

Git básico

¿Qué es git?

- Git es una herramienta de control de versiones.
- Una herramienta de control de versiones permite guardar copias de seguridad de cada etapa del proceso.
- Esto, unido a un repositorio web, nos permite disponer de copias de seguridad en la nube y distribuir nuestro código fácilmente en su última versión.
- Git es muy complejo, solo veremos lo más más básico.

¿Cómo usar git?

THIS IS GIT. IT TRACKS COLLABORATIVE WORK
ON PROJECTS THROUGH A BEAUTIFUL
DISTRIBUTED GRAPH THEORY TREE MODEL.

COOL. HOW DO WE USE IT?

NO IDEA. JUST MEMORIZE THESE SHELL
COMMANDS AND TYPE THEM TO SYNC UP.
IF YOU GET ERRORS, SAVE YOUR WORK
ELSEWHERE, DELETE THE PROJECT,
AND DOWNLOAD A FRESH COPY.



Repositorio

- Para crear un repositorio en una carpeta:

```
git init
```

- Para copiar un repositorio en la nube:

```
git clone [url]
```

- Para añadir una correspondencia con un repositorio en la nube:

```
git remote add origin [url]
```

Commit

- Un commit es un punto de restauración. Para realizarlo primero hay que añadir los archivos al commit:

```
git add [archivo]
```

- Para añadir todos los archivos de la carpeta al commit:

```
git add *
```

- Para realizar el commit:

```
git commit -m [mensaje de confirmación]
```

- Para ver si todo está confirmado:

```
git status
```

Commit

- A veces no queremos añadir cambios sino eliminar archivos del repositorio:

```
git rm [archivo]
```

- O indicar que el nombre/ruta cambió:

```
git mv [nombre1/ruta1] [nombre2/ruta2]
```

Restaurar punto

- Para ver los commit que cambian un fichero:
`git log [archivo]`
- Para ver los commit de todo el repositorio:
`git log`
`git log --oneline` → Sin detalles
- Para restaurar al último commit:
`git checkout HEAD [archivo]`
`git checkout HEAD *` → Para todo el repo.
- Para restaurar a un commit concreto:
`git checkout [CODIGO] [archivo]/[*]`

Remoto

- Para traer cambios desde el remoto:
`git pull origin`
- Para subir cambios al remoto
`git push origin`