

Práctico 2: Git y GitHub

Duarte Federico

Actividades

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

- 1.1. • **¿Qué es GitHub?**

- 1.1.1. GitHub es un repositorio remoto que nos permite poder ir subiendo nuestros proyectos y sus modificaciones en el tiempo para luego poder tener acceso desde cualquier lugar con conexión a internet, se puede crear un repositorio público como remoto, ver repositorios de otros usuarios, colaborar en distintos proyectos colaborativos etc.

- 1.2. • **¿Cómo crear un repositorio en GitHub?**

- 1.2.1. Para crear un repositorio en GitHub se deben seguir una serie de pasos, primero hay que ingresar en la página con nuestro usuario y contraseña en este punto hay varias formas de llegar a la sección para poder crea un nuevo repositorio, la que yo uso es : posicionarse en el icono de mi usuario en la parte derecha superior de la página, hacer click ,luego ir a la pestaña “ your Repositories” o “tus proyectos” y luego hacer click en el botón “new” o “nuevo”. Una vez allí dentro se piden ciertos datos como el nombre del repositorio, una pequeña descripción , si es privado o público y otras opciones. Allí se debe hacer click en “Create repository” o “Crear repositorio” y ya estaría listo para su uso

- 1.3. • **¿Cómo crear una rama en Git?**

- 1.3.1. Se debe ejecutar el comando “git branch” y el nombre de la nueva rama que se quiera crear, de modo tal que el comando si la nueva rama se llama nueva es :“ git branch nueva”

- 1.4. • **¿Cómo cambiar a una rama en Git?**

- 1.4.1. Primero deberías saber sobre que rama estas posicionado, para esto se ejecuta el comando “git branch” y con un asterisco “*” te marca sobre que rama estás posicionado. Para cambiar de rama se ejecuta “git checkout “ y el nombre de la rama , por ejemplo : git checkout nueva.

- 1.5. • **¿Cómo fusionar ramas en Git?**

- 1.5.1. Para poder fusionar las ramas que tengamos en nuestro proyecto hay que primero estar en la rama principal de del mismo esto se logra usando el comando “checkout” y una vez posicionado en esa rama principal hay que ejecutar el comando “git merge “ y la rama a fusionar, por ejemplo : git merge nueva. De este modo la rama nueva se fusiona con la rama principal.

- 1.6. • **¿Cómo crear un commit en Git?**

- 1.6.1. Los commits en Git se crean mediante el comando “git commit -m “Comentario ””.

- 1.7. • **¿Cómo enviar un commit a GitHub?**

- 1.7.1. Para que los cambios a nivel local de los commits se suban al repositorio de GitHub hay que utilizar el comando “git push -u origin master” por defecto.

- 1.8. • **¿Qué es un repositorio remoto?**

- 1.8.1. Es un servicio para guardar los repositorios locales en forma remota dentro de sus servidores para poder abrirlo y editarlo desde cualquier lugar que el usuario desee

- 1.9. • **¿Cómo agregar un repositorio remoto a Git?**
 - 1.9.1. Para poder hacer uso del repositorio, que está en el servidor, de forma local lo que se hace es clonar el mismo para así poder tenerlo localmente. En git se usa el comando “git clone “ y la url del repositorio.
- 1.10. • **¿Cómo empujar cambios a un repositorio remoto?**
 - 1.10.1. Para empujar los cambios desde el repositorio local al remoto se utiliza el comando “git push -u origin master” para hacerlo por primera vez y luego con poner solo “git push” es suficiente, de este modo se suben los cambios realizados al repositorio.
- 1.11. • **¿Cómo tirar de cambios de un repositorio remoto?**
 - 1.11.1. Para tirar o hacer pull en un repositorio local, es decir traer lo del repositorio remoto al local, se debe usar el comando “git pull origin master”. Así lo que sucede es que los cambios se hacen a nivel local desde el repositorio remoto.
- 1.12. • **¿Qué es un fork de repositorio?**
 - 1.12.1. Un fork es un repositorio que se ha sido creado para que la comunidad pueda usarlo sin que esto afecte al código original. El fork es una copia de repositorios en forma remota, es decir que se copia el repositorio de el usuario a en el repositorio del usuario b.
- 1.13. • **¿Cómo crear un fork de un repositorio?**
 - 1.13.1. Para crear un fork hay que ir al repositorio que se quiera usar , luego hacer click en el botón fork, una vez en esa sección se pueden editar ciertos datos como el nombre y la descripción del mismo, luego se debe hacer click en el botón Create fork y ya se puede hacer uso de ese repositorio en tu cuenta
- 1.14. • **¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?**
 - 1.14.1. Primero hay que hacer un fork del repositorio en cuestión, así tenemos el repositorio en nuestra cuenta de GitHub, para luego hacer un clon de ese fork a nivel local. Una vez se tenga localmente se debe hacer un branch o rama agregando los cambios que queremos realizar. Luego se suben los cambios realizados al fork que hicimos en GitHub. Para finalizar debemos dirigirnos al repositorio original, del cual hicimos el fork y hacer click en el botón “Compare & Pull Request”, se completan los datos como el título y se describe lo que se hizo para luego hacer click en “Create Pull request” para finalizar.
- 1.15. • **¿Cómo aceptar una solicitud de extracción?**
 - 1.15.1. Dentro de GitHub hay que dirigirse a la pestaña Pull Request , una vez allí dentro hay que hacer click en el botón Merge Pull Request, así se confirmará la fusión con los cambios
- 1.16. • **¿Qué es una etiqueta en Git?**
 - 1.16.1. Las etiquetas en Git hacen referencia a la versión, refiriéndose a cambios de relevancia dentro del proyecto
- 1.17. • **¿Cómo crear una etiqueta en Git?**
 - 1.17.1. Para crear una etiqueta en Git se debe utilizar el comando -“git tag” nombre-
- 1.18. • **¿Cómo enviar una etiqueta a GitHub?**
 - 1.18.1. Las etiquetas se pushean a GitHub con el comando -“git push “ nombre- en caso de querer hacerlo con una sola etiqueta, para poder subir todas las etiquetas se utiliza el comando -“git push – tags”
- 1.19. • **¿Qué es un historial de Git?**
 - 1.19.1. Es el registro histórico de los commits en el repositorio literalmente es un historial con los cambios, quien los hizo, que se cambio etc.
- 1.20. • **¿Cómo ver el historial de Git?**
 - 1.20.1. En Git los commits se ven con el comando -“git log”-
- 1.21. • **¿Cómo buscar en el historial de Git?**
 - 1.21.1. Para buscar en el historial de Git se pueden usar varios comandos, en principio si se quiere buscar cambios respecto a la palabras claves se usa -“git log -p -s “palabra”” -. Pero si se quiere buscar los cambios realizados por alguien específico se puede utilizar el comando -“git log – author “nombre del autor””-
- 1.22. • **¿Cómo borrar el historial de Git?**

- 1.22.1. Hay Varias formas de borrar el historial de git, una de ellas es con el comando -"got rebase -i <commit-base>, con el cual se pueden borrar los commits.
- 1.23. • **¿Qué es un repositorio privado en GitHub?**
 - 1.23.1. Es un repositorio al cual solo tiene acceso el creador del mismo y los miembros a los que él da acceso .
- 1.24. • **¿Cómo crear un repositorio privado en GitHub?**
 - 1.24.1. Al crear un repositorio se debe marcar la opción de privado.
- 1.25. • **¿Cómo invitar a alguien a un repositorio privado en GitHub?**
 - 1.25.1. Hay que estar en el repositorio privado, una vez dentro se va a la pestaña de Settings, allí a la izquierda , debajo de general está el apartado Collaborators, y allí se debe hacer click sobre Add people, en la ventana emergente que se abre se coloca el mail o bien el nombre de usuario para enviar la invitación.
- 1.26. • **¿Qué es un repositorio público en GitHub?**
 - 1.26.1. Es un repositorio que es público para todos, se puede clonar, o bien hacer un fork o colaborar.
- 1.27. • **¿Cómo crear un repositorio público en GitHub?**
 - 1.27.1. A la hora de crear el repositorio simplemente se tilda la opcion de de public y ya quedaría configurado de ese modo, También podemos hacer que un repositorio pase de público a privado en la pestaña de Settings, luego en General, allí hay que dirigirse a la parte inferior, a Danger Zone y se toca en el botón Change visibility.
- 1.28. • **¿Cómo compartir un repositorio público en GitHub?**
 - 1.28.1. En la página Code del proyecto hay que hacer click en el boton "< > Code" y allí se despliega un menú con el link de nuestro proyecto