Blob con github y fastpages

<https://github.com/fastai/fastpages>

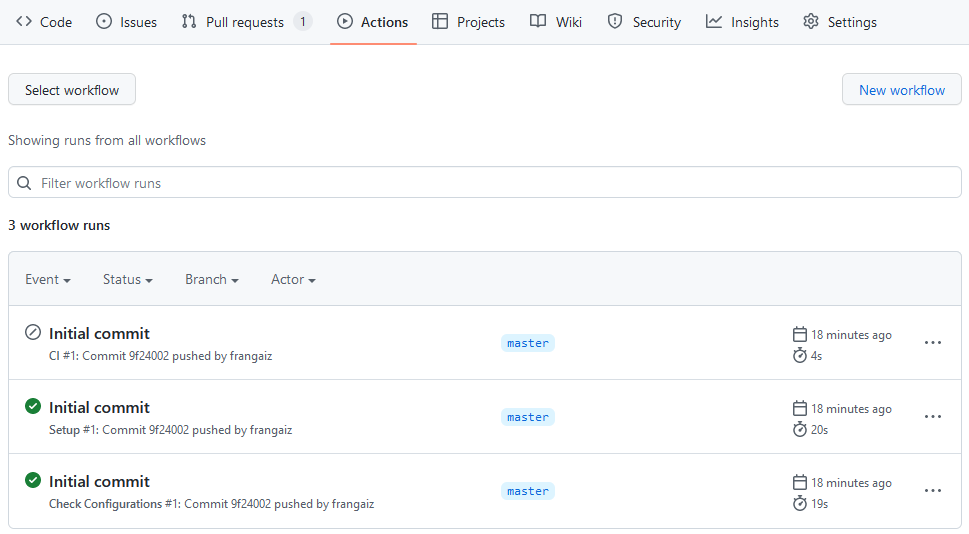
<https://www.youtube.com/watch?v=L0boq3zqazI>

NOTA: en este sitio se usa otra cosa parecida tb de fastai, que es fast\_template, no fastpages. Yo lo hice y era mas fácil incluso: <https://maxlein.github.io/fastbook/2020/05/28/app_blog.html>

NOTA: por debajo usa un generador de sitios estáticos llamado jekill

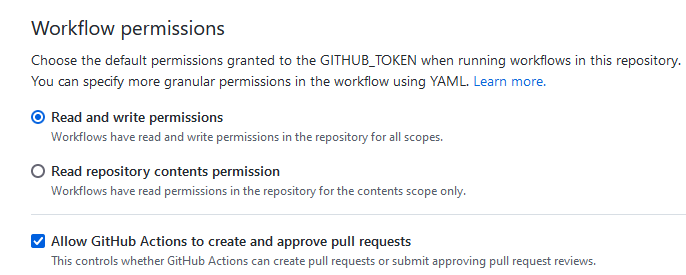
NOTA: he creado un repo llamado “paw” en <https://github.com/frangaiz/paw>, y el blog se puede ver en <https://frangaiz.github.io/paw/>

1. Se pueden añadir notebooks de jupiter, markdown e incluso Word docx.
2. Entrar en el repo de fastpages de fastai (<https://github.com/fastai/fastpages>) y, en las instrucciones de setup buscar un enlace que genera una copia de ese repo (<https://github.com/fastai/fastpages/generate>)
3. En readme del repo (<https://github.com/fastai/fastpages>) están parte de las instrucciones.
4. Tb recomienda mirar el video de <https://www.youtube.com/watch?v=L0boq3zqazI>
5. Al clonar el repo usando <https://github.com/fastai/fastpages/generate> hay que llamar al repo como se quiera PERO NO con {your-username}.github.io. Por ej., yo pongo “paw”
6. A los 30 sg., se recibe un pull-request en el nuevo repo con instrucciones de creación del blog. OJO: puede ocurrir que la ejecución del action que construye el sitio falle (ver <https://forums.fast.ai/t/fastpages-initial-commit-setup-fail/96069>). Esto se ve en el menú de actions.



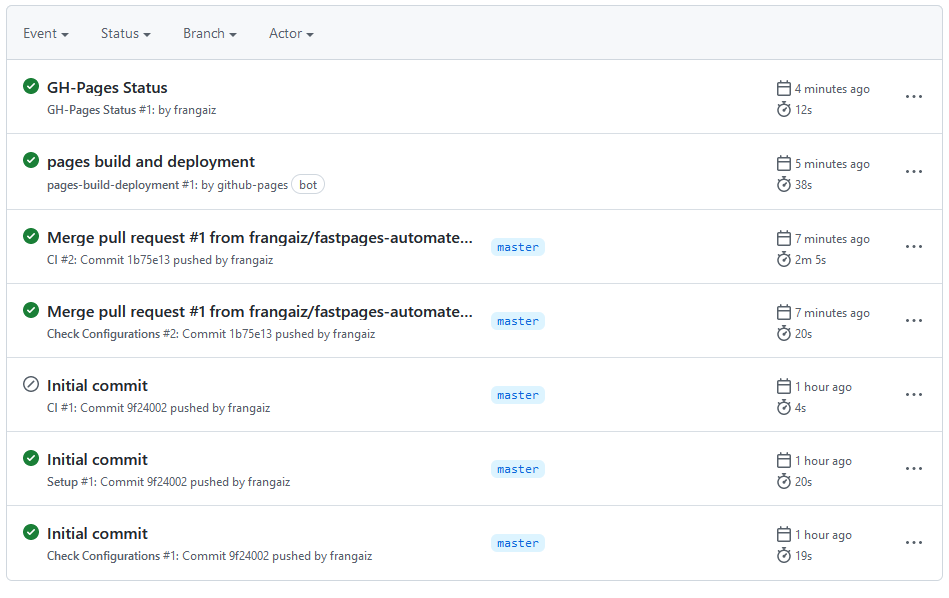
Esto quedó así una vez corregido, pero estaba en rojo cuando falló

Hay que ir a settings, en la categoría de Actions / General / Workflow permissions, y dejarlo así:



Luego ir a la lista de acciones y re-ejecutar la acción que falló.

1. El pull request se llama “Initial Setup”. Las instrucciones del Pull-request dicen que:
   1. Crear un par de claves SSH en <https://8gwifi.org/sshfunctions.jsp>. Dejar el passphrase en blanco
   2. Ir a Settings / Secrets del repositorio que se está creando (ej <https://github.com/frangaiz/paw/settings/secrets/actions/new>). click New repository secret. Copy and paste the **Private Key** into the Value field. This includes the "---BEGIN RSA PRIVATE KEY---" and "--END RSA PRIVATE KEY---" portions. **In the Name field, name the secret SSH\_DEPLOY\_KEY.**
   3. Ir a Settings / Keys de repositorio creado (ej <https://github.com/frangaiz/paw/settings/keys>). click the Add deploy key button. Paste your **Public Key** from step 1 into the Key box. In the Title, name the key anything you want, for example fastpages-key. Finally, **make sure you click the checkbox next to Allow write access** (pictured below), and click Add key to save the key.
2. Ahora hay que aceptar el pull-request (merge el pull-reques)
3. Tras eso, las actions construyen el sitio, que, si no se usa un dominio público, estará en <https://frangaiz.github.io/paw/> . La creación tarda unos minutos, tras lo que se ve el estado de las action así:



1. Sdf

# Configurar el blog

Ir al fichero [\_config.yml](https://github.com/frangaiz/paw/blob/master/_config.yml) y configurar:

title: fastpages

description: An easy to use blogging platform with support for Jupyter Notebooks.

github\_username: frangaiz

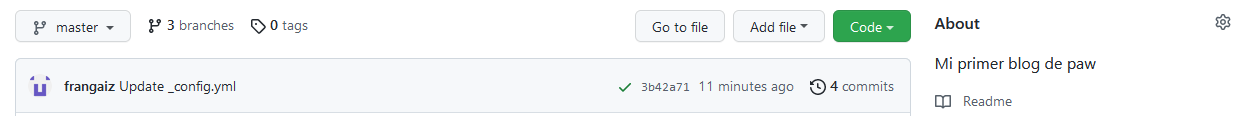
# you can comment the below line out if your repo name is not different than your baseurl

github\_repo: "paw"

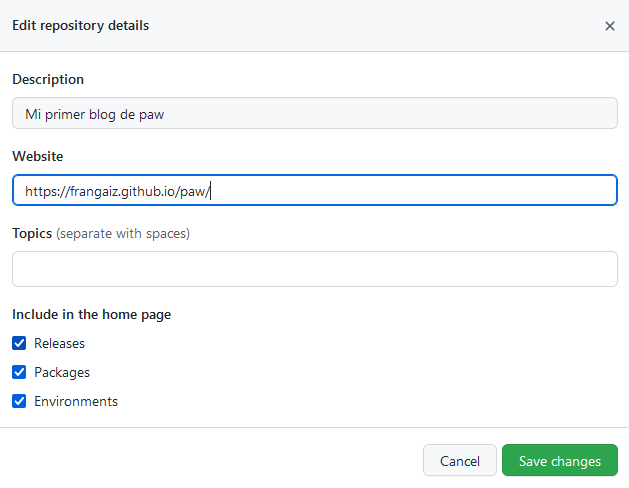
Acabar con un commit del fichero, lo que pondrá en marcha las action que refrescan el blog.

Ver más opciones de configuración en <https://github.com/fastai/fastpages#site-wide-configuration-options>

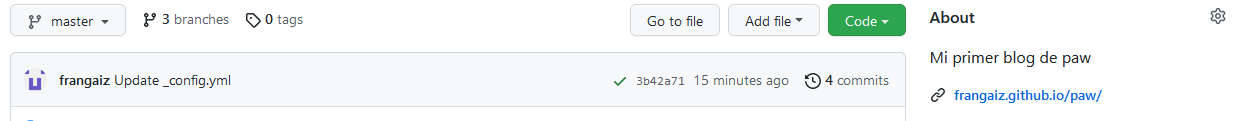
Tb se puede editar la parte de about del repo:



Y añadir la URL del blog



para que quede más fácil acceder a él.



# Cambiar la home

Editar el fichero index.html

# Añadir contenido

Copiar en los directorios \_posts, o \_word los ficheros que se quieren publicar, con la convención de nombre: YYYY-MM-DD-\*.md o YYYY-MM-DD-\*.docx.

Ir al directorio dónde se va a colocar el contenido, seleccionar “Add file”, subir ficheros y hacer commit. Los actions se pondrán en marcha y actualizarán el sitio.

El contenido queda un poco feo si no se configura un poco.

# Configuración de cada post particular

Ver <https://github.com/fastai/fastpages#customizing-blog-posts-with-front-matter>

Se puede cambiar el título el resumen, habilitar comentarios…

Por ejemplo se puede añadir:

---

toc: true

layout: post

description: A minimal example of using markdown with fastpages.

categories: [markdown]

title: An Example Markdown Post

---

El title es el que aparece en la página principal, y le sigue la descripción…

Ver el restos de cosas que se puede customizar en <https://github.com/fastai/fastpages#customizing-blog-posts-with-front-matter>.

Tb se pueden añadir permalinks: <https://jekyllrb.com/docs/permalinks/>

# Writing Blog Posts With Microsoft Word

Save your Microsoft Word documents into the /\_word folder with the same naming convention (YYYY-MM-DD-\*.docx) specified for notebooks.

*Note:* [alt text](https://support.office.com/en-us/article/add-alternative-text-to-a-shape-picture-chart-smartart-graphic-or-other-object-44989b2a-903c-4d9a-b742-6a75b451c669) in Word documents are not yet supported by fastpages, and will break links to images.

## Specifying front-matter for Word documents

fastpages does not have a robust way to specify [front matter](https://jekyllrb.com/docs/front-matter/) for Word documents. At the moment, you can only specify front matter globally for all Word documents by editing [\_action\_files/word\_front\_matter.txt](https://github.com/fastai/fastpages/blob/master/_action_files/word_front_matter.txt).

To specify unique front matter per Word document, you will need to convert Word to markdown files manually. You can follow the steps in this [blog post](https://www.fast.ai/2020/01/18/gitblog/), which walk you through how to use [pandoc](https://pandoc.org/installing.html) to do the conversion. Note: If you wish to customize your Word generated blog post in markdown, make sure you delete your Word document from the \_word directory so your markdown file doesn’t get overwritten!

*If your primary method of writing blog posts is Word documents, and you plan on always manually editing markdown files converted from Word, you are probably better off using* [*fast\_template*](https://github.com/fastai/fast_template) *instead of fastpages.*