

<https://bluuweb.github.io/tutorial-github/guia/>

Git

// Iniciar un nuevo repositorio

// Crear la carpeta oculta .git

git init

// Ver que archivos no han sido registrados

git status

git status -s // Solo muestra los archivos modificados

git status -s -b // Vemos información de la rama maestra

git status -sb //Hace lo mismo que el comando anterior

// Agregar todos los archivos para que esté pendiente de los cambios

git add .

// Crear commit (fotografía del proyecto en ese momento)

git commit -m "primer commit"

git commit -am "más comandos agregados" // Realiza el add . y
commit más mensaje al mismo tiempo

git commit --amend // Para editar un commit, como por ej: el mensaje

// Muestra la lista de commit del mas reciente al más antiguo

git log

git log --oneline // Muestra en una línea los commit realizados

git log --oneline --decorate --all --graph // Muestra en una línea los
commit realizados pero más elegante

// Viajamos al commit en específico f52f3da y eliminamos los cambios
futuros

git reset --hard f52f3da

Para deshacer cambio cuando se a enviado a stage con add, utilizar

git reset HEAD index.html //HEAD es el ultimo commit de master

git reset --soft 903e327 //Para regresar a un commit y devolverlo al
stage

//Para deshacer los cambio no se ha hecho, ni add, ni commit utiliza

```
git checkout . //Para deshacer todos los cambios
git checkout -- index.html
```

Es un sistema de versiones distribuido.

```
git remote //Agregar ordenadores remotos
git remote add origin https://github.com/bluuweb/tutorial-github.git
git remote -v
```

```
git push //Enviar código remoto
git pull -u nombre master //Recuperar datos remoto
git fetch
git push --tags
```

```
git remote show origin. //Inspecting a Remote
```

// Crear un tags

```
git tag versionAlpha -m "versión alpha"
git tag // Listar tags
git tag -d nombreTags // Borrar tags
git tag -a nombreTag f52f3da -m "version alpha" // Hacer una versión
en un commit anterior ej: f52f3da
git show nombreTag // Mostrar información del tag
```

//Crea una nueva rama

```
git branch <nombreRama>
```

//Nos muestra en que rama estamos

```
git branch
```

//Nos movemos a la nueva rama

```
git checkout <nombreRama>
```

//Para hacer merge mover a la rama master

```
git merge <nombreRama>
```

//Eliminar una rama

```
git branch -d <nombreRama>
```