

GUÍA OFICIAL

FX | FRAMEWORK

Framework de trabajo para equipos AI-Native
v1.0

Por Flink X · Mayo 2026 · Buenos Aires

Sobre esta guía

FX surge de la necesidad de **sistematizar una forma de trabajo que ya está sucediendo**. Para hacerlo reunimos a **ingenieros, product managers, agilistas, AI engineers, formadores** y otros roles, con el objetivo de ponernos de acuerdo en lo que está pasando hoy y proponer un marco de buenas prácticas para equipos AI-native.

No esperamos llegar a una versión definitiva y estática. La práctica AI-native cambia demasiado rápido como para inmortalizar una guía definitiva. Esperamos seguir recibiendo feedback de muchos otros equipos y profesionales que están experimentando y construyendo conocimiento sobre **ways of working AI-first** — y juntos mantener el marco vigente.

En síntesis

- **Qué es:** es un framework de trabajo para equipos donde conviven personas de **carbono** (humanos) y de **silicio** (agentes de IA), colaborando como miembros del mismo squad.
- **Para quién:** nace con foco en equipos que desarrollan **productos digitales** que ya usan o planean usar agentes como parte sustancial de su flujo de trabajo.
- **De dónde parte:** toma los frameworks ágiles tradicionales como punto de partida y los extiende donde la lógica AI-native exige nuevos flujos de trabajo y toma de decisión.
- **Qué cambia:** introduce cambios en elementos tales como **roles, ceremonias y artefactos** que hasta ahora eran standard en muchos equipos y organizaciones.
- **Qué mantiene:** muchos de los valores y principios siguen vigentes: **el empirismo, trabajo iterativo, y la convicción de que necesitamos formar equipos y no sólo grupos de trabajo**, son algunos de ellos. FX cambia el sistema operativo, no la filosofía.

Contenidos

- 00 Prólogo: por qué FX Framework
- 01 Definiciones fundacionales
- 02 Roles
- 03 Eventos
- 04 Artefactos
- 05 Definition of Done expandida
- 06 Glosario
- 07 Cierre y próximas iteraciones

00 · Prólogo: por qué FX Framework

UNA REVOLUCIÓN COGNITIVA

- **Hasta ahora los frameworks fueron relativamente estables; el contexto, volátil.** FX parte de la premisa de que el framework es ahora por lo menos tan volátil como el contexto.
- **En la revolución industrial, el humano dejó de ejecutar tareas para pasar a auditar las máquinas.** Algo similar ocurre ahora, con una diferencia esencial: por primera vez delegamos acciones **cognitivas**, no sólo físicas. Estamos ante una **revolución cognitiva**.
- **El esfuerzo de producir código baja mucho.** En paralelo, la capacidad de tomar decisiones se facilita: al refinar, podemos **prototipar casi con el mismo esfuerzo**. El proceso decisorio siempre fue clave; ahora se potencia.
- **Sube la importancia del proceso de calidad.** Antes esperábamos que calidad algún día deje de ser una etapa separada con propuestas como TDD; ahora este proceso pasa a ser un **activo central** del equipo.
- **El humano-auditor pasa a ser una tarea clave.** No es un rol nuevo: es una característica central de los equipos AI-native.

CONTEXTO

Los frameworks ágiles que conocemos fueron diseñados bajo un supuesto implícito: el trabajo lo hacen personas. Toda la maquinaria de roles, ceremonias y artefactos asume que detrás de cada tarea hay un humano que entiende, decide y ejecuta.

Hoy ese supuesto ya no se cumple. **Una parte significativa del trabajo lo ejecutan agentes.** Los agentes generan código, escriben tests, redactan tickets, analizan logs, atienden clientes, etc.

No se trata de "implementar una nueva herramienta". Es un **cambio disruptivo en el sistema**. Cambia **quién** hace, **cómo** se coordina, **qué** significa que algo está hecho, y nos hace replantearnos supuestos como **valor y productividad**, entre otros.

01 · Definiciones fundacionales

1.1 Equipo AI-Native

Un equipo AI-Native es aquel donde los agentes son **miembros declarados** — con identidad, rol y contexto — y no herramientas que se invocan ad hoc.

1.2 Personas de carbono y personas de silicio

FX usa una distinción operativa, no metafísica:

- **Personas de carbono:** los humanos del squad. Tienen criterio, contexto implícito, responsabilidad legal y la capacidad de decir "esto no se delega". **La accountability del trabajo del squad es siempre y exclusivamente del carbono.**
- **Personas de silicio:** los agentes que el squad gobierna. Tienen identidad declarada, rol asignado y contexto curado. Cada agente tiene un **owner humano** que responde por su performance.

Buscamos que **el humano se establezca en su rol de auditor** sin transferir accountability.

1.3 Prompt-first workflow

En FX, la historia de usuario sigue siendo la unidad mínima de trabajo. Lo que cambia es el flujo a su alrededor — el **prompt-first workflow**: un workflow donde se especifica con qué agentes vamos a ejecutar la historia, qué prompts les vamos a dar, qué contexto consumen, dónde un humano interviene a validar, y cómo se loguea cada paso para auditoría posterior.

Le decimos **prompt-first** porque el prompt pasa a ser un **artefacto de importancia central en todo el proceso**: es versionado, revisado, reutilizable, asignado a un agente con identidad declarada. Si el código fue durante veinte años el activo central del equipo, en FX ese rol pasa a compartirse entre **el código, los prompts, los contextos curados y las definiciones de checkpoint**.

Ejemplo concreto

Pongamos como ejemplo la siguiente historia de usuario: "**Como usuario quiero filtrar las transacciones por rango de fechas**". Así es como se ejecuta esta historia en un equipo tradicional vs en un equipo AI-native. Aclaramos que son ejemplos ilustrativos y no prescriptivos.

En un equipo tradicional: el PO redacta la historia y se refina en equipo. Un dev toma la historia al comenzar el sprint, escribe el código del filtro en backend, hace el componente UI en frontend, escribe tests, abre un PR, espera code review, ajusta, mergea. Tiempo medido en story points; calidad medida por la tasa de bugs post-merge.

En un equipo AI-native, el mismo trabajo se diseña como un prompt-first workflow:

- **Paso 0 — Product Engineer (con asistencia de un Story Agent):** redacta la historia inicial a partir de la necesidad del negocio. El agente sugiere casos de uso, criterios de aceptación tentativos y dependencias visibles. Output: historia escrita y validada por el PE.
- **Paso 1 — Spec Agent (silicio):** toma la historia, lee el contexto del módulo de transacciones del repo, propone un contrato de API, casos de borde y criterios de aceptación verificables. Output: spec en markdown.
- **Checkpoint humano A — Arquitecto:** revisa el spec en 5 minutos. Aprueba, ajusta o reasigna. Si rechaza, el spec vuelve al backlog con feedback.
- **Paso 2 — Test Agent (silicio):** convierte la spec aprobada en un set de tests (unitarios e integración) que cubren los casos de borde. Los tests fallan (rojo) porque la implementación no existe.
- **Paso 3 — Implementation Agent (silicio):** genera el código que pasa los tests. Itera hasta verde.
- **Checkpoint humano B — Engineer:** revisa código y tests. Valida que el código no introduce regresión en otras partes del módulo, que la spec se respeta, y que los tests realmente cubren los criterios de aceptación. Aprueba o pide ajustes.
- **Paso 4 — UX Agent (silicio):** genera el componente de filtro siguiendo el design system. Output: PR de frontend.
- **Checkpoint humano C — Designer / PM:** valida UX y aprueba el merge.

Lo que cambia respecto a la versión tradicional: el dev no escribe código de la feature, escribe **la especificación que los agentes consumen**. La productividad deja de medirse por velocidad de tipeo y pasa a medirse por la calidad de los prompts y los checkpoints. La auditoría no se hace sobre el commit ("¿quién escribió esta línea?") sino sobre el flujo ("¿qué agente, con qué prompt, contra qué contexto, aprobado por quién?").

Anatomía de un prompt-first workflow

Un prompt-first workflow bien diseñado define:

- **Agentes participantes:** cada uno identificado por su tarjeta del Agent Roster (nombre, rol, modelo, owner).
- **Prompts en uso:** referencias versionadas a la Prompt Library, no improvisaciones de chat.
- **Contexto que consume cada agente:** qué archivos, qué documentos, qué porciones del repo. El contexto curado es el activo principal del equipo.
- **Checkpoints humanos:** dónde un humano accountable revisa, qué decide en ese punto, y qué pasa si rechaza.
- **Logging y auditoría:** qué se guarda de cada paso para reconstruir la decisión más adelante (prompt, contexto, output, identidad del revisor humano, timestamp).

02 · Roles

2.1 Product Engineer (PE)

Mantiene las responsabilidades de un product manager. Es el responsable del producto y del valor que el mismo entrega tanto a los usuarios como al negocio. Al mismo tiempo, el Product Engineer es **semi-técnico**: puede leer specs, entender prompts, evaluar outputs técnicos y prototipar. No es ingeniero senior, pero sí lo suficientemente técnico para participar en el diseño de los prompt-first workflows.

Responsabilidades expandidas en FX:

- Validar que los prompt-first workflows generan valor de negocio, no sólo eficiencia técnica.
- Decidir, junto al squad, qué partes del flujo se delegan a agentes y cuáles se mantienen humanas.
- Participar activamente en el diseño de los workflows prompt-first, no sólo en la validación final.
- Aceptar el incremento sólo cuando cumple la Definition of Done expandida (§05).

2.2 FX Coach (FXC)

Mantiene las responsabilidades de un Scrum Master o Team Coach. Al mismo tiempo acompaña al equipo y a la organización en la **transformación AI-first**. Hacia adentro del squad —coachea, facilita, sostiene el framework—, hacia afuera —construye los puentes con la organización para que el squad pueda operar.

Responsabilidades hacia el squad:

- **Coachear y facilitar.** Acompañar al equipo en el trabajo iterativo. Facilitar las ceremonias del framework. Remover impedimentos. Proteger al squad de la "ansiedad de IA": foco en outcomes, no en demostrar que se usa IA.
- **Sostener la coherencia del framework.** Mantener viva la Trust Boundaries Matrix (§4.5). Asegurar trazabilidad: prompts versionados, decisiones de agentes documentadas, contexto auditable. Velar por la salud sistémica del agent roster (sin ser su owner técnico — ese rol es del AI Engineer).

Responsabilidades hacia la organización:

- **Navegar la organización.** Ayudar al squad a moverse entre áreas, comités, gobierno y lineamientos corporativos. El squad construye; el FX Coach despeja el camino.

- **Generar acuerdos.** Entre el squad y los stakeholders, y con áreas como legal, compliance, riesgo o infraestructura. Es el rol que destraba donde el equipo choca con la política organizacional.
- **Sostener la mirada de transformación.** Mantener viva la conversación de adopción AI-first más allá del squad. Acompañar a otros squads y áreas que vienen detrás. Customizar y evolucionar la aplicación del marco según el contexto, y traer aprendizajes al framework global.

2.3 AI-Native Squad

Entrega incrementos de producto que cumplen la Definition of Done expandida, **gobernando un agent roster** que ejecuta partes sustanciales del trabajo.

El squad — composición humana

- **Engineers:** orquestan agentes en lugar de codear directamente. Diseñan workflows, revisan outputs, integran sistemas. La habilidad central deja de ser "saber escribir código" y pasa a ser **saber implementar IA en procesos**.
- **Quality Engineers (QEs):** revisan outputs de agentes con criterio. Es un rol que puede ser una persona dedicada o rotar dentro del squad.
- **Product Designer:** mismo mandato que en frameworks ágiles tradicionales, pero participa en el diseño de los prompt-first workflows.

El agent roster — agentes bajo gobernanza

Los agentes **no son miembros plenos del squad**. Son personas de silicio que el squad gobierna. Cada agente activo tiene una **tarjeta de identidad** en el **Agent Roster (§4.5)**:

- Nombre y rol declarado ("Reviewer-01", "TestGen-Backend", etc.).
- Contexto que consume: datasets, documentos, sistemas a los que tiene acceso.
- Modelo subyacente y versión.
- Owner humano: la persona que responde por su performance y firma sus outputs.
- Métricas de performance vigentes.

Roles adicionales

Dependiendo del tamaño de la organización y del momento del equipo, estos roles pueden participar con dedicación parcial o total en un squad.

- **AI Engineer.** Owner técnico del sistema de silicio. Diseña los prompt-first workflows, mantiene la **Prompt Library (§4.4)** y es **owner del Agent Roster (§4.3)**: actualiza tarjetas de identidad, audita la consistencia de los agentes con la Prompt Library, y propone altas, bajas y cambios al squad.
- **Arquitecto de Software.** Supervisa la arquitectura del código y de los sistemas que el squad construye, en línea con la **estrategia de negocio y el roadmap**. En un equipo donde gran parte del código lo escriben agentes, este rol es crítico: los agentes optimizan local sin ver la foto grande, y alguien tiene que cuidar la coherencia arquitectónica del producto. Valida decisiones de delegación que tengan impacto estructural. Muchos developers están migrando hacia este rol — en equipos AI-native la arquitectura del producto pasa a ser una de las habilidades centrales.

03 · Eventos

Todos los eventos de los marcos iterativos siguen vigentes (Sprint, Sprint Review, etc.). FX no los reemplaza. Sólo describimos aquí los eventos donde FX hace cambios sustanciales o suma algo nuevo.

La tendencia son **equipos más pequeños que van a mayor velocidad**. En ese contexto, el standard actual de iteraciones de dos semanas queda demasiado largo. FX propone iteraciones de **una semana como standard**. Los timeboxes de los eventos que siguen están calibrados a ese ciclo.

3.1 Sprint Planning + Agent Briefing

Timebox: la mitad del timebox tradicional — **hasta 1 hora para un sprint de 1 semana**. Con prompt-first workflows declarados de antemano, parte del trabajo de definición se desplaza al Refinamiento (§3.4) y el planning se concentra en confirmar y arrancar. Se agrega un bloque dedicado: el **Agent Briefing**.

Estructura sugerida:

1. Bloque 1 — Sprint Goal y selección de items (igual que en planning tradicional).
2. Bloque 2 — Plan de trabajo humano (igual que en planning tradicional).
3. Bloque 3 — Agent Briefing: para cada item, ¿qué agentes participan? ¿qué prompt-templates se usan? ¿dónde están los checkpoints humanos? ¿qué prompts hay que adaptar?

3.2 Daily Sync

Timebox: 15 minutos. La daily **continúa prácticamente sin modificaciones**. Se agrega un update corto sobre las respuestas obtenidas de los agentes y si necesitan ajustes para ejecutar en el día.

3.3 Retrospectiva

Timebox: hasta 1 hora semanal — puede ser quincenal en equipos chicos. La retro mantiene el foco en el **equipo y sus interacciones, relaciones, comunicación y acuerdos de trabajo**. Es un espacio humano: comunicación, alineación, qué nos sostuvo, qué nos frenó.

De la retro salen acciones concretas, con owner y plazo.

3.4 Refinamiento

En FX el refinamiento es un **checkpoint formal a mitad de semana** — puede ubicarse post-daily. Tiene dos objetivos:

4. Revisar el próximo Sprint Goal y los items candidatos a entrar.
5. Validar los agentes requeridos para esos items: ¿tenemos lo que necesitan (contexto, prompts, owners)?

El output del refinamiento es la lista de items **listos para Sprint Planning**, con sus prompt-first workflows esbozados.

04 · Artefactos

Mismo caso que los eventos. Sólo describimos aquí cambios sustanciales o agregados.

4.1 Sprint Backlog (extendido)

Con una extensión respecto al uso tradicional: cada ítem del Sprint Backlog incluye, además de su descripción y criterios de aceptación, su **workflow asociado** — qué agentes participan, qué prompts se usan, dónde están los checkpoints humanos.

4.2 Definition of Done expandida

Tratada en detalle en §05. Aquí va el principio: en FX la DoD agrega criterios sobre los outputs de agentes, prompts versionados, decisiones auditables y métricas de calidad observadas por el squad.

4.3 Agent Roster

Lista viva de los agentes activos del squad, con la tarjeta de identidad de cada uno (§2.3). Es un artefacto vivo: se actualiza en el Refinamiento (§3.4) o cuando el AI Engineer detecta que un agente necesita ajuste.

Cada agente activo debe tener un owner humano declarado y un contexto curado.

Un agente sin owner es un agente sin dueño. Un agente sin contexto curado es una automatización ciega. Si en cualquier momento la lista contiene un agente que no cumple esto, el FX Coach lo retira o re-asigna.

4.4 Prompt Library

Repositorio versionado de prompts y workflows prompt-first del squad. No es un Notion abandonado: vive en el repositorio del producto, con changelog y review.

Ningún prompt productivo vive sólo en la cabeza de una persona.

La Prompt Library es lo que hace replicable y transferible el trabajo del squad. Es también el activo que se preserva cuando hay rotación.

4.5 Trust Boundaries Matrix

Matriz que define, para cada categoría de tarea, tres niveles de delegación:

- **Autónomo:** el agente decide y ejecuta sin supervisión por instancia (sí auditoría agregada).
- **Asistido:** el agente propone, un humano valida antes de ejecutar.
- **Humano-only:** la decisión nunca se delega a un agente.

Ninguna decisión crítica se ejecuta sin un humano accountable.

La matriz se revisa al menos una vez por trimestre. Lo que hoy es "asistido" puede pasar a "autónomo" cuando los datos lo respaldan, y al revés.

Responsabilidad siempre humana

Algunas decisiones son —por ahora— **100% humanas**, sin importar cuánto mejoren los modelos. La razón es simple: estas decisiones pueden tener consecuencias legales, contractuales o estratégicas que requieran un humano accountable. Algunos ejemplos son:

- La IA no valida el dominio final del negocio.
- La IA no aprueba arquitectura crítica.
- La IA no acepta incrementos como "done".
- La IA no autoriza despliegues a producción.
- La IA no firma compromisos con clientes externos.

05 · Definition of Done expandida

La DoD es el contrato compartido entre el squad y el negocio sobre qué significa "hecho". En los frameworks ágiles tradicionales incluye criterios de calidad técnica (tests, code review, deploy). En FX, la DoD se expande con criterios específicos del trabajo humano-agente.

Lo que sigue es un ejemplo y una sugerencia, no algo fijo. Cada squad arma su propia DoD colectivamente y la **evoluciona cuando es necesario**.

5.1 Criterios standard sugeridos

- **Outputs de agentes revisados por un humano del squad.** Cada output que pasa a producción tiene un humano que lo aprobó o validó por muestreo según la Trust Boundaries Matrix.
- **Prompts utilizados versionados en la Prompt Library.** Si el output del item se generó con un prompt nuevo o modificado, ese prompt está en la library con su changelog.
- **Decisiones de agentes auditables.** Para cada decisión que tomó un agente, se puede reconstruir: prompt usado, contexto consumido, modelo, output, owner.
- **Métricas de calidad observadas.** Hallucination rate, tasa de aceptación humana, costo por outcome — el squad las inspecciona sprint a sprint y define umbrales propios.

06 • Glosario

Agent Roster — Artefacto FX. Lista viva de agentes activos del squad con su tarjeta de identidad.

Carbono (persona de) — Humano del squad. Tiene la accountability legal y de criterio del trabajo del squad.

DoD expandida — Definition of Done con criterios FX adicionales: outputs revisados, prompts versionados, decisiones auditables, métricas de calidad observadas.

Prompt-first workflow — Workflow declarado en torno a una historia, donde los agentes ejecutan pasos y los humanos validan en checkpoints definidos. Reemplaza al flujo informal alrededor de la historia.

Owner humano — El humano accountable por la performance de un agente. Cura su contexto y revisa sus outputs.

Prompt Library — Artefacto FX. Repositorio versionado de prompts y workflows del squad.

Silicio (persona de) — Agente con identidad declarada, rol asignado y contexto curado, gobernado por el squad humano. No es miembro pleno: ejecuta, no firma.

Trust Boundaries Matrix — Artefacto FX. Define qué decisiones son autónomas, asistidas o humano-only.

07 · Cierre y próximas iteraciones

ESTADO DE ESTA VERSIÓN

Esta es la **versión v1.0** del FX Framework. Cubre la estructura mínima: roles, eventos, artefactos, DoD expandida y glosario.

CURADO, NO INVENTADO

FX se inscribe en una conversación más amplia. Esta guía toma como referencia y extiende ideas de: **Scrum, spec-driven development, modelos de relación humano-agente, la literatura sobre sistemas multi-agente y research empírico sobre identidad de agentes.**

CÓMO COLABORAR

FX Framework es un sistema en evolución, en base a conversaciones con expertos de todo el mundo. Si estás aplicando FX en tu organización, tus aprendizajes alimentan la próxima versión. Escríbenos a info@flink-x.com.



Framework de trabajo para equipos AI-Native
flink-x.com · info@flink-x.com