

- **¿Qué es GitHub?**

Es una plataforma donde podemos guardar y gestionar repositorios de Git de manera remota

- **¿Cómo crear un repositorio en GitHub?**

Directamente desde su website: iniciando sesión y creando el repositorio donde dice “New repository”, deberás otorgarle un nombre, elegir si va a ser público o privado y tendrás algunas casillas opcionales.

- **¿Cómo crear una rama en Git?**

```
% git branch <branch_name>
```

- **¿Cómo cambiar a una rama en Git?**

```
% git checkout <branch_name>
```

- **¿Cómo fusionar ramas en Git?**

Cambiamos a la rama principal o rama que absorberá a la otra y hacemos:

```
% git merge <branch_feature>
```

- **¿Cómo crear un commit en Git?**

```
% git commit -m “Algún mensaje para saber de que se trata el commit”
```

- **¿Cómo enviar un commit a GitHub?**

```
% git push -u origin main
```

- **¿Qué es un repositorio remoto?**

Es un directorio que posee el “tracking” que hace Git, solo que está alojado en la nube

- **¿Cómo agregar un repositorio remoto a Git?**

```
% git remote add origin <url_del_repositorio>
```

- **¿Cómo empujar cambios a un repositorio remoto?**

```
% git push
```

- **¿Cómo tirar de cambios de un repositorio remoto?**

```
% git pull //actualiza automáticamente tu repo local con lo que tenes en el remoto
```

% git fetch //permite analizar las diferencias que tiene el repo local con el remoto

- **¿Qué es un fork de repositorio?**

Realizar una copia de un repositorio de otro usuario y así poder modificarlo sin afectar el original

- **¿Cómo crear un fork de un repositorio?**

Ir al repositorio en GitHub y apretar el botón “Fork”

Luego si quieres llevarlo a local

% git clone <repo>

- **¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?**

Es una forma de proponer cambios a un repositorio en GitHub

Primero debemos hacer el fork, luego creamos una branch en la cual trabajamos, una vez finalizado realizamos el commit al fork.

Dentro de nuestro fork vamos a donde dice “New pull request”, de esta manera solicitamos al creador del original, este lo revisará, propondrá cambios (en caso de ser necesario) y luego aceptará (o declinará) el pull request, de esta forma fusionando y adquiriendo las mejoras que realizaste

- **¿Cómo aceptar una solicitud de extracción?**

Se accede a la solicitud en la pestaña de “Pull requests” dentro del repositorio

Luego se examinan los cambios

Se proponen ajustes dejando comentarios

Se aprueba

En la fusión se resuelven en caso de haber conflictos, de forma manual

- **¿Qué es una etiqueta en Git?**

Es una forma de marcar un punto específico en el historial de un repositorio. Es como un marcador permanente que apunta a un commit particular. A diferencia de las ramas, las etiquetas no cambian con el tiempo y siempre apuntan al mismo commit.

Es una manera de establecer puntos de referencias, como por ejemplo la versión 2.0 de un programa.

- **¿Cómo crear una etiqueta en Git?**

% git tag -a <tag_name> -m <message> //etiqueta anotada

% git tag <tag_name> //etiqueta ligera

- **¿Cómo enviar una etiqueta a GitHub?**

```
% git tag // Ver todas las tags
% git show <tag_name> //Mostrar tag anotada

% git push origin <nombre_de_la_etiqueta> //Enviar tag al repo en GitHub
% git push origin --tags //Enviar todas las tags
```

- **¿Qué es un historial de Git?**

Es la secuencia de commits realizados, están ordenados de manera cronológica y poseen información sobre el autor, la fecha, el mensaje descriptivo y las modificaciones que se realizaron en los archivos

- **¿Cómo ver el historial de Git?**

Lista de commits, los muestra en sentido inverso cronológico, es el más básico

```
% git log
```

Flags: – oneline – graph – author=”name”, etc

- **¿Cómo buscar en el historial de Git?**

```
% git log -S”cadena”
% git log -G”Exp regular”
```

- **¿Cómo borrar el historial de Git?**

```
% git reset
```

Con este comando deshacemos commits y vuelve el puntero HEAD al anterior y tiene flags:

- soft
- mixed
- hard

```
% git rebase -i
```

permite editar el historial de Git de forma interactiva y puedes usarlo para eliminar, reordenar o combinar commits

- **¿Qué es un repositorio privado en GitHub?**

Es un repositorio al que solamente pueden acceder el propietario y los usuarios a los que este les haya concedido acceso

- **¿Cómo crear un repositorio privado en GitHub?**

Cuando creamos un nuevo repositorio en GitHub, nos da la opción de hacerlo público o privado

- **¿Cómo invitar a alguien a un repositorio privado en GitHub?**

En GitHub, entramos al repositorio privado, dentro vamos a:

Settings > Collaborators > Add people > Agregamos al usuario por username, email o name

Puedes darle permiso de: read, write o admin

GitHub limita la cantidad de personas que puedes agregar dentro de 24 horas. Si quieres agregar más debes considerar crear una organización

- **¿Qué es un repositorio público en GitHub?**

Es un repositorio a que cualquier persona tiene acceso para ver y clonar el código

- **¿Cómo crear un repositorio público en GitHub?**

Cuando creamos un nuevo repositorio en GitHub, nos da la opción de hacerlo público o privado

- **¿Cómo compartir un repositorio público en GitHub?**

Se puede compartir mediante la URL, a través de redes sociales desde GitHub, embed el código en alguna página web o directamente en la comunidad de GitHub