

BUENAS PRACTICAS PARA REALIZAR COMMITs

Commit Messages

Estructura del Mensaje

El mensaje de un commit consistirá en 2 diferentes partes separadas por una línea en blanco:

- Título: asunto
- Cuerpo

El título consiste en el tipo y asunto del mensaje.

Type / Tipo

El tipo es contenido en el título y puede ser de alguno de los siguientes casos:

feat: Una nueva característica.

fix: Se soluciono un bug.

docs: Se realizaron cambios en la documentación.

style: Se aplicó formato, comas y puntos faltantes, etc; **Sin cambios en el código.**

refactor: Refactorización del código en producción.

test: Se añadieron pruebas, refactorización de pruebas; **Sin cambios en el código.**

chore: Actualización de tareas de build, configuración del admin. de paquetes; **Sin cambios en el código.**

Subject / Asunto

El asunto no debe contener más de 50 caracteres, debe iniciar con una letra mayúscula y no terminar con un punto. Debemos ser objetivos al momento de redactar nuestro commit.

Body / Cuerpo

No todos los commits son lo suficientemente complejos como para necesitar de un cuerpo, sin embargo es opcional y se usan en caso de que el commit requiera una explicación y contexto. Utilizamos el cuerpo para explicar el ¿Qué y Por qué? de un commit y no el ¿Cómo? Al escribir el cuerpo, requerimos de una línea en blanco entre el título y el cuerpo, además debemos limitar la longitud de cada línea a no más de 72 caracteres.

Algo para tener en cuenta.

Arreglar un commit con amend

Llevas un rato trabajando, has cambiado unos cuantos archivos. Terminas de desarrollar un feature o resolver un bug. Haces el commit con los archivos modificados y, de repente, te das cuenta de que te has dejado algo. Ya sea un test unitario, algo de documentación, CSS o cualquier otra cosa.

Con suerte, no afecta mucho, con algo menos de suerte rompes todo el build con ese commit incompleto. No pasa nada, haces otro commit justo después y todo arreglado.

Sin embargo, es una mala filosofía tener commits incoherentes, incompletos o incluso que rompan el build, ya que si alguien elige retroceder a algún estado anterior del proyecto, no debería encontrarse con un commit que haga que las cosas no funcionen como deberían.

Si en vez de hacer un nuevo commit, ejecutas:

git commit -amend -no-edit

Actualizas el último commit que hayas hecho con los cambios que hayas añadido a Git. Así lo dejas todo empaquetado en un solo commit y mantienes el Git limpio.