

22 de abril de 2020

David Martin Vergues

Curos Git y GitHub

UDEMY course

# Tema 1

## ¿Cómo funciona GIT?

Básicamente con git conseguimos tener un proyecto distribuido. A ese proyecto de le denomina repositorio. Distribuido significa que cada integrante del proyecto tiene una copia de ese repositorio en local.

Git establece una línea del tiempo desde que se crea el repo hasta el día de hoy. Durante esa línea del tiempo vamos tomando fotografías de como se encuentra el proyecto, los llamados commits, de tal manera que siempre podremos volver a alguno de esos momentos.

## Primeros pasos

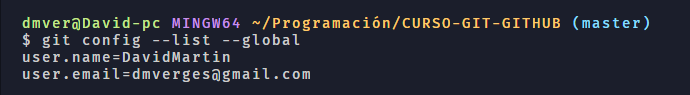
Lo primero que debemos hacer es guardar un nombre de usuario y un mail en nuestra máquina para presentarnos quienes somos en GIT, estas credenciales serán las usadas en todos los repositorios que tengamos en local. Podemos optar por tener un nombre y un mail específicos por repositorio. Todo depende si añadimos al comando la palabra global.

Usamos: config global





Para saber si se ha guardado bien hacemos



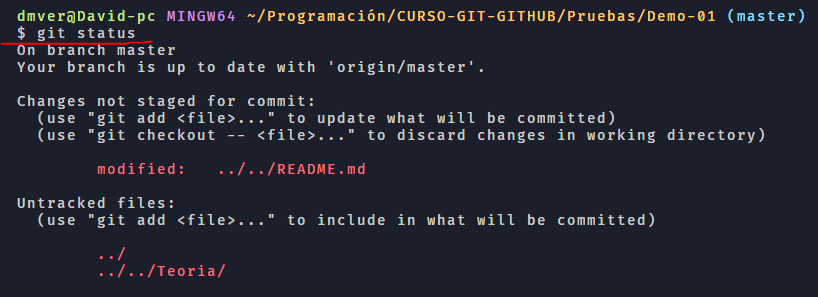
Podemos ir directamente al archivo q almacena esta información en .gitconfig y alterarla(no recomendable)

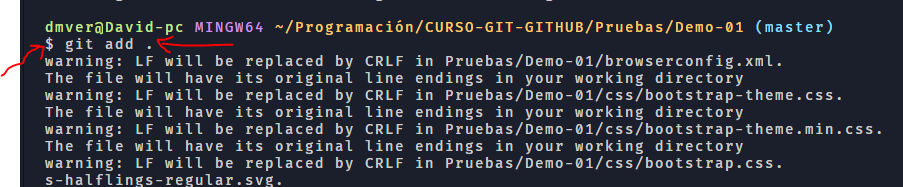




Usamos: status add commit .gitignore

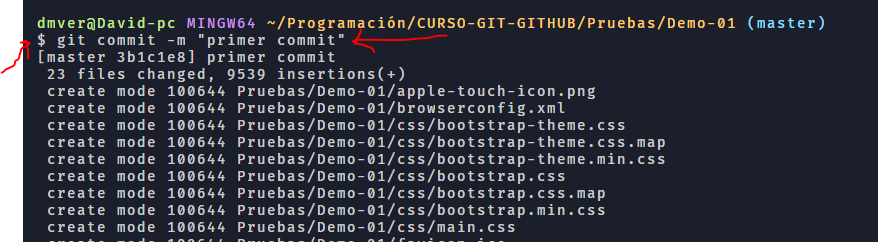
Cuando queremos ver el estado de git usamos git status y si hay cambias que hay que incluir usamos git add.





Al importar un repositorio hecho en Linux en un entorno Windows el final de las líneas en diferente, en Linux usan LF (line feed) y en Windows CR (carriage return).

Para tomar la foto del proyecto actual usamos commit -m “mensaje”



Si queremos descartar algún tipo de archivo, para q no sea seguido por git creamos un archivo .gitignore con la ruta a los archivos que no queremos controlar.



Usamos: checkout

Imaginemos que por error modificamos un archivo o eliminamos su contenido. Con el comando checkout recuperamos el proyecto como estaba en el último commit. Esto también sirve para recuperar carpetas borradas

