

# Git & GitHub pt. 2

**Autor:** José Guillermo Sandoval Huerta

**Fecha:** 31 octubre 2023

## Índice

Comandos avanzados.....	2
Pull Request.....	2
Fork .....	3
Rebase.....	4

## Comandos avanzados

Los comandos avanzados de Git son herramientas que permiten a los usuarios de Git realizar tareas más complejas y personalizadas en su flujo de trabajo de control de versiones. Estos comandos son útiles para tener mayor flexibilidad y control sobre el repositorio. A continuación, se presentan algunos de estos comandos.

### Pull Request

También conocido como PR es una característica fundamental en sistemas de control de versiones como Git, sirve para facilitar la colaboración y la revisión de código en proyectos de desarrollo de software [1].

Realizar un Pull Request es un proceso que va desde (1) hacer Fork a un repositorio, (2) clonar el repositorio, (3) crear una rama nueva, (4) realizar cambios, (5) confirmar y subir los cambios, (6) Crear el pull request, donde en el repositorio original se encontrara una opción para crea un nuevo Pull Request. Se tendrá que hacer clic en el botón "New Pull Request", se verificara los cambio y se da clic en el botón "Create Pull Request". Por ultimo (7) el colaborador inicial tendrá que aprobar el pull request y luego fusionará el pull request en el repositorio principal.

Entre sus principales funcionalidades están:

- **Solicitar una fusión de cambios:** Un Pull Request es una solicitud formal que un colaborador hace para fusionar sus cambios desde una rama (branch) de trabajo en su repositorio con otra rama, generalmente la rama principal del proyecto, como master o main. Esto permite que otros revisen y consideren los cambios antes de que se fusionen en la rama principal.
- **Facilita la revisión de código:** Los Pull Requests proporcionan un espacio para que los colaboradores revisen y comenten los cambios propuestos. Esto fomenta la revisión de código, lo que ayuda a mejorar la calidad del código y a detectar posibles errores antes de que se fusionen en la rama principal.
- **Discusión y colaboración:** Los comentarios en un Pull Request permiten a los colaboradores discutir los cambios propuestos. Esto fomenta la colaboración, la resolución de problemas y la comunicación efectiva entre los miembros del equipo de desarrollo.
- **Seguimiento de cambios:** Los Pull Requests mantienen un registro de los cambios propuestos y de la discusión que los rodea. Esto es útil para entender por qué se realizaron ciertos cambios y para mantener un historial de decisiones.
- **Integración continua:** Muchas plataformas de control de versiones y hospedaje de repositorios, como GitHub y GitLab, pueden configurarse para ejecutar automáticamente pruebas y verificaciones de calidad en los cambios propuestos antes de la fusión. Esto asegura que los cambios no rompan el código existente.
- **Autorización y control de acceso:** Los Pull Requests a menudo permiten que los mantenedores del proyecto tengan un control más granular sobre quién tiene permiso para fusionar cambios en la rama principal. Esto ayuda a garantizar la calidad del código y la seguridad del proyecto.

- **Documentación y contexto:** Los Pull Requests pueden incluir descripciones detalladas que explican el propósito de los cambios, proporcionando contexto adicional para los revisores.

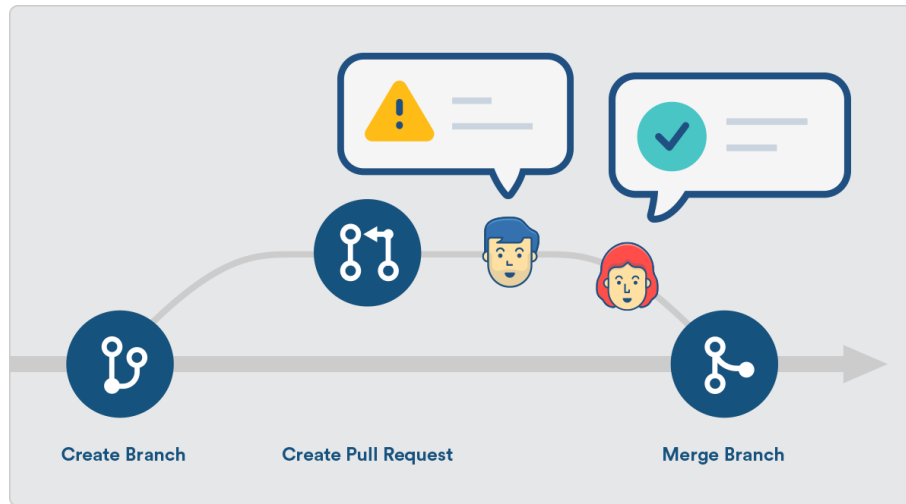


Imagen 1. Pull Request.

## Fork

Fork es una copia de un repositorio de Git existente al que se puede trabajar de manera independiente. Normalmente, se realiza un fork de un repositorio de un tercero o de un proyecto de código abierto para contribuir a él sin afectar directamente al repositorio original. Al hacer un fork, se crea una copia en la cuenta del usuario [2].

Para realizar un fork hay que realizar lo siguiente (1) ir al repositorio al que se quiere hacer fork, (2) en la parte superior derecha del repositorio, hacer click en el botón "Fork", (3) seleccionar la ubicación del fork, (4) una vez que se haya completado el fork, la página se direccionará a la nueva copia del repositorio.

Algunos aspectos importantes son los siguientes:

- **Propósito principal:** Los forks se utilizan comúnmente para colaborar en proyectos de código abierto. Cuando haces un fork de un proyecto de código abierto, puedes clonar la copia en tu máquina local, realizar cambios, y luego enviar un pull request al repositorio original para que los propietarios del proyecto consideren fusionar tus cambios.
- **Independencia:** El fork es completamente independiente del repositorio original. Esto significa que puedes hacer cambios en tu fork sin afectar al proyecto original y viceversa.
- **Clonación y sincronización:** Después de hacer un fork, puedes clonar tu copia en tu máquina local y trabajar en ella como en cualquier otro repositorio Git. Puedes sincronizar tu fork con el repositorio original de forma periódica para mantenerlo actualizado con los cambios realizados en el proyecto original.
- **Colaboración:** Cuando haces un fork de un proyecto y envías un pull request, estás proponiendo cambios al proyecto original. Los propietarios del proyecto original revisarán tus cambios y pueden decidir si los incorporan o no en su repositorio.

- **Diferencias con una rama:** A diferencia de crear una rama en un repositorio existente, un fork es una copia independiente del repositorio. Las ramas se utilizan para trabajar en cambios dentro del mismo repositorio, mientras que los forks se utilizan para crear una copia separada del repositorio original.

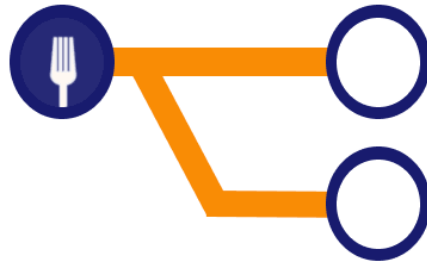


Imagen 2. Fork.

## Rebase

El comando **git rebase** se utiliza para modificar el historial de confirmaciones. Permite reorganizar, combinar o eliminar confirmaciones, lo que puede ayudar a mantener un historial de cambios más limpio y coherente. Sin embargo, se debe usar con cuidado, ya que puede causar conflictos si se utiliza en colaboración con otros [3].

El uso de git rebase es común para evitar la creación de fusiones (merges) innecesarias y mantener un historial de confirmaciones más limpio.

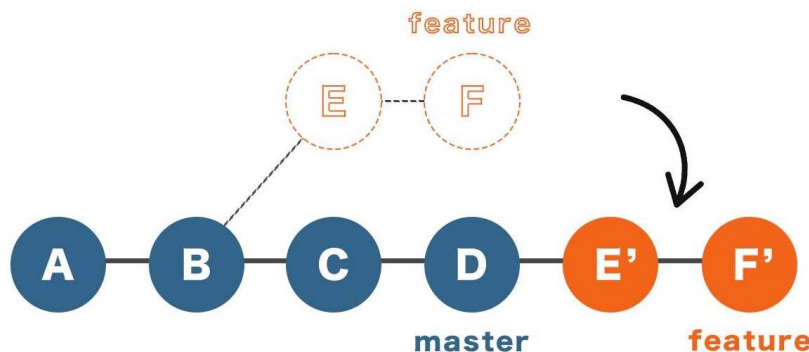


Imagen 3. Ejemplo de Rebase.

## Referencias

- [1] J. Domingo Muñoz, «¿Cómo colaborar en un proyecto de software libre? ¿Qué es un Pull Request?», josedomingo.org, 2023. [En línea]. Available: <https://www.josedomingo.org/pledin/2022/09/que-es-pull-requests/#:~:text=Podemos%20definir%20un%20Pull%20Request,acci%C3%B3n%20no%20tiene%20mucho%20sentido..> [Último acceso: 31 Octubre 2023].
- [2] K. T. «¿Qué es fork en Git?», keepcoding, 20 Julio 2022. [En línea]. Available: <https://keepcoding.io/blog/que-es-fork-en-git/>. [Último acceso: 31 Octubre 2023].
- [3] J. Holcombe, «Git Avanzado: Comandos Avanzados Además de los Básicos», Kinsta, 21 Agosto 2023. [En línea]. Available: <https://kinsta.com/es/blog/git-avanzado/>. [Último acceso: 31 Octubre 2023].