

Gestión de Ramas en Git

Ramas Principales

- **main o master:** Contiene el código listo para producción.
- **develop:** Utilizada para integrar cambios antes de fusionarlos con la rama principal.

Ramas Secundarias

- **Ramas de características:** Creación de nuevas funcionalidades. Se originan desde `develop` y se reintegran a `develop` al completarse.
- **Ramas de corrección de errores:** Para solucionar bugs. Se crean desde `develop` o `main/master`.
- **Ramas de hotfix:** Para arreglos críticos en producción. Se crean desde `main/master` y se fusionan de vuelta a `main/master` y `develop`.

Mensajes de Commit

Formato y Convenciones

- **Formato:** Mensajes claros y descriptivos, con una breve descripción.
- **Tipos de commits:**
 - `feat`: Nueva funcionalidad.
 - `fix`: Corrección de errores.
 - `docs`: Cambios en la documentación.
 - `style`: Formato y estilos.
 - `refactor`: Cambios en el código sin afectar la funcionalidad.
 - `test`: Añadir o corregir pruebas.
 - `chore`: Cambios en la configuración o entorno de desarrollo.
- **Detalles adicionales:** Usar el cuerpo del mensaje para más detalles y el pie de página para referencias a issues o tickets.

Nombres de Ramas

Convenciones

- Usar nombres descriptivos que indiquen el propósito de la rama, como:
 - `feature/nueva-funcionalidad`
 - `bugfix/corregir-error-x`
 - `hotfix/corregir-urgente`
 - `release/v1.2.3`

Revisiones de Código y Pull Requests

Uso de Pull Requests (PR)

- Utilizar PRs para la revisión de código antes de fusionar cambios.
- Cada PR debe incluir una descripción clara del propósito de los cambios y referencias a issues o tickets relacionados.