



# Encuentro Práctico Git 1

Objetivo: Utilizar los comandos de git para crear un repositorio y subirlos a GitHub

```
mkdir labgit  
cd labgit
```

```
git init  
inicializo el repo para que sea reconocido por git
```

```
ll ( o ll -a)  
veo que se agrega la carpeta .git
```

```
git status  
reviso el estado
```

```
vim code.txt  
archivo: code.txt
```

```
ll
```

## **DIPLOMATURA DEVOPS - GUIA PRACTICA**

```
git status
```

para ver los untraked

```
git add code.txt
```

lo meto en el area de stage (index) para luego comitearlo

```
git status
```

verifico que este en "to committed"

```
git commit -m "primer commit"
```

verifico que obtengo un id del commit

```
git status
```

veo que no hay pendientes

```
vim code.txt
```

le pongo "segunda modificacion"

```
git status
```

veo q el archivo esta en "modified"

```
git add code.txt
```

lo paso al area de stage

```
git commit -m "Segundo commit - modificacion a code"
```

```
git status
```

veo que no hay nada pendiente en el working clean

```
git log -3
```

veo el historial de commits

```
vim code.txt
```

le pongo "tercera modificacion"

```
git status
```

veo si hay cambios

```
git diff
```

veo los cambios en el working area

```
git commit add code.txt
```

## **DIPLOMATURA DEVOPS - GUIA PRACTICA**

```
git commit -m "tercer commit al archivo code"
```

### **-> Manejo de Ramas**

```
git branch  
  consulto en que rama estoy
```

```
git checkout -b develop  
  creo rama develop como copia del master
```

```
ll
```

```
vim code2.txt  
  le pongo "archivo: code2.txt"
```

```
git status  
  verifico estado
```

```
git add code2.txt
```

```
git commit -m "Subo archivo code2 en rama dev"
```

```
git branch  
  reviso las ramas que tengo, puedo tener vs
```

```
ll  
veo los 2 archivos
```

```
git checkout master  
  me cambio a la rama master
```

```
ll  
veo que solo tengo 1 archivo
```

### **-> Actualizar rama master con lo de develop**

```
git branch  
  veo en q rama estoy
```

```
git log  
  veo los logs
```

```
git checkout develop  
  me cambio a la rama de develop
```

## **DIPLOMATURA DEVOPS - GUIA PRACTICA**

```
git log
veo que tengo un comit mas
```

```
git checkout master
vuelvo a la rama master
```

```
git merge develop
trae los cambios de develop a master
compara los historiales de commit y me los trae
```

```
ll
veo que ahora en rama master tengo el archivo code2.txt
```

```
git log
veo que no hay diferencias entre commit y merges...
```

entonces para diferenciarlos aparece los git-flow donde defino que los cambios a master deben venir por un pool request u otros esquemas de branches

### **-> Repositorio web**

Creo en mi cuenta de la web de github (usuario) el repositorio de nombre "mylab"

```
clono el repo de github
git clone https://github.com/marceamx/mylab.git
```

```
git remote -v
veo que no tiene datos
```

si es necesario configuro mis datos de acceso locales al repo (reemplazar usuario por el de uds)

```
git config user.name "marceamx"
git config user.email "marceamx@gmail.com"
```

```
git remote add origin https://github.com/marceamx/mylab.git
agrego el link al repo remoto desde mi local
```

```
git remote -v
veo los datos del repositorio remoto y su alias "origin"
```

## **DIPLOMATURA DEVOPS - GUIA PRACTICA**

git push origin master  
subo a mi repo remoto la rama master

ingreso a la web <https://github.com/usuario/labgit2> y veo de forma grafica mi repositorio

modifico en la web el archivo code2.txt y al commit como descripcion le pongo "cambio desde la web"

vuelvo a mi repo local y modifico el archivo y lo commiteo

vi code2.txt  
git add code2.txt  
git commit -m "Cambio Local"

ahora los repos estan desincronizados

git push origin master  
intento subir mi repo

no me deja por los cambios en remoto  
entonces tengo que bajar primero los datos remotos

git pull origin master  
me avisa del conflicto

git status  
veo que archivo tiene el conflicto

vi code2.txt  
modifico y dejo el texto (codigo) que quiero quede

git add code2.txt

git commit -m "Reparo conflicto"

git log  
veo como quedo el historial

git push origin master  
actualizo el repo remoto  
ahora si lo sube correctamente

