

GIT - Comandos básicos

Como crear un repositorio:

Agregar archivos y directorios a un repositorio

Respalidar archivos y directorios en repositorios.

git init Para comenzar nuestro proyecto. Crea 2 áreas una de ensayo y otra de repositorio

git add nombreArchivo Agrega el archivo al área de ensayo para hacer seguimiento, en general el archivo queda de color verde, antes sino tiene seguimiento esta de color rojo.

git add . Agrega todos los archivos de ese directorio para hacer seguimiento.

git commit -m "Comentario" Pasa los archivos del área de ensayo al repositorio.

git status -s Nos muestra el status de los archivos si estan agregados o no.

git log -- online Nos muestra todos los commit realizados en el repositorio.

git reset --hard (codigo commit) - Nos lleva al commit indicado por el codigo, pero perdemos el actuado actual del codigo. (hacer esto puede provocar perdida de código)

git commit -am "Comentario" - Nos hace un add. y commit en un solo paso.

git pull <https://github.com/JuanAntonaccio/PrimerRepositorioEjemploGit> main

Esto trae los cambios desde el repositorio a mi proyecto, puede ser main o master en donde este parado en mi repositorio.

El comando "git pull" se utiliza para traer los cambios del repositorio remoto a tu repositorio local. En este caso, al ejecutar:

git pull <https://github.com/JuanAntonaccio/PrimerRepositorioEjemploGit> main

Estás indicando a Git que obtenga los cambios del repositorio remoto ubicado en la URL "<https://github.com/JuanAntonaccio/PrimerRepositorioEjemploGit>" y de la rama "main" (rama principal) y los aplique en tu repositorio local. Es especialmente útil cuando trabajas en equipo y necesitas actualizar tu código con las últimas modificaciones hechas por otros colaboradores en el repositorio remoto.

git clone <https://github.com/JuanAntonaccio/PrimerRepositorioEjemploGit.git>

El comando "git clone" se utiliza para crear una copia local de un repositorio remoto en tu máquina.

Estás indicando a Git que clone el repositorio ubicado en la URL en un nuevo directorio en tu máquina con el mismo nombre del repositorio. Git copiará todos los archivos, ramas y la historia del repositorio remoto a tu máquina, permitiéndote trabajar en el proyecto de forma local.

Este comando es útil para comenzar a colaborar en un proyecto existente o para tener una copia local de un repositorio remoto.

git branch nombreRama

Estás indicando a Git que cree una nueva rama con el nombre "nombreRama" en tu repositorio local. Esta nueva rama estará basada en el commit actual en el que te encuentras. Las ramas en Git te permiten trabajar en diferentes líneas de desarrollo de manera independiente, lo que es útil para separar funcionalidades, experimentar con cambios sin afectar la rama principal, entre otros.

Después de ejecutar este comando, la nueva rama se creará, pero no cambiarás a esa rama automáticamente. Para cambiar a la nueva rama que acabas de crear, puedes usar el comando "git checkout nombreRama".

git checkout nombreRama

Estás indicando a Git que cambie tu directorio de trabajo a la rama llamada "nombreRama". Esto significa que empezarás a trabajar en la rama seleccionada, lo que te permitirá hacer cambios y realizar operaciones en esa rama en particular.

Es importante recordar que al cambiar de rama con "git checkout", tu directorio de trabajo se actualizará con el contenido y commits de esa rama.

Además de cambiar a una rama específica, el comando "git checkout" en Git tiene otras opciones útiles:

1. ***Crear y cambiarte a una nueva rama:*** Puedes crear una nueva rama y cambiarte a ella en un solo comando utilizando la opción "-b". Por ejemplo, para crear y cambiar a una nueva rama llamada "nuevaRama", ejecutarías: git checkout -b nuevaRama.
2. ***Recuperar un archivo eliminado:*** Si has eliminado un archivo y deseas recuperarlo del último commit, puedes utilizar git checkout -- nombreArchivo.
3. ***Desvincular cambios locales en un archivo:*** Si hiciste cambios locales en un archivo y deseas deshacerlos para que vuelva a su estado tal como está en el último commit, puedes usar git checkout -- nombreArchivo.

git merge nombreRama

Para hacer git merge tenemos que estar en la rama principal, y nos une la rama con la rama principal. Podemos tener conflictos.

El comando "git merge" se utiliza para combinar cambios de una rama en otra en Git.

Básicamente, fusiona el historial de una rama en otra para incorporar los cambios realizados en la rama que quieres fusionar.

Al ejecutar git merge nombreRama, por ejemplo, estás indicando a Git que traiga los cambios de la rama "nombreRama" a la rama en la que te encuentras actualmente. Git intentará fusionar los cambios automáticamente, pero en caso de conflictos (cuando se modifican las mismas líneas en ambas ramas), se generará un conflicto que deberá ser resuelto manualmente.

El comando "git merge" es fundamental para integrar el trabajo realizado en diferentes ramas y mantener actualizado tu código base.