

Instantly share code, notes, and snippets.

dasdo / **GIT.md**



Last active 2 days ago

<> **Code** Revisions 14 Stars 474 Forks 285

Lista de Comandos en GIT

GIT.md

Configuración Básica

Configurar Nombre que salen en los commits

```
git config --global user.name "dasdo"
```

Configurar Email

```
git config --global user.email dasdo1@gmail.com
```

Marco de colores para los comando

```
git config --global color.ui true
```

Iniciando repositorio

Iniciamos GIT en la carpeta donde esta el proyecto

```
git init
```

Clonamos el repositorio de github o bitbucket

```
git clone <url>
```

Añadimos todos los archivos para el commit

```
git add .
```

Hacemos el primer commit

```
git commit -m "Texto que identifique por que se hizo el commit"
```

subimos al repositorio

```
git push origin master
```

GIT CLONE

Clonamos el repositorio de github o bitbucket

```
git clone <url>
```

Clonamos el repositorio de github o bitbucket ?????

```
git clone <url> git-demo
```

GIT ADD

Añadimos todos los archivos para el commit

```
git add .
```

Añadimos el archivo para el commit

```
git add <archivo>
```

Añadimos todos los archivos para el commit omitiendo los nuevos

```
git add --all
```

Añadimos todos los archivos con la extensión especificada

```
git add *.txt
```

Añadimos todos los archivos dentro de un directorio y de una extensión específica

```
git add docs/*.txt
```

Añadimos todos los archivos dentro de un directorio

```
git add docs/
```

GIT COMMIT

Cargar en el HEAD los cambios realizados

```
git commit -m "Texto que identifique por que se hizo el commit"
```

Agregar y Cargar en el HEAD los cambios realizados

```
git commit -a -m "Texto que identifique por que se hizo el commit"
```

De haber conflictos los muestra

```
git commit -a
```

Agregar al último commit, este no se muestra como un nuevo commit en los logs. Se puede especificar un nuevo mensaje

```
git commit --amend -m "Texto que identifique por que se hizo el commit"
```

GIT PUSH

Subimos al repositorio

```
git push <origien> <branch>
```

Subimos un tag

```
git push --tags
```

GIT LOG

Muestra los logs de los commits

```
git log
```

Muestras los cambios en los commits

```
git log --oneline --stat
```

Muestra graficos de los commits

```
git log --oneline --graph
```

GIT DIFF

Muestra los cambios realizados a un archivo

```
git diff  
git diff --staged
```

GIT HEAD

Saca un archivo del commit

```
git reset HEAD <archivo>
```

Devuelve el ultimo commit que se hizo y pone los cambios en staging

```
git reset --soft HEAD^
```

Devuelve el ultimo commit y todos los cambios

```
git reset --hard HEAD^
```

Devuelve los 2 ultimo commit y todos los cambios

```
git reset --hard HEAD^^
```

Rollback merge/commit

```
git log  
git reset --hard <commit_sha>
```

GIT REMOTE

Agregar repositorio remoto

```
git remote add origin <url>
```

Cambiar de remote

```
git remote set-url origin <url>
```

Remover repositorio

```
git remote rm <name/origin>
```

Muestra lista repositorios

```
git remote -v
```

Muestra los branches remotos

```
git remote show origin
```

Limpiar todos los branches eliminados

```
git remote prune origin
```

GIT BRANCH

Crea un branch

```
git branch <nameBranch>
```

Lista los branches

```
git branch
```

Comando -d elimina el branch y lo une al master

```
git branch -d <nameBranch>
```

Elimina sin preguntar

```
git branch -D <nameBranch>
```

GIT TAG

Muestra una lista de todos los tags

```
git tag
```

Crea un nuevo tags

```
git tag -a <verison> - m "esta es la versión x"
```

GIT REBASE

Los rebase se usan cuando trabajamos con branches esto hace que los branches se pongan al día con el master sin afectar al mismo

Une el branch actual con el master, esto no se puede ver como un merge

```
git rebase
```

Cuando se produce un conflicto no das las siguientes opciones:

cuando resolvemos los conflictos --continue continua la secuencia del rebase donde se pauso

```
git rebase --continue
```

Omite el conflicto y sigue su camino

```
git rebase --skip
```

Devuelve todo al principio del rebase

```
git rebase --abort
```

Para hacer un rebase a un branch en específico

```
git rebase <nameBranch>
```

OTROS COMANDOS

Lista un estado actual del repositorio con lista de archivos modificados o agregados

```
git status
```

Quita del HEAD un archivo y le pone el estado de no trabajado

```
git checkout -- <file>
```

Crea un branch en base a uno online

```
git checkout -b newlocalbranchname origin/branch-name
```

Busca los cambios nuevos y actualiza el repositorio

```
git pull origin <nameBranch>
```

Cambiar de branch

```
git checkout <nameBranch/tagname>
```

Une el branch actual con el especificado

```
git merge <nameBranch>
```

Verifica cambios en el repositorio online con el local

```
git fetch
```

Borrar un archivo del repositorio

```
git rm <archivo>
```

Fork

Descargar remote de un fork

```
git remote add upstream <url>
```

Merge con master de un fork

```
git fetch upstream
```



```
git merge upstream/master
```

[Load earlier comments...](#)

alexdeveloper8 commented on Jul 29, 2021

Amigo puedo colaborar con más

Nemo3003 commented on Jul 30, 2021

Genial!

PedroBonet commented on Aug 1, 2021

Gracias

IrvingGeyler commented on Sep 28, 2021

Muchas gracias.

aNDREUET648 commented on Oct 11, 2021

Aporte interesante. Muchas gracias

estherff commented on Oct 27, 2021

para que sirve git branch -M master que significa la M?

En este enlace tienes una muy buena explicación <https://platzi.com/blog/cambios-en-github-master-main/>

julianmagra commented on Feb 16, 2022

Que hace el git -u? Hay de donde ver todos los comandos? ?

aNDREUET648 commented on Feb 17, 2022

Hola, el git -u no se que hace, pero el que sí funciona es el git push -u. Por lo de ver todos los comandos te aconsejo un curso de w3schools, son bastante buenos todos los que tienen y el de git está particularmente bastante bien.

<https://www.w3schools.com/git/default.asp>

Salud
Andreu

aNDREUET648 commented on Feb 17, 2022 via email 

Hola, el git -u no se que hace, pero el que sí funciona es el git push -u. Por lo de ver todos los comandos te aconsejo un curso de w3schools, son bastante buenos todos los que tienen y el de git está particularmente bastante bien.

<https://www.w3schools.com/git/default.asp>

Salud

Andreu

El mié, 16 feb 2022 a las 21:56, Julian Nicolas Magra (<
@.***>) escribió:

...

giancarlos1230 commented on Jul 4, 2022

Excelent

santiCBA commented on Jul 27, 2022

Tremendo aporte!

cristian545 commented on Aug 2, 2022

muchas gracias por la gran ayuda

Caro-Huell commented on Aug 19, 2022

gracias por laayuda

jacobgomez666 commented on Sep 27, 2022

muy claro, gracias crack.

MichGM commented on Sep 27, 2022

Muchas gracias Bro! Muy util está informacion!!

adolfnavarro commented on Oct 20, 2022

Gracias :)

EdsonSalas commented on Nov 2, 2022

Excelente aporte, muchas gracias.

leoangel094 commented on Nov 15, 2022

una ayuda por favor.
Que hace git config pull.rebase false/true ?

PazDaiana commented on Dec 23, 2022

.

ricardorivaspereda commented on May 23, 2023

Excelente info! Se agradece

IddleMaster commented on Sep 4, 2023

Como puedo conectar con un repositorio del cual descargue en lugar de clonar?, o si despues de los cambios con el clone los puedo guardar de igual forma?

marcelozepeunam commented on Sep 12, 2023

Muchas gracias por la aportación, la verdad es muy útil

RgeditV1 commented on Feb 22, 2024

Muchisimas gracias bro

EduEzMez commented on Apr 8, 2024

Muchas gracias!!

jose-erazo commented on Apr 9, 2024

Gracias por la información.

JuanEstenbanQuirama commented on Apr 23, 2024

```
git add . # Agregar todos los archivos
git add /path/to/file # Agregar un archivo concreto
git diff --cached
git commit -m 'comment'
git commit --amend -m 'my corrected comment' # Para corregir el mensaje del commit anterior.
git push -u origin master
git checkout -b 'robranch' #o (git switch robranch) # Cambiar de branch
git push origin robranch
git branch -a # Ver las branches
git branch -m # renombrar la branch actual
git merge # Une otra branch con la branch en la te encuentras
git merge --squash # Igual que el anterior pero junta todo los commits(es necesario hacer un nuevo
add . y git commit -m "mensaje" )
git fetch --all # Actualizar las branches
git rm --cached myarchivo.txt # Elimina el archivo en todo el historial
git remote update origin --prune # Actualiza las branches remotas localmente
git stash # ir al commit anterior pero salvando los datos por si acaso quieres usar luego.
git stash pop # para regresar al stash previo (al que no se le ha hecho commit).
git stash push -m "mensaje" # Para hacer stash ("salvar los cambios" sin commit) y agregar un mensaje a
dicho stash.
git stash list # ver la lista de stashes.
git init
git config --global user.name "Your Name Comes Here"
git config --global user.email you@yourdomain.example.com
git config --list # para saber el nombre e email configurados en git
git status
```

AstroMark11 commented on Jun 19, 2024

Información muy útil, muchas gracias

nesyita commented on Aug 26, 2024

excelente es muy util saber los comandos y para que se utilizan gracias me ayudo muchisimo

Raff4-Gauna commented on Sep 17, 2024

Genial Gracias!!!

Mani8217 commented on Oct 2, 2024

Genial !