

## Principales convenciones git

### 1. Nombres de Ramas

- **main o master:** Ramas principales
- **develop:** Rama de desarrollo en proyectos
- **feature/:** Ramas para nuevas funcionalidades.
- **hotfix/:** Ramas para correcciones rápidas en producción.
- **release/:** Ramas para preparar una nueva versión.
- **bugfix/:** Ramas para corregir bugs.
- **chore/:** Ramas para tareas menores o mantenimiento.

### 2. Mensajes de Commit

- **Especificidad:** Deben ser claros, concisos y se debe indicar qué se hizo.
- **Convención de tipo:** Iniciar con el tipo de cambio, seguido de dos puntos y un breve resumen.
- **Formato:**
  - **Primera línea:** Breve resumen del cambio.
  - **Segunda línea:** Descripción más detallada.

### 3. Uso de git pull vs git fetch

- **git pull:** Trae los cambios remotos y fusiona automáticamente.
- **git fetch:** Trae los cambios remotos sin fusionarlos, permitiendo revisar antes de fusionar manualmente.

### 4. Convenciones de Merge

- **merge commit:** Para fusionar cambios desde una rama hacia otra, conservando el historial de commits.
- **Squash and merge:** Fusiona todos los commits en uno solo para mantener un historial más limpio.
- **Rebase:** Reorganiza los commits sobre la rama base para mantener un historial lineal.

## 5. Uso de Pull Requests o Merge Requests

- **Discusión y revisión:** Los cambios son revisados antes de realizar merge.
- **Requerir aprobaciones:** Configurar el repositorio para requerir aprobaciones antes de fusionar.
- **Comentarios en el código:** Dejar comentarios en los Pull Requests para mejorar la calidad del código.

## 6. Etiquetado (Tagging)

- **Versiones:** Utilizar etiquetas para marcar versiones de software.
- **Semver:** Seguir las convenciones de versionado semántico, como MAJOR.MINOR.PATCH.

## 7. Rebase interactivo (`git rebase -i`)

- **Limpieza de commits:** Permite reordenar, combinar o eliminar commits antes de fusionar.

## 8. Uso de `.gitignore`

- **Exclusión de archivos:** Especificar qué archivos o directorios no deben ser incluidos en el control de versiones, como archivos temporales, configuraciones locales o dependencias generadas.

## 9. Convenciones para Branch Protection

- **Protección de ramas:** Configurar el repositorio para proteger ramas importantes como `main` o `master`, requiriendo revisiones de código o que las pruebas pasen antes de poder fusionar.

## 10. Uso de GPG para Firmar Commits

- **Seguridad y autenticidad:** Firmar commits y etiquetas con GPG para asegurar que provienen de una fuente confiable.