

TP2- Git y GitBub

Carolina Benitez

1. ¿Qué es GitHub?

Es una plataforma para alojar y compartir repositorios de control de versiones basada en Git de manera pública o privada. Permite a los programadores colaborar en proyectos y trabajar en equipo.

2. ¿Cómo crear un repositorio en GitHub?

Para subir un repositorio a github desde nuestro git se clickea en “New repository”, donde se coloca el nombre, una descripción y si queremos que sea público o privado. Al finalizar, damos click en “Create Repository. Luego de crear el repositorio, GitHub muestra instrucciones para:

Subir archivos manualmente

Clonar el repositorio con Git usando el comando `git clone`

Vincular un repositorio existente.

3. ¿Cómo crear una rama en Git?

Para crear una nueva rama, se utiliza el comando `git branch nueva rama`.

Esto no significa que estemos posicionados en la nueva rama, ya que seguimos parados en la rama main.

4. ¿Cómo cambiar a una rama en Git?

Para cambiar a la nueva rama, se utiliza el comando `git checkout nueva rama`.

5. ¿Cómo fusionar ramas en Git?

Para fusionar ramas en Git, primero se posiciona en la rama principal (main) donde queremos fusionar los cambios usando el comando `git checkout main`. Luego, para fusionar e incorporar los cambios de la rama nueva en la rama principal se utiliza el comando `git merge nueva rama`.

6. ¿Cómo crear un commit en Git?

Primero se realizan los cambios en los archivos de tu proyecto que quieras guardar en el commit. Luego, se agregan los cambios usando el comando `git add`. Una vez hecho esto, se crea el commit con el comando `git commit` agregando un mensaje que describa los cambios realizados: `git commit -m "Descripción de los cambios"`.

7. ¿Cómo enviar un commit a GitHub?

Primero necesitamos hacer un push (clonar) al repositorio de GitHub a nuestra máquina local:

```
git clone https://github.com/tu\_usuario/tu\_repositorio.git
```

```
cd tu_repositorio
```

Se realizan cambios en los archivos del repositorio y se agregan:

```
git add nombre_del_archivo o git add .
```

Crea un commit de tus cambios:

```
git commit -m "Descripción de los cambios"
```

Para enviar los commits al repositorio en GitHub, los empujamos con:

```
git push origin nombre_de_la_rama
```

8. ¿Qué es un repositorio remoto?

Un repositorio remoto en Git es una versión del repositorio que está guardado en Internet y puede ser compartido con otros, permitiendo colaborar con otros programadores y mantener una copia segura del código.

9. ¿Cómo agregar un repositorio remoto a Git?

Para vincular tu repositorio local con uno remoto, se utiliza el comando `git remote add` nombre del repositorio remoto.

Para verificar que se agregó correctamente, se utiliza el comando `git remote -v`.

10. ¿Cómo empujar cambios a un repositorio remoto?

Para empujar cambios a un repositorio se utiliza `git push origin nombre_de_la_rama`.

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

Para tirar los cambios del repositorio remoto se utiliza el comando `git pull`. Para descargar y fusionar los cambios del repositorio remoto con tu rama local: `git pull origin nombre_de_la_rama`.

12. ¿Qué es un fork de repositorio?

Un fork de repositorio es una copia local de un repositorio remoto existente a tu cuenta, lo que te da independencia para trabajar sobre ese proyecto sin afectar el repositorio original.

13. ¿Cómo crear un fork de un repositorio?

Para crear un fork de un repositorio en Github, se accede al repositorio que quieras forkear, creando una copia en tu cuenta. Para trabajar en tu fork localmente, se realiza una clonación en una carpeta, procedemos a realizar algún cambio, lo agregamos al stage y lo commiteamos. Luego, mandamos los cambios a el forking que hicimos con un `git push origin main`.

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

Para enviar una solicitud de pull request a un repositorio en GitHub, nos dirigimos a pull request, damos click en new pull request y luego en create pull request en donde podemos colocar un mensaje describiendo la razón de los cambios.

15. ¿Cómo aceptar una solicitud de extracción?

El autor del repositorio original puede ver en sus pull requests el mensaje que le enviamos, y si considera aceptar esta modificación, puede clicar en Merge pull request y de esta manera sumará a su repositorio los cambios realizados.

16. ¿Qué es una etiqueta en Git?

Una etiqueta es un marcador o referencia a un punto específico en la historia del repositorio. Se usa para indicar versiones de software importantes en el desarrollo.

17. ¿Cómo crear una etiqueta en Git?

Git utiliza dos tipos de etiquetas: ligeras y anotadas. Una etiqueta ligera es un puntero a un commit, para crearla se utiliza `git tag v1.0`. Las anotadas contienen el nombre del etiquetador, correo electrónico y fecha, para crearla se utiliza `git tag -a v1.0 -m "Versión 1.0 estable"`.

18. ¿Cómo enviar una etiqueta a GitHub?

Primero, verificamos las etiquetas creadas en tu repositorio local con `git tag`.

Luego, para subir (empujar) una etiqueta en particular al repositorio remoto, se utiliza `git push origin v1.0`. Para subir todas las etiquetas creadas, se utiliza `git push origin --tags`.

19. ¿Qué es un historial de Git?

El historial de Git es un registro de todos los cambios realizados en un repositorio. Contiene información sobre cada commit, como el autor, la fecha, el mensaje enviado y el identificador único del commit. Se utiliza para mantener el control de tu código.

20. ¿Cómo ver el historial de Git?

Se procede a utilizar el comando `git log`. Esto muestra una lista de commits en orden cronológico inverso. Para ver un resumen de los commits, cada uno representado en una sola línea, se utiliza el comando `git log --oneline`.

21. ¿Cómo buscar en el historial de Git?

Para buscar en el historial de commits, se pueden utilizar varios comandos y opciones que te permiten filtrar y localizar commits específicos.

Para buscar commits que contengan una palabra o frase específica en el mensaje de commit, se utiliza `git log --grep="palabra clave"`

Para buscar commits que modificaron un archivo específico, se utiliza `git log --nombre_del_archivo`

Para buscar commits en un rango de fechas específico, se utiliza `git log --since="fecha_especifica" o --until="fecha_especifica"`

Para encontrar commits hechos por un autor específico, se utiliza `git log --author="Nombre del Autor"`.

22. ¿Cómo borrar el historial de Git?

Borrar el historial de Git no es una operación común, ya que Git está diseñado para mantener un registro permanente de los cambios. Sin embargo, existen algunas formas de hacerlo, dependiendo de tu objetivo.

Para borrar todo el historial pero conservar los archivos se utiliza el comando `rm -rf .git`.

Para borrar el historial de commits de una rama específica se utiliza:

`git checkout --orphan nueva-rama` (crea una nueva rama sin historial)

`git add .`

`git commit -m "reinicio del historial"`

`git branch -D main` (elimina la antigua rama principal)

`git branch -M main` (renombrar la nueva rama como main)

`git push --force origin main` (fuerza la actualización en el remoto).

Para borrar un commit en particular sin borrar todo el historial se utiliza `git rebase -i HEAD-N` (reemplaza N con el número de commits que quieras modificar).

23. ¿Qué es un repositorio privado en GitHub?

Un repositorio privado en GitHub es un tipo de repositorio en el que el contenido solo es accesible para usuarios específicos que fueron autorizados. Limita el acceso a los colaboradores que elijas.

24. ¿Cómo crear un repositorio privado en GitHub?

Se procede ingresando a la página de creación de repositorios, hacer clic en el botón "+" y seleccionar "New Repository". Completa la información del repositorio (nombre del repositorio, descripción) y selecciona la configuración de privacidad.

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

Accede al repositorio, clickea en la pestaña "Settings" del repositorio. Selecciona "Collaborators", clickea en el botón "Add people" e ingresa el nombre de usuario de GitHub de la persona que deseas invitar. Selecciona el nivel de acceso que deseas otorgar y clickea en el botón "Add" para enviar la invitación.

26. ¿Qué es un repositorio público en GitHub?

Un repositorio público en GitHub es un repositorio con contenido accesible a cualquier persona en Internet. Permite que cualquier persona pueda ver, clonar y, si tienen los permisos adecuados, contribuir al proyecto.

27. ¿Cómo crear un repositorio público en GitHub?

Se procede ingresando a la página de creación de repositorios, hacer clic en el botón “+” y seleccionar “New Repository”. Completar la información del repositorio (nombre del repositorio, descripción) y seleccionar la configuración de privacidad.

28. ¿Cómo compartir un repositorio público en GitHub?

Para compartir tu repositorio, accede a tu repositorio, copia la URL de tu repositorio que se encuentra en un cuadro de texto que dice "<> Code":

Se puede copiar la URL directamente haciendo clic en el botón de copiar a la derecha de la URL.

2) <https://github.com/Carolina-Benitez/mi-primer-repositorio.git>

3) <https://github.com/Carolina-Benitez/conflict-exercise.git>