BIG school

Apuntes

Curso de Desarrollo con IA

DE 0 A PRODUCCIÓN

DÍA 3

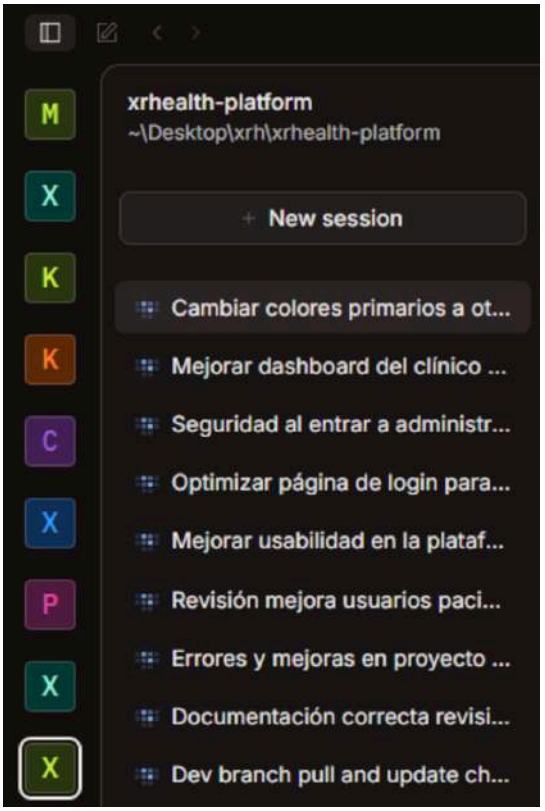
Índice

De código a producción: Testing, CI/CD y releases con IA

Más IA = Más velocidad = más riesgo oculto

La capacidad de generar código a velocidades masivas mediante agentes es una ventaja competitiva, pero también un riesgo sistémico. El código generado por IA sin tests es deuda técnica por diseño. La IA carece de contexto de negocio profundo y no comprende las ramificaciones de seguridad de cada línea que escribe.

- La IA genera código más rápido de lo que podemos revisarlo.
- Múltiples tareas en paralelo = pérdida de contexto.
- Código que "funciona" no significa código seguro o correcto.
- El ritmo de cambio supera la capacidad de entender qué cambia.
- Deuda técnica invisible que se acumula sin que nadie la vea.



La IA no conoce tu negocio

- ✓ La IA genera código rápido... pero no lo entiende profundamente.
- ✓ No conoce el contexto completo de tu sistema.
- ✓ No sabe qué casos edge importan en tu negocio.
- ✓ Producción exige confianza, no esperanza.

Principio de Responsabilidad Profesional: Aplicando la máxima de "**un gran poder conlleva una gran responsabilidad**", el desarrollador debe asumir que es el único responsable legal y profesional del código en producción. Si el sistema falla y compromete datos o ingresos, el cliente o el empleador no pedirán cuentas a Claude o a GPT; la responsabilidad recae íntegramente en el humano que aprobó el despliegue.

Testing cuando hay IA de por medio

Generar código con IA ≠ validar código. Sin tests, la IA construye sobre arena

La IA genera código rápido. Demasiado rápido para revisarlo manualmente.

Sin tests, cada cambio es una apuesta:

- ✗ ¿Sigue funcionando el login después de este cambio?
- ✗ ¿Rompí algo al refactorizar?
- ✗ ¿El código nuevo de la IA hace lo que yo creo que hace?

Con tests, tienes una red de seguridad que lo verifica por ti automáticamente.

¿Qué testear cuando programas con IA?

En un flujo donde el código se autogenera, los tests son la primera y más crítica barrera de defensa. Ignorar el testing es aceptar un caos técnico inevitable.

No se trata de testearlo todo, sino de blindar el valor de negocio. Priorizamos bajo este criterio:

Tip: Podemos delegar en la IA la redacción de la suite de pruebas, pero el desarrollador debe definir y validar que los escenarios de prueba sean realistas y cubran los casos de borde (edge cases) del negocio.

CI/CD mínimo viable

¿Qué es CI/CD?

CI — Integración Continua
Cada vez que alguien sube código, se compila y se pasan los tests automáticamente. Si algo rompe, se detecta al instante.

CD — Entrega/Despliegue Continuo
Si el código pasa los tests, llega a producción solo, sin que nadie tenga que pulsar ningún botón.

CI/CD es el guardián automático de tu producción.

¿Qué es Pull Request o PR?

Trabajas en tu rama, terminas un cambio y quieres integrarlo en main.
Un PR es una petición formal de "quiero meter este código".
Antes de que entre en main, dos cosas ocurren automáticamente:

- El CI lanza todos los checks (lint, tests, build...)
- Otros devs pueden revisar el código antes de aprobarlo.

Si algo falla, el merge queda bloqueado. Main nunca recibe código roto.

Las reglas de oro del CI/CD

- Toda feature entra vía Pull Request a main
- Nunca se hace push directo a main
- Todo PR ejecuta automáticamente:
    - Lint: el código tiene buen formato.
    - Validaciones: el código es seguro.
    - Tests: el código funciona.
- Si algo falla → el PR se bloquea.
- main siempre está lista para producción

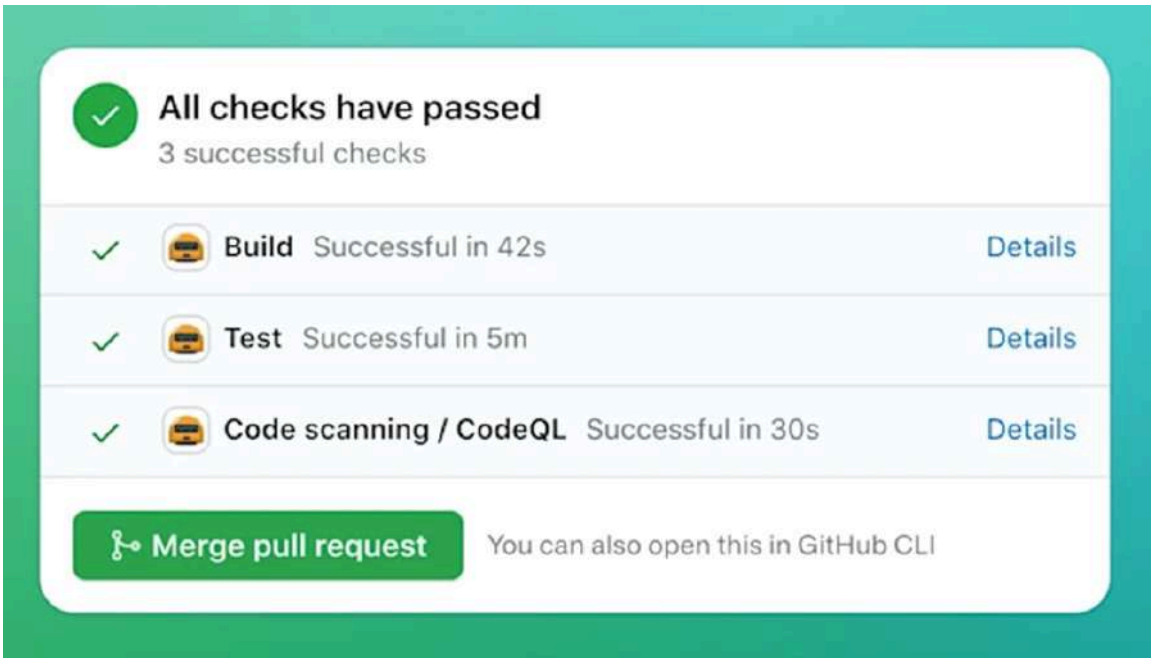
Truco: puedes bloquear la rama main de forma que solo acepte merges a través de pull requests.

GitHub Actions

La herramienta de CI/CD integrada en GitHub — no necesitas nada externo.

Cómo funciona:

- creas un archivo .yml en .github/workflows/
    - su redacción delegamos a la IA, y debe definir pasos claros como instalación de dependencias, ejecución de linter y paso exitoso de la suite de tests.
- defines cuándo se ejecuta (en cada PR, en cada push…)
- GitHub levanta una máquina virtual y ejecuta tus pasos
- si algo falla, te avisa y bloquea el PR



https://github.com/features/actions

CI/CD inteligente gracias a la IA

Code Review automático con agentes IA

Cada PR recibe una revisión automática de un agente.

El agente revisa:

- ¿Sigue las convenciones del proyecto?
- ¿Hay problemas de rendimiento obvios?
- ¿La lógica tiene sentido?
- ¿Faltan tests para algún caso?
- ¿Hay problemas de seguridad?

Algunas Herramientas: CodeRabbit, GitHub Copilot, Cursor Rules

Security Review automático con Claude

La seguridad no es un añadido; es un requisito arquitectónico.

¿Por qué es importante la revisión de seguridad en cada PR?

- La IA genera codigo rápido pero puede introducir vulnerabilidades
- SQL injection, XSS, secrets expuestos, dependencias vulnerables...

claude-code-security-review: GitHub Action oficial de Anthropic

- Analiza cada PR automáticamente con Claude
- Deja comentarios directamente en el PR
- Detecta: inyecciones, auth debil, datos sensibles, lógica insegura

Repositorio oficial: github.com/anthropics/claude-code-security-review

Release Please

El versionado de software suele sufrir de hastío burocrático, lo que lleva a documentaciones pobres. Resolvemos esto mediante el flujo **Release Please** de Google.

Funcionamiento de Release Please

Tras cada merge en main:

- Se crea automáticamente un Release PR
- Se actualiza package.json con la nueva versión

Al hacer merge del Release PR:

- Segundo deploy a producción
- Se crea el tag vX.Y.Z en git
- Se publica el GitHub Release
- Las release notes se generan con IA (Extra que añadimos nosotros)

Repositorio oficial: github.com/googleapis/release-please

Conclusiones

La inteligencia artificial ha dejado de ser una promesa para convertirse en el estándar operativo.

En la ingeniería de software actual, aprender a orquestar estas herramientas no es una opción, sino un imperativo para no quedar obsoleto.

El futuro del desarrollo pertenece a quienes asumen la responsabilidad profesional de guiar la tecnología hacia sistemas seguros, escalables y de alta calidad.

BIG school

{m}

Conviértete en Desarrollador con IA

MATRICÚLATE AQUÍ