

Principales convenciones git en el desarrollo de software

1. Nombrado de las Ramas

- Rama Principal (Main/Master): La rama principal que contiene el código de producción. Generalmente se llama main o master.
- Ramas de Característica: Las ramas que se ramifican para implementar nuevas funcionalidades. El nombre debe ser un concepto que describa la funcionalidad que está siendo agregada, por ejemplo, feature/login-page o feature/user-auth.
- Ramas de Solución de Errores: Las ramas que se bifurcan para solucionar problemas en el código. Un ejemplo de ello: bugfix/fix.
- Ramas de Lanzamiento: Se utilizan para implementar lanzamientos de la aplicación. Ejemplo: lanzamiento/v1.0.0.
- Ramas de Hotfix: Se utilizan para realizar correcciones críticas directamente en la rama principal. Ejemplo: hotfix/urgent-security-patch.

2. Mensajes de commit

- Claros e Informativos: Los mensajes deben ser claros e informativos sobre lo que se logró en el commit. Ejemplo: arreglar el error de validación del inicio de sesión, agregar autenticación de usuario.
- Tiempo Verbal: Por ejemplo: "Add", "Fix", "Remove". Imperativo presente.
- Longitud: Mantén la longitud del mensaje corta (idealmente, menos de 50 caracteres para la primera línea). Luego, una nueva línea en blanco y, opcionalmente, una descripción adicional.

3. Uso de commits Atómicos

- Un Cambio por Commit: Cada commit debe representar un único cambio atómico. Esto facilita la reversión en caso de necesidad.
- Frecuencia: Commits lo suficientemente seguido como para que no hayan muchos cambios acumulados en un commit.

4. Uso de Pull Requests o Solicitudes de Merge

- Revisión: Usa Pull Requests para que otros revisen y discutan los cambios antes de fusionarlos en la rama principal.
- Comentarios: Asegúrate de que la descripción de la Pull Request explique claramente los cambios.
- Integración Continua: Idealmente, cada Pull Request debe ser validado automáticamente antes de ser aceptado.

5. Rebase vs Merge

- Rebase: Se utiliza para mantener un historial de confirmaciones limpio y lineal, por ejemplo, cuando queremos fusionar una rama de funcionalidad con la rama principal que se ha actualizado en el ínterin.
- Merge: Se usa principalmente para fusionar ramas después de que han sido revisadas en Solicitudes de Extracción. Puede romper el historial lineal, pero mantiene un registro completo de todas las operaciones.

Simulación de commits

1.

```
Feat: Add dark-mode
```

2.

```
Fix: Remove wrong color
```



3.

```
Feat: Add search system
```

```
Added a search system to upgrade user experience
```

4.

Refactor: Refactor app.js

Refactores code for better performance

5.

Fix: Remove innecessary files

Files with useless code were removed

Estos “commits” fueron hechos en obsidian. Utilizando buenas prácticas y en inglés.