

| Working area  | Stagin area   | Local repository  | REMOTE REPOSITORY  |
|---|---|---|--|
| Untracked   | Tracked   | Confirmado  | REMOTO   |
| Ctrol + S (guardar)                                 |   |   |  |
| <b>Git add</b> para pasar a tracked en staging area | <b>Git commit -m "..."</b> para confirmar el cambio |   |  |
|   |   | <b>Git push origin main</b> para subir cambios al repo remoto (1° PULL 2° PUSH) |  |
| <b>Git status</b> me muestra ESTA COLUMNA           |   |   | <b>Git pull origin main</b> para traer cambios del repo remoto al repo local (1° PULL 2° PUSH) |

### 1) Creo archivos en la terminal, en mi compu **Working area (Untracked)**

```
pwd
mkdir "NuevaCarpeta"
cd "NuevaCarpeta"

touch nuevoarchivo.xx
mkdir "OtraCarpeta"
mv nuevoarchivo.xx ../OtraCarpeta o bien lo renombro: nuevonombre.yy
(mv -r para carpetas)
rm nuevoarchivo.xx
(rm -r) (rmdir) (para carpetas)
nano nombredelarchivoeditar
```

```
ls (visibles)
ls -a (visibles + ocultos)
ls -R (visibles + ocultos + contenido carpetas)
```

```
.git es mi repo
rm -r .git (para eliminarlo)
```

```
rm * (borra todo)
git checkout (los trae de vuelta)
```

### 2) Creo un repo remoto en github (+ verde)

En el repositorio q creé busco el link en **CODE** y copio la dirección, por ejemplo:  
<https://github.com/GiRoldan/PrimerExamen1.git>

### 3) Creo un repo local en mi working area

```
git init
git config user.name "..." (ya los hice con config -global)
git config user.email "..." (ya los hice con config -global)
```

### 4) Añado los ficheros a la **Staging Area (Tracked)**

```
git add nombredelarchivo / . (para todos)

git status
```

### 5) Confirmo los cambios, los archivos pasan al **repo local**

```
git commit -m "..."

git log (historial de commits)
:q (para salir)
```

### 6) Subo los cambios a github con la ruta del **repo remoto** del punto 2

**git remote add origin** <https://TOKEN@github.com/GiRoldan/PrimerExamenII.git>

o bien:

**git remote add origin** <https://github.com/GiRoldan/PrimerExamenII.git>

## 7) Subo los cambios a github

**git push origin main** (la 1º vez pongo todo, pero luego ya puedo hacer directamente git push)

## 8) Creo otra rama (tiene q estar el main al final en la terminal)

**git branch** "NombredelaRama"

## 9) Para ir a esa nueva rama

**git checkout** "NombredelaRama"

## 10) En la nueva rama repito:

- el paso 1 de crear archivos,
- el paso 4 de add
- el paaso 5 de commit

## 11) Para subir los archivos a esa nueva rama

**git push -u origin** "NombredelaRama"

el -u crea una rama que no estaba en el remoto, una rama que he creado en el repo local, con -u subo esa rama al remoto

## 12) NOTA:

Para clonar uso el link del punto 2

**git clone** <https://github.com/GiRoldan/PrimerExamenII.git>

Se clona no solo el repositorio, se clona todo, la carpeta .git, los commits, etc, todo

## 13) Comandos Git

**mkdir** "Nombre de la carpeta" // Crea un nuevo directorio

**cd** "Nombre de la carpeta" // Para ir a esa nueva carpeta

**touch** archivo.extension // Para crear un nuevo archivo

**mv** archivo.extension ./NuevaDirección // Para mover el archivo/carpeta de lugar

**mv** "Nombre de la carpeta" "Nuevo nombre" // Modifica el nombre del archivo/carpeta

**rm** archivo.ext // Elimina un archivo

**rmdir** "Nombre de la carpeta" // Borra la carpeta con un solo comando

**rm -r** "Directorio" // Eliminación forzada

**ls** // Muestra el listado de archivos visibles

**ls -a** // Muestra el listado de archivos visbles e incluso ocultos

**ls -R** // Muestra el listado de archivos y los archivos que éstos contienen

**cp -r** "Nombre de la carpeta" "Otra carpeta" // Copia una carpeta y todo su contenido en otra

**git init** // crea un repositorio

**git config user.name** "Nombre de Usuario" // Nombre de Usuario de mi repositorio

**git config user.email** "Email de Usuario" // Email de Usuario de mi repositorio

**git config user.name --global** "Nombre de Usuario" // Nombre de Usuario de todos mis repositorios

**git config user.email --global** "Email de Usuario" // Email de Usuario de todos mis repositorios

**git add** "Nombre del Archivo".extensión // Agregar un archivo pendiente para commitear

`git add .` // Agregar todos los archivos pendientes para commitear

`git commit -m "Mensaje conciso del commit"` // Para commitear archivos

`git status` // Estado de mis archivos respecto al repositorio

`git log` // Lista los commits creados

`git remote add origin URLdelrepositorioengithub` // Para vincular el repositorio local con el repositorio remoto

`git remote -v` // Para chequear con qué repositorio remoto está vinculado mi repositorio local

`git clone https://token@elrestodelarutadelrepositorioremoto` // Para clonar el repositorio remoto en una pc, se hace una sola vez, se clonan todos los archivos

`git branch -M` // Cambia el nombre de la rama de master a main

`git push origin main` // Para enviar archivos de mi repositorio local al remoto, a la rama principal

`git push` // Para enviar archivos al repositorio de la rama que ya elegí

`git pull origin main` // Para actualizar mi repositorio local, de la rama principal

`git pull` // Para actualizar mi repositorio local, de la rama que ya elegí

`git branch "Nombre de una nueva rama"` // Para crear una nueva rama

`git branch -D "Nombre de la rama"` // Elimina la rama si aún no tiene cambios fusionados con la rama main

`git checkout "Nombre de la rama"` // Para movernos de una rama a otra, es directo si no tenemos cambios, sino debemos previamente eliminarlos (deshaciendo los cambios) o confirmándolos (`git commit -m "Mensaje"`)

`git push origin "Nueva rama"` // Para subir los cambios a la "Nueva rama", debo estar trabajando en esa rama

`git pull origin "Nueva rama"` // Para traer los cambios de la "Nueva rama"

`git rm -cached` archivo // Se utiliza para dejar de dar seguimiento a cualquier archivo

`git rm "Nombre del archivo".extensión` // Elimina un archivo desde un repositorio Git