

git Instalación

Acción	Comando
--------	---------

Instalar Git	Ir a Git SCM
--------------	--------------

Instalar Git en Mac	1) Instalar Homebrew si no está instalado:
---------------------	--

	\$ /bin/bash -c " \$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh) "
--	---

2) En la terminal poner:

\$ brew install git

Ver la versión	\$ git --version
----------------	------------------

Repositorios (repositories)

Acción	Comando
--------	---------

Crear un repositorio*	\$ git init
-----------------------	-------------

Ver la situación actual de los archivos	\$ git status
---	---------------

Añadir un archivo al repositorio	\$ git add file_name.ext
----------------------------------	--------------------------

Añadir todos los archivos que contiene la carpeta	\$ git add . \$ git add -A
---	-------------------------------

Repositorios (repositories) (cont)

Mostrar las diferencias de archivos que no se han enviado aún al staging area	\$ git diff
---	-------------

Borrar el caché de un repositorio	\$ git rm --cached file_name/ext archivo**
-----------------------------------	---

* Primero se abre en la terminal la carpeta que contiene el proyecto del que se quiere inicializar un repositorio: \$ cd folder_with_project

**Esto se hace cuando se quiere agregar un archivo al .gitignore (archivo que permite guardar archivos que no se quiere que sean públicos en el repositorio)

>_ Configuraciones

Acción	Comando
--------	---------

Configurar el nombre de usuario para todo el sistema	\$ git config --global user.name "my_username"
--	--

Configurar el nombre de usuario solo para el repositorio actual	\$ git config user.name "my_username"
---	---------------------------------------



By paulafarias (paulafarias)

Published 5th October, 2020.

Last updated 22nd October, 2020.

Page 1 of 4.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

>_ Configuraciones (cont)

Configurar el email para todo el sistema `$ git config --global user.email "my_email"`

Configurar el email solo para el repositorio actual `$ git config user.email "my_email"`

Habilitar la colorización en la línea de comando `$ git config --global color.ui auto`

Eliminar todas las configuraciones globales realizadas `$ git config --global --unset-all`

Commit - Files

Acción	Comando
--------	---------

Realizar el commit de los archivos que están en staging area	<code>\$ git commit -m "Commit message"</code>
--	--

Mostrar todos los commits realizados	<code>\$ git log</code> <code>\$ git log --oneline</code>
--------------------------------------	--

Revertir el commit a uno anterior*	<code>\$ git revert commit_ID</code>
------------------------------------	--------------------------------------

Commit - Files (cont)

Deshacer todos los commits que se hicieron después del `commit_ID`, preservando los cambios localmente* `$ git reset commit_ID`

Enumerar el historial de versión para un archivo, incluidos los cambios de nombre `$ git log --follow file_name`

Cambiar el nombre de un archivo `$ git mv origin_file_name new_file_name`

*Para revertir o resetear el commit, se necesita el `commit_ID` del commit al cual se quiere volver, el cual lo encontramos cuando buscamos los commits realizados con `$ git log`

Ramas (branches)

Acción	Comando
--------	---------

Crear una nueva rama (branch)	<code>\$ git branch new_branch_name</code>
-------------------------------	--

Ver todas las ramas del repositorio	<code>\$ git branch</code>
-------------------------------------	----------------------------



By paulafarias (paulafarias)

Published 5th October, 2020.

Last updated 22nd October, 2020.

Page 2 of 4.

Sponsored by Readable.com

Measure your website readability!

<https://readable.com>

👤 Ramas (branches) (cont)

Cambiar de rama a donde se quiere subir el archivo

```
$ git checkout branch_name
```

Crear una nueva rama y "commit" a ella

```
$ git checkout -b new_branch_name
```

Volver a la rama master

```
$ git checkout master
```

Unir una rama con la rama activa actualmente

```
$ git merge merge_branch_name
```

Borrar una rama

```
$ git branch -d delete_branch_name
```

Forzar el borrado de una rama (force-delete)

```
$ git branch -D delete_branch_name
```

🔗 Repositorios remotos (GitHub)

Acción	Comando
--------	---------

Conectar un repositorio <i>local</i> con un repositorio <i>remoto</i> (en GitHub) *	<pre>\$ git remote add origin repository_url.git</pre>
---	--

Listar los repositorios <i>remotos</i> a los que se está conectado	<pre>\$ git remote -v</pre>
--	-----------------------------

🔗 Repositorios remotos (GitHub) (cont)

Cambiar la url del repositorio *remoto* al que se está conectado

```
$ git remote set-url origin new_repository_url
```

Actualizar el repositorio *local* para que tenga los últimos cambios del repositorio *remoto*

```
$ git pull origin master
$ git pull origin branch_name
```

Subir los últimos cambios del repositorio *local* al repositorio *remoto*

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

Crear una copia al repositorio *local*, clonando un repositorio *remoto*

```
$ git clone repository_url
```

* Se hace una sola vez para conectar los repositorios.



By paulafarias (paulafarias)

Published 5th October, 2020.

Last updated 22nd October, 2020.

Page 3 of 4.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

☰ Pasos generales para un proyecto

```
1) Abrir en la terminal la carpeta que contiene el proyecto.
$ cd folde_r_w_it_h_pr_oject

2) Crear un repositorio local en la carpeta del proyecto.*
$ git init

3) Añadir al repositorio local los archivos que se encuentran en la
carpeta del proyecto.
$ git add .
$ git add file_ nam e.e xt

4) Realizar el commit de los archivos que están en staging area.
$ git commit -m " Commit messag e"

Si se quiere subir a un repositorio remoto (Github):

5) Conectar el repositorio local con un repositorio remoto (en
Github)**
$ git remote add origin repos_ito_ry_ u rl .g it

6) Subir los últimos cambios del repositorio local al repositorio
remoto
$ git push origin branch _na me
```

* Se crea solo una vez.

** Se conecta solo una vez.

📄 README. File (Markdown Syntax)

Acción	Markdown
Títulos (headers)	# ##
Lista	* -
Checklist	- [] - [x]
Negrita (bold)	** text **
Itálica	<i>_text_</i>
Coloca una línea horizontal	---
Link a un archivo que se encuentra en el mismo repositorio	Texto
Link a una página externa con texto propio	Texto
Mencionar a un usuario (recibe notificación)	@user
Emoji	:emoji_id:



By paulafarias (paulafarias)

Published 5th October, 2020.

Last updated 22nd October, 2020.

Page 4 of 4.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>