

Instantly share code, notes, and snippets.

dasdo / **GIT.md**

Last active 23 hours ago

☆ Star

<> Code

🔗 Revisions 14

☆ Stars 154

🔗 Forks 110

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 directorios

```
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 ultimo 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 <version> - 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
```

 **ghost** commented on 15 Mar 2017 • edited by ghost ▾

Hola, dasdo

añadí información en mi forked repository acerca de GIT DIFF, pero no encuentro la opción de hacer un pull request, si te sirve, podrias añadirlo.

<https://gist.github.com/andrec1234/a8d45f48f2422ad6c153c4ddabad9248>

Saludos

 **Federico2017** commented on 20 Mar 2017

Hola, buenas noches, tengo terminada la carpeta de un primer proyecto, necesito realizar un github_url.txt, necesito apoyo

 **allam89** commented on 30 Jan 2018

v:> me vale xdx

 **allam89** commented on 30 Jan 2018

<https://gist.github.com/dasdo/9ff71c5c0efa037441b6>

 **allam89** commented on 30 Jan 2018

holo v:< dance warriorsv:

 **Sisimitv** commented on 30 Jan 2018

Allam Gay V:

 **allam89** commented on 30 Jan 2018

fredy hueco v:

 **IvanAjquejay** commented on 30 Jan 2018

mejor callese allam89

 **raquec** commented on 30 Jan 2018

allam gay balla a dormir :V

 **allam89** commented on 30 Jan 2018

arnulfo no se pierda en la noche xd v:

 **Juank343** commented on 30 Jan 2018

No mamen, malditos! :v

 **dantedelao89** commented on 15 Apr 2019

Hola, como hago para ver en mi local qué archivos y lineas cambiaron en el repositorio github al hacer pull al local?

 **Federico-Taborda** commented on 29 Jun 2019

yo cree un commit y me olvide de ponerle add . a los archivos primero. Ahora cuando pongo git log me salta error. Como borro un commit?



xtealer commented on 11 Jul 2019 • edited ▼

Puedes volver a una version anterior usando `git reset --hard HEAD~1` (eliminara el ultimo commit hecho) tambien puedes usar `git reset --hard <sha1-commit-id>` para ser mas especifico (eliminar un commit por sha1-id). Puedes encontrar mas informacion [aqui](#) (ingles).



02ernesto commented on 10 Aug 2019

Excelente información :)



AnderAlcarraz commented on 21 Oct 2019

Donde puedo encontrar mayo información



hadiazb commented on 10 Jan 2020

Buen aporte



JuanBustosV commented on 2 Feb 2020

Muchas gracias, información muy útil, clara y concisa.



EduardoMay commented on 14 Feb 2020

Excelente información




jgranados4 commented on 9 Mar 2020

hola que tal como le hace para poner texto asi en github soy nuevo en esto


  **EduardoMay** commented on 10 Mar 2020

Cual texto?

...

 **jgranados4** commented on 10 Mar 2020

los texto de comando como los pusiste

 **oscarzatell** commented on 16 Jun 2020

genial

 **Jesusvelasquez** commented on 22 Oct 2020

Excelente aporte!

 **sebasbeltran95** commented on 29 Dec 2020

gracias

 **jPablo23** commented on 1 Feb

Gracias, Me tome un descanso de programar y ahora que voy a retomar todo, me di cuenta que olvide muchas cosas... jajajajaja

 **jgranados4** commented on 5 Feb

Hola como le hiciste para implementar texto asi con git.md en tu repositorio

  **LaPolola** commented on 6 Feb

Holi!

<https://docs.github.com/es/github/writing-on-github/basic-writing-and-formatting-syntax>

<https://docs.github.com/es/github/writing-on-github/basic-writing-and-formatting-syntax>

Ahí te dice como tenes que formatear, el tema de los emojis es cuestión de que encuentres una página de emojis donde puedas copiarlos y los pegues en el readme, te detecta el formato.

entra al repo > click en readme.md> click en RAW. <Ahí ves el formato en crudo del readme>

Espero te haya sido útil la info.

Saludos!

El vie, 5 de feb. de 2021 a la(s) 12:33, Juan Gustavo Granados Merchán (notifications@github.com) escribió:

...



JuanMachucaE commented on 9 Mar

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



majocardozo commented on 17 Mar

cual es la diferencia de hacer:
git push -u origin ramaremota --tags
con hacer:
git push origin ramaremota --tags
que hace el -u??



lore198429 commented on 5 Jun

el mejor contenido encontrado



alexdeveloper8 commented on 29 Jul

Amigo puedo colaborar con más



Nemo3003 commented on 30 Jul



commented on 30 Jul

Genial!



PedroBonet commented on 1 Aug

Gracias



IrvingGeyler commented 21 days ago

Muchas gracias.



aNDREUET648 commented 8 days ago

Aporte interesante. Muchas gracias