

GitHub



¿Git y GitHub son lo mismo?



¿Git y GitHub son lo mismo?

Git es un control de versiones.

GitHub es un servicio de alojamiento en la nube de código fuente basado en el sistema de control de versiones que git ofrece para manejar repositorios.



¿GitHub es único?



¿GitHub es único?

No, puesto que existen otros servicios de alojamiento de código fuente como:

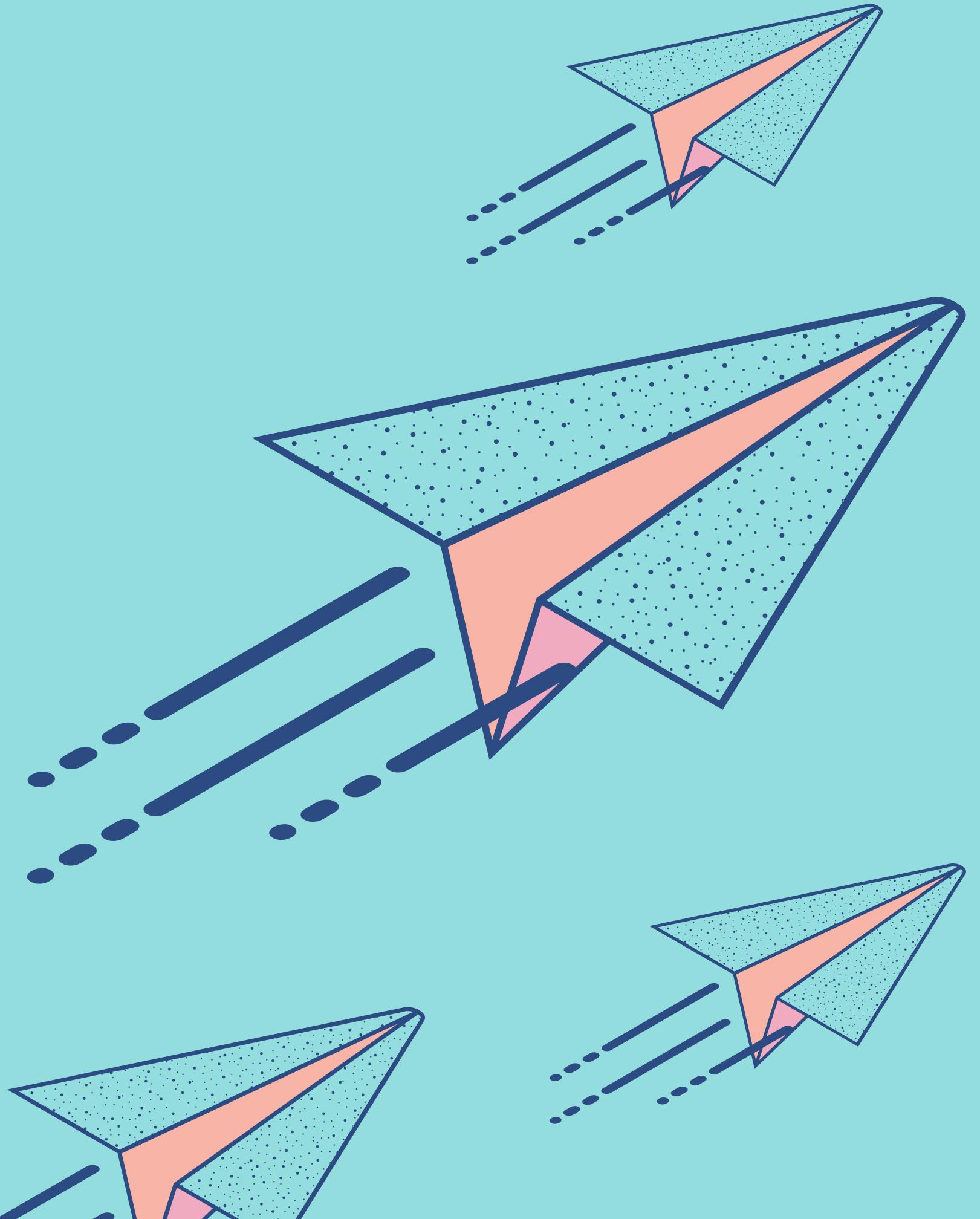
Bitbucket se centra en repositorios remotos, creado por Atlassian.

GitLab basado en la nube y una plataforma DevOps que facilita a los desarrolladores probar, monitorear e implementar código.



Repositorios remotos

Los repositorios remotos son repositorios que están hospedados en un servidor y que servirá de punto de sincronización entre diferentes repositorios locales.



Navegando por GitHub

Let's go ...

Mi perfil y mis repositorios

Repository > Code

Buscar perfiles, repositorios y otros

Repository > Pull Requests

Projects

Repository > Actions

Organizations

Repository > Projects

Navegando por GitHub

Let's go ...

Repository > Wiki

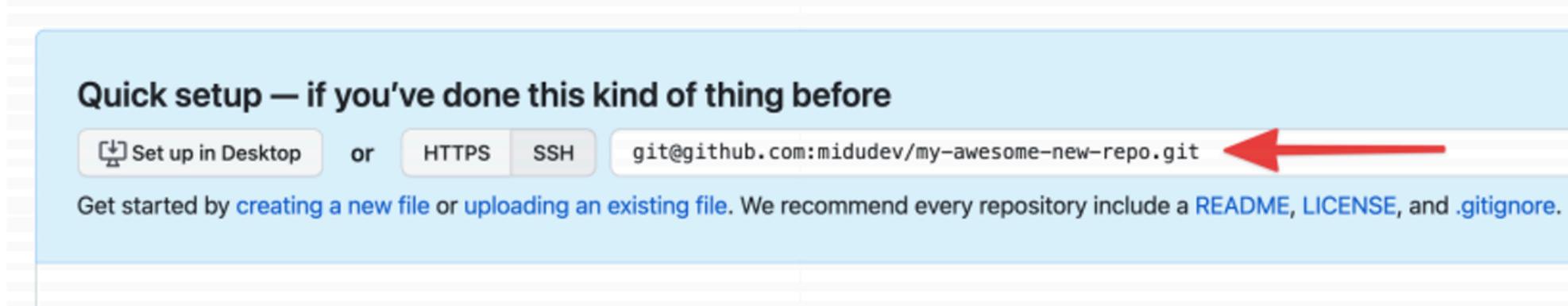
Repository > Security

Repository > Insights

Repository > Settings

Enlazar un repositorio local con un repositorio remoto

Como alias podemos usar cualquier nombre que queramos, pero por defecto usamos **origin** para indicar que el repositorio remoto que estamos sincronizando es el remoto principal.



Cada vez más, GitHub está utilizando las direcciones SSH por defecto. Por ello, es buena idea que siempre intentes utilizar estas

De esta forma, para añadir un repositorio remoto a nuestro repositorio local, tenemos que ejecutar:

```
# añadimos como origin la dirección ssh del repositorio remoto  
$ git remote add origin git@github.com:midudev/my-awesome-new-repo.git
```

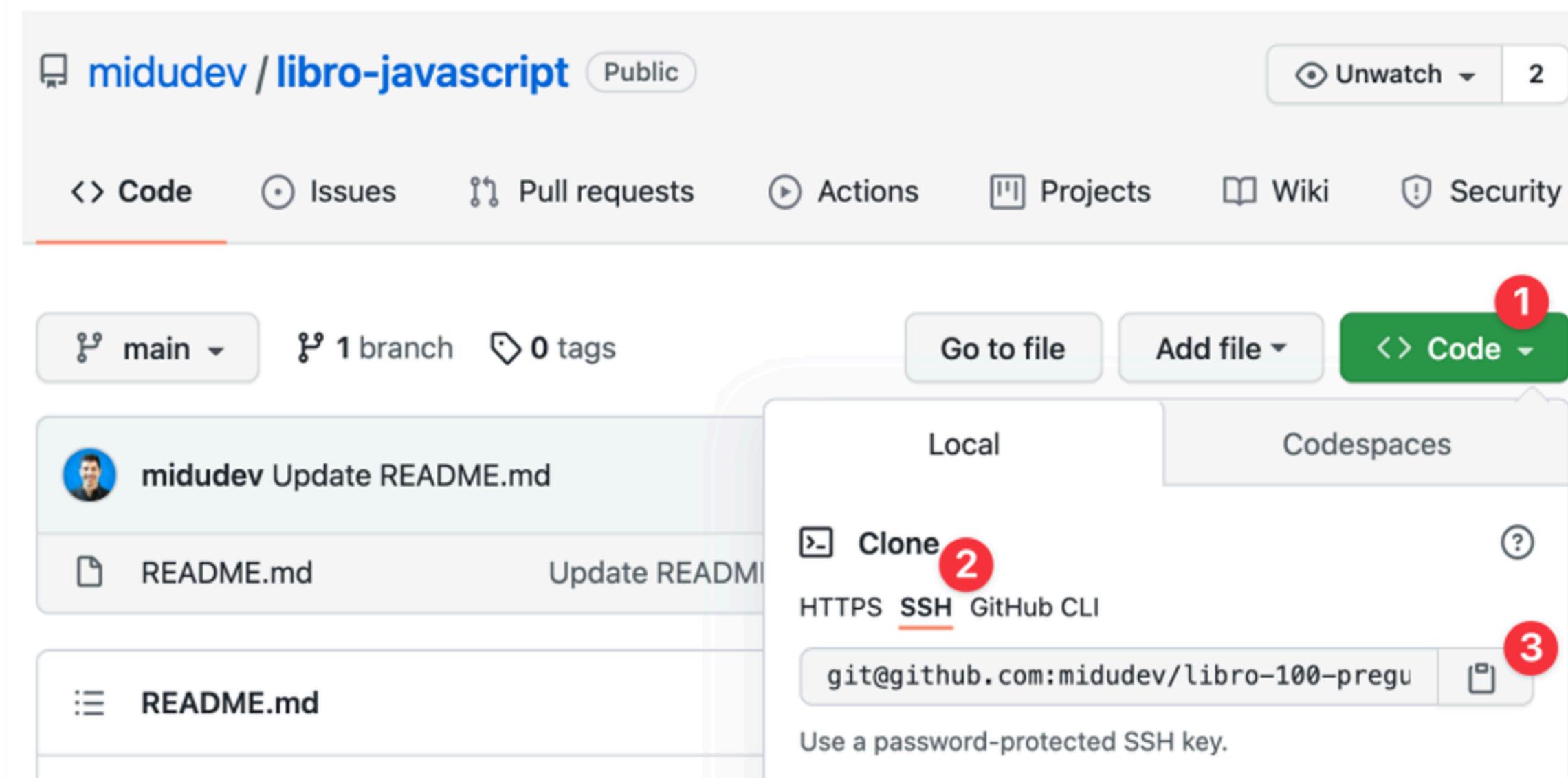
Ten en cuenta que un repositorio local, en realidad, puede tener enlazados tantos **repositorios remotos como queramos**. Normalmente sólo será uno, como en este caso, pero más adelante veremos que es útil tener también otros repositorios

Generar key SSH

- Listamos las llaves SSH que ya tenemos
 - \$ ls -al ~/.ssh
- Crear key SSH
 - ssh-keygen -t rsa -b 4096 -C "tu.email@gmail.com"
 - press enter
 - passphrase
- Poner en marcha nuestro key SSH
 - eval "\$(ssh-agent -s)"
- Añadir key SSH
 - ssh-add ~/.ssh/id_rsa
- Copia en portapapeles
 - clip < ~/.ssh/id_rsa.pub
- Pegar en la sección “key”
 - <https://github.com/settings/ssh/new>

Clonando repositorio

Para clonar un repositorio remoto necesitamos saber su dirección. Esta dirección puede ser su dirección *HTTPS* o *SSH*. Normalmente, la forma preferida, debería ser usando *SSH*.



Primero pulsas el botón Code. En el diálogo, selecciona SSH si no está ya activado. Después, pulsa el icono del lateral para que se copie la dirección en tu portapapeles

Clonando repositorio

```
# Para clonar un repositorio remoto con la dirección SSH  
$ git clone git@github.com:midudev/libro-javascript.git
```

```
Cloning into 'libro-javascript'...  
remote: Enumerating objects: 12, done.  
remote: Counting objects: 100% (12/12), done.  
remote: Compressing objects: 100% (7/7), done.  
remote: Total 12 (delta 2), reused 3 (delta 0), pack-reused 0  
Receiving objects: 100% (12/12), done.  
Resolving deltas: 100% (2/2), done.
```

```
# Para clonar un repositorio remoto con la dirección HTTPS  
$ git clone https://github.com/midudev/libro-javascript.git
```

Escribiendo en el repositorio remoto

- El primer parámetro sería el alias del repositorio remoto.
- El segundo sería el nombre de la rama sobre la que queremos enviar los cambios.

Por ejemplo, para enviar nuestros *commits* al repositorio remoto *origin* y la rama *main*, deberíamos ejecutar lo siguiente:

```
# subimos los cambios del repositorio local  
# al repositorio remoto con alias origin y a la rama main  
$ git push origin main
```

Esto hará que nuestros cambios locales se reflejen en el repositorio remoto y otras personas podrán verlos y sincronizar sus repositorios locales para trabajar con ellos.

Más adelante veremos cómo podemos enviar una rama distinta a la principal. Por ahora no nos preocuparemos por ello.

Creando una rama remota

```
# Creamos una rama en el repositorio local
$ git switch -c website
Switched to a new branch 'website'

# Enviar la rama a nuestro repositorio remoto
# para ello usamos git push:
# git push <alias-repositorio> <rama>
$ git push origin website

Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'website' on GitHub by visiting:
remote:     https://github.com/midudev/manual-git/pull/new/website
remote:
To github.com:midudev/manual-git.git
 * [new branch]      website -> website

# Ten en cuenta que si intentas enviar una rama
# que no existe en el repositorio local
# tendrás un error:
$ git push origin rama-que-no-existe

error: src refspec rama-que-no-existe does not match any
error: failed to push some refs
```

Ya tenemos nuestra rama en el repositorio remoto. Ahora, podemos empezar a crear commits en nuestro repositorio local y enviarlos al repositorio remoto a la rama que hemos creado.

Eliminar ramas de mi repositorio local que ya no se usan

```
# ejecutamos el mismo comando que antes  
# pero sin la opción --dry-run  
$ git remote prune origin
```

```
Pruning origin  
URL: git@github.com:midudev/midu.dev.git  
* [pruned] origin/feat/add-lighthouse-ci  
* [pruned] origin/imgbot  
* [pruned] origin/newsletter
```

Comandos

Para sincronizar con un repositorio remoto, sincronizar ramas, bajar cambios y más.

`git remote add origin <url>`

`git push origin <rama>`

`git remote -v`

`git fetch`

`git remote prune origin`

`git clone <url_repositorio>`

`git branch -a`

`git switch <rama_remota>`