

---

# Práctica Jekyll

Implantación de Aplicaciones Web

José Juan Sánchez Hernández

Curso 2024/2025

# Índice

- 1 Práctica: Creación de blogs con Jekyll y GitHub Pages** **1**
- 1.1 Creación de un contenedor Docker con Jekyll . . . . . 1
  - 1.1.1 `jekyll new` . . . . . 2
  - 1.1.2 `jekyll build` . . . . . 2
  - 1.1.3 `jekyll serve` . . . . . 2
- 1.2 Publicar una web estática en GitHub Pages . . . . . 3
- 1.3 *Themes* para Jekyll . . . . . 3
  
- 2 Referencias** **4**
  
- 3 Licencia** **5**

# Índice de figuras

# Índice de cuadros

# 1 Práctica: Creación de blogs con Jekyll y GitHub Pages

**Jekyll** es un generador de **sitios web estáticos** que nos permite crear de forma sencilla blogs, sitios webs personales o webs para proyectos. Los sitios webs generados con **Jekyll** no usan una base de datos, el contenido del sitio web está escrito en archivos de texto plano en formato **Markdown** (o **Textile**) y plantillas **Liquid**.

**Jekyll** es el motor de **GitHub Pages**, un servicio que ofrece **GitHub** a sus usuarios para que puedan publicar sitios webs estáticos alojados en los repositorios que tienen en **GitHub**.

## 1.1 Creación de un contenedor Docker con Jekyll

Si buscamos la **imagen oficial de Jekyll en Docker Hub** encontraremos el **repositorio oficial en GitHub**.

Existen tres imágenes oficiales con Jekyll:

- `jekyll/jekyll`: Default image.
- `jekyll/minimal`: Very minimal image.
- `jekyll/builder`: Includes tools.

En esta práctica utilizaremos la imagen por defecto `jekyll/jekyll`.

Para consultar todos los comandos que podemos ejecutar con `jekyll` podemos ejecutar el siguiente comando:

```
1 docker run -it --rm -u $(id -u):$(id -g) -v "$PWD:/srv/jekyll" jekyll/jekyll
  jekyll
```

Deberíamos obtener la siguiente salida:

```
1 Usage:
2
3   jekyll <subcommand> [options]
4
5 Options:
6   -s, --source [DIR]   Source directory (defaults to ./)
7   -d, --destination [DIR] Destination directory (defaults to ./_site)
8       --safe           Safe mode (defaults to false)
9   -p, --plugins PLUGINS_DIR1[,PLUGINS_DIR2[,...]] Plugins directory (
10                          defaults to ./_plugins)
11       --layouts DIR   Layouts directory (defaults to ./_layouts)
12       --profile        Generate a Liquid rendering profile
```

```
12     -h, --help          Show this message
13     -v, --version      Print the name and version
14     -t, --trace       Show the full backtrace when an error occurs
15
16 Subcommands:
17 docs
18 import
19 build, b              Build your site
20 clean                 Clean the site (removes site output and metadata file)
21                       without building.
22 doctor, hyde         Search site and print specific deprecation warnings
23 help                 Show the help message, optionally for a given
24                       subcommand.
25 new                  Creates a new Jekyll site scaffold in PATH
26 new-theme           Creates a new Jekyll theme scaffold
27 serve, server, s    Serve your site locally
```

### 1.1.1 jekyll new

Este comando nos permite crear la estructura de directorios y los archivos necesarios de un nuevo proyecto Jekyll.

```
1 docker run -it --rm -u $(id -u):$(id -g) -v "$PWD:/srv/jekyll" jekyll/jekyll
  jekyll new blog
```

### 1.1.2 jekyll build

Este comando nos permite generar un sitio HTML estático a partir del contenido del proyecto Jekyll.

```
1 docker run -it --rm -u $(id -u):$(id -g) -v "$PWD:/srv/jekyll" jekyll/jekyll
  jekyll build
```

**Nota:** Tenga en cuenta que este comando tendrá que ejecutarlo dentro del directorio donde tenga el contenido del blog.

### 1.1.3 jekyll serve

Este comando nos permite servir de forma local un sitio HTML estático generado a partir del contenido del proyecto Jekyll.

```
1 docker run -it --rm -p 4000:4000 -u $(id -u):$(id -g) -v "$PWD:/srv/jekyll"
  jekyll/jekyll jekyll serve --force_polling
```

La opción `--force_polling` permite que el contenido del sitio se vaya generando automáticamente cuando existe algún cambio en los archivos del proyecto.

**Nota:** Tenga en cuenta que este comando tendrá que ejecutarlo dentro del directorio donde tenga el contenido del blog.

Para ver nuestro sitio web sólo tenemos que abrir un navegador web y acceder a la dirección de localhost en el puerto 4000:

```
1 http://127.0.0.1:4000
```

También podríamos acceder a través de la dirección IP de nuestra máquina.

## 1.2 Publicar una web estática en GitHub Pages

Se recomienda seguir los pasos de la [documentación oficial de GitHub Pages](#).

## 1.3 Themes para Jekyll

[En esta web](#) puedes encontrar una extensa colección de *themes* para crear web estáticas con Jekyll.

## 2 Referencias

- [Jekyll](#)
- [GitHub Pages](#)
- [Jekyll en Wikipedia](#)



## **3 Licencia**

Esta página forma parte del curso Implantación de Aplicaciones Web de José Juan Sánchez Hernández y su contenido se distribuye bajo una licencia Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional.