

Git e Github

Github pages



Índice

Introdução	3
Sites dinâmicos, sites estáticos e Jekyll	3
Por que usar sites estáticos?	4
GitHub Pages para usuários ou organizações	5
GitHub Pages para projetos	5

Introdução

GitHub Pages é um serviço GitHub que nos permite hospedar nossos projetos e mostrá-los ao vivo em uma página web estática sem a necessidade de pagar por hospedagem ou mesmo ter conhecimento de servidores ou DevOps.

É muito simples e não leva mais do que 5 minutos para ser montado.

As páginas do GitHub são páginas web públicas hospedadas e publicadas através do GitHub. A maneira mais rápida de se colocar em funcionamento é usar o Jekyll Theme Chooser para carregar um tema pré-construído. Você pode então modificar o conteúdo e o estilo de suas páginas do GitHub.

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

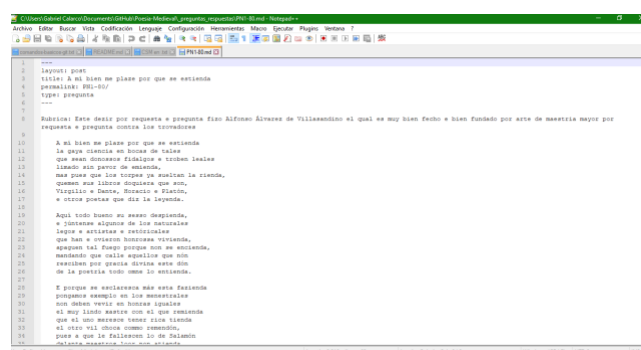
Sites dinâmicos, sites estáticos e Jekyll

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. Ambos os *sites dinâmicos*, assim como aqueles criados e gerenciados por sistemas de gerenciamento de conteúdo como *Drupal*, *WordPress* e *Omeka*, extraem informações de um banco de dados para completar o conteúdo de uma página da web.

Quando pesquisamos um livro na Amazon.com, por exemplo, a página de resultados da pesquisa não existe como uma página HTML completa; em vez disso, a Amazon.com tem um modelo para a página de resultados da pesquisa, que inclui elementos que todas as páginas de resultados da pesquisa compartilham (como o menu principal e o logotipo da Amazon) e consulta um banco de dados para inserir os resultados de nossa pesquisa nesse modelo.

Os *sites estáticos* não utilizam um banco de dados para armazenar informações; em vez disso, todas as informações exibidas em cada página da web já estão contidas em seu arquivo HTML. As páginas HTML que compõem um site estático podem ser escritas inteiramente à mão ou parte deste trabalho pode ser delegado usando uma ferramenta como a Jekyll.

Jekyll é um software que nos ajuda a "gerar" ou criar um *web site estático* (Jekyll é freqüentemente referido como um "gerador de site estático"). Jekyll usa modelos de páginas para aqueles elementos como menus principais e rodapés que normalmente são repetidos em todas as páginas que compõem um site e, portanto, são muito demorados se escrevermos manualmente o HTML para incluí-los em cada página da web. Com Jekyll, estes modelos são combinados com outros arquivos com informações específicas (por exemplo, um arquivo para cada post do blog) para gerar páginas HTML completas para os visitantes do site. Jekyll não precisa consultar bancos de dados ou criar uma nova página HTML (ou completar uma página parcial) quando os usuários visitam uma página web; ele já tem as páginas HTML completas e só as atualiza quando/se alguma vez elas mudarem.



```
1 ---
2 title: A mi bien se piensa por que se entiende
3 permalink: /mi-bien-se-piensa/
4 type: post
5 ---
6
7 Subtítulo: Este texto por requesta e pregunta fizo Alfonso Sánchez de Villaseñor el qual se muy bien fecho e bien fundado por arte de maestría mayor por
8 requesta e pregunta contra los torreadores
9
10 A mi bien se piensa por que se entiende
11 la que muestra en boca de tales
12 que mas dimesse fidalgo e todos tales
13 llando sus puros de enredo,
14 mas puse que los torpes ya sacian la sienda,
15 quenen sus linderos de sacan,
16 Virgilio e Dante, Boccacio e Plauto,
17 e otros puros que diz la leyenda.
18
19 Aquí todo bueno se muy deprende,
20 e dimesse algunos de los naturales
21 laque e entiendo e entiendo
22 que han e omean hombres viciados,
23 después del tiempo por que se entiendo,
24 mandando que calla aquellos que son
25 viciados por que se entiendo
26 de la postura todo como lo entiendo.
27
28 E porque se entienda más esta fazienda
29 pongo un ejemplo de los naturales
30 son más viciados en horas iguales
31 el muy lindo sacan con el que entiendo
32 que el uno sacan como sacan,
33 el otro sacan como sacan,
34 puse a que se fazienda lo de sacan
35 sacan sacan como sacan.
```

Código para uma página Jekyll em formato md



Código da mesma página Jekyll, mas em formato html, como pode ser visto no navegador.

Note que quando alguém se refere a um "website Jekyll", está na verdade se referindo a um website estático (HTML puro) que foi criado usando Jekyll. Jekyll é um software que cria websites. Jekyll não está "rodando" o site ao vivo, mas é um "gerador de site estático": é uma ferramenta que cria os arquivos do site estático que são então hospedados em um servidor web, como é feito com qualquer outro site HTML.

Como os sites estáticos são apenas arquivos de texto (sem um banco de dados para complicar as coisas), é possível *versioná-los* facilmente, ou seja, usar uma ferramenta para acompanhar as diferentes versões do site ao longo do tempo, acompanhando as mudanças nos arquivos de texto que compõem o site.

O controle de versão é muito útil quando queremos combinar as duas versões (por exemplo, dois estudantes escrevem um post no blog e queremos combinar suas duas versões) ou quando queremos comparar arquivos para procurar diferenças entre eles (por exemplo, "Como foi o projeto descrito na página 'Sobre' original?"). O controle de versões é muito útil quando se trabalha em equipe (por exemplo, permite combinar e acompanhar o trabalho de diferentes pessoas), mas também é útil quando se cria ou administra um site por conta própria.

Por que usar sites estáticos?

Opções como *Drupal*, *WordPress* e *Omeka* são úteis para criar sites complexos e interativos como o Amazon ou uma edição digital interativa de um romance, mas para muitos blogs, sites de projetos e portfólios on-line, um site estático (como um site criado com o Jekyll) pode fazer tudo o que é necessário ao mesmo tempo em que oferece algumas vantagens:

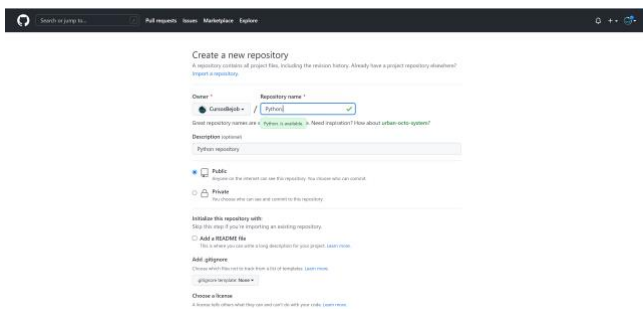
- **Manutenção:** As atualizações e a manutenção precisam ser feitas com muito menos frequência (menos de uma vez por ano).
- **Preservação:** Ao não utilizar um banco de dados, os arquivos de texto que compõem seu site são tudo o que precisamos manter para preservação e replicação. É fácil fazer o backup do site ou enviá-lo para um repositório institucional.
- **Aprendizagem:** Como não há banco de dados e não há muitos arquivos de código que forneçam funções que talvez nem precisemos, há muito menos componentes em nosso site; é mais fácil revisá-los e saber o que cada um faz. É muito mais fácil se tornar um usuário básico e avançado da Jekyll.
- **Mais personalização:** Como aprender a dominar nosso site é fácil, coisas que definitivamente queremos fazer, como mudar o visual (o "tema") de um site criado por *Jekyll*, são muito mais fáceis do que alterar o visual de um site *WordPress* ou *Drupal*.
- **Hospedagem gratuita:** Enquanto muitas ferramentas do site como *Drupal*, *WordPress* e *Omeka* são gratuitas, a hospedagem (pagando alguém para exibir seus arquivos do site aos visitantes do site) pode custar dinheiro.
- **Controle de versão:** Hospedagem nas *GitHub Pages* significa que nosso site está vinculado à interface visual do *GitHub* para controle da versão do *git*, de modo que as mudanças em nosso site podem ser rastreadas e, se necessário, revertidas para o estado anterior de qualquer post ou página do blog.

Isto inclui arquivos carregados que podemos querer armazenar no site, tais como silabos e posts antigos.

- **Segurança:** Não existe um banco de dados a ser protegido contra ataques maliciosos.
- **Velocidade:** Arquivos mínimos do site e nenhum banco de dados para consulta resultam em tempos de carregamento de páginas mais rápidos.

GitHub Pages para usuários ou organizações

Vamos começar criando um website para nosso usuário ou organização. Para fazer isso, precisamos ir ao GitHub e criar um novo repositório.



Este repositório deve começar com o nome de nosso usuário ou organização seguido do github.io.

Por exemplo: username.github.io

A partir daí, podemos começar a gerar nossos *commits*. Assim como faríamos com qualquer repositório; ou clonando nosso projeto:

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

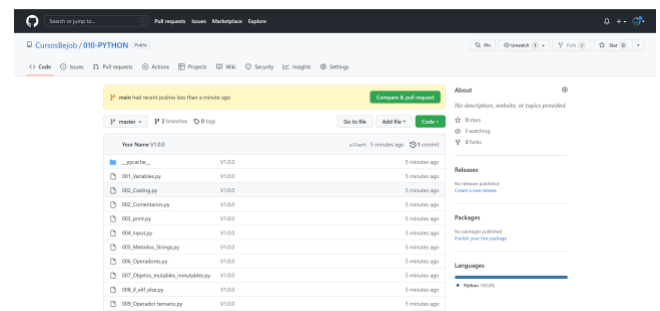
o haciendo una vinculación:

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

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

GitHub Pages para projetos

Como podemos ver, gerar um website com *GitHub* é bastante fácil. Mas não podemos fazê-lo apenas para nosso usuário ou organização, podemos também utilizá-lo com nossos projetos. Para isso, temos que ir ao *GitHub* e criar um novo repositório, ou usar um já criado por nós.



Quando começarmos a criar nossos *commits* neste repositório e nosso código estiver pronto, criaremos uma filial com o nome *gh-pages*:

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

Agora, vamos a fazer um *push* dessas para mudanças de nosso repositório local (computador) para nosso repositório remoto (*GitHub*):

```
miPC@miPC MINGW64 /g/WORKSPACE/011
GitHub/Proyecto GIT (master)
$ git remote add origin
https://github.com/nombreDeUsuario/nombr
eDeUsuario.github.io
```

Agora podemos ver essas mudanças indo para:

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

Todos os *commits* e mudanças que começarmos a gerar no ramo de *gh-pages* serão refletidos nesta URL.

GitHub Pages é uma ótima maneira de mostrar nossos projetos ao mundo através de um website. Vem a calhar para um portfólio, landing page e até mesmo para criar um blog. Tudo de graça e com repositórios ilimitados.

Infelizmente não podemos usar código do lado do servidor (Python, Ruby, PHP, etc.); portanto, se quisermos apenas mostrar alguns projetos de código aberto, hospedar um blog ou até mesmo compartilhar nosso CV, o *GitHub Pages* é uma opção interessante.