

Trabajo practico unidad 2

Genaro Luna

- 1) GitHub es una comunidad que funciona a suerte de “red social” de programadores. Al crearte una cuenta en GitHub podés crear repositorios donde subir tus códigos y proyectos y dejarlos públicos o privados en tu perfil, además que podés guardar o clonar otros repositorios de otras personas las cuales también podés seguir y ellos a vos.
- 2) Vamos a nuestro perfil creado en GitHub y vamos a la parte de nuevo repositorio, al clickear ahí tendremos que seleccionar un nombre para nuestro repo (que es opcional pero recomendado) y luego de crearlo copiar y pegar los comandos que nos aparecerán en la consola de git para subir nuestro proyecto al nuevo repositorio.
- 3) Una rama en git se crea con el comando `git branch nombre-rama`
- 4) Para cambiar a una rama en git se usa el comando `git checkout nombre-rama`
- 5) Primero nos posicionamos en el main y luego hacemos `git merge nombre-rama-a-fusionar` para fusionar las ramas
- 6) Para crear un commit en git utilizamos el comando `git commit nombre-commit`
- 7) Un commit se envía a GitHub con el comando `git push origin nombre-rama`
- 8) Un repositorio remoto es una versión de tu proyecto que se almacena en la nube en plataformas como GitHub o GitLab
- 9) Con el comando `git remote add origin url-del-repositorio-remoto` podés agregar un repositorio remoto a git, o con `git clone url-del-repositorio-remoto` también podés hacerlo descargando el repo
- 10) Con el comando `git push origin main` podés empujar los cambios a un repositorio remoto
- 11) Con `git fetch` o `git pull` podés tirar los cambios de un repositorio remoto
- 12) Un fork de un repositorio es una copia de un repositorio que se aloja en tu cuenta de GitHub
- 13) En GitHub, vas al repositorio al que quieres crearle un fork, y al ingresar en la parte superior derecha se encuentra el botón para crear el fork. Puedes cambiar el nombre del repositorio y su descripción
- 14) Hay que ir al repositorio en GitHub que se desee crear la pull request y apretamos el botón con el mismo nombre y le damos a crear nueva pull request, con eso mandamos nuestra solicitud
- 15) Vamos a nuestro repositorio en GitHub y apretamos el botón pull request, donde encontraremos la petición enviada y podremos aceptar
- 16) Las etiquetas o mejor dicho tags, son identificadores de puntos importantes y específicos cuando del historial de un repositorio.
- 17) Los tags se crean con el comando `git tag -a nombre -m descripción`

- 18) Se utiliza el comando `git push origin nombre-etiqueta`
- 19) El historial es el registro completo de los cambios del repositorio
- 20) Para ver el historial, se utiliza `git log`
- 21) Para poder buscar en el historial hay varias formas, una de ellas es con el comando `git log --author=nombre-autor`
- 22) Con el comando `git reset --hard HEAD~1`, se puede eliminar el ultimo commit del historial
- 23) Un repositorio privado en GitHub es como un repositorio común, solo que únicamente las personas a las cuales vos les des acceso a ese repositorio son las que podrían visualizarlo, de lo contrario es únicamente visible por vos
- 24) Para crear un repositorio privado en GitHub basta con crear un nuevo repositorio y en el apartado de privacidad poner privado
- 25) Para invitar a alguien a nuestro repositorio privado debemos configurarlo con la opción Collaborators and teams y luego en la opción add collaborator escribir el mail de la persona que desees que tenga acceso a tu repositorio
- 26) Un repositorio publico en GitHub es la contra parte de un repositorio privado, este repositorio puede verlo cualquier persona que ingrese a tu perfil de GitHub y lo vea
- 27) Un repositorio público se crea de la misma manera que uno privado, con la diferencia de que en el apartado de seguridad solo ponemos que sea publico
- 28) Para compartir un repositorio público en GitHub puede hacerse pasando el link del mismo o dando tu nombre de cuenta de GitHub para que la gente pueda acceder a el