

Trabajo Practico Numero 2: Trabajo colaborativo

- 1) Git Hub es un sitio web que te permite subir tus proyectos de programación u otros archivos. De esta forma quedan guardados en internet, con el fin de que sean visualizados por otros usuarios, editados, o copiados. También ofrece visualizar proyectos de otros usuarios y editarlos (con permiso).
- 2) Para crear un repositorio en Git Hub, primero debes ingresar en tu cuenta y luego dirigirse a la esquina superior derecha. En esa esquina está el símbolo **+**, que es el acceso para crear el repositorio. Git Hub te dirige a otra página donde pones nombre a tu repositorio y lo configuras.
- 3) Para crear una rama en Git, abro mi editor de código con el archivo donde ya previamente inicialice un repositorio local (es decir en mi PC) y guarde con la función `add` .
En este momento solo hay una rama llamada `main`. Para crear otra, llamada `rama2`, escribo `git Branch rama2` Para trabajar en esa rama debo moverme a ella entonces escribo `git checkout rama2` .
- 4) Para cambiar de rama en Git, debo escribir el comando `git checkout` y después del espacio el nombre de la rama a la que quiero cambiarme.
- 5) Para fusionar ramas en Git primero tengo que estar en la rama a la que le quiero fusionar la otra. Por ejemplo, tengo la rama `mine` y la rama `rama2`. Debo fusionar `rama2` en `mine`. Entonces me muevo a la rama `mine`
`Git checkout mine`
Una vez allí ya le puedo fusionar `rama2` con el siguiente comando:
`Git merge rama2`
- 6) Para ir guardando con etiquetas o nombres los cambios que voy haciendo y poder encontrarlos en el caso de que se desee revisarlos uso la función `commit`. Para ello tengo que escribir en la terminal. `git commit -m "definición de variables"`. En este ejemplo hago un `commit` que me registra en que momento definí las variables.
- 7) Para que los `commits` se guarden en el repositorio de GitHub debo escribir en la terminal de mi editor de código `git push`
- 8) Se llama repositorio remoto al repositorio que se encuentra en un servidor de la web , por ejemplo en Git Hub.
- 9) Para agregar un repositorio remoto a nuestro Git, es decir un repositorio de la web, debo escribir en la terminal `git remote add` y al lado escribir un nombre y la dirección de la pagina web.

- 10) El termino empujar se refiere a la acción de subir los avances o cambios de tu proyecto de tu repositorio local a tu repositorio remoto, es decir al repositorio que se encuentra en un servidor web. Para ello se utiliza el comando `git push`
- 11) Es lo contrario a empujar, es decir “jalaremos” los cambios que se hayan realizado en repositorio remoto, hecho por otro usuario a nuestro repositorio local, o sea nuestra PC. Para eso se utiliza el comando `git pull`
- 12) Se llama fork a la acción de traer a tu repositorio local un proyecto de otro usuario que haya subido a su repositorio remoto. Realiza una copia de ese proyecto para que se pueda editar sin afectar el original.
- 13) Para crear un fork se necesita el url con el repositorio que se quiera copiar, se entra en Git Hub con tu usuario. Y Git Hub te ofrece la opción **fork** a la derecha de la pantalla. Luego el repositorio aparece en tu cuenta. Puedes bajarlo a tu repositorio local con la opción **Code** , allí se te ofrece el link para que puedas tenerlo en tu computadora con el comando `git clone [link]`. Este comando creara una copia del repositorio en tu computadora
- 14) Podes editarlo en tu repositorio y ofrecerle la opción al usuario que vea las ediciones que hayas realizado con el debes dirigirte al url del repositorio remoto desde tu cuenta, y dirigirte a la opción `pull request` Si el otro usuario acepta la invitación tus cambios se acoplaran a su repositorio, es decir al repositorio original .
- 15) En la página de Git Hub , en el menú de la izquierda se encuentra la; opción de `pull request` , allí aparecen las solicitudes de los usuarios que quieren editar los proyectos subidos
- 16) Las etiquetas en Git nos sirven para referenciar commit importantes que hayamos realizado en nuestro repositorio con el objetivo de encontrarlos fácilmente luego. Se podría decir que destacan los commit más importantes. Este asociado al commit, no es uno.
- 17) Existen dos tipos de etiquetas o tag. Las ligeras y las anotadas. Las ligeras las utilizamos cuando no necesitamos escribirle mucha referencia. Y las anotadas son más extensas. La desventaja de las anotadas es que son más complejas de crear. Las ligeras se crean con el comando `git tag [nombre] -m [información]`
- 18) Podemos enviar una sola etiqueta a nuestro repositorio remoto o enviar todas las etiquetas que hayamos realizado a lo largo de nuestro proyecto, volvemos a usar el comando `git push [nombre de la etiqueta]`. Si queremos enviar todas las etiquetas debemos comandar `git push origins--tags`

- 19) El historial de Git, es toda la secuencia de commits que fuimos realizando a lo largo de nuestro proyecto.
- 20) Para poder ver el historial de Git, debemos comandar `git log`. Si queremos ver los últimos commit podemos comandar `git log -4` entonces se mostrarán los últimos 4 commits
- 21) Para borrar un archivo del historial de Git, primero debemos clonar el repositorio en nuestro repositorio local. Y luego comandar `rm-[nombre del archivo]`. Luego es conveniente realizar un commit para indicar este cambio. En este punto el archivo ya no se encuentra en el repositorio de Git, pero si en el historial, para borrarlo del historial se debe comandar `git filter-branch --tree-filter 'rm -f [nombre-del-archivo]' --prune-empty HEAD`
- 22) Un repositorio privado de GitHub es un repositorio creado que solo pueden ver usuarios seleccionados o solo el usuario que lo creo. Se utiliza para evitar plagios de proyectos.
- 23) Para crear un repositorio privado debo elegirlo como opción cuando lo creo, entre la opción público o privado.
- 24) Para invitar personas a un repositorio privado debo dirigirme en el repositorio a la sección settings o acceso y debo clickear en colaboradores, allí debo escribir los nombres de los usuarios de las personas que se desee que accedan a este repositorio.
- 25) El repositorio público en GitHub es aquel en el que tienen acceso todos los usuarios de GitHub.
- 26) Para crear un repositorio público debo ir al icono + en la esquina superior derecha y selección new repository, allí nos da la opción de elegir si queremos que sea público o privado.
- 27) Para compartir un repositorio público necesito el url del mismo, para ello en el repositorio debo dirigirme a la pestaña código (code en inglés), y luego a la local, allí se encuentra el código de nuestro repositorio para compartir

28) Para crear un repositorio remoto en nuestra PC debo dirigirme a la carpeta donde están mis archivos con mis proyectos, allí con el click derecho me dirijo a la opción **open git bash here.**

Se abre una terminal donde podemos escribir. Allí mismo escribimos `git init` seguido de `git add` . Ahora todos los archivos y carpetas se agregan al escenario de git. Lo conveniente despues de estos pasos es realizar un commit, por ejemplo `git commit -m "Inicio de proyecto"`