



Vanessa Marlene Sanchez



GIT



Explicaciones con gatitos

GUIÁ

Programming
& TechCats





Vanessa Marlene Sanchez

Intro



Hola, soy gatito GIT, quiero compartirte +30 comandos con explicaciones simples ¿estás listo/a?





Vanessa Marlene Sanchez

Antes de iniciar



Antes que nada, decirte que son explicaciones, los comandos son mucho más complejos pero lo haremos de la manera más simple posible.





Vanessa Marlene Sanchez

Git config

Imagina que estás en el espacio (en este caso tu consola) y dices ¡Vamos a nombrarnos y a nombrar nuestros correos!



Entonces cuando utilizas:

git config --global user.name

"Marlene"

git config --global user.email

"marlenemsanchez@gmail.com"

Es como iniciar iniciar un registro en tu nave de como te llamas y qué email quieres registrar.





Vanessa Marlene Sanchez

Git init

Imagina que estás en el espacio (en este caso tu consola) y dices, ¡Vamos a despegar!



Entonces cuando utilizas **git init** estás comenzando un repositorio y esto creará una “staging area” y un repositorio local.

Es como iniciar a despegar una nave desde tu computadora donde tus archivos van a irse al espacio (Github, Bickbucket, etc).



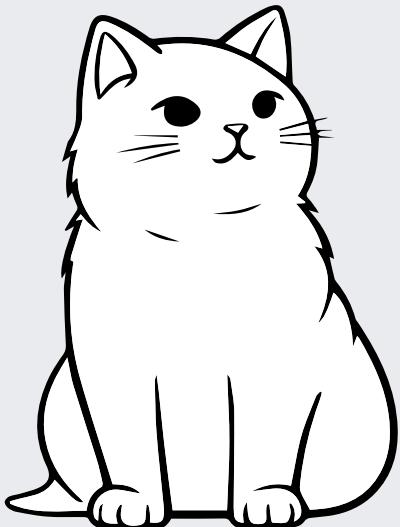


Vanessa Marlene Sanchez

GIT add

Imagina que estás en el espacio (en este caso tu consola) y dices:
“Necesitamos agregar todo lo que necesitamos para irnos al espacio” , en este caso son tus documentos.

Entonces cuando utilizas **git add <nombre_archivo>** o **git add .** (agregar todos) estas agregando tus archivos.



Por lo cual estás añadiendo tus archivos.

Es como subir todo tu equipaje a tu nave espacial.





Vanessa Marlene Sanchez

GIT status

Imagina que estás en el espacio (en este caso tu consola) y dices:
“Necesitamos verificar el estado de nuestras naves” , en este caso son tus documentos.

Entonces cuando utilizas **git status** va a arrojarte información sobre los archivos de tu rama actual (ahorita veremos lo de las ramas)

Por lo cual estás verificando que todo está dentro de la nave



Vanessa Marlene Sanchez

GIT commit

Imagina que estás en el espacio (en este caso tu consola) y dices:

“Necesitamos enviar mensajes al espacio”, en este caso es un mensaje que dirá qué estás haciendo o creando (hay parámetros para hacer commits pero solo piensa por ahora que es un mensaje)

Entonces cuando utilizas **git commit -m “mensaje”**

Por lo tanto es como almacenar un mensaje que será muy útil para decirle a tu equipo qué es

Es como mandar un mensaje a otras naves.





Vanessa Marlene Sanchez

GIT clone



Imagina que estás en el espacio (en este caso tu consola) y dices: “Necesitamos duplicar nuestras naves”, en este caso es como clonar dos naves y sean idénticas. Desde el espacio hacia tu nave (remoto-local). Entonces cuando utilizas **git clone <enlace_repositorio_github>** Por lo tanto es como clonar lo que tiene una nave en el espacio a una nave local.





Vanessa Marlene Sanchez

GIT pull

Imagina que estás en el espacio (en este caso tu consola) y dices:

“Necesitamos actualizar lo que tienen nuestras naves”, en este caso es como extraer y descargar contenido al instante a tu nave (remoto-local).

Entonces cuando utilizas **git pull** extraes y descargas contenido desde un repositorio remoto a tu repositorio local, reflejando tal contenido.



git pull





Vanessa Marlene Sanchez

GIT push



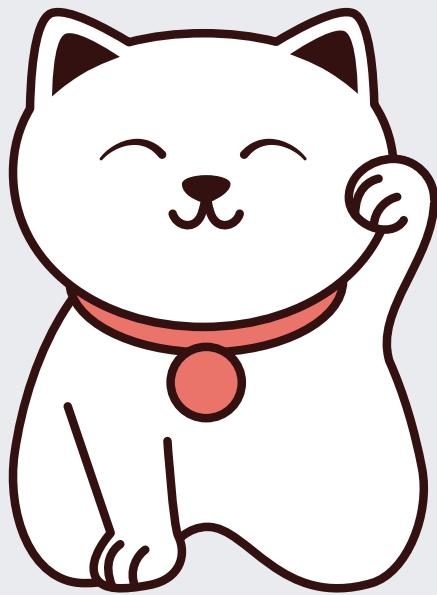
Imagina que estás en el espacio (en este caso tu consola) y dices:
“Necesitamos mandar todo lo que hay en nuestra nave”, en este caso es como ¡mandar a volar tu nave!
Entonces cuando utilizas **git push** para mandar todo lo de tu local a remoto (repositorio) (local-remoto)
Por lo tanto es mandar lo que tiene nuestra nave a un espacio





Vanessa Marlene Sanchez

GIT branch



Imagina que estás en el espacio pero ahora quieres crear muchos espacios, o dos (en este caso en tu consola) y dices: "Necesitamos crear otro espacio para evitar una catástrofe y hacer todos los cambios en un solo espacio para no afectar al espacio principal" (main/master).

Entonces **git branch** te permite crear, enumerar, cambiar nombre, etc (de una rama).





Vanessa Marlene Sanchez

GIT checkout

Imagina que estás en el espacio pero ahora quieres cambiarte de espacio (en este caso tu consola) y dices: "Necesitamos cambiar de espacio para almacenar ahí nuestras naves y evitar el colapso".

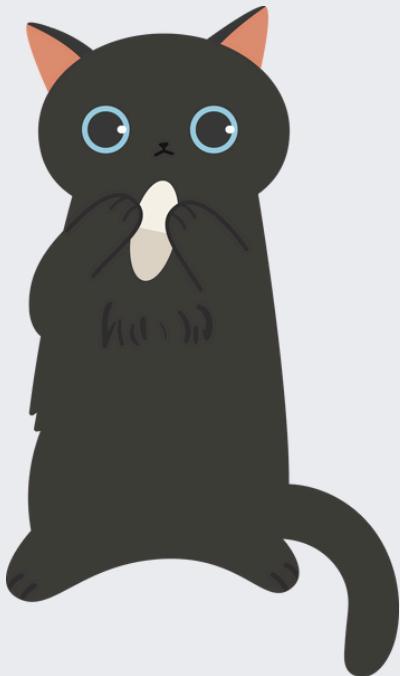
Entonces **git checkout <rama>** permite cambiarte de una rama a otra para que puedas checar archivos y commits.





Vanessa Marlene Sanchez

GIT revert



Imagina que estás en el espacio ahora quieras revertir algo (en este caso tu consola) y dices:
“Necesitamos revertir un punto de nuestra nave ¡Todo va a colapsar! ”.

Entonces **git revert** va a deshacer cambios que se realizaron en el historial de confirmaciones de nuestro repositorio.





Vanessa Marlene Sanchez

GIT remote -v

Imagina que estás en el espacio ahora quieres saber en qué espacio estás conectado (en este caso tu consola) y dices: "Necesitamos verificar que el espacio es el correcto porque haremos el lanzamiento".

Entonces **git remote -v** va a mostrarte una lista de conexiones remotas





Vanessa Marlene Sanchez

GIT restore



Imagina que estás en el espacio y quieres restaurar los paquetes que enviaste desde tu nave (en este caso tu consola) y dices: "Necesitamos hacer un punto de restauración de todo lo que enviamos".

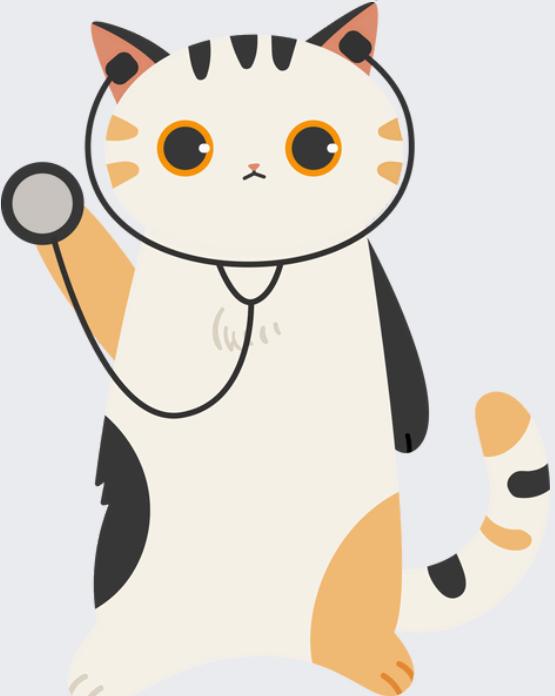
Entonces **git restore <archivo>** va a ayudarte a restaurar un archivo en su estado por defecto. Elimina los cambios en tu directorio como si no hubiera pasado nada.





Vanessa Marlene Sanchez

GIT log



Imagina que estás en el espacio y quieres verificar todos los mensajes que mandaste a otras naves desde tu nave (en este caso tu consola) y dices: "Necesitamos verificar que los mensajes han sido enviados de manera exitosa".

Entonces **git log** va a ayudarte a visualizar un historial completo de tus commits a través de una lista cronológica que te permite ver los cambios, a continuación de explicaré qué vas a encontrar ahí.





Vanessa Marlene Sanchez

GIT log 2

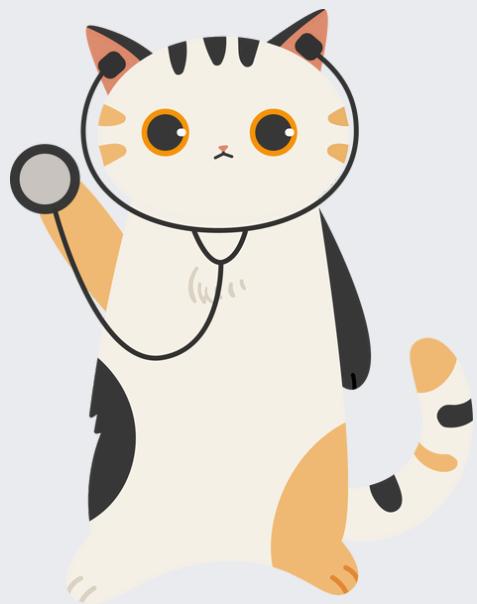
Como te mencione anteriormente verás un historial de confirmaciones (commmits) y vas a encontrar lo siguiente:

SHA: Es un identificador de cada commit (Secure Hasg Algorithm) hace referencia a un commit.

Autor: Verás el nombre y email del autor que hizo el commit.

Fecha: Verás la fecha y hora en la que se hizo el commit

Mensaje: Es un mensaje descriptivo, que te permite ver el commit y los cambios realizados





Vanessa Marlene Sanchez

GIT diff

Imagina que estás en el espacio y quieres ver las diferencias que hay entre las naves y sus espacios (en este caso tu consola) y dices: "Necesitamos verificar la diferenciación de cada nave/espacio".

Entonces **git diff** va a ayudarte a ver las diferencias que existen entre las confirmaciones.

Puedes utilizar `git diff <hash commit1> <hash commit 2>`

`git HEAD` : muestra el commit que tenemos guardado en nuestro repositorio local.

Vemos la diferencia entre 2 ramas





Vanessa Marlene Sanchez

Git checkout -b

Imagina que estás en el espacio y quieres crear un nuevo espacio y cambiarte a él rápidamente (en este caso tu consola) y dices: "Necesitamos crear y cambiar de espacio en 5 segundos, ¡La nave está por romperse, tenemos 5 segundos! ".

Entonces **git checkout -b <nombre_rama>** va a ayudarte a crear una rama e ir a ella



Vanessa Marlene Sanchez

GIT clean



Imagina que estás en el espacio (en este caso tu consola) y quieres omitir algunos paquetes que están en tu nave porque hay un gran conflicto y dices:
“Necesitamos omitir algunos paquetes y su lanzamiento porque la nave está por dejar de funcionar”.

Entonces **git clean** va a ayudarte a eliminar archivos y el seguimiento de tu directorio.





Vanessa Marlene Sanchez

Git commit --amend



Imagina que estás en el espacio (en este caso tu consola) y quieres modificar el último mensaje que envió tu nave hacia otro espacio remoto y dices:
“Necesitamos modificar el último mensaje que envió nuestra nave porque no fue muy claro y las otras naves pueden confundirse”.

Entonces **git commit --amend** va a ayudarte a modificar tu commit más reciente, esto es útil cuando olvidas preparar un archivo o haces alguna omisión en el mensaje de confirmación (commit).





Vanessa Marlene Sanchez

GIT log -1



Imagina que estás en el espacio (en este caso tu consola) y quieres ver el último mensaje que envió tu nave hacia otro espacio remoto y dices: "Necesitamos ver el último mensaje que envió nuestra nave porque mi nave envió uno y quiero confirmarlo".

Entonces **git log -1** va a ayudarte a ver tu commit más reciente, esto es útil cuando quieras ver la última confirmación





Vanessa Marlene Sanchez

GIT help



Imagina que estás en el espacio (en este caso tu consola) y quieres obtener todos los comandos que puedes utilizar para despegar tu nave y dices: "Necesitamos ver todos los comandos para verificar cuales son útiles y cuándo utilizarlos".

Entonces **git help** va a ayudarte a ver un listado de comandos que puedes utilizar en GIT



Vanessa Marlene Sanchez

GIT help init



Imagina que estás en el espacio (en este caso tu consola) y quieres obtener información sobre cómo utilizar el comando para despegar tu nave (git init) y dices: “Necesitamos verificar la información del comando de despegue (init)”.

Entonces **git help init** va a ayudarte a ver un listado de comandos que puedes utilizar en GIT en torno al comando git init





Vanessa Marlene Sanchez

GIT --stat



Imagina que estás en el espacio (en este caso tu consola) y quieres obtener información sobre todo lo relacionado a tus mensajes enviados a otras naves y aquello que eliminaste y dices:
“Necesitamos verificar la información de los mensajes que incluso se han eliminado”.

Entonces **git --stat** va a ayudarte a ver las inserciones y eliminaciones de una confirmación

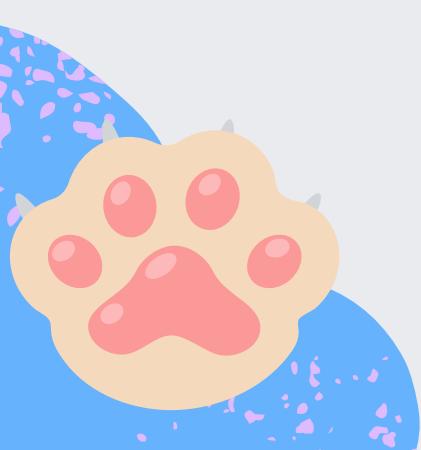




Vanessa Marlene Sanchez

GIT --bisect

Imagina que estás en el espacio (en este caso tu consola) y quieras obtener información sobre si hay algún mensaje con algún problema o error y dices : “Necesitamos verificar si los mensajes realmente están funcionando”. Entonces **git --bisect good** o **git bisect bad** donde se te llevará al primer commit y luego al siguiente, y así sucesivamente, para ver si alguna confirmación contiene algún error o no





Vanessa Marlene Sanchez

GIT ...mv



Imagina que estás en el espacio (en este caso tu consola) y quieres renombrar el nombre de tus cajas que vas a lanzar al espacio y dices :
“Necesitamos cambiar el nombre para identificar cada caja enviada”.
Entonces **git mv [archivo_original] [archivo_renombrado]** va a ayudarte a cambiar el nombre de tus archivos y prepararlos para el siguiente commit





Vanessa Marlene Sanchez

Git commit -a -m



Imagina que estás en el espacio (en este caso tu consola) y quieres lanzar un nuevo mensaje al espacio y dices : “Necesitamos mandar un nuevo mensaje cifrado de advertencia porque la nave parece presentar un fallo”. Entonces **git commit -a -m “segundo comentario”** va a ayudarte a añadir comentarios en donde te encuentres trabajando (local).





Vanessa Marlene Sanchez

Git rm --cached



Imagina que estás en el espacio (en este caso tu consola) y quieres borrar un paquete que enviaste al espacio y dices : “Necesitamos borrar el paquete porque hay un error y puede explotar!”.

Entonces **git rm --cached archivo** va a ayudarte a borrar un archivo de tu repositorio





Vanessa Marlene Sanchez

Git stash pop



Imagina que estás en el espacio (en este caso tu consola) y quieres recuperar los cambios que realizaste porque al parecer la nave funciona y dices : “Necesitamos recuperar la información de la nave”. Entonces **git stash pop** va a ayudarte a recuperar esos cambios.





Vanessa Marlene Sanchez

Git switch



Imagina que estás en el espacio (en este caso tu consola) y quieres cambiar de rama y dices :
"Necesitamos cambiar de espacio y solamente crear uno y movernos a él".

Entonces **git switch -c**

<new_branch> va a ayudarte a crear una rama y moverte. A diferencia de git branch que se asignan más responsabilidades porque puedes moverte, deshacer cambios, restaurar archivos y tener mayor control.

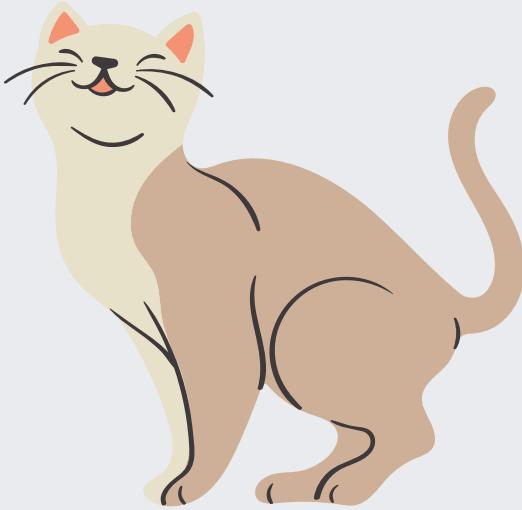
git switch solo permite crear y moverte en ramas





Vanessa Marlene Sanchez

Git tag



Imagina que estás en el espacio (en este caso tu consola) y quieres crear una etiqueta y dices : “Necesitamos crear una etiqueta de la nave”. Entonces **git tag -a v2.0 -m “this is version 2.0 tag”** va a ayudarte a marcar un commit específicamente con un identificador que sea simple y legible





Vanessa Marlene Sanchez

Git fsk



Imagina que estás en el espacio (en este caso tu consola) y quieres identificar los problemas con tu espacio y dices : “Necesitamos verificar qué problemas o inconsistencias hay en nuestro espacio”.

Entonces **git fsk** va a ayudarte a comprobar la integridad de tus archivos e identificar cuáles archivos son corruptos





Vanessa Marlene Sanchez

Git prune



Imagina que estás en el espacio (en este caso tu consola) y que tu nave tiene objetos innecesarios por lo que te piden que limpies tu nave y dices : "Necesitamos limpiar la nave de objetos no necesarios, e innalcanzables ".

Entonces **git prune** va a ayudarte a reliminar objetos que no están aportando un objeto como tal en alguna rama accesible optimizando nuestro repositorio





Vanessa Marlene Sanchez

Git blame



Imagina que estás en el espacio (en este caso tu consola) y que el capitán de la sala de control te dice “Necesitamos generar una revisión, ya que hay código que está generando interferencia y no tiene sentido en nuestras naves”

Entonces **git blame** es como si inspeccionarás cada línea de código e identificarás quién fue el responsable de agregar esas partes que no contribuyen al proyecto.





Vanessa Marlene Sanchez

Git blame -e



Imagina que estás en el espacio (en este caso tu consola) y que el capitán de la sala de control te dice “Necesitamos generar una revisión, ya que hay código que está generando interferencia y no tiene sentido en nuestras naves, verifica quién fue”

Entonces **git blame -e [filename]** es como si inspeccionarás cada línea de código e identificarás quién fue el responsable, donde quieras identificar el email del propietario que se encargo de agregar esas partes que no contribuyen al proyecto.





Vanessa Marlene Sanchez

Antes de finalizar



Agradecerte tu apoyo, si te gustaría que hubieran más formatos con gatitos apreciaríamos si nos indicas en los comentarios que quieres que sigamos explicandote.

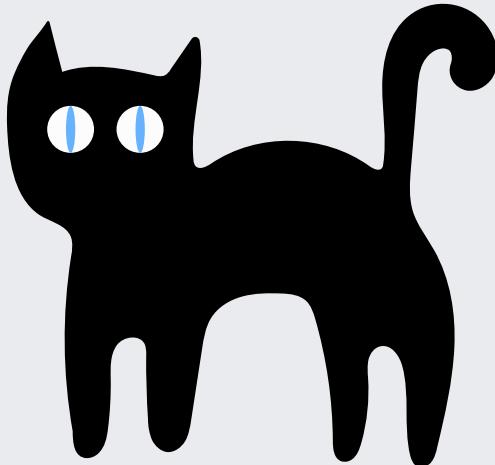
Ha sido una gran travesía, pero esperamos todo esto te ayude a comprender más los comandos.
Se despide por ahora GItFlowing jr





Vanessa Marlene Sanchez

Anuncio gatuno



En nombre de todos mis amigos gatitos
te damos las gracias por el apoyo, GIT
está a punto de finalizar, pero te
explicaremos sobre SQL, JS, y más.





Vanessa Marlene Sanchez



Gracias por leer

Si te gusto el contenido, agradecería mucho si pudieses compartirlo.

Espera las siguientes actualizaciones.

