

1 - ¿Qué es GitHub?

GitHub es una plataforma en línea que permite a los desarrolladores almacenar, administrar y colaborar en proyectos de software utilizando el sistema de control de versiones Git.

2 - ¿Cómo crear un repositorio en GitHub?

En github, luego de iniciar sesión, haz clic en 'New', asigna un nombre al repositorio y luego haz clic en 'Create repository'.

3 - ¿Cómo crear una rama en Git?

Podemos crear una rama de Git, utilizamos el comando: `"git branch new_branch"`

4 - ¿Cómo cambiar a una rama en Git?

Para cambiar a una rama en Git, usa el comando:

`"git checkout new_branch"`

5 - ¿Cómo fusionar ramas en Git?

Para fusionar ramas en Git, usa el comando `"git merge"`. Por ejemplo, si estamos en release/latest y queremos fusionar las nuevas funcionalidades que están en main, nos posicionamos sobre release/latest y ejecutamos `git merge main`.

6 - ¿Cómo crear un commit en Git?

Para crear un commit de git primero hay que tener cambios en los archivos de nuestro repositorio, una vez estén hechos los cambios podemos optar por `"git add ."` que agregar todos los archivos, en cambio si solo quieres agregar un archivo específico ponemos `"git add readme.md"`, una vez hecho eso tenemos que utilizar el comando `"git commit -m 'feat: new readme'"` para guardar los cambios en la branch.

7 - ¿Cómo enviar un commit a GitHub?

Enviamos el "commit" a GitHub con el comando `"git push origin <branch>"`, en el

que branch puede ser MAIN / MASTER / RELEASE / ETC ...

8 - ¿Qué es un repositorio remoto?

Un repositorio remoto es una versión de tu proyecto Git alojada en un servidor en línea (como GitHub o GitLab). Permite compartir código, colaborar con otros y tener copias de seguridad de tu trabajo.

9 - ¿Cómo agregar un repositorio remoto a Git?

Para agregar un repositorio remoto a Git, utiliza el comando “git remote add” por ejemplo quedaría “git remote add origin <URL>”, “origin” es el nombre que le das al repositorio remoto por convención, se usa siempre “origin” y URL puede ser un repositorio de GitHub, GitLab, etc ...

10 - ¿Cómo empujar cambios a un repositorio remoto?

Para “empujar” que en realidad se usa mas el termino “push”, utilizamos el comando “git push” que completo quedaria algo asi “git push origin <branch>”, que origin es por cómo nombramos el repositorio remoto (explicado en la anterior respuesta), si le ponemos “remotex” tendría que ser “git push remotex <branch>”.

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

Para “tirar” que en realidad se usa más el término “pull” o “fetch”, tenemos dos caminos para llegar a la respuesta correcta.

Tenemos por un lado “git pull” que quedaria algo asi “git pull origin <branch>”, este comando va a “tirar” todos los cambios de la rama remota y los va a fusionar con la rama local, si es la misma la fusionará automáticamente, si es distinta (o si la rama local está desfasada de la remota) el IDE que usemos nos dirá que tenemos “conflictos” a solucionar.

Por el otro tenemos “git fetch” que quedaria algo asi “git fetch origin <branch>”,

que si bien sirve para traer cambios pero no los fusiona automáticamente, esto sirve por si tenemos alguna BRANCH en el remoto que no la estamos viendo en el local, sirve más cuando se trabaja en conjunto con otros colaboradores.

12 - ¿Qué es un fork de repositorio?

Un "fork" de un repositorio es una copia personal de un repositorio ajeno en tu propia cuenta de GitHub. Te permite modificar el código original sin afectar al repositorio principal.

13 - ¿Cómo crear un fork de un repositorio?

Para crear un "fork", primero accedemos al repositorio que queremos copiar, por ejemplo este repositorio, luego tocamos en el botón que dice "Fork", le ponemos un nombre y le damos a "Create fork".

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

Para enviar un "pull request" a un repositorio, primero modifica los archivos deseados. Luego, al subir los cambios a GitHub, selecciona la opción para crear un pull request. Se te pedirá confirmar la creación del PR y describir los cambios realizados.

15 - ¿Cómo aceptar una solicitud de extracción?

En GitHub una vez creada la PR solo debemos de tocar en "Merge Pull Request"

16 - ¿Qué es una etiqueta en Git?

En Git, una etiqueta (tag) es una instantánea que se utiliza para marcar puntos específicos en la historia de un repositorio.

17 - ¿Cómo crear una etiqueta en Git?

Para crear una etiqueta en Git tenemos que ejecutar el comando "git tag <nombre de etiqueta>".

18 - ¿Cómo enviar una etiqueta a GitHub?

Para enviar una etiqueta a GitHub, usa el comando “git push origin <nombre etiqueta>”. Si quieres enviar todas las etiquetas a la vez, usa git push origin --tags.

19 - ¿Qué es un historial de Git?

El historial de Git es un registro cronológico de todos los cambios realizados en un repositorio, permitiendo rastrear la evolución del proyecto, revertir a versiones anteriores y colaborar eficientemente.

20 - ¿Cómo ver el historial de Git?

Para ver el historial de Git, usa el comando “git log”. Este comando muestra una lista de los commits realizados, junto con información como el autor, la fecha y el mensaje del commit.

21 - ¿Cómo buscar en el historial de Git?

Para buscar en el historial de Git, puedes usar el comando git log con diferentes opciones de filtrado. Aquí tienes algunas formas breves:

Por palabras clave:

git log --grep="palabra clave": Busca commits que contengan la "palabra clave" en su mensaje.

Por autor:

git log --author="nombre": Filtra commits realizados por un autor específico.

Por fecha:

git log --since="fecha" o git log --until="fecha": Filtra commits por rango de fechas.

Por contenido:

git log -S"texto": busca cambios que han agregado o quitado el "texto".

Por Archivo:

`git log <nombre_archivo>`: Muestra solo los commit que afectaron el archivo determinado.

Combinando filtros:

Puedes combinar estas opciones para realizar búsquedas más específicas.

22 - ¿Cómo borrar el historial de Git?

El historial de Git se puede modificar, pero no se "elimina" en el sentido tradicional. En realidad, se reescribe, creando un nuevo historial que reemplaza al anterior. Por ejemplo “`git rebase -i`” edita commits locales.

23 - ¿Qué es un repositorio privado en GitHub?

Un repositorio privado en GitHub es lo mismo que el local no mas, con la característica de que solo las personas a las que invites explícitamente pueden acceder a él. A diferencia de los repositorios públicos, donde cualquiera puede ver el código, los repositorios privados ofrecen privacidad y control sobre quién puede ver y contribuir al proyecto.

24 - ¿Cómo crear un repositorio privado en GitHub?

Al crear el repositorio solo elegimos la opción que sea privado.

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

Debemos ir a Settings > Collaborators > Add people

ahí buscamos por username, full name o email

26 - ¿Qué es un repositorio público en GitHub?

Un repositorio público en GitHub es un espacio de almacenamiento en línea para código y archivos, accesible para cualquier usuario de la plataforma. Esto significa que cualquiera puede ver, clonar y, en algunos casos, contribuir al proyecto. Es ideal para proyectos de código abierto y para compartir trabajo con la comunidad.

27 - ¿Cómo crear un repositorio público en GitHub?

Al crear el repositorio seleccionamos la opción "Public"

28 - ¿Cómo compartir un repositorio público en GitHub?

Simplemente compartiendo el link del repositorio.

CONSIGNA 2:

<https://github.com/FrancoPetrozzelli/TP2-Ejercicio2>

COSIGNA 3: <https://github.com/FrancoPetrozzelli/conflict-exercise>