

Git con HolaMundo

Roolback

Volver de nuestra version 1.1(con errores) a la version 1.0(sin errores).

Ver la version de Git

```
~ { git --version  
git version 2.32.0 (Apple Git-132)
```

Configurar Git

Configurar el name y email

```
~ { git config --global user.name "Nicolas Schurmann"  
~ { git config --global user.email nicolas@gmail.com
```

Configurar editor por defecto

```
~ { git config --global core.editor "code --wait"
```

la opcion --wait es para que la terminal se quede esperando a que se cierre el editor.

Ver el archivo de configuracion

```
~ { git config --global -e  
hint: Waiting for your editor to close the file... █
```

Configurar core.autocrlf para win or linux/mac

Para Win

```
~ { git config --global core.autocrlf true █
```

Para Linux/Mac

```
~ { git config --global core.autocrlf input █
```

Inicializar un proyecto de Git en la carpeta miweb, la cual estara vacia.

```
workspace/miweb { git init  
Initialized empty Git repository in /Users/nicolasschurmann/workspace/miweb/.git/
```

Creamos un archivo con el nombre archivo1.txt y le agregamos texto. Luego vemos el estado

```
workspace/miweb { git status  
On branch main  
  
No commits yet  
  
Untracked files:  
  (use "git add <file>..." to include in what will be committed)  
    archivo1.txt  
  
nothing added to commit but untracked files present (use "git add" to track)
```

Agregar archivos a la etapa de stage

```
workspace/miweb { git add archivo1.txt
```

Volvemos a ver el estado

```
workspace/miweb | git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   archivo1.txt
```

Podemos ver que tenemos cambios que nuestros archivos ya estan en la etapa de stage, para ser comprometidos. Todos los archivos que se encuentren de color verde estan listos para ser comprometidos.

Volvemos a crear un archivo con el nombre archivo2 y le agregamos texto. Luego vemos el estado

```
workspace/miweb | git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   archivo1.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        archivo2.txt
```

Como vemos el archivo2.txt todavia no esta siendo seguido por Git, asi que lo vamos a agregar con el comando que ya vimos anteriormente.

```
workspace/miweb | git add archivo2.txt
```

Volvemos a ver el estado

```
workspace/miweb | git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   archivo1.txt
        new file:   archivo2.txt
```

Ahora podemos ver que se encuentran dos archivos agregados a nuestra etapa de stage.

Ahora vamos a modificar el archivo2.txt para ver que le pasa.

```
workspace/miweb | git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   archivo1.txt
    new file:   archivo2.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   archivo2.txt
```

Como podemos ver se encuentran agregados archivo1 y archivo2 en nuestra etapa de stage, pero tambien nos esta diciendo que existen cambios que no se encuentran en una etapa de stage, y estos cambios se refieren al archivo2. Lo que quiere decir esto es que se realizaron cambios en el archivo2 y que si lo agregamos al stage esos cambios son los que se agregaran y lo que tenia previamente el archivo2 en el stage se modificara con el nuevo contenido. Ojo no estamos pasando los archivos como tal sino los cambios que hemos realizado.

```
workspace/miweb | git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   archivo1.txt
    new file:   archivo2.txt
```

Vamos a comprometer(commit) nuestro trabajo

```
workspace/miweb | git commit -m "Commit inicial"
[main (root-commit) 3d0c515] Commit inicial
2 files changed, 2 insertions(+)
create mode 100644 archivo1.txt
create mode 100644 archivo2.txt
```

Eliminamos el archivo2.txt y luego vemos el estado.

```
workspace/miweb | rm archivo2.txt
workspace/miweb | git status
On branch main

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:    archivo2.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

Ahora veremos los cambios que no se encuentran en la etapa de stage. Así que agregaremos los cambios a la etapa de stage

```
workspace/miweb | git add archivo2.txt
workspace/miweb | git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:    archivo2.txt
```

Ahora vemos que nuestro archivo2 se encuentra en la etapa de stage y ahora vamos a realizar un commit de este cambio.

```
workspace/miweb | git commit -m "Eliminando archivo 2"
[main a769090] Eliminando archivo 2
 1 file changed, 2 deletions(-)
 delete mode 100644 archivo2.txt
workspace/miweb | git status
On branch main
nothing to commit, working tree clean
```

Otra forma mas rapida de eliminar un archivo.

```
workspace/miweb | git rm archivo1.txt
rm 'archivo1.txt'
```

Ahora veremos que nuestro archivo1 se encuentra en la etapa de stage para que se le haga un commit. Si quisieramos sacar un archivo de la etapa de stage escribimos el siguiente comando.

```
workspace/miweb | git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:    archivo1.txt

workspace/miweb | git restore --staged archivo1.txt
workspace/miweb | git status
On branch main
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:    archivo1.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

Ahora vemos que archivo1 ya no se encuentra en la etapa de stage.

Descartar los cambios

```
workspace/miweb | git restore archivo1.txt
workspace/miweb | git status
On branch main
nothing to commit, working tree clean
```

Ignorar archivos

Vamos a crear dos archivos uno donde estas nuestras credenciales (users and passwd) el cual tendra el nombre de .env y el otro tendra el nombre de .gitignore en el que agregaremos el nombre del archivo .env de la misma forma podemos agregar carpetas por decir node_modules/ ,una vez hecho esto agregamos .gitignore a la etapa de stage.

```
workspace/miweb | git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .env

nothing added to commit but untracked files present (use "git add" to track)
workspace/miweb | git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore

nothing added to commit but untracked files present (use "git add" to track)
workspace/miweb | git add .gitignore
workspace/miweb | git commit -m "agregando archivo gitignore"
[main b323fa8] agregando archivo gitignore
1 file changed, 2 insertions(+)
create mode 100644 .gitignore
```

```
gitignore
1  .env
2  node_modules/
```

Verificar en que rama nos encontramos

```
workspace/miweb | git branch
* main
```

Crear una rama

```
workspace/miweb | git checkout -b ramab
Switched to a new branch 'ramab'
workspace/miweb | git branch
main
* ramab
```

Cambiar de rama

```
workspace/miweb | git checkout main
Switched to branch 'main'
```

Realizar un merge

Ojo debemos estar en la rama main

```
workspace/miweb | git merge ramab
Updating 886b51c..b513afa
Fast-forward
 archivo2.txt | 3 ++-
1 file changed, 2 insertions(+), 1 deletion(-)
```

Subir nuestro proyecto a la nube

Lo primero que debemos hacer es crear un repositorio, previamente debemos tener una cuenta en GitHub.

```
workspace/miweb [ git remote add origin https://github.com/nschurmann/miweb.git
workspace/miweb [ git push -u origin main
Username for 'https://github.com': nschurmann
Password for 'https://nschurmann@github.com':
Enumerating objects: 22, done.
Counting objects: 100% (22/22), done.
Delta compression using up to 16 threads
Compressing objects: 100% (13/13), done.
Writing objects: 100% (22/22), 1.91 KiB | 1.91 MiB/s, done.
Total 22 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/nschurmann/miweb.git
* [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

Revisar el historial de nuestro repositorio

```
workspace/miweb [ git log
```

```
workspace/miweb [ git log --oneline
```