Refactorización inteligente

Buenas prácticas para mejorar el código, sin romperlo

PABLO DEL ÁLAMO

Introducción

¿Tu código se ve caótico?

> La refactorización puede ser la solución, pero si no lo haces bien, puede causar más daño que beneficio.
>
> Aquí te dejo buenas prácticas para refactorizar de forma inteligente y segura.

PABLO DEL ÁLAMO

No refactorices sin un motivo claro

Hazlo solo cuando sea necesario.

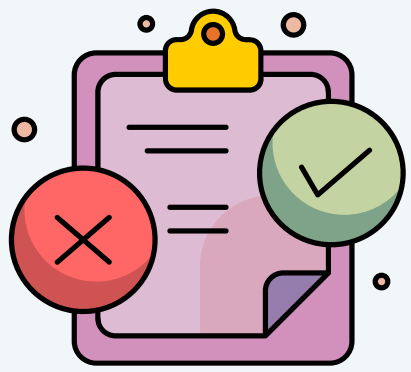
- Casos ideales:
  - Código duplicado.
  - Funciones demasiado largas.
  - Complejidad innecesaria.
  - Código que nadie entiende.

PABLO DEL ÁLAMO

Escribe tests antes de refactorizar

- Si el código no tiene tests, añade algunos básicos primero.

- Los tests actúan como método de seguridad para validar que no rompes nada durante la refactorización.



PABLO DEL ÁLAMO

Cambia de forma incremental

- No refactorices todo de golpe.
- Aborda pequeñas secciones del código en cada paso.
- Ejemplo: En lugar de reescribir una clase entera, refactoriza una función cada vez.

PABLO DEL ÁLAMO

Conserva la funcionalidad

La refactorización no es el momento de añadir nuevas características.

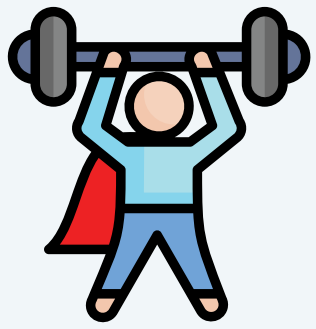
Tu objetivo es mejorar el diseño, no cambiar el comportamiento.

PABLO DEL ÁLAMO

Usa herramientas para refactorizar

Aprovecha IDEs modernos y herramientas como:

- IntelliJ IDEA, VS Code.
- Plugins para detectar código duplicado o mal formateado.
- Linters para identificar problemas.



PABLO DEL ÁLAMO

Mejora la legibilidad del código

Apunta a un diseño limpio y entendible:

- Cambia nombres genéricos por descriptivos. Ejemplo: calcX() → calculaTotalImpuestos().

- Divide funciones largas en otras más pequeñas.

PABLO DEL ÁLAMO

Reduce la deuda técnica

Ataca el código que más problemas causa.
Enfócate en partes del sistema que:

- Sean difíciles de entender.
- Requieran cambios frecuentes.

PABLO DEL ÁLAMO

Refactoriza solo lo que entiendes

No toques lo que no comprendas bien.

Probablemente haya alguien que ya haya trabajado con ello, a quien puedas acudir para solicitar ayuda.

PABLO DEL ÁLAMO

Refactoriza con frecuencia, pero con cuidado

- Haz de la refactorización un hábito, no una tarea gigantesca.

- Planifica pequeñas sesiones frecuentes, para mantener el código limpio.

PABLO DEL ÁLAMO

Conclusión

Refactorizar no se trata solo de limpiar el código, sino de invertir en la calidad y el futuro de tu software.

Aplicando estas prácticas, podrás mejorar la legibilidad, reducir errores y mantener tu sistema flexible y escalable.

Hazlo con cuidado, escribe tests y aborda los cambios de forma incremental.

Refactorizar no es un lujo, es una necesidad para cualquier desarrollador que aspire a la excelencia.

PABLO DEL ÁLAMO

¿Te ha resultado útil?

- Comparte esta guía con tu equipo o amigos desarrolladores.
- Guárdala para tenerla siempre a mano.
- ¡Dale un like o comenta si tienes preguntas!

PABLO DEL ÁLAMO