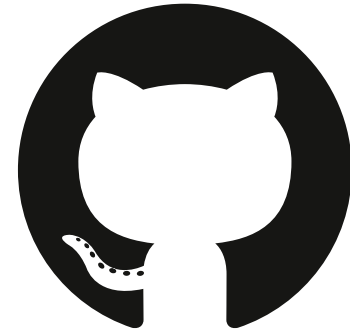


**git**



**Digital**House >  
Coding School

# Cheatsheet git

---

ELIANA CHÁVEZ VALENCIA

# Configuración general

## git config user.email

Muestra el correo electrónico que está configurado en el momento.

**Indicando el correo:** Añadiendo el correo configura esta dirección de correo

**--global:** Es una bandera que configura en todo el sistema este correo, de manera que cada vez que se inicie un nuevo repositorio estará por defecto configurado el correo que se haya especificado

Debe ser el correo que está vinculado con la cuenta de GitHub o el almacenamiento de repositorios que se utilice

## git config user.name

Se comporta de la misma manera que el comando git config user.email

Este nombre es el que aparecerá cuando se realicen los commits

## git remote -v

Permite verificar con que repositorio en la nube está conectado este repositorio local, si no aparece nada es porque todavía no se ha conectado.

# Creación e inicialización de repositorios

## git init

Crea un repositorio local, comienza a hacer seguimiento de todo lo que se coloque en esa carpeta (repositorio) ¡Solo se hace una vez!

## git remote add origin <url>

Conecta un repositorio local con un repositorio remoto.

## git clone <url>

Clona un repositorio remoto en mi computadora, convirtiéndolo en un repositorio local conectado con dicho repositorio remoto. Solo se hace una vez; cuando ya se tiene el repositorio clonado no hay que volver a hacerlo a menos que se haya eliminado por completo de la computadora.

# Información

## git status

Muestra el estado de todos los archivos

## git diff

Muestra las diferencias que existen entre el último commit y los cambios actuales que estamos realizando.

## git log

Muestra todos los commits del repositorio indicando autor, fecha, código y descripción.

**--online:** Muestra de manera resumida los commits

# Interacción con los archivos parte 1

## git add <nombre archivos>

Agrega a la zona de stage\* los archivos especificados. Con punto se agregan todos los archivos que hayan sufrido modificación.

\*Zona en la que están preparados los archivos para ser comiteados y subidos a la nube

## git commit -m "Mensaje descriptivo"

Commitea los cambios que estén en el área de stage y los convierte en la nueva versión actual.

## git push origin <nombre rama>

Sube o sincroniza los cambios locales al repositorio remoto

## git pull origin <nombre rama>

Descarga los cambios del repositorio remoto que no se tengan en el repositorio local

## git restore

Restaura los cambios anteriores

**Con punto:** Regresa todos los cambios que no hayan sido añadidos al stage área

**Indicando el nombre de un archivo o**

**carpeta:** regresa todos los cambios de ese archivo o carpeta que no hayan sido añadidos al stage área

**--staged:** elimina del stage área los archivos especificados o todos poniendo un punto



# Interacción con los archivos parte 2

**git checkout -- <nombre archivo>**

Deshace los cambios que no estén en el stage área de dicho archivo

**git reset HEAD <nombre archivo>**

Elimina del stage area un archivo especificado

# Ramas

## git branch

Muestra todas las ramas del repositorio y resalta la rama actual en la que nos encontramos

**-a** : lista todas las ramas incluyendo aquellas del repositorio remoto

**-m <nombre nuevo>** : Cambia el nombre de la rama actual

**-d <nombre rama>**: elimina la rama especificada

**Indicando el nombre de una rama:** crea una nueva rama con dicho nombre

## git switch <nombre rama>

Cambia de rama

## git checkout

**Indicando nombre de una rama:** cambia a la rama especificada

**Indicando código de un commit:** apunta a la versión especificada

## git merge <nombre rama>

Fusiona la rama especificada con la rama actual.

Es decir que recibe todos los cambios de la rama especificada.