1.Github es una pagina la cual nos permite subir nuestros repositorios a internet, permitiéndonos elegir si nuestro repositorio es publico o privado.

2.Para crear un repositorio en Github necesitamos ir a la pagina web, para posteriormente buscar la opción de “New repository”. Una vez hagamos clic en esa opción deberemos colocar el nombre, descripción y visibilidad del repositorio, luego de hacer eso ya deberíamos tener nuestro repositorio en Github.

3.Para crear una nueva rama en Git debemos usar el comando “Branch”, por ejemplo: git Branch ejemplo .

4.Para cambiar a otra rama en Git debemos usar el comando “Checkout”, por ejemplo: git checkout ejemplo2 . Vale recordar que antes de cambiar de rama se debería guardar el progreso que realizamos en la rama actual.

5.Para combinar ramas en Git usaremos el comando “merge”, por ejemplo supongamos que estamos en la rama “ejemplo” y queremos combinarla con la rama “ejemplo2”. Para eso realizaríamos “git merge ejemplo2”, lo cual uniria las 2 ramas.

6.Para crear un comit usaremos “commit”, por ejemplo digamos que arreglamos algún error y queremos dejarlo por escrito que lo hicimos. En ese caso usaremos “git commit -m “Arregle el error X””.

7.Para enviar un commit a Github primero debemos establecer la conexión entre Git y Github. Luego de hacer eso debemos escribir el commit deseado en Git, para posteriormente hacer (git add .) luego (git push -u origin master) y con eso ya debería estar subido nuestro commit a Github.

8.Un repositorio remoto es una versión del proyecto alojada en un servidor, lo que permite un fácil acceso a otros miembros si se trabaja en equipo, a si mismo esta se guarda en la nube. Lo que facilita acceder al proyecto desde distintos equipos.

9.Para agregar un repositorio remoto debemos tener el link de este, para luego hacer en git (git remote add origin “URL”), una vez hecho eso ya estaría conectado nuestro git y el repositorio remoto.

10.Para empujar cambios a un repositorio remoto debemos usar “push”.

11.Para tirar cambios de un repositorio remoto debemos usar “pull”.

12.Un Fork es una copia de un repositorio el cual se almacena en tu cuenta de Github, sin necesidad de descargar el archivo de forma local. Esto permite trabajar en códigos ajenos sin la necesidad de modificar el original, permitiendo presentar las cambios realizados al repositorio al autor mediante un “pull request”

13.Para crear un Fork debemos dirigirnos al repositorio original y buscar el botón que dice Fork, una vez apretado ya tendremos nuestro copia del repositorio.

14.Para enviar un Pull request debemos ir a nuestro Fork, luego seleccionar la opción “Compare & pull request”. Luego deberemos completar un cuestionario acerca de los cambios realizados, luego agregamos una descripción de los cambios realizados y le damos a la opción “Create pull request”.

15.Para aceptar un Pull request deberemos abrir el mismo y revisar los cambias realizados a nuestro repositorio, luego de analizarlo debemos elegir de que modo queremos añadirlo a nuestro código. Luego de eso solo tenemos que confirmar la unión y ya tendríamos el Pull request integrado a nuestro código.

16.Las etiquetas en Git se usan para marcar metas importantes dentro del proyecta en el cual se esta trabajando, ya sean fechas de lanzamiento o otros eventos importantes.

17.Hay diferentes formas de crear tags en Git, dependiendo de cuanto detalle o información se quiera especificar. Si queremos crear un tag solo con tombre tendríamos que usar (git tag “nombre”), si queremos agregar un mensaje (git tag -a “Nombre” -m “mensaje”). Y si a esta ultima lo queremos adjuntar a algún commit debemos usar (git tag -a “Nombre” hash -m “mensaje”).

18.Para subir estos tags a GitHub podemos usar (git push –tags), a si mismo esto guarda los cambios realizados hasta el momento.

19.El historial en Git es el registro de todos los commits realizados en un repositorio, permitiendo visualizar datos como quien realizo ciertas modificaciones, cuando se hicieron las mismas y que cosa se modifico.

20.Para ver el historial en Git podemos usar el comando (git reflog), el cual nos mostrara todos los cambios realizados.

21.Git ofrece varas formas de buscar dentro del historial, algunas de ellas son (git log) la cual permite ver todos los commits. (git log –oneline) el cual busca

commits en una línea en especifico, (git log –decorate –all –graph –oneline) el cual grafica los commits, (git log “ramaX” .. “ramaY”) permite buscar commits que se encuentren en ramaX pero no en ramaY y (git log –follow “archivo”) para buscar commits donde el archivo cambio.

22.Hay 3 formas de borrar el historial:
1- soft: En este caso se retrocede el puntero HEAD a un commit especificado, pero no altera el STAGE. (git reset –soft).

2-mixed: Es igual al soft, solo que en este caso también se retrocede el STAGE al ultimo commit. (git reset –mixed).

3-hard:Es igual que el mixed, pero además modifica los archivos y carpetas del proyecto. (git reset –hard).

23.Un repositorio privado en GitHub es un repositorio en el cual solo pueden visualizar personas que elijamos, esto permite limitar quienes tienen acceso a nuestro progreso y la capacidad de copiar o enviar Pull request.

24.Para crear un repositiorio privado devemos seleccionar la opción de privado cuando estemos creando nuestro repossitorio en Github.

25.Para invitar personas a nuestro repositorio privado nececitaremos su nombre de usuario, una vez lo tengamos debemos dirigirnos a nuestro repositorio ir a opciones seleccionar colaboradores y buscar a dicha persona ahí. Una vez lo encontremos le podremos asignar cuanto poder tendrá dentro del repositorio.

26.Un repositorio publico en GitHub es aquel el cual puede ser visualizado por cualquiera sin necesidad de permisos adicionales, a si mismo esto permite que cualquiera pueda usar nuestro código como así también mandar Pull request. Aunque vale recalcar que no pueden editar el código base sin que uno de el visto bueno antes.

27. Crear un repositorio publico es fácil, solo debemos seleccionar la opción de publico mientras lo estemos creando.

28.La forma mas fácil de compartir un repositorio publico es brindando el URL del mismo, ya que es publico para cualquier persona.