

Advanced commands (Pull Request, Fork, Rebase)

Pull Request (PR):

Un Pull Request es una propuesta de cambio que se envía a un repositorio en GitHub (u otras plataformas similares) para su revisión e inclusión en el repositorio fuente. Es una forma de proponer cambios y discutirlos antes de fusionarlos con el código principal.

Flujo de trabajo típico:

1. Fork: `git clone https://github.com/repo/original.git`
 - Clonar el repositorio original.
2. Rama: `git checkout -b mi-rama`
 - Crear y cambiar a una nueva rama para los cambios.
3. Commit: `git add . + git commit -m "Mis cambios"`
 - Agregar y confirmar los cambios en la rama.
4. Pull Request:
 - `hub pull-request -b repo:rama-destino -h usuario:mi-rama`
 - Crear un Pull Request desde la línea de comandos.
 - Desde la interfaz de GitHub, se crea un Pull Request seleccionando la rama de origen (mi-rama) y destino (rama-destino).
5. Iniciar el proceso de revisión de cambios:
 - Una vez creado el Pull Request, los colaboradores inician el proceso de revisión examinando los cambios propuestos y realizando comentarios.

Revisión:

Los colaboradores revisan los cambios, discuten y sugieren modificaciones si es necesario.

Flujo de trabajo típico:

1. Revisión de cambios:
 - Los colaboradores revisan los cambios y pueden comentar directamente en la interfaz de GitHub.
2. Sugerir modificaciones:
 - Los colaboradores pueden sugerir modificaciones o mejoras a través de comentarios en el código.
 - Aprobar cambios:
 - Si los cambios son aprobados, el mantenedor del proyecto o colaborador con los permisos necesarios puede aprobar el Pull Request.
3. Merge:
 - `git checkout rama-principal + git pull origin rama-principal + git merge mi-rama`
 - Fusionar la rama del usuario con la rama principal desde la línea de comandos.

Si los cambios son aprobados, se procede a fusionar la rama del usuario con la rama principal. Esto se realiza desde la interfaz de GitHub o mediante comandos de Git, como el ejemplo proporcionado.

Fork:

Hacer un fork es el proceso de crear una copia personal de un repositorio ajeno. Esto permite hacer cambios sin afectar el proyecto original.

Casos típicos de uso:

- Cuando se desea contribuir a un proyecto, pero no se tiene acceso de escritura.
- Si se desea experimentar con cambios sin afectar el proyecto original.

Flujo de trabajo típico:

1. Hacer un fork del repositorio original:
 - `git fork https://github.com/repo/original.git`
 - Realizar un fork del repositorio original para crear una copia personal en la cuenta del usuario.
2. Clonar el repositorio bifurcado:
 - `git clone https://github.com/usuario/mi-fork.git`
 - Clonar el repositorio bifurcado en la máquina local.
3. Crear y cambiar a una nueva rama para los cambios:
 - `git checkout -b mi-rama`
 - Crear una nueva rama específica para los cambios y cambiar a ella.
4. Agregar y confirmar los cambios en la rama:
 - `git add . + git commit -m "Mis cambios"`
 - Agregar y confirmar los cambios realizados en la rama.
5. Subir la rama con los cambios al repositorio remoto:
 - `git push origin mi-rama`
 - Subir la rama con los cambios al repositorio remoto en GitHub.
6. Crear el Pull Request en GitHub seleccionando la rama de origen y destino:
 - Desde la interfaz de GitHub, se crea un Pull Request seleccionando la rama de origen (mi-rama) y destino.

Rebase:

Rebase es un comando de Git utilizado para integrar cambios de una rama en otra. Se utiliza a menudo para mantener un historial de proyecto lineal y evitar commits de fusión innecesarios.

Flujo de trabajo típico:

1. Obtener los últimos cambios del repositorio remoto:
 - `git fetch origin`
 - Obtener los últimos cambios del repositorio remoto utilizando `git fetch`.
2. Aplicar los cambios locales sobre los cambios obtenidos:
 - `git rebase origin/rama-remota`
 - Utilizar `git rebase` para aplicar los cambios locales sobre los obtenidos.
3. Resolver conflictos:
 - Si hay conflictos, pausar el proceso para resolverlos, utilizando herramientas como `git mergetool` o resolviéndolos manualmente.
4. Continuar el rebase después de resolver conflictos:
 - `git add . + git rebase --continue`
 - Después de resolver conflictos, continuar el proceso de rebase.
5. Subir los cambios rebasados al repositorio remoto:
 - `git push origin mi-rama`

- Subir la rama con los cambios rebasados al repositorio remoto en GitHub.