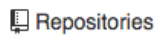




# Trabajar con Git y GitHub

Para trabajar con este software de control de versiones, primero crearemos una cuenta en <https://github.com> y nos instalaremos el cliente de **git** (ver los enlaces al final del tutorial).

Una vez creada la cuenta en github.com para hospedar nuestro repositorio, vamos a la pestaña:

 Repositories

y creamos un nuevo repositorio. Será el repositorio “remoto” (el que está en github.com):


 New

El trabajo del día a día lo haremos en nuestra máquina local (en la copia local del repositorio), y de cuando en cuando iremos subiendo nuestro trabajo al repositorio remoto.

Sólo tenemos que inventarnos el nombre del repositorio. Será público (gratuito), y querremos que lo inicialice con un archivo README, aunque esté vacío al principio:

Owner

Repository name


 pacastillo ▾

 / 


swap1415 ✓

Great repository names are short and memorable. Need inspiration? [Click here](#)

Description (optional)

☒  Public

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

☐  Private


You choose who can see and commit to this repository.

☒ Initialize this repository with a README

This will let you immediately clone the repository to your computer. Skip

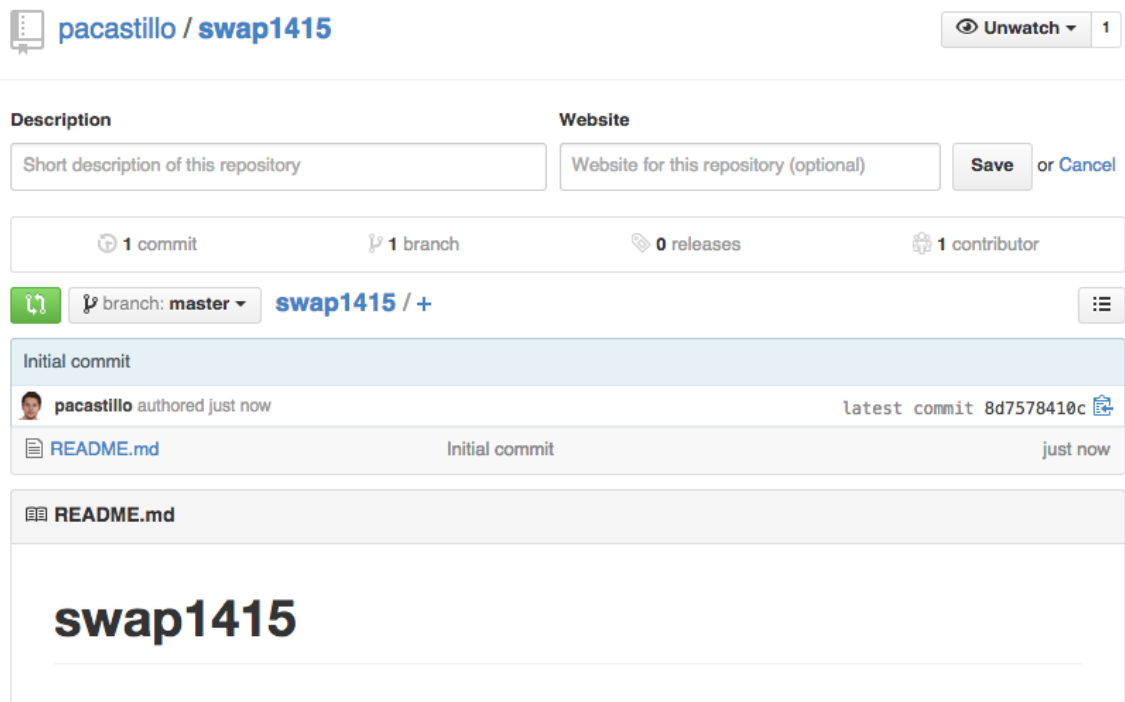
Add .gitignore: **None** ▾

 | 

Add a license: **None** ▾ 

Create repository

Una vez creado, nos mostrará el contenido en ese momento. Sólo tendremos el archivo README:



The screenshot shows the GitHub interface for a repository named 'swap1415' by user 'pacastillo'. At the top, there's a header with the repository name and a 'Unwatch' button. Below this, there are two input fields: 'Description' (with placeholder text 'Short description of this repository') and 'Website' (with placeholder text 'Website for this repository (optional)'). To the right of these fields are 'Save' and 'Cancel' buttons. Below the input fields, there's a summary bar showing '1 commit', '1 branch', '0 releases', and '1 contributor'. Underneath this, there's a green 'Clone' button, a dropdown menu for 'branch: master', and a 'swap1415 / +' button. Below the clone section, there's a section for 'Initial commit' showing that 'pacastillo' authored it 'just now'. It also shows the 'latest commit 8d7578410c' and a link to the commit. Below this, there's a section for 'README.md' showing the initial commit. The main content area displays the 'README.md' file, which contains the text 'swap1415'.

Ya tenemos el repositorio preparado. Ahora podemos trabajar en nuestra máquina en local e ir subiendo las cosas al repositorio remoto (en github.com).

Para ello, copiamos la dirección de nuestro repositorio:

HTTPS clone URL

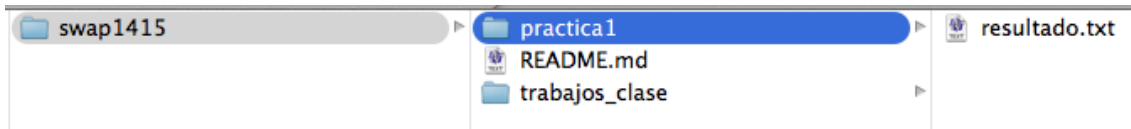
<https://github.com>



Y en el terminal de nuestro ordenador ejecutamos el git clone:

```
mac:tmp pedro$
mac:tmp pedro$ git clone https://github.com/pacastillo/swap1415.git
Cloning into 'swap1415'...
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
Checking connectivity... done.
mac:tmp pedro$
mac:tmp pedro$ ls -la swap1415/
total 8
drwxr-xr-x  4 pedro  staff  136 28 feb 23:59 .
drwxr-xr-x  4 pedro  staff  136 28 feb 23:59 ..
drwxr-xr-x 13 pedro  staff  442 28 feb 23:59 .git
-rw-r--r--  1 pedro  staff   11 28 feb 23:59 README.md
mac:tmp pedro$
mac:tmp pedro$
mac:tmp pedro$
```

Ahora, en este tutorial, vamos a crear varias carpetas y archivos. Por ejemplo, la carpeta donde vamos a ir entregando los trabajos de clase y la carpeta para la práctica 1:



Antes que nada, debemos añadir esos archivos y carpetas al repositorio local, para que estén bajo el control de git. Si no, no podremos subirlas al repositorio remoto en github.com. Para añadir archivos, usamos el comando “add”, y para confirmar cualquier acción sobre el repositorio, el comando “commit” del git:

```
mac:swap1415 pedro$
mac:swap1415 pedro$ git add practical1/resultado.txt
mac:swap1415 pedro$ git add trabajos_clase/leeme.txt
mac:swap1415 pedro$
mac:swap1415 pedro$ git commit -a -m "comentario describiendo lo que subimos"
[master 48ae6a5] comentario describiendo lo que subimos
 2 files changed, 1 insertion(+)
 create mode 100644 practical1/resultado.txt
 create mode 100644 trabajos_clase/leeme.txt
mac:swap1415 pedro$
```

Con esto ya están bajo el control de git, pero aún no las hemos subido al remoto. Para ello, usaremos el comando “push” del git:

```
mac:swap1415 pedro$
mac:swap1415 pedro$ git push
warning: push.default is unset; its implicit value is changing in
Git 2.0 from 'matching' to 'simple'. To squelch this message
and maintain the current behavior after the default changes, use:

    git config --global push.default matching

To squelch this message and adopt the new behavior now, use:

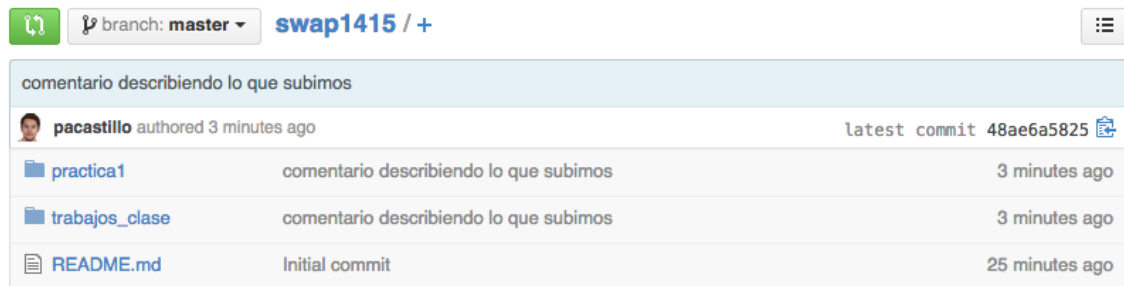
    git config --global push.default simple

See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

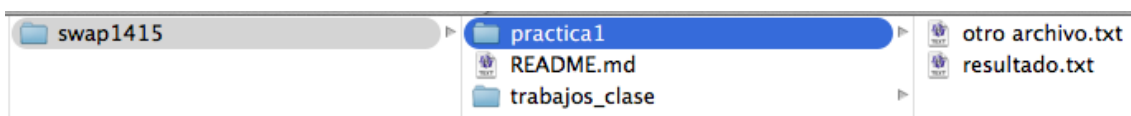
Counting objects: 7, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (6/6), 469 bytes | 0 bytes/s, done.
Total 6 (delta 0), reused 0 (delta 0)
To https://github.com/pacastillo/swap1415.git
 8d75784..48ae6a5  master -> master
mac:swap1415 pedro$
mac:swap1415 pedro$ _
```

Obtendremos muchos mensajes y diversa información. El programa pedirá el nombre de usuario y contraseña para poder subir contenido al repositorio remoto. En la captura anterior no aparecen los mensajes correspondientes ya que para mi cuenta tengo configuradas las claves SSH de forma que no me solicite la identificación más.

Todo debe funcionar correctamente y quedar almacenado en el repositorio en github.com. Podemos comprobarlo desde la interfaz web:



Con esto habríamos entregado la práctica 1 y un trabajo de clase. Si tras subirlo al servidor queremos modificar algo y actualizarlo en el repositorio remoto, sólo debemos trabajar con el archivo que corresponda en nuestra máquina en nuestra carpeta (en el repositorio local bajo el control de git):



Luego volvemos a usar los comandos “commit” y “push” del git:

```
mac:swap1415 pedro$
mac:swap1415 pedro$ git add practica1/otro\ archivo.txt
mac:swap1415 pedro$ git commit -a -m "añadido un archivo y modificado el primero"
[master 59b3fdd] añadido un archivo y modificado el primero
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 practica1/otro archivo.txt
mac:swap1415 pedro$ git push
warning: push.default is unset; its implicit value is changing in
Git 2.0 from 'matching' to 'simple'. To squelch this message
and maintain the current behavior after the default changes, use:

    git config --global push.default matching

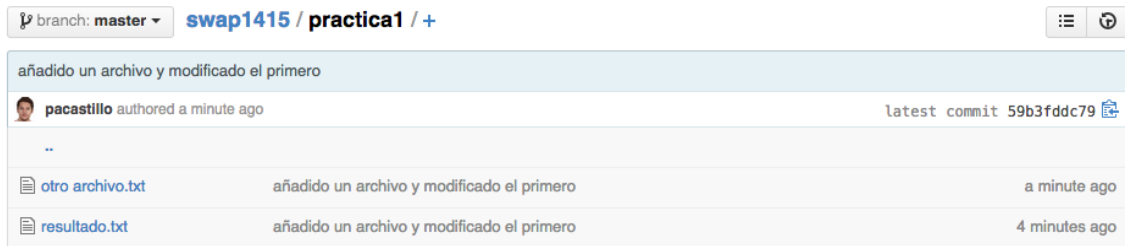
To squelch this message and adopt the new behavior now, use:

    git config --global push.default simple

See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 428 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/pacastillo/swap1415.git
a47fcc5..59b3fdd master -> master
mac:swap1415 pedro$ _
```

Todo se actualiza en el repositorio remoto:



Finalmente, si quisiéramos eliminar uno de los archivos, usamos el comando “rm” del git, con lo que lo eliminamos del sistema de archivos local, y luego lo confirmaremos con el comando “commit” y con un “push”:

```
mac:swap1415 pedro$
mac:swap1415 pedro$ git rm practical1/otro\ archivo.txt
rm 'practical1/otro archivo.txt'
mac:swap1415 pedro$ git commit -a -m "borrar un archivo"
[master 0abae3e] borrar un archivo
 1 file changed, 0 insertions(+), 0 deletions(-)
 delete mode 100644 practical1/otro archivo.txt
mac:swap1415 pedro$ git push
warning: push.default is unset; its implicit value is changing in
Git 2.0 from 'matching' to 'simple'. To squelch this message
and maintain the current behavior after the default changes, use:

    git config --global push.default matching

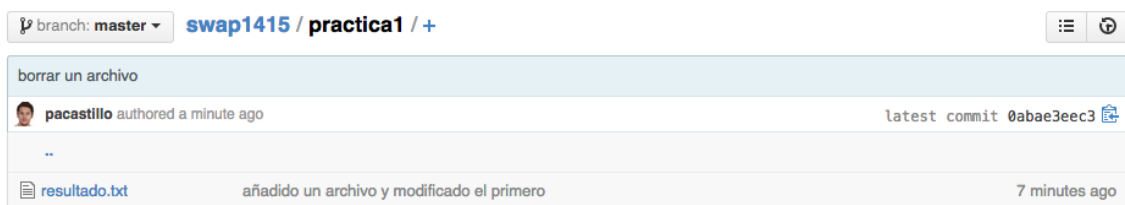
To squelch this message and adopt the new behavior now, use:

    git config --global push.default simple

See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 370 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/pacastillo/swap1415.git
 59b3fdd..0abae3e master -> master
mac:swap1415 pedro$
mac:swap1415 pedro$
```

Y en el repositorio remoto desaparecerá el archivo:



Esta herramienta de control de versiones es tremendamente útil. Tiene muchas más funciones que éstas básicas comentadas aquí. En los enlaces al final de este documento se puede encontrar información adicional sobre cómo obtener el software cliente de git y sobre las posibilidades de esta herramienta.

## **Referencias**

<http://es.wikipedia.org/wiki/Git>

<http://rogerdudler.github.io/git-guide/index.es.html>

<http://gitimmersion.com/>

<http://git-scm.com/book/es/v1>

<http://www.psicobyte.com/descargas/ZenDeGit2.pdf>

<http://www.alvaroremesal.net/blog-alvaroremesal/recuperar-archivos-antigos-con-git>

<http://algunostutoriales.blogspot.com.es/2014/12/introduccion-git-chuletario-basico.html>

[https://github.com/oslugr/curso-git/blob/master/texto/uso\\_basico.md](https://github.com/oslugr/curso-git/blob/master/texto/uso_basico.md)

<http://guides.beanstalkapp.com/version-control/git-on-windows.html>

<https://msysgit.github.io/>

<http://www.thegeekstuff.com/2011/08/git-install-configure/>

<http://git-scm.com/download/linux>

<https://code.google.com/p/git-osx-installer/>

<http://guides.beanstalkapp.com/version-control/git-on-mac.html>

<http://ivanprego.com/programacion/como-instalar-y-configurar-git-en-mac-os-x/>

<http://coolestguidesontheplanet.com/install-update-latest-version-git-mac-osx-10-9-mavericks/>



# Trabajar con Markdown en GitHub



Lo habitual es subir a github.com archivos de código (texto plano). En esos casos, cuando accedemos a través de la interfaz web a esos archivos subidos, el sistema web nos muestra el contenido de forma similar a:



Es exactamente el archivo que tenemos en la copia del repositorio que almacenamos (en local) en nuestra máquina.

Sin embargo, hay muchas ocasiones en las que nos conviene presentar la información de una forma más completa y más cuidada, con formatos de letra, datos tabulados, incluyendo enlaces e imágenes.

En nuestro repositorio podemos subir archivos de gráficos (PNG o JPG, p.ej.) y archivos Markdown (extensión .md) que github.com nos mostrará aplicando los formatos que establezcamos.

## El lenguaje Markdown

Markdown es un lenguaje de marcado muy sencillo y ligero. Básicamente, lo que hace es convertir el texto marcado en documentos XHTML bien formados. Este lenguaje se basa en el uso de ciertos caracteres para incluir y usar ciertos tipos de formatos.

Por ejemplo, anteponiendo el carácter # a una línea de texto, convertiremos esa línea en una cabecera de primer nivel. Si anteponeamos ## estaremos indicando que ese texto lo queremos como cabecera de segundo nivel:

```
# Cabecera de primer nivel
## Cabecera de segundo nivel
```

Si una palabra o texto queda encerrada entre dos caracteres \* estaremos poniéndola en cursiva, mientras que si la encerramos entre doble \*\* estaremos poniéndola en negrita:

```
*cursiva*
**negrita**
```

También podemos enlazar otras páginas o archivos, de forma que cuando github.com nos muestre el archivo .md formateado, veamos y podamos pinchar sobre un enlace (como estamos acostumbrados) para ir a otra página. La sintaxis para incluir un enlace es:

```
texto fuera del [enlace](http://www.dominio.com/carpeta/archivo.html)
```



De la misma forma, github.com nos puede mostrar imágenes en el texto formateado, si las incluimos en el archivo .md con el formato:

```
! [img] (http://www.dominio.com/carpeta/archivo.png)
```

Existen muchos más formatos que podemos usar en un archivo .md pero con los vistos tendremos más que suficiente para crear documentos bien presentados para entregar los trabajos y prácticas. Al final de este mini-tutorial incluimos una serie de enlaces donde aprender todos los detalles del lenguaje, extensiones, etc.

## Un ejemplo de uso

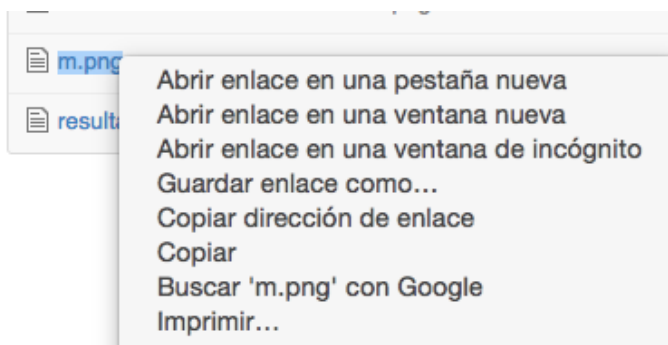
Vamos a hacer un ejemplo completo suponiendo que para la entrega de nuestra práctica-1 queremos entregar un archivo .md en el que mostremos los resultados con formatos de texto bien presentados e incluyendo una tabla de datos y una imagen.

Para ello, vamos a subir a la carpeta “practical1” de nuestro repositorio de la asignatura un archivo de texto plano con extensión .md y un archivo de imagen con extensión .PNG. El proceso a seguir es el mismo que vimos en el primer tutorial sobre git+github

Una vez que los tengamos subidos a nuestro repositorio remoto en github.com:



podemos obtener el enlace a la imagen (opción “Copiar enlace” de nuestro navegador):



Ahora estamos en disposición de editar el archivo .md que tenemos en nuestro repositorio para especificar los formatos, incluir un enlace, una imagen y una tabla.

En la primera parte de nuestro archivo Markdown incluiremos una cabecera de primer nivel (línea 1), texto plano sin formato (línea 4), una cabecera de segundo nivel (línea 6), una palabra en cursiva (línea 8), otra palabra en negrita (línea 10), y según la sintaxis



que hemos visto antes, un enlace a Google (línea 12) e incluiremos una imagen que hemos subido al repositorio y de la que hemos obtenido la URL (línea 16):

```
fich.md | Adiós! | #
1 # El título
2
3 Esto es un ejemplo de archivo que incluye tipografías, formatos, etc usando
4 el lenguaje Markdown.
5
6 ## El subtítulo
7
8 Esto va en cursiva
9
10 Esto va en negrita
11
12 Aquí enlazamos a la [web de google](http://www.google.com)
13
14 Aquí incluimos una imagen
15
16 ![imagen](https://github.com/pacastillo/swap1415/blob/master/practical1/m.png)
17
18
```

Esta primera parte del archivo “fich.md” se nos mostrará en github.com como sigue:



En la segunda parte de dicho archivo, vamos a incluir dos tablas (formato un tanto especial usando guiones, | y dos puntos) y una cabecera de tercer nivel (formato ####):

```

18
19 Y aquí terminamos poniendo una tabla:
20
21 dog | bird | cat
22 :-- | :--: | --:
23 foo | foo | foo
24 bar | bar | bar
25 baz | baz | baz
26
27
28 | Left align | Right align | Center align |
29 :-----: | :-----: | :-----: |
30 This      |          This      |      This
31 column    |        column    |    column
32 will     |         will     |    will
33 be       |          be      |    be
34 left     |        right    |   center
35 aligned  |        aligned  |   aligned
36
37
38 y con texto más grande
39
40 ### ¡Adiós!
41

```

Esta segunda parte del archivo “fich.md” se nos mostrará en github.com como sigue:

Y aquí terminamos poniendo una tabla:

dog	bird	cat
foo	foo	foo
bar	bar	bar
baz	baz	baz

Left align	Right align	Center align
This	This	This
column	column	column
will	will	will
be	be	be
left	right	center
aligned	aligned	aligned

y con texto más grande

**¡Adiós!**

Este lenguaje tiene una gran funcionalidad, siendo sumamente flexible, sencillo y ligero. Para conocer más detalles acerca del mismo, mostramos en la siguiente sección un buen número de páginas con tutoriales y páginas de ayuda.

## **Referencias**

<http://en.wikipedia.org/wiki/Markdown>

<http://es.wikipedia.org/wiki/Markdown>

<http://daringfireball.net/projects/markdown/>

<https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>

<http://markdowntutorial.com/lesson/1/>

<https://guides.github.com/features/mastering-markdown/>

<https://help.github.com/articles/markdown-basics/>

<https://confluence.atlassian.com/display/STASH/Markdown+syntax+guide>

<http://xlson.com/2010/11/09/getting-started-with-github-pages.html>

<https://pages.github.com>

[http://www.tablesgenerator.com/markdown\\_tables](http://www.tablesgenerator.com/markdown_tables)