Jekyll for a Website 🌰

Introducción

Este documento trata de explicar cuales son los pasos básicos para utilizar Jekyll para construir una pagina web.

Que es Jekyll 👹?

Jekyll es un pequeño programa hecho en Ruby que transforma archivos `*.html`, markdown (`*.md` o `*.markdown`), css y demás en una web estática, nada más y nada menos, o lo que es lo mismo, una web de las de toda la vida.

Por qué Jekyll?

Jekyll, al igual que otros sistemas para hacer web estáticas [1] tiene sus ventajas y sus desventajas, pero podría decirse que para webs pequeñas o medias, las ventajas suelen ser mayores que las desventajas, bastante de largo.

Las principales **desventajas** se podría decir que son:

- Tiene una **curva de aprendizaje un poco empinada**. No es tan fácil y visual el diseño y la gestión, y suelen ser y utilizar herramientas enfocadas más a programadores. Sin embargo no es que sea un obstáculo muy difícil de superar, y una vez que practicamos un poco y nos familiarizamos, realizar cambios no es más difícil que realizar cambios en un documento de texto.
- Si se busca hace una web muy compleja, dinámica y de gran tamaño, las cosas se complican bastante.

Lo más curioso es que estas dos desventajas no es lo sean significativamente cuando las comparamos con lo que nos pasaría en plataformas que suelen tener famas de más sencillas, como puede ser Wordpress.

Ventajas:

- Más simple.
- Más ligero.
- Más seguro.
- Más barato (gratis?).
- Es de código abierto y libre.
- El contenido es tuyo, siempre.
- La puedes servir como tu quieras.

Como lo consigue?

- No hay un gestor de contenido, solo contenido. Lo único que hace Jekyll es trasformar el

contenido en un formato de pagina web, ayudándote a rehusar el contenido de la misma manera que hace Wordpress. Pero a diferencia de Wordpress que lo tiene que hacer cada vez que alguien visita la pagina, Jekyll solo lo hace cada vez que la pagina se compila, lo que ocurre solo cuando cambias el contenido.

- Como el contenido es estático y no dinámico, no se necesita una base de datos y por lo tanto es más rápido y seguro. No passwords, no ataques y normalmente no hay caídas.
- Como el resultado final es una web estática, en la que solo es necesario servir los archivos de esa web hay docenas de *hostings* donde podrás almacenar la pagina web de manera gratuita, o de una manera muy muy económica y que incluso la pueden pueden llegar a construir la web por ti a partir de tu archivos originales —código fuente. Este es el caso de GitHub, son su GitHub Pages, que sirve gratis las paginas cuyo código fuente se hospeda en una repo[2] de GitHub. Todo gratis hasta 1 GB. Dudo que necesites mucho más.
- En el caso de que no quisieras usar GitHub, hay otras plataformas parecidas —hostings para Git— o que simplemente sirven para publicar paginas web estáticas. Las opciones son amplias. Si contratas un hosting cualquiera, también se pueden servir ahí, ya que solo hay que subir los archivos por FTP —por ejemplo— una vez que se compila la web.
- No se puede decir que el contenido no sea tuyo en otras plataformas como Wordpress, pero si que está atrapado. Con Jekyll el contenido son simples archivos de texto, ya sea en formato html o markdown que creas y editas en tu ordenador, subes a una repo, es transforman en una pagina web y se sirven. En WP el contenido está atrapado dentro de una base de datos, y recuperarlo suele ser algo un tanto tedioso.

Que necesito?

Algunas de las cosas que necesito son:

- [x] Installar Git
- [ ] Installar Jekyll[3]
- [ ] Instalarse un editor de texto como es debido, como puede ser Atom o Sublime Text $ $ $, puedes usar Notepad++ si lo prefieres, pero no es lo mejor [1 & 2]. Yo uso Sublime ☠. Pero vamos que para los gustos los colores.
- [ ] Instalarse un editor de Markdown[4]. Lo puedes hacer con el editor de texto, pero... normalmente es mucho rollo ya que quieres ver un poco como queda, y es lo que los editores de markdown suelen ayudarte a ver. Yo uso Typora, que creo que lo hay para Windows, y que ahora es gratis porque está en beta y que es un tipo de editor WYSIWYG. Pero hay otros.
- [ ] Instalar GitHub Desktop.
- [ ] Aprender sobre Git. Más que nada porque cuando estás usando GitHub para poder subir y bajar archivos lo tienes que hacer con Git. Git no solo que va a servir para esto, sino para hacer control de versiones de otros documentos y proyectos que estén formados por archivos de texto fundamentalmente. Lo que sobran son cursos en internet:
  - [ ] Software Carpentry
    - [ ] Git
    - [ ] unix shell (si tiens win2 10 y vas a usar bash puede ser util)

    - [ ] GitHub Lab
- [ ] Mira que plantilla te gusta y hacerle un fork.
- [ ] Cuando tengas el fork, puedes añadirme como colaborador.
- [ ] Si quieres puedes publicar la web right away para ver como queda, como he hecho yo —fíjate en la URL— después de hacer un par de cambios.

Que más tengo que saber?

A ver que me pueda venir a la mente...

- Que si quieres aprender un poco de html a lo mejor este curso puede estar bien. Pero yo nunca he hecho ninguno, suelo ir aprendiendo sobre la marcha. Aunque a lo mejor me miro el que te he mandado.
- Lo mismo con markdown.
- Jekyll tiene algo adicional que es lo que le permite que tenga plantillas y no repetir información, es liquid template system or markup. Básicamente es lo que permite a jekyll hacer algunas de las acciones tipo `for` [5] y cosas por el estilo... le da esteroides al html. Te permite por ejemplo tener una parte de la web en un archivo html —por ejemplo el footer o pie— y que se repita en todas las paginas, sin tener que repetir la información.

A partir de aquí es más o menos ir probando y ver que pasa. He estado echando un ojo a la plantilla

---

---

1. Como Octopress, Hugo, etc.↩

2. Repo es repositorio, que es donde almacenas tus archivos originales o código fuente cuando estás usando Git o otro sistema de control de cambios. ↩

3. Yo esto nunca lo he hecho —no suelo usar windows 💩, pero si tienes Windows 10 💩💩💩💩💩 parece que lo mejor es instalar Windows Subsystem for Linux, dejarse de ~~mierdas~~ *tonteridas* y utilizar bash.↩

4. Markdown no es más que un `html` aguado y fácil para que escribir un documento no sea un dolor de dedos de todo lo que le tienes que poner cuando haces un html. Este documento está hecho en markdown, y básicamente lo que hago es escribir como en el word. Es más, para algunas cosas es mejor que el puñetero word.↩

5. Lo que viene siendo un loop, repetir una acción sobre algo hasta que se acaba ese algo. ↩