

## MEMORIA COMANDOS PRÁCTICA 1

A lo largo de esta memoria explicaremos brevemente los comandos utilizados en la práctica e incluiremos una captura del código introducido en la terminal.

### 1.Git Clone

```
● @LCS-26 → /workspaces/ci-cd (main) $ git clone https://github.com/gitt-3-pat/p1
Cloning into 'p1'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 5 (from 1)
Receiving objects: 100% (6/6), done.
○ @LCS-26 → /workspaces/ci-cd (main) $ █
```

Este comando nos permite clonar un repositorio a partir de su URL en el nuestro.

### 2.Git Status

```
● @LCS-26 → /workspaces/ci-cd (main) $ git clone https://github.com/gitt-3-pat/p1
Cloning into 'p1'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 5 (from 1)
Receiving objects: 100% (6/6), done.
○ @LCS-26 → /workspaces/ci-cd (main) $ █
```

Git Status muestra el estado actual del repositorio local. Incluye cambios que aún no se han guardado (staged o unstaged), la rama en la que estás trabajando o archivos nuevos que no están siendo rastreados.

### 3.Git Add .

```
● @LCS-26 → /workspaces/ci-cd (main) $ git clone https://github.com/gitt-3-pat/p1
Cloning into 'p1'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 5 (from 1)
Receiving objects: 100% (6/6), done.
○ @LCS-26 → /workspaces/ci-cd (main) $ █
```

Git Add añade archivos al área de preparación (staging area). Esto prepara a los archivos para ser incluidos en el próximo commit.

### 4.Git Commit

```
● @LCS-26 → /workspaces/ci-cd (main) $ git clone https://github.com/gitt-3-pat/p1
Cloning into 'p1'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 5 (from 1)
Receiving objects: 100% (6/6), done.
○ @LCS-26 → /workspaces/ci-cd (main) $ █
```

Este comando guarda los cambios en el historial del repositorio. Un commit representa un punto de control del desarrollo.

## 5. Git Push

```
● @LCS-26 → /workspaces/ci-cd (main) $ git clone https://github.com/gitt-3-pat/p1
Cloning into 'p1'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 5 (from 1)
Receiving objects: 100% (6/6), done.
○ @LCS-26 → /workspaces/ci-cd (main) $ █
```

Git Push sincroniza los cambios y envía los commits locales al repositorio remoto.

## 6. Git Checkout

```
● @LCS-26 → /workspaces/ci-cd (main) $ git clone https://github.com/gitt-3-pat/p1
Cloning into 'p1'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 5 (from 1)
Receiving objects: 100% (6/6), done.
○ @LCS-26 → /workspaces/ci-cd (main) $ █
```

Git Checkout permite restaurar archivos o cambiar de rama. Para crear una nueva rama utilizamos flag -b.