

### git Instalación

Acción	Comando
--------	---------

Instalar Git	Ir a <a href="#">Git SCM</a>
--------------	------------------------------

Instalar Git en Mac	1) Instalar <a href="#">Homebrew</a> si no está instalado: <pre>\$ /bin/bash -c "\$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"</pre>
---------------------	--

	2) En la terminal poner: <pre>\$ brew install git</pre>
--	--

Ver la versión	<pre>\$ git --version</pre>
----------------	-----------------------------

### Repositorios (repositories)

Acción	Comando
Crear un repositorio*	<pre>\$ git init</pre>
Ver la situación actual de los archivos	<pre>\$ git status</pre>
Añadir un archivo al repositorio	<pre>\$ git add file_name.ext</pre>
Añadir todos los archivos que contiene la carpeta	<pre>\$ git add . \$ git add -A</pre>
Mostrar las diferencias de archivos que no se han enviado aún al staging area	<pre>\$ git diff</pre>
Borrar el caché de un archivo**	<pre>\$ git rm --cached file_name.ext</pre>

\* 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	<pre>\$ git config --global user.name "my_username"</pre>
Configurar el nombre de usuario solo para el repositorio actual	<pre>\$ git config user.name "my_username"</pre>

### >\_ Configuraciones (cont)

Configurar el email para todo el sistema	<pre>\$ git config --global user.email "my_email"</pre>
Configurar el email solo para el repositorio actual	<pre>\$ git config user.email "my_email"</pre>
Habilitar la colorización en la línea de comando	<pre>\$ git config --global color.ui auto</pre>
Eliminar todas las configuraciones globales realizadas	<pre>\$ git config --global --unset-all</pre>

### Commit - Files

Acción	Comando
Realizar el commit de los archivos que están en staging area	<pre>\$ git commit -m "Commit message"</pre>
Mostrar todos los commits realizados	<pre>\$ git log \$ git log --oneline</pre>
Revertir el commit a uno anterior*	<pre>\$ git revert commit_ID</pre>
Deshacer todos los commits que se hicieron después del <code>commit_ID</code> , preservando los cambios localmente*	<pre>\$ git reset commit_ID</pre>
Enumerar el historial de versión para un archivo, incluidos los cambios de nombre	<pre>\$ git log --follow file_name</pre>
Cambiar el nombre de un archivo	<pre>\$ git mv original_file_name new_file_name</pre>

\* 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)	<pre>\$ git branch new_branch_name</pre>
Ver todas las ramas del repositorio	<pre>\$ git branch</pre>



### 👤 Ramas (branches) (cont)

Cambiar de rama a donde se quiere subir el archivo	<code>\$ git checkout branch_name</code>
Crear un nueva rama y "cambiar" a ella	<code>\$ git checkout -b new_branch_name</code>
Volver a la rama master	<code>\$ git checkout master</code>
Unir una rama con la rama activa actualmente	<code>\$ git merge merging_branch_name</code>
Borrar una rama	<code>\$ git branch -d deleting_branch_name</code>
Forzar el borrado de una rama (force-delete)	<code>\$ git branch -D deleting_branch_name</code>

### 🌐 Repositorios remotos (GitHub)

Acción	Comando
Conectar un repositorio <i>local</i> con un repositorio <i>remoto</i> (en GitHub) *	<code>\$ git remote add origin repository_url.git</code>
Listar los repositorios <i>remotos</i> a los que se está conectado	<code>\$ git remote -v</code>
Cambiar la url del repositorio <i>remoto</i> al que se está conectado	<code>\$ git remote set-url origin new_repository_url</code>
Actualizar el repositorio <i>local</i> para que tenga los últimos cambios del repositorio <i>remoto</i>	<code>\$ git pull origin master</code> <code>\$ git pull origin branch_name</code>
Subir los últimos cambios del repositorio <i>local</i> al repositorio <i>remoto</i>	<code>\$ git push -u origin master</code> <code>\$ git push origin branch_name</code>
Crear una copia al repositorio <i>local</i> , clonando un repositorio <i>remoto</i>	<code>\$ git clone repository_url</code>

\* Se hace una sola vez para conectar los repositorios.

### ☰ Pasos generales para un proyecto

- 1) Abrir en la terminal la carpeta que contiene el proyecto.  
`$ cd folder_with_project`
  - 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_name.ext`
  - 4) Realizar el commit de los archivos que están en *staging area*.  
`$ git commit -m "Commit message"`
- Si se quiere subir a un repositorio remoto (Github):**
- 5) Conectar el *repositorio local* con un *repositorio remoto* (en GitHub)\*\*  
`$ git remote add origin repository_url.git`
  - 6) Subir los últimos cambios del *repositorio local* al *repositorio remoto*  
`$ git push origin branch_name`

\* Se crea solo una vez.

\*\* Se conecta solo una vez.

### 📄 README. File (Markdown Syntax)

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



By **paulafarias** (paulafarias)

Published 5th October, 2020.

Last updated 22nd October, 2020.

Page 2 of 2.

Sponsored by **ApolloPad.com**

Everyone has a novel in them. Finish

Yours!

<https://apollopadd.com>