



Introducción a git

Comenzaremos especificando algunos conceptos básicos que se utilizarán en el entorno git.

Git es un sistema de control de versiones distribuido cuyo objetivo es el de permitir mantener una gran cantidad de código a una gran cantidad de programadores eficientemente.

Repositorio: Base de datos fundamental para el diseño; en general elementos de software necesarios para la programación.

¿Que hay dentro del directorio git?

- **config**: Contiene las opciones de configuración de este repositorio. La configuración específica a este repositorio se define en este archivo . El uso más común de este archivo es especificar la ubicación de cada repositorio remoto y definir algunas opciones claves, como ser si tu repositorio posee o no un área de trabajo.
- **FETCH_HEAD**: Contiene SHAs-entre parentesis definir SHA de todas las ramas que fueron actualizadas por el último git fetch.
- **HEAD**: Representa el ref actual. En la mayoría de los casos éste es probablemente refs/heads/master.
- **index**: Es el área de staging conteniendo información acerca de los archivos que serán incluídos en el próximo commit.
- **hooks**: Es un directorio que rápidamente se convertirá en tu mejor amigo: contiene distintos scripts que son ejecutados en determinados momentos al trabajar con Git, como por ejemplo después de cada commit o antes de cada rebase. Una serie completa de artículos acerca de los hooks está en camino.
- **logs**: Contiene la historia de las distintas ramas. Aparentemente se lo utiliza principalmente con el comando reflog.
- **objects**: Es el depósito interno de objetos, indexados por SHAs.
- **rebase-apply**: Es la base para el comando rebase y también para git am. Si te atreves, es posible ajustar el archivo patch dentro de este directorio.
- **refs**: Es la copia maestra de todos los refs que existen en tu repositorio, ya sean stashes, tags, ramas locales o remotas.

Comandos básicos

➤ **git init**

Se crea una carpeta oculta llamada **.git** y contiene toda la información necesaria para que podamos realizar las versiones (copias de seguridad) de nuestro proyecto.



➤ **git clone /path/to/repository**

crea una copia local del repositorio.

Si utilizas un servidor remoto, ejecuta:

➤ **git clone username@host:/path/to/repository**

Tu repositorio local esta compuesto por tres "árboles" administrados por git:

- El primero es tu Directorio de trabajo que contiene los archivos.
- El segundo es el Index que actúa como una zona intermedia.
- El último es el HEAD que apunta al último commit realizado.

Puedes registrar cambios (añadirlos al **Index**) usando:

➤ **git add**

Este es el primer paso en el flujo de trabajo. Para hacer commit a estos cambios usa:

➤ **git commit -m "Commit message"**

Ahora el archivo está incluido en el HEAD, pero aún no en tu repositorio remoto.

Para enviar estos cambios a tu repositorio remoto ejecuta:

➤ **git push**

Para actualizar tu repositorio local al commit más nuevo, ejecuta:

➤ **git pull**