

## Guía

# Kanban

---

Desarrollo de los temas del State of the art



Actualizado a junio de 2026

## Sobre este documento

Este documento desarrolla en profundidad los temas del mapa de referencia (State of the art) para gestionar el flujo de trabajo del conocimiento con Kanban: visualizar el trabajo, limitar el trabajo en curso, medir el flujo y mejorar de forma evolutiva, que está disponible en: [Skill Arena](#).

Úsalo como material de consulta para mantener y contrastar tu práctica con el conocimiento profesional actual.

Se complementa con la plataforma de entrenamiento y evaluación en Skill Arena. En el área [Kanban](#) puedes realizar pruebas de entrenamiento para contrastar y mejorar tu nivel de conocimiento y si lo deseas también puedes obtener un diploma que acredita curricularmente la solvencia y vanguardia profesional en esta área.



## Estado del conocimiento

El conocimiento sobre Kanban evoluciona a ritmo lento en su núcleo —principios, prácticas y métricas llevan más de una década estables— y progresivo en su frontera: las guías de referencia se revisaron en 2025, el forecasting probabilístico gana terreno y la incorporación de la IA a la gestión de flujo cambia cada pocos 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.

## Índice

Capítulos del desarrollo, en el orden del State of the art.

### **Bloque A · Fundamentos conceptuales**

1. Origen y naturaleza de Kanban
2. Kanban como método de cambio evolutivo
3. Principios de gestión del cambio y de prestación de servicio

### **Bloque B · Las prácticas**

4. Definir y visualizar el trabajo
5. Limitar el WIP y el sistema pull
6. Gestionar el flujo
7. Hacer explícitas las políticas
8. Bucles de feedback y cadencias
9. Mejorar de forma colaborativa y evolutiva

### **Bloque C · Diseño y operación del sistema**

10. El tablero y el sistema Kanban
11. Diseñar un sistema con STATIK (introdutorio)

### **Bloque D · Métricas y previsión**

12. Las cuatro métricas de flujo
13. Visualización del flujo (CFD, scatterplot)
14. Service Level Expectation (SLE)
15. Forecasting probabilístico (Monte Carlo)

### **Bloque E · Frontera**

16. Más allá del equipo: madurez organizacional (KMM) y Flight Levels
17. IA y gestión de flujo: en las dos direcciones

## BLOQUE A · FUNDAMENTOS CONCEPTUALES

## Capítulo 1. Origen y naturaleza de Kanban

### ESTABLECIDO

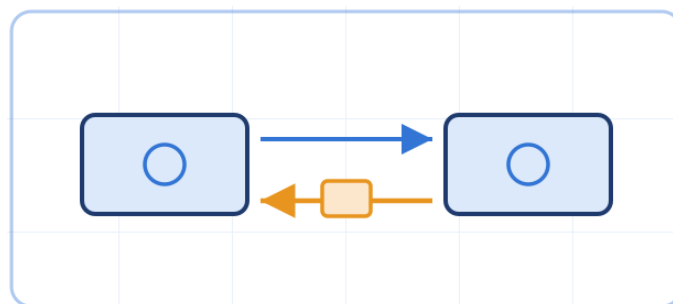
*De dónde viene Kanban y qué es realmente: un método de gestión del flujo de trabajo, no un tablero con columnas.*

### Introducción

Pocas palabras del vocabulario ágil se usan tanto y se entienden tan poco como "Kanban". Para muchos equipos designa simplemente el tablero con columnas donde mueven sus tareas. Ese tablero existe y es valioso, pero es solo una práctica de algo más amplio: un método de gestión cuyo objeto no son las personas ni las tareas sueltas, sino el flujo de valor que las atraviesa. Este capítulo recorre de dónde viene esa idea y qué se transformó al traerla al trabajo del conocimiento.

### 1.1 Del kanban de Toyota al Kanban del conocimiento

El término japonés kanban (看板) puede traducirse como "tablero o tarjeta de señalización". Taiichi Ohno lo empleó en el Sistema de Producción Toyota (finales de los años cuarenta) para nombrar las tarjetas que coordinaban la cadena de montaje: cuando una estación consumía piezas, la tarjeta viajaba aguas arriba y autorizaba a reponer exactamente lo consumido. Nada se producía sin esa señal. El resultado era un sistema pull: la demanda real tiraba de la producción, evitando la sobreproducción y el inventario innecesario.



*La tarjeta kanban original: una señal que viaja aguas arriba y autoriza a producir solo lo que se ha consumido.*

Medio siglo después, esa lógica se adaptó al trabajo del conocimiento. El conjunto de prácticas que hoy llamamos Kanban tomó forma sobre 2006-2007 en equipos de software (el caso fundacional fue un equipo de Corbis, con David J. Anderson y Daniel Vacanti entre sus impulsores) y se extendió después a casi cualquier dominio del conocimiento. En el traslado se conserva lo esencial —visualizar señales, limitar lo que está en curso, dejar que la capacidad tire del trabajo— y se transforma el objeto: ya no fluyen piezas físicas por una cadena, sino ítems de trabajo intelectual por un proceso que, sin ayuda, es invisible.

**La diferencia clave con la fábrica.** En la fábrica el inventario se ve: ocupa estanterías. En el trabajo del conocimiento es invisible: vive en bandejas de correo y documentos a medias. Kanban existe, ante todo, para hacer visible ese trabajo y poder gestionarlo.

### 1.2 Qué problema resuelve: gestionar el flujo, no a las personas

Kanban parte de una observación incómoda: en la mayoría de las organizaciones nadie ve cuánto trabajo hay realmente en curso, dónde se atasca ni cuánto tarda de verdad en completarse. Las consecuencias son conocidas: comenzar mucho y terminar poco, urgencias que adelantan a todo,

plazos que nadie puede prever. La respuesta tradicional es presionar a las personas; la de Kanban, hacer visible el trabajo y su proceso, y gestionar el flujo —dónde se acumula, dónde espera, a qué ritmo avanza— en lugar de la ocupación de quienes lo realizan.

Por eso Kanban se define hoy como una estrategia para optimizar el flujo de valor a través de un proceso, buscando el equilibrio entre tres cualidades: eficacia (entregar lo que los interesados necesitan cuando lo necesitan), eficiencia (aprovechar bien la capacidad disponible) y predictibilidad (prever la entrega con incertidumbre aceptable).

### 1.3 El ecosistema actual de guías

Quien estudie Kanban en 2026 encontrará definiciones que no coinciden literalmente, y conviene saber por qué. El cuerpo de conocimiento se mantiene en dos referencias vivas y complementarias. Por un lado, el método Kanban sistematizado por David J. Anderson y mantenido por Kanban University: principios de gestión del cambio y de servicio, seis prácticas generales, cadencias, clases de servicio y STATIK. Por otro, The Kanban Guide (Vacanti y Coleman, revisada en mayo de 2025), una definición mínima en torno a tres prácticas y al concepto de Definition of Workflow, complementada desde julio de 2025 por el Open Guide to Kanban, adaptación comunitaria abierta.

No son métodos rivales: describen la misma disciplina con distinto nivel de detalle y terminología. Esta guía adopta el cuerpo común a ambas y, cuando un concepto pertenece claramente a una, lo indica.

**Tablero ≠ método.** Un tablero kanban es una herramienta de visibilidad; Kanban es el método de gestión que da sentido a esa herramienta. Si el equipo actualiza el tablero pero no actúa sobre lo que muestra —límites, bloqueos, envejecimiento—, tiene un tablero, no tiene Kanban.

### Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- Explicar el origen Lean de Kanban y qué se conserva y qué se transforma al trasladarlo al trabajo del conocimiento.
- Distinguir Kanban (método de gestión del flujo) de "un tablero con columnas" (una de sus prácticas).
- Situar las referencias actuales del cuerpo de conocimiento (método Kanban, The Kanban Guide, Open Guide) y explicar por qué coexisten.

### Errores y antipatrones frecuentes

**Reducir Kanban a un tablero.** Es el error más extendido: instalar una herramienta de columnas y dar el método por implantado. Sin gestión del flujo, límites ni mejora, el tablero solo decora el proceso de siempre.

**Confundirlo con el kanban físico de fábrica.** El sistema de tarjetas de inventario de Toyota es el antecedente, no el método: aplicar literalmente la lógica de reposición de piezas al trabajo intelectual lleva a conclusiones erróneas.

### Para profundizar

- [BoK Scrum Manager — Kanban: origen y definición](#)
- [The Kanban Guide \(mayo 2025\)](#)
- [Kanban University](#)

## BLOQUE A · FUNDAMENTOS CONCEPTUALES

## Capítulo 2. Kanban como método de cambio evolutivo

### ESTABLECIDO

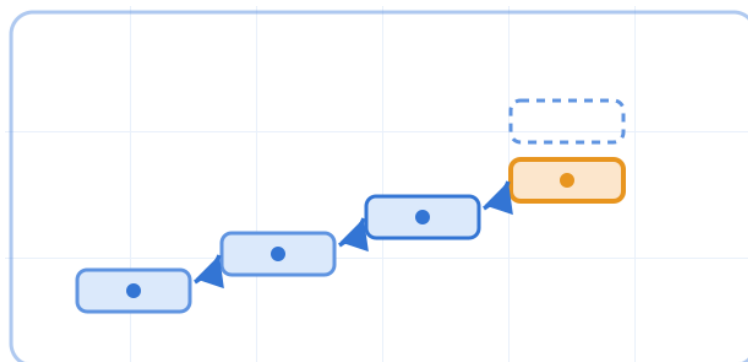
*Kanban no prescribe roles ni eventos: se superpone a lo que ya haces y lo mejora de forma incremental.*

### Introducción

La mayoría de los marcos de trabajo se adoptan sustituyendo lo anterior: nuevos roles, nuevas reuniones, nueva estructura. Kanban hace lo contrario. No prescribe roles, ni eventos obligatorios, ni una forma de organizar el equipo: se aplica sobre el proceso que ya existe, lo hace visible y lo mejora desde dentro, paso a paso. Esa naturaleza evolutiva no es un detalle de estilo: es la decisión de diseño que explica cómo se adopta, por qué genera menos resistencia y cómo convive con otros marcos.

### 2.1 Qué es y qué no es

Kanban es un método de gestión del flujo de trabajo. No es un marco de proceso completo (no dice quién decide qué se construye), no es una metodología de proyecto (no define fases ni hitos) y no exige reorganizar el equipo. Por eso puede aplicarse a un equipo de desarrollo, a un departamento jurídico o a la gestión personal de una sola persona: allí donde haya trabajo que fluye por estados, Kanban tiene algo que aportar.



*Empezar desde lo que se hace ahora y mejorar por pasos pequeños: el siguiente paso siempre está a la vista.*

### 2.2 Evolutivo, no revolucionario

"Implantar Kanban" mediante una reorganización disruptiva es una contradicción en los términos. La adopción correcta empieza desde lo que se hace ahora: se visualiza el proceso real, se acuerdan límites y políticas sobre ese proceso, y a partir de ahí cada mejora es un cambio pequeño, acordado y reversible. Esta aproximación tiene una razón profunda: los cambios impuestos a gran escala disparan la resistencia de las personas y destruyen información valiosa sobre por qué el proceso actual es como es. El cambio evolutivo, en cambio, conserva lo que funciona, apoya cada paso en datos y permite retroceder si un experimento sale mal.

**Implicación práctica.** Si la adopción de Kanban en tu organización empieza con un organigrama nuevo y una semana de formación obligatoria para todos, no está empezando con Kanban: está empezando con la clase de cambio disruptivo que Kanban existe para evitar.

## 2.3 Kanban y Scrum: complementarios, no rivales

La duda práctica más frecuente: ¿Kanban o Scrum? Está mal planteada. Scrum es un marco con roles, eventos y artefactos definidos; Kanban es un método de gestión del flujo que no prescribe nada de eso. Precisamente por eso pueden combinarse: un equipo Scrum puede visualizar el flujo de su sprint, limitar su WIP, medir cycle time y usar previsión probabilística sin dejar de hacer Scrum. La combinación se conoce popularmente como Scrumban, y en la práctica es una de las formas más comunes de aplicar Kanban en equipos que ya trabajan con marcos ágiles.

Las diferencias clásicas (Kniberg y Skarin las catalogaron ya en 2009) ayudan a entender qué aporta cada uno: Scrum prescribe roles, Kanban no; Scrum trabaja con iteraciones de tiempo fijo, Kanban con cadencias; Scrum limita el WIP por iteración, Kanban por estado del flujo; Scrum necesita estimaciones de velocidad, Kanban no las prescribe.

### Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- Argumentar por qué Kanban es evolutivo y no revolucionario, y qué implica eso para su adopción.
- Situar Kanban frente a Scrum: diferencias de enfoque y formas de combinarlos (Scrumban).

### Errores y antipatrones frecuentes

**Tratar Kanban como un marco prescriptivo.** Buscar en Kanban los roles, eventos y artefactos "oficiales" que hay que implantar. No existen: lo que existe son prácticas que se aplican sobre el proceso propio.

**"Implantar Kanban" con una reorganización disruptiva.** Cambiar estructura, herramientas y proceso de golpe en nombre de Kanban contradice su principio fundacional de empezar desde lo que se hace ahora.

### Para profundizar

- [BoK Scrum Manager — Kanban: origen y definición](#)
- [Kanban University](#)

## BLOQUE A · FUNDAMENTOS CONCEPTUALES

## Capítulo 3. Principios de gestión del cambio y de prestación de servicio

### ESTABLECIDO

*Los dos conjuntos de principios fundacionales que dan sentido a las prácticas y guían cada decisión de implementación.*

### Introducción

Las prácticas de Kanban —visualizar, limitar, medir— son fáciles de copiar y, copiadas sin más, fáciles de vaciar de contenido. Lo que evita esa degeneración son los principios: las convicciones sobre cómo cambian las organizaciones y para qué existe un servicio. El método Kanban los organiza en dos conjuntos, y cada uno responde a una pregunta distinta: cómo introducir el cambio, y al servicio de quién está el sistema.

### 3.1 Principios de gestión del cambio

El primer conjunto gobierna cómo se adopta y evoluciona Kanban:

- Empezar desde lo que se hace ahora: respetar el proceso, los roles y las responsabilidades existentes; el punto de partida es la realidad, no un diseño ideal.
- Acordar perseguir la mejora mediante cambio evolutivo: el compromiso compartido de avanzar por pasos pequeños, acordados y basados en evidencia.
- Fomentar el liderazgo en todos los niveles: las mejoras pueden proponerse y liderarse desde cualquier puesto, no solo desde la dirección.

### 3.2 Principios de prestación de servicio

El segundo conjunto recuerda para qué existe el sistema:

- Entender y centrarse en las necesidades y expectativas del cliente: el flujo se optimiza para entregar valor a alguien, no para lucir métricas internas.
- Gestionar el trabajo y dejar que las personas se auto-organicen alrededor de él: se dirige el flujo de los ítems, no a las personas.
- Revisar regularmente la red de servicios y sus políticas para mejorar los resultados: el sistema completo —no solo cada equipo— es objeto de mejora.

### 3.3 De los principios a las decisiones

Los principios no son decoración doctrinal: cada uno se traduce en decisiones concretas. "Empezar desde lo que se hace ahora" decide que el primer tablero modele el proceso real con sus defectos. "Gestionar el trabajo, no a las personas" decide que el tablero siga ítems y no tenga carriles por persona. "Centrarse en el cliente" decide que las clases de servicio se definan por el coste del retraso para quien recibe, no por la comodidad de quien produce. Cuando una decisión de implementación no puede justificarse desde algún principio, suele ser síntoma de que se está copiando una práctica sin su porqué.

**La prueba del porqué.** Ante cualquier elemento del sistema —un límite, una columna, una política— pregunta de qué principio se deriva. Si nadie sabe responder, el elemento es cosmética heredada y es candidato a revisarse.

## Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- Enumerar y explicar ambos conjuntos de principios.
- Relacionar cada principio con una decisión práctica de implementación.

## Errores y antipatrones frecuentes

**Prácticas sin principios.** Adoptar tablero y límites de WIP sin los principios que les dan sentido produce sistemas de cartón piedra: la forma de Kanban sin su función.

## Para profundizar

- [Kanban University](#)
- [Open Guide to Kanban \(julio 2025\)](#)

## BLOQUE B · LAS PRÁCTICAS

## Capítulo 4. Definir y visualizar el trabajo

## ESTABLECIDO

*Hacer visible el trabajo y el proceso real por el que fluye, formalizado hoy como Definition of Workflow (DoW).*

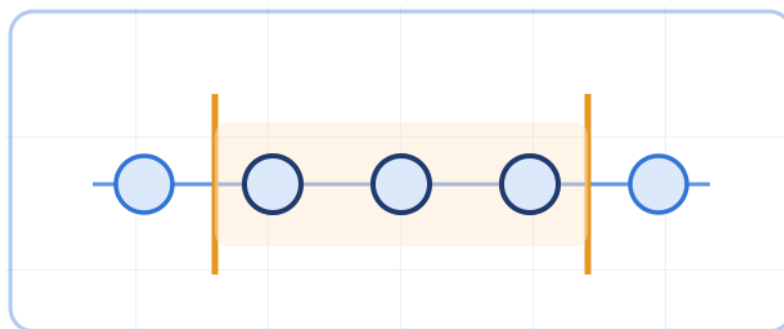
### Introducción

Todo sistema Kanban empieza aquí. Antes de limitar, medir o mejorar nada, el equipo necesita ver su trabajo: qué ítems existen, por qué estados pasan y dónde están ahora mismo. Pero visualizar no es solo dibujar columnas: la versión actual de esta práctica exige primero definir el flujo de manera explícita, y solo después representarlo. Ese acuerdo previo tiene nombre propio en la guía vigente: Definition of Workflow.

### 4.1 La Definition of Workflow (DoW)

La DoW es el entendimiento explícito y compartido de qué significa "flujo" para un equipo concreto. Es el concepto fundamental del que depende todo lo demás: las métricas, la SLE y la propia gestión del tablero solo significan algo respecto a una DoW. Como mínimo, una DoW define seis elementos:

- Qué unidades de valor fluyen por el sistema: los ítems de trabajo.
- Cuándo se considera que un ítem está empezado y cuándo terminado (puede haber más de un punto de cada).
- Los estados por los que pasa un ítem entre el inicio y el fin; lo que está entre ambos puntos es el trabajo en curso (WIP).
- Cómo se controlará el WIP entre el inicio y el fin.
- Las políticas explícitas que gobiernan el paso de los ítems por cada estado.
- Una Service Level Expectation (SLE): la previsión de cuánto debería tardar un ítem de inicio a fin.



*La DoW marca dónde empieza y dónde termina el flujo: solo entonces "WIP", "cycle time" o "terminado" significan algo.*

**Por qué importa tanto el inicio y el fin.** Cada métrica de flujo (capítulo 12) se define a partir de los puntos de empezado y terminado de la DoW. Dos equipos con el mismo trabajo y distinta DoW obtendrán números distintos, y ambos serán correctos: las métricas se interpretan siempre respecto a la DoW que las define.

## 4.2 Visualizar el flujo real, no el ideal

La visualización de la DoW es el tablero kanban. No hay un formato prescrito —columnas, filas, colores y señales son libres—, pero sí un criterio innegociable: el tablero debe representar el proceso real, con sus esperas, sus retrabajos y sus pasos incómodos, no el proceso que al equipo le gustaría tener. Visualizar el proceso idealizado convierte el tablero en propaganda; visualizar el real hace aflorar de inmediato los problemas que el método existe para tratar: colas invisibles, trabajo estancado, demanda que entra por canales laterales.

Mapear los estados es un ejercicio de observación, no de diseño: se sigue el recorrido de varios ítems reales recientes y se anotan los pasos y esperas por los que pasaron de verdad. Ese mapa, con todas sus imperfecciones, es el punto de partida que pide el principio de "empezar desde lo que se hace ahora".

### Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- Describir el flujo de trabajo real de un servicio y representarlo en estados.
- Definir los elementos mínimos de una Definition of Workflow para un caso concreto.
- Justificar por qué visualizar es el primer paso y qué problemas hace aflorar.

### Errores y antipatronos frecuentes

**Copiar el tablero de otro equipo.** Cada tablero visualiza una DoW concreta; importar columnas ajenas es importar el proceso de otro, no visualizar el propio.

**Modelar el proceso deseado en lugar del real.** Un tablero que representa cómo "debería" fluir el trabajo oculta precisamente los problemas que hay que ver.

### Para profundizar

- [The Kanban Guide \(mayo 2025\)](#)
- [BoK Scrum Manager — Tableros kanban: conceptos](#)

## BLOQUE B · LAS PRÁCTICAS

## Capítulo 5. Limitar el WIP y el sistema pull

### ESTABLECIDO

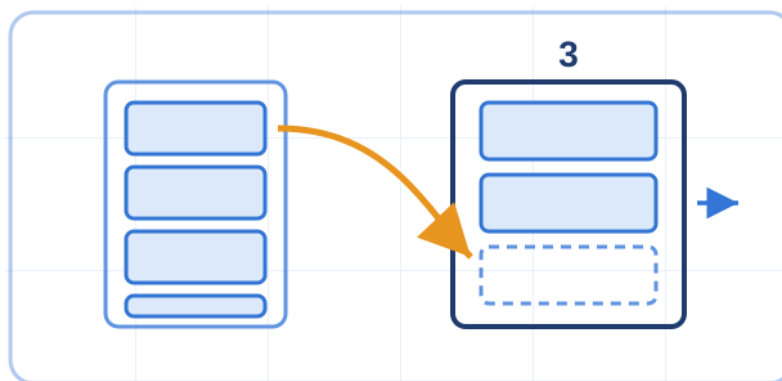
*Limitar el trabajo en curso convierte el sistema en pull: solo se empieza trabajo nuevo cuando hay capacidad para terminarlo.*

### Introducción

Si hubiera que reducir Kanban a una sola intervención, sería esta: poner un límite explícito a la cantidad de trabajo en curso. Es la práctica que separa un sistema Kanban de un simple tablero de tareas, la que produce los efectos más inmediatos y la que más resistencia cultural genera, porque contradice la intuición de que "cuanto más se empieza, más se avanza".

### 5.1 Por qué limitar lo que está en curso

El trabajo en curso (WIP, work in progress) es todo ítem empezado y no terminado. Cada ítem en curso reclama atención, genera cambios de contexto, envejece y bloquea valor que aún no llega a nadie. Cuando el WIP crece sin control, el sistema entero se ralentiza: todo está "en marcha" y nada termina. Limitar el WIP invierte la lógica: en lugar de empujar trabajo nuevo hacia un sistema saturado, se acota cuánto puede estar en curso y se concentra la capacidad en terminar lo empezado. El lema clásico lo resume: stop starting, start finishing.



*El hueco libre dentro del límite es la señal pull: el sistema tira del siguiente ítem solo cuando hay capacidad.*

### 5.2 El sistema pull

El límite de WIP es lo que convierte el tablero en un sistema pull. Mientras el WIP está al límite, no entra nada nuevo; cuando un ítem termina y el WIP cae por debajo del control acordado, ese hueco es la señal de capacidad que autoriza a seleccionar (tirar de) el siguiente ítem. La dirección del control se invierte: no es la demanda la que empuja trabajo hacia dentro, es la capacidad disponible la que tira del trabajo hacia el sistema. Las excepciones aceptables al límite —si las hay— deben ser explícitas y formar parte de la DoW, no decisiones improvisadas bajo presión.

Conviene distinguir dos cosas que se confunden a menudo: el WIP es una métrica (cuántos ítems están en curso ahora); el límite de WIP es una política (cuántos permitimos que estén). La métrica se observa; la política se acuerda y se respeta. La guía vigente habla en general de "controlar" el WIP, dejando flexible el mecanismo; el límite numérico explícito por estado sigue siendo la forma más común y la más fácil de auditar.

### 5.3 La relación cuantitativa: la Ley de Little

La intuición "menos en curso = terminar antes" tiene base matemática. La Ley de Little establece que, en un sistema estable, el cycle time medio es igual al WIP medio dividido por el throughput medio. Manteniendo el ritmo de salida, duplicar el WIP duplica el tiempo medio que cada ítem pasa dentro del sistema. No hace falta dominar la formulación para usar su consecuencia práctica: el WIP es la palanca más directa sobre el tiempo de entrega y sobre la predictibilidad, porque un sistema con menos cosas en vuelo se comporta de forma más regular.

**Ocupado no es lo mismo que entregar.** Un equipo puede estar al 100 % de ocupación con un flujo desastroso: todos trabajando, nada terminándose. Kanban optimiza la salida de valor, no la ocupación de las personas; aceptar cierta holgura es el precio de fluir.

#### Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- Explicar el efecto de limitar el WIP sobre el cycle time y la predictibilidad.
- Distinguir WIP (métrica) de límite de WIP (política).
- Explicar cómo el límite de WIP genera la señal pull del sistema.

#### Errores y antipatronos frecuentes

**Límites inexistentes o no respetados.** Un límite que se ignora cada vez que llega algo "urgente" no es una política: es un número pintado en el tablero.

**Empezar trabajo sin señal de capacidad.** Si el trabajo entra porque alguien lo pide y no porque hay hueco, el sistema sigue siendo push, tenga el aspecto que tenga.

**Confundir estar ocupado con entregar.** Optimizar la ocupación individual maximiza el WIP y degrada el flujo: es exactamente la variable equivocada.

#### Para profundizar

- [The Kanban Guide \(mayo 2025\)](#)
- [ProKanban — The Kanban Guide](#)

## BLOQUE B · LAS PRÁCTICAS

## Capítulo 6. Gestionar el flujo

### ESTABLECIDO

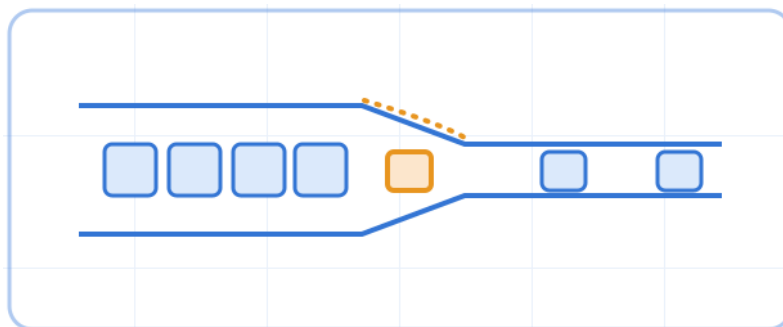
*El trabajo diario del sistema: detectar bloqueos y cuellos de botella, vigilar el envejecimiento y buscar un flujo suave y predecible.*

### Introducción

Visualizar y limitar prepara el terreno; gestionar el flujo es el trabajo de cada día. Un sistema Kanban no se atiende solo: los ítems se bloquean, se acumulan ante un estado, envejecen en silencio. Gestionar activamente significa mirar el tablero como se mira un sistema vivo —dónde se atasca, qué lleva demasiado tiempo quieto— y actuar sobre ello hoy, no en la retrospectiva del mes que viene.

### 6.1 Cuellos de botella y bloqueos

Un cuello de botella es el estado del flujo cuya capacidad limita el rendimiento de todo el sistema: el trabajo se acumula delante de él y escasea después de él. La señal en el tablero es inconfundible: una columna sistemáticamente llena seguida de columnas vacías. La respuesta correcta no es presionar a quienes trabajan en esa fase, sino aliviar el sistema: dirigir capacidad al cuello de botella, dejar de alimentarlo por encima de lo que procesa y atacar las causas de su lentitud.



*El trabajo se acumula delante del estrechamiento y escasea después: la firma visual de un cuello de botella.*

Los bloqueos son otra cosa: ítems que no pueden avanzar por una dependencia o impedimento concreto (una aprobación pendiente, un dato que falta, un tercero que no responde). Se señalizan visualmente sobre la tarjeta y se gestionan de forma activa: cada bloqueo tiene un responsable de desbloquearlo y una antigüedad visible. Un ítem bloqueado que nadie persigue es valor parado que envejece.

### 6.2 El envejecimiento del trabajo en curso

La herramienta más infravalorada de la gestión diaria es la edad del ítem (Work Item Age): cuánto tiempo lleva en curso cada ítem desde que empezó. A diferencia del cycle time —que solo se conoce cuando el ítem ya terminó—, la edad se conoce ahora, mientras aún se puede actuar. Comparar la edad de cada ítem en curso con lo que suele tardar el sistema (la SLE del capítulo 14) identifica de forma temprana los ítems en riesgo: ese que lleva nueve días en curso cuando el 85 % termina en ocho no necesita esperar a fallar para recibir atención.

**La pregunta correcta en la reunión diaria.** No es "¿qué hizo cada uno ayer?", sino "¿qué impide fluir a los ítems más viejos del tablero?". La conversación pasa de las personas al trabajo, que es exactamente donde Kanban quiere ponerla.

### Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- Identificar un cuello de botella y un ítem envejecido a partir del estado del tablero.
- Explicar la diferencia entre gestionar el flujo del trabajo y gestionar la utilización de las personas.

### Errores y antipatrones frecuentes

**Optimizar la ocupación individual a costa del flujo.** Mantener a todos siempre ocupados infla el WIP y alarga las colas: el sistema rinde menos cuanto más "lleno" está.

**Ignorar ítems bloqueados o estancados.** El bloqueo señalado que nadie gestiona se normaliza; el tablero acaba mostrando una arqueología de tarjetas inmóviles.

### Para profundizar

- [The Kanban Guide \(mayo 2025\)](#)
- [ActionableAgile](#)

## BLOQUE B · LAS PRÁCTICAS

## Capítulo 7. Hacer explícitas las políticas

## ESTABLECIDO

*Las reglas del sistema —criterios de avance, clases de servicio, límites, priorización— deben ser explícitas, visibles y compartidas.*

### Introducción

Todo equipo tiene políticas: criterios sobre cuándo algo está listo para avanzar, qué se atiende primero, cómo se tratan las urgencias. La diferencia está en si son explícitas o viven en la cabeza de algunos. Las políticas implícitas generan fricción permanente y hacen imposible mejorar: no se puede revisar una regla que nadie ha formulado.

### 7.1 Qué políticas hacer explícitas

Las políticas típicas de un sistema Kanban incluyen: los criterios de salida de cada estado, los límites de WIP y sus excepciones, las clases de servicio y su tratamiento, los criterios de selección al reponer, y el tratamiento de bloqueos y urgencias. Hechas explícitas y visibles, convierten discusiones de opinión en decisiones de sistema: no se debate si un ítem puede avanzar, se comprueba si cumple el criterio.

### 7.2 Cualidades de una buena política

No basta con escribir reglas; tienen que ser reglas que funcionen. Una buena política es:

- Concisa y simple: cabe en una frase que cualquiera recuerda.
- Bien definida: no admite interpretaciones contradictorias.
- Visible: está donde se toman las decisiones que regula.
- Aplicada siempre: una política con excepciones discrecionales no es una política.
- Fácilmente modificable: cambiarla cuesta una conversación, no un comité.

**Las políticas son del equipo.** Hacer explícita una política no la congela: la expone a mejora. Porque está escrita y compartida, cualquiera puede proponer cambiarla, y el cambio beneficia a todos.

### Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- Redactar políticas explícitas para los estados de un tablero.
- Evaluar una política existente contra las cualidades de una buena política.

### Errores y antipatronos frecuentes

**Políticas implícitas que solo conocen algunos.** El conocimiento tribal excluye a los nuevos y hace el sistema dependiente de personas concretas.

**Políticas rígidas que nadie revisa.** La política escrita hace años que ya no refleja la realidad es tan dañina como la implícita: enseña al equipo a ignorar las reglas.

### Para profundizar

- [Open Guide to Kanban \(julio 2025\)](#)
- [Kanban University](#)

## BLOQUE B · LAS PRÁCTICAS

## Capítulo 8. Bucles de feedback y cadencias

### ESTABLECIDO

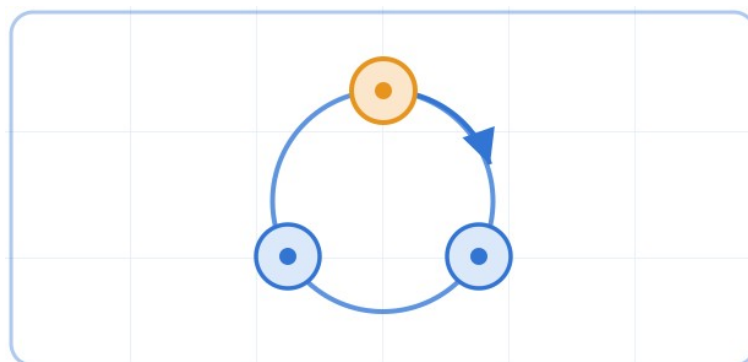
*Las reuniones recurrentes del método que institucionalizan la revisión y la mejora, empezando por un subconjunto pequeño.*

### Introducción

Un sistema sin bucles de feedback visualiza pero no aprende. Kanban institucionaliza el aprendizaje mediante cadencias: reuniones recurrentes, cada una con un propósito y un ritmo propios, donde la información del sistema se convierte en decisiones. La palabra cadencia no es casual: frente a las iteraciones de tiempo fijo, Kanban desacopla los ritmos —se revisa, se repone y se entrega cada cosa a su frecuencia natural.

### 8.1 Las siete cadencias del método

El método Kanban (Anderson) define siete cadencias que cubren desde la operación diaria hasta la estrategia: la reunión Kanban (diaria, sobre el estado del flujo), la reunión de reposición (selección de qué entra al sistema), la planificación de entrega, la revisión de prestación del servicio (¿cumplimos lo que el cliente espera?), la revisión de operaciones (equilibrio entre servicios), la revisión de riesgos y la revisión de estrategia. Juntas forman una red de bucles a distintas escalas de tiempo, donde cada reunión alimenta a las demás.



*Cadencias: bucles recurrentes a distintos ritmos; se empieza por pocas y cada reunión termina en decisiones.*

### 8.2 Por dónde empezar

Importar las siete cadencias de golpe es un error clásico: multiplica reuniones antes de que el sistema genere la información que las justifica. La mayoría de las implementaciones nuevas empiezan con un subconjunto: la reunión Kanban diaria, la reunión de reposición y, cuando el servicio madura, la revisión de prestación del servicio. Y casi siempre la forma más sensata de introducirlas es reutilizar reuniones existentes cambiando su contenido: la daily que ya existe pasa a girar en torno al tablero y a los ítems envejecidos; la reunión de priorización que ya existe se convierte en reposición.

**El criterio de una cadencia sana.** Toda cadencia debe terminar en decisiones: ítems desbloqueados, trabajo seleccionado, una política ajustada. Una reunión recurrente de la que no salen decisiones es un informe de estado con otro nombre, y es candidata a desaparecer.

## Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- Nombrar las cadencias del método y explicar el propósito de las tres de arranque.
- Decidir qué cadencias introducir primero en un contexto dado, reutilizando reuniones existentes.

## Errores y antipatrones frecuentes

**Importar las siete cadencias de golpe.** Instala la burocracia del método antes que su sustancia; las reuniones sin información que procesar se vacían y desprestigian el sistema.

**Reuniones sin decisión asociada.** La cadencia que solo informa no es un bucle de feedback: el bucle se cierra cuando la información cambia algo.

## Para profundizar

- [Kanban University](#)
- [Open Guide to Kanban \(julio 2025\)](#)

## BLOQUE B · LAS PRÁCTICAS

## Capítulo 9. Mejorar de forma colaborativa y evolutiva

## ESTABLECIDO

*La mejora continua como experimento: hipótesis, datos compartidos y pasos pequeños, no rediseños impuestos.*

### Introducción

La última práctica cierra el círculo: el sistema hecho visible, limitado y medido se convierte en objeto de mejora permanente. El cómo importa tanto como el qué: la mejora es colaborativa (la propone y valida el equipo, con datos que todos ven) y evolutiva (experimentos pequeños y reversibles, no grandes rediseños).

### 9.1 La mejora como experimento

La forma madura de mejorar es el experimento: una hipótesis explícita ("si bajamos el límite de WIP de 5 a 4, el cycle time del percentil 85 bajará"), una métrica de verificación acordada y un plazo para evaluar. Si los datos la confirman, el cambio se queda; si no, se revierte sin drama. Plantearlo así obliga a predecir —y por tanto a entender el sistema— y despersonaliza el resultado: fracasa la hipótesis, no quien la propuso.

Los datos compartidos son la condición: las métricas del bloque D dan un modelo común sobre el que razonar. Sin datos, la mejora se decide por jerarquía o elocuencia; con datos, por evidencia.

### 9.2 Por qué evolutivo reduce la resistencia

El cambio impuesto activa resistencia por una razón sana: las personas defienden un proceso que conocen frente a uno que no han elegido. El evolutivo desactiva esa dinámica: cada paso es pequeño (riesgo percibido bajo), acordado (nadie lo sufre, todos lo deciden) y reversible (equivocarse no es catastrófico). Matiz: evolutivo no significa diminuto; si la evidencia pide un cambio profundo, el equipo puede acometerlo. Lo evolutivo está en cómo se decide y se valida, no en el tamaño.

**Conexión con las cadencias.** Las mejoras no esperan a una reunión: cualquier momento vale para ajustar; las cadencias del capítulo 8 añaden el ritmo regular para revisar con perspectiva.

### Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- Plantear una mejora como experimento con hipótesis y métrica de verificación.
- Explicar por qué el cambio evolutivo reduce la resistencia frente al cambio impuesto.

### Errores y antipatronos frecuentes

**Mejoras impuestas sin datos.** El cambio que llega por decreto y sin evidencia enseña al equipo que las métricas son decorativas y que opinar es inútil.

**Grandes rediseños en lugar de iteración.** El rediseño total descarta lo aprendido y apuesta todo a una carta; la iteración conserva lo que funciona y arriesga poco en cada paso.

### Para profundizar

- [The Kanban Guide \(mayo 2025\)](#)
- [Open Guide to Kanban \(julio 2025\)](#)

## BLOQUE C · DISEÑO Y OPERACIÓN DEL SISTEMA

## Capítulo 10. El tablero y el sistema Kanban

### ESTABLECIDO

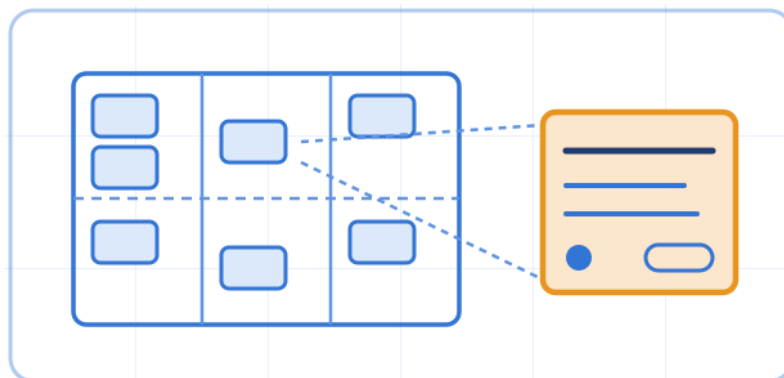
*Los elementos que componen un sistema concreto: tablero, tarjeta, clases de servicio y gestión visual de bloqueos, diseñados a medida del servicio.*

### Introducción

Los capítulos anteriores describen qué hace Kanban; este describe con qué. Un sistema Kanban se materializa en unos pocos elementos de diseño —el tablero, la tarjeta, las clases de servicio, las señales de bloqueo— y la calidad del sistema depende de que cada uno esté diseñado para el servicio concreto al que sirve. No hay tablero estándar, igual que no hay proceso estándar.

### 10.1 El diseño del tablero

Las columnas representan los estados de la DoW, en el orden real del flujo. Cuando conviven trabajos de naturaleza distinta —tipos de ítem, productos, clases de servicio—, los carriles horizontales (swimlanes) separan visualmente esas corrientes sin romper la lectura del flujo. Sobre esa estructura se rotulan los límites de WIP y las políticas de salida de cada estado, de modo que el tablero no solo muestre dónde está el trabajo, sino las reglas del juego. Las columnas de espera ("listo para X") merecen existir como estados propios: las colas son donde el tiempo se pierde, y solo se gestionan si se ven.



*Columnas para los estados, carriles para las corrientes de trabajo y la tarjeta como portadora de la información que el flujo necesita.*

### 10.2 El diseño de la tarjeta

La tarjeta (o ticket) es la unidad de información del sistema. Debe llevar lo que el flujo necesita para fluir, ni más ni menos: identificación clara del ítem, tipo o clase de servicio (a menudo por color), fecha de inicio o edad visible, señal de bloqueo cuando lo hay y, si aplica, fecha comprometida. Cada dato de la tarjeta debería responder a una pregunta que alguien hace de verdad delante del tablero; los campos que nadie consulta son ruido que esconde lo importante.

### 10.3 Clases de servicio

No todo el trabajo merece el mismo tratamiento. Las clases de servicio clasifican los ítems por su coste de retraso y asignan a cada clase una política de tratamiento: la clase urgente (un incidente que para el negocio) salta colas y puede romper límites de forma acordada; la de fecha fija (un requisito regulatorio con plazo) se ordena para llegar a tiempo; la estándar fluye por orden; la intangible (mejora interna, deuda técnica) avanza con capacidad reservada para que nunca quede

eternamente postergada. Definidas con el cliente en mente, las clases convierten la priorización diaria en aplicación de política y no en negociación caso a caso.

**A medida, no de plantilla.** Las plantillas de tablero de las herramientas son puntos de partida cómodos y sistemas finales malos. El tablero correcto se parece al servicio que representa; si dos servicios distintos tienen tableros idénticos, al menos uno de los dos está mal modelado.

### Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- Diseñar un tablero y una tarjeta adecuados a un servicio concreto.
- Definir clases de servicio y su tratamiento a partir del coste de retraso.

### Errores y antipatronos frecuentes

**Tablero genérico copiado.** El "Pendiente / En curso / Hecho" universal no modela ningún servicio real: oculta las colas y los estados donde el trabajo de verdad espera.

**Tarjetas sin la información que el flujo necesita.** Si para saber el estado real de un ítem hay que abrir tres aplicaciones, el tablero ha dejado de ser el punto de verdad.

**Ausencia de señalización de bloqueos.** Los bloqueos invisibles no se gestionan: un ítem parado sin señal es indistinguible de uno avanzando.

### Para profundizar

- [BoK Scrum Manager — Tableros kanban: conceptos](#)
- [BoK Scrum Manager — Ejemplo de tablero kanban](#)

## BLOQUE C · DISEÑO Y OPERACIÓN DEL SISTEMA

**Capítulo 11. Diseñar un sistema con STATIK (introductorio)****ESTABLECIDO**

*Una forma repetible y colaborativa de arrancar un sistema Kanban, servicio a servicio y de manera iterativa.*

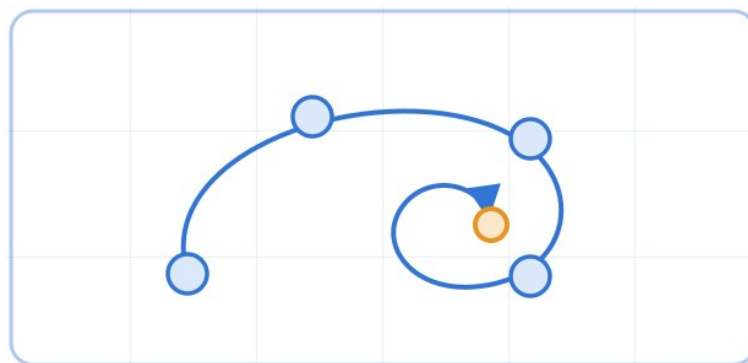
**Introducción**

Saber qué es un sistema Kanban no responde a la pregunta práctica: ¿por dónde empiezo? STATIK (Systems Thinking Approach to Introducing Kanban) es la respuesta del método: un recorrido de pensamiento sistémico que lleva de "tenemos un servicio con problemas" a "tenemos un sistema Kanban diseñado para él". Dos rasgos lo definen: se aplica a cada servicio por separado (cada servicio tiene su demanda, su flujo y su sistema) y es iterativo, no secuencial: los pasos se revisitan a medida que cada uno enseña algo sobre los anteriores.

**11.1 El recorrido**

En su versión introductoria, el recorrido de STATIK pasa por estos pasos:

- Identificar el servicio y sus fuentes de insatisfacción: qué se entrega, a quién, y qué les duele hoy a quienes lo reciben y a quienes lo prestan.
- Entender la demanda: qué tipos de trabajo llegan, por qué canales, con qué ritmo y qué espera cada uno.
- Analizar la capacidad: qué puede entregar el sistema de forma realista, observando datos recientes.
- Modelar el flujo de trabajo: por qué estados pasa de verdad cada tipo de ítem, de la solicitud a la entrega.
- Descubrir las clases de servicio: qué tratamientos distintos pide la demanda según su coste de retraso.
- Diseñar el sistema Kanban: tablero, límites, políticas y cadencias que dan respuesta a todo lo anterior.



*STATIK es un recorrido iterativo: cada paso ilumina los anteriores y el diseño converge por vueltas sucesivas.*

El orden enumera, no manda: al modelar el flujo se descubre demanda que no se había visto; al analizar la capacidad se reinterpretan las insatisfacciones. Por eso STATIK se recorre en taller colaborativo, con las personas que viven el servicio, y se vuelve sobre los pasos hasta que el diseño converge. Existe un nivel más avanzado del enfoque —análisis profundo de stakeholders, criterios de satisfacción del cliente como punto de partida— que queda fuera del alcance de fundamentos: aquí basta saber que existe.

**STATIK no termina.** El mismo recorrido sirve para revisar un sistema en marcha: cuando el servicio cambia —nueva demanda, nuevo equipo, nuevas insatisfacciones—, volver a pasar por STATIK es la forma ordenada de rediseñar.

### Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- Recorrer los pasos centrales de STATIK sobre un servicio de ejemplo.
- Reconocer que STATIK es un proceso de pensamiento iterativo, no una checklist secuencial.

### Errores y antipatrones frecuentes

**Aplicar STATIK como pasos rígidos y de una sola pasada.** Tratarlo como checklist produce un diseño hecho con la información del primer día, que es siempre la peor disponible.

**Hacerlo en solitario.** El STATIK que ejecuta un coach o un jefe sin el equipo produce el sistema de esa persona, no el del servicio: nace sin el conocimiento ni el compromiso de quienes lo operarán.

### Para profundizar

- [Kanban University](#)
- [BoK Scrum Manager — Tableros kanban: conceptos](#)

## BLOQUE D · MÉTRICAS Y PREVISIÓN

## Capítulo 12. Las cuatro métricas de flujo

### ESTABLECIDO

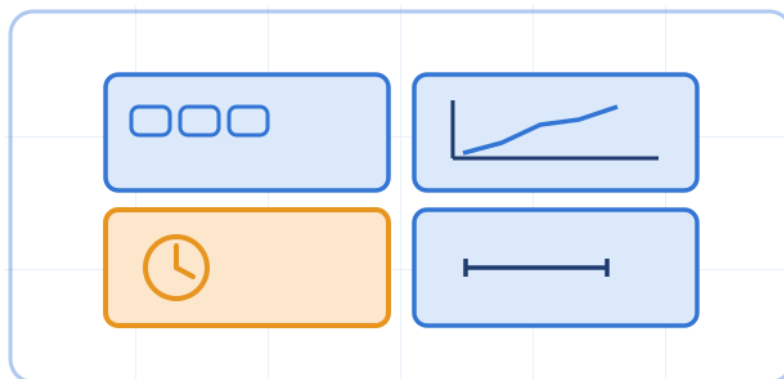
*WIP, Throughput, Work Item Age y Cycle Time: el mínimo de medición de cualquier sistema Kanban y la base de toda previsión.*

### Introducción

Operar un sistema Kanban exige medirlo. La guía vigente fija cuatro métricas de flujo obligatorias — el mínimo, no el máximo— y todas se definen sobre la DoW del capítulo 4: empezado y terminado significan lo que la DoW del equipo diga que significan. Las cuatro juntas responden a las preguntas esenciales: cuánto hay en curso, a qué ritmo sale, cuánto lleva dentro lo que aún no salió y cuánto tardó lo que ya salió.

### 12.1 Las cuatro métricas

- **WIP (trabajo en curso):** el número de ítems empezados y no terminados. Fotografía instantánea de la carga del sistema.
- **Throughput (rendimiento):** el número de ítems terminados por unidad de tiempo. Se mide en conteo exacto de ítems, sin ponderar por tamaño ni puntos.
- **Work Item Age (edad del ítem):** el tiempo transcurrido desde que un ítem en curso empezó hasta hoy. Es el indicador adelantado: habla del trabajo sobre el que aún se puede actuar.
- **Cycle Time (tiempo de ciclo):** el tiempo transcurrido entre que un ítem empezó y terminó. Es el indicador atrasado: solo existe cuando ya no hay nada que hacer con ese ítem.



*Las cuatro métricas mínimas; la edad del ítem en curso (acento) es la única que avisa a tiempo.*

### 12.2 Adelantadas y atrasadas: por qué la edad es la alerta

La distinción entre indicadores adelantados y atrasados organiza el uso diario. WIP y Work Item Age describen el presente y permiten intervenir: un WIP que crece semana a semana anuncia un sistema que empieza más de lo que termina; un ítem cuya edad supera lo habitual está pidiendo ayuda hoy. Throughput y Cycle Time describen el pasado y alimentan la previsión y la mejora. Equipos enteros miden con esmero su cycle time y desatienden la edad de lo que está en curso: miran el retrovisor con lupa mientras conducen sin parabrisas.

### 12.3 Nombres distintos, métricas iguales

La guía vigente admite expresamente que estas medidas reciban otros nombres siempre que se usen como están definidas: Cycle Time aparece como Flow Time en algunos marcos, Throughput como Delivery Rate o velocidad de entrega. La advertencia es práctica: cada herramienta (Jira, Azure

DevOps, Businessmap...) usa su propia nomenclatura, y a veces la misma palabra designa cosas distintas. Antes de comparar números, hay que comprobar qué definición —y qué puntos de inicio y fin— usa cada fuente.

**La métrica no decide: informa.** Por sí solas, las cuatro métricas no significan nada; valen en cuanto informan alguna de las prácticas: ajustar un límite, atender un ítem envejecido, revisar una política. Medir sin actuar es coleccionar números.

### Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- Definir las cuatro métricas y clasificar cada una como indicador adelantado o atrasado.
- Diagnosticar un sistema a partir de sus métricas (por ejemplo, un WIP creciente semana a semana).

### Errores y antipatrones frecuentes

**Confundir throughput con velocity.** El throughput cuenta ítems terminados, sin ponderar; la velocity de Scrum suma puntos estimados. Mezclarlos invalida cualquier previsión basada en flujo.

**Medir solo cycle time e ignorar la edad.** El cycle time llega cuando ya es tarde; sin vigilar la edad de los ítems en curso, el equipo descubre los problemas en la autopsia.

### Para profundizar

- [The Kanban Guide \(mayo 2025\)](#)
- [ActionableAgile](#)

## BLOQUE D · MÉTRICAS Y PREVISIÓN

**Capítulo 13. Visualización del flujo (CFD, scatterplot)****ESTABLECIDO**

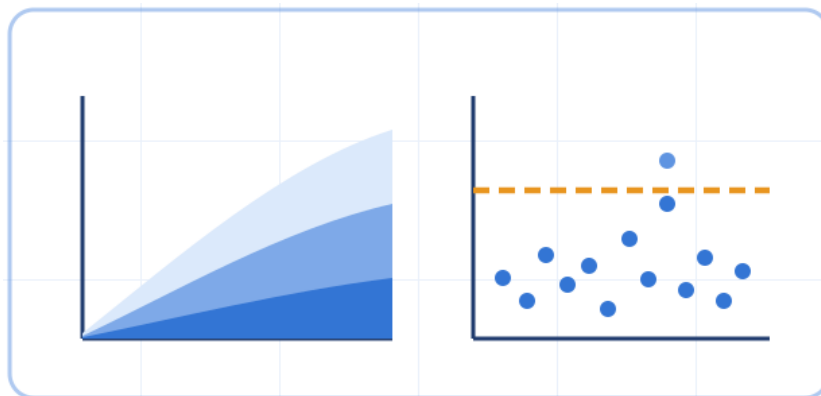
*El diagrama de flujo acumulado y el scatterplot de cycle time: las dos vistas que convierten los datos del flujo en conversación.*

**Introducción**

Los números de las métricas, en tablas, dicen poco; dibujados, hablan. Dos visualizaciones se han convertido en el estándar para leer un sistema Kanban: el diagrama de flujo acumulado, que muestra la salud del sistema completo a lo largo del tiempo, y el scatterplot de cycle time, que muestra la distribución real de los tiempos de entrega. Saber leerlas es alfabetización básica del flujo.

**13.1 El diagrama de flujo acumulado (CFD)**

El CFD apila, día a día, cuántos ítems hay en cada estado del flujo: cada banda de color es un estado, y la banda crece o encoge según entra y sale trabajo. Su gramática es simple y potente: la distancia vertical entre dos fronteras de banda es el WIP de ese estado en ese día; la distancia horizontal aproxima cuánto tarda un ítem en atravesarlo. Las señales de problema saltan a la vista: una banda que se ensancha de forma sostenida es trabajo acumulándose ante un estado (el cuello de botella del capítulo 6); bandas superiores planas indican que nada termina; escalones bruscos delatan entregas por lotes.



*CFD a la izquierda (el WIP es la distancia vertical entre bandas) y scatterplot con su percentil 85 a la derecha.*

**13.2 El scatterplot de cycle time y los percentiles**

El scatterplot dibuja un punto por cada ítem terminado: en el eje horizontal, cuándo terminó; en el vertical, cuánto tardó. El resultado nunca es una línea: es una nube, porque el cycle time de un sistema real es una distribución con cola larga —muchos ítems rápidos y unos pocos muy lentos—. Sobre la nube se trazan líneas de percentil: la del 85 %, por ejemplo, marca el tiempo por debajo del cual terminó el 85 % de los ítems. Esa lectura es la base de la SLE del capítulo siguiente.

Aquí está el salto de madurez analítica: hablar en percentiles en lugar de promedios. El promedio de una distribución con cola larga es una ficción que casi ningún ítem cumple, y promete sistemáticamente de menos. "El 85 % termina en 8 días o menos" informa; "de media tardamos 5 días" confunde.

**Los gráficos son para decidir.** Un CFD que nadie mira en la revisión del servicio, un scatterplot que no cambia ninguna conversación con el cliente: visualización decorativa. Cada gráfico debe tener un foro donde se use y decisiones que dependan de él.

### Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- Leer un CFD e interpretar qué indica el ensanchamiento o estrechamiento de bandas.
- Leer percentiles en un scatterplot de cycle time y explicar por qué sustituyen al promedio.

### Errores y antipatrones frecuentes

**Mirar promedios en lugar de distribuciones.** En distribuciones de cola larga, el promedio infrarrepresenta el riesgo: la mitad mala de la realidad desaparece del discurso.

**Gráficos que nadie usa para decidir.** La visualización sin foro ni decisiones asociadas degenera en informe ritual: cuesta esfuerzo y no cambia nada.

### Para profundizar

- [ActionableAgile](#)
- [ProKanban — The Kanban Guide](#)

## BLOQUE D · MÉTRICAS Y PREVISIÓN

## Capítulo 14. Service Level Expectation (SLE)

### ESTABLECIDO

*La previsión de cuánto debería tardar un ítem, expresada como tiempo más probabilidad, basada en el cycle time histórico.*

### Introducción

¿Cuánto tardará esto? Es la pregunta que todo cliente hace y que muchos equipos responden mal: con una promesa voluntarista o con un "depende" inútil. La SLE es la respuesta profesional: una previsión calculada de los propios datos del sistema, formulada con la honestidad de la probabilidad.

### 14.1 Qué es y cómo se formula

La Service Level Expectation es la previsión de cuánto debería tardar un ítem en fluir de su punto de inicio a su punto de fin. Tiene siempre dos partes inseparables: un período de tiempo y una probabilidad asociada. "El 85 % de los ítems se terminará en 8 días o menos" es una SLE; "tardamos unos 8 días" no lo es. La probabilidad no es un adorno: reconoce que el sistema tiene variabilidad y que prometer certeza sería mentir. La SLE forma parte de la Definition of Workflow y se visualiza en el propio flujo, donde guía las decisiones diarias.

### 14.2 De dónde sale y para qué se usa

La SLE se calcula del cycle time histórico: el percentil elegido sobre el scatterplot del capítulo anterior. Si el percentil 85 de los últimos meses es 8 días, esa es la SLE al 85 %. Cuando el sistema es nuevo, una estimación provisional sirve de arranque y se sustituye por el cálculo real en cuanto hay histórico. Su uso más valioso es diario: comparar la edad de cada ítem en curso con la SLE señala a tiempo los ítems en riesgo —el que lleva 6 días cuando la SLE es "85 % en 8" merece atención hoy.

**Previsión, no promesa ni objetivo.** La SLE describe cómo se comporta el sistema; no es un compromiso contractual ni una meta que imponer al equipo. Convertirla en objetivo invita a manipular los datos, y entonces deja de servir para prever con honestidad.

### Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- Calcular una SLE a partir del cycle time histórico y expresarla correctamente (tiempo + probabilidad).
- Distinguir una SLE de un compromiso de fecha sin base estadística.

### Errores y antipatrones frecuentes

**La SLE como objetivo impuesto.** Usarla como vara de medir al equipo la invierte: de previsión honesta pasa a cifra que defender, con los datos como primera víctima.

**SLE sin probabilidad asociada.** "Lo entregamos en 8 días" sin percentil es una promesa disfrazada de dato: el día que un ítem tarde 12, la confianza se rompe.

### Para profundizar

- [The Kanban Guide \(mayo 2025\)](#)
- [ProKanban — The Kanban Guide](#)

## BLOQUE D · MÉTRICAS Y PREVISIÓN

## Capítulo 15. Forecasting probabilístico (Monte Carlo)

### EN CONSOLIDACIÓN

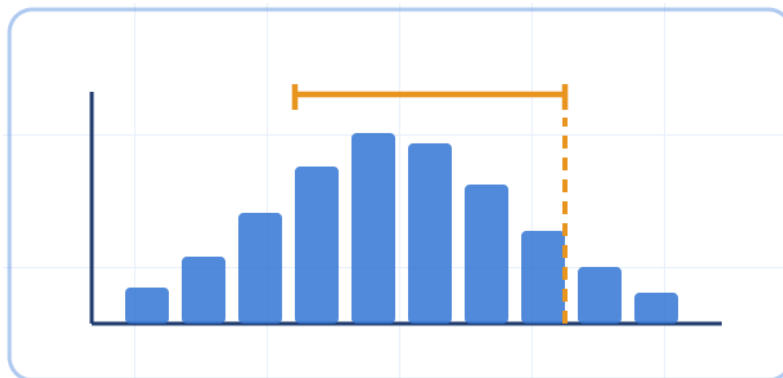
*Prever entregas simulando miles de escenarios sobre los datos históricos: rangos con niveles de confianza en lugar de fechas de punto único.*

### Introducción

La SLE responde por un ítem; las preguntas grandes son por muchos: ¿cuándo estarán estos veinte ítems?, ¿cuántos cabrán antes de fin de trimestre? La respuesta tradicional —estimar cada pieza y sumar— es laboriosa y sistemáticamente optimista. El forecasting probabilístico responde de otra manera: deja que los datos del propio sistema simulen el futuro.

### 15.1 Cómo funciona una simulación Monte Carlo

La idea es simple: si el throughput diario reciente del equipo es conocido, el futuro probable puede simularse muestreando ese historial. Para responder "¿cuándo terminaremos 20 ítems?", la simulación toma días al azar del histórico de throughput y los encadena hasta sumar 20 ítems terminados, anotando cuántos días hicieron falta; repite el proceso miles de veces, y la distribución de resultados dice qué fechas son probables y cuáles no. El mismo mecanismo, a la inversa, responde "¿cuántos ítems caben hasta tal fecha?". Solo requiere datos que el sistema ya genera (el throughput histórico) y está soportado por herramientas accesibles, de productos comerciales a hojas de cálculo.



*Miles de escenarios simulados forman una distribución: la respuesta es un rango con su nivel de confianza, no una fecha.*

### 15.2 Leer el resultado: rangos y confianza

El resultado de la simulación nunca es una fecha: es una distribución, que se comunica como rango con niveles de confianza —"90 % de probabilidad de terminar los 20 ítems en 12 a 18 días"—. Leerla exige el mismo cambio de mentalidad que los percentiles: el percentil 50 no es "la fecha", es la fecha que se incumplirá la mitad de las veces; los compromisos serios se toman en percentiles altos (85, 95) asumiendo el coste de la prudencia. Y la simulación hereda las limitaciones de sus datos: si el contexto ha cambiado drásticamente —equipo nuevo, tipo de trabajo distinto—, el histórico antiguo simula un sistema que ya no existe.

**Por qué "en consolidación".** La técnica está madura, validada y soportada por herramientas; lo que aún no es universal es el hábito. Cada vez más equipos la usan como sustituto directo de la estimación tradicional, y es candidata firme a pasar a ESTABLECIDO en próximas revisiones del mapa.

## Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- Explicar qué hace una simulación Monte Carlo y qué datos necesita.
- Interpretar un resultado probabilístico (rango + percentiles de confianza) frente a una estimación de punto único.

## Errores y antipatrones frecuentes

**Tomar el percentil 50 como "la fecha".** Es elegir el escenario que falla una de cada dos veces y comunicarlo como compromiso: el desengaño está garantizado por construcción.

**Alimentar la simulación con datos obsoletos.** Monte Carlo proyecta el pasado; si el sistema que generó esos datos ya no existe, la previsión es precisa sobre el sistema equivocado.

## Para profundizar

- [ActionableAgile — Monte Carlo](#)
- [ActionableAgile](#)

## BLOQUE E · FRONTERA

## Capítulo 16. Más allá del equipo: madurez organizacional (KMM) y Flight Levels

### EN CONSOLIDACIÓN

*Dos modelos para crecer desde los fundamentos: la hoja de ruta de madurez del KMM y la conexión entre niveles de Flight Levels.*

### Introducción

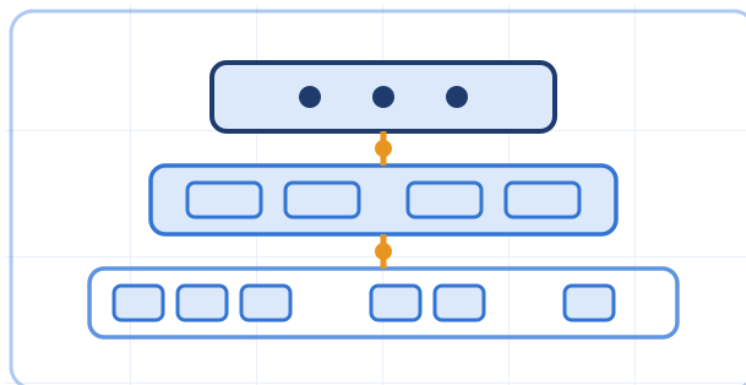
Los fundamentos de esta guía operan sobre todo a escala de equipo o servicio. Tarde o temprano aparece la pregunta de la escala: ¿cómo se extiende esto a un departamento, a un portfolio, a una organización? Dos modelos dominan esa conversación, y conviene presentarlos con precisión porque, pese a citarse juntos, son de naturaleza distinta. Ambos se presentan aquí como horizonte: el profesional de fundamentos debe saber que existen y qué problema resuelve cada uno, no dominarlos.

### 16.1 El Kanban Maturity Model (KMM)

El KMM es un modelo de madurez organizacional: organiza más de 150 prácticas (de Kanban y de gestión general) en siete niveles de madurez, desde la organización inconsciente del flujo hasta la preparada para reinventarse. Su valor es de hoja de ruta evolutiva: indica qué prácticas tienen sentido en cada nivel y cuáles fracasarán si se adoptan antes de tiempo, evitando el error de exigir a una organización inmadura comportamientos de madurez alta. Una precisión que su propio nombre invita a olvidar: el KMM es un modelo de madurez de la organización en su conjunto —cultura, resultados y prácticas—, del que Kanban es una parte, no el todo.

### 16.2 Flight Levels

Flight Levels, de Klaus Leopold, responde a otra pregunta: no cuán madura es la organización, sino dónde mirar para mejorar el flujo de extremo a extremo. La metáfora es la altitud de vuelo: el nivel operativo (los equipos y su trabajo diario), el nivel de coordinación (el flujo entre equipos que comparten un resultado) y el nivel estratégico (las apuestas de la organización). Su tesis central: optimizar equipos aislados no mejora el conjunto si nadie gestiona el flujo entre niveles; agilizar veinte equipos no agiliza una organización cuyas iniciativas mueren en las dependencias. Sus autores insisten en algo que el emparejamiento habitual con el KMM tiende a confundir: Flight Levels no es un modelo de madurez —un nivel operativo puede estar más maduro que el estratégico— sino un modelo de agilidad de negocio para conectar estrategia, coordinación y ejecución.



*Tres niveles de vuelo —operativo, coordinación, estrategia— y la conexión entre ellos como objeto de gestión.*

**Horizonte, no prerequisite.** Ninguno de los dos modelos hace falta para empezar: un servicio con su DoW, sus límites y sus métricas es Kanban completo y valioso. Escalar es la continuación natural de unos fundamentos que funcionan, no la condición para tenerlos.

### Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- Explicar qué problema resuelven el KMM y Flight Levels, a quién van dirigidos y en qué se diferencian.
- Situar la madurez organizacional como continuación natural de los fundamentos, no como prerequisite.

### Errores y antipatronos frecuentes

**Saltar al escalado sin fundamentos.** Diseñar el sistema de coordinación de una organización cuyos equipos no visualizan ni limitan nada es construir el tejado sin la casa.

**Leer el KMM como un modelo "sobre Kanban".** Su nombre engaña: es un modelo de madurez organizacional con Kanban como una de sus partes; reducirlo a niveles de uso del tablero lo malinterpreta.

### Para profundizar

- [Kanban Maturity Model](#)
- [Flight Levels Academy](#)

## BLOQUE E · FRONTERA

## Capítulo 17. IA y gestión de flujo: en las dos direcciones

**EMERGENTE**

*La IA aplicada al flujo y el flujo aplicado a la IA: el frente más volátil del mapa, con la gobernanza como condición.*

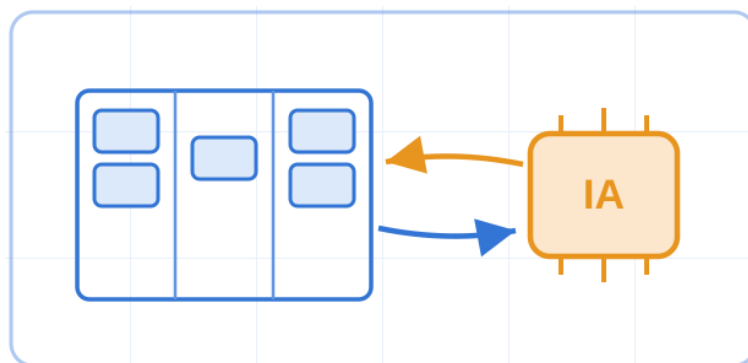
### Introducción

Ningún puntero de este mapa cambia tan deprisa como este. La inteligencia artificial está entrando en la gestión de flujo por dos puertas a la vez, y conviene distinguirlas porque plantean preguntas distintas: la IA como instrumento que ayuda a gestionar el flujo de trabajo humano, y Kanban como instrumento que gobierna el flujo de trabajo de la propia IA. Las dos direcciones avanzan rápido, ninguna tiene todavía consenso asentado, y en ambas la condición de entrada es la misma: gobernanza, supervisión humana y trazabilidad.

#### 17.1 La IA gestiona el flujo: tres niveles

La primera dirección admite una escala de tres niveles según cuánta decisión se delega. El primero es la analítica predictiva: modelos que aprenden del histórico del sistema para anticipar cuellos de botella, predecir fechas y señalar ítems en riesgo —una extensión natural de las métricas del bloque D, donde la IA propone y las personas deciden—. El segundo es la automatización asistida: el tablero ejecuta acciones bajo reglas o sugerencias del modelo (etiquetar, priorizar, mover de estado al cumplirse condiciones, avisar a quien corresponda). El tercero son los agentes autónomos: sistemas que ejecutan flujos de trabajo multi-paso de principio a fin, ya en despliegue comercial en las plataformas de gestión del trabajo desde 2026.

La escala importa porque el riesgo no se reparte igual: la analítica equivocada desinforma, la automatización equivocada actúa, y el agente equivocado actúa muchas veces seguidas y deprisa. De ahí el criterio general de adopción prudente: empezar por los niveles que informan, exigir trazabilidad de cada acción automática, y delegar decisiones de flujo solo donde el error es barato y reversible.



*Dos direcciones: la IA ayuda a gestionar el tablero, y el tablero —límites de WIP, estado de revisión— gobierna a la IA.*

#### 17.2 El flujo gestiona a la IA: Kanban para el trabajo de agentes

La segunda dirección es la más reveladora para quien conoce el método: los equipos que trabajan con agentes de IA están redescubriendo Kanban como sistema de gobierno de esos agentes. El patrón que se está asentando es reconocible al instante: una tarjeta por sesión de agente, columnas que reflejan su estado (preparado, ejecutando, esperando revisión, integrado), y —la pieza clave— un límite de WIP sobre la ejecución paralela: no lanzar más agentes de los que se pueden revisar, y revisar lo terminado antes de arrancar trabajo nuevo.

La razón de fondo conecta con todo el bloque B: cuando los agentes multiplican la capacidad de hacer, el cuello de botella del sistema se desplaza a la validación humana. La persona que revisa se convierte en el estado limitante del flujo, y todo lo que esta guía enseña sobre cuellos de botella, WIP y envejecimiento aplica literalmente: trabajo de agente sin revisar es inventario que envejece, y paralelizar agentes sin límite es la versión 2026 de empezar mucho y terminar poco.

**La lección de fondo.** La IA no jubila a Kanban: lo vuelve más necesario. Cuanto más barato es producir trabajo, más valioso es el sistema que decide cuánto entra, lo hace visible y garantiza que algo humano lo valida antes de contar como terminado.

### 17.3 Gobernanza como condición

En ambas direcciones, la adopción sería comparte tres exigencias. Supervisión humana proporcional al riesgo: las decisiones de flujo con coste de error alto —priorizar el trabajo de otros, comprometer fechas, descartar demanda— permanecen en manos humanas. Trazabilidad: toda acción de la IA sobre el sistema queda registrada y es explicable; un tablero donde las tarjetas se mueven sin que nadie sepa por qué ha dejado de ser un punto de verdad. Y reversibilidad: las automatizaciones se introducen como los experimentos del capítulo 9, con hipótesis, métrica y vuelta atrás. Este es, además, el capítulo del mapa con fecha de caducidad más corta: lo aquí descrito refleja junio de 2026 y obligará a revisión frecuente.

#### Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- Distinguir los tres niveles de IA aplicada al flujo: analítica predictiva, automatización asistida y agentes autónomos.
- Razonar qué decisiones de flujo conviene delegar y cuáles requieren supervisión humana.
- Aplicar los conceptos de Kanban (WIP, cuello de botella, revisión como estado) al gobierno del trabajo de agentes de IA.

#### Errores y antipatronos frecuentes

**Delegar decisiones de flujo sin trazabilidad ni gobernanza.** Un sistema cuyo estado cambia por decisiones inexplicables deja de ser gestionable: la visibilidad que Kanban construye se destruye por dentro.

**Confundir automatización de tareas con mejora del flujo.** Acelerar pasos sueltos no mejora el sistema si el cuello de botella está en otra parte; la pregunta sigue siendo dónde se atasca el valor, no qué se puede automatizar.

**Paralelizar agentes sin límite de WIP de revisión.** Lanzar más trabajo de agente del que las personas pueden validar acumula inventario sin revisar: mucho movimiento, poco terminado de verdad.

#### Para profundizar

- [AI-powered Kanban boards \(2026\)](#)
- [Tableros Kanban para sesiones de agentes \(2026\)](#)

---

© 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.