

## Práctico 2: Git y GitHub

Nombre: Gabriel Hernan Coceres

### Actividades

Respuestas de las preguntas en orden

- 1) GitHub es un sitio donde podemos compartir nuestros proyectos de forma privada o pública, es como una red social donde las personas comparten sus proyectos, siguen a otras personas, pueden copiar repositorios etc.
- 2) Se puede crear un repositorio desde el mismo sitio en GitHub u otra manera es crear un repositorio local y enlazándolo a una dirección en GitHub, primero comenzando con el comando de “git init”, luego con el comando “git remote add origin (link del repositorio)” enlazamos el repositorio local al que esta alojado en GitHub.
- 3) Para crear una rama en Git es necesario usar el comando de “git branch -m (El nombre de la rama que queremos crear)”
- 4) Para fusionar una rama en Git con otra es necesario usar el comando de “git merge (Nombre de la rama que queremos fusionar)”. Esto lo que va a hacer es que aplicará todos los cambios que se hicieron en la rama que seleccionamos en la rama en donde estamos posicionados al momento de usar “git merge”
- 5) Para crear un commit primero necesitamos agregar los cambios realizados a un estado que seria “listo” o “un área de preparación” usando el comando de “git add .” en este caso el punto quiere decir que agregamos todos los cambios, pero se puede reemplazar por un archivo en específico o varios, una carpeta o todos los archivos con determinada extensión anteponiendo un \* antes de la extensión. Luego de agregar los cambios podemos hacer un commit con el comando “git commit -m (Nombre descriptivo del commit)”. Con “-m” podemos nombrar el commit que estamos haciendo.
- 6) Para enviar un commit al repositorio de GitHub sí es la primera vez necesitamos hacerlo con el comando “git push -u origin master”, master se debe cambiar por el nombre de la rama principal que estamos utilizando, por defecto es master (Algunos usan “main”) Luego para hacer los proximos solo basta con usar “git push”
- 7) Un repositorio remoto es un lugar donde podemos alojar nuestro proyecto en internet para compartirlo o almacenarlo de forma privada, de esta forma nuestro proyecto no solo va a estar en nuestra pc si no que podemos acceder a el desde otro dispositivo ya que se encuentra en internet. Al subirlo tambien

podemos hacerlo publico y permitir que otras personas puedan descargarlo y tambien colaborar desarrollando nuevas versiones.

- 8) Para agregar un nuevo repositorio remoto podemos usar el comando de “git remote add origin (la url del repositorio que deseamos agregar)”
- 9) Con el comando “git push” podemos empujar un commit al repositorio de GitHub
- 10) Con el comando “git fetch” podemos obtener/descargar los cambios del repositorio sin aplicarlos a nuestra rama
- 11) Hacer un “fork” de un repositorio es generar una copia de un repositorio con otra url, nos sirve por ejemplo para poder crear una versión distinta de un proyecto aunque no tengamos el permiso de poder cambiarlo ya que es como si estuviéramos haciendo un proyecto paralelo.
- 12) Para crear un fork de otro repositorio necesitamos ir hasta la url del repositorio que elegimos, y arriba a la derecha esta para seleccionar “Fork” o en español se encuentra como “Tenedor” al presionar en el automáticamente se nos genera una copia del repositorio en nuestro perfil y podemos hacerle un “git clone” para poder trabajar en el de forma local en nuestro pc.
- 13) Siguiendo la respuesta 12, luego de tener el repositorio en nuestro pc y terminemos de hacer lo cambios en el proyecto. Tenemos que ir a la url del proyecto original y hacemos click en “Nueva solicitud de extraccion”, seleccionamos la rama que contiene nuestros cambios y creamos la solicitud.
- 14) Para aceptar una solicitud de extraccion debemos dirigirnos al repositorio en cuestion y en la pestaña donde dice “Pull request” podemos ver todas las solicitudes pendientes, al elegir una y leer su descripcion y corroborar todos los cambios echos si estamos de acuerdo podemos presionar le boton de "Merge pull request" para hacer la fusion a nuestro proyecto con los cambios nuevos propuestos.
- 15) Las etiquetas son “nombres” que podemos asignarles a nuestros commits cuando los vemos en el historial de cambios, es una forma para poder diferenciarlos entre si de una manera mas legible o resaltar los mas importantes.
- 16) Para crear una etiqueta en git usamos el comando de “git tag (nombre de la etiqueta)” seguido del codigo de referencia al commit que queremos asignarle el tag. En caso de que no agreguemos el código de referencia a un commit este se va a hacer por defecto al último commit que hicimos en la rama.
- 17) Simplemente con un push se puede enviar las etiquetas que creamos en git a GitHub.
- 18) Para ver el historial de git usamos el comando de “git log” nos va a mostrar el historial de los cambios que se hicieron el repositorio.
- 19) Para borrar el historial de git necesitamos usar el comando “git reset (el hash del commit) --hard” En el hash del commit debemos poner apartir de que commit

queremos mantener el historial, luego de eso hay que hacer un push con la opcion “-f” por ejemplo: “git push -f” lo que va a hacer “-f” es forzar la sobreescritura del historial.

- 20) Un repositorio privado en GitHub es un sitio en la plataforma que nos permite almacenar nuestro proyecto de forma privada. Para que las personas no puedan interactuar con él.
- 21) Al crear un repositorio GitHub nos pregunta si queremos que sea público o privado, si es que tenemos un repositorio publico creado y queremos hacerlo privado podemos cambiarlo desde la configuración del repositorio desde Github.
- 22) Para invitar a alguien a nuestro repositorio y que pueda participar en sus cambios podemos agregarlo a la lista de “colaboradores” que te permite GitHub gestionar desde su configuración.
- 23) Un repositorio público en Github es un sitio donde puedo almacenar mi proyecto para que todas las personas que quieran puedan verlo y probarlo.
- 24) Al entrar a nuestra cuenta de GitHub arriba a la izquierda podemos visualizar nuestros repositorios y un botón verde “nuevo” que nos permite crear un nuevo repositorio, seguido de eso configuramos para que el repositorio tenga las características que deseemos.
- 25) Podemos compartir un repositorio de GitHub simplemente con la url del mismo, asegurándonos de que la persona tenga permiso para poder verlo en el caso de que este privado.