

GIT: Control de Versiones

¿Qué es una Pull Request (PR)?

Una Pull Request en GitHub es una solicitud para integrar cambios (commits) de una rama de desarrollo a otra, comúnmente de una rama de un fork o de una rama secundaria a la rama principal o main. Las PRs son fundamentales para la revisión de código, discusión sobre los cambios propuestos y una gestión de código fuente colaborativa y organizada.

Las PRs ofrecen múltiples beneficios como la revisión de código, la colaboración, un historial detallado de cambios y la integración con sistemas y otras herramientas de automatización.

¿Cuál es el proceso necesario para subir los cambios de una rama de trabajo?

Branching: Crear una nueva rama dentro del mismo.

Desarrollo: Realizar cambios en el código y hacer commits en la rama.

Actualización la rama de trabajo: Asegurarse de que la rama local esté actualizada con la rama principal.

Push al Repositorio Remoto: Subir los cambios a GitHub.

Creación de la Pull Request: Una vez que los cambios están listos, se crea una PR hacia el repositorio o rama objetivo.

Revisión y Discusión: Otros miembros del equipo revisan los cambios, proponen mejoras o discuten los aspectos del código.

Aprobación y Merge: Tras la aprobación, los cambios se integran (merge) en la rama principal o de desarrollo, según corresponda..

¿Qué se hace en el proceso de revisión?

Las Pull Requests son centrales para el proceso de revisión de código. Aquí es donde los miembros del equipo examinan los cambios propuestos y hacen comentarios o sugerencias. Una buena revisión de código debe ser minuciosa, centrada en mejorar tanto la calidad del código como la habilidad del desarrollador.

GitHub proporciona varias herramientas para facilitar la colaboración y revisión en PRs, incluyendo:

- **Comentarios en Línea:** Permiten a los revisores comentar directamente sobre líneas específicas del código.
- **Discusiones Generales:** Para comentarios más generales sobre la PR en su conjunto.
- **Solicitudes de Cambios:** Los revisores pueden solicitar cambios específicos antes de aprobar la PR.

¿Cómo se gestiona la integración de una PR?

Una vez que una PR ha sido aprobada, se puede "mergear" o integrar en la rama destino. Este proceso puede realizarse automáticamente a través de GitHub si no hay conflictos, o manualmente si se requiere una resolución de conflictos.

Los conflictos ocurren cuando hay discrepancias entre el código en la PR y el código en la rama destino. GitHub ofrece herramientas para resolver estos conflictos, pero a veces es necesario hacerlo manualmente en el entorno local.

¿Cuáles son los comandos para trabajar con el repositorio remoto?

Los comandos que nos ayudarán con esta tarea son:

git fetch: descarga los cambios del repositorio remoto, pero no los fusiona automáticamente en tu rama local. Este comando actualiza tu repositorio local con referencias de todas las ramas que están en el remoto, lo que te permite ver qué ramas han avanzado. Al usar git fetch, puedes revisar los cambios en el repositorio remoto antes de decidir integrarlos en tu rama de trabajo local. Es una forma de asegurarte de qué cambios se van a integrar antes de hacerlo efectivamente. Este comando es considerado un comando seguro porque no modifica tu estado de trabajo actual. Solo actualiza la información que Git tiene del repositorio remoto.

git merge <nombre-rama>: fusiona los cambios de una rama en otra. Cuando ejecutas `git merge <nombre-rama>`, Git intenta combinar automáticamente los cambios de la rama especificada en la rama en la que te encuentras actualmente. Si no hay conflictos, Git realizará la fusión automáticamente. Si hay conflictos, tendrás que resolverlos manualmente antes de completar la fusión. Este comando es fundamental en el flujo de trabajo de Git, ya que permite integrar el trabajo realizado en diferentes ramas y mantener actualizada la rama principal con todas las características y correcciones realizadas.

git pull: actualiza tu rama local con los cambios más recientes del repositorio remoto. Este comando es una combinación de `fetch` seguido de `merge`. Cuando ejecutas este comando, Git primero recupera las actualizaciones (commits) disponibles en la rama del repositorio remoto (como `origin/main/master`) y luego automáticamente intenta fusionarlos en tu rama local actual. Este comando es útil para mantener tu repositorio local sincronizado con los cambios que otros miembros del equipo han subido al repositorio remoto.