

1) .

- a) GitHub es un controlador de versiones utilizado para registrar los cambios en un archivo o conjunto de archivos de modo que se puede recuperar versiones específicas más adelante
- b) Los beneficios son que puedes tener acceso a tu trabajo en caso de alguna eventualidad. Pérdida/robo/cambio de pc o si los archivos se corrompen
- c) Entrar a la página, te registras y le das a nuevo repositorio. Le pones un nombre y listo
- d) Son ramificaciones del código que te permite trabajar en él sin afectar los archivos principales. Una vez que se vea que la versión es estable se puede fusionar la rama con la rama principal
- e) Para crear una nueva rama se utiliza `git branch "nombre rama"`
- f) Con `git checkout "nombre de la rama"` puedes cambiar entre ramas
- g) Te posicionas en la rama a la que deseas que se fusionen las otras ramas y utilizas `git merge "nombre de la rama"`
- h) Hacer un commit se utiliza `git commit -m "Felipe"`
- i) Para enviar el commit se utiliza `git push origin main`
- j) Un repositorio remoto es el que se encuentra externo a nuestra computadora
- k) Para vincular a nuestro repositorio utilizamos `git remote add origin url`
- l) Utilizamos `git push origin main`
- m) Utilizamos `git clone` o `git pull`
- n) Un fork es una copia de un repositorio en un espacio diferente de tu cuenta
- o) Para crear un fork entras a la página de GitHub y en el repositorio habrá un botón que se llama "fork"
- p) Vas al repositorio en el que te interesa crear una bifurcación, le das click en solicitar incorporación de cambios
- q) En tu repositorio habrá una opción de solicitudes de incorporación, luego seleccionas la que estás interesado en aceptar y seleccionas aprobar
- r) Se utilizan principalmente para marcar las diferentes versiones de un proyecto
- s) Para crear un tag se utiliza el comando `git tag`
- t) Puedes utilizar `git push origin "nombre etiqueta"`
- u) Es un registro de todos los commits que hiciste
- v) Para buscar en un historial se utiliza `git log`
- w) Para borrar el historial utilizamos `git rebase`
- x) Es un repositorio en el cual nadie puede ver o ingresar
- y) Al momento de crear el repositorio tildas la opción de privado
- z) Haces click en configuraciones, acceso, "colaboraciones", y agregas el nombre de esa persona y lo envías
- aa) Se utiliza para compartir software de código abierto
- bb) Nuevo repositorio, le editas los parámetros y publicas
- cc) Copias el enlace y lo compartes con quien quieras
- dd) contiene todos los archivos de tu proyecto y el historial de revisiones de cada uno de ellos.
- ee) Bajas el repo con `git clone` y lo mantendrías actualizado con los compañeros usando `pull`