

Comparta instantáneamente código, notas y fragmentos.

dasdo / GIT.md



Última actividad Hace 4 días

<> Código Revisiones 14 Estrellas 489 Tenedores 290

Lista de comandos en GIT

GIT.md

Configuración básica

Configurar Nombre que salen en los commits

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

Configurar correo electrónico

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

Marco de colores para los comandos.

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

Iniciando el repositorio

Iniciamos GIT en la carpeta donde está 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 compromiso

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

Subimos al repositorio

```
git push origin master
```

CLON DE GIT

Clonamos el repositorio de github o bitbucket

```
git clone <url>
```

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

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

AGREGAR GIT

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/
```

Confirmación de GIT

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 las muestras

```
git commit -a
```

Agregando el último compromiso, este no se muestra como un nuevo compromiso en los registros. 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>
```

Enviamos una etiqueta

```
git push --tags
```

REGISTRO DE GIT

Muestra los logs de los commits

```
git log
```

Muestras los cambios en los commits

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

Muestra gráfica de los commits

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

Diferencia de Git

Muestra los cambios realizados a un archivo

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

CABEZA DE GIT

Saca un archivo del commit

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

Devuelve el último compromiso que se hizo y pone los cambios en la puesta en escena.

```
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^^
```

Revertir fusión/confirmación

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

GIT REMOTO

Agregar repositorio remoto

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

Cambiar de control remoto

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

Removedor de repositorio

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

Muestra lista de repositorios

```
git remote -v
```

Muestra las sucursales remotas

```
git remote show origin
```

Limpiar todas las ramas eliminadas

```
git remote prune origin
```

RAMA GIT

Crea una rama

```
git branch <nameBranch>
```

Lista de sucursales

```
git branch
```

Comando -d elimina el rama y lo une al maestro

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

Elimina el sin preguntar

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

ETIQUETA GIT

Muestra una lista de todos los tags.

```
git tag
```

Crea una nueva etiqueta

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

REBASE DE GIT

Los rebase se usan cuando trabajamos con sucursales esto hace que los sucursales se pongan al día con el maestro 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 pausa

```
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 una rama 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 una sucursal en base a uno en línea

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

Busca los cambios nuevos y actualiza el repositorio

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

Cambiar de sucursal

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

Une el sucursal 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>
```

Tenedor

Descargar control remoto de un fork

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

Fusionar con master de un fork

```
git fetch upstream
```



```
git merge upstream/master
```

[Cargar comentarios anteriores...](#)

Pedro Bonet comentó el 1 de agosto de 2021

Gracias

IrvingGeyler comentó el 28 de septiembre de 2021

Muchas gracias.

aNDREUET648 comentó el 11 de octubre de 2021

Aporte interesante. Muchas gracias

Estherff comentó el 27 de octubre de 2021

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

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

julianmagra comentó el 16 de febrero de 2022


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

aNDREUET648 comentó el 17 de febrero de 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 comentó el 17 de febrero de 2022 por correo electrónico 

Hola, el git -u no se que hace, pero el que sí funciona es el git push -u. Por lo de ver todos lo 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, Julián Nicolás Magra (< ***@***.***>) escribió:

...

giancarlos1230 comentó el 4 de julio de 2022

Excelente

santiCBA comentó el 27 de julio de 2022

¡Tremendo aporte!

cristian545 comentó el 2 de agosto de 2022

muchas gracias por la gran ayuda

Caro-Huell comentó el 19 de agosto de 2022

Gracias por la ayuda

jacobgomez666 comentó el 27 de septiembre de 2022

muy claro, gracias crack.

MichGM comentó el 27 de septiembre de 2022

¡Muchas gracias hermano! Muy util esta informacion!!

adolfnavarro comentó el 20 de octubre de 2022

Gracias :)

EdsonSalas comentó el 2 de noviembre de 2022

Excelente aporte, muchas gracias.

leoangel094 comentó el 15 de noviembre de 2022

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

PazDaiana comentó el 23 de diciembre de 2022

.

ricardorivaspereda comentó el 23 de mayo de 2023

¡Excelente información! Se agradece.

IddleMaster commented el 4 de septiembre de 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 el 12 de septiembre de 2023

Muchas gracias por la aportación, la verdad es muy útil

RgeditV1 commented el 22 de febrero de 2024

Muchisimas gracias bro

EduEzMez commented el 8 de abril de 2024

Muchas gracias!!

jose-erazo commented el 9 de abril de 2024

Gracias por la información.

JuanEstenbanQuirama commented el 23 de abril de 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 git
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 el 19 de junio de 2024

Información muy útil, muchas gracias

nesyita commented el 26 de agosto de 2024

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

Raff4-Gauna commented el 17 de septiembre de 2024

Genial Gracias!!!

Mani8217 commented el 2 de octubre de 2024

Genial !

Johanlo28 commented el 24 de febrero

gracias

gDev-Gerard comentó Hace 4 días

Muchísimas gracias