

## CHEATSHEET

### Crear

Clonar un repositorio existente

```
$ git clone https://github.com/MarthaCaro/Mochila_Martha_Caro.git
```

Crear un nuevo repositorio local

```
$ git init
```

### Configurar

Configurar la información del usuario utilizada en todos los repositorios locales

```
$ git config --global user.name "MarthaCaro"
```

```
$ git config --global user.email "m.carovila@gmail.com"
```

### Trabajando en escenarios de Git

Muestra los archivos modificados en el directorio de trabajo, próximos para realizarles un commit.

```
$ git status
```

Agrega los archivos para realizarles su próximo commit

```
$ git add archivo.txt
```

Agrega todos los archivos para realizarles su próximo commit

```
$ git add .
```

Captura y guarda los cambios realizados en el momento actual del proyecto o zona de montaje (staging area)

```
$ git commit -m "nombre del commit"
```

### Ramas y Fusiones

Crea una nueva rama en el commit actual

```
$ git Branch "nombre de la rama"
```

Sirve para movilizarte entre las ramas creadas con el git branch

```
$ git checkout
```

Te permite ver todos los commits en el historial del repositorio.

```
$ git log
```

Te permite unificar las líneas creadas por git Branch en una sola rama

```
$ git merge
```

Actualizar y borrar
Te permite subir los commits desde tu rama (branch) local en tu repositorio git local al repositorio remoto, no olvides realizar previamente los commits que necesites.
\$ git push origin ( nombre de la rama)
Sirve para extraer y descargar el contenido actualizado desde un repositorio remoto y actualizar al instante el repositorio local para reflejar dicho contenido.
\$ git pull
Borra el archivo del proyecto sin borrarlo del historial del versiones
\$ git rm