

TP UNIDAD 2

Alumno: Juan Ignacio Rouge

PREGUNTAS:

- ¿Qué es GitHub?
- ¿Cómo crear un repositorio en GitHub?
- ¿Cómo crear una rama en Git?
- ¿Cómo cambiar a una rama en Git? • ¿Cómo fusionar ramas en Git?
- ¿Cómo crear un commit en Git?
- ¿Cómo enviar un commit a GitHub?
- ¿Qué es un repositorio remoto?
- ¿Cómo agregar un repositorio remoto a Git?
- ¿Cómo empujar cambios a un repositorio remoto?
- ¿Cómo tirar de cambios de un repositorio remoto?
- ¿Qué es un fork de repositorio?
- ¿Cómo crear un fork de un repositorio?
- ¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?
- ¿Cómo aceptar una solicitud de extracción?
- ¿Qué es una etiqueta en Git?
- ¿Cómo crear una etiqueta en Git?
- ¿Cómo enviar una etiqueta a GitHub?
- ¿Qué es un historial de Git?

- ¿Cómo ver el historial de Git?
- ¿Cómo buscar en el historial de Git?
- ¿Cómo borrar el historial de Git?
- ¿Qué es un repositorio privado en GitHub?
- ¿Cómo crear un repositorio privado en GitHub?
- ¿Cómo invitar a alguien a un repositorio privado en GitHub?
- ¿Qué es un repositorio público en GitHub?
- ¿Cómo crear un repositorio público en GitHub?
- ¿Cómo compartir un repositorio público en GitHub?

Respuestas:

1: GitHub es una plataforma online para alojar y gestionar repositorios Git. Ayuda a la colaboración entre programadores al dar herramientas como control de versiones, issues, pull requests y revisiones de código.

2: Para crear un repositorio en GitHub hay que seguir estos pasos: Primero, se inicia sesión en GitHub, después haces click en tu foto de perfil y vas a tu perfil, después creas un nuevo repositorio haciendo click en el signo + que está arriba a la derecha, dentro de la creación del repositorio pones el nombre del repositorio y la descripción, también puedes ponerlo en privado o público y agregar un archivo readme y luego haces click en crear repositorio.

3: Para crear una rama en Git hay que seguir estos pasos: Abrir la terminal en el directorio donde quieres crear la rama. Luego se escribe este comando (sin comillas): "git branch feature".

4: Para cambiarnos a la rama que creamos escribimos el comando (sin comillas): git switch feature. También, para confirmar que estamos en la nueva rama podemos escribir el comando (sin comillas): git branch. Esto te va a mostrar todas las ramas, y la activa estará señalada con un asterisco. Para fusionar ramas en git, primero vamos a una rama con el comando que puse anteriormente. Ahora supongamos que queremos fusionar otra rama llamada main, con la rama feature. Vamos a ir a la rama feature y vamos a escribir en la consola el comando (sin comillas): "git merge main". Esto va a integrar los cambios de la rama main a la rama feature.

5: Para realizar un commit en Git, seguimos estos pasos: Primero realizamos y guardamos los cambios hechos a los archivos de nuestro repositorio. Despues usamos el siguiente comando para agregar esos cambios a el area de preparacion (sin comillas) : "git add <archivo>". Y si deseamos agregar todos los cambios usamos el comando (sin comillas): "git add . ". El siguiente paso es crear el commit, para eso usamos el siguiente comando (sin comillas): "git commit -m", despues del -m podemos agregar una descripcion, el mensaje debe estar escrito entre comillas.

6: Para enviar un commit a GitHub primero vinculamos el repositorio local con el de GitHub, utilizamos el siguiente comando (sin comillas): "git remote add origin <URL-del-repositorio-en-GitHub>". Despues, utilizamos el siguiente comando en la consola (sin comillas): "git push origin <nombre-de-la-rama>"

7: Un repositorio remoto es una versión de un repositorio que está almacenada en un servidor en línea o en una red, en lugar de estar en la computadora local del programador. Nosotros usamos GitHub para subir los repositorios y así hacerlos remotos.

8: Para agregar un repositorio remoto a Git, primero tienes que tener un repositorio remoto. Luego nos vamos a la consola y nos posicionamos en el proyecto. Después agregamos el repositorio remoto con este comando (sin comillas): `git remote add origin <URL>` (borramos el texto que dice URL y ahí pegamos el link del repositorio remoto). Y por último podemos subir el código al repositorio remoto si hicimos cambios con estos comandos en la consola (sin comillas): `git branch -M main` y `git push -u origin main`.

9: Para empujar cambios a un repositorio remoto usamos los comandos `git branch -M main` y `git push -u origin main`. Antes asegúrate de haber hecho todos los pasos anteriores.

10: Para retirar cambios de un repositorio remoto, primero nos fijamos en qué rama estamos con el comando (sin comillas): `git branch`, si estamos en la rama deseada ejecutamos el comando (sin comillas): `git pull origin main`, si no estamos en la rama deseada ejecutamos el comando (sin comillas): `git checkout (nombre de la rama)` y luego ejecutamos el comando del paso anterior.

11: Un fork en Git es una copia de un repositorio que se hace en otra cuenta, hay varias plataformas que permiten fork, como por ejemplo GitHub o GitLab. Esto nos permite hacer cambios a un repositorio sin afectar al repositorio original.

12: Para hacer fork a un repositorio en GitHub primero localizamos el repositorio al que queremos hacer fork. Luego nos dirigimos arriba a la derecha a la opción que dice fork. Le cambiamos el nombre y la descripción y listo, ponemos crear fork y ya puedes empezar a hacer cambios sin afectar el repositorio original.

13: Primero vamos a nuestro repositorio, luego buscamos la pestaña pull requests, y ahí seleccionamos como destino el repositorio original (base repository) y como origen elegimos la rama en la que hicimos cambios (compare). Después escribimos un título y una descripción con los cambios que hicimos y listo, clickeamos en create pull request.

14: Podemos aceptar la pull request clickeando en merge pull request, con esta opción unimos los cambios a la rama principal. También podemos usar confirm merge, y con esta opción confirmamos la fusión.

15: En Git, una etiqueta es un marcador fijo que se usa para señalar puntos importantes a lo largo del proyecto, como versiones de lanzamiento. Es similar a una rama, pero no cambia con el tiempo.

16: Para crear una etiqueta podemos usar 2 comandos (sin comillas): "git tag nombre-del-tag", con este comando creamos una etiqueta básica que solo apunta un commit sin información adicional. También podemos usar este comando (sin comillas): "git tag -a nombre-del-tag -m 'Descripción de la etiqueta'" con esta etiqueta anotamos el nombre del autor, la fecha y un mensaje de descripción. El mensaje si debe estar entre comillas.

17: El historial de Git es el registro de todos los cambios que se hicieron en un repositorio, ordenados en commits. Permite ver qué modificaciones se hicieron, quién las hizo y cuándo.

18: Para ver el historial es fácil, solo tenemos que usar el comando (sin comillas): "git log". Esto nos muestra el registro de los commits con orden cronológico inverso, del más nuevo al más viejo. También podemos usar el comando (sin comillas): "git log --oneline" para verlo de forma más compacta.

19: Para buscar cambios en el log podemos usar varios tipos de comandos para filtrar los commits. Primero podemos buscar un commit con una frase o palabra específica con el comando (sin comillas): "git log --grep="palabra o frase"" (la palabra o frase va entre comillas). Luego podemos buscar un commit que esté afectando a un archivo en específico con el comando (sin comillas): "git log -- nombre-del-archivo". También podemos buscar cambios realizados en un rango de fechas con el comando (sin comillas): "git log --since="A-M-D" --until="A-M-D"" (las fechas van entre comillas en el orden año mes y día). Otra opción es si sabemos el hash del commit es buscarlo con el hash, el comando que utilizamos es (sin comillas): "git show hash-del-commit". Con este otro método podemos buscar un commit que cambie una línea específica de un archivo, el comando es (sin comillas): "git blame nombre-del-archivo". Y por último podemos buscar commits relacionados con una rama con el siguiente comando (sin comillas): "git log nombre-de-la-rama".

20: Para borrar el historial de commits de manera local, tenemos que seguir los siguientes pasos:

1: Ejecutar el comando (sin comillas): "git rebase -i HEAD~5" donde está el 5 podemos poner la cantidad de commits que queramos ver, en este caso nos mostrará los últimos 5.

2: Cambiamos la palabra de pick a drop al costado de cada commit que queramos eliminar. Si no lo queremos eliminar dejamos pick.

3: Guardar y cerrar el editor.

También podemos usar el comando (sin comillas): `git reset --hard hash_del_commit` Este comando sirve para empezar de nuevo desde un commit en específico, borrando todos los commits posteriores al seleccionado

21: Un repositorio privado en GitHub, es un repositorio que solo vos o las personas que elijan pueden acceder. Que a diferencia del público, cualquiera lo puede ver, clonar o modificar.

22: Para hacer un repositorio privado en GitHub, seguimos los mismos pasos para crear un repositorio que mencioné anteriormente. Pero en la parte inferior en vez de dejar seleccionado público, seleccionamos privado.

23: Para invitar colaboradores ingresamos al repositorio privado y arriba a la derecha nos va a salir un botón grande que dice invitar colaboradores a este repositorio. Hacemos click y probablemente se nos abra una ventana donde nos pida nuestra contraseña para verificar que somos nosotros. Luego abajo nos aparece una ventana donde podemos escribir el nombre de usuario, nombre completo o correo electrónico de la persona que queramos invitar a nuestro repositorio.

24: Un repositorio público al contrario del privado permite que cualquiera acceda a él sin necesidad de ser invitado. Cualquiera puede clonarlo, hacer cambios y verlo.

25: Para crear un repositorio público, lo hacemos de la misma manera que el privado, solo que dejamos seleccionada la opción por defecto que es público.

26: Para compartir un repositorio público de GitHub, solamente tenemos que copiar la url y enviarla a la persona que deseamos que lo vea. Al ser público no necesita ningún tipo de permiso para acceder a él.