

## Contents

<b>1</b>	<b>Git</b>	<b>2</b>
1.1	Checkout . . . . .	2
1.1.1	Ir a una rama . . . . .	2
1.1.2	Ir a un commit en particular . . . . .	2
1.2	Switch . . . . .	2
1.2.1	Ir a una rama . . . . .	2
1.2.2	Crear una rama nueva . . . . .	2
1.3	Status . . . . .	2
1.3.1	Ver el estado general . . . . .	2
1.4	Log . . . . .	2
1.4.1	Ver el historial de commits . . . . .	2
1.4.2	Ver el ultimo commit . . . . .	2
1.4.3	Ver el grafo de commits . . . . .	3
1.4.4	Ver los cambios del commit . . . . .	3
1.4.5	Grepear los mensajes de commits . . . . .	3
1.4.6	Grepear los diffs de commits . . . . .	3
1.4.7	Diferencias de commits entre dos ramas . . . . .	3
1.5	Diff . . . . .	3
1.5.1	Ver diferencias entre ultimo commit y el estado actual	3
1.5.2	Ver diferencias entre ultimo commit y el estado actual de un archivo particular . . . . .	3
1.5.3	Ver diferencias de patches entre dos ramas . . . . .	3
1.5.4	Ver diferencias de patches entre dos ramas en un archivo especifico . . . . .	3
1.6	Show . . . . .	4
1.6.1	Mostrar el commit de un hash en particular . . . . .	4
1.7	Bisect . . . . .	4
1.7.1	Arranca el modo bisect . . . . .	4
1.7.2	Marcar el primer commit bueno . . . . .	4
1.7.3	Marcar el commit en el que estas como bueno . . . . .	4
1.7.4	Marcar el commit en el que estas como malo . . . . .	4
1.7.5	Finalizar el modo bisect . . . . .	4
1.8	Blame . . . . .	4
1.8.1	Ver linea a linea quien es el autor del ultimo commit .	4
1.9	Stash . . . . .	4
1.9.1	Guardar el estado de git sin necesidad de commitear .	5
1.9.2	Restaurar lo guardado . . . . .	5

# 1 Git

- <>: Representa una "variable". No hay que ponerlas.
- La gran mayoría de las opciones de un mismo comando se pueden combinar!

## 1.1 Checkout

### 1.1.1 Ir a una rama

```
git checkout <branch>
```

### 1.1.2 Ir a un commit en particular

```
git checkout <hash>
```

## 1.2 Switch

### 1.2.1 Ir a una rama

```
git switch <branch>
```

### 1.2.2 Crear una rama nueva

```
git switch -c <branch>
```

## 1.3 Status

### 1.3.1 Ver el estado general

```
git status
```

## 1.4 Log

### 1.4.1 Ver el historial de commits

```
git log
```

### 1.4.2 Ver el ultimo commit

```
git log -1
```

### 1.4.3 Ver el grafo de commits

Para recordarlo esta buena la regla mnemotécnica "A DOG".

```
git log --all --decorate --oneline --graph
```

### 1.4.4 Ver los cambios del commit

```
git log --patch
```

### 1.4.5 Grepear los mensajes de commits

```
git log --grep=<texto>
```

### 1.4.6 Grepear los diffs de commits

```
git log -G <texto> --patch
```

### 1.4.7 Diferencias de commits entre dos ramas

```
git log <branchX>..<branchY>
```

## 1.5 Diff

### 1.5.1 Ver diferencias entre ultimo commit y el estado actual

```
git diff
```

### 1.5.2 Ver diferencias entre ultimo commit y el estado actual de un archivo particular

```
git diff <archivo>
```

### 1.5.3 Ver diferencias de patches entre dos ramas

```
git diff <branchX>..<branchY>
```

### 1.5.4 Ver diferencias de patches entre dos ramas en un archivo especifico

```
git diff <branchX>..<branchY> -- <archivo>
```

## **1.6 Show**

### **1.6.1 Mostrar el commit de un hash en particular**

```
git show <hash>
```

## **1.7 Bisect**

Git de alto nivel!! Realiza una búsqueda binaria interactivamente, preguntandote en cada paso. Te permite encontrar el commit "que lo rompe" de una manera re piola.

### **1.7.1 Arranca el modo bisect**

```
git bisect start
```

### **1.7.2 Marcar el primer commit bueno**

```
git bisect good <hash>
```

### **1.7.3 Marcar el commit en el que estas como bueno**

```
git bisect good
```

### **1.7.4 Marcar el commit en el que estas como malo**

```
git bisect malo
```

### **1.7.5 Finalizar el modo bisect**

```
git bisect reset
```

## **1.8 Blame**

### **1.8.1 Ver linea a linea quien es el autor del ultimo commit**

```
git blame <archivo>
```

## **1.9 Stash**

Este esta re bueno. Si tenes cosas sin commitear y tenes que cambiar de rama o lo que sea esto te sirve para "guardar" tus cosas momentaneamente

### 1.9.1 Guardar el estado de git sin necesidad de commitear

```
git stash
```

### 1.9.2 Restaurar lo guardado

```
git stash pop
```