
Git & GitHub

Introducción a Git

Git es una tecnología imprescindible para el trabajo colaborativo eficiente. Con la ayuda de esta tecnología varias personas pueden trabajar en un mismo proyecto de forma simultánea sin perjudicarse entre sí. Además, nos permite llevar un control meticuloso de los cambios que realizamos en nuestros archivos.

Conceptos Básicos de Git

- Repositorio: Es el lugar en que se almacenan los datos actualizados e históricos de cambios.
- Revisión: Una revisión es una versión determinada de la información que se gestiona.
- Etiqueta (Tag): Los tags permiten identificar de forma fácil revisiones importantes en el proyecto.
- Rama (Branch): Un conjunto de archivos puede ser ramificado o bifurcado en un punto en el tiempo.
- Cambio (Change): Un cambio representa una modificación específica de un documento bajo el control de versiones.
- Desplegar (Checkout): Es crear una copia de un trabajo local desde el repositorio.
- Confirmar (Commit): Confirmar es escribir o mezclar los cambios realizados en la copia de trabajo del repositorio.
- Conflicto: Un conflicto se produce cuando diferentes partes realizan cambios en el mismo documento.
- Cabeza (Head): Se refiere a la última confirmación, ya sea en el tronco ("trunk") o en una rama ("branch").

Git maneja sus datos como un conjunto de copias instantáneas de un sistema de archivos miniatura. Cada que se confirma un cambio, Git toma una foto del aspecto de todos tus archivos en ese momento y guarda una referencia a esa copia instantánea.

Comandos básicos en Git

- `git init`: Crea un nuevo repositorio local Git.
- `git clone`: Se usa para copiar un repositorio.
- `git add`: Se usa para agregar archivos al área de preparación.
- `git commit`: Crea una instantánea de los cambios y la guarda en el directorio git.
Por ejemplo: `git commit -m "El mensaje que acompaña al commit va aquí"`
- `git config`: Puede ser usado para establecer una configuración específica de usuario, como el email, nombre de usuario y tipo de formato.
Por ejemplo: `git config --global user.email ejemplo@ejemplo.com`
- `git status`: Muestra la lista de los archivos que se han cambiado junto con los archivos que están por ser preparados o confirmados.
- `git branch`: Este comando es tu herramienta de administración de ramas de uso general.

- `git checkout`: Además de extraer las confirmaciones y las revisiones de archivos antiguos, `git checkout` también sirve para navegar por las ramas existentes.
- `git clean`: Elimina los archivos sin seguimiento de tu directorio de trabajo.
- `git fetch`: Con este comando, se descarga una rama de otro repositorio junto con todas sus confirmaciones y archivos asociados.
- `git commit --amend`: Pasar la marca `--amend` a `git commit` permite modificar la confirmación más reciente.

Trabajando con Repositorios en GitHub

Ramas (Branches)

Las ramas en GitHub son esencialmente una copia del proyecto. Son útiles para desarrollar características aisladas sin afectar el proyecto principal. Aquí están algunos comandos básicos:

- `git branch`: Muestra las ramas existentes.
- `git Branch [nombre_rama]`: Crea una nueva rama.
- `git checkout [nombre_rama]`: Cambia a una rama específica.

Fusiones (merge)

La fusión es el proceso de unir los cambios de una rama a otra. Esto se hace con el comando `git merge [nombre_rama]`, que fusiona la rama especificada.

Conflictos

Los conflictos ocurren cuando hay cambios contradictorios en el mismo archivo entre dos ramas. GitHub tiene un editor de conflictos que permite resolver conflictos de fusión simples. Para conflictos mas complejos, es posible que debas resolverlos localmente desde la línea de comandos.