

CONFIGURAR GIT

Establece el nombre que estará anexado a las transacciones:

```
$ git config --global user.name "<user-name>"
```

Establece el email que estará anexado a las transacciones:

```
$ git config --global user.email "<user-email>"
```

Establece el editor vscode como editor predeterminado:

```
$ git config --global core.editor "code --wait"
```

`--wait` le indica a Git que debe esperar a cerrar el editor para continuar

Configuraciones a nivel de sistema, usuario y a nivel de repositorio

```
$ git config --system     👉 [path]/etc/gitconfig
```

```
$ git config --global     👉 ~/.gitconfig
```

OPERACIONES BÁSICAS EN LOCAL

Crea un nuevo repositorio local:

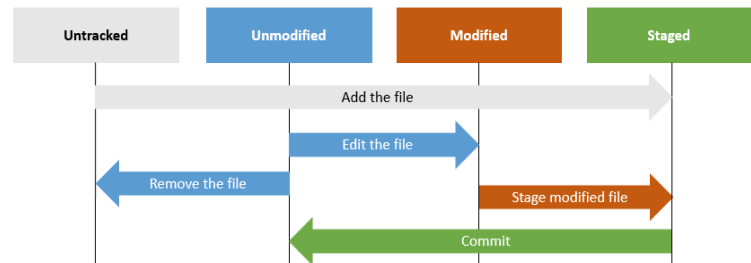
```
$ git init <path?>
```

Agrega un archivo al área de preparación (staging)

```
$ git add <file>
```

Crea una nuevo objeto de confirmación de los archivos:

```
$ git commit <path?> -m "message"
```



SINCRONIZAR CAMBIOS

Descarga los últimos cambios del origen remoto y rama específica:

```
$ git fetch origin/<branch-name>
```

Fusiona la rama remota con la rama local actual:

```
$ git merge origin/<branch-name>
```

HISTORIAL

Listar commit, con su ID en un gráfico de bifurcaciones:

```
$ git log --graph --oneline --decorate
```

```
* 9c3a2f1 (HEAD -> main, origin/main, tag: v1.2.0) chore(release): v1.2.0
| * 7b1d4e8 (feature/auth) feat(auth): agregar login con JWT
|/
| * a4f7b2d (origin/feature/ui) Merge branch 'feature/ui' into main
|/
| * d2e8a6c feat(ui): nuevo componente Header
| * b1c9f3a fix(ui): corregir layout responsive
```

CLONAR REPOSITORIOS

Clonar repositorio vía https:

```
$ git clone https://<host>/<user>/<repo-name>.git
```

Clonar una rama específica:

```
$ git clone --branch <branch-name> <remote-url>
```

Clonar repositorio vía ssh:

```
$ git clone git@github.com:<user>/<repo-name>.git
```

TRABAJAR CON RAMAS

Crear una nueva rama sin cambiarse a ella:

```
$ git branch <branch-name>
```

Crea y cambia a una nueva rama en un solo paso:

```
$ git checkout -b <branch-name>
```

Crea y cambia a una nueva rama basada en una rama remota:

```
$ git checkout -b <branch-name> <origin>/<branch>
```