Donde gastar los merges

| **Comando** | **Características** | **Cuando usarlo** |
| --- | --- | --- |
| git rebase rama | Pone los commits en la nueva rama siempre después de los que ya hay, es decir que los reordena | La rama origin/develop en nuestra develop La rama origin/release en nuestra release La rama origin/master en nuestra master |
| git merge --squash rama | Junta todos los “microcommits” de la rama en uno solo en la rama destino | De una rama feature a develop |
| git merge --ff-only rama | Nunca crea nuevos commits en la rama destino pero falla si es necesario crear un nuevo commit, cosa que nunca debería pasar, por lo que --ff-only ayuda a detectar si hay algún problema  SOLO HACE MERGE SI PUEDE HACER UN FAST FORWARD  SI VA MAL Y NO HACEMOS -FF HARÍA UN COMMIT EXTRA | De develop a Release De release a master |
| git merge rama | Es el merge normal que puede crear nuevos commits | De una rama develop a feature. Para resolver los conflictos desde nuestra rama feature y no *ensuciar* develop. |

Squash de la privada a develop

Todos los días hacer merges-develop normales

Rebase: fetch , git rebase origin/develop