

- GitHub es un sitio web que permite almacenar, compartir y colaborar en proyectos que usan Git como sistema de control de versiones.
- Para crear un nuevo repositorio debemos hacer click en “new repository” , le agregamos un nombre y elegimos si sera público o privado.
- Las ramas se crean por defecto al iniciar un nuevo repositorio, pero para crear nuevas ramas se utiliza el siguiente comando: `git branch nombre_rama`
- Para cambiar de rama se utiliza el siguiente comando: `git checkout nombre_rama`
- Para fusionar una rama con la principal se escribe: `git checkout main ; git merge nombre_rama`
- Para crear un commit se utiliza el comando: `git commit -m "Descripción breve del cambio"`
- Para enviar un commit a GitHub escribimos: `git push -u “nombre del repositorio” main`
- Un repositorio remoto es una versión del proyecto que esta guardado en un servidor de la nube como por ejemplo GitHub.
- Para agregar un repositorio remoto a Git se usa: `git remote add “nombre del repositorio” https://github.com/usuario/nombre-repo.git`
- Para hacer un “push” al repositorio remoto se hace con el comando: `git push -u “nombre del repositorio” main`
- Para realizar un “pull” del repositorio remoto usamos: `git pull “nombre del repositorio” main`
- Un fork es una copia de un repositorio que se crea en tu cuenta de GitHub.
- Para hacer un fork se deben realizar los siguientes pasos:
  - 1) Estando en el repositorio en GitHub hacemos click en el boton “fork”
  - 2) Copiamos el fork a la computadora con el siguiente codigo: `git clone https://github.com/tu-usuario/nombre-del-repo.git`  
`- cd nombre-del-repo`
  - 3) Realizamos los cambios necesarios y los subimos al fork: `git add .`  
`git commit -m "Descripción de los cambios"`  
`git push “nombre del repositorio” main`
- Un Pull Request es una solicitud para que los dueños de un repositorio revisen y fusionen los cambios que hicieron otras personas en su copia del proyecto.
- Luego de verificar los cambios y asegurarse de que no hayan conflictos en el código, para aceptar el pull request hacemos click en “merge pull request” y luego en “confirm merge”
- Una etiqueta (tag) en Git es un marcador o referencia que se usa para señalar puntos importantes en la historia de un repositorio, como versiones de software.
- Para crear una etiqueta simple se usa el comando: `git tag “nombre de la etiqueta”`, para hacer una etiqueta con mas información se escribe: `git tag -a “nombre de la etiqueta” -m "Descripción detallada"`
- Para enviar una etiqueta al repositorio remoto se usa: `git push “nombre del repositorio” “nombre de la etiqueta”`
- El historial de Git es el registro de todos los commits realizados en un repositorio. Este historial muestra quién hizo qué cambios y cuándo, permitiendo rastrear la evolución del proyecto y revertir errores si es necesario.

- Para ver el historial básico usamos: `git log`
- Se puede buscar en el historial de Git ya sea por autor, fecha, mensaje o cambios en archivos.
  - 1) Para buscar por mensaje: `git log --grep="palabra clave"`
  - 2) Para buscar por autor: `git log --author="nombre"`
  - 3) Para buscar dentro de un rango de fechas: `git log --since="2024-01-01" --until="2024-03-01"`
  - 4) Para buscar por cambios en un archivo específico: `git log -- nombre-del-archivo`
- En git no se puede simplemente borrar el historial , pero si se puede reescribirlo o eliminarlo del repositorio local y remoto.
  - 1) Para borrar el historial localmente usamos: `rm -rf .git`  
`git init`  
`git add .`  
`git commit -m "Reinicio del historial"`
  - 2) Para reescribir el historial y borrar commits antiguos: `git rebase -i HEAD~N` (reemplazar N por el número de commits a editar)
  - 3) Para borrar el historial en GitHub y subir uno nuevo: `rm -rf .git`  
`git init`  
`git add .`  
`git commit -m "Nuevo inicio del historial"`
  - 4) Para borrar un commit en específico: `git rebase -i <commit-anterior>`
- Un repositorio privado en GitHub es un repositorio oculto al público, accesible solo para los propietarios y colaboradores autorizados.
- Para hacer un repositorio privado seguimos estos pasos:
  - 1) En GitHub hacemos click en New Repository.
  - 2) Asignamos un nombre al repositorio.
  - 3) En la sección "Visibility", seleccionamos "Private".
  - 4) Hacemos click en "Create repository".
- Para invitar a alguien a unirse a nuestro repositorio:
  - 1) Vamos a la pestaña "Settings".
  - 2) En el menú lateral, seleccionamos "Manage access".
  - 3) Hacemos click en "Invite a collaborator".
  - 4) Ingresamos el nombre de usuario o correo de la persona y enviamos la invitación
- Un repositorio público en GitHub es un repositorio accesible para cualquier persona en Internet.
- Para hacer un repositorio público debemos seguir el mismo procedimiento que el que seguimos para hacer uno privado con la unica diferencia de que en la sección "visibility" seleccionamos "public".
- Para compartir el repositorio con la gente podemos compartir el link del repositorio, agregar colaboradores o permitir forks y pull requests para contribuciones externas.