

La Guía Oficial de K-Agile v.2.0

La Guía Definitiva de K-Agile: Las Reglas del Juego para una Inteligencia Artificial Confiable, Explicable y Decisional

1. Introducción y propósito.

K-Agile (Knowledge-Centric Agile) es un marco de trabajo diseñado para guiar el desarrollo, la evolución y el mantenimiento de **sistemas complejos intensivos en conocimiento**, en particular **soluciones avanzadas de inteligencia artificial, inteligencia de decisiones (*Decision Intelligence*) y sistemas de apoyo a la decisión**, desarrollados en contextos de investigación y desarrollo (I+D).

K-Agile se **fundamenta en los principios de la filosofía Agile** —iteración, empirismo, inspección y adaptación continua— y adopta la **estructura operativa de la metodología Scrum** (roles, eventos, artefactos y ciclos temporales). Sobre esta base, K-Agile introduce una extensión estructural crítica: **el conocimiento explícito y trazable como activo central del proceso de desarrollo**, alrededor del cual coevolucionan los modelos, los algoritmos y los mecanismos de decisión.

Esta guía define de forma **normativa y exhaustiva** los principios, roles, eventos, artefactos y reglas que constituyen K-Agile. Su objetivo es proporcionar un marco **técnicamente riguroso, verificable y coherente** que permita a organizaciones y equipos **gestionar explícitamente la incertidumbre científica, técnica y decisional** inherente al desarrollo de sistemas de IA, donde el valor no reside únicamente en la funcionalidad, sino en la **calidad de las decisiones que el sistema es capaz de generar**.

En K-Agile, el desarrollo se orienta explícitamente a la construcción de **capacidades de decisión**, entendidas como la integración de:

- **modelos de conocimiento** (hipótesis, reglas, restricciones, ontologías),
- **solvers y motores de razonamiento** (optimizadores, planificadores, sistemas híbridos IA-OR),
- **funciones objetivo y criterios de evaluación** que formalizan el concepto de valor,
- **y mecanismos de explicabilidad y trazabilidad**, que permiten comprender por qué una decisión es tomada y bajo qué supuestos.

K-Agile trata los **solvers** como artefactos cognitivos evolucionables, no como componentes estáticos. Cada iteración contribuye a refinar:

- la formulación del problema,



- las funciones objetivo,
- los compromisos (*trade-offs*),
- y los límites de aplicabilidad de las decisiones generadas.

La **explicabilidad** no se considera un requisito posterior, sino una propiedad estructural del sistema. K-Agile exige que toda capacidad de decisión pueda trazarse hasta el conocimiento que la justifica, las evidencias que la validan y los criterios que la gobiernan, permitiendo auditoría, control y confianza operacional.

Asimismo, K-Agile integra explícitamente el concepto de **inteligencia colectiva**, entendida como la combinación estructurada de:

- conocimiento humano (expertos, usuarios, decisores),
- conocimiento algorítmico (modelos, solvers, aprendizaje automático),
- y conocimiento contextual (datos, restricciones operativas, señales del entorno).

El marco facilita que esta inteligencia colectiva se formalice, se itere y se mejore de manera sistemática, evitando que el conocimiento crítico permanezca tácito o aislado en individuos o modelos opacos.

K-Agile no prescribe técnicas concretas de aprendizaje automático, optimización, simulación o razonamiento. Dichas técnicas se integran dentro del marco en función del **conocimiento que deba generarse, validarse y operacionalizarse en cada Sprint**, manteniendo una trazabilidad directa entre hipótesis, evidencias, decisiones algorítmicas y comportamiento observable del sistema.

La aplicación parcial del marco, la omisión de alguno de sus elementos fundamentales o su reinterpretación arbitraria **no constituye K-Agile**, del mismo modo que una aplicación parcial de Scrum no constituye Scrum.

2. Definición de K-Agile.

K-Agile (Knowledge-Centric Agile) es un **marco de trabajo empírico** mediante el cual equipos multidisciplinares abordan **problemas complejos y adaptativos** a través de la **generación, formalización, validación y operacionalización de conocimiento**, de forma **iterativa e incremental**.

K-Agile se basa en los principios de la **filosofía Agile** y adopta la estructura fundamental de la **metodología Scrum**, incluyendo el uso de roles definidos, eventos formales, artefactos explícitos y ciclos temporales fijos. Sobre esta base, K-Agile introduce una extensión esencial: **el conocimiento explícito y trazable como unidad central de progreso y de valor**.

El objetivo de K-Agile es **transformar conocimiento explícito y validado en capacidades operativas de decisión**, entendidas como la habilidad de un sistema para evaluar alternativas, optimizar resultados y actuar de acuerdo con **criterios formales**, tales como funciones objetivo, restricciones, reglas, modelos y mecanismos de razonamiento. Estas capacidades se materializan mediante **solvers, motores de decisión y componentes algorítmicos**, cuya evolución está gobernada por el conocimiento que los sustenta.

En K-Agile, el progreso no se mide por la cantidad de funcionalidad entregada ni por el volumen de trabajo realizado, sino por la **reducción demostrable de incertidumbre relevante** y por la **mejora verificable de la calidad de las decisiones** que el sistema es capaz de generar. Cada iteración debe producir conocimiento que sea explícito, inspeccionable y utilizable, o bien descartar conocimiento previamente asumido mediante evidencia.

K-Agile exige que toda capacidad de decisión sea **explicable y trazable**, de modo que el comportamiento del sistema pueda vincularse de forma directa a:

- el conocimiento que lo justifica,
- las evidencias que lo validan,
- y los criterios que gobiernan la decisión.

La **explicabilidad** no se considera un atributo opcional, sino una propiedad estructural del marco, necesaria para garantizar control, confianza y uso responsable de sistemas de inteligencia artificial y de inteligencia de decisiones.

Asimismo, K-Agile reconoce explícitamente la **inteligencia colectiva** como un componente esencial del sistema. El marco está diseñado para integrar de manera sistemática el conocimiento humano (expertos, usuarios, decisores), el conocimiento algorítmico (modelos, solvers, aprendizaje automático) y el conocimiento contextual (datos, restricciones operativas y señales del entorno), evitando que el conocimiento crítico permanezca tácito, aislado u opaco.

K-Agile no prescribe técnicas concretas de investigación, aprendizaje automático, optimización, simulación o razonamiento. Dichas técnicas se seleccionan y emplean dentro del marco en función del **conocimiento que deba generarse, validarse y operacionalizarse en cada iteración**, manteniendo coherencia entre investigación, desarrollo tecnológico y validación práctica.

K-Agile es:

- **ligero en estructura**, para permitir adaptación continua en entornos de alta incertidumbre;
- **explícito en conocimiento**, garantizando trazabilidad entre hipótesis, evidencias y decisiones;
- **empírico en su funcionamiento**, basado en inspección frecuente y adaptación informada;
- **y riguroso en su disciplina metodológica**, exigiendo cumplimiento íntegro de sus roles, eventos, artefactos y reglas.

La aplicación parcial del marco, la omisión de alguno de sus elementos fundamentales o su reinterpretación arbitraria **no constituye K-Agile**, del mismo modo que una aplicación parcial de Scrum no constituye Scrum.

3. Teoría de K-Agile.

K-Agile se fundamenta en una **teoría empírica del conocimiento aplicado**, diseñada específicamente para entornos en los que la principal fuente de complejidad no es la implementación técnica, sino la **incertidumbre epistemológica** asociada a la comprensión del problema, a la formulación de hipótesis y a la toma de decisiones.

En este contexto, K-Agile adopta y extiende el empirismo clásico sobre el que se sustentan Agile y Scrum. El empirismo en K-Agile establece que el conocimiento válido surge de la experiencia, pero añade una condición esencial: **solo el conocimiento explícito, inspeccionable y validado puede gobernar decisiones operativas**.

En K-Agile, el conocimiento no se considera un subproducto del desarrollo ni una consecuencia indirecta de la implementación. Por el contrario, se define como el **objeto central de inspección, validación y adaptación**. El marco asume que los sistemas intensivos en conocimiento —como los sistemas de inteligencia artificial, optimización o inteligencia de decisiones— no pueden diseñarse de forma completamente predictiva, dado que:

- el propio problema evoluciona a medida que se investiga,
- las hipótesis iniciales pueden resultar incompletas o incorrectas,
- los criterios de decisión y las funciones objetivo se refinan progresivamente,
- y los contextos de uso influyen de manera determinante en la validez del conocimiento generado.

Como consecuencia, K-Agile rechaza los enfoques deterministas de diseño y adopta un modelo de **aprendizaje iterativo controlado**. El marco estructura el desarrollo en ciclos iterativos e incrementales, en los que cada Sprint constituye un **experimento epistemológico controlado**, con un objetivo cognitivo explícito, criterios de validación definidos y resultados inspeccionables.

Cada Sprint está orientado a:

- reducir una incertidumbre relevante,
- generar o refinar conocimiento explícito,
- y mejorar de forma verificable la capacidad de decisión del sistema.

Desde esta perspectiva, el progreso no se mide por el volumen de trabajo realizado, la cantidad de funcionalidad entregada o la velocidad de ejecución, sino por:

- la **reducción demostrable de incertidumbre crítica**,
- la **madurez del conocimiento operacionalizado**,
- y la **mejora objetiva de la calidad de las decisiones** que el sistema es capaz de generar.

K-Agile asume que el conocimiento evoluciona en distintos niveles de madurez y que no todo el conocimiento generado es inmediatamente utilizable. Por ello, el marco exige mecanismos explícitos para:

- diferenciar conocimiento exploratorio de conocimiento validado,
- evitar la toma de decisiones basadas en supuestos no contrastados,
- y mantener trazabilidad entre hipótesis, evidencias, modelos, solvers y decisiones.

Finalmente, la teoría de K-Agile reconoce que el conocimiento relevante para la toma de decisiones no es exclusivamente algorítmico. El marco se apoya en una concepción de **inteligencia colectiva**, en la que el conocimiento humano, el conocimiento algorítmico y el conocimiento contextual se integran de forma sistemática y explícita dentro del proceso empírico.

En conjunto, la teoría de K-Agile proporciona el fundamento conceptual que permite al marco **alinear investigación, desarrollo tecnológico y validación práctica** en entornos de alta incertidumbre, manteniendo el control epistemológico del sistema y garantizando que cada iteración contribuya a decisiones más informadas, explicables y fiables.

4. Principios fundamentales de K-Agile.

Los principios fundamentales de K-Agile son **normativos** y gobiernan todas las decisiones metodológicas, organizativas y técnicas. Su incumplimiento invalida la correcta aplicación del marco.

En K-Agile, los principios no se interpretan como valores aspiracionales. Se aplican como **restricciones de diseño** del proceso y como **criterios de conformidad** del marco. En particular, en sistemas de IA y *Decision Intelligence*, estos principios aseguran que el comportamiento del sistema permanezca gobernado por conocimiento explícito, evidencias verificables y límites de validez inspeccionables.

4.1 Centralidad del conocimiento.

El conocimiento explícito es el **principal activo** del sistema y el eje organizador del proceso de desarrollo.

En K-Agile, el conocimiento incluye, como mínimo:

- hipótesis y supuestos,
- modelos conceptuales y formales,
- criterios de decisión y funciones objetivo,
- restricciones, reglas y heurísticas,
- evidencias empíricas y resultados de validación,
- límites de aplicabilidad y niveles de confianza.

Toda actividad realizada dentro de K-Agile debe justificarse por su contribución directa o indirecta a:

- generar conocimiento nuevo,
- refinar conocimiento existente,
- validar conocimiento en contexto,
- o operacionalizar conocimiento en capacidades de decisión.

El trabajo que no aporta avance cognitivo verificable o que no reduce incertidumbre relevante se considera desperdicio metodológico.

4.2 Empirismo riguroso.

Las decisiones se toman exclusivamente sobre la base de conocimiento derivado de la experiencia y validado mediante evidencias.

En sistemas de IA, el empirismo riguroso implica que:

- ningún comportamiento del sistema se acepta como válido sin evidencia,
- ningún resultado “que funciona” se considera suficiente sin validación,
- y ninguna decisión de diseño se mantiene por inercia si las evidencias la contradicen.

Los supuestos no contrastados solo son aceptables cuando:

- se formulan explícitamente como hipótesis,
- se registran como incertidumbre,
- y se planifica su contrastación mediante criterios definidos.

El empirismo riguroso obliga a separar de forma explícita:

- conocimiento exploratorio,
- conocimiento validado,
- y conocimiento operacionalizado.

4.3 Trazabilidad epistemológica.

Toda capacidad del sistema debe poder trazarse hasta conocimiento explícito documentado en la Base de Conocimiento.

La trazabilidad epistemológica exige que cualquier decisión o recomendación generada por el sistema sea vinculable, como mínimo, a:

- el modelo o formulación que la gobierna (incluida la función objetivo y restricciones),
- las evidencias que justifican dicho modelo,
- y los supuestos y límites bajo los cuales se considera válido.

En sistemas de IA y *Decision Intelligence*, la ausencia de trazabilidad constituye un riesgo crítico, ya que impide:

- auditoría,
- control operativo,
- análisis de fallo,
- y gobernanza responsable.

4.4 Co-evolución.

El conocimiento, los algoritmos, los mecanismos de decisión y los contextos de validación evolucionan de forma conjunta e inseparable.

En K-Agile, no se optimiza un solver sin revisar:

- la formulación del problema,
- los criterios y funciones objetivo,

- la definición de calidad de decisión,
- y la validez contextual de las evidencias utilizadas.

La co-evolución implica que todo cambio relevante en cualquiera de los siguientes elementos exige inspección y posible adaptación de los demás:

- conocimiento explícito,
- modelos y datos,
- solvers y estrategias de decisión,
- métricas y procedimientos de validación,
- y supuestos operativos.

4.5 Validación contextual.

El conocimiento solo se considera válido cuando ha sido contrastado en contextos representativos de uso real o simulado.

K-Agile rechaza la validación aislada del contexto de decisión. En sistemas de IA, métricas abstractas no son suficientes si no representan las condiciones reales de operación, incluyendo:

- restricciones del entorno,
- distribución de casos,
- interacción humano-sistema,
- y coste de error o incertidumbre.

La validación contextual debe producir evidencias que permitan responder, de forma explícita:

- en qué condiciones el conocimiento es aplicable,
- con qué nivel de confianza,
- y cuáles son sus límites o modos de fallo.

4.6 Explicitación sistemática.

Todo supuesto implícito relevante debe hacerse explícito tan pronto como sea identificado.

La explicitación sistemática es un requisito previo para:

- inspección,
- validación,
- trazabilidad,
- y reutilización del conocimiento.

En sistemas de IA, esto incluye hacer explícitos supuestos sobre:

- calidad y sesgos de datos,

- estabilidad de distribuciones,
- definición de “éxito” y funciones objetivo,
- tolerancias a error,
- y reglas de decisión humanas integradas (inteligencia colectiva).

El conocimiento tácito que permanece implícito reduce la capacidad del sistema para ser controlado y aumenta el riesgo operacional.

4.7 Disciplina metodológica.

Los roles, eventos, artefactos y reglas definidos en K-Agile son inmutables dentro de la versión vigente del marco.

K-Agile no se adapta mediante omisiones o modificaciones informales. La disciplina metodológica exige que:

- los eventos existan y produzcan sus resultados normativos,
- los artefactos se mantengan transparentes y actualizados,
- y las definiciones (incluida la Definición de “Known”) se apliquen de manera consistente.

Cualquier evolución del marco solo puede realizarse mediante versionado formal de esta guía. Las variaciones no documentadas no constituyen versiones válidas de K-Agile.

4.8 Co-creación e inteligencia colectiva.

K-Agile se fundamenta en el principio de **co-creación e inteligencia colectiva**, entendidas como la integración explícita y sistemática del conocimiento humano, algorítmico y contextual en el proceso de generación y validación de decisiones.

En K-Agile, el conocimiento relevante para la toma de decisiones no reside exclusivamente en modelos, datos o solvers. Incluye de manera explícita:

- el conocimiento experto de dominio,
- la experiencia de usuarios y decisores,
- los juicios humanos incorporados en criterios, restricciones y funciones objetivo,
- y el conocimiento algorítmico derivado de modelos y aprendizaje automático.

La co-creación exige que este conocimiento colectivo:

- se haga explícito,
- se formalice cuando sea posible,
- se valide mediante evidencias,
- y se integre de forma trazable en las capacidades de decisión del sistema.

K-Agile rechaza los enfoques en los que el conocimiento humano permanece tácito, no documentado o externalizado fuera del sistema. El conocimiento no explicitado no puede ser inspeccionado, validado ni gobernado, y por tanto incrementa el riesgo decisional.

La inteligencia colectiva se considera un **activo estructural del sistema**, no un mecanismo informal de coordinación. Su correcta integración es condición necesaria para:

- la explicabilidad de las decisiones,
- la adaptación a contextos cambiantes,
- y la gobernanza responsable de sistemas de inteligencia artificial.

5. El Equipo K-Agile.

El **Equipo K-Agile** es la unidad fundamental mediante la cual K-Agile opera. Es responsable colectivo de **generar, formalizar, validar y operacionalizar conocimiento**, y de transformarlo en **capacidades operativas de decisión** explicables, trazables y contextualmente válidas.

El Equipo K-Agile es responsable último de que el sistema desarrollado:

- incorpore conocimiento explícito y validado,
- reduzca incertidumbre relevante,
- integre inteligencia humana y algorítmica,
- y mejore de forma verificable la calidad de las decisiones que genera.

El Equipo K-Agile está compuesto por **cuatro roles definidos e inmutables**:

- Knowledge Owner
- Equipo de Investigación y Desarrollo
- Decision Wizard
- K-Agile Facilitator

Estos roles existen para garantizar la correcta aplicación del empirismo cognitivo, la integración de inteligencia colectiva y la disciplina metodológica del marco.

5.1 Naturaleza del Equipo K-Agile.

El Equipo K-Agile es **autoorganizado y multidisciplinar**.

- *Autoorganizado* significa que el equipo decide internamente **cómo** transformar Ítems de Conocimiento en Incrementos de Conocimiento Operacional, sin dirección externa sobre los medios técnicos, cognitivos o decisionales.
- *Multidisciplinar* significa que el equipo dispone colectivamente de todas las competencias necesarias para abordar el problema, incluyendo investigación científica, ingeniería, optimización, análisis de decisión y comprensión del comportamiento humano.

Ninguna entidad externa dirige cómo el Equipo K-Agile realiza su trabajo. Las partes interesadas pueden influir en **qué conocimiento es valioso**, pero nunca en **cómo se genera, valida o formaliza**.

5.2 Responsabilidad colectiva.

El Equipo K-Agile es responsable, de forma colectiva, de:

- la calidad del conocimiento generado;
- la validez y suficiencia de las evidencias producidas;
- la trazabilidad entre conocimiento, solvers y decisiones;
- la integración coherente de conocimiento humano y algorítmico;
- y la adecuación del sistema a su contexto real de uso.

La especialización funcional no exime de responsabilidad colectiva. No existen propietarios individuales del resultado final.

5.3 El Knowledge Owner.

El **Knowledge Owner** es responsable de **maximizar el valor cognitivo y decisional del sistema**.

Es la única persona responsable del **Knowledge Backlog** y de asegurar que el conocimiento priorizado responde a las **incertidumbres más relevantes** del problema y a los objetivos estratégicos.

Sus responsabilidades incluyen, como mínimo:

- definir y priorizar Ítems de Conocimiento según su impacto en la decisión;
- explicitar hipótesis, supuestos y objetivos cognitivos;
- asegurar la alineación entre conocimiento, estrategia y contexto de uso;
- decidir la cancelación de un Sprint cuando el Objetivo Cognitivo pierde validez;
- mantener coherencia global del conocimiento generado.

El Knowledge Owner **no decide cómo se implementan modelos, solvers o algoritmos**. Su responsabilidad es **qué conocimiento debe existir y por qué**.

5.4 El Equipo de Investigación y Desarrollo.

El **Equipo de Investigación y Desarrollo** está compuesto por los profesionales que realizan el trabajo necesario para producir Incrementos de Conocimiento Operacional.

Este equipo puede incluir perfiles de:

- investigación científica,
- ingeniería de datos y modelos,
- aprendizaje automático,
- optimización y simulación,

- diseño experimental,
- validación técnica y decisional,
- **psicología, neurociencia y antropología**, cuando el comportamiento humano influye en las decisiones del sistema.

Cuando el sistema interactúa con personas o incorpora juicio humano, el conocimiento sobre comportamiento humano se considera **conocimiento técnico del sistema**, no información auxiliar.

El Equipo de Investigación y Desarrollo es responsable de:

- transformar Ítems de Conocimiento en conocimiento explícito;
- diseñar y ejecutar experimentos técnicos y humanos;
- interpretar resultados cognitivos y conductuales;
- producir evidencias alineadas con la Definición de “Known”;
- mantener la coherencia técnica y epistemológica del sistema.

5.5 El Decision Wizard

El **Decision Wizard** es responsable de **maximizar el rendimiento decisional del conocimiento experto**, transformándolo en **sabiduría operacional**.

La sabiduría operacional se entiende como la capacidad del sistema para:

- evaluar alternativas,
- gestionar *trade-offs* complejos,
- optimizar resultados bajo restricciones,
- y tomar decisiones consistentes, explicables y responsables.

Las responsabilidades del Decision Wizard incluyen, como mínimo:

- diseñar y gobernar **funciones objetivo** y criterios de evaluación;
- definir compromisos entre objetivos técnicos, humanos y contextuales;
- seleccionar, diseñar o adaptar **solvers y motores de decisión**;
- integrar conocimiento experto humano en estructuras formales de decisión;
- incorporar restricciones cognitivas, culturales y psicológicas;
- garantizar la **explicabilidad y trazabilidad** de las decisiones.

El Decision Wizard no prioriza qué conocimiento debe generarse (Knowledge Owner) ni dirige la ejecución general del trabajo (Equipo de Investigación y Desarrollo). Su responsabilidad es **cómo el conocimiento existente se convierte en decisiones óptimas y viables**.

5.6 El K-Agile Facilitator.

El **K-Agile Facilitator** es responsable de asegurar que K-Agile se entiende y se aplica correctamente.

Sirve al Equipo K-Agile mediante:

- la protección de los principios y del empirismo cognitivo;
- la facilitación efectiva de los eventos K-Agile;
- la eliminación de impedimentos metodológicos;
- la promoción de la transparencia del conocimiento y del progreso cognitivo.

El K-Agile Facilitator **no toma decisiones de contenido** ni define soluciones técnicas. Su función es **garantizar la integridad metodológica del marco**.

5.7 Inteligencia colectiva e interacción entre roles.

Los roles del Equipo K-Agile son **complementarios y no jerárquicos**:

- el Knowledge Owner define **qué conocimiento es valioso**;
- el Equipo de Investigación y Desarrollo define **cómo se genera y valida**;
- el Decision Wizard define **cómo el conocimiento se transforma en decisiones**;
- el K-Agile Facilitator asegura **que el marco se aplica correctamente**.

Esta interacción permite integrar **inteligencia colectiva**, combinando conocimiento humano, algorítmico y contextual de forma explícita, trazable y gobernable.

5.8 Tamaño y composición del equipo.

El Equipo K-Agile debe ser lo suficientemente pequeño como para mantener comunicación efectiva y lo suficientemente grande como para contener todas las competencias necesarias.

Equipos excesivamente grandes reducen la transparencia cognitiva y dificultan la inspección.

5.9 Conformidad del Equipo K-Agile.

Un equipo solo puede considerarse un Equipo K-Agile si:

- existen explícitamente los cuatro roles definidos;
- los roles ejercen sus responsabilidades normativas;
- el equipo opera de forma autoorganizada;
- y las decisiones del sistema están gobernadas por conocimiento explícito, validado y formalizado.

La adopción nominal de los roles sin cumplimiento real de sus responsabilidades **no constituye K-Agile**.

6. El Knowledge Owner.

El **Knowledge Owner** es responsable de **maximizar el valor cognitivo y decisional del sistema**. Es la autoridad última sobre **qué conocimiento debe existir, por qué es relevante y en qué orden debe generarse o validarse**.

El Knowledge Owner es la **única persona responsable del Knowledge Backlog** y actúa como custodio de la coherencia epistemológica y estratégica del sistema a lo largo de su evolución.

Este rol existe para garantizar que el esfuerzo del equipo se dirige a reducir las **incertidumbres más relevantes**, y no simplemente a producir resultados técnicos o funcionales aislados.

6.1 Propósito del rol.

El propósito del **Knowledge Owner** es asegurar que cada iteración del sistema contribuye de forma explícita a:

- generar conocimiento relevante para la toma de decisiones,
- reducir incertidumbre crítica,
- y aumentar la capacidad del sistema para actuar de manera informada, explicable y responsable.

En K-Agile, el valor no se define por la cantidad de funcionalidad entregada, sino por la **calidad del conocimiento operacionalizado**. El **Knowledge Owner** es responsable de proteger este principio.

6.2 Responsabilidades fundamentales.

Las responsabilidades del **Knowledge Owner** incluyen, como mínimo:

- definir qué conocimiento es relevante para el sistema en cada momento;
- priorizar **incertidumbres**, no tareas ni soluciones;
- formular y mantener Ítems de Conocimiento claros, explícitos y verificables;
- asegurar que los Ítems de Conocimiento reflejan hipótesis, supuestos y objetivos cognitivos bien definidos;
- mantener la coherencia estratégica entre conocimiento, contexto de uso y objetivos del sistema;
- decidir la cancelación de un Sprint cuando el Objetivo Cognitivo pierde validez;
- aceptar o rechazar Incrementos de Conocimiento Operacional en función de su impacto decisional y su alineación con la Definición de “Known”.

El **Knowledge Owner** no dirige la **ejecución técnica**, ni decide cómo se implementan modelos, solvers o algoritmos. Su responsabilidad es **orientar el aprendizaje del sistema**.

6.3 Gestión del Knowledge Backlog.

El **Knowledge Backlog** es la expresión operativa de la estrategia cognitiva del sistema. El Knowledge Owner es responsable de que este backlog sea:

- explícito,
- ordenado,
- comprensible,
- y orientado a reducción de incertidumbre.

Cada Ítem de Conocimiento debe, como mínimo:

- formular una pregunta, hipótesis o incertidumbre relevante;
- indicar el impacto esperado en la toma de decisiones;
- identificar supuestos clave;
- y proponer criterios preliminares de validación.

El orden del Knowledge Backlog refleja **prioridad cognitiva**, no esfuerzo técnico ni facilidad de implementación.

6.4 Relación con la inteligencia colectiva.

El Knowledge Owner es responsable de **integrar inteligencia colectiva** en el Knowledge Backlog.

Esto incluye identificar, capturar y formalizar conocimiento procedente de:

- expertos humanos de dominio,
- neurocientíficos, psicólogos y antropólogos cuando el comportamiento humano influye en la decisión;
- análisis de contexto operativo;
- y resultados emergentes del propio sistema.

El Knowledge Owner debe asegurar que este conocimiento:

- no permanezca tácito,
- se formule explícitamente,
- y sea inspeccionable y validable dentro del marco.

6.5 Relación con el Decision Wizard.

El Knowledge Owner y el **Decision Wizard** colaboran estrechamente, pero tienen responsabilidades distintas y no intercambiables.

- El Knowledge Owner define **qué conocimiento es necesario** para mejorar la decisión.
- El Decision Wizard define **cómo ese conocimiento se transforma en decisiones óptimas**.

El Knowledge Owner no diseña funciones objetivo ni solvers, pero debe comprender su impacto decisional para priorizar correctamente el conocimiento que los gobierna.

6.6 Autoridad y límites.

El Knowledge Owner tiene autoridad para:

- ordenar el Knowledge Backlog;
- clarificar o reformular Ítems de Conocimiento;
- cancelar un Sprint cuando el aprendizaje perseguido deja de ser relevante;
- rechazar Incrementos que no aportan valor cognitivo o decisional.

El Knowledge Owner **no tiene autoridad** para:

- asignar tareas al equipo;
- imponer soluciones técnicas;
- alterar eventos o artefactos del marco;
- ni sustituir el juicio empírico por decisiones jerárquicas.

6.7 Cualidades del Knowledge Owner.

El Knowledge Owner debe poseer:

- comprensión profunda del dominio del problema;
- capacidad para formular buenas preguntas y detectar incertidumbre relevante;
- pensamiento sistémico y decisional;
- criterio para distinguir conocimiento prometedor de conocimiento operativo;
- y habilidad para integrar perspectivas técnicas, humanas y contextuales.

No es necesario que sea un experto técnico en implementación, pero sí un **experto en el conocimiento que gobierna la decisión**.

6.8 Conformidad del rol.

Un Knowledge Owner cumple su rol únicamente cuando:

- el Knowledge Backlog refleja incertidumbres reales y relevantes;
- los Ítems de Conocimiento son claros y verificables;
- las prioridades se basan en impacto decisional;
- y el aprendizaje del sistema es coherente a lo largo del tiempo.

La adopción nominal del rol sin el ejercicio real de estas responsabilidades **no constituye K-Agile**.

7. El Equipo de Investigación y Desarrollo.

El **Equipo de Investigación y Desarrollo** es responsable de realizar todo el trabajo necesario para producir **Incrementos de Conocimiento Operacional**, es decir, conocimiento explícito, validado y utilizable para la toma de decisiones reales.

Este equipo constituye el **núcleo operativo** de K-Agile y es el principal responsable de que el sistema evolucione de manera empírica, rigurosa y controlada.

7.1 Naturaleza del Equipo de Investigación y Desarrollo.

El Equipo de Investigación y Desarrollo es **autoorganizado, multifuncional y responsable colectivamente del resultado**.

- *Autoorganizado* significa que el equipo decide internamente cómo abordar los Ítems de Conocimiento seleccionados para un Sprint, sin dirección externa sobre métodos, técnicas o soluciones.
- *Multifuncional* significa que el equipo dispone colectivamente de todas las capacidades necesarias para generar, validar y operacionalizar conocimiento.
- *Responsable colectivamente* significa que el resultado del Sprint pertenece al equipo en su conjunto, independientemente de la especialización individual de sus miembros.

No existen jerarquías internas que sustituyan el juicio empírico del equipo.

7.2 Alcance de responsabilidad.

El Equipo de Investigación y Desarrollo es responsable de:

- transformar Ítems de Conocimiento en conocimiento explícito;
- diseñar y ejecutar experimentos técnicos, humanos y contextuales;
- implementar y refinar modelos, algoritmos, prototipos y simulaciones;
- generar evidencias alineadas con los criterios de validación acordados;
- evaluar hipótesis y descartar conocimiento incorrecto o no aplicable;
- producir Incrementos de Conocimiento Operacional coherentes y trazables.

El equipo es responsable tanto del **conocimiento generado** como del **conocimiento descartado**, siempre que este descarte esté respaldado por evidencia.

7.3 Composición y perfiles.

El Equipo de Investigación y Desarrollo puede estar compuesto por perfiles diversos, incluyendo, cuando sea necesario:

- investigadores científicos y de dominio;
- ingenieros de datos, modelos y sistemas;
- especialistas en aprendizaje automático y optimización;

- expertos en simulación y análisis experimental;
- analistas de decisión y validación;
- **psicólogos, neurocientíficos y antropólogos**, cuando el comportamiento humano influye en las decisiones del sistema.

En K-Agile, el conocimiento sobre el comportamiento humano se considera **conocimiento técnico del sistema** cuando afecta a la calidad, comprensión o uso de las decisiones generadas.

7.4 Relación con el conocimiento humano.

Cuando el sistema interactúa con personas o incorpora juicio humano, el Equipo de Investigación y Desarrollo es responsable de:

- formular hipótesis sobre comportamiento humano;
- diseñar experimentos y validaciones con participantes reales o simulados;
- interpretar resultados conductuales de manera rigurosa;
- convertir observaciones humanas en conocimiento explícito y validable.

El equipo debe evitar que el conocimiento humano permanezca implícito, informal o no trazable.

7.5 Relación con el Knowledge Owner.

El Equipo de Investigación y Desarrollo colabora estrechamente con el **Knowledge Owner**, pero mantiene independencia operativa.

- El Knowledge Owner define **qué conocimiento es prioritario**.
- El Equipo de Investigación y Desarrollo decide **cómo se genera y valida**.

El equipo puede cuestionar la formulación de Ítems de Conocimiento, proponer reformulaciones o identificar incertidumbres no previstas, pero no altera unilateralmente la prioridad del Knowledge Backlog.

7.6 Relación con el Decision Wizard.

El Equipo de Investigación y Desarrollo colabora con el **Decision Wizard** para:

- proporcionar conocimiento validado que alimente funciones objetivo y solvers;
- evaluar el comportamiento de los mecanismos de decisión;
- analizar el impacto de cambios en criterios, restricciones o trade-offs;
- detectar inconsistencias entre conocimiento generado y decisiones producidas.

El equipo no define funciones objetivo finales ni gobierna la sabiduría operacional, pero es responsable de suministrar la evidencia que las sustenta.

7.7 Autoorganización y toma de decisiones.

El Equipo de Investigación y Desarrollo decide internamente:

- cómo organizar su trabajo durante el Sprint;
- qué técnicas de investigación o implementación utilizar;
- cómo dividir o combinar tareas;
- y cómo responder a nueva información emergente.

Las decisiones se toman de manera empírica, basadas en evidencia y alineadas con el Objetivo Cognitivo del Sprint.

7.8 Responsabilidad sobre la calidad.

El Equipo de Investigación y Desarrollo es responsable de la **calidad epistemológica** del trabajo realizado.

Esto incluye asegurar que:

- las hipótesis están claramente formuladas;
- los experimentos son reproducibles cuando sea posible;
- las evidencias son suficientes y comprensibles;
- los límites de validez del conocimiento están explícitos;
- y la trazabilidad entre conocimiento y decisiones es mantenida.

La velocidad o el volumen de trabajo nunca justifican la degradación de la calidad cognitiva.

7.9 Conformidad del Equipo de Investigación y Desarrollo.

Un Equipo de Investigación y Desarrollo cumple su rol únicamente cuando:

- produce Incrementos de Conocimiento Operacional válidos;
- genera evidencia explícita y verificable;
- opera de forma autoorganizada;
- y mantiene coherencia técnica, cognitiva y contextual.

La adopción nominal del rol sin el ejercicio real de estas responsabilidades **no constituye K-Agile**.

8. El Decision Wizard.

El **Decision Wizard** es responsable de **transformar conocimiento explícito y validado en decisiones óptimas, explicables y contextualmente viables**. Es el custodio de la **sabiduría operacional** del sistema.

En K-Agile, la sabiduría operacional se define como la capacidad del sistema para **tomar buenas decisiones de forma consistente**, integrando conocimiento experto, evidencia empírica, restricciones del contexto y criterios de valor explícitos.

El Decision Wizard existe para asegurar que el sistema **no solo sabe**, sino que **decide bien**.

8.1 Propósito del rol.

El propósito del Decision Wizard es **maximizar el rendimiento decisional del conocimiento disponible**.

Esto implica garantizar que el conocimiento generado por el sistema se traduzca en:

- criterios de decisión bien definidos,
- funciones objetivo coherentes,
- *trade-offs* explícitos,
- y mecanismos de decisión (solvers) alineados con los objetivos reales del sistema.

El Decision Wizard gobierna el paso crítico entre **conocimiento y acción**.

8.2 Ámbito de responsabilidad.

El Decision Wizard es responsable de:

- diseñar, formalizar y mantener **funciones objetivo**;
- definir y gobernar **criterios de optimización y *trade-offs*** entre objetivos en conflicto;
- seleccionar, diseñar o adaptar **solvers y motores de decisión**;
- integrar conocimiento experto humano en estructuras formales de decisión;
- incorporar restricciones técnicas, humanas, cognitivas, éticas y contextuales;
- asegurar que las decisiones producidas son **explicables, trazables y auditables**;
- evaluar el impacto decisional del conocimiento generado.

El Decision Wizard **no prioriza qué conocimiento debe generarse** (Knowledge Owner) ni ejecuta el trabajo técnico completo (Equipo de Investigación y Desarrollo). Su responsabilidad es **cómo el conocimiento existente se convierte en decisiones de alta calidad**.

8.3 Funciones objetivo y criterios de valor.

Las **funciones objetivo** son el principal instrumento de la sabiduría operacional.

El Decision Wizard es responsable de asegurar que dichas funciones:

- formalizan explícitamente el concepto de valor del sistema;
- reflejan compromisos reales entre objetivos en conflicto;
- incorporan penalizaciones y restricciones relevantes;
- y pueden evolucionar a medida que se genera nuevo conocimiento.

En K-Agile, una función objetivo nunca es definitiva. Es una **hipótesis decisional** que debe ser inspeccionada, validada y refinada empíricamente.

8.4 Solvers y mecanismos de decisión.

El Decision Wizard gobierna los **solvers** como artefactos cognitivos, no como componentes puramente técnicos.

Esto incluye:

- seleccionar paradigmas de resolución adecuados (optimización, heurísticas, planificación, sistemas híbridos);
- evaluar la adecuación de un solver al tipo de decisión;
- ajustar estrategias de resolución en función del contexto;
- y garantizar coherencia entre solver, función objetivo y conocimiento subyacente.

Un solver que produce resultados correctos pero no explicables, no trazables o no alineados con el valor del sistema **no es aceptable en K-Agile**.

8.5 Integración del conocimiento humano e inteligencia colectiva.

El Decision Wizard es el principal responsable de **integrar inteligencia colectiva** en las decisiones del sistema.

Esto implica transformar conocimiento humano —incluido el aportado por psicólogos, neurocientíficos y antropólogos— en:

- restricciones humanas explícitas,
- criterios de aceptación,
- penalizaciones cognitivas,
- y límites de aplicabilidad decisional.

El Decision Wizard asegura que el conocimiento humano **no quede como opinión externa**, sino que se incorpore de forma formal y trazable en los mecanismos de decisión.

8.6 Explicabilidad y trazabilidad.

La **explicabilidad** es una responsabilidad estructural del Decision Wizard.

Toda decisión generada por el sistema debe poder explicarse en términos de:

- función objetivo aplicada,
- criterios considerados,
- conocimiento utilizado,
- y supuestos activos.

El Decision Wizard garantiza que esta trazabilidad se mantenga incluso cuando se emplean modelos complejos o híbridos.

8.7 Relación con el Knowledge Owner.

El Decision Wizard y el **Knowledge Owner** colaboran de forma continua, pero con responsabilidades claramente diferenciadas:

- El Knowledge Owner define **qué conocimiento es relevante** para mejorar la decisión.
- El Decision Wizard define **cómo ese conocimiento se usa para decidir**.

El Decision Wizard puede identificar carencias de conocimiento o inconsistencias decisionales, y comunicarlás al Knowledge Owner para su priorización en el Knowledge Backlog.

8.8 Relación con el Equipo de Investigación y Desarrollo.

El Decision Wizard colabora con el **Equipo de Investigación y Desarrollo** para:

- validar empíricamente el comportamiento de los solvers;
- evaluar el impacto de cambios en funciones objetivo;
- analizar sensibilidad y robustez de las decisiones;
- y refinar mecanismos de decisión a partir de nueva evidencia.

El Decision Wizard **no sustituye** al Equipo de Investigación y Desarrollo, pero gobierna el uso decisional del conocimiento que este produce.

8.9 Cualidades del Decision Wizard.

El Decision Wizard debe poseer:

- pensamiento sistémico y decisional;
- comprensión profunda del dominio del problema;
- capacidad para formalizar conocimiento experto;
- criterio para gestionar *trade-offs* complejos;
- sensibilidad a factores humanos, cognitivos y contextuales;
- y habilidad para explicar decisiones complejas de forma comprensible.

No es necesario que implemente todo el sistema, pero sí que **comprenda profundamente cómo decide**.

8.10 Conformidad del rol.

Un Decision Wizard cumple su rol únicamente cuando:

- las funciones objetivo son explícitas y comprensibles;
- los solvers están alineados con el valor del sistema;
- los *trade-offs* están documentados y gobernados;
- las decisiones son explicables y trazables;

- y la sabiduría operacional mejora de Sprint en Sprint.

La adopción nominal del rol sin el ejercicio real de estas responsabilidades **no constituye K-Agile**.

9. El K-Agile Facilitator.

El K-Agile Facilitator es responsable de **asegurar la correcta comprensión, adopción y aplicación de K-Agile** como marco empírico centrado en el conocimiento y la decisión.

Este rol existe para **proteger la integridad metodológica del marco**, garantizar que el empirismo cognitivo se aplique de forma efectiva y que el sistema no derive hacia prácticas predictivas, opacas o centradas únicamente en la entrega técnica.

El K-Agile Facilitator **no es un gestor de personas ni un director técnico**. Su función es servir al Equipo K-Agile asegurando que el marco se utilice como fue definido.

9.1 Propósito del rol.

El propósito del K-Agile Facilitator es:

- habilitar la generación efectiva de conocimiento explícito;
- garantizar la transparencia cognitiva de los artefactos;
- asegurar la inspección y adaptación continuas;
- y proteger al equipo frente a desviaciones metodológicas.

En K-Agile, el Facilitator es el **custodio del proceso empírico**, no del contenido.

9.2 Ámbito de responsabilidad.

El K-Agile Facilitator es responsable de:

- asegurar que los **eventos K-Agile** ocurren y cumplen su propósito normativo;
- facilitar la correcta ejecución de Sprint Planning, Daily, Review y Retrospective;
- promover la transparencia del Knowledge Backlog, la Base de Conocimiento y el progreso cognitivo;
- identificar y eliminar impedimentos metodológicos y organizativos;
- apoyar al equipo en la adopción del empirismo cognitivo;
- y fomentar la mejora continua del uso del marco.

El K-Agile Facilitator **no toma decisiones sobre qué conocimiento priorizar**, ni sobre cómo se diseñan solvers o funciones objetivo.

9.3 Protección del empirismo cognitivo.

Una responsabilidad clave del K-Agile Facilitator es **proteger el empirismo cognitivo** frente a prácticas que lo degraden.

Esto incluye intervenir cuando:



- se toman decisiones basadas en suposiciones no explicitadas;
- se presenta actividad como si fuera aprendizaje;
- se oculta incertidumbre para aparentar progreso;
- se evita descartar hipótesis incorrectas;
- o se sustituyen evidencias por opiniones de autoridad.

El Facilitator actúa como **mecanismo de defensa del aprendizaje real**.

9.4 Facilitación de eventos.

El K-Agile Facilitator facilita los eventos del marco asegurando que:

- la **Planificación del Sprint** define objetivos cognitivos claros y criterios de validación explícitos;
- el **Daily K-Agile** inspecciona conocimiento y no solo tareas;
- la **Revisión del Sprint** evalúa conocimiento y decisiones, no demos superficiales;
- la **Retrospectiva del Sprint** mejora la calidad del aprendizaje, no solo la dinámica del equipo.

El Facilitator vela por que cada evento produzca sus **resultados normativos**.

9.5 Gestión de impedimentos.

El K-Agile Facilitator ayuda a eliminar impedimentos que afectan a:

- la generación de conocimiento;
- la validación empírica;
- la trazabilidad entre conocimiento y decisión;
- o la correcta aplicación del marco.

Los impedimentos pueden ser técnicos, organizativos, culturales o metodológicos. El Facilitator no resuelve todos directamente, pero **asegura que se hagan visibles y se aborden**.

9.6 Relación con los demás roles.

El K-Agile Facilitator colabora estrechamente con todos los roles del Equipo K-Agile:

- con el **Knowledge Owner**, para asegurar que el Knowledge Backlog es transparente y comprensible;
- con el **Equipo de Investigación y Desarrollo**, para proteger su autoorganización y foco empírico;
- con el **Decision Wizard**, para garantizar que la sabiduría operacional se gobierna de forma explícita y trazable.

El Facilitator **no sustituye** a ninguno de estos roles ni interfiere en sus responsabilidades de contenido.

9.7 Autoridad y límites.

El K-Agile Facilitator tiene autoridad para:

- intervenir en eventos cuando se desvían de su propósito;
- señalar incumplimientos del marco;
- proponer mejoras en la aplicación de K-Agile.

El K-Agile Facilitator **no tiene autoridad** para:

- priorizar el Knowledge Backlog;
- imponer decisiones técnicas o decisionales;
- modificar roles, eventos o artefactos del marco;
- ni sustituir la evidencia por criterio personal.

9.8 Cualidades del K-Agile Facilitator.

El K-Agile Facilitator debe poseer:

- comprensión profunda del marco K-Agile;
- capacidad para detectar anti-patronos cognitivos;
- habilidades de facilitación y comunicación;
- criterio para distinguir aprendizaje real de actividad aparente;
- y sensibilidad para proteger la autoorganización del equipo.

No necesita ser experto en el dominio del problema, pero sí **experto en el proceso empírico**.

10. Artefactos de K-Agile.

Los artefactos de K-Agile representan **conocimiento, progreso cognitivo y capacidad de decisión**. Están diseñados para maximizar la transparencia, habilitar inspección rigurosa y soportar adaptación informada.

Todos los artefactos de K-Agile son **obligatorios**. La ausencia, degradación o falta de actualización de un artefacto invalida la transparencia del marco y compromete la validez de la inspección.

En K-Agile, los artefactos no son documentación. Son **instrumentos operativos** mediante los cuales el equipo gobierna la evolución del conocimiento y su conversión en decisiones.

10.1 Base de Conocimiento.

La **Base de Conocimiento** es el repositorio vivo que contiene el conocimiento explícito que gobierna el sistema. Es el núcleo epistemológico de K-Agile.

La Base de Conocimiento debe contener, como mínimo:

- **Hipótesis y supuestos** explícitos, incluyendo condiciones de validez.
- **Modelos** conceptuales y/o formales (incluyendo modelos de decisión).

- **Reglas, restricciones y heurísticas**, cuando existan.
- **Funciones objetivo y criterios de valor**, incluyendo *trade-offs* documentados.
- **Solvers y mecanismos de decisión**: formulaciones, estrategias, parámetros relevantes y límites.
- **Evidencias**: resultados experimentales, simulaciones, pruebas, validaciones humanas o contextuales.
- **Criterios de validez y límites de aplicabilidad**, incluyendo modos de fallo conocidos.
- **Niveles de madurez** del conocimiento y grado de confianza asociado.

El conocimiento que no se incorpora a la Base de Conocimiento **no se considera existente** a efectos de K-Agile.

La Base de Conocimiento debe permitir responder, de manera inspeccionable, a las siguientes preguntas:

- ¿Qué sabe el sistema y con qué grado de confianza?
- ¿Bajo qué supuestos opera?
- ¿Qué evidencia respalda sus decisiones?
- ¿Qué no sabe aún y qué riesgos implica?

La responsabilidad de mantener su transparencia es colectiva, con énfasis en:

- el Equipo de Investigación y Desarrollo para su actualización factual,
- el Decision Wizard para su coherencia decisonal,
- y el Knowledge Owner para su relevancia estratégica.

10.2 Knowledge Backlog.

El **Knowledge Backlog** es una lista ordenada de **Ítems de Conocimiento** que representan preguntas, hipótesis, incertidumbres u objetivos cognitivos relevantes para el sistema.

El Knowledge Backlog es la **única fuente autorizada** para priorizar trabajo cognitivo. Ningún trabajo que no esté representado en el Knowledge Backlog puede considerarse prioritario dentro de K-Agile.

Cada Ítem de Conocimiento debe incluir, como mínimo:

- una formulación explícita de la incertidumbre o conocimiento buscado;
- la motivación decisonal (por qué importa para decidir mejor);
- supuestos asociados y dependencias;
- criterios preliminares de validación;
- tipo de conocimiento esperado (exploratorio, formal, empírico u operacional);
- riesgos principales si el conocimiento es incorrecto o incompleto.

La ordenación del Knowledge Backlog refleja **prioridad epistemológica y decisional**, no esfuerzo técnico.

El Knowledge Owner es la única persona responsable de su ordenación, pero el refinamiento es responsabilidad conjunta con el Equipo de Investigación y Desarrollo y el Decision Wizard cuando afecte a criterios de decisión.

10.3 Sprint Knowledge Backlog.

El **Sprint Knowledge Backlog** es el subconjunto del Knowledge Backlog seleccionado para un Sprint, junto con un plan explícito para generar, validar y operacionalizar el conocimiento correspondiente.

El Sprint Knowledge Backlog debe incluir, como mínimo:

- los Ítems de Conocimiento seleccionados;
- el desglose del plan en actividades inspeccionables;
- criterios de validación definidos para el Sprint;
- evidencias esperadas (experimentos, simulaciones, pruebas humanas o contextuales);
- riesgos de validez y supuestos críticos activos.

El Sprint Knowledge Backlog representa una **predicción cognitiva**, no un compromiso de alcance fijo. Puede ajustarse durante el Sprint a medida que emerge nuevo conocimiento, siempre que:

- no se comprometa el Objetivo Cognitivo del Sprint;
- no se relajen los criterios de validación;
- y se mantenga coherencia epistemológica.

Solo el Equipo de Investigación y Desarrollo puede modificar el Sprint Knowledge Backlog durante el Sprint.

10.4 Incremento de Conocimiento Operacional.

El **Incremento de Conocimiento Operacional** es el conjunto integrado de conocimiento que, al final del Sprint, cumple las condiciones necesarias para ser utilizado en decisiones reales.

El Incremento de Conocimiento Operacional es la suma coherente de todo el conocimiento operacionalizado hasta la fecha, incluyendo el conocimiento incorporado en:

- modelos y reglas,
- funciones objetivo y *trade-offs*,
- solvers y mecanismos de decisión,
- y criterios de explicabilidad y trazabilidad.

Un Incremento de Conocimiento Operacional debe cumplir obligatoriamente:

1. **Explicitación**

El conocimiento debe estar formalizado de manera explícita en la Base de Conocimiento.

2. **Evidencia**
Debe existir evidencia verificable que respalde el conocimiento incorporado, incluyendo sus límites de aplicabilidad.
3. **Validez (Definición de “Known”)**
El conocimiento operacionalizado debe cumplir la Definición vigente de “Known”.
4. **Coherencia**
El incremento debe ser coherente con el conocimiento operacionalizado previamente. Las contradicciones deben resolverse o documentarse explícitamente.
5. **Trazabilidad y explicabilidad**
Debe ser posible trazar el comportamiento decisional del sistema a conocimiento explícito, evidencias y criterios de decisión.

El incremento puede no implicar nuevas funcionalidades visibles. Un incremento es válido cuando produce **mejora verificable de la capacidad de decisión** o reducción demostrable de incertidumbre crítica con evidencia suficiente.

10.5 Transparencia de artefactos.

El estado real del conocimiento solo puede inspeccionarse si los artefactos reflejan fielmente dicho estado.

Cuando los artefactos no están actualizados o no son transparentes:

- la inspección pierde validez;
- la adaptación se degrada;
- el riesgo cognitivo y operativo se incrementa.

Mantener la transparencia de los artefactos es responsabilidad colectiva del Equipo K-Agile. El K-Agile Facilitator es responsable de promover dicha transparencia y de eliminar impedimentos que la comprometan.

10.6 Evidencias y trazabilidad epistemológica.

En K-Agile, **no existe conocimiento válido sin evidencia**, ni evidencia útil sin trazabilidad explícita hacia las decisiones que gobierna.

Las **evidencias** constituyen un subartefacto normativo implícito que acompaña a todos los demás artefactos y permite verificar la validez del conocimiento generado.

10.6.1 Naturaleza de las evidencias.

Las evidencias pueden adoptar distintas formas, incluyendo, pero no limitándose a:

- resultados experimentales controlados;
- simulaciones y análisis comparativos;
- métricas cuantitativas y análisis estadísticos;
- validaciones con expertos humanos;
- estudios de comportamiento humano y cognitivo;

- pruebas en contextos operativos reales o representativos.

La forma de la evidencia debe ser **adecuada al tipo de conocimiento** que se pretende validar. K-Agile no prescribe un único tipo de evidencia, pero exige que su adecuación sea explícita y justificable.

10.6.2 Requisitos normativos de la evidencia.

Toda evidencia utilizada en K-Agile debe cumplir, como mínimo, los siguientes requisitos:

1. **Explicitación**
La evidencia debe estar documentada y accesible desde la Base de Conocimiento.
2. **Vinculación epistemológica**
Debe existir una relación explícita entre:
 - la evidencia,
 - el Ítem de Conocimiento que valida,
 - y la decisión o capacidad operacional que gobierna.
3. **Contextualización**
La evidencia debe indicar claramente:
 - el contexto en el que fue obtenida,
 - las condiciones bajo las cuales es válida,
 - y los límites de su aplicabilidad.
4. **Inspeccionabilidad**
La evidencia debe poder ser revisada, cuestionada y reinterpretada a la luz de nueva información.

La evidencia que no cumple estos requisitos **no puede utilizarse** para declarar conocimiento como válido o “Known”.

10.6.3 Trazabilidad epistemológica completa.

K-Agile exige **trazabilidad epistemológica completa** desde la decisión hasta el conocimiento y la evidencia que la sustenta.

Debe ser posible trazar, de forma explícita y verificable:

Decisión → solver → función objetivo → conocimiento explícito → evidencia → supuestos

La ausencia de cualquiera de estos eslabones constituye un **riesgo crítico** para la gobernanza del sistema y es incompatible con K-Agile.

El **Decision Wizard** es responsable de asegurar la trazabilidad decisional.

El **Equipo de Investigación y Desarrollo** es responsable de la trazabilidad experimental.

El **Knowledge Owner** es responsable de la coherencia global de dicha trazabilidad.

10.7 Estados de madurez del conocimiento.

En K-Agile, el conocimiento **no es binario** (conocido / no conocido). Evoluciona a través de **estados de madurez explícitos**, que permiten gobernar el riesgo cognitivo y decisional del sistema.

Todo Ítem de Conocimiento y todo elemento de la Base de Conocimiento debe encontrarse **explícitamente** en uno de los siguientes estados.

10.7.1 Estados de madurez.

1. Propuesto

El conocimiento ha sido formulado como pregunta, hipótesis o suposición, pero no ha sido contrastado.

- Evidencia: inexistente o preliminar.
- Uso decisional: no permitido.
- Riesgo: alto.

2. Explorado

Se han realizado análisis iniciales, experimentos preliminares o validaciones limitadas.

- Evidencia: parcial o exploratoria.
- Uso decisional: solo exploratorio o asistido.
- Riesgo: medio-alto.

3. Validado

Existe evidencia suficiente que respalda el conocimiento en un contexto definido.

- Evidencia: explícita, verificable y contextualizada.
- Uso decisional: permitido con restricciones explícitas.
- Riesgo: controlado.

4. Operacionalizado

El conocimiento se utiliza activamente para gobernar decisiones reales mediante modelos, reglas, funciones objetivo o solvers.

- Evidencia: validada y trazable a decisiones.
- Uso decisional: pleno dentro de límites definidos.
- Riesgo: gestionado.

5. Generalizado

El conocimiento ha demostrado validez en múltiples contextos o escenarios relevantes.

- Evidencia: acumulativa y robusta.
- Uso decisional: amplio, con alta confianza.
- Riesgo: bajo, pero nunca nulo.

10.7.2 Transiciones permitidas

Las transiciones entre estados de madurez **no son automáticas** ni implícitas.

- Toda transición debe estar respaldada por evidencia adicional.
- Toda regresión de estado (por nueva evidencia contradictoria) es válida y debe documentarse.
- El conocimiento puede permanecer indefinidamente en un estado si no existe evidencia suficiente para avanzar.

El progreso en K-Agile se mide, en parte, por el **movimiento explícito del conocimiento hacia estados de mayor madurez**, no por la cantidad de ítems creados.

10.7.3 Relación con la Definición de “Known”.

La **Definición de “Known”** establece el umbral mínimo de madurez requerido para que un conocimiento pueda gobernar decisiones reales.

Por defecto:

- solo el conocimiento en estado **Validado** u **Operacionalizado** puede considerarse “Known”;
- cualquier excepción debe documentarse explícitamente como riesgo asumido.

10.7.4 Gobernanza del riesgo cognitivo.

La gestión explícita de los estados de madurez permite:

- evitar decisiones basadas en conocimiento inmaduro;
- entender el riesgo asociado a cada decisión;
- justificar explícitamente por qué una decisión es aceptable;
- y adaptar el sistema cuando la evidencia cambia.

La ocultación o ambigüedad sobre el estado de madurez del conocimiento **invalida la inspección empírica** y es incompatible con K-Agile.

11. Eventos de K-Agile.

K-Agile prescribe un conjunto de **eventos formales y obligatorios** que estructuran el trabajo del Equipo K-Agile y garantizan la aplicación efectiva del **empirismo cognitivo**.

Los eventos existen para crear **regularidad**, habilitar **inspección rigurosa** y permitir **adaptación informada** del conocimiento, de los mecanismos de decisión y del propio marco de trabajo.

En K-Agile, los eventos **no son reuniones**. Son **mecanismos epistemológicos** mediante los cuales el equipo inspecciona el estado del conocimiento, evalúa su validez y adapta su estrategia cognitiva.

La omisión sistemática de cualquiera de los eventos, o su ejecución sin producir los resultados normativos definidos, **invalida la correcta aplicación del marco**.

11.1 Conjunto de eventos.

Los eventos de K-Agile son:

- Planificación del Sprint
- Daily K-Agile
- Revisión del Sprint
- Retrospectiva del Sprint

Estos eventos ocurren dentro del **Sprint**, que actúa como contenedor temporal del ciclo empírico.

11.2 Principios que gobiernan los eventos.

Todos los eventos de K-Agile se rigen por los siguientes principios:

- **Orientación al conocimiento**
Cada evento debe inspeccionar conocimiento explícito, no solo actividad o progreso aparente.
- **Empirismo riguroso**
Las decisiones que emergen de los eventos deben basarse en evidencia y no en suposiciones implícitas.
- **Transparencia cognitiva**
El estado real del conocimiento, su madurez y sus límites deben ser visibles durante los eventos.
- **Adaptación informada**
Los eventos deben conducir a ajustes explícitos en el Knowledge Backlog, los artefactos o la estrategia decisional.

11.3 Time-boxing y disciplina temporal.

Todos los eventos de K-Agile son **time-boxed**.

El límite temporal no existe para acelerar decisiones, sino para:

- reducir riesgo cognitivo,
- forzar inspección frecuente,
- y evitar acumulación de conocimiento no validado.

La duración concreta de cada evento debe ser proporcional a la duración del Sprint y a la complejidad del problema abordado.

11.4 Participación y responsabilidad.

La participación en los eventos de K-Agile está determinada por su propósito cognitivo:

- El **Knowledge Owner** participa para asegurar alineación estratégica del conocimiento.
- El **Equipo de Investigación y Desarrollo** participa como principal generador y validador de conocimiento.
- El **Decision Wizard** participa para evaluar impacto decisional, funciones objetivo y solvers.

- El **K-Agile Facilitator** asegura que el evento cumple su propósito y protege el empirismo.

La ausencia sistemática de alguno de los roles relevantes degrada la calidad del evento.

11.5 Eventos como mecanismos de inspección y adaptación.

Cada evento de K-Agile cumple una función específica dentro del ciclo empírico:

- La **Planificación del Sprint** inspecciona incertidumbre y define objetivos cognitivos.
- El **Daily K-Agile** inspecciona el estado actual del conocimiento y los impedimentos.
- La **Revisión del Sprint** inspecciona el Incremento de Conocimiento Operacional y su impacto decisional.
- La **Retrospectiva del Sprint** inspecciona la calidad del proceso de aprendizaje y generación de conocimiento.

En conjunto, los eventos garantizan que K-Agile opere como un sistema de **aprendizaje controlado**, no como un proceso predictivo.

11.6 Consecuencias del incumplimiento.

Cuando los eventos:

- se convierten en reportes de estado,
- evitan exponer incertidumbre,
- sustituyen evidencia por opinión,
- o no producen adaptaciones explícitas,

el marco deja de operar de forma empírica.

El K-Agile Facilitator es responsable de detectar estas desviaciones y de promover correcciones.

11.7 Relación con el Sprint.

Los eventos existen **para servir al Sprint**, no al revés.

Cada Sprint debe contener exactamente:

- una Planificación del Sprint,
- múltiples Daily K-Agile,
- una Revisión del Sprint,
- y una Retrospectiva del Sprint.

Un Sprint sin alguno de estos eventos **no constituye un Sprint válido en K-Agile**.

12. El Sprint.

El **Sprint** es la unidad central de K-Agile. Es un **bloque de tiempo fijo** durante el cual el Equipo K-Agile ejecuta un **experimento epistemológico completo**, orientado explícitamente a la **reducción de incertidumbre relevante** y a la mejora verificable de la capacidad de decisión del sistema.

En K-Agile, el Sprint no es un contenedor de tareas ni un periodo de entrega funcional. Es un **mecanismo empírico disciplinado** que impone límites temporales al riesgo cognitivo y fuerza la inspección frecuente del conocimiento generado.

Cada Sprint debe producir un **Incremento de Conocimiento Operacional** inspeccionable, respaldado por **evidencias explícitas**, y coherente con la Definición vigente de “Known”.

12.1 Naturaleza epistemológica del Sprint.

Cada Sprint constituye un **experimento epistemológico controlado**.

Esto implica que todo Sprint debe tener, de forma explícita:

- una **incertidumbre principal** que se desea reducir;
- un **Objetivo Cognitivo** claramente formulado;
- **criterios de validación** definidos antes de ejecutar el trabajo;
- y **evidencias esperadas** que permitan evaluar los resultados.

Un Sprint sin hipótesis, sin criterios de validación o sin evidencia prevista **no constituye un Sprint válido en K-Agile**.

12.2 Duración del Sprint.

Los Sprints tienen una duración **fija y consistente** a lo largo del proyecto.

La duración máxima recomendada de un Sprint es de **un mes calendario**. Duraciones más cortas pueden utilizarse cuando el nivel de incertidumbre o el riesgo cognitivo lo requieran.

La duración del Sprint establece el **horizonte máximo de riesgo cognitivo aceptable**. Sprints excesivamente largos incrementan la probabilidad de generar conocimiento obsoleto, no validable o incorrecto.

Una vez iniciado un Sprint:

- su duración no puede modificarse;
- el Objetivo Cognitivo no puede cambiarse;
- y los criterios de validación no pueden relajarse.

12.3 Objetivo Cognitivo del Sprint.

Cada Sprint debe tener **un único Objetivo Cognitivo**.

El Objetivo Cognitivo expresa **qué debe saberse mejor** al finalizar el Sprint y **qué incertidumbre relevante se pretende reducir**. Debe formularse de manera clara, verificable y comprensible para todo el Equipo K-Agile.

El Objetivo Cognitivo proporciona coherencia interna al Sprint y actúa como referencia constante para:

- la planificación,
- la inspección diaria,

- la evaluación del Incremento,
- y la adaptación posterior.

El Objetivo Cognitivo **no se expresa como una lista de tareas ni como una promesa funcional**, sino como un avance de conocimiento.

12.4 Contenido obligatorio del Sprint.

Todo Sprint incluye necesariamente:

- una **Planificación del Sprint**;
- trabajo de investigación, modelado, implementación y validación;
- uno o varios **Daily K-Agile**;
- una **Revisión del Sprint**;
- y una **Retrospectiva del Sprint**.

La omisión de cualquiera de estos eventos reduce la capacidad de inspección y adaptación, y degrada el carácter empírico del Sprint.

12.5 Cambios durante el Sprint.

Durante el Sprint, el contenido del **Sprint Knowledge Backlog** puede ajustarse a medida que emerge nuevo conocimiento.

Estos ajustes están permitidos únicamente si:

- no comprometen el Objetivo Cognitivo del Sprint;
- no degradan los criterios de validación establecidos;
- y mantienen coherencia epistemológica.

El Sprint no es un contrato de alcance fijo, pero sí un **compromiso cognitivo**.

12.6 Cancelación del Sprint.

Un Sprint puede ser cancelado antes de finalizar su bloque de tiempo.

Solo el **Knowledge Owner** tiene la autoridad para cancelar un Sprint, aunque puede hacerlo bajo la influencia del Equipo de Investigación y Desarrollo, del Decision Wizard o del K-Agile Facilitator.

La cancelación es apropiada cuando el Objetivo Cognitivo pierde validez, por ejemplo:

- cuando una hipótesis central resulta incorrecta de forma temprana;
- cuando el contexto operativo cambia de manera sustancial;
- cuando nueva evidencia invalida el experimento en curso.

Cuando se cancela un Sprint:

- el conocimiento generado hasta el momento debe inspeccionarse;

- el conocimiento validado se incorpora a la Base de Conocimiento;
- los Ítems de Conocimiento no completados regresan al Knowledge Backlog.

La cancelación de un Sprint **no constituye un fallo**, sino una aplicación correcta del empirismo.

12.7 Sprint fallido.

Un Sprint se considera **fallido** cuando no produce un Incremento de Conocimiento Operacional que cumpla la Definición de “Known”.

Un Sprint fallido indica, típicamente:

- formulación incorrecta del Objetivo Cognitivo;
- criterios de validación inadecuados;
- supuestos implícitos no explicitados;
- o evidencia insuficiente.

Un Sprint fallido **no implica un fallo del equipo**, pero sí requiere inspección explícita y adaptación en la Retrospectiva.

12.8 Resultado normativo del Sprint.

Al final de cada Sprint debe existir, de forma explícita:

- un **Incremento de Conocimiento Operacional** inspeccionable;
- un conjunto de **evidencias** que respalden el conocimiento incorporado;
- una actualización coherente de la **Base de Conocimiento**;
- y una adaptación informada del **Knowledge Backlog**.

La ausencia de cualquiera de estos elementos invalida el Sprint desde el punto de vista de K-Agile.

13. Planificación del Sprint.

La **Planificación del Sprint** es el evento mediante el cual el Equipo K-Agile establece el **marco cognitivo del experimento** que se ejecutará durante el Sprint.

Durante este evento se definen explícitamente:

- las **incertidumbres objetivo** que se desean reducir;
- el **conocimiento esperado** al finalizar el Sprint;
- y los **criterios de validación** que permitirán evaluar el resultado.

La Planificación del Sprint existe para asegurar que el trabajo del Sprint esté **orientado al aprendizaje relevante**, y no a la ejecución de actividad no dirigida.

13.1 Propósito de la Planificación del Sprint.

El propósito de la Planificación del Sprint es responder, de forma explícita y verificable, a las siguientes preguntas fundamentales:

1. ¿Qué incertidumbre relevante se va a abordar en este Sprint?
2. ¿Qué conocimiento debe existir al finalizar el Sprint si el experimento tiene éxito?
3. ¿Qué evidencia será suficiente para considerar dicho conocimiento válido?
4. ¿Cómo se transformará ese conocimiento en capacidad de decisión?

Si cualquiera de estas preguntas no puede responderse de forma clara, el Sprint **no debe iniciarse**.

13.2 Participantes y responsabilidades.

La Planificación del Sprint requiere la participación activa de todos los roles del Equipo K-Agile:

- El **Knowledge Owner** propone y prioriza las incertidumbres a abordar.
- El **Equipo de Investigación y Desarrollo** evalúa la viabilidad epistemológica y técnica del experimento.
- El **Decision Wizard** analiza el impacto decisional esperado y la coherencia con funciones objetivo y solvers.
- El **K-Agile Facilitator** asegura que el evento cumple su propósito y protege el empirismo cognitivo.

La planificación no es una negociación de alcance, sino una **construcción colectiva del experimento**.

13.3 Selección de Ítems de Conocimiento.

Durante la Planificación del Sprint, el Equipo K-Agile selecciona un conjunto de **Ítems de Conocimiento** del Knowledge Backlog para conformar el **Sprint Knowledge Backlog**.

La selección se basa en:

- relevancia de la incertidumbre;
- impacto esperado en la capacidad de decisión;
- dependencias epistemológicas;
- y riesgo cognitivo asumido.

Los Ítems de Conocimiento seleccionados deben contribuir de manera directa al **Objetivo Cognitivo del Sprint**.

13.4 Definición del Objetivo Cognitivo del Sprint.

El **Objetivo Cognitivo del Sprint** se define durante la Planificación del Sprint.

Debe formularse como una **afirmación de aprendizaje esperado**, por ejemplo:

- comprensión de un fenómeno;
- validación o refutación de una hipótesis;
- delimitación de condiciones de validez;

- o refinamiento de criterios decisionales.

El Objetivo Cognitivo debe ser:

- único;
- explícito;
- verificable;
- y comprensible para todo el Equipo K-Agile.

13.5 Criterios de validación.

Durante la Planificación del Sprint se definen los **criterios de validación** que permitirán evaluar si el Objetivo Cognitivo ha sido alcanzado.

Estos criterios deben especificar:

- qué evidencias se consideran aceptables;
- en qué condiciones dichas evidencias son válidas;
- qué umbral de suficiencia debe alcanzarse;
- y qué resultados implican rechazo o reformulación de la hipótesis.

Los criterios de validación **no pueden relajarse** durante el Sprint.

13.6 Diseño del experimento epistemológico.

La Planificación del Sprint incluye el diseño explícito del **experimento epistemológico** que se ejecutará.

Este diseño debe contemplar, como mínimo:

- métodos de investigación o modelado;
- experimentos técnicos o humanos;
- simulaciones o pruebas;
- análisis de sensibilidad o robustez;
- y mecanismos de captura de evidencia.

El diseño debe ser proporcional al riesgo cognitivo y a la criticidad de la decisión.

13.7 Plan de trabajo del Sprint.

El **Sprint Knowledge Backlog** resultante incluye un plan de trabajo que describe **cómo** el equipo pretende alcanzar el Objetivo Cognitivo.

Este plan no constituye un compromiso de alcance fijo, sino una **hipótesis operativa** sujeta a inspección y adaptación.

El Equipo de Investigación y Desarrollo es responsable de su elaboración y ajuste durante el Sprint.

13.8 Compromiso cognitivo.

Al finalizar la Planificación del Sprint, el Equipo K-Agile asume un **compromiso cognitivo**:

- compromete su esfuerzo a reducir la incertidumbre definida;
- compromete la aplicación de criterios de validación rigurosos;
- y compromete la transparencia del conocimiento generado, incluso si este contradice expectativas previas.

Este compromiso sustituye al compromiso tradicional de entrega.

13.9 Resultado normativo de la Planificación del Sprint.

Al finalizar la Planificación del Sprint deben existir, de forma explícita:

- un **Objetivo Cognitivo del Sprint** claramente formulado;
- un **Sprint Knowledge Backlog** coherente;
- criterios de validación explícitos;
- y un diseño claro del experimento epistemológico.

La ausencia de cualquiera de estos elementos invalida la Planificación del Sprint desde el punto de vista de K-Agile.

14. Daily K-Agile.

El **Daily K-Agile** es un evento **diario, formal y obligatorio** cuyo propósito es **inspeccionar el progreso cognitivo del Sprint** y **adaptar el plan de trabajo inmediato** para maximizar la probabilidad de alcanzar el Objetivo Cognitivo del Sprint.

En K-Agile, el Daily K-Agile **no existe para coordinar tareas**, sino para **evaluar el estado del conocimiento en generación**, detectar desviaciones epistemológicas y ajustar el experimento antes de que el riesgo cognitivo se materialice.

14.1 Propósito epistemológico del Daily K-Agile.

El propósito principal del Daily K-Agile es responder, de forma explícita y recurrente, a las siguientes preguntas:

1. ¿Qué conocimiento nuevo se ha generado desde el último Daily?
2. ¿Qué evidencia se ha obtenido y qué indica realmente?
3. ¿Qué incertidumbres permanecen activas o han cambiado de naturaleza?
4. ¿Sigue siendo válido el diseño del experimento epistemológico?
5. ¿Qué debe adaptarse hoy para maximizar aprendizaje y validez?

Estas preguntas desplazan el foco desde la ejecución de tareas hacia la **evolución real del conocimiento**.

14.2 Naturaleza del evento.

El Daily K-Agile es un evento **time-boxed** y de **frecuencia diaria**.

La frecuencia diaria existe para:

- reducir el tiempo durante el cual el equipo puede avanzar con supuestos incorrectos;
- detectar tempranamente evidencia contradictoria;
- y limitar la acumulación de conocimiento implícito no inspeccionado.

El Daily K-Agile no es una reunión de reporte de estado ni un mecanismo de control jerárquico. Es un **instrumento de inspección colectiva del experimento en curso**.

14.3 Participantes y responsabilidades.

El Daily K-Agile es un evento del **Equipo de Investigación y Desarrollo**, pero puede contar con la participación activa de otros roles cuando sea necesario.

- El **Equipo de Investigación y Desarrollo** inspecciona el conocimiento generado y decide adaptaciones inmediatas.
- El **Decision Wizard** participa cuando los hallazgos afectan funciones objetivo, solvers o criterios decisionales.
- El **Knowledge Owner** puede participar para observar la evolución cognitiva y detectar desviaciones estratégicas.
- El **K-Agile Facilitator** asegura que el evento se centre en conocimiento y no derive en coordinación de tareas.

La participación debe estar siempre alineada con el propósito cognitivo del evento.

14.4 Inspección del progreso cognitivo.

Durante el Daily K-Agile, el progreso se evalúa en términos de **conocimiento**, no de actividad.

Esto implica inspeccionar explícitamente:

- hipótesis confirmadas, refutadas o debilitadas;
- evidencias obtenidas y su calidad;
- supuestos que han dejado de ser válidos;
- riesgos epistemológicos emergentes;
- y cambios en el estado de madurez del conocimiento.

El uso de métricas de actividad o esfuerzo como sustituto del progreso cognitivo **no es aceptable en K-Agile**.

14.5 Adaptación del plan de trabajo inmediato.

A partir de la inspección, el equipo adapta su plan de trabajo diario.

Estas adaptaciones pueden incluir:

- rediseñar un experimento;

- cambiar el orden de actividades;
- profundizar en una hipótesis emergente;
- descartar líneas de trabajo inconsistentes;
- o introducir nuevas pruebas o validaciones.

La adaptación debe ser **proporcional a la evidencia observada** y siempre orientada a proteger el Objetivo Cognitivo del Sprint.

14.6 Relación con el Objetivo Cognitivo del Sprint.

El Daily K-Agile existe para **proteger el Objetivo Cognitivo del Sprint**.

Cuando el Daily revela que:

- el experimento no está produciendo evidencia relevante,
- los supuestos centrales son incorrectos,
- o el riesgo cognitivo es mayor de lo esperado,

el equipo debe adaptar inmediatamente su enfoque o escalar la situación a la Planificación o incluso considerar la cancelación del Sprint.

Persistir sin adaptación constituye una violación del empirismo.

14.7 Transparencia y explicitación.

El Daily K-Agile exige **transparencia cognitiva total**.

Esto implica que:

- la incertidumbre debe hacerse visible;
- los resultados negativos deben compartirse;
- las dudas deben expresarse explícitamente;
- y los errores tempranos deben valorarse como información valiosa.

Ocultar incertidumbre para aparentar progreso **invalida el propósito del evento**.

14.8 Resultado normativo del Daily K-Agile.

Al finalizar cada Daily K-Agile debe existir, de forma explícita:

- una comprensión compartida del estado actual del conocimiento;
- una lista clara de adaptaciones inmediatas;
- una identificación de riesgos cognitivos activos;
- y, cuando sea necesario, una actualización del Sprint Knowledge Backlog.

Si el Daily no produce adaptación informada, ha fallado en su propósito.

15. Revisión del Sprint.

La **Revisión del Sprint** es el evento mediante el cual el Equipo K-Agile **inspecciona el Incremento de Conocimiento Operacional**, evalúa su **validez epistemológica y decisonal**, y **adapta el Knowledge Backlog** en función del aprendizaje generado.

En K-Agile, la Revisión del Sprint **no es una demostración de resultados técnicos** ni una validación superficial de avances aparentes. Es un **acto formal de evaluación del conocimiento** y de su capacidad para gobernar decisiones reales.

15.1 Propósito de la Revisión del Sprint.

El propósito de la Revisión del Sprint es responder, de manera explícita y basada en evidencia, a las siguientes preguntas fundamentales:

1. ¿Qué conocimiento ha sido efectivamente generado durante el Sprint?
2. ¿Qué conocimiento ha alcanzado el nivel requerido de validez y madurez?
3. ¿Qué evidencia respalda dicho conocimiento y cuáles son sus límites?
4. ¿Cómo afecta este conocimiento a la capacidad de decisión del sistema?
5. ¿Qué incertidumbres han sido reducidas y cuáles permanecen abiertas?

Estas preguntas desplazan el foco desde “qué se ha hecho” hacia **qué se sabe ahora mejor que antes**.

15.2 Objeto de inspección.

El objeto central de la Revisión del Sprint es el **Incremento de Conocimiento Operacional**.

Durante la Revisión se inspecciona explícitamente:

- el conocimiento incorporado en la Base de Conocimiento;
- su estado de madurez y conformidad con la Definición de “Known”;
- las evidencias que lo respaldan;
- los supuestos que permanecen activos;
- y su impacto sobre modelos, funciones objetivo, solvers y decisiones.

La ausencia de evidencia explícita invalida cualquier afirmación de conocimiento operativo.

15.3 Participantes y responsabilidades

La Revisión del Sprint es un evento del **Equipo K-Agile**, con participación de los roles clave y, cuando sea relevante, de partes interesadas externas al equipo.

- El **Equipo de Investigación y Desarrollo** presenta el conocimiento generado y la evidencia asociada.
- El **Decision Wizard** evalúa el impacto del incremento sobre la calidad, coherencia y explicabilidad de las decisiones.

- El **Knowledge Owner** inspecciona el valor cognitivo y decisional del incremento.
- El **K-Agile Facilitator** asegura que el evento se centre en conocimiento y evidencia, y no derive en justificación de actividad.

Las partes interesadas pueden observar y aportar contexto, pero no sustituyen el juicio empírico del equipo.

15.4 Evaluación de la validez del conocimiento.

Durante la Revisión del Sprint, el conocimiento generado se evalúa explícitamente en función de:

- su grado de validación empírica;
- la adecuación de la evidencia presentada;
- la claridad de sus límites de aplicabilidad;
- y su coherencia con el conocimiento previamente operacionalizado.

El conocimiento puede ser:

- aceptado como válido y operacional;
- aceptado con restricciones explícitas;
- reclasificado a un estado de madurez inferior;
- o descartado si la evidencia lo contradice.

Descartar conocimiento incorrecto constituye un **resultado positivo del Sprint**, no un fallo.

15.5 Impacto decisional.

Un criterio central de la Revisión del Sprint es el **impacto decisional** del Incremento de Conocimiento Operacional.

Esto implica evaluar:

- si las decisiones del sistema son ahora más robustas;
- si los *trade-offs* están mejor comprendidos;
- si las funciones objetivo reflejan mejor el valor real;
- y si los solvers operan bajo supuestos más precisos.

Un incremento que no mejora la capacidad de decisión ni reduce incertidumbre relevante **no cumple el propósito del Sprint**, incluso si produjo actividad técnica significativa.

15.6 Adaptación del Knowledge Backlog.

Como resultado de la Revisión del Sprint, el **Knowledge Backlog** debe adaptarse explícitamente.

Esta adaptación puede incluir:

- incorporación de nuevas incertidumbres identificadas;

- reformulación de Ítems de Conocimiento existentes;
- cambio de prioridad en función del aprendizaje;
- o eliminación de ítems que han dejado de ser relevantes.

El Knowledge Owner es responsable de estas adaptaciones, basándose en el aprendizaje generado durante la Revisión.

15.7 Transparencia y explicitación del aprendizaje.

La Revisión del Sprint exige **transparencia total** respecto a lo aprendido.

Esto incluye hacer explícitos:

- los resultados esperados y no esperados;
- los errores de hipótesis;
- los supuestos que resultaron incorrectos;
- y las limitaciones del conocimiento obtenido.

La ocultación de resultados negativos o ambiguos compromete la validez del marco empírico.

15.8 Resultado normativo de la Revisión del Sprint.

Al finalizar la Revisión del Sprint deben existir, de forma explícita:

- un Incremento de Conocimiento Operacional inspeccionado;
- una evaluación clara de su validez y límites;
- una actualización coherente de la Base de Conocimiento;
- y una adaptación informada del Knowledge Backlog.

Si la Revisión no produce adaptación explícita, el evento ha fallado en su propósito.

16. Retrospectiva del Sprint.

La **Retrospectiva del Sprint** es el evento mediante el cual el Equipo K-Agile **inspecciona y mejora** la calidad de su proceso de **generación, validación y operacionalización del conocimiento**.

En K-Agile, la Retrospectiva **no se centra en comportamientos individuales ni en eficiencia operativa**, sino en la **eficacia epistemológica del sistema de trabajo**. Su propósito es mejorar **cómo el equipo aprende**, no solo cómo colabora.

16.1 Propósito de la Retrospectiva del Sprint.

El propósito de la Retrospectiva del Sprint es identificar y aplicar mejoras que incrementen:

- la calidad del conocimiento generado;
- la solidez de la validación empírica;
- la claridad de la explicitación de supuestos;

- la trazabilidad entre conocimiento y decisiones;
- y la eficacia del uso del marco K-Agile.

La Retrospectiva existe para evitar que errores epistemológicos se repitan y para reforzar prácticas que producen aprendizaje fiable.

16.2 Objeto de inspección

El objeto de la Retrospectiva del Sprint es el **proceso cognitivo** seguido durante el Sprint.

Esto incluye inspeccionar explícitamente:

- la formulación del Objetivo Cognitivo;
- la calidad del diseño del experimento epistemológico;
- la adecuación de los criterios de validación;
- la gestión de incertidumbre y supuestos;
- el uso efectivo de evidencias;
- la integración de conocimiento humano y algorítmico;
- y la coherencia entre conocimiento generado y decisiones producidas.

La Retrospectiva **no inspecciona personas**, inspecciona **el sistema de trabajo**.

16.3 Participantes y responsabilidades.

La Retrospectiva del Sprint es un evento del **Equipo K-Agile completo**.

- El **Equipo de Investigación y Desarrollo** aporta visión sobre la ejecución del experimento.
- El **Knowledge Owner** evalúa la adecuación del enfoque cognitivo y estratégico.
- El **Decision Wizard** inspecciona la calidad de la conversión de conocimiento en decisiones.
- El **K-Agile Facilitator** facilita el evento y protege su foco epistemológico.

La participación debe ser abierta, crítica y orientada a mejora real.

16.4 Inspección de la calidad del aprendizaje.

Durante la Retrospectiva, el equipo debe analizar explícitamente preguntas como:

- ¿Formulamos correctamente la incertidumbre inicial?
- ¿Definimos criterios de validación adecuados?
- ¿Detectamos pronto evidencias contradictorias?
- ¿Explicitamos suficientemente los supuestos?
- ¿Confundimos actividad con aprendizaje?
- ¿Descartamos conocimiento incorrecto con la suficiente rapidez?

Estas preguntas permiten identificar **fallos sistemáticos de aprendizaje**, no errores puntuales.

16.5 Inspección de la disciplina metodológica.

La Retrospectiva también inspecciona el uso del marco K-Agile en sí mismo.

Esto incluye evaluar:

- si los eventos cumplieron su propósito normativo;
- si los artefactos fueron transparentes y actualizados;
- si los estados de madurez del conocimiento se aplicaron correctamente;
- si la Definición de “Known” fue respetada;
- y si existieron desviaciones metodológicas.

La mejora del marco en uso es tan importante como la mejora del conocimiento generado.

16.6 Identificación de mejoras concretas.

La Retrospectiva del Sprint debe producir **mejoras concretas y accionables**.

Estas mejoras pueden afectar a:

- la formulación de Ítems de Conocimiento;
- el diseño de experimentos;
- los criterios de validación;
- la integración de conocimiento humano;
- el uso de solvers y funciones objetivo;
- o la facilitación de los eventos.

Al menos **una mejora identificada** debe incorporarse explícitamente al siguiente Sprint.

16.7 Relación con el riesgo cognitivo.

La Retrospectiva permite gestionar el **riesgo cognitivo acumulado**.

El equipo debe identificar:

- dónde se asumió riesgo innecesario;
- dónde se fue excesivamente conservador;
- y cómo equilibrar exploración y validación en futuros Sprints.

La gestión explícita del riesgo cognitivo es una condición necesaria para la evolución controlada del sistema.

16.8 Resultado normativo de la Retrospectiva del Sprint.

Al finalizar la Retrospectiva del Sprint deben existir, de forma explícita:

- una evaluación compartida de la calidad del aprendizaje del Sprint;
- una lista priorizada de mejoras del proceso cognitivo;
- al menos una mejora comprometida para el siguiente Sprint;
- y, cuando corresponda, ajustes al uso del marco K-Agile.

Si la Retrospectiva no produce mejoras aplicables, el evento ha fallado en su propósito.

17. Definición de “Known” y “Wisdom”.

En K-Agile, no todo conocimiento generado puede utilizarse para gobernar decisiones reales. La **Definición de “Known”** establece el umbral mínimo de validez epistemológica requerido para que un conocimiento pueda considerarse fiable. La **Definición de “Wisdom”** establece el umbral adicional requerido para que dicho conocimiento pueda gobernar decisiones óptimas mediante solvers.

Ambas definiciones son **normativas, explícitas y obligatorias**. El incumplimiento de cualquiera de ellas invalida el uso operativo del conocimiento o de la decisión resultante.

17.1 Definición de “Known”.

Un conocimiento se considera **“Known”** cuando ha alcanzado un nivel de madurez suficiente para ser **utilizado como base fiable** en la toma de decisiones.

Un conocimiento solo puede declararse “Known” cuando cumple **simultáneamente** todas las condiciones siguientes:

1. **Explicitación formal**

El conocimiento está formulado de manera explícita y accesible en la Base de Conocimiento, incluyendo:

- hipótesis o afirmaciones claras;
- supuestos activos;
- contexto de validez;
- y límites conocidos.

2. **Evidencia suficiente**

Existe evidencia empírica verificable que respalda el conocimiento, obtenida mediante:

- experimentación,
- simulación,
- validación contextual,
- o análisis humano riguroso, según corresponda.

3. **Contextualización**

El conocimiento especifica claramente:

- en qué condiciones es válido,



- para qué tipo de decisiones es aplicable,
 - y qué ocurre fuera de dichos límites.
4. **Estado de madurez adecuado**
El conocimiento se encuentra, como mínimo, en estado **Validado** según los estados de madurez definidos en K-Agile.
5. **Trazabilidad epistemológica**
Es posible trazar el conocimiento hasta:
- las evidencias que lo validan,
 - los supuestos que lo condicionan,
 - y las decisiones que puede gobernar.

Un conocimiento que no cumple cualquiera de estas condiciones **no puede considerarse “Known”**, independientemente de su plausibilidad o aceptación informal.

17.2 Alcance y límites del conocimiento “Known”.

El conocimiento “Known” **no implica certeza absoluta**.

En K-Agile, “Known” significa:

- suficientemente validado,
- explícitamente limitado,
- y gobernable en términos de riesgo.

Todo conocimiento “Known” conserva:

- incertidumbre residual,
- dependencia contextual,
- y posibilidad de revisión futura.

Declarar conocimiento como “Known” **no lo congela**, pero sí lo habilita para uso responsable.

17.3 Definición de “Wisdom”.

En K-Agile, la **Wisdom (sabiduría operacional)** es una propiedad emergente del sistema que aparece **cuando el conocimiento “Known” se transforma correctamente en decisiones óptimas**.

Un sistema posee “Wisdom” cuando:

El conocimiento validado se ha incorporado de forma coherente en **funciones objetivo, criterios de decisión y solvers**, produciendo decisiones que maximizan el valor definido bajo restricciones explícitas y contextualmente válidas.

La Wisdom **no es conocimiento adicional**. Es conocimiento **correctamente aplicado**, en base a experiencia y a un objetivo determinado.

17.4 Condiciones normativas de “Wisdom”.

Una capacidad de decisión solo puede considerarse **Wise** cuando se cumplen simultáneamente las siguientes condiciones:

1. **Conocimiento subyacente “Known”**
Todas las variables, reglas, restricciones y criterios utilizados por el solver están basados exclusivamente en conocimiento declarado como “Known”.
2. **Función objetivo explícita y justificada**
Existe una función objetivo claramente definida que:
 - formaliza el concepto de valor;
 - incorpora *trade-offs* reales;
 - y refleja prioridades humanas, técnicas y contextuales.
3. **Solver adecuado al tipo de decisión**
El solver utilizado es coherente con:
 - la naturaleza del problema,
 - el nivel de incertidumbre,
 - y los requisitos de explicabilidad y control.
4. **Trazabilidad decisional completa**
Es posible trazar cualquier decisión hasta:
 - la función objetivo aplicada,
 - el conocimiento utilizado,
 - las evidencias que lo validan,
 - y los supuestos activos en ese momento.
5. **Validación del resultado decisional**
Las decisiones producidas han sido evaluadas empíricamente, demostrando:
 - mejora frente a decisiones previas,
 - robustez frente a variaciones,
 - y comportamiento aceptable en contexto real o simulado.

Una decisión que utiliza conocimiento “Known” pero no cumple estas condiciones **no constituye Wisdom**.

17.5 Relación entre “Known” y “Wisdom”.

La relación entre ambos conceptos es jerárquica y no intercambiable:

- **Known** es condición necesaria pero no suficiente para **Wisdom**.
- **Wisdom** no puede existir sin **Known**.
- Puede existir conocimiento “Known” que aún no se haya convertido en **Wisdom**.

- Nunca puede existir Wisdom basada en conocimiento no “Known”.

Esta distinción permite a K-Agile separar claramente:

- validez epistemológica (saber),
- de calidad decisional (decidir bien).

17.6 Responsabilidades sobre “Known” y “Wisdom”.

- El **Equipo de Investigación y Desarrollo** es responsable de generar y validar conocimiento hasta alcanzar el estado “Known”.
- El **Decision Wizard** es responsable de transformar conocimiento “Known” en Wisdom mediante funciones objetivo y solvers adecuados.
- El **Knowledge Owner** es responsable de decidir cuándo un conocimiento es suficientemente relevante para ser declarado “Known”.
- El **K-Agile Facilitator** es responsable de asegurar que estas definiciones se aplican rigurosamente.

17.7 Consecuencias del incumplimiento.

Utilizar conocimiento no “Known” para decisiones reales, o declarar Wisdom sin cumplir sus condiciones normativas, constituye un **fallo crítico del marco K-Agile**.

Dicho fallo incrementa:

- el riesgo cognitivo,
- el riesgo decisional,
- y el riesgo operativo del sistema.

En K-Agile, **no decidir es preferible a decidir mal basándose en conocimiento no validado**.

18. Seguimiento del progreso.

En K-Agile, el progreso **no se mide por la cantidad de trabajo realizado**, la velocidad de ejecución ni la entrega incremental de funcionalidades. El progreso se evalúa exclusivamente mediante:

- la **reducción demostrable de incertidumbre relevante**;
 - la **madurez del conocimiento generado**;
 - y la **mejora verificable de la capacidad de decisión del sistema**;
- todo ello **respaldado por evidencias explícitas, trazables y contextualizadas**.

Esta definición de progreso es normativa. Cualquier métrica o criterio que no refleje alguno de estos tres ejes **no constituye progreso en K-Agile**.

18.1 Principio rector del seguimiento del progreso

El principio rector del seguimiento del progreso en K-Agile es el siguiente:

Un sistema progresa cuando **sabe más, sabe mejor o decide mejor** que antes, y puede demostrarlo con evidencia.

El progreso es, por tanto, **cognitivo y decisional**, no operativo.

18.2 Reducción de incertidumbre.

La **reducción de incertidumbre relevante** es el primer indicador fundamental de progreso.

En K-Agile, la incertidumbre se considera reducida cuando:

- una hipótesis es validada o refutada;
- un supuesto implícito se hace explícito y se evalúa;
- se delimitan claramente las condiciones de validez de un conocimiento;
- o se identifican con precisión los límites de aplicabilidad de una decisión.

Reducir incertidumbre **no implica confirmar hipótesis**. Refutar una hipótesis central constituye un avance significativo del progreso si está respaldado por evidencia suficiente.

El seguimiento del progreso exige que:

- las incertidumbres activas estén explícitamente identificadas;
- se pueda observar su evolución entre Sprints;
- y se documente qué incertidumbres han sido cerradas, reformuladas o reemplazadas.

18.3 Madurez del conocimiento.

El segundo eje del progreso es la **evolución del conocimiento a través de los estados de madurez definidos en K-Agile**.

El progreso se observa cuando:

- el conocimiento transita de *Propuesto* a *Explorado*;
- de *Explorado* a *Validado*;
- y de *Validado* a *Operacionalizado* o *Generalizado*.

El volumen de conocimiento creado **no es un indicador de progreso** si dicho conocimiento permanece inmaduro o no validado.

El seguimiento del progreso debe permitir responder, de forma explícita:

- qué conocimiento ha aumentado su nivel de madurez;
- qué conocimiento ha retrocedido por nueva evidencia;
- y qué conocimiento permanece estancado y por qué.

La regresión de madurez, cuando está justificada por evidencia, **no es un fracaso**, sino una manifestación correcta del empirismo.

18.4 Mejora de la capacidad de decisión.

El tercer eje —y el más crítico en K-Agile— es la **mejora verificable de la capacidad de decisión**.

Una mejora de la capacidad de decisión puede manifestarse, entre otros, en:

- decisiones más robustas frente a incertidumbre;
- mejor gestión de *trade-offs*;
- mayor alineación entre decisiones y valor real;
- incremento de la explicabilidad y comprensión de las decisiones;
- reducción del riesgo operativo o cognitivo;
- o mayor adecuación a contextos humanos y organizativos.

La mejora decisional debe estar **empíricamente respaldada**, mediante:

- comparación con decisiones previas;
- análisis de sensibilidad;
- validación en contexto real o simulado;
- o evaluación por expertos humanos, cuando corresponda.

Una decisión “más rápida” o “más automática” **no implica progreso** si no es también una decisión mejor.

18.5 Evidencias como base del seguimiento.

Todo seguimiento del progreso en K-Agile debe estar respaldado por **evidencias explícitas**.

No se aceptan como indicadores de progreso:

- percepciones subjetivas;
- métricas de esfuerzo o actividad;
- cumplimiento de planes sin validación;
- ni resultados no reproducibles o no contextualizados.

Las evidencias utilizadas para evaluar progreso deben:

- estar documentadas en la Base de Conocimiento;
- ser trazables a Ítems de Conocimiento;
- y poder relacionarse con decisiones concretas.

Sin evidencia, no hay progreso verificable.

18.6 Instrumentos de seguimiento del progreso.

El seguimiento del progreso en K-Agile se apoya principalmente en:

- la evolución del **Knowledge Backlog**;
- los cambios de estado de madurez del conocimiento;
- los Incrementos de Conocimiento Operacional;
- y los resultados de las Revisiones del Sprint.

Estos instrumentos permiten inspeccionar progreso **Sprint a Sprint**, sin necesidad de métricas externas artificiales.

18.7 Rechazo explícito de métricas tradicionales

K-Agile **rechaza explícitamente** el uso de métricas tradicionales de progreso como indicadores primarios, incluyendo:

- velocidad (*velocity*);
- número de tareas completadas;
- cantidad de código producido;
- o porcentaje de cumplimiento de planes.

Estas métricas pueden utilizarse como información secundaria, pero **no definen progreso en K-Agile**.

El uso de métricas operativas como sustituto del progreso cognitivo constituye un **anti-patrón metodológico**.

18.8 Responsabilidades en el seguimiento del progreso.

- El **Equipo de Investigación y Desarrollo** es responsable de producir evidencias del progreso cognitivo.
- El **Decision Wizard** es responsable de evidenciar la mejora decisional.
- El **Knowledge Owner** es responsable de evaluar el progreso en términos de valor cognitivo y estratégico.
- El **K-Agile Facilitator** es responsable de asegurar que el progreso se evalúa conforme a los principios del marco.

18.9 Resultado normativo del seguimiento del progreso.

En todo momento debe ser posible responder, de forma explícita y verificable, a las siguientes preguntas:

- ¿Qué incertidumbres se han reducido recientemente?
- ¿Qué conocimiento ha madurado y hasta qué nivel?
- ¿En qué decisiones decide mejor el sistema ahora que antes?
- ¿Qué evidencias respaldan estas afirmaciones?

Si estas preguntas no pueden responderse, **no existe progreso demostrable en K-Agile**.

19. Conformidad con K-Agile.

La **conformidad con K-Agile** define las condiciones bajo las cuales una implementación, proyecto u organización puede afirmar legítimamente que opera conforme a este marco.

K-Agile es un **marco normativo e integral**. Su correcta aplicación requiere la adopción **coherente, explícita y completa** de todos los elementos definidos en esta guía. La conformidad no es gradual ni interpretativa: **se cumple o no se cumple**.

19.1 Criterio general de conformidad

Una implementación es conforme con K-Agile únicamente cuando:

- adopta **todos los principios fundamentales** del marco;

- implementa **todos los roles definidos**, con responsabilidades efectivamente ejercidas;
- utiliza **todos los artefactos obligatorios**, manteniéndolos actualizados y transparentes;
- ejecuta **todos los eventos prescritos**, cumpliendo su propósito epistemológico;
- aplica de forma rigurosa la **Definición de “Known”** y la **Definición de “Wisdom”**;
- y evalúa el progreso exclusivamente conforme a los criterios definidos en K-Agile.

La ausencia, debilitamiento o sustitución informal de cualquiera de estos elementos **rompe la conformidad del marco**.

19.2 Conformidad de principios.

Una implementación conforme con K-Agile debe demostrar, de manera observable, que:

- el **conocimiento explícito** es el eje organizador del trabajo;
- el **empirismo riguroso** gobierna las decisiones;
- existe **trazabilidad epistemológica** entre conocimiento, evidencia y decisión;
- se respeta la **co-evolución** entre conocimiento, solvers y contexto;
- la **validación contextual** es obligatoria;
- los supuestos se **explicitan sistemáticamente**;
- y se mantiene una **disciplina metodológica estricta**.

La violación sistemática de cualquiera de estos principios invalida la conformidad, incluso si los roles o eventos existen nominalmente.

19.3 Conformidad de roles.

Para ser conforme con K-Agile, deben existir y ejercerse de manera efectiva los siguientes roles:

- **Knowledge Owner**, como autoridad cognitiva y estratégica;
- **Equipo de Investigación y Desarrollo**, como generador y validador de conocimiento;
- **Decision Wizard**, como custodio de la sabiduría operacional;
- **K-Agile Facilitator**, como garante del proceso empírico.

La asignación nominal de roles sin ejercicio real de sus responsabilidades **no constituye conformidad**.

La fusión informal de roles que degrade sus funciones normativas también invalida la conformidad.

19.4 Conformidad de artefactos.

Una implementación conforme con K-Agile debe mantener, de forma explícita y actualizada:

- una **Base de Conocimiento** viva y trazable;
- un **Knowledge Backlog** priorizado por incertidumbre e impacto decisional;
- un **Sprint Knowledge Backlog** por Sprint;
- y un **Incremento de Conocimiento Operacional** al final de cada Sprint.

La existencia de conocimiento implícito, no documentado o no trazable constituye una **violación directa de la conformidad**.

19.5 Conformidad de eventos.

Todos los eventos prescritos por K-Agile deben ejecutarse:

- con la frecuencia definida;
- dentro de sus límites temporales;
- y cumpliendo su propósito epistemológico específico.

En particular:

- la Planificación del Sprint debe definir un **Objetivo Cognitivo explícito**;
- el Daily K-Agile debe inspeccionar **conocimiento y evidencia**, no tareas;
- la Revisión del Sprint debe evaluar **validez y capacidad de decisión**;
- la Retrospectiva del Sprint debe producir **mejoras explícitas del proceso cognitivo**.

La conversión de eventos en rituales administrativos o reportes de estado **invalida la conformidad**.

19.6 Conformidad de decisiones.

Una implementación conforme con K-Agile debe poder demostrar que:

- las decisiones operativas se basan exclusivamente en conocimiento declarado como **“Known”**;
- las decisiones óptimas cumplen la definición de **“Wisdom”**;
- las funciones objetivo y solvers son explícitos, trazables y justificados;
- y las decisiones pueden ser explicadas y auditadas.

El uso de decisiones opacas, no trazables o basadas en supuestos no validados **no es compatible con K-Agile**.

19.7 Exclusiones explícitas.

No constituyen K-Agile, aunque utilicen terminología similar:

- implementaciones ágiles centradas únicamente en entrega funcional;
- marcos de MLOps sin gobierno explícito del conocimiento;
- procesos de IA basados en experimentación no trazable;
- adaptaciones que eliminen o debiliten roles, artefactos o eventos;
- reinterpretaciones del marco sin versionado explícito.

El uso del nombre K-Agile sin cumplimiento normativo **no implica conformidad**.

19.8 Evaluación de conformidad.

La conformidad con K-Agile puede evaluarse mediante:

- inspección de artefactos;
- observación de eventos;
- trazabilidad entre conocimiento, evidencia y decisiones;
- y verificación del uso de las definiciones normativas del marco.

K-Agile está diseñado para ser **auditable**, precisamente porque todo lo relevante es explícito.

19.9 Responsabilidad sobre la conformidad.

La responsabilidad última de la conformidad con K-Agile recae en:

- la organización que adopta el marco;
- y los responsables de su aplicación efectiva.

K-Agile no impone certificaciones externas. Impone **disciplina epistemológica observable**.

20. Nota final normativa.

K-Agile es un **marco normativo abierto y gratuito**, diseñado para su aplicación en contextos de alta complejidad cognitiva, investigación avanzada, inteligencia artificial e inteligencia de decisiones. Su uso no está restringido por licencia, pero sí por **disciplina metodológica**.

K-Agile **existe y funciona únicamente como un todo coherente**. No es un conjunto de prácticas independientes ni un catálogo de recomendaciones. La eliminación, sustitución o reinterpretación arbitraria de cualquiera de sus principios, roles, artefactos, eventos o definiciones normativas **invalida su aplicación**.

En K-Agile:

- el **conocimiento explícito, validado y trazable** es la **unidad central de progreso**;
- la **evidencia** es el único fundamento legítimo para afirmar aprendizaje;
- y la **sabiduría operacional** —expresada mediante decisiones óptimas, explicables y contextualizadas— es el criterio último de valor.

K-Agile rechaza explícitamente los enfoques en los que:

- el conocimiento permanece implícito;
- las decisiones se toman sobre supuestos no validados;
- la actividad se confunde con progreso;
- o la complejidad se oculta en sistemas opacos.

Este marco no promete rapidez, ni certeza absoluta, ni éxito garantizado.

Promete algo más exigente y necesario: **rigor epistemológico, control decisional y aprendizaje verificable**.

K-Agile no sustituye el juicio humano, pero **lo hace explícito, gobernable y responsable**.

No elimina la incertidumbre, pero **la gestiona de forma consciente y trazable**.

No garantiza decidir bien siempre, pero **garantiza no decidir a ciegas**.

“K-Agile, un sistema progresa únicamente cuando **conoce mejor, decide mejor y puede demostrarlo**”.

Con esta nota, la **Guía Oficial de K-Agile** queda cerrada en su versión vigente. Toda evolución futura deberá realizarse mediante **versionado explícito** de este documento, preservando la coherencia y los principios aquí establecidos.

Alineación con el marco legal europeo en materia de inteligencia artificial

K-Agile está diseñado de forma **compatible por construcción** con los principios del marco legal de la Unión Europea en materia de inteligencia artificial, incluyendo los requisitos relativos a **gobernanza, trazabilidad, explicabilidad, gestión del riesgo y supervisión humana**.

El marco K-Agile no sustituye las obligaciones legales aplicables ni constituye, por sí mismo, una certificación de cumplimiento normativo. No obstante, proporciona una **estructura metodológica explícita** que facilita de manera significativa el cumplimiento, la demostrabilidad y la auditoría de dichas obligaciones.

En particular, K-Agile:

- exige **trazabilidad completa** entre conocimiento, evidencia y decisiones, facilitando la rendición de cuentas;
- impone **gestión explícita del riesgo cognitivo y decisional**, alineada con enfoques de evaluación de riesgos;
- integra la **explicabilidad** como propiedad estructural de las decisiones, no como añadido posterior;
- establece **roles claros de responsabilidad**, evitando ambigüedad en la gobernanza del sistema;
- y promueve la **supervisión humana informada**, mediante la integración explícita de inteligencia colectiva.

Estas características permiten que sistemas desarrollados bajo K-Agile puedan ser **inspeccionados, justificados y evaluados** conforme a los principios de la regulación europea vigente en cada momento.

La responsabilidad última del cumplimiento legal recae en las organizaciones que desarrollan, despliegan u operan los sistemas. K-Agile proporciona el marco metodológico necesario para que dicho cumplimiento sea **viable, verificable y sostenible** a lo largo del tiempo.