

Los 10 puntos que vuelven vendible tu app hecha con IA

CATEGORÍA · CÓDIGO Y DESARROLLO

Hiciste una app con IA y se ve increíble en tu pantalla, pero sigue siendo un demo: la primera persona que pague le va a encontrar las grietas.

QUÉ VAS A ENCONTRAR

- 01 Qué es y qué resuelve
- 02 Para qué sirve
- 03 Dónde aplicarlo
- 04 Cómo implementarlo paso a paso
- 05 Comprobación y errores frecuentes

FICHA DEL RECURSO

Qué es, para qué sirve y cómo se implementa

FUNCIÓN

Hiciste una app con IA y se ve increíble en tu pantalla, pero sigue siendo un demo: la primera persona que pague le va a encontrar las grietas.

PARA QUÉ SIRVE

Hiciste una app con IA y se ve increíble en tu pantalla, pero sigue siendo un demo: la primera persona que pague le va a encontrar las grietas. Lo que separa un juguete que se cae en días de un producto que la gente paga son 10 capas que casi nadie te enseña: front-end comprimido sin filtrar tus llaves, base de datos con RLS para que nadie toque datos ajenos, control de versiones, APIs de verdad, hosting, seguridad, rate limiting, caché, escalabilidad y monitoreo.

DÓNDE APLICARLO

Nivel Avanzado. Temas: profesionalizar-app, vibecoding.

CÓMO IMPLEMENTARLO

- 1 Confirma requisitos: plan de Claude, herramientas y accesos que la tarea necesita.
- 2 Prepara el material (archivos, cuentas, datos) antes de empezar.
- 3 Ejecuta el método sobre un caso real pequeño, no sobre el proyecto completo.
- 4 Verifica el resultado contra lo que esperabas antes de repetirlo o automatizarlo.
- 5 Cuando funcione dos veces seguidas, conviértelo en rutina o skill.

EL RECURSO EN DETALLE

Qué resuelve y cómo funciona

Hiciste una app con IA y se ve increíble en tu pantalla, pero sigue siendo un demo: la primera persona que pague le va a encontrar las grietas.

Lo que separa un juguete que se cae en días de un producto que la gente paga son 10 capas que casi nadie te enseña: front-end comprimido sin filtrar tus llaves, base de datos con RLS para que nadie toque datos ajenos, control de versiones, APIs de verdad, hosting, seguridad, rate limiting, caché, escalabilidad y monitoreo.

Aquí te explicamos cada una desde cero, aunque no programes, y te damos el prompt exacto para que Claude Code audite e implemente capa por capa, más una auditoría maestra que revisa las 10 de un jalón.

ANTES DE DARLO POR HECHO

Comprobación y errores frecuentes

- 1 Aplica el método sobre un caso pequeño y compara el resultado con lo que esperabas.
- 2 Si el resultado no sale, revisa primero los requisitos: plan, versión del modelo y accesos.
- 3 Anota lo que tuviste que corregir a mano: eso es lo que hay que añadir a tus instrucciones permanentes.
- 4 Cuando funcione dos veces seguidas, automatízalo como rutina o skill.

DATOS DEL RECURSO

Tipo: Guía paso a paso

Categoría: Código y desarrollo

Nivel: Avanzado

Temas: profesionalizar-app, vibecoding

El ecosistema de IA cambia rápido: confirma en el repositorio que el proyecto sigue activo y que los comandos no han cambiado desde la fecha de esta ficha.

SIGUIENTE PASO

Instálalo hoy, en pequeño.

Elige la ruta 1, pégale el enlace a Claude Code y pruébalo en un caso real chico. Un recurso instalado y probado vale más que diez guardados para después.

MÁS GUÍAS GRATIS EN [377CREATIVE.COM/RECURSOS](https://377creative.com/recursos)