

Git y Github

Github pages



Índice

Introducción	3
Sitios dinámicos, sitios estáticos y Jekyll	3
¿Por qué usar sitios estáticos?	4
GitHub Pages para usuarios u organizaciones	5
GitHub Pages para proyectos	5

Introducción

GitHub Pages es un servicio de GitHub que nos permite alojar nuestros proyectos y mostrarlos en vivo en una página web estática sin necesidad de pagar por hosting o siquiera tener conocimientos en servidores o DevOps.

Es algo muy sencillo y que no toma más de 5 minutos configurar.

Las páginas de GitHub son páginas web públicas alojadas y publicadas a través de GitHub. La forma más rápida de ponerse en marcha es usar el Selector de temas de Jekyll para cargar un tema prefabricado. Luego puede modificar el contenido y el estilo de sus páginas de GitHub.

Página oficial:

<https://docs.github.com/en/pages>

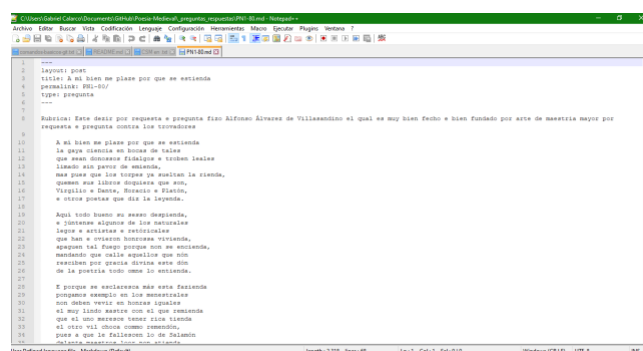
Sitios dinámicos, sitios estáticos y Jekyll

Tanto los *sitios web dinámicos*, como los que son creados y administrados por gestores de contenidos tales como *Drupal*, *WordPress* y *Omeka* extraen información de una base de datos para completar el contenido de una página web.

Cuando buscamos un libro en Amazon.com, por ejemplo, la página de resultados de búsqueda no existe como una página HTML completa; en cambio, Amazon.com tiene una plantilla para la página de resultados de búsqueda, que incluye elementos que comparten todas las páginas de resultados (como el menú principal y el logotipo de Amazon) y consulta una base de datos para insertar en esa plantilla los resultados de la búsqueda que nosotros realizamos.

Los *sitios web estáticos* no usan una base de datos para almacenar información; por el contrario, toda la información que se muestra en cada página web ya está contenida en tu correspondiente archivo HTML. Las páginas HTML que componen un sitio estático se pueden escribir completamente a mano o se puede delegar parte de este trabajo usando una herramienta como Jekyll.

Jekyll es un software que nos ayuda a “generar” o crear un *sitio web estático* (Jekyll muchas veces es caracterizado como un “generador de sitios estáticos”). Jekyll utiliza plantillas de página para aquellos elementos como menús principales y pies de página que normalmente se repiten en todas las páginas que componen un sitio y por ende consumen mucho tiempo si escribimos manualmente el HTML para incluirlos en cada página web. Con Jekyll, estas plantillas se combinan con otros archivos con información específica (por ejemplo, un archivo para cada post de un blog) para generar páginas HTML completas para los visitantes de un sitio. Jekyll no necesita consultar bases de datos ni crear una nueva página HTML (o completar una parcial) cuando los usuarios visitan una página web, sino que ya cuenta con las páginas HTML completas y solo las actualiza cuando/si alguna vez cambian.



```

1  ---
2  layout: post
3  title: A mi hijo me piace por que se acurando
4  permalink: /hi-0/
5  type: post
6  ---
7
8  Rubrica: Esta obra por requesta a requesta fira Alfonso Álvarez de Villaseñor el qual se muy bien hecho e bien fundado por arte de maestria mayor por
9  requesta a requesta contra los curanderos.
10
11  A mi hijo me piace por que se acurando
12  la que ciencia en boca de tales
13  que sean hombres fideles e viciosos lasias
14  limado sin pavor de escucha,
15  mas para que los viciosos se escucha la ciencia,
16  quemen sus libros dequiere que son,
17  Virgilio e Dante, Horacio e Plauto,
18  e otros poetas que dia la leyenda.
19
20  Aqui todo bueno es rezo dequiere,
21  e algunos algunos de los curanderos
22  laque e aristas e reuicorias
23  que son e curanderos viciados,
24  apagan tal fuego por que son se escucha,
25  mandando que vicia apague que son
26  reuicorias por quere diera esta dia
27  de la guerra todo uno la escucha.
28
29  E porque se acuranderos más esta faciendo
30  programo ejemplo en los monasterios
31  son malos vicios en buena iguiera
32  el muy lindo marte con el que reuicoria
33  que el muy marte reuicoria fira laque
34  el otro vil obra como reuicoria,
35  para e que la faliendo la de faliendo
36  ---
37  ---
38  ---
39  ---
40  ---
41  ---
42  ---
43  ---
44  ---
45  ---
46  ---
47  ---
48  ---
49  ---
50  ---
51  ---
52  ---
53  ---
54  ---
55  ---
56  ---
57  ---
58  ---
59  ---
60  ---
61  ---
62  ---
63  ---
64  ---
65  ---
66  ---
67  ---
68  ---
69  ---
70  ---
71  ---
72  ---
73  ---
74  ---
75  ---
76  ---
77  ---
78  ---
79  ---
80  ---
81  ---
82  ---
83  ---
84  ---
85  ---
86  ---
87  ---
88  ---
89  ---
90  ---
91  ---
92  ---
93  ---
94  ---
95  ---
96  ---
97  ---
98  ---
99  ---
100  ---

```

Código de una página de Jekyll en formato md



Código de la misma página de Jekyll pero en formato html, tal como puede verse en el explorador

Hay que tener en cuenta que cuando alguien se refiere a un “sitio web de Jekyll”, en realidad se refiere a un sitio web estático (HTML puro) que se ha creado utilizando Jekyll. Jekyll es un software que crea sitios web. Jekyll no está “ejecutando” el sitio en vivo, sino que es un “generador de sitios estáticos”: es una herramienta que crea los archivos del sitio estático que luego son alojados en un servidor web, como se hace con cualquier otro sitio web HTML.

Dado que los sitios estáticos no son más que archivos de texto (sin una base de datos que complique las cosas), es posible *versionarlos* fácilmente, es decir, usar una herramienta para llevar un registro de las diferentes versiones del sitio a lo largo del tiempo rastreando los cambios en los archivos de texto que lo componen.

El control de versiones es muy útil cuando deseamos combinar ambas versiones (por ejemplo, dos estudiantes escriben una publicación de blog juntos y deseamos combinar sus dos versiones) o cuando queremos comparar archivos para buscar diferencias entre ellos (por ejemplo, “¿Cómo se describía el proyecto en la página ‘Acerca de’ original?”). El control de versiones es muy útil cuando se trabaja en equipo (por ejemplo, permite combinar y rastrear el trabajo de diferentes personas), pero también es útil al crear o ejecutar un sitio web por nuestra propia cuenta.

¿Por qué usar sitios estáticos?

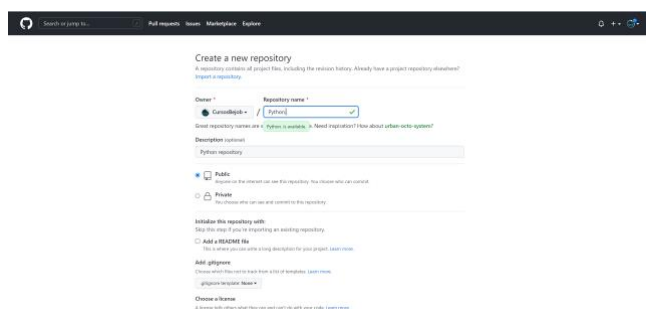
Opciones como *Drupal*, *WordPress* y *Omeka* son útiles para la creación de sitios web complejos e interactivos como Amazon o una edición digital interactiva de una novela, pero para muchos blogs, sitios web de proyectos y portafolios en línea, un sitio web estático (como un sitio web creado con Jekyll) puede hacer todo lo que se necesita al mismo tiempo que proporciona algunas ventajas:

- **Mantenimiento:** Las actualizaciones y el mantenimiento necesitan realizarse con mucha menos frecuencia (menos de una vez al año).
- **Preservación:** Al no emplear base de datos, los archivos de texto que componen tu sitio son todo lo que necesitamos guardar para preservar y replicarlo. Resulta sencillo hacer un respaldo del sitio o enviarlo a un repositorio institucional.
- **Aprendizaje:** Debido a la ausencia de base de datos y a que no hay una gran cantidad de archivos de código que brinden funciones que quizás ni siquiera necesitamos, hay muchos menos componentes en nuestro sitio web; es más fácil revisarlos y saber lo que hace cada uno. Es mucho más fácil convertirse en un usuario de Jekyll básico y avanzado.
- **Más personalización:** Dado que aprender a dominar nuestro sitio web es fácil, cosas que definitivamente queremos hacer, como cambiar el aspecto (el “tema”) de un sitio creado por *Jekyll*, resultan mucho más sencillas que alterar el aspecto de un sitio en *WordPress* o *Drupal*.
- **Alojamiento gratuito:** Si bien muchas herramientas de sitios web como *Drupal*, *WordPress* y *Omeka* son gratuitas, alojarlas (pagar a alguien para que muestre los archivos de tu sitio web a los visitantes del sitio) puede costar dinero.

- **Control de versiones:** Hospedar en *GitHub Pages* significa que nuestro sitio está vinculado a la interfaz visual de *GitHub* para el control de versiones de *git*, por lo que puede realizarse un seguimiento de los cambios en nuestro sitio y, si fuera necesario, volver al estado anterior de cualquier publicación de blog, o página. Esto incluye archivos cargados que tal vez deseemos almacenar en el sitio, como programas de estudio y publicaciones antiguas.
- **Seguridad:** No hay una base de datos a la que haya que proteger de posibles ataques maliciosos.
- **Velocidad:** Los archivos mínimos del sitio web y la inexistencia de una base de datos para consultar resultan en un tiempo de carga de página más rápida.

GitHub Pages para usuarios u organizaciones

Empecemos creando un sitio web para nuestro usuario u organización. Para ello tenemos que ir a GitHub y crear un nuevo repositorio.



Dicho repositorio forzosamente debe de empezar con nuestro nombre de usuario u organización seguido de github.io.

Por ejemplo: nombreDeUsuario.github.io

A partir de esto podemos empezar a generar nuestros *commits*. Tal como lo haríamos con cualquier repositorio; ya sea clonando nuestro proyecto:

```
$ git clone
https://github.com/nombreDeUsuario/nombr
eDeUsuario.github.io
```

o haciendo una vinculación:

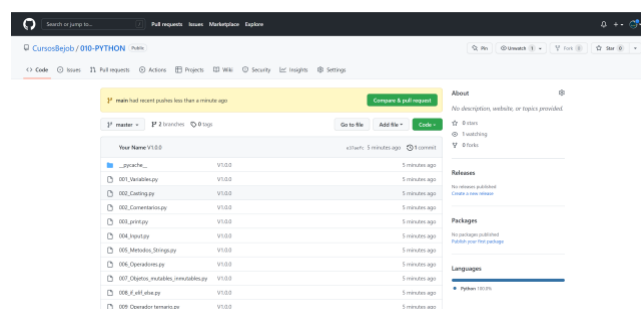
```
miPC@miPC MINGW64 /g/WORKSPACE/011
GitHub/Proyecto GIT (master)

$ git remote add origin
https://github.com/nombreDeUsuario/nombr
eDeUsuario.github.io
```

¡Y listo! Una vez hecho esto, todos los cambios dentro del repositorio remoto (*GitHub*) los podemos ir viendo en [**http://nombreDeUsuario.github.io](http://nombreDeUsuario.github.io).

GitHub Pages para proyectos

Como podemos ver, generar un sitio web con *GitHub* es bastante fácil. Pero no sólo lo podemos hacer para nuestro usuario u organización, también podemos usarlo con nuestros proyectos. Para ello, tenemos que ir a *GitHub* y crear un nuevo repositorio, o usar uno que ya hayamos creado.



Cuando empecemos a crear nuestros *commits* en este repositorio y nuestro código esté listo, vamos a crear una rama con el nombre *gh-pages*:

```
miPC@miPC MINGW64 /g/WORKSPACE/011
GitHub/Proyecto GIT (master)
$ git branch gh-pages
```

Ahora, haremos un *push* con esos cambios de nuestro repositorio local (computadora) a nuestro repositorio remoto (GitHub):

```
miPC@miPC MINGW64 /g/WORKSPACE/011
GitHub/Proyecto GIT (master)
$ git push origin gh-pages
```

Ahora ya podemos ver dichos cambios entrando en:

<http://nombreDeUsuario.github.io/repositorio.>

Todos los *commits* y cambios que empecemos a generar en la rama *gh-pages*, se verán reflejados en esta URL.

GitHub Pages es una excelente forma de mostrar nuestros proyectos al mundo a través de un sitio web. Viene bien para un portafolio, landing page e incluso para crear un blog. Todo de manera gratuita y con repositorios ilimitados.

Desgraciadamente no podemos usar código del lado del servidor (Python, Ruby, PHP, etc.); así que, si sólo queremos mostrar algunos proyectos de código abierto, alojar un blog o incluso compartir nuestro currículum, *GitHub Pages* es una opción interesante.