**Git**

¿Qué es Git?

“Git, desarrollado por Linus Torvalds para el desarrollo del kernel de Linux, es hoy en día el **sistema de control de versiones** más popular. Permite a los desarrolladores trabajar localmente y luego "empujar" sus cambios a un repositorio remoto cuando están listos. Por ejemplo:



El control de versiones es fundamental para cualquier proyecto de desarrollo de software o documentación que requiera mantenimiento o actualización continua.

Además es distribuido y descentralizado. Esto significa que cada desarrollador tiene su propio repositorio completo y puede trabajar de manera independiente sin depender de un servidor central.”

Conceptos importantes:

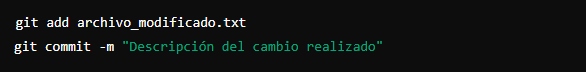
-*Repositorio*:

Un repositorio de Git es una colección de archivos de tu proyecto y un historial completo de todas las modificaciones realizadas a esos archivos. Cada repositorio contiene una carpeta oculta llamada .git donde se almacenan las configuraciones y el historial de cambios del repositorio.

-*Commit*:

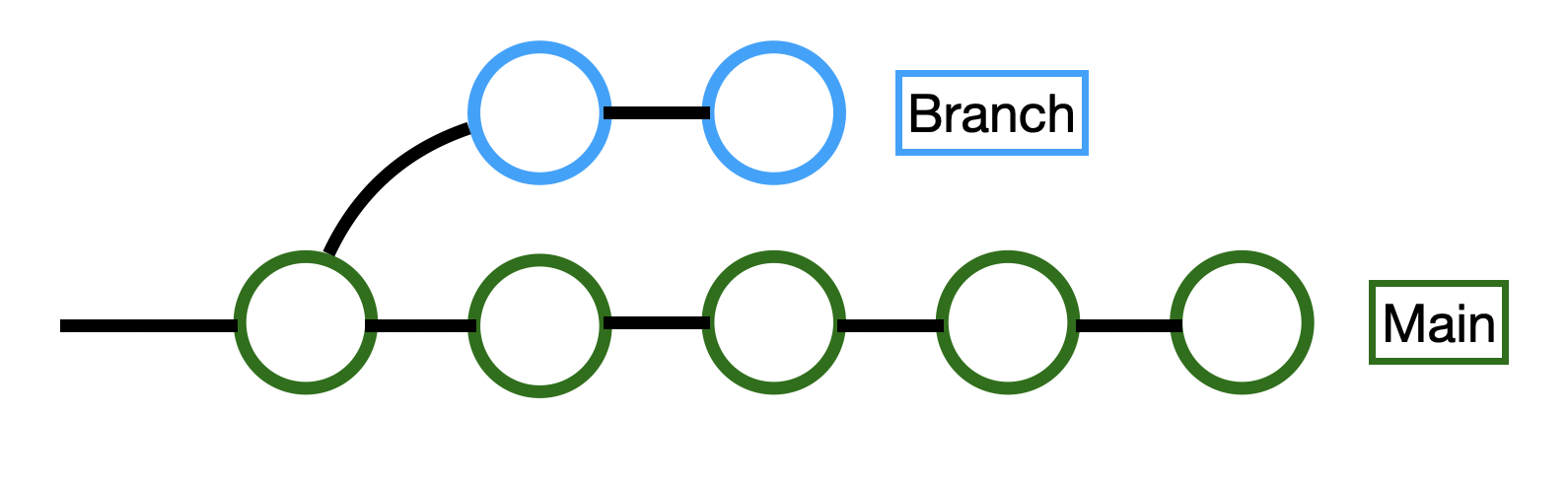
Un commit en Git es como una instantánea del estado actual de tu proyecto. Los commits almacenan información sobre qué cambios se han hecho, quién los ha hecho y cuándo.

Ejemplo: Para guardar un cambio en el repositorio, primero añades los archivos modificados al área de preparación (staging area) y luego realizas un commit:

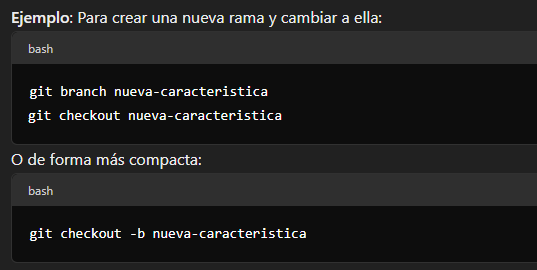


-*Ramas*:

Una rama es una versión separada del proyecto que puedes crear para trabajar en nuevas características o solucionar problemas sin afectar la versión principal. Las ramas permiten la experimentación y la colaboración en paralelo.

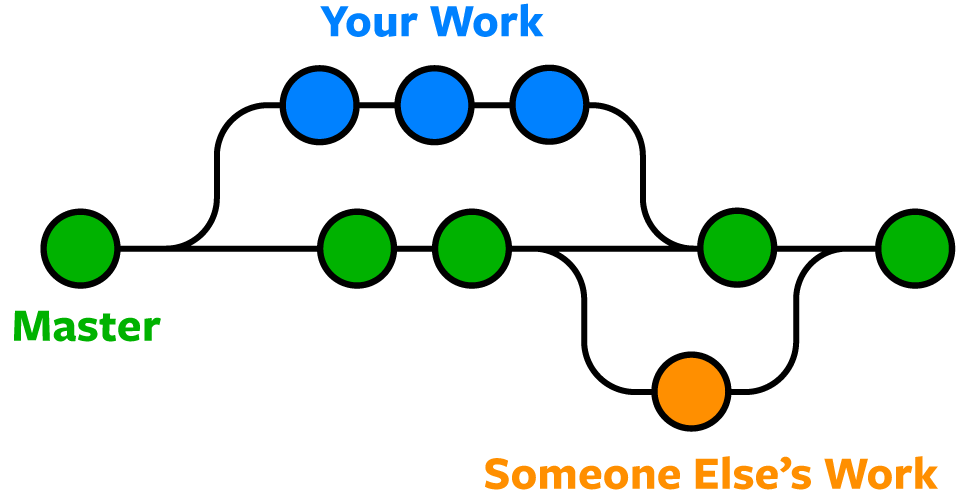


Las ramas en Git permiten a los desarrolladores trabajar en características, arreglos o experimentos separados del flujo principal de desarrollo (generalmente la rama master o main).



-*Fusión*:

La fusión es el proceso de combinar los cambios de una rama en otra. Cuando una nueva característica está lista, los cambios realizados en la rama de la característica se fusionan en la rama principal.



Comando para fusionar una rama de características de vuelta en la rama principal



-*Clon*:

Clonar un repositorio significa hacer una copia completa del repositorio en tu máquina local. Esto te permite trabajar en tu propio entorno y hacer cambios sin afectar el repositorio original.

Para clonar un repositorio, necesitas su URL, que puede ser obtenida de plataformas de hospedaje como GitHub, GitLab, o Bitbucket.



-*Pull request*

En plataformas como GitHub, una solicitud de extracción es una forma de proponer los cambios que has hecho en una rama y pedir que se integren en la rama principal.

Comandos:

**git init**

Para crear repositorio local.

**git clone**

Para copiar un repositorio.

**git add**

Agregar archivos para el commit.

Si querés agregar todos los cambios se usa “git add .” o agregar varios archivos separándolos por espacios.



**git commit -m**

Guardará el mensaje en el directorio

**git status**

Muestra el estado actual del repositorio

**git log**

Para ver una lista de todos los commits realizados, incluyendo sus identificadores, autores, fechas y mensajes.

git log --oneline: Muestra un historial más conciso, mostrando solo los mensajes de los commits en una sola línea.

git log --graph: Muestra el historial con una representación gráfica de las ramas y las relaciones entre los commits.

git log --author para filtrar los commits por autor:



**git push**

Para enviar confirmaciones locales a la rama maestra del repositorio remoto.

**git branch**

Para cambiar de una rama a otra.

**git branch -d**

Para borrar la rama

**git checkout**

Para cambiar entre ramas o versiones de un proyecto en Git. Puedes usarlo para moverte entre diferentes estados del repositorio, como ramas de desarrollo, commits anteriores y etiquetas.



**git checkout SHA**

Si deseas moverte a un commit anterior en la historia, primero debes conocer su identificador único (SHA). Luego, puedes usar git checkout con el identificador del commit:



**git checkout -b**

Crea ramas y te lleva a ella.

**git remote**

Te permite ver todos los repositorios remotos. El siguiente comando listará todas las conexiones junto con sus URLs:

**git pull**

Fusiona todos los cambios del repo remoto con el directorio local

**git merge**

Para fusionar ramas

**git diff --base**

Para hacer una lista de conflictos. Para poder ver conflictos con respecto al archivo base.

**git diff**

Para ver una lista de todos los conflictos presentes. Permite ver las diferencias entre la versión actual de tus archivos y la versión anterior.

Si has preparado cambios en la staging area usando git add, puedes usar git diff --staged para ver las diferencias entre la versión preparada y la última versión commit.