

UT 2.3 GIT. Staging y trabajo remoto

ENTORNOS DE DESARROLLO

1. Diferencia entre ficheros unstaged

Imaginemos que estás haciendo cambios sobre diferentes ficheros ya controlados, pero no recuerdas qué cambios han sido.



1. Diferencia entre ficheros unstaged

```
$ git diff
```

Muestra las diferencias desde el último commit de aquellos ficheros que no están en modo stage

```
diff --git a/LICENSE b/LICENSE
```

```
index 7e4922d..442669e 100644
```

```
--- a/LICENSE
```

```
+++ b/LICENSE
```

```
@@ -1,4 +1,4 @@
```

```
-Copyright (c) 2012 Envy Labs LLC
```

```
+Copyright (c) 2012 Code School LLC
```

Línea eliminada

Línea añadida

1. Diferencia entre ficheros staged

- También es posible ver las diferencias entre ficheros en estado “staged”.

```
$ git add LICENSE
```

```
$ git diff
```

No hay diferencias desde el momento en los cambios han pasado a modo *staged*

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

Para ver diferencias de ficheros *staged*

```
diff --git a/LICENSE b/LICENSE
```

```
index 7e4922d..442669e 100644
```

```
--- a/LICENSE
```

```
+++ b/LICENSE
```

```
@@ -1,4 +1,4 @@
```

```
-Copyright (c) 2012 Envy Labs LLC
```

```
+Copyright (c) 2012 Code School LLC
```



2. Sacar ficheros del estado staged

- ¿Qué ocurre si un fichero que ha pasado a modo stage se necesita revertir de ese estado sin deshacer esos cambios?

```
$ git status
```

```
# On branch master
```

```
# Changes to be committed:
```

```
#   (use "git reset HEAD <file>..." to unstage)
```

```
#
```

```
#   modified:   LICENSE
```

```
#
```

```
$ git reset HEAD LICENSE
```

```
Unstaged changes after reset:
```

```
M  LICENSE
```

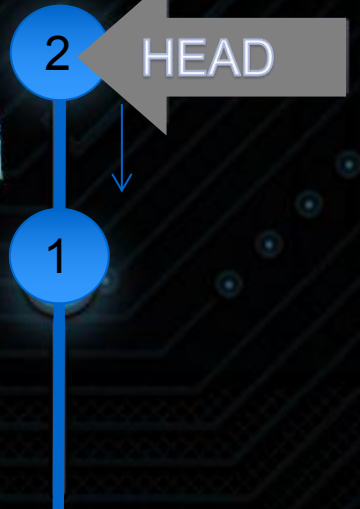
Sugerencia de comando para unstage



HEAD referencia al último commit en la línea temporal actual en la que nos encontramos



master



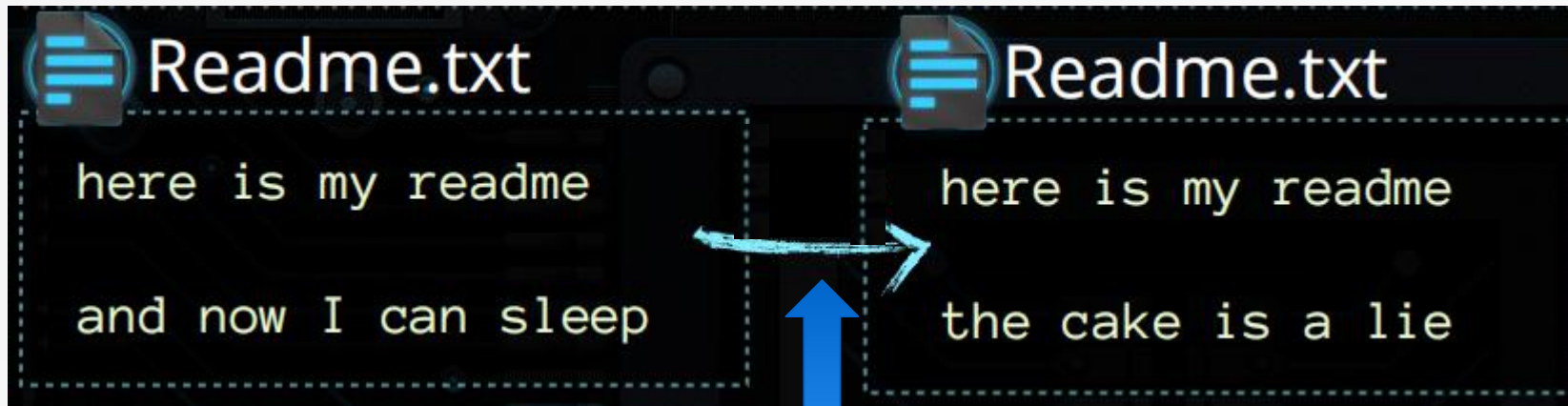
3. Revertir todos los cambios

```
$ git status
# On branch master
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#   modified:   LICENSE
#
$ git checkout -- LICENSE
$ git status
nothing to commit (working directory clean)
```

Desaparecen los cambios realizados sobre ese fichero desde el último commit

4. Saltar Staging

- Hemos modificado un fichero ya versionado Readme.txt



Se edita el fichero

4. Saltar Staging

- GIT nos permite saltar el paso por el área de preparación si así lo vemos conveniente

Añade los cambios de todos los ficheros ya versionados

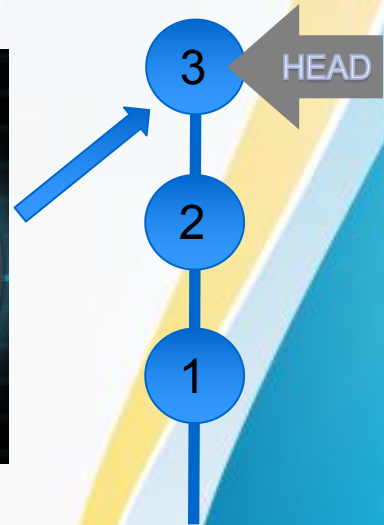
```
$ git commit -a -m "Modify readme"
```

```
[master d00fefc] Modify readme
```

```
1 files changed, 1 insertions(+), 1 deletions(-)
```



¡NO AÑADE LOS NUEVOS FICHEROS!



4. Deshacer un commit

- ¡Vaya!, la hemos liado en nuestro último commit. No hay problema, podemos revertirlo.

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

Resetea todo un commit al estado staged

```
$ git status
```

Esto implica que la línea temporal volverá al commit anterior a HEAD

```
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#   modified:   README.txt
#
```

Ahora ya se pueden volver a hacer cambios y re-commitear



5. Añadir fichero al último commit

- Igual lo que hemos olvidado añadir es un fichero en el último commit. Podemos usar el parámetro *amend*

Unir al último commit

Nuevo mensaje para commit

```
$ git add todo.txt  
$ git commit --amend -m "Modify readme & add todo.txt."  
  
[master fe98ef9] Modify readme and add todo.txt.  
2 files changed, 2 insertions(+), 1 deletions(-)  
create mode 100644 todo.txt
```



3

2

1

TODO LO QUE HAYA PASADO A ESTADO STAGE SE AÑADIRÁ AL ÚLTIMO COMMIT

6. Otros comandos útiles

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

Deshacer último commit, poner cambios como staged

```
$ git commit --amend -m "New Message"
```

Cambiar el último commit

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

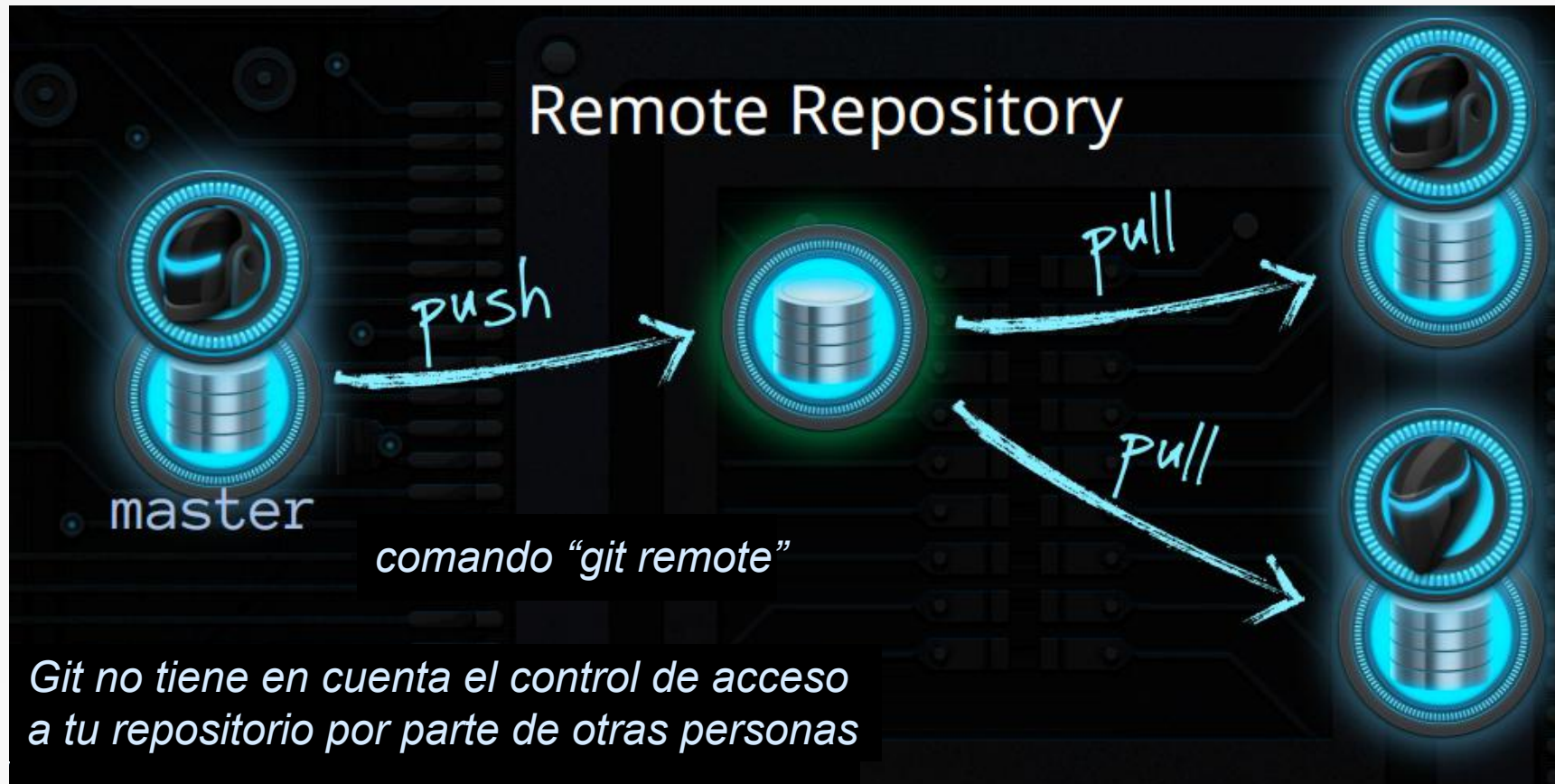
Deshacer el último commit y todos los cambios

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





Deshacer los últimos 2 commits y todos los cambios

7. ¿Cómo compartir?

- GIT fue creado para poder trabajar en equipo, pudiendo éste ser compartido con otras personas

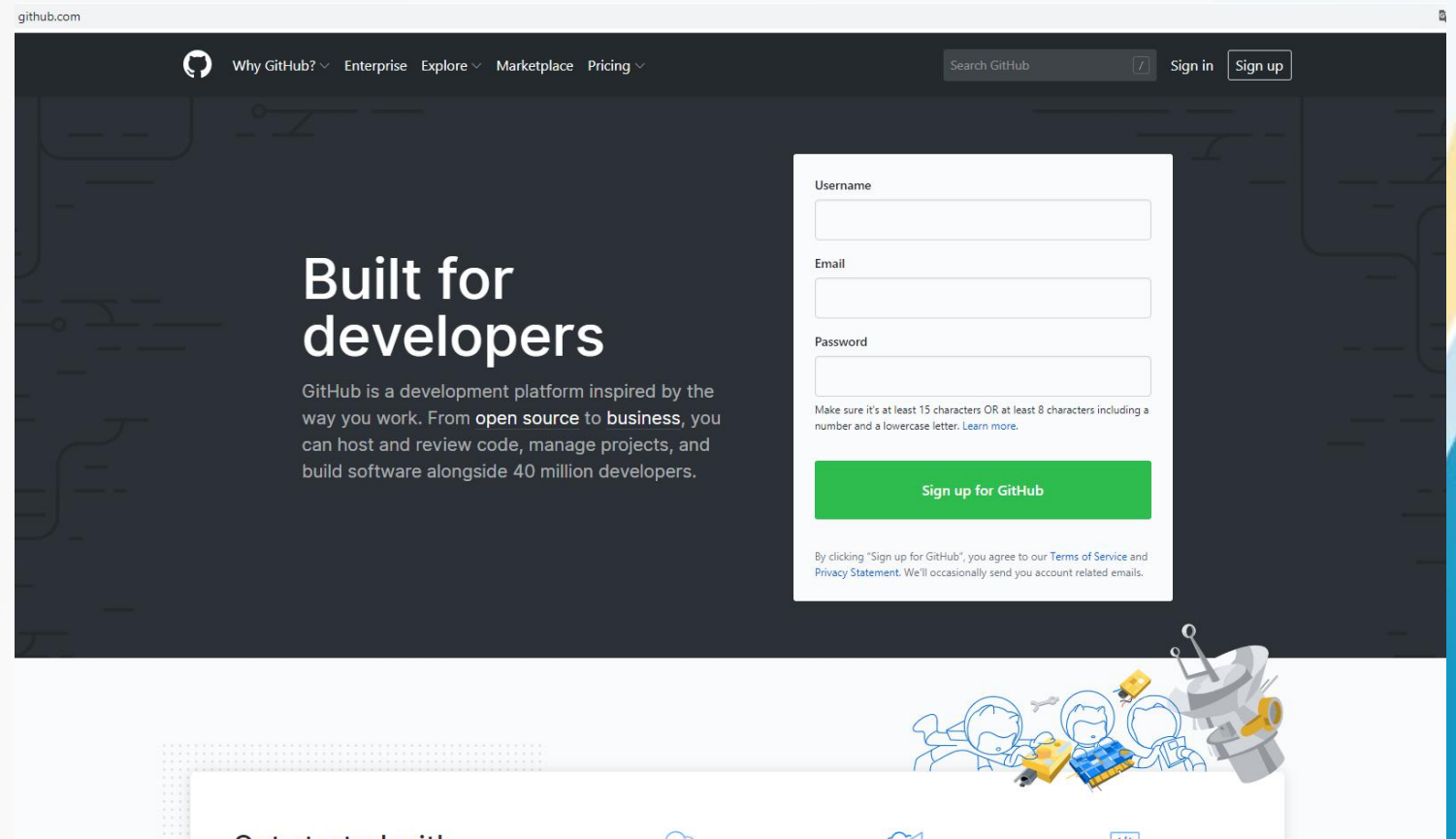


7. ¿Cómo compartir?

- Git no tiene en cuenta el control de acceso a tu repositorio por parte de otras personas
- Soluciones Hosted (autogestionado)
 - GitHub 
 - BitBucket 
- Soluciones Self Managed (gestión propia)
 - Gitis 
 - Gitorious 

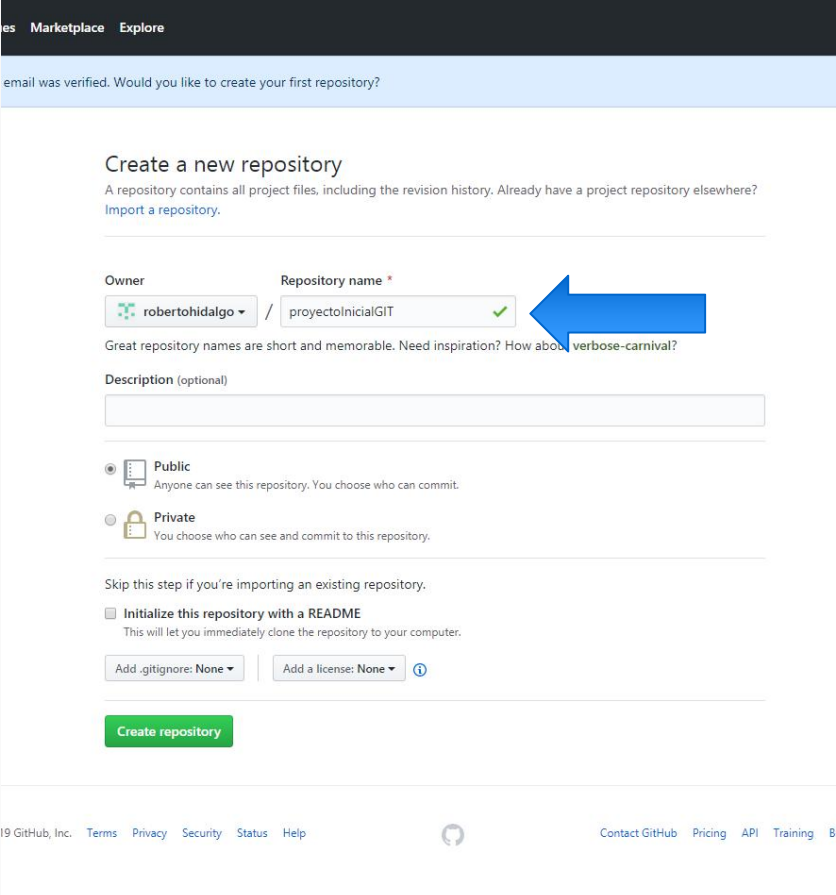
7. ¿Cómo compartir?

- Para compartir en GitHub primero necesitamos tener una cuenta para poder subir todo lo que necesitemos.
- <https://github.com>



7. ¿Cómo compartir?

- Una vez registrados, debemos crear nuestro repositorio en GitHub



es Marketplace Explore

email was verified. Would you like to create your first repository?

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Owner: robertohidalgo / Repository name: proyectoInicialGIT ✓

Great repository names are short and memorable. Need inspiration? How about [verbose-carnival](#)?

Description (optional)

☒ Public
Anyone can see this repository. You choose who can commit.


☐ Private
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☐ Initialize this repository with a README
This will let you immediately clone the repository to your computer.

Add .gitignore: None | Add a license: None ⓘ

[Create repository](#)

© 19 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#)  [Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#)

7. ¿Cómo compartir?

Using the Hero World guide, you'll start a branch, write comments, and open a pull request.

[Read the guide](#)

robertohidalgo / proyectoInicialGIT

Unwatch 1 Star 0 Fork 0

[Code](#) [Issues 0](#) [Pull requests 0](#) [Projects 0](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

Quick setup — if you've done this kind of thing before

[Set up in Desktop](#) or [HTTPS](#) [SSH](#) <https://github.com/robertohidalgo/proyectoInicialGIT.git>

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# proyectoInicialGIT" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/robertohidalgo/proyectoInicialGIT.git
git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/robertohidalgo/proyectoInicialGIT.git
git push -u origin master
```

...or import code from another repository


You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)


7. ¿Cómo compartir?

...or push an existing repository from the command line


```
git remote add origin https://github.com/robertohidalgo/proyectoInicialGIT.git  
git push -u origin master
```



Orden para
enlazar nuestro
repositorio local
con un nuevo
repositorio
remoto



El nombre de nuestro nuevo
repositorio remoto
(puede ser cualquier otro,
aunque por defecto se suele
llamar "origin")



Dirección web donde está alojado
el repositorio remoto


7. ¿Cómo compartir?

- Puede ser que necesitemos saber la lista de remotos que están vinculados a nuestro repositorio local

```
$ git remote -v
```

```
origin https://github.com/robertohidalgo/proyectoInicialGIT.git (fetch)
```

```
origin https://github.com/robertohidalgo/proyectoInicialGIT.git (push)
```



“origin” es un
vínculo a estas
URLs

7. ¿Cómo compartir?


- Para **subir** (push) el repositorio local al remoto se debe usar el comando “push”.

Alias del repositorio remoto

Rama local a subir

```
$ git push -u origin master
```

Username for 'https://github.com': robertohidalgo
Password for 'https://robertohidalgo@github.com':
Contando objetos: 6, listo.
Comprimiendo objetos: 100% (3/3), listo.
Escribiendo objetos: 100% (6/6), 556 bytes | 278.00 KiB/s, listo.
Total 6 (delta 0), reused 0 (delta 0)
To https://github.com/robertohidalgo/proyectoInicialGIT.git
* [new branch] master -> master



7. ¿Cómo compartir?

The screenshot shows a GitHub repository page for 'robertohidalgo / proyectoInicialGIT'. At the top, there are buttons for 'Unwatch' (1), 'Star' (0), and 'Fork' (0). Below this is a navigation bar with tabs for 'Code', 'Issues' (0), 'Pull requests' (0), 'Projects' (0), 'Wiki', 'Security', 'Insights', and 'Settings'. The main content area starts with a message 'No description, website, or topics provided.' and an 'Edit' button. Below this is a section with statistics: '2 commits', '1 branch', '0 releases', and '0 contributors'. A blue arrow points to the '2 commits' text. Underneath are buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and a green 'Clone or download' button. A commit history table follows, showing a commit by 'Roberto' titled 'Añado el documento2' with the latest commit hash '0deb377' and a timestamp of '31 minutes ago'. Below this, two files are listed: 'subdirectorio' (Añado el documento2, 31 minutes ago) and 'documento1' (Añado el documento1, 32 minutes ago). At the bottom of the main content area is a light blue box with the text 'Help people interested in this repository understand your project by adding a README.' and a green 'Add a README' button. The footer contains copyright information '© 2019 GitHub, Inc.', links for 'Terms', 'Privacy', 'Security', 'Status', and 'Help', the GitHub logo, and links for 'Contact GitHub', 'Pricing', 'API', 'Training', 'Blog', and 'About'.

robertohidalgo / proyectoInicialGIT

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Security Insights Settings

No description, website, or topics provided. Edit

Manage topics

2 commits 1 branch 0 releases 0 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

Roberto	Añado el documento2	Latest commit 0deb377 31 minutes ago
subdirectorio	Añado el documento2	31 minutes ago
documento1	Añado el documento1	32 minutes ago

Help people interested in this repository understand your project by adding a README. Add a README

© 2019 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About

7. ¿Cómo compartir?

- Muestra la misma información que “git log”

The screenshot displays the GitHub interface for the repository 'robertohidalgo / proyectoInicialGIT'. At the top, there are buttons for 'Unwatch' (1), 'Star' (0), and 'Fork' (0). Below this is a navigation bar with links for 'Code', 'Issues' (0), 'Pull requests' (0), 'Projects' (0), 'Wiki', 'Security', 'Insights', and 'Settings'. A dropdown menu shows the current branch as 'master'. The main section is titled 'Commits on Oct 11, 2019' and lists two commits:

- Añado el documento2**
Roberto committed 37 minutes ago
Commit hash: 0deb377
- Añado el documento1**
Roberto committed 38 minutes ago
Commit hash: 509483d

Each commit entry includes a copy icon, the commit hash, and a code icon. At the bottom, there are 'Newer' and 'Older' buttons for navigating through the commit history.

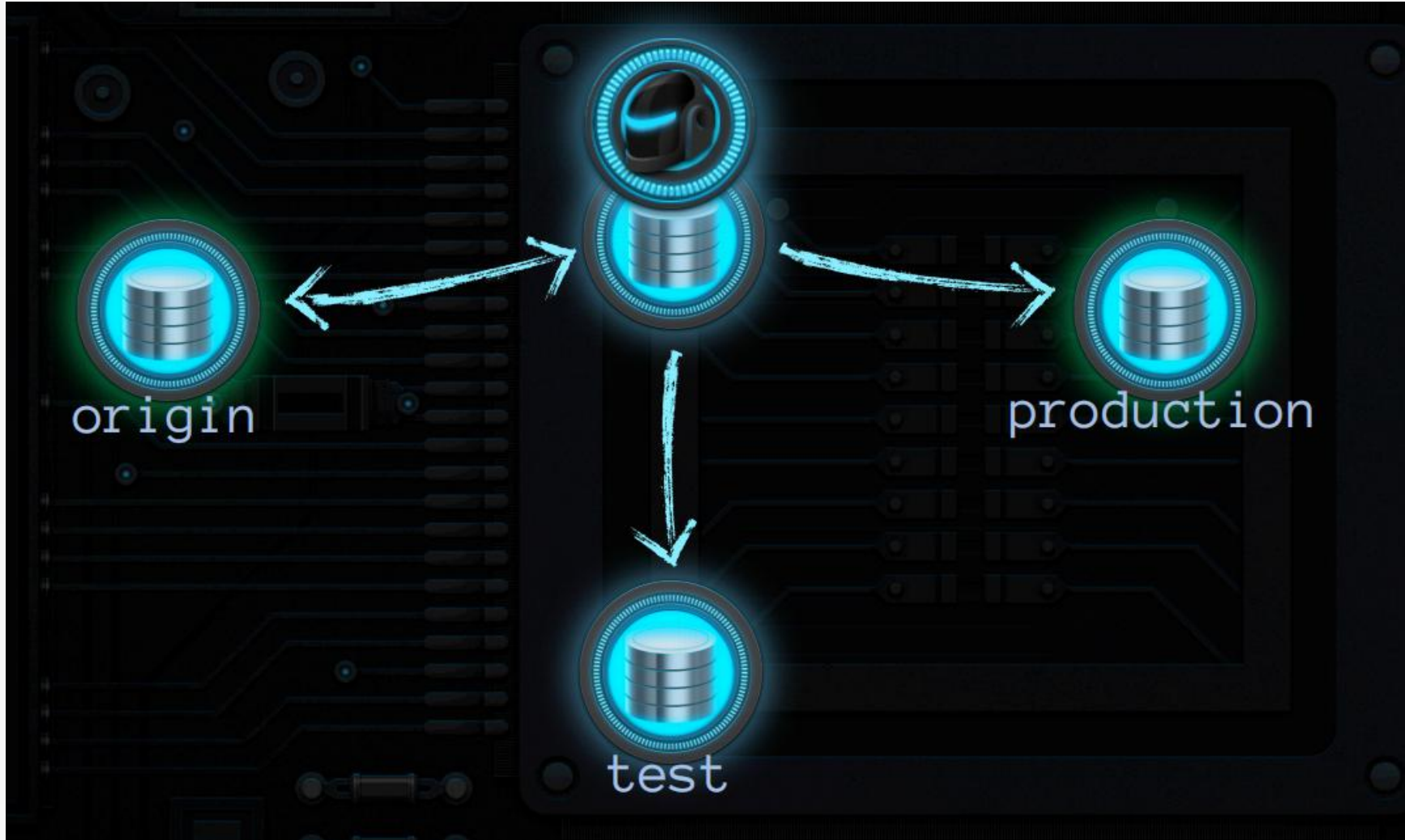
7. ¿Cómo compartir?

- Para **traer** (pull) los datos del repositorio remoto al repositorio local usaremos la instrucción “pull”

```
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Desempaquetando objetos: 100% (3/3), listo.
Desde https://github.com/robertohidalgo/proyectoInicialGIT
  0deb377..e883d0a  master    -> origin/master
Actualizando 0deb377..e883d0a
Fast-forward
 documento1 | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```



8. Trabajo con múltiples remotos



8. Trabajo con múltiples remotos

- Añadir un nuevo remoto

```
$ git remote add <name> <address>
```

- Eliminar un remoto

```
$ git remote rm <name>
```

- Subir cambios a remoto

```
$ git push -u <name> <branch>
```



Normalmente, master

9. Recordatorio de comandos útiles

- Mejor no los hagas después de hacer 'push' al remoto

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

Deshacer último commit, poner cambios como staged

```
$ git commit --amend -m "New Message"
```

Cambiar el último commit

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

Deshacer el último commit y todos los cambios

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

Deshacer los últimos 2 commits y todos los cambios