

# Homogeneización

## Resumen

---

Para hacer un git commit, al dejar un mensaje descriptivo también podemos utilizar palabras reservadas en el equipo. Como se explicó en clase nos debemos de poner de acuerdo para hacer los commits y los cambios. Asimismo, que sean mensajes claros, por lo cual hice este documento para poder explicarles un poco de la forma en la que yo, Isaac, hago mis commits. Pueden seguirla ustedes, o no, pero es para hacer un ejemplo

## Forma general

---

La forma reservada que utilice tiene la estructura siguiente :

**palabraReservada** [ **documentoCambiado.extensión** ] : “ **mensaje descriptivo.** ”

La razón de hacer esto es eficientar el tiempo que tomamos al leer los cambios. Cuando leemos, muchas veces lo hacemos de una manera muy rápida y al identificar qué acción se hizo y en qué archivo puede ser más rápido para seguir el flujo. Por ejemplo, si nos estamos enfocando a un solo archivo, no nos interesarían otros archivos, por lo tanto solo veríamos si hubo cambios en lo que estamos haciendo

## Palabras reservadas

---

- **feat** : indica que se añadieron cosas, por ejemplo, añadir un archivo, añadir una función, añadir una sección de la página web.
- **fix** : arreglar un error o funcionalidad parcial o totalmente. Por ejemplo, arreglar un título, o arreglar una falta de ortografía.
- **docs** : esto solo es para la documentación, es cuando añadimos o modificamos el README.md u otro archivo que sea de documentación. Sin embargo, también lo podemos usar cuando añadimos ciertos comentarios que ayudan a entender el código.
- **refactor** : reestructurar o reescribir el código pero que no cambia la funcionalidad, solo la estructura.
- **style** : para los estilos o visuales.
- **perf** : es para mejorar el rendimiento, puede ser parecido a fix o refactor, pero en este caso sí cambia cosas, pero como tal no las arregla, solo son mejorar. Por ejemplo, si en un juego mejoro la *hitbox*.
- **del** : eliminación, puede ser de un archivo, sección, comentario, etc. Todas las eliminaciones.

## Nombres archivos y carpetas

---

En particular para el proyecto utilizaremos JavaScript, HTML y CSS, de cierta manera es un proyecto corto y pequeño, sin embargo, mi propuesta es que todos los archivos y carpetas se nombren de manera descriptiva.

Y antes de que me digan que mi Github no lleva esa organización es porque para mis proyectos personales es diferente, lol.