

Comandos Básicos de Git

José Canumán Ch.

1 Comandos Básicos de Git

1.1 git — version

Muestra la versión de Git. También se utiliza para verificar si Git está instalado o no.

1.2 git config

Establecer

```
1 git config --global user.name "Nombre Apellido"
2 git config --global user.email "micorreo@gmail.com"
```

Verificar

```
1 git config --global user.name
2 git config --global user.email
```

1.3 git init

Convierte un proyecto existente sin control de versiones (espacio de trabajo) en un repositorio de Git o crea un nuevo repositorio vacío de Git.

Ejecutar este comando crea un subdirectorio `.git`, que contiene todos los metadatos para el nuevo repositorio. Estos metadatos incluyen subdirectorios para objetos, referencias (refs) y archivos de plantilla. También se crea un archivo `HEAD` que apunta al commit actualmente revisado.

1.4 git clone

Descarga un repositorio Git existente a tu computadora local.

```
1 git clone -b nombre_de_la_rama <url_git>
```

El argumento `-b` te permite especificar una rama específica para clonar en lugar de la rama a la que apunta `HEAD` en el repositorio remoto, usualmente la rama `master`.

git init vs git clone: A un nivel alto, ambos pueden usarse para inicializar un nuevo repositorio Git. Sin embargo, `git clone` depende de `git init`. Internamente, `git clone` primero llama a `git init` para crear un nuevo repositorio. Luego, copia los datos del repositorio existente.

1.5 git status

Muestra:

- Rama actual
- Archivos que tienen diferencias entre el Espacio de trabajo ↔ Área de preparación (archivos nuevos no rastreados y cambios no preparados)
- Archivos que tienen diferencias entre el Área de preparación ↔ Repositorio Git local (cambios no confirmados)

1.6 git add

Añade cambios en el espacio de trabajo al área de preparación.

Espacio de trabajo → Área de preparación

1.7 git commit

Añade cambios en el área de preparación al repositorio Git local.

```
1 git commit
```

Área de preparación → Repositorio Git local

```
1 git commit -a
```

Espacio de trabajo → Repositorio Git local (Los archivos no rastreados no están incluidos, solo aquellos que han sido añadidos con `git add` en algún momento)

```
1 git commit -m 'mensaje del commit'
```

1.8 git push & git pull

Push: Añade cambios en el repositorio Git local al repositorio remoto.

```
1 git push <remoto> <local>
```

Repositorio Git local → Repositorio Git remoto

Pull: Actualiza el repositorio Git local desde el repositorio remoto correspondiente.

```
1 git pull <remoto> <local>
```

Repositorio Git local ← Repositorio Git remoto

1.9 git branch

Lista todas las ramas locales.

```
1 git branch -a
```

Lista todas las ramas remotas también

```
1 git branch -d <rama>
```

Elimina la rama especificada

```
1 git branch <nueva rama>
```

Crea una nueva rama

1.10 git checkout

Navega entre diferentes ramas.

```
1 git checkout <rama>
```

```
1 git checkout -b <nueva rama>
```

Crea una nueva rama a partir de la rama actual y cámbiate a ella.

1.11 git merge

Integra cambios de múltiples ramas en una sola.

```
1 git merge <rama>
```

1.12 git remote

Administra las conexiones a repositorios remotos.

Te permite ver qué repositorios remotos están actualmente conectados, así como agregar nuevas conexiones o eliminar las existentes.

```
1 git remote -v
```

Lista todas las conexiones remotas

```
1 git remote add <nombre> <url>
```

Crea una nueva conexión remota

```
1 git remote rm <nombre>
```

Elimina una conexión a un repositorio remoto

```
1 git remote rename <nombre antiguo> <nombre nuevo>
```

Cambia el nombre de una conexión remota

1.13 git fetch

Actualiza el repositorio Git local desde el repositorio remoto correspondiente. `git fetch` no cambia tu espacio de trabajo, mantiene el contenido obtenido separado hasta que se haga un merge.

```
1 git fetch <remoto> <local>
```

```
1 git checkout <remoto>/<local>
```

Para ver el cambio

git fetch vs git pull: `git pull = git fetch + git merge`

1.14 git log

Muestra los snapshots confirmados.

1.15 .gitignore

Un archivo de texto que indica qué archivos y carpetas ignorar en un proyecto.