

TRABAJO PRÁCTICO N°2

GIT Y GITHUB

Actividad 1:

Contestar las siguientes preguntas utilizando las guías y documentación proporcionada (Desarrollar las respuestas):

1- ¿Qué es GitHub?

Es una plataforma de internet donde se pueden alojar repositorios de scripts o código mediante Git, permite realizar cambios que se alojen tanto en la web como en la computadora local, lo que facilita el trabajo colaborativo.

2- ¿Cómo crear un repositorio en GitHub?

Salvo se indique lo contrario, en los comandos git si aparecen comillas "" se debe tener en cuenta que son para expresar el tipo de información que contienen dentro, pero el comando no se escribe con dichas comillas.

Primero se debe crear un usuario en la página GitHub.com, ingresando mail, nombre de usuario y password. Luego cuando se entra al usuario creado, en la esquina superior derecha hay una opción con el símbolo "+", si se clickea allí y elige "Nuevo repositorio", se abre una ventana con las opciones del nuevo repositorio a crear, como el creador o "owner", nombre, si es público o privado, si se le quiere agregar una descripción, etc. Finalmente en la parte inferior se elige "Crear repositorio" y se crea el mismo, que estará vacío hasta que se le cargue un código.

3- ¿Cómo crear una rama en Git?

Una rama Git o branch se crea utilizando el comando: git branch "nombre de la rama"

4- ¿Cómo cambiar a una rama en Git?

Cambiar de una rama a otra en Git se realiza mediante el comando: git checkout "nombre de la rama ya creada"

5- ¿Cómo fusionar ramas en Git?

El comando a utilizar (falta especificar que rama fusionaremos) es: git merge, lo que hace este comando es acoplar o integrar en la rama en la cual estamos parados dentro de Git, la "otra rama" que ya existe y cuyo nombre se debe conocer. Entonces para fusionar ramas se debe estar dentro de la rama en la cual fusionaremos la "otra rama" con el comando: git merge "otra rama"

6- ¿Cómo crear un commit en Git?

Para crear un commit, es decir un punto de control de los cambios que hemos realizado, se debe utilizar el comando: git commit -m "mensaje"

Entre comillas escribiremos un mensaje indicando los cambios hechos o alguna referencia de utilidad. A diferencia de los ejemplos anteriores, **aquí el mensaje sí va entre comillas**.

7- ¿Cómo enviar un commit a GitHub?

Una vez creado un GitHub como se indicó previamente, para enviar un commit a dicho GitHub se utiliza el comando: `git push origin "main o master"` sube los cambios de la rama local master o main al repositorio remoto, sin hacer un seguimiento automático de la rama master o main en dicho repositorio remoto. Comando alternativo: `git push -u origin "main o master"` sube los cambios de la rama local master o main al repositorio remoto, pero de esta forma rastrea automáticamente la rama master o main en dicho repositorio remoto, por lo que en futuras ejecuciones del comando, Git recordará a qué rama se hace referencia.

8- ¿Qué es un repositorio remoto?

Es un repositorio que se aloja en un servidor remoto, al que se accede mediante una página web como GitHub o GitLab teniendo un usuario en dicha página, estos servidores permiten la interacción con los repositorios locales de Git que uno tenga.

9- ¿Cómo agregar un repositorio remoto a Git?

Para agregar un repositorio remoto se debe utilizar el comando: `git remote add origin "url del repositorio"`

10- ¿Cómo empujar cambios a un repositorio remoto?

Para empujar o hacer un push de cambios a un repositorio remoto, dentro del repositorio local Git se debe utilizar el comando: `git push origin "nombre de rama"`

11- ¿Cómo tirar de cambios de un repositorio remoto?

Para tirar, obtener o hacer un pull de cambios de un repositorio remoto hacia un repositorio local, se debe utilizar el comando: `git pull origin "nombre de rama"`

12- ¿Qué es un fork de repositorio?

Un fork es una copia de un repositorio remoto de un Usuario1 en el repositorio remoto de otro Usuario2. De esta forma el Usuario2 puede realizar cambios sobre el repositorio copiado sin que afecte dicho repositorio en la cuenta del Usuario1. Se utiliza la palabra fork como bifurcación, ya que se interpreta que la copia y posterior modificación del repositorio genera una bifurcación del original.

13- ¿Cómo crear un fork de un repositorio?

Dentro del repositorio remoto, por ejemplo GitHub, se clickea en la opción "Fork" y luego se elige la cuenta destino donde se realizará dicho Fork o clone.

14- ¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?

Para crear una solicitud de pull request a un repositorio remoto, se debe estar en el repositorio propio dentro de GitHub, ingresar a la opción “Pull request”, indicar en un mensaje (la página deja espacio para que escribamos las consideraciones que querramos anotar) los cambios realizados en el repositorio correspondiente y luego clicar sobre “Create pull request” para que GitHub envíe la solicitud al otro usuario para que acepte las modificaciones hechas.

15- ¿Cómo aceptar una solicitud de extracción?

Para aceptar la solicitud de extracción, en GitHub, alojado en la parte superior derecha hay una opción que indica “you pull requests”, se clickea sobre esa opción, se revisan los cambios y para aceptarlos se clickea sobre la opción “merge pull request” luego se confirma mediante la opción “confirm merge”

16- ¿Qué es un etiqueta en Git?

Una etiqueta es un indicador en el historial del Git que sirve para indicar versiones o hitos específicos, es similar a una rama excepto que no se modifica en el tiempo ya que la etiqueta se fija en un commit dado.

17- ¿Cómo crear una etiqueta en Git?

Para crear una etiqueta en Git se debe utilizar el comando: `git tag “nombre de etiqueta”`

Comando alternativo: `git tag -a “nombre de la etiqueta” -m “mensaje haciendo alguna aclaración pertinente”`

18- ¿Cómo enviar una etiqueta a GitHub?

Se envía una etiqueta a GitHub mediante el comando: `git push -tags`

19- ¿Qué es un historial de Git?

Un historial de Git es un listado donde se registran todos los commits realizados en el repositorio, de esta forma se puede visualizar los cambios que se han ido sucediendo en el mismo.

20- ¿Cómo ver el historial de Git?

Para generar el listado de registros de Git se utiliza el comando: `git log`

21- ¿Cómo buscar en el historial de Git?

Se pueden hacer diversos tipos de búsquedas utilizando diferentes criterios.

Búsqueda mediante palabra clave, se utiliza el comando: `git log -S “palabra clave”`

Búsqueda mediante autor: `git log -author=“nombre del autor”`

Búsqueda mediante modificación de un archivo en particular: `git log --“nombre de archivo.extensión”`

Búsqueda mediante palabras ingresadas en el mensaje del commit: `git log --grep="palabra buscada"`

22- ¿Cómo borrar el historial de Git?

Para borrar un commit se debe utilizar el comando: `git rebase` este comando permite a su vez editar, reordenar y modificar commits.

23- ¿Qué es un repositorio privado en GitHub?

Es un repositorio remoto donde solo tiene acceso el usuario propietario y a quien él haya otorgado permiso de acceso.

24- ¿Cómo crear un repositorio privado en GitHub?

Al crear un nuevo repositorio en GitHub, se debe elegir la opción "Private" que se encuentra debajo de la opción "Public", y luego "creat repository" para crearlo.

25- ¿Cómo invitar a alguien a un repositorio privado en GitHub?

Dentro de GitHub se debe acceder a la opción "Settings" luego se accede a "Collaborators" donde se ingresa los nombres de usuario o mail de los colaboradores a invitar, y finalmente se envía la invitación para su aceptación.

26- ¿Qué es un repositorio público en GitHub?

Es un repositorio remoto el cual es visible para cualquier usuario, permitiendo que lo vean y clonen el código que tenga dentro el repositorio público.

27- ¿Cómo crear un repositorio público en GitHub?

Al crear un nuevo repositorio en GitHub, se debe elegir la opción "Public" que se encuentra por encima de la opción "Private", y luego "creat repository" para crearlo.

28- ¿Cómo compartir un repositorio público en GitHub?

Se puede compartir el repositorio remoto enviando el enlace URL a otras personas.