EcomLabs

GUÍA PASO A PASO

Revisión de código con IA: checklist en dos niveles

Qué revisa la IA, qué hace el autor con cada hallazgo y qué revisa después una persona: reglas de negocio, decisiones y datos. Con la plantilla de descripción del cambio, CODEOWNERS para las rutas sensibles y las señales para no aprobar.

PARA	Equipos o personas que usan IA para revisar código antes del merge
PROBADO CON	Node 22.22.3 · patrones de CODEOWNERS comprobados con git check-ignore · práctica propia, sin cifras externas
ACTUALIZADA	19 de noviembre de 2026
DOCUMENTACIÓN	google.github.io/eng-practices/review
ARTÍCULO	ecomlabs.dev/g/review
AUTOR	Luis Morales · @luismoralesdev

Índice

Antes de empezar	Cómo usar esta checklist
Nivel 1	Lo que revisa la IA
Entre niveles	Lo que hace el autor
Nivel 2	Lo que revisa la persona
Antes del merge	Cuándo no aprobar
Ajustes	Una persona en las rutas sensibles

ANTES DE EMPEZAR

Cómo usar esta checklist

Dos niveles, en orden. El nivel 1 lo hace la IA sobre el diff, después de que pasan lint, tipos y pruebas. El autor responde cada hallazgo. El nivel 2 lo hace una persona, y solo su aprobación permite el merge.

NIVEL	QUIÉN	REVISA	DECIDE
0	CI	Formato, tipos, pruebas	Pasa o no pasa
1	IA	Lo que está en el diff	Nada: reporta
2	Persona	Lo que no está en el diff	Aprobar o no

NIVEL 1

Lo que revisa la IA

Dale el diff y un archivo de instrucciones versionado con las reglas de negocio cortas del repo. Pídele que reporte sin editar. Cada punto debe poder demostrarse con una entrada o una línea:

- [ ] Entradas límite: `null`, lista vacía, 0, negativos, el valor exacto del umbral.
- [ ] Qué pasa cuando una llamada falla, tarda o no devuelve nada.
- [ ] Un `await` que falta, un `catch` vacío, una promesa sin manejar.
- [ ] Números o textos mágicos sin nombre.
- [ ] Nombres y comentarios que dicen algo distinto de lo que hace el código.
- [ ] Ramas nuevas sin prueba, y pruebas que pasarían aunque el código estuviera mal.
- [ ] Pruebas borradas, saltadas o con el valor esperado cambiado.
- [ ] Secretos, tokens o datos personales en código, logs o fixtures.
- [ ] El diff contra cada regla de negocio escrita en el archivo de instrucciones.
- [ ] Al final: la lista de lo que no revisó.

Formato de cada hallazgo: `ruta:línea · bloquea | pregunta | menor · qué falla · entrada que lo demuestra`.

ENTRE NIVELES

Lo que hace el autor

Ningún hallazgo queda sin respuesta antes de pedir revisión humana:

HALLAZGO	QUÉ HACES
Bloquea, con entrada	Reprodúcelo, corrígelo y agrega la prueba con esa entrada.
Bloquea, sin entrada	Búscala tú. Si no la encuentras, responde por qué no aplica.
Pregunta	Responde en el PR. Si la respuesta es una regla, pásala al archivo de instrucciones.
Menor	Corrígelo si es barato; si no, déjalo anotado.
«Sin hallazgos»	Copia en el PR lo que la IA dijo que no revisó.

Y escribe la descripción para la persona, no para la IA. Esta plantilla le da lo que no está en el diff:

```
## Qué cambia
Una o dos frases.

## Regla de negocio que aplica
Ej.: envío gratis desde 999 MXN de subtotal después de descuentos.

## Rutas sensibles que toca
Pagos, migraciones, middleware, datos personales: cuáles y por qué.

## Qué no se probó
Lo que no cubren las pruebas y cómo lo verificaste a mano.

## Hallazgos de la IA
Bloquea: corregidos (lista) · Respondidos sin cambio: (lista con la razón)
No revisado según la IA: (lista)
```

NIVEL 2

Lo que revisa la persona

No repitas lo mecánico. Lee los hallazgos y las respuestas, lee el diff completo y gasta el tiempo aquí:

- [ ] El cambio cumple la regla de negocio que dice la descripción, y la regla sigue vigente.
- [ ] El archivo de instrucciones no contradice lo que se acordó; si cambió la regla, se actualiza en el mismo PR.
- [ ] El cambio debería existir: no agrega algo que no quieres en el sistema.
- [ ] Respeta las decisiones tomadas (librerías, lo que se aceptó no resolver).
- [ ] Quién llama a esto, qué queda público y qué dato es personal.
- [ ] A dónde van los logs y quién los lee.
- [ ] Dinero, inventario, pedidos pagados: el caso que cuesta caro si falla.
- [ ] Las respuestas del autor a los hallazgos se sostienen.
- [ ] Puedes explicar con tus palabras qué hace el cambio y por qué.

ANTES DEL MERGE

Cuándo no aprobar

- [ ] Un hallazgo «bloquea» sin corrección ni respuesta.
- [ ] Toca una ruta protegida y su dueño no la revisó.
- [ ] Una prueba cambió, se saltó o se borró sin razón escrita.
- [ ] Toca dinero, inventario o datos personales y la descripción no dice qué regla aplica.
- [ ] «Sin hallazgos» sin la lista de lo que no se revisó.
- [ ] Mezcla un refactor con un cambio de comportamiento.
- [ ] Empeora la salud del código, aunque funcione.

Si nada de eso aplica y el cambio mejora el código aunque no sea perfecto, aprueba.

AJUSTES

Una persona en las rutas sensibles

```
# .github/CODEOWNERS
/lib/payments/           @tu-usuario
/db/migrations/          @tu-usuario
/middleware.ts           @tu-usuario
/.github/revision-ia.md  @tu-usuario
/.github/CODEOWNERS      @tu-usuario
```

En GitHub, activa «Require review from Code Owners» en la regla de protección de la rama principal. El dueño necesita permiso de escritura. La `/` inicial ancla a la raíz y la `/` final cubre todo lo de dentro; la última línea que coincide gana.

> **CUIDADO**
> Un comentario de la IA nunca es una aprobación. Si trabajas solo, haz el nivel 2 en otro momento, con esta lista y no con la de la IA.