

+

1. Configuración de git

```
$ git config --global user.name <nombre apellido>
```

Establece el nombre que quedará asociado a los commits y tags.

```
$ git config --global user.email <correo@correo.cl>
```

Establece el email que quedará asociado a los commits y etiquetas.

2. Inicializar un proyecto

```
$ git init <nombre-proyecto>
```

Crea un nuevo repositorio local, si se indica el <nombre-proyecto>, git creará un nuevo directorio con el nombre <nuevo-proyecto> e inicializará un repositorio dentro de este. Si no se especifica <nombre-proyecto>, se inicializará un nuevo repositorio en el directorio actual

```
$ git clone <url-proyecto>
```

Descarga un proyecto con todo su detalle desde un repositorio remoto

3. Uso cotidiano

```
$ git status
```

Indica el estado del directorio de trabajo. Las opciones que incluyen son archivos *new*, *straged*, y *modified*. Además indica el nombre de la rama, el commit actual y los cambios que tienen un commit pendiente.

```
$ git add <archivo>
```

agrega un archivo al “stage area”. Usar “.” en vez de la ruta del archivo para agregar todos los archivos con cambios del directorio actual, indicados en el árbol de directorio.

```
$ git diff <archivo>
```

muestra las diferencias entre los directorio de trabajo y el “staging area”

```
$ git checkout -- <archivo>
```

Descarta cambios en el directorio de trabajo

\$ git commit -m <mensaje>

Crea un nuevo commit de los cambios agregados al “staging area”. El commit debe tener un mensaje

\$ git rm <archivo>

remueve el archivo del directorio de trabajo y del “staging area”

\$ git stash

Deja los cambios actuales del directorio de trabajo en el stash para usarlos después.

\$ git stash pop

Borrar un stash específico de todos los stashes

4. Manejo de ramas

\$ git branch [-a]

Lista de todas las ramas locales del repositorio. Con -a : muestra todas las ramas, incluyendo las remotas.

\$ git branch <nombre-rama>

Crear una nueva rama, referenciando a la HEAD actual

\$ git checkout [-b] <nombre-rama>

Cambia el directorio de trabajo a la rama especificada. con [-b] : git creará la rama especificada en el caso de que no exista

\$ git merge <nombre-rama>

Une la rama especificada en la rama donde te encuentras actualmente

\$ git branch [-d] [-D] <nombre-rama>

Elimina la rama seleccionada, solo si ya fue unida con otra, -D en vez de -d fuerza el borrado

5. Sincronizando repositorios

```
$ git fetch <remote>
```

Recupera los cambios remotos, sin actualizar la rama trackeada

```
$ git pull <remote>
```

Recupera los cambios remotos y actualiza la rama actual (genera un merge de los cambios remotos con tus cambios locales).

```
$ git push
```

Sube los cambios locales al repositorio remoto

```
$ git push - <remote> <rama>
```

sube la rama local al repositorio remoto. Deja una copia en el canal de subida (upstream)