

Resumen convenciones git.

Armin Hucke

- Al inicio de cada mensaje de commit se debe primero especificar el **tipo** de commit como prefijo.

- Feat: Commit que agrega una nueva funcionalidad al programa (versión acotada de “feature”).

Ejemplo:

feat: add 2FA through SMS

- Fix: Commit que arregla algun bug en el código.

Ejemplo:

fix: disallow username form from accepting null values

- Perf: Commit que se refiere a haber refactorizado una porción de código para mejorar rendimiento.

Ejemplo:

perf: dump memory after <action/time elapsed>

- Build: Commit que cambia aspectos como las dependencias o la herramienta de build.

Ejemplo:

Build: update dependencies

- Entre estos tipos también se encuentran: style, test, docs, ops, refactor y chore.

- El alcance (scope) del commit es una parte opcional del mensaje que especifica que parte especifica en la que afecta el commit.

Ejemplo:

```
feat(authentication): add 2FA through SMS
```

- El verbo que inicia la descripción del commit debe siempre estar en **presente imperativo**.

Ejemplo incorrecto:

```
feat: added 2FA through SMS
```

- Se espera que el mensaje del commit tenga como máximo 50 caracteres.

Commits simulados:

1.

```
git init  
echo "mini taller semana 2" >> README.md  
git add README.md  
git commit -m "chore: first commit"
```
2.

```
git add Main.java  
git commit -m "feat: create main java class"
```
3.

```
git add pom.xml  
git commit -m "build: add dependencies that will be needed"
```
4.

```
git commit -m "style: remove excessive whitespace"
```
5.

```
git commit -m "refactor: use recursion for fib sequence  
calculation"
```

