



# 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

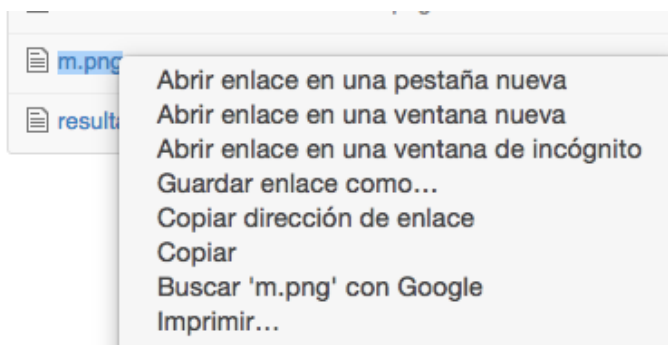
Vamos a hacer un ejemplo completo suponiendo que para la entrega de nuestra práctica-1 queremos entregar un archivo .md en el que mostremos los resultados con formatos de texto bien presentados e incluyendo una tabla de datos y una imagen.

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)