



onext
technology

Taller · TRGCON 2026

Ata en corto a tus agentes: SDD en directo

Bernat López, CEO de onext · **Manel Marcos**, cofundador y responsable técnico

La Nave, Madrid · 2026-10-22 · 15:30–18:00

Qué vas a ver esta tarde

Qué te llevas



El flujo completo, de la idea a la documentación actualizada, y dónde decide una persona.



Un *harness* mínimo, el contexto en el que trabaja la IA (constitución, especificación y un control de verificación), sobre el repo de ejemplo.



El criterio para decidir qué gobierna la especificación y qué se deja al agente.

Agenda

15:30	La promesa rota Acto 1 · Bernat
15:40	Entendiendo las limitaciones Acto 2 · Bernat
15:50	El nuevo paradigma (el <i>harness</i>) Acto 3 · Bernat
16:30	De la spec al código, en directo Acto 4 · Manel
16:45	Pausa 15 min
17:00	Móntalo tú Acto 5 · Manel
17:45	Qué se queda en tu equipo, y preguntas Acto 6 · Manel y Bernat

Acto 1

La promesa rota

Nos dijeron que la IA nos haría 10x más productivos.

En algunos casos aceleramos.

En muchos otros, estamos generando más código... **y más problemas.**

Con la IA, los devs tenemos superpoderes.

O eso dice internet:

David Bonilla reposteado



Alfredo

@brolag

Adiós programadores.

La IA ya los reemplazó.

Hice esta app en 10 minutos sin saber nada de programación.

<http://localhost:3000/>

16:16 · 06 ene 26 · **164K** Visualizaciones

618 Republicaciones **51** Citas

4.880 Me gusta

265 Elementos guardados

El problema del vibe coding

Febrero de 2025: **Andrej Karpathy** (ex-OpenAI, ex-Tesla) le pone nombre.

Vibe coding

Te dejas llevar, aceptas lo que propone la IA y ajustas hasta que parece que funciona.

El problema

**No escala.
Y menos en equipo.**

Nació para **prototipos desechables**.

Giro de guion

Creían haber ido
un 20 % más rápido

Tardaron
un 19 % más

Estudio controlado de **METR (2025)**: desarrolladores con experiencia tardaron un 19 % más con asistentes de IA y creían haber ido un 20 % más rápido.

La sensación es de velocidad. Sin método, el resultado es retrabajo.

Muestra pequeña y herramientas de principios de 2025.

Dos años, la misma señal

Qué ve DORA, el programa de investigación de Google sobre equipos de software, a medida que sube la adopción de IA.

2024

Ritmo de entrega **-1,5 %**

Estabilidad **-7,2 %**

Estimado, a medida que subía la adopción de IA.

2025

Ritmo de entrega **Positiva**

Estabilidad **Negativa**

Relación con la adopción de IA. Casi 5.000 profesionales.

La velocidad la pone la herramienta. La estabilidad, el método.

Fuente: DORA, «Announcing the 2024 DORA report» (23-10-2024) y «Announcing the 2025 DORA Report» (23-09-2025), Google Cloud Blog. Correlaciones en encuestas, no causas; las cifras de 2024 son estimaciones y 2025 no da porcentaje para la estabilidad.

Más rápido no es más fiable

Mejora que declaran 293 responsables de tecnología e ingeniería, frente a su situación antes de la IA.



Velocidad de release

+27 %

de mejora hoy

Esperan **+67 %** en 1–2 años



Productividad

+20 %

de mejora hoy

Esperan **+39 %** en 1–2 años



Resiliencia operativa

+16 %

de mejora hoy

Esperan **+27 %** en 1–2 años

Con IA, el software sale antes. De las tres, la resiliencia es la que menos mejora.

Fuente: Bain, «AI in Software Engineering: The Reliability Paradox» (19-10-2026). Bain Tech and Engineering Survey, abril de 2026, n = 293. Mediana del aumento frente a la situación previa a la IA; cifras redondeadas y declaradas por los encuestados.

A superhero with dark curly hair, wearing a blue suit with a yellow belt and yellow boots, and a black mask, is flying through a city. The city is constructed from blue and white blocks, resembling a digital or pixelated environment. In the background, there are various blocky buildings, a blue car with orange flames, a blue helicopter, and a large blue building with a yellow warning sign. The sky is blue with white clouds and a few birds. The superhero is in the center, flying towards the right with a determined expression.

Sin método, el superhéroe deja la ciudad patas arriba.

Con IA, escribir código ya es lo barato. Comprobarlo cuesta lo mismo que antes.

«Funcionaba en mi prompt.» Lo dijo entre los escombros.

Acto 2

Entendiendo las limitaciones

Todo superhéroe tiene sus villanos. Y estos tres no salen en los anuncios.

El olvidadizo · El ilusionista · La deriva



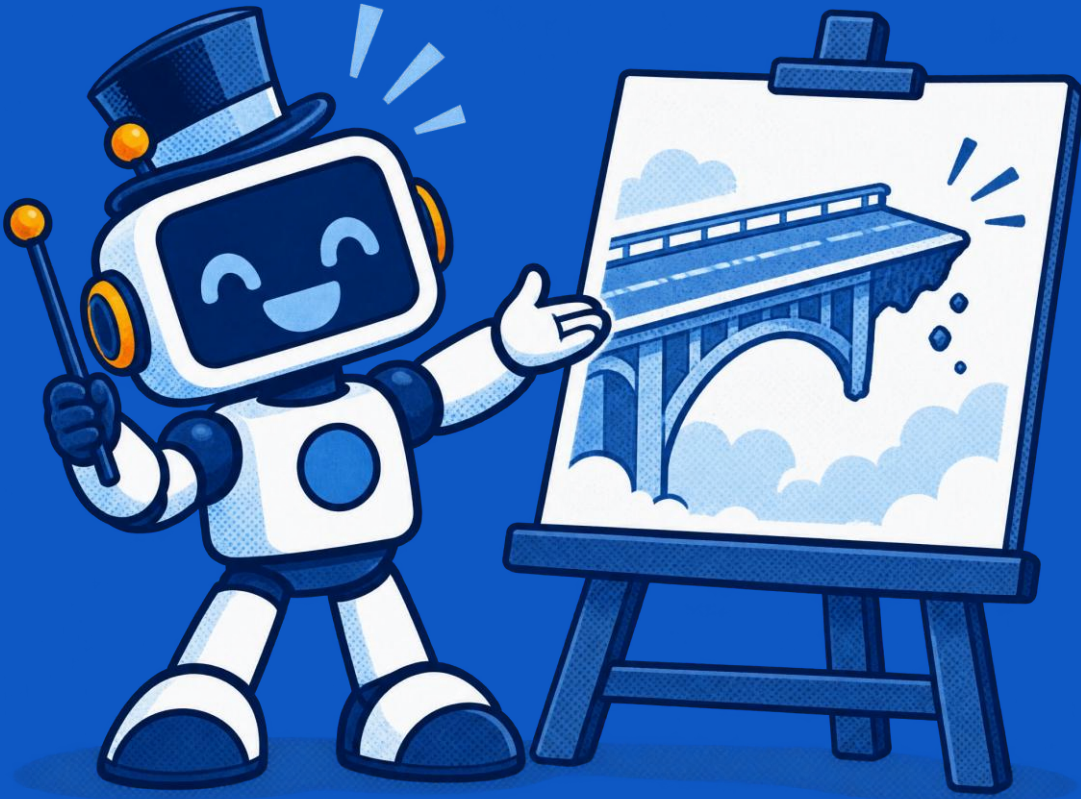
Villano

El olvidadizo

Cada mañana borra la pizarra de las reglas.

Límite real: memoria limitada

Riesgo: inconsistencias



Villano

El ilusionista

Código que parece perfecto. Y no lo es.

Límite real: plausible no es cierto

Riesgo: defectos silenciosos



Villano

La deriva

Le pides una baldosa y rehace el puerto.

Límite real: deriva con el tiempo

Riesgo: regresiones

Manifiesto

Rules over tools

Las reglas de tu equipo, por encima de cualquier herramienta.

El modelo no es la fuente de verdad.

El chat no es el sistema.

El código no es el único artefacto.

Lo que sabe tu equipo vive fuera del modelo: **escrito, gobernado y verificable.**

Acto 3

El nuevo paradigma

Contra los villanos no hace falta un modelo mejor: hace falta un cuartel general.

El *harness* es donde trabaja la IA: **contexto + spec + controles.**

Y *human on the loop*: **una persona decide donde hay riesgo.**

Un sistema de trabajo para desarrollar con agentes

El *harness*: contexto + spec + controles. Y *human on the loop*: una persona decide donde hay riesgo.



Contexto

Las reglas y el conocimiento de tu casa, escritos fuera del modelo y cargados en cada tarea (ingeniería de contexto).



Spec

Qué se construye y cómo se comprueba, aprobado antes del código (SDD: Spec-Driven Development, desarrollo guiado por especificaciones).



Controles

Comprobaciones automáticas que avisan o frenan lo que no cumple antes de producción.



Human on the loop: **una persona decide donde hay riesgo**, no en cada paso.

La IA amplifica. El equipo decide. El método escala.

Qué es el desarrollo guiado por especificaciones (SDD)

Cada especificación es una misión en el tablero: qué, para qué y cómo sabremos que está cumplida.

Programar a base de prompts

- El código es la fuente de verdad
- Se pide y se prueba a ver si funciona
- Mantener es reescribir
- Cada persona lo hace a su manera

Guiado por especificaciones

- La especificación es la fuente de verdad
- Criterios comprobables, aprobados antes del código
- Mantener es actualizar la especificación
- Consistencia entre las distintas personas del equipo



La especificación vive con el código

- En el **repositorio** y en el **mismo** *pull request*.
- **Versionada** con el producto.

Si vive en otro sitio, se desactualiza. Y nadie la vuelve a abrir.

El cuartel general de tu equipo



El *harness* (contexto + spec + controles) tiene:



El libro de reglas

el contexto: la constitución del proyecto



El tablero de misiones

la spec: las especificaciones



Las alarmas

los controles: avisan o frenan

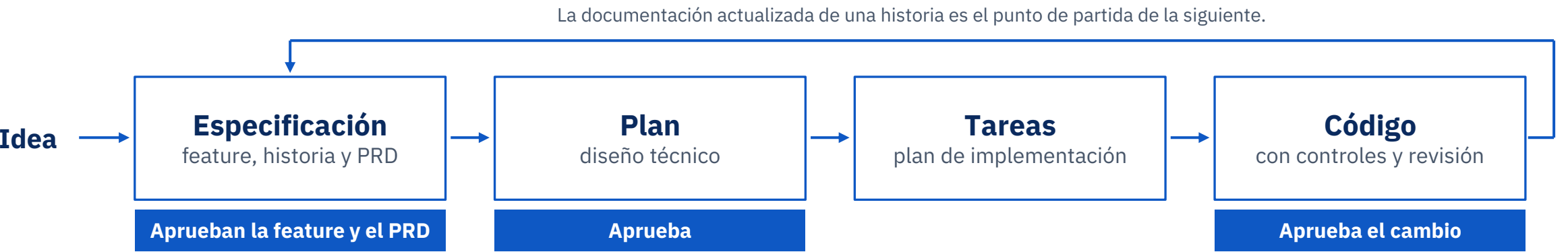


La silla donde decide una persona

feature, PRD, plan y el cambio antes de integrarlo

El *harness* se queda en el equipo: es suyo y lo hace crecer con cada proyecto.

El contexto y el proceso



Cuatro puntos de decisión: feature, PRD (documento de requisitos de producto), plan y el cambio.

Documentación del producto lo que la IA sabe de tu producto: escrita, versionada y en el repo. El <i>harness</i> ingeniería de contexto: cómo trabaja la IA en tu casa.	Especificaciones funcionales y casos de uso		Reglas de diseño gráfico sistema de diseño		Reglas de arquitectura estándares y seguridad		Decisiones tomadas qué se decidió y por qué	
	Constitución reglas del proyecto	Subagentes especialistas por fase	MCP conexión con herramientas	Comandos pasos repetibles	Skills habilidades	Hooks controles automáticos	Gates controles de calidad antes de avanzar	Modelos el adecuado a cada tarea

Arriba, **el proceso con el que trabajáis**, de la idea a la documentación actualizada. Abajo, **el contexto en el que trabaja la IA**.

El libro de reglas: la constitución

Los agentes solo son tan buenos como lo que saben de tu casa. Su *harness* lleva:



Las reglas de la casa, escritas para que la IA las aplique siempre.



El conocimiento del producto: módulos, decisiones y documentación.



Agentes especializados por fase; los que revisan son distintos de los que construyen.



Controles automáticos que frenan lo que no cumple antes de producción.

```
# Constitución · [tu producto]
```

```
## Arquitectura  
Módulos, capas y dependencias permitidas
```

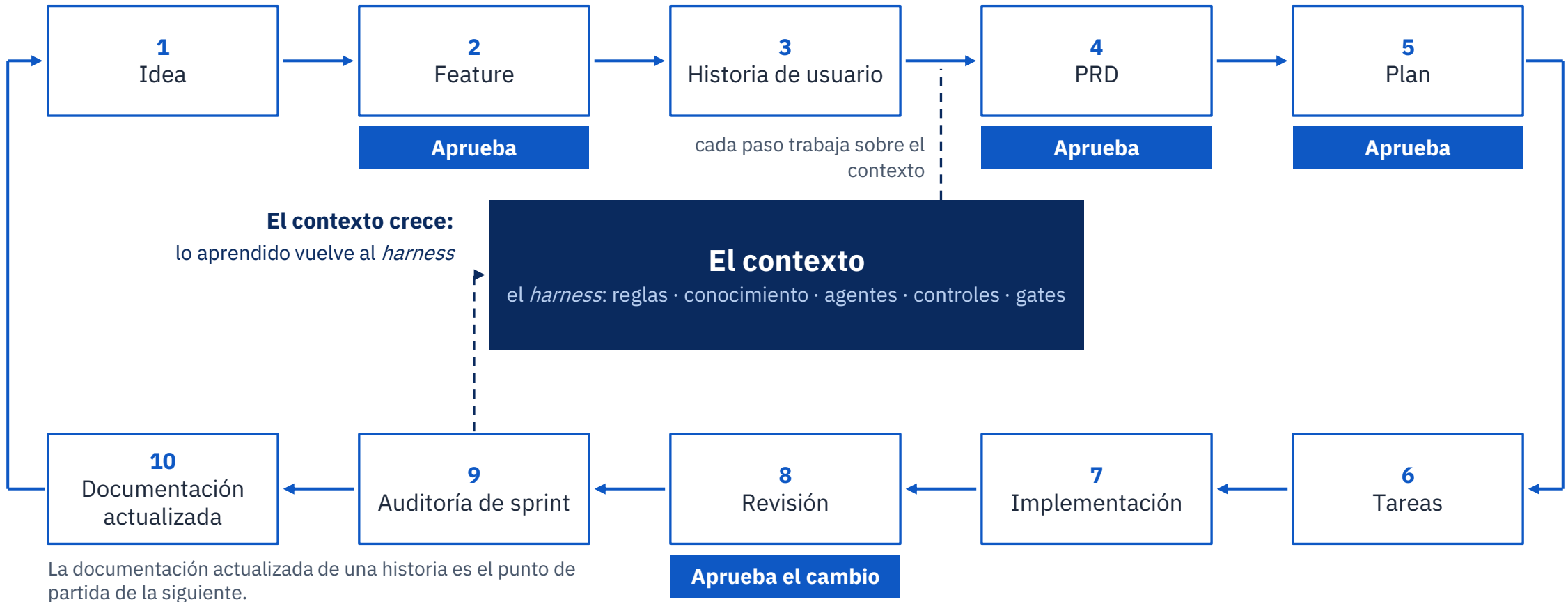
```
## Reglas de la casa  
Estándares, seguridad, accesibilidad
```

```
## Zonas sensibles  
API pública y migraciones de datos:  
no se tocan sin plan aprobado
```

```
## Cómo se trabaja aquí  
Plantillas de feature, PRD y plan; quién revisa  
qué
```

Lo que el olvidadizo no puede borrar: **se carga en cada tarea.**

Cómo trabaja el equipo: diez pasos en bucle



Cuatro puntos de decisión, y nada más: la feature, el PRD, el plan y el cambio antes de integrarlo. El resto no lleva firma: *human on the loop*, una persona decide donde hay riesgo.

Dónde vive cada cosa: la carpeta docs/

La del repo de la práctica. Cada paso del bucle escribe en su carpeta.

docs/

00_methodology/

01_roadmap/sprints/

02_discovery/

03_delivery/hu/

05_modules/

06_design_principles/

07_architecture_principles/

10_audits/

_templates/

WORKFLOW.md y compañía

El método en una página: el flujo y sus reglas, para leer antes de empezar.

Los sprints: qué historias entran y cómo se cierra cada uno.

Ideas, features y spikes, y las reglas de negocio ya validadas.

Una carpeta por historia: HU, PRD, plan, tareas, revisión y CHANGELOG.

La documentación viva de cada módulo: funcional, técnica y casos de uso.

Las reglas de diseño de la interfaz.

Las reglas de arquitectura: quality gates, decisiones tomadas y contratos.

Los veredictos: revisiones, auditorías de sprint e informes de QA de extremo a extremo.

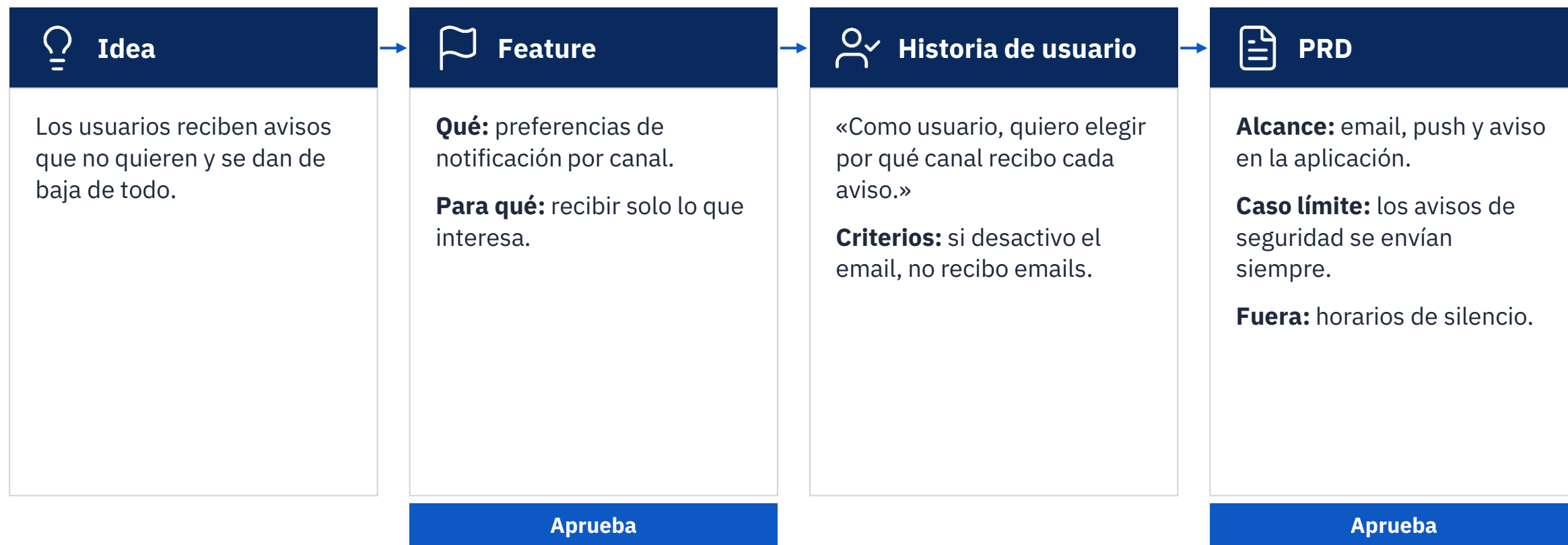
Las 15 plantillas de los artefactos: idea, feature, HU, PRD, plan, tarea, revisión...

En la raíz: el flujo, la arquitectura, la regla de implementación y el estado del MVP.

La documentación no es un anexo: **es el estado del proyecto**, y es lo que lee el agente antes de cada tarea.

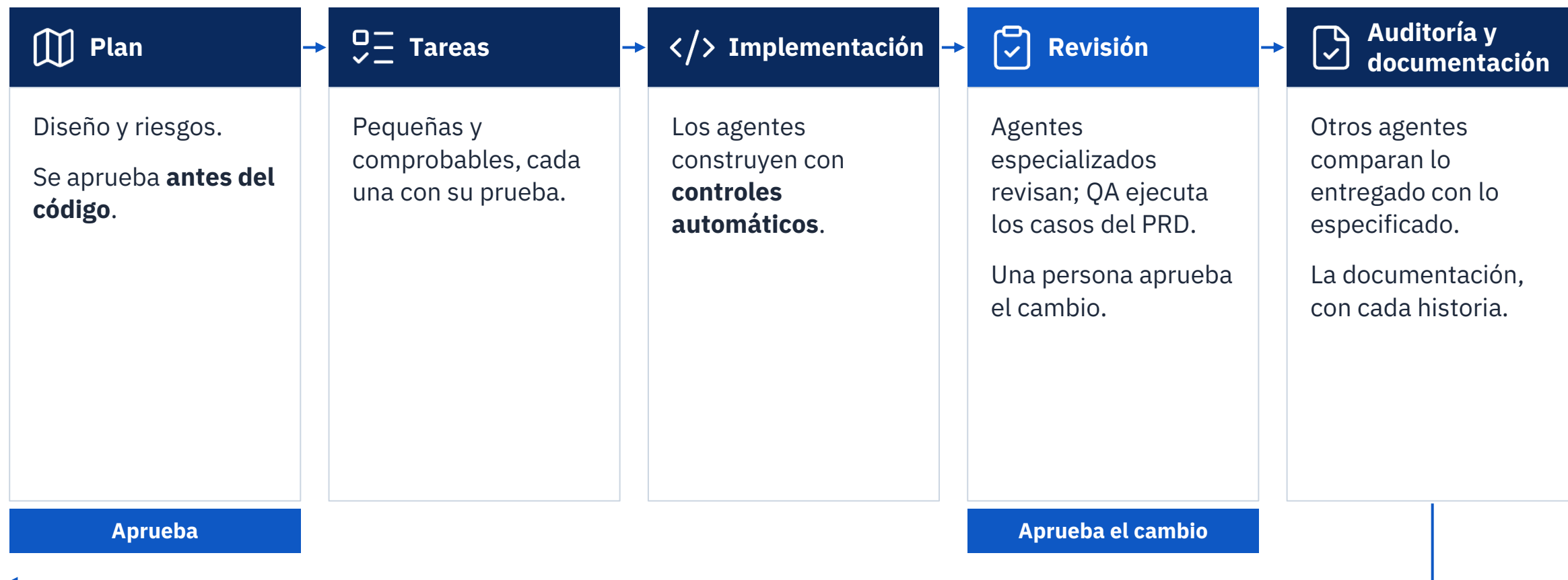
Una misión bien escrita: de la idea al PRD

Ejemplo ilustrativo: preferencias de notificación en una aplicación SaaS.



Criterios que se pueden comprobar. «**Funcionaba en mi prompt.**» Eso no cuenta.

Del plan a la documentación actualizada



La documentación actualizada de una historia es el punto de partida de la siguiente.

Producto trabaja de la idea al PRD. Ingeniería, del plan a la revisión.

Cuando algo falla, se arregla donde nació

Nunca se parchea el código para compensar una mala especificación.

 **Fallo de comportamiento** → Se corrige en la **especificación** **Si no, mañana...**

 **Problema de diseño** → Se corrige en el **plan** ...el olvidadizo vuelve a leer la especificación equivocada y reintroduce el fallo. Con mucha seguridad.

 **Desorden en la ejecución** → Se corrige en las **tareas**

«Funcionaba en mi prompt.» Ya no es excusa: **ahora hay especificación.**



Los superpoderes los tiene la herramienta. **El criterio es del equipo.**

La IA amplifica. El equipo decide. El método escala.

Acto 4

De la spec al código, en directo

Una feature completa: una persona la aprueba, y aprueba el PRD y el plan antes del código.

El agente implementa y un control frena lo que no cumple la especificación.

Una persona aprueba el cambio **antes de integrarlo**.

Pausa

Café, y el portátil a punto

Volvemos a las 17:00 con «Móntalo tú».

Si aún no tienes el repo de ejemplo, clónalo ahora: está en onext.es/trgcon26.

Trae una duda de tu proyecto: **la usamos después.**

Acto 5

Móntalo tú

Arranca tu propio *harness* mínimo sobre el repo de ejemplo.

Una constitución, una especificación y un control de verificación.

Lo que puedas replicar **el lunes**.

Acto 6

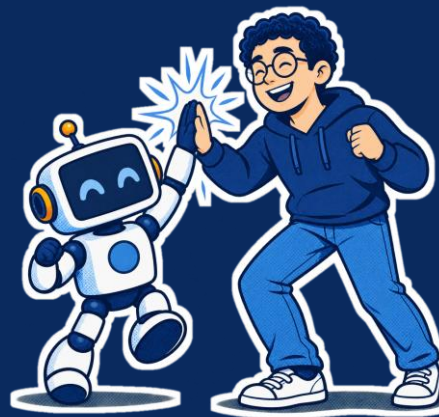
Qué se queda en tu equipo

La auditoría de sprint: otros agentes comparan lo entregado con lo especificado.

El sprint se cierra con su archivo y sus novedades; la documentación ya está al día.

El método se queda en tu equipo. **Y preguntas.**

Gracias. ¿Preguntas?



Los superpoderes, para la herramienta. El criterio, para tu equipo.

- ¿Dónde viven hoy las reglas de tu proyecto: escritas o en la cabeza de alguien?
- ¿Dónde firma una persona antes de producción, y dónde firma por costumbre?
- Si mañana cambiarais de asistente, ¿qué perderíais?

¿Lo montamos con tu equipo?

Pásate por el stand de onext · Diagnóstico gratuito



onext.es/trgcon26

presentación y repositorio

Bernat López

CEO de onext

Manel Marcos

Cofundador y responsable técnico

onext.es