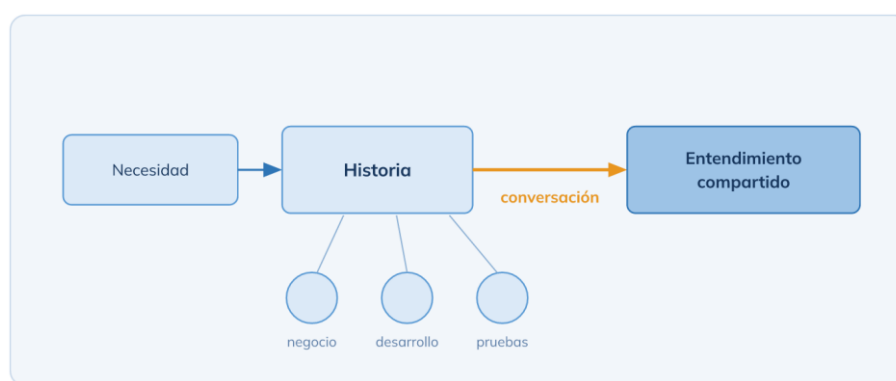


State of the art

Historias de Usuario

Mapa de referencia del conocimiento profesional vigente



Actualizado a septiembre de 2026

Sobre este documento

Este documento es un mapa curado de los conocimientos, prácticas y modelos empleados profesionalmente a la fecha de su publicación para trabajar con historias de usuario y con los demás elementos del Product Backlog, desde escribirlas cuando aportan y conversarlas hasta juzgar su calidad, dividir las sin perder valor, mantener el contexto del conjunto y verificar que el resultado responde a la intención, también cuando quien redacta o quien implementa es una inteligencia artificial. Para cada uno, este mapa lo sitúa, indica su grado de adopción en la práctica profesional actual y orienta sobre dónde encontrar información de referencia.

Úsalo como observatorio y punto de referencia para contrastar si el conocimiento profesional que empleas está alineado con la práctica actual y en vanguardia.

Se complementa con un documento de desarrollo que profundiza en cada concepto y con la plataforma de entrenamiento y evaluación en Skill Arena. En el área [Historias de Usuario](#) puedes contrastar tu nivel de conocimiento y, si lo superas, obtener un diploma que acredita curricularmente la solvencia y vanguardia profesional en esta área.



Estado del conocimiento

El conocimiento sobre las historias de usuario está asentado en su canon y se mueve muy deprisa en su frontera: las piezas que lo fijan tienen entre veinte y veinticinco años y siguen vigentes sin revisiones mayores, mientras una franja de práctica gana adopción sin ser universal (medir la calidad con criterios explícitos, el mapa de ejemplos, dimensionar por medida empírica, preservar el contexto de las conversaciones) y la capa de inteligencia artificial y agentes cambia en cuestión de meses. En los distintos apartados del documento, las etiquetas (ESTABLECIDO, EN CONSOLIDACIÓN, EMERGENTE) ayudan a identificar la madurez de cada concepto:

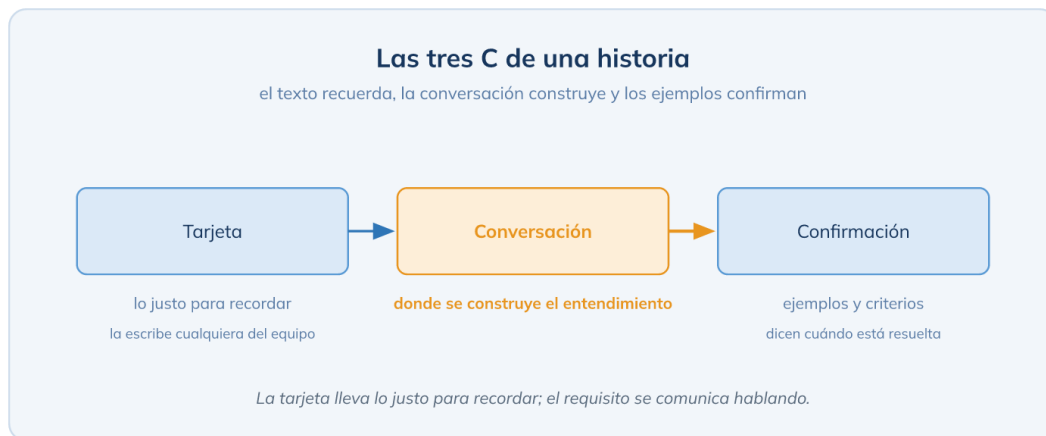
ESTABLECIDO consenso asentado; conocimiento que se da por necesario.

EN CONSOLIDACIÓN gana adopción con rapidez; aún no universal pero ya relevante.

EMERGENTE frontera reciente; alta relevancia y alta volatilidad.

Bloque A — La historia de usuario: qué es y qué lugar ocupa

Una historia es el mecanismo con el que un equipo construye entendimiento compartido sobre una porción pequeña de valor, y no un formato de requisito.



La historia como promesa de conversación: origen, las tres C y la plantilla · ESTABLECIDO

Una historia de usuario describe brevemente una necesidad desde la perspectiva de quien la tiene, y su valor está en la conversación que promete más que en el texto que la enuncia. Se sostiene en tres partes que Ron Jeffries formuló en 2001, la tarjeta con lo justo para recordar, la conversación donde se construye el entendimiento y la confirmación en forma de ejemplos o pruebas de aceptación. La plantilla «Como..., quiero..., para...» nació en Connextra para forzar el quién y el porqué, y funciona como ruedas de aprendizaje que conviene saber quitar.

Por qué está aquí ahora. El canon no se ha movido en veinticinco años y sigue explicando los fallos de hoy. Jeffries dejó dicho en 2008 que una historia no es un documento de requisitos, el glosario de Agile Alliance califica la plantilla de «training wheels» y avisa de que dedicarle esfuerzo no tiene mucho sentido, y Mike Cohn insiste en que lo escrito es un recordatorio y el requisito real emerge en la conversación, los ejemplos y las decisiones. El BoK de Scrum Manager recoge la definición, las tres C y la plantilla con el léxico en español.

Dónde mirar. [Jeffries — Card, Conversation, Confirmation](#) · [Agile Alliance — User story template](#) · [Scrum Manager BoK — Historia de usuario](#)

Qué dicen los marcos: de la neutralidad de Scrum a la institucionalización de SAFe · ESTABLECIDO

La Guía de Scrum no menciona las historias de usuario y solo pide que los elementos del Product Backlog tengan descripción, orden y tamaño, mientras SAFe y Disciplined Agile las formalizan con INVEST y añaden las historias habilitadoras (enabler stories). Kanban es neutral en cuanto al formato, y el BABOK ágil y el material de IREB las tratan como una técnica de requisitos entre varias. Conviene saber qué exige cada marco para no defender como norma lo que es una elección del equipo.

Por qué está aquí ahora. Las posiciones están escritas y son estables. Scrum.org lo enuncia sin rodeos desde 2017, que Scrum no las prescribe ni las requiere; LeSS dice que el Product Backlog no contiene historias, sino elementos; SAFe mantiene su artículo de referencia con los cuatro tipos de

habilitador. El debate sobre el propio término sigue vivo, con propuestas de sustituirlo por «elemento del backlog» para acabar con la confusión entre técnica y obligación.

Dónde mirar. [Guía de Scrum 2020](#) · [Scrum.org — Mito 4: el backlog no tiene que ser historias](#) · [SAFe — Enablers](#)

El entendimiento compartido como criterio de cuánto escribir · ESTABLECIDO

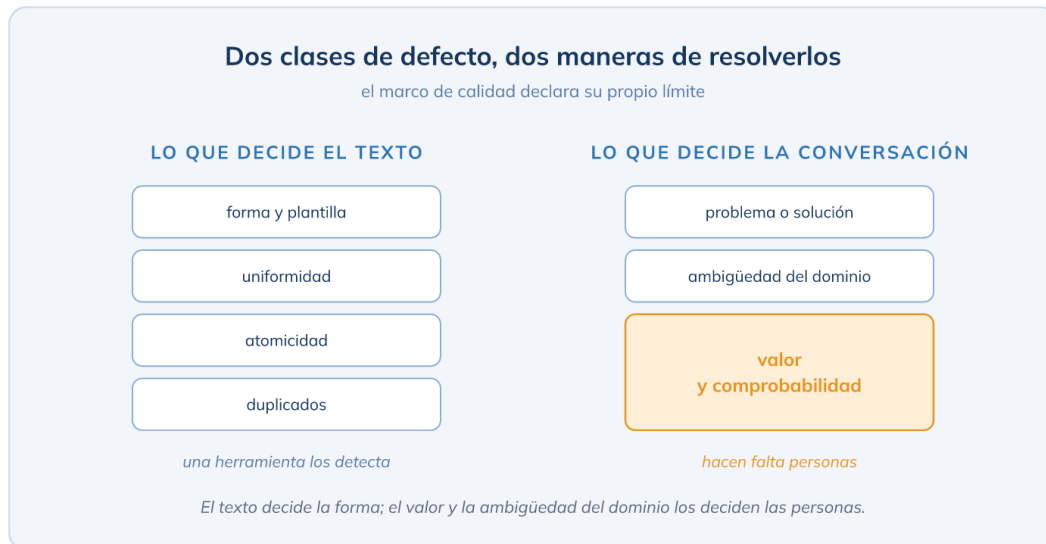
El entendimiento compartido tiene una parte explícita, escrita y verificable, y otra implícita que vive en las cabezas de quienes trabajan juntos, y puede ser verdadero o solo aparente. Ese es el criterio que decide cuánto conviene escribir en cada caso, porque escribir de más no crea entendimiento y escribir de menos deja el trabajo a merced de la interpretación. Es el hilo que recorre toda la skill, de la tarjeta a la verificación del resultado.

Por qué está aquí ahora. El modelo de Glinz y Fricker da nombre a lo que la práctica ágil venía haciendo por intuición, y explica el fracaso más común de las historias, el falso acuerdo que nadie detecta hasta la revisión. Adzic lo resume en una consigna que ordena el oficio, contar historias en lugar de escribirlas, y Patton recuerda que compartir documentos no es compartir entendimiento. Con la inteligencia artificial el criterio gana peso, porque producir texto ha dejado de costar y el entendimiento sigue costando lo mismo.

Dónde mirar. [Glinz — Requirements Engineering, reflexión personal \(con el modelo de 2015\)](#) · [Adzic y Evans — Fifty Quick Ideas to Improve Your User Stories](#) · [Fowler — UserStory](#)

Bloque B — La calidad de una historia y del conjunto

La calidad de una historia se juzga por si sostiene una conversación productiva, y la del backlog por si el conjunto sigue siendo manejable.



INVEST y su relectura: guía de trabajo, no lista de comprobación · ESTABLECIDO

INVEST nombra seis propiedades deseables de una historia (independiente, negociable, valiosa, estimable, pequeña y comprobable) y funciona como guía de diagnóstico cuando una historia se resiste. Sus propiedades tiran en direcciones opuestas, porque la independencia y el valor empujan hacia historias grandes y la estimabilidad y el tamaño hacia pequeñas, de modo que el peso de cada una cambia según lo cerca que esté la historia de construirse. Su autor la ha revisado en público, con una lectura escalable de «pequeña» y una propiedad añadida de extremo a extremo.

Por qué está aquí ahora. Es el criterio de calidad más citado del oficio y también el más mal usado. La encuesta de REFSQ de 2016 encontró que solo uno de cada cuatro equipos lo aplica y concluyó que es una guía de trabajo y no una lista de verificación, útil sobre todo en equipos que empiezan. El propio Wake publicó en 2021 su matización, y el BoK de Scrum Manager la recoge como guía conversacional y no como formulario de calidad.

Dónde mirar. [Wake — INVEST in Good Stories, and SMART Tasks](#) · [Wake — All You Need is INVEST? No!](#) · [Scrum Manager BoK — INVEST](#)

Defectos y ambigüedad: lo que decide el texto y lo que decide la conversación · EN CONSOLIDACIÓN

El marco Quality User Story ordena trece criterios de calidad en tres categorías, sintácticos, semánticos y pragmáticos, y declara su propio límite, porque deja fuera el valor y la comprobabilidad al no poder decidirse leyendo el texto. Esa frontera reparte el trabajo, ya que la forma, la uniformidad y la atomicidad las detecta una herramienta o un modelo de lenguaje, mientras confundir problema con solución o resolver una ambigüedad de dominio exige a las personas. La ambigüedad tiene además una cara útil, porque señala dónde hay conocimiento tácito que conviene aflorar en lugar de eliminarlo por escrito.

Por qué está aquí ahora. El marco tiene detrás una medición poco común en este terreno, más de mil historias de dieciocho organizaciones con un 56 % que presentaba al menos un defecto, y los más frecuentes eran precisamente los mecánicos. La lista de palabras y construcciones de riesgo está documentada desde el trabajo sobre fuentes lingüísticas de ambigüedad y los «requirements smells». Sigue en consolidación porque es riguroso y con herramienta, pero poco conocido fuera del ámbito académico.

Dónde mirar. [Lucassen y otros — Quality User Story framework \(REJ 2016\)](#) · [Femmer y otros — Requirements Smells](#) · [Ferrari y otros — Ambigüedad y conocimiento tácito](#)

El catálogo de antipatrones de la historia · ESTABLECIDO

Los modos de fallo de una historia tienen nombre y remedio conocidos, desde la historia que en realidad es una tarea técnica y el usuario genérico que no informa de nada, hasta la solución disfrazada de necesidad, la épica que se presenta como historia, la fragmentación que deja piezas sin valor y la historia tratada como contrato. Reconocer el patrón ahorra la discusión, porque cada uno pide una acción distinta, dividir, fusionar, reescribir, reclasificar o descartar.

Por qué está aquí ahora. Es consenso practicante con nombres estables, publicado por Pichler, Cohn y el catálogo de antipatrones del Product Backlog de Wolpers, y el BoK de Scrum Manager señala el más frecuente en español, escribir tareas técnicas en lugar de valor. La lista cambia poco de un año a otro, lo que la hace útil como instrumento de diagnóstico rápido en el refinamiento.

Dónde mirar. [Pichler — Why User Stories Fail](#) · [Pichler — 10 Tips for Writing Good User Stories](#) · [Humanizing Work — ¿Todo el backlog tiene que ser historias?](#)

Salud del backlog y la Definition of Ready · ESTABLECIDO

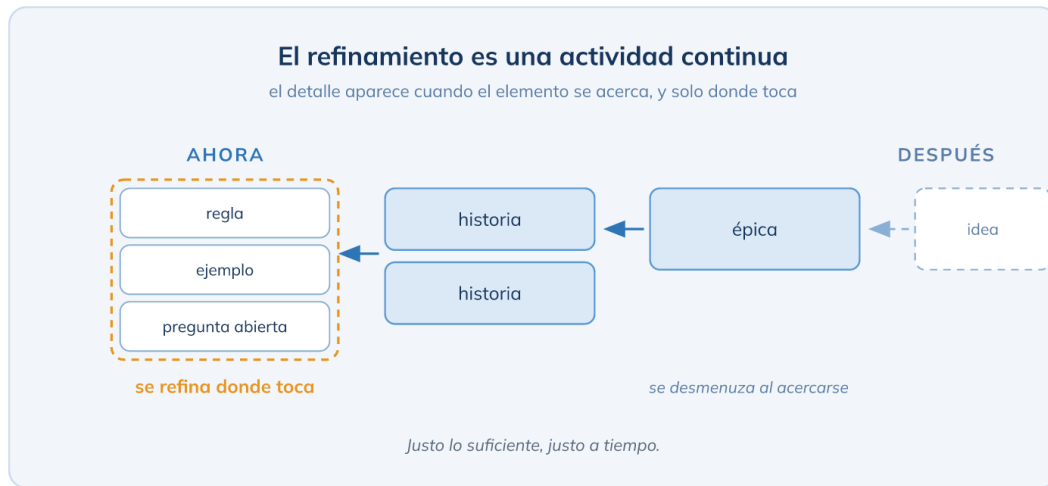
Un backlog sano cumple criterios que no se ven en una historia aislada, como la ausencia de duplicados y de conflictos, la uniformidad de formato, el detalle en proporción a la cercanía y la visibilidad de las dependencias. A eso se añade la higiene del conjunto, la edad de los elementos, la caducidad de lo que ya no responde al objetivo y la decisión de declarar en bancarrota un backlog que nadie puede leer. La Definition of Ready ayuda mientras funciona como guía de conversación, y estorba cuando se convierte en una puerta que el equipo usa para devolver trabajo.

Por qué está aquí ahora. El acrónimo DEEP de Pichler es la formulación más difundida de esa salud, y la Guía de Scrum no contiene la Definition of Ready y habla de elementos que se consideran listos para su selección. Cohn ha publicado por qué le parece peligrosa y Scrum.org ha explicado varias veces por qué no está en la guía, mientras el BoK de Scrum Manager la documenta como práctica con su versión para trabajo asistido por inteligencia artificial.

Dónde mirar. [Pichler — Make the Product Backlog DEEP](#) · [Cohn — Definition of Ready: por qué es peligrosa](#) · [Scrum Manager BoK — A punto](#)

Bloque C — La conversación y su confirmación

La historia se completa hablando y se cierra con ejemplos: quién conversa, en qué formato, qué queda escrito como criterio y cómo se vuelve verificable.



Refinamiento continuo: ni traspaso ni reunión · ESTABLECIDO

El refinamiento es la actividad de desmenuzar y precisar los elementos del backlog, y funciona cuando ocurre de forma continua, en grupos pequeños, con horizonte corto y con el detalle justo a tiempo. Sus dos degeneraciones son conocidas, el traspaso en el que alguien refina solo y entrega tickets, y la reunión periódica que se convierte en entrada de datos. La inteligencia artificial ha añadido una versión nueva del traspaso, en la que producto genera con un modelo y desarrollo interpreta con otro sin que nadie haya conversado.

Por qué está aquí ahora. La Guía de Scrum define el refinamiento como acto y no como evento, y Scrum.org lleva años desmontando la idea de que sea una reunión obligatoria. El BoK de Scrum Manager fija la referencia en español, con el porcentaje de capacidad que suele dedicarse y el rechazo del término «grooming». Lo que la investigación sobre entrevistas de requisitos ha aportado es un matiz importante, que la conversación detecta dónde difieren las interpretaciones más de lo que extrae conocimiento tácito.

Dónde mirar. [Guía de Scrum 2020](#) · [Scrum Manager BoK — Mantenimiento de la pila del producto](#) · [Scrum.org — ¿Qué es una historia de usuario?](#)

Quién conversa y en qué formato: Three Amigos y talleres de escritura · ESTABLECIDO

La conversación productiva necesita tres perspectivas presentes, negocio, desarrollo y pruebas, a las que se suman diseño u operaciones cuando el trabajo lo pide. Eso es lo que nombra Three Amigos, tres puntos de vista y no tres personas, con la función de hacer chocar interpretaciones distintas antes de construir. El taller de escritura de historias ordena ese trabajo en grupo, y la escala de complejidad de Liz Keogh decide cuándo la incertidumbre es tan alta que lo que procede es un spike o un experimento.

Por qué está aquí ahora. El glosario de Agile Alliance fija las tres perspectivas y los alias con que circula la práctica, y advierte de sus dos deformaciones, limitarla a tres personas o convertirla en reunión formal recurrente. La escala de Keogh, de «todos sabemos hacerlo» a «nadie lo ha hecho

nunca», sigue siendo la herramienta más simple para decidir si una historia está lista para conversarse o pide investigación previa.

Dónde mirar. [Agile Alliance — Three Amigos](#) · [Keogh — Estimating Complexity](#) · [Cohn — User Stories \(guía de referencia\)](#)

Criterios de aceptación: formatos, límites y confusiones · ESTABLECIDO

Los criterios de aceptación son las condiciones observables que dirán que la necesidad está resuelta, y admiten varios formatos según lo que convenga, lista de condiciones, escenarios «Dado que / Cuando / Entonces», reglas con ejemplos o tablas. No forman parte de Scrum, mientras la Definition of Done sí, y mezclarlos produce dos problemas a la vez, criterios que repiten la calidad general y una Definition of Done que cambia con cada historia. Cuando el comportamiento es probabilístico, la confirmación deja de ser un criterio binario y pasa a ser un umbral sobre un conjunto de evaluación.

Por qué está aquí ahora. Los formatos están fijados y documentados, incluido el patrón «Dado que / Cuando / Entonces» en el glosario de Agile Alliance, y el BoK de Scrum Manager mantiene la página de referencia en español con su sección de inteligencia artificial. La separación entre criterios y Definition of Done es hoy uno de los asuntos más tratados en la comunidad Scrum, señal de que sigue confundiendo en la práctica.

Dónde mirar. [Agile Alliance — Given-When-Then](#) · [Scrum Manager BoK — Criterios de aceptación](#) · [Cohn — Requisitos no funcionales como historias](#)

Reglas, ejemplos y Example Mapping · EN CONSOLIDACIÓN

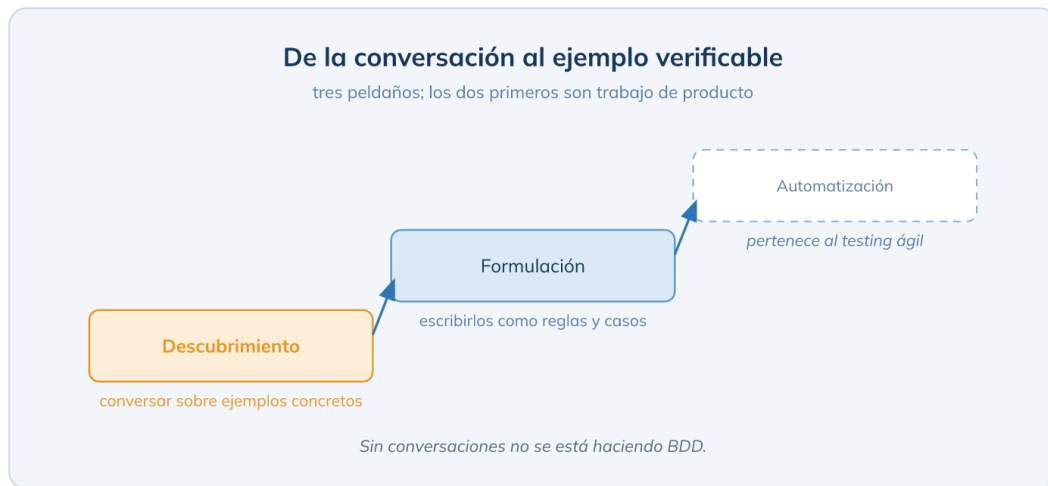
Concretar una historia consiste en encontrar sus reglas de negocio y darles ejemplos, incluidos los casos límite y los negativos, con ejemplos que ilustren en lugar de agotar. Example Mapping ordena ese trabajo en una sesión corta con cuatro tipos de tarjeta, la historia, sus reglas, los ejemplos de cada regla y las preguntas que nadie sabe responder. El mapa resultante se lee como una señal, porque muchas preguntas indican que la historia no está lista, muchas reglas que hay que partirla y una regla con demasiados ejemplos que en realidad son varias reglas.

Por qué está aquí ahora. La técnica tiene autor, protocolo y duración declarada, unos veinticinco minutos para una historia bien entendida, y prescribe expresamente no escribir Gherkin en la sesión. Gana adopción en la comunidad de BDD y de pruebas ágiles, con variantes como Feature Mapping, aunque sigue siendo minoritaria en los equipos de producto, que es exactamente el hueco que esta skill cubre.

Dónde mirar. [Wynne — Introducing Example Mapping](#) · [Smart — Feature Mapping](#) · [Adzic — Specification by Example \(recursos\)](#)

De los ejemplos a la verificabilidad: BDD sin herramienta · ESTABLECIDO

El desarrollo guiado por el comportamiento se ordena en tres peldaños, descubrimiento, formulación y automatización, y los dos primeros pertenecen al trabajo de producto. Es una manera de conversar sobre ejemplos concretos, no una herramienta ni una técnica de pruebas, y produce como efecto lateral una documentación que sigue siendo verdad porque se ejecuta. La automatización pertenece al testing ágil y aquí solo se enuncia, con la advertencia de que empezar por ella es el error más repetido.



Por qué está aquí ahora. Sus autores lo dicen con claridad, Keogh que sin conversaciones no se está haciendo BDD en absoluto y Hellesøy que BDD no es automatización de pruebas, y el BoK de Scrum Manager mantiene la página con esa lectura. El estado es establecido en el discurso y en las fuentes, aunque la práctica sin automatización sigue siendo minoritaria, y ese contraste conviene tenerlo presente al implantarlo.

Dónde mirar. [North — What's in a Story?](#) · [Keogh — A Third Conversational Pattern in BDD](#) · [Scrum Manager BoK — Behavior-Driven Development](#)

Preservar el contexto de la conversación · EN CONSOLIDACIÓN

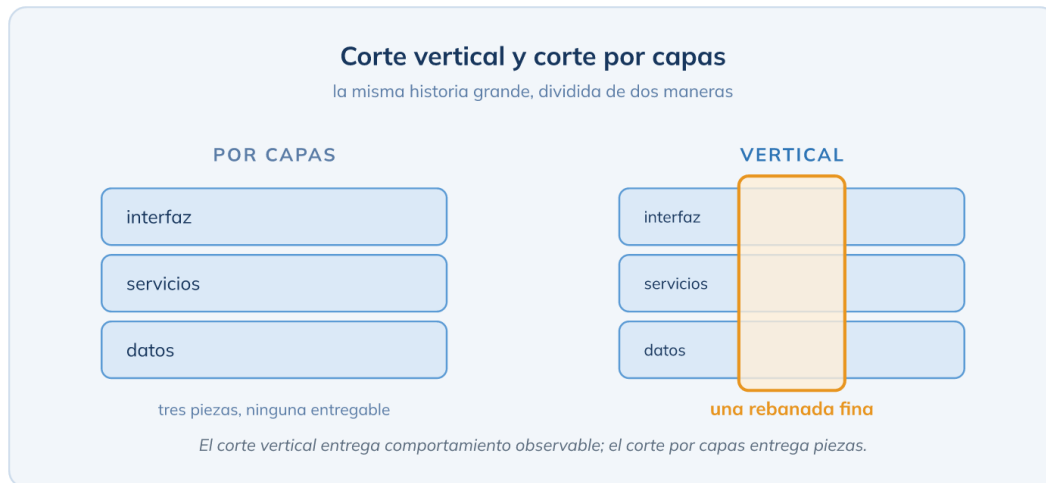
De una conversación de refinamiento merece la pena conservar cuatro cosas, las reglas acordadas, los ejemplos que las fijan, las preguntas abiertas con su responsable y las decisiones con su procedencia. La tarjeta se puede tirar cuando la historia se entrega, y el conocimiento que produjo, no. Scrum Manager propone llevar al terreno del producto el registro de decisiones que la arquitectura de software usa desde hace más de una década, con media página por decisión relevante.

Por qué está aquí ahora. La práctica está extendida en piezas sueltas, en la documentación viva que deja la especificación con ejemplos y en el propio mapa de ejemplos, pero no tiene nombre canónico ni formato acordado, y por eso la ficha figura en consolidación. El registro de decisiones de producto es propuesta de la casa por analogía con los registros de arquitectura, y el material lo dice al presentarlo. Con asistencia de inteligencia artificial el problema se agrava, porque buena parte del contexto pasa a vivir en conversaciones de chat que el equipo no ve.

Dónde mirar. [Adzic — Specification by Example y documentación viva](#) · [Wynne — el mapa de ejemplos como registro de la conversación](#) · [Skill Arena — Product Owner](#)

Bloque D — Dividir manteniendo el valor

La división de historias (story splitting) es la habilidad más pedida y la menos enseñada: por qué se divide, con qué patrón, con qué procedimiento y cómo se juzga el corte.



Por qué, cuánto y cuándo dividir · ESTABLECIDO

Trabajar en lotes pequeños acorta el ciclo de aprendizaje, reduce el riesgo de cada entrega y permite descartar lo que no aporta, y esa es la razón de fondo para dividir. El tamaño recomendable ha bajado durante veinte años, y la banda de heurísticas conviene sustituirla por la medida empírica del equipo, dimensionando cada elemento contra su expectativa de servicio (right-sizing) en lugar de igualar tamaños. Épica, tema y funcionalidad son contenedores por tamaño y no categorías rígidas, y el momento de dividir llega cuando la historia se acerca a construirse, no cuando entra en el backlog.

Por qué está aquí ahora. El lote pequeño es una de las capacidades que DORA sostiene con datos año tras año, y la escala de referencia ha cambiado tanto que INVEST hablaba en 2003 de unas pocas semanas-persona por historia mientras SAFe habla hoy de pocos días. En la comunidad Kanban el dimensionado empírico se ha separado de igualar tamaños, matiz que evita convertir la división en un ritual de estimación.

Dónde mirar. [DORA — Working in small batches](#) · [ProKanban — Right-sizing no es igualar tamaños](#) · [Cohn — Épicas, funcionalidades y temas](#)

Corte vertical y cortes técnicos legítimos · ESTABLECIDO

Un corte vertical entrega un cambio de comportamiento observable y atraviesa las capas que haga falta; el corte por capas produce piezas que nadie puede desplegar por separado. Hay tres cortes técnicos legítimos y conviene conocer sus límites, el esqueleto andante (walking skeleton) que integra los componentes con lo mínimo, la historia habilitadora que prepara el terreno y el spike que compra información. Queda abierto el debate sobre por dónde empezar, con interfaz primero y datos simulados o con servicios primero tras un conmutador de funcionalidad.

Por qué está aquí ahora. El esqueleto andante y la bala trazadora llevan dos décadas documentados, y Adzic añadió el correctivo práctico de ponerle muletas para que la primera rebanada sea útil y no solo demostrable. Cohn incluye cortar por fronteras técnicas entre los cinco

errores de división más frecuentes, y Humanizing Work publica el criterio que distingue la historia técnica honesta de la tarea disfrazada.

Dónde mirar. [Adzic — Forget the walking skeleton, put it on crutches](#) · [Hunt y Thomas — Tracer bullets and prototypes](#) · [Humanizing Work — Dealing with technical stories](#)

El catálogo de patrones de división · ESTABLECIDO

Los patrones de división son un catálogo de trece movimientos con procedencia conocida, entre ellos los pasos del flujo con una rebanada fina que lo atravesase, las variaciones de reglas de negocio, las variaciones de datos, los caminos e interfaces, lo simple antes que lo complejo, el esfuerzo mayor aislado, las cualidades del sistema aplazadas, el segmento de usuarios y aprender antes que ganar. Cada patrón responde a una señal distinta en la historia, y reconocer la señal es lo que convierte el catálogo en un instrumento y no en una lista que memorizar.

Por qué está aquí ahora. Las fuentes son estables y públicas, de las veinte formas de Wake en 2005 a los nueve patrones de Humanizing Work y el acrónimo SPIDR de Cohn, que agrupa cinco de los más productivos. El catálogo unificado es síntesis editorial sobre patrones establecidos, y Agile Alliance advierte de que hay poco consenso sobre cómo dividir, motivo para enseñar el criterio de elección junto a los patrones.

Dónde mirar. [Humanizing Work — Guide to Splitting User Stories](#) · [Wake — Twenty Ways to Split Stories](#) · [Cohn — SPIDR](#)

Del patrón al corte: procedimiento, elección y dependencias · EN CONSOLIDACIÓN

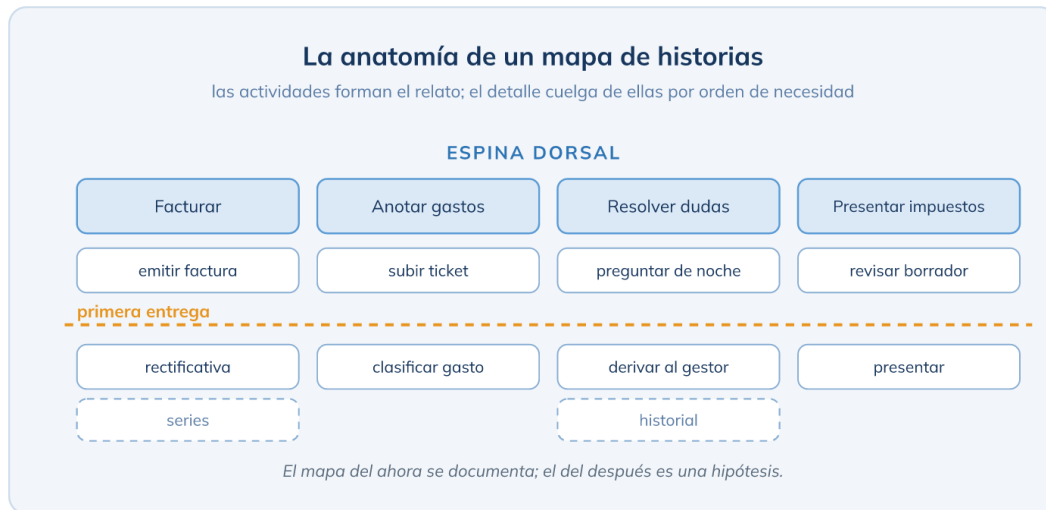
El procedimiento empieza antes de aplicar ningún patrón, comprobando que la historia cumple INVEST salvo el tamaño, porque lo que suele faltar es el valor y entonces procede recombinar en lugar de partir. Los cortes candidatos se juzgan con criterios explícitos, conservar valor, ser verificables, permitir despriorizar algo y dar piezas parecidas, y una prueba descarta los falsos cortes, comprobar si todas las piezas son obligatorias para completar la grande. Cada corte introduce dependencias, que piden hacerse visibles antes de gestionarse.

Por qué está aquí ahora. El procedimiento explícito, el método de la hamburguesa y la práctica deliberada con Elephant Carpaccio pertenecen a una escuela concreta más que al consenso general, y de ahí el estado. Humanizing Work publica su precondition y el dato de que la fluidez se alcanza en unas tres horas de práctica.

Dónde mirar. [Humanizing Work — ¿Historias indivisibles?](#) · [Adzic — El método de la hamburguesa](#) · [Cockburn — Elephant Carpaccio](#)

Bloque E — El mapa de historias y el contexto del backlog

Una lista ordenada pierde el relato; el mapa de historias (user story mapping) devuelve el contexto y permite decidir qué conjunto de historias forma una entrega con sentido.



Lo que se pierde en un backlog plano · EN CONSOLIDACIÓN

Una lista ordenada por prioridad esconde el relato de lo que hace un usuario, las relaciones entre elementos y la posibilidad de ver qué conjunto forma una entrega con sentido. Las alternativas propuestas son varias y ninguna se ha impuesto, desde las jerarquías de la herramienta y el tablero de backlog hasta el árbol de oportunidades y el mapa de impacto. La defensa de la lista plana con etiquetas sigue siendo un argumento serio, porque una jerarquía mal usada ordena carpetas en lugar de trabajo.

Por qué está aquí ahora. El diagnóstico es compartido desde que Patton lo formuló en 2008, y sigue publicándose con datos nuevos, con artículos recientes sobre por qué la jerarquía del backlog está rota y con la pregunta explícita de si debería ser plano. Que el problema esté claro y la solución no lo esté es lo que mantiene la ficha en consolidación.

Dónde mirar. [Patton — The new user story backlog is a map](#) · [Bastow — Por qué la jerarquía del backlog está rota](#) · [Hinshelwood — ¿Debería tu backlog ser plano?](#)

El mapa de historias: anatomía y construcción · ESTABLECIDO

El mapa coloca a los usuarios y sus actividades en un eje narrativo, la espina dorsal (backbone), y cuelga de él las tareas y el detalle en orden de necesidad. Se construye en cinco pasos, encuadrar, mapear el panorama en amplitud antes que en profundidad, explorar, cortar entregas viables y cortar la estrategia de desarrollo, con la regla de refinar solo donde toca. Distinguir el mapa del ahora, documentado con datos, del mapa del después, que es una hipótesis hasta que se libera, evita el error de tratar el segundo como un plan.

Por qué está aquí ahora. La técnica tiene una fuente única y estable, con la terminología literal de Patton fijada desde 2008 y su libro de 2014 sin ediciones posteriores, más su referencia rápida y su lista de los cinco errores más frecuentes. Aparece ya en los objetivos de aprendizaje de

certificaciones de producto, y el BoK de Scrum Manager la recoge, aunque con una estructura que conviene contrastar con la canónica.

Dónde mirar. [Patton — Story Map Concepts \(referencia rápida\)](#) · [Patton — Five common story mapping mistakes](#) · [Scrum Manager BoK — Mapa de historias](#)

Cortar por resultado y mantener el mapa vivo · EN CONSOLIDACIÓN

Del mapa se cortan tres cosas distintas, una entrega con su resultado esperado y su métrica, un experimento o producto mínimo viable, y la estrategia de desarrollo que empieza por un esqueleto que atraviesa el sistema. El corte hecho por capacidad del equipo se reconoce porque no puede enunciar qué cambia para nadie cuando se entregue. El mapa se mantiene con lo aprendido, y esa actualización es lo que lo distingue de un mural que decora la pared después del taller.

Por qué está aquí ahora. Patton define el producto mínimo viable como el producto más pequeño que alcanza los resultados buscados, lo que traslada el criterio del alcance al resultado, y su material vincula cada rebanada a un resultado observable. Se predica más de lo que se practica, y las cifras de adopción disponibles son secundarias y antiguas, de modo que la ficha se apoya en las fuentes primarias y no en encuestas.

Dónde mirar. [Patton — It's All in How You Slice It](#) · [Kniberg — Making sense of MVP](#) · [Patton — Dual Track Development is not Duel Track](#)

Mapas vecinos y el mapa dentro de la herramienta · EN CONSOLIDACIÓN

Cuatro técnicas se confunden con frecuencia y responden a preguntas distintas, el mapa de historias para el relato de uso, el mapa del recorrido del usuario para la experiencia y sus emociones, el mapa de impacto para conectar objetivo y comportamientos, y el árbol de oportunidades para elegir dónde intervenir. Se encadenan bien, porque el descubrimiento alimenta el mapa y el mapa alimenta el taller de historias y sus criterios. Al llevar el mapa a la herramienta aparecen tres modos de fallo, la jerarquía que sustituye a los dos ejes, el mapa gigante que nadie lee y el mapa físico que se abandona tras el taller.

Por qué está aquí ahora. Cada técnica está establecida por separado y bien documentada, mientras la guía de elección es síntesis editorial, y así se presenta. La relación con la construcción del backlog en una inception está publicada y enlaza con la skill de Agile Inception, y los modos de fallo son consenso practicante que las propias herramientas recogen ya en su documentación.

Dónde mirar. [Adzic — Impact Mapping](#) · [Nielsen Norman Group — Journey Mapping 101](#) · [Torres — Opportunity Solution Trees](#)

Bloque F — No todo es una historia

El Product Backlog admite muchos tipos de elemento, y forzar la plantilla donde no encaja produce usuarios falsos y conversaciones equivocadas.



Elemento del Product Backlog frente a historia · ESTABLECIDO

Scrum solo pide que un elemento del Product Backlog tenga descripción, orden y tamaño, y la enumeración de tipos que aparecía en versiones anteriores de su guía desapareció en 2020. En un backlog real conviven defectos, spikes con su pregunta y su límite de tiempo, trabajo técnico, restricciones y requisitos no funcionales, cumplimiento normativo, e hipótesis con su resultado esperado. Cada tipo se confirma de una manera distinta, y por eso cada uno pide una conversación distinta.

Por qué está aquí ahora. La Guía de Scrum es explícita en lo que exige y en lo que dejó de enumerar, LeSS insiste en que el backlog contiene elementos y no historias, y la guía troncal Product Owner 2.0 de Scrum Manager lo enuncia en la misma línea, que un elemento del backlog no tiene por qué ser una historia de usuario. El BoK completa el mapa en español con las páginas de la pila del producto y de los requisitos ágiles.

Dónde mirar. [Guía de Scrum 2020](#) · [Scrum Manager BoK — Pila del producto](#) · [Skill Arena — Product Owner](#)

Trabajo técnico sin usuario falso · ESTABLECIDO

Cuando el trabajo no tiene un usuario visible hay tres salidas honestas, encontrar al beneficiario real y su valor económico, enunciarlo en claro como elemento técnico o habilitador sin forzar la plantilla, o convertirlo en una restricción transversal o en un criterio de la Definition of Done. La deshonestas es conocida y produce ficciones que no comunican nada, del tipo «como servidor, quiero...». Distinguir la historia técnica honesta de la tarea disfrazada es una de las decisiones más frecuentes del refinamiento, con un matiz para los equipos de plataforma, donde quien desarrolla es un cliente real.

Por qué está aquí ahora. Humanizing Work publica el criterio con ejemplos, SAFe institucionaliza los habilitadores en cuatro tipos y Cohn propone la sintaxis de funcionalidad de FDD para cuando no hay

usuarios cerca. El BoK de Scrum Manager señala como error frecuente escribir tareas técnicas en lugar de valor, y a la vez legitima la historia técnica, lo que obliga a enseñar la distinción con precisión.

Dónde mirar. [Humanizing Work — Dealing with technical stories](#) · [SAFe — Enablers](#) · [Cohn — No todo tiene que ser una historia: funcionalidades de FDD](#)

Representaciones alternativas y cómo elegir el formato · EN CONSOLIDACIÓN

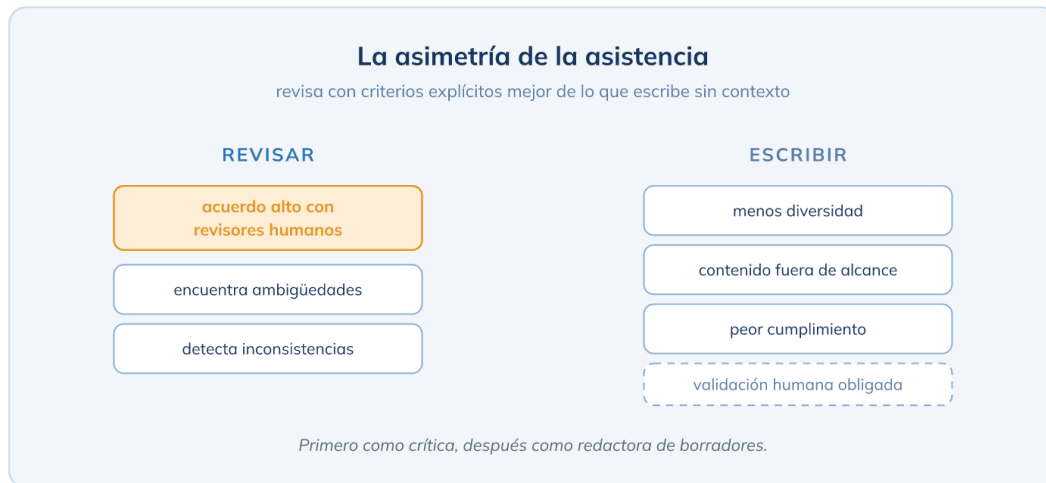
Hay formatos con autor y uso conocido para lo que la historia no expresa bien, la job story cuando manda la situación y no el perfil, el caso de uso y su rebanada para flujos con muchas alternativas o errores costosos, la funcionalidad de FDD para componentes sin usuario cercano, la hipótesis con su resultado para lo que hay que validar, la notación EARS para requisitos que va a leer una máquina, y el pitch o el documento de diseño para decisiones que necesitan argumento. Cinco preguntas bastan para elegir, quién se beneficia, qué se ignora, qué conversación hace falta, quién lo va a leer y qué cuesta equivocarse.

Por qué está aquí ahora. Cada formato está establecido y publicado por su autor, y lo que falta en el oficio es el criterio para elegir entre ellos, que aquí se presenta como guía de decisión y no como método nuevo. El manual básico de Scrum Manager al que esta skill toma el relevo ya ofrecía una tabla de contexto y técnica apropiada, y esta ficha la recoge ampliada con los formatos que han aparecido después.

Dónde mirar. [Klement — Replacing the user story with the job story](#) · [Jacobson, Spence y Kerr — Use-Case 2.0](#) · [Singer — Shape Up: escribir el pitch](#)

Bloque G — La inteligencia artificial en el trabajo con historias

La inteligencia artificial abarata escribir, y mueve el valor profesional al contexto que se le da, a la crítica de lo que devuelve y a la verificación de lo que se construye.



La asimetría: la inteligencia artificial revisa mejor de lo que escribe · EN CONSOLIDACIÓN

Como evaluadora de historias, con criterios explícitos, la asistencia alcanza un acuerdo alto con revisores humanos y encuentra ambigüedades e inconsistencias que se escapan a las reglas. Como autora rinde peor de lo que aparenta, porque produce historias de calidad superficial comparable pero menos diversas, con más contenido fuera de alcance y con peor cumplimiento de los criterios de calidad. De ahí se deriva un orden de uso, primero como crítica y después como redactora de borradores, variantes, criterios y cortes candidatos.

Por qué está aquí ahora. Es el hallazgo más consistente de la investigación reciente, replicado en varios estudios entre 2023 y 2026, y el que más contradice la intuición de quien empieza a usarla para generar backlog. Scrum Manager ya lo sostiene en su propio material, en el BoK y en la skill de Agile Inception, y en el mercado han aparecido herramientas que asumen ese papel de crítica, como el revisor de historias de Mountain Goat.

Dónde mirar. [Quattrocchi y otros — ¿Pueden los LLM generar historias y evaluar su calidad?](#) · [Ronanki y otros — ChatGPT en la evaluación de calidad de historias](#) · [Scrum Manager BoK — Historia de usuario](#)

Contexto, invención y procedencia · EMERGENTE

Sin evidencia y sin estructura, lo que devuelve un modelo es genérico, y el contexto que de verdad cambia el resultado es el del descubrimiento, la épica, las personas reales, las reglas conocidas y las decisiones ya tomadas, con el límite de no pegar datos de clientes ni notas crudas de entrevistas. Lo que se inventa admite catálogo, usuarios y necesidades sintéticas, reglas plausibles pero falsas, criterios sin evidencia y falso consenso, y se detecta barato exigiendo la fuente de cada criterio y marcando los supuestos. Conviene distinguir en la propia historia lo aportado, lo inferido y lo generado, y quién lo validó.

Por qué está aquí ahora. El estudio de campo con Product Owners que ya trabajan así apunta al mismo problema, que la salida es demasiado genérica porque falta el contexto y no cita el material

de origen. Las revisiones sistemáticas confirman que la mayor parte de la investigación está en fase temprana y documentan la frecuencia de la invención, y el artículo 50 del Reglamento europeo de inteligencia artificial, aplicable desde agosto de 2026, hace de la revisión humana con responsable identificable algo más que una buena práctica.

Dónde mirar. [Steghöfer — Faster than the Team, Faster than the Customer](#) · [Nielsen Norman Group — Usuarios sintéticos](#) · [Reglamento europeo de IA, artículo 50](#)

Riesgos y el patrón de trabajo que no delega el juicio · EMERGENTE

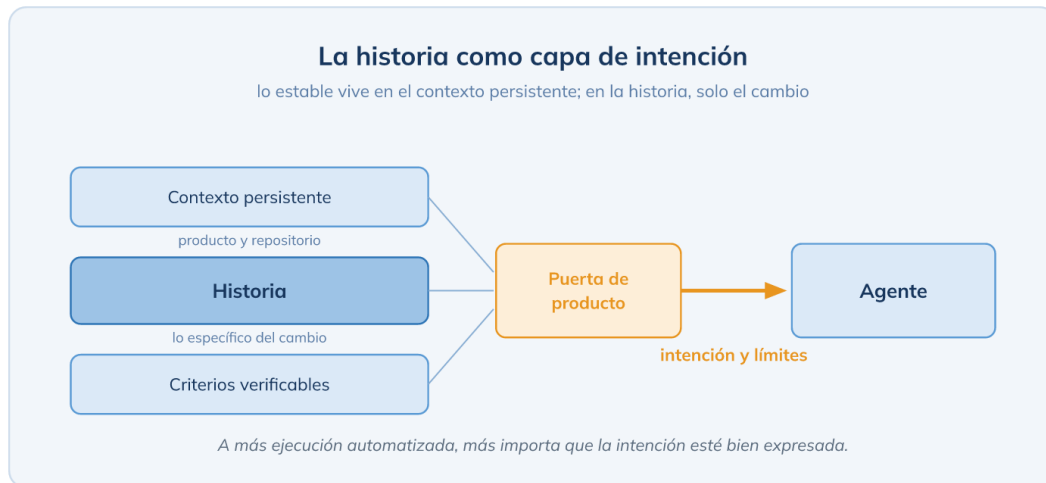
Los riesgos documentados son cinco y se refuerzan entre sí, la invención con apariencia de solvencia, el sesgo de precisión que concede autoridad a lo bien acabado, la homogeneización de las propuestas, la sobreproducción de backlog y la sustitución de la conversación por el chat. Scrum Manager propone un patrón de trabajo en seis pasos, evidencia, borrador asistido, crítica, conversación, refinamiento y confirmación, frente al antipatrón de pedir veinte historias y llamar backlog al resultado. El papel del profesional se desplaza a dar contexto, preguntar, contrastar, encontrar el corte, decidir y verificar la intención.

Por qué está aquí ahora. La evidencia sobre el sesgo de automatización viene de décadas de investigación en otros dominios y sostiene algo incómodo, que la formación y las advertencias no lo previenen, lo que obliga a diseñar el proceso en lugar de confiar en la prudencia. El sesgo de precisión aplicado a producto está documentado por quienes forman a equipos en división de historias, y el patrón de seis pasos es propuesta de la casa, no consenso del oficio.

Dónde mirar. [Parasuraman y Manzey — Complacencia y sesgo en el uso de la automatización](#) · [Humanizing Work — Sesgo de precisión y IA en gestión de producto](#) · [Skill Arena — IA aplicada al trabajo ágil](#)

Bloque H — La historia como entrada de un flujo con agentes

Cuando quien implementa es un agente, la historia deja de ser un recordatorio para la conversación y pasa a ser la expresión verificable de una intención.



De la historia a la spec: la intención como fuente de verdad · EMERGENTE

Las herramientas de desarrollo con agentes trabajan sobre especificaciones que contienen historias con sus criterios, sus escenarios, sus casos límite, sus supuestos y sus ambigüedades marcadas, de modo que la spec contiene a la historia y no la sustituye. El desarrollo guiado por especificaciones (spec-driven development) es una tendencia en evaluación, con defensores que ven la intención convertida en fuente de verdad y críticos que ven el retorno del ciclo en cascada. Lo que sí cambia el resultado es la disciplina de contexto, dejar lo estable del producto y del repositorio en el contexto persistente y en la historia solo lo específico del cambio.

Por qué está aquí ahora. Las herramientas tienen meses de vida y su documentación ya prescribe formatos concretos, incluidas plantillas con historias comprobables por separado y marcas explícitas de lo que falta aclarar. Las posiciones enfrentadas están publicadas, con análisis de campo de Thoughtworks y críticas argumentadas desde el lado del ciclo en cascada, y el BoK de Scrum Manager fija ya la relación, que el desarrollo guiado por especificaciones no sustituye a las historias de usuario, las complementa.

Dónde mirar. [GitHub Spec Kit — desarrollo guiado por especificaciones](#) · [Böckeler — Understanding spec-driven development](#) · [Scrum Manager BoK — Spec-Driven Development](#)

Lo que el agente necesita y el criterio que se verifica solo · EMERGENTE

Un agente necesita en la historia lo que un compañero nuevo necesitaría para hacer la tarea sin preguntar, el objetivo y el usuario real, las reglas y los ejemplos canónicos, los casos límite, las restricciones del cambio, un criterio de éxito verificable, el contexto del sistema, lo que queda fuera de alcance y lo que no debe cambiar. Las dudas se marcan como entregable, porque el agente no pregunta y rellena los huecos por su cuenta. Pocos criterios y canónicos funcionan mejor que muchos, y bajo agentes INVEST se relee, con lo comprobable convertido en verificable por máquina.

Por qué está aquí ahora. Las guías de las herramientas coinciden en lo esencial, criterios completos de lo que sería una buena solución, decir lo que queda fuera de alcance y no dar por bueno lo que

no se puede verificar. Hay además una razón medida para la contención, que el cumplimiento de instrucciones se degrada a medida que crece su número, y trabajos recientes muestran que los agentes no piden aclaraciones cuando el enunciado es ambiguo. La relectura de INVEST bajo agentes es propuesta de Scrum Manager.

Dónde mirar. [Spec Kit — plantilla de especificación](#) · [GitHub — obtener buenos resultados del agente de codificación](#) · [Claude Code — buenas prácticas](#)

La puerta de requisitos desde producto · EMERGENTE

Antes de que un agente arranque, producto revisa seis cosas, que la intención sea la real y no la atribuida, que estén las restricciones y lo que no debe cambiar, que los criterios sean observables, que el alcance sea manejable, que el qué no lleve decisiones técnicas dentro y que las dudas estén marcadas. La pregunta que ordena el refinamiento en este escenario es qué necesita decidir una persona y qué se puede delegar. El papel se desplaza a diseñar los controles del bucle en lugar de revisar cada línea, con el principio de que a más ejecución automatizada, más importa que la intención y los límites estén bien expresados y verificados.

Por qué está aquí ahora. La evidencia de campo desaconseja fiarse de la percepción, porque el estudio de METR midió a desarrolladores expertos un 19 % más lentos con asistencia mientras ellos creían haber ido más rápidos, y el informe DORA de 2025 encontró adopción casi universal con una confianza mucho menor. Las capacidades que DORA asocia al rendimiento con inteligencia artificial incluyen precisamente el lote pequeño y el foco en el usuario, que son el contenido de esta skill.

Dónde mirar. [METR — estudio con desarrolladores expertos](#) · [DORA — informe 2025](#) · [Böckeler — humanos y agentes en el bucle](#)

Nota sobre las referencias

Los enlaces incluidos estaban disponibles y verificados en la fecha de actualización. Dado el ritmo de cambio del área, algunos pueden modificarse; cada revisión del mapa actualiza también sus referencias.

© 2026 Scrum Manager®. Esta obra se publica bajo licencia Creative Commons Atribución – No Comercial 4.0 Internacional (CC BY-NC 4.0). Los formadores y centros oficiales de Scrum Manager quedan licenciados bajo los términos CC BY 4.0 para su actividad formativa.