

Convenciones de Nombrado de Ramas

[←Regresar](#)

Este documento establece las reglas para el nombrado de ramas en nuestro flujo de trabajo de Git, basado en GitFlow, con el fin de mantener un desarrollo organizado y estructurado.

Ramas Principales

master

- Contiene el código en producción.
- Solo se fusionan versiones estables y probadas.
- No se realizan desarrollos directos en esta rama.

qa

- Contiene código en pruebas de calidad.
- Permite realizar pruebas automatizadas y manuales.
- Contiene el código en desarrollo.
- Se fusionan las ramas de características (features), correcciones de errores (bugfix) y mejoras (enhancements).
- Es la base para la integración y pruebas antes de pasar a producción.

develop

- Rama de integración: Recibe los cambios de feature/, fix/ y enhancement/ antes de una versión estable.
- Siempre funcional: Debe mantenerse estable y en estado operativo, evitando código roto.
- No se usa en producción: Sirve para pruebas internas antes de pasar a release o main.
- Flujo controlado de fusión: Se integran cambios mediante pull requests con revisión.

Ramas Auxiliares

feature/{nombre-descriptivo}

- Se crean a partir de develop.
- Se utilizan para desarrollar nuevas características o funcionalidades.
- Se deben fusionar de vuelta en develop una vez completadas.
- Ejemplo: feature/agregar-autenticacion

bugfix/{nombre-descriptivo}

- Se crean a partir de develop.
- Se usan para corregir errores detectados en el entorno de desarrollo.
- Se fusionan en develop tras su corrección.
- Ejemplo: bugfix/corregir-error-login

release/{version}

- Se crean a partir de develop cuando se prepara una nueva versión para producción.

- Se usan para realizar pruebas finales, documentación y correcciones menores.
- Se fusionan en main y develop al finalizar.
- Ejemplo: release/1.2.0

hotfix/{nombre-descriptivo}

- Se crean a partir de main para corregir errores críticos en producción.
- Se fusionan en main y develop después de su corrección.
- Ejemplo: hotfix/solucionar-error-pago

enhancement/{nombre-descriptivo}

- Se crean a partir de develop.
- Se utilizan para mejoras en el código o optimización de funcionalidades existentes.
- Se fusionan en develop.
- Ejemplo: enhancement/mejorar-ui-dashboard

chore/{nombre-descriptivo}

- Se usan para tareas de mantenimiento del proyecto, como actualizaciones de dependencias o cambios en la configuración.
- Se crean a partir de develop y se fusionan de vuelta en develop.
- Ejemplo: chore/actualizar-librerias

Consideraciones Generales

- Usar nombres descriptivos y en minúsculas.
- Separar palabras con guiones -.
- Mantener las ramas actualizadas con develop antes de fusionarlas.
- Eliminar las ramas después de fusionarlas para evitar desorden.

Formato recomendado para mensajes de commit

- feat □ Nueva funcionalidad o característica
- fix □ Corrección de un bug
- refactor □ Refactorización sin cambios en funcionalidad
- perf □ Mejora de rendimiento
- test □ Agregar o modificar pruebas
- docs □ Cambios en la documentación
- style □ Cambios de formato, espacios, comas (sin afectar código)
- chore □ Tareas de mantenimiento (ej: actualizar dependencias)
- ci □ Configuración de integración continua (CI/CD)
- build □ Cambios en el sistema de build o dependencias
- revert □ Revertir un commit anterior

Ejemplo

```
git commit -m "feat(user): agregar opción de recuperación de contraseña"
```

[←Regresar](#)

From:
<http://wiki.adacsc.co/> - Wiki

Permanent link:
<http://wiki.adacsc.co/doku.php?id=ada:howto:sicoferp:factory:new-migracion-sicoferp:front:nombrado-de-ramas-git>

Last update: **2025/03/13 12:40**

