

Práctico 2: Git y GitHub

Alumno: Resende De Moura Juan Ignacio

Comisión : 24

1) ¿Qué es GitHub?

GitHub es una plataforma que me permite guardar proyectos con control de versiones usando Git. Es como un espacio online donde puedo subir mis repositorios para tenerlos respaldados, compartirlos con otras personas, colaborar en equipo y ver el historial de cambios que fui haciendo en mis archivos. También se usa mucho para proyectos de software, tanto personales como profesionales.

2) ¿Cómo crear un repositorio en GitHub?

Para crear un repositorio en GitHub, primero tengo que iniciar sesión en la página. Una vez adentro, hago clic en el botón verde que dice "New" (o "Nuevo repositorio"). Ahí le pongo un nombre, puedo agregar una descripción si quiero, elijo si va a ser público o privado, y por lo general no marco nada más (ni README, ni gitignore, ni licencia) para evitar conflictos si ya tengo un proyecto armado en mi computadora. Después hago clic en "Create repository", y listo.

3) ¿Cómo crear una rama en Git?

Crear una rama en Git es útil cuando quiero trabajar en una parte del proyecto sin afectar la rama principal. Para hacerlo uso el siguiente comando:

```
git branch nombre-de-la-rama
```

Eso crea una rama nueva, pero me quedo en la rama actual hasta que yo decida cambiarme.

4) ¿Cómo cambiar a una rama en Git?

Una vez que tengo una rama creada, puedo moverme a ella con el comando:

```
git checkout nombre-de-la-rama
```

Eso me permite trabajar ahí como si fuera una versión paralela del proyecto. También, si quiero crear y cambiarme en un solo paso, puedo usar

`git checkout -b nombre-de-la-rama`

5) ¿Cómo fusionar ramas en Git?

Cuando termino de trabajar en una rama y quiero juntar esos cambios con otra (por lo general, la principal), me voy a la rama principal con:

`git checkout main` (o `master`, según cómo se llame), y después uso: `git merge nombre-de-la-rama`

Eso copia los cambios de esa rama a la que estoy parado. Si no hay conflictos, se fusiona todo sin problema.

6) ¿Cómo crear un commit en Git?

Después de hacer cambios en los archivos del proyecto, tengo que guardarlos con un commit. Primero agrego los archivos que quiero incluir con:

`git add .`

(el punto significa todos los archivos modificados). Después hago el commit con:

`git commit -m "mensaje descriptivo"`

Ese mensaje es para saber qué hice en ese momento del proyecto.

7) ¿Cómo enviar un commit a GitHub?

Una vez hecho el commit, si quiero subirlo a GitHub, uso:

`git push`

Pero antes, tengo que haber agregado un repositorio remoto y haberlo configurado. La primera vez que subo una rama uso:

`git push --set-upstream origin nombre-de-la-rama`

Después de eso, `git push` alcanza.

8) ¿Qué es un repositorio remoto?

Un repositorio remoto es una versión del repositorio que está en la nube (por ejemplo, en GitHub). Es donde puedo guardar mis archivos de forma online, compartirlos con otros y mantener una copia sincronizada con la de mi computadora.

9) ¿Cómo agregar un repositorio remoto a Git?

Para agregarlo, primero tengo que crear el repositorio en GitHub (vacío), y después usar este comando en mi terminal:

```
git remote add origin https://github.com/mi-usuario/mi-repo.git
```

Eso le dice a Git que ese es el repositorio remoto con el nombre "origin". Solo se hace una vez.

10) ¿Cómo empujar cambios a un repositorio remoto?

Para subir los cambios que tengo en mi máquina a GitHub, uso:

```
git push
```

Si es la primera vez que subo esa rama, tengo que hacerlo así:

```
git push --set-upstream origin nombre-de-la-rama
```

Después, con solo usar `git push` ya alcanza en las próximas veces.

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

Si alguien más hizo cambios en el repositorio remoto y quiero traerlos a mi máquina, uso:

```
git pull
```

Eso descarga y fusiona los cambios del repositorio remoto a mi rama actual.

12) ¿Qué es un fork de repositorio?

Un fork es una copia de un repositorio de otra persona que se guarda en mi cuenta de GitHub. Me sirve para hacer mis propias modificaciones sin afectar el proyecto original. Es muy usado para colaborar con otros proyectos o probar cosas nuevas a partir de otro código.

13) ¿Cómo crear un fork de un repositorio?

Para hacer un fork, voy al repositorio original en GitHub y hago clic en el botón "Fork" (que aparece arriba a la derecha). Automáticamente se crea una copia en mi cuenta, y desde ahí ya puedo clonarlo, hacer cambios, subir commits, etc.

14. ¿Cómo se envía una solicitud de extracción (*pull request*) a un repositorio en GitHub?

1. Forkeá o cloná el repositorio.

2. Creá una nueva rama y hacé tus cambios.
3. Subí la rama a tu repositorio.
4. En GitHub, hacé clic en "Compare & pull request".
5. Agregá un mensaje y descripción, y enviá la solicitud.

15. ¿Cómo se acepta una solicitud de extracción en GitHub?

1. Ingresá al repositorio.
2. Abrí la pestaña "Pull requests".
3. Seleccioná la solicitud pendiente.
4. Revisá los cambios.
5. Hacé clic en "Merge pull request" y luego en "Confirm merge".

16. ¿Qué es una etiqueta (*tag*) en Git?

Es una referencia que marca un punto específico en la historia del repositorio, normalmente usada para señalar versiones (ej: v1.0).

17. ¿Cómo se crea una etiqueta en Git?

```
git tag v1.0
```

O con un mensaje:

```
git tag -a v1.0 -m "Primera versión estable"
```

18. ¿Cómo se envía una etiqueta a GitHub?

```
git push origin v1.0
```

O para enviar todas:

```
git push origin --tags
```

19. ¿Qué es el historial de Git?

Es el registro de todos los *commits* realizados en el repositorio, mostrando qué se cambió, cuándo y por quién.

20. ¿Cómo se visualiza el historial de Git?

`git log`

O de forma resumida:

`git log --oneline`

21. ¿Cómo se realiza una búsqueda en el historial de Git?

Usá:

`git log --grep="palabra clave"`

Para buscar por autor:

`git log --author="Nombre"`

22. ¿Cómo se borra el historial de Git?

No se recomienda, pero se puede sobrescribir el historial con:

`git checkout --orphan nueva-rama`

`git add .`

`git commit -m "Nuevo inicio"`

`git push --force origin nueva-rama`

Esto elimina el historial anterior del repositorio remoto.

23. ¿Qué es un repositorio privado en GitHub?

Es un repositorio al que solo tienen acceso el creador y las personas que él autorice.

24. ¿Cómo se crea un repositorio privado en GitHub?

1. Ingresá a GitHub.
2. Hacé clic en "New repository".
3. Completá el nombre y descripción.
4. Seleccioná "Private".
5. Hacé clic en "Create repository".

25. ¿Cómo se invita a alguien a colaborar en un repositorio privado en GitHub?

1. Entrá al repositorio.
2. Hacé clic en "Settings" > "Collaborators".

3. Agregá el nombre de usuario.

4. Enviá la invitación.

26. ¿Qué es un repositorio público en GitHub?

Es un repositorio visible para cualquier persona, que puede clonarlo, observarlo o hacer *pull requests*.

27. ¿Cómo se crea un repositorio público en GitHub?

1. Ingresá a GitHub.

2. Hacé clic en "New repository".

3. Completá los datos y seleccioná "Public".

4. Hacé clic en "Create repository".

28. ¿Cómo se comparte un repositorio público en GitHub?

Simplemente copió la URL del repositorio desde la barra de direcciones o haciendo clic en el botón "Code" > "HTTPS", y compartila.