

Contenido

Dotar a nuestro sitio de contenido para publicar

✓ Objetivo:

- Asociar una plantilla o theme al sitio web
- Estructurar contenido para mejorar la visualización del sitio.
- Organizar los contenidos que vamos a publicar (Organizaremos por directorios y ficheros)

Páginas referenciadas o de consulta

- <https://gohugo.io/content-management/organization/>
- <https://desarrolloweb.com/articulos/generar-contenido-site-hugo>
- Plantillas disponibles en hugo <https://themes.gohugo.io/>
- Plantilla usada como ejemplo [relearn: https://themes.gohugo.io/themes/hugo-theme-relearn/](https://themes.gohugo.io/themes/hugo-theme-relearn/)

Contenido

Ahora que tenemos una plantilla, queda lo principal, que es aportar contenido a nuestro sitio web. Son sitios estáticos, por lo que nuestro objetivo es aportar ficheros de texto con datos, información, imágenes, efectos visuales para que sean publicados en la web. La idea va a ser aportar ficheros en formato [markdown](#) cuyo contenido queremos

¿Qué estructura tiene el contenido de las páginas?

- Ahora toca aportar contenido a nuestro sitio web. Las páginas Tipos de páginas de contenido. Agrupar las páginas por tipos.
- En Hugo el contenido se organiza como luego se va a ver en la web. Anidamos el contenido en subdirectorios (**secciones**):
content/<directorios>
- Pensar en **front-matter** - **archetypes**
- Plantillas específicas para cada modelo

Creando contenido: markdown

Para crear nuevo contenido vamos a generar páginas de texto con formato **markdown** ([Hablamos más abajo sobre el tema](#)). Podemos consultar las siguientes referencias para el texto en markdown. Son las páginas que dirigen la web de hugo

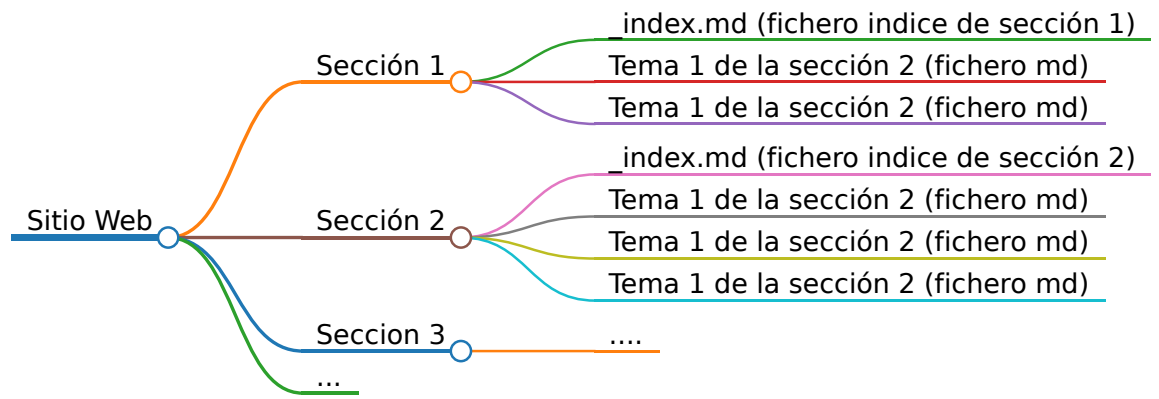
- [Daring Fireball: Markdown, John Gruber \(Creator of Markdown\)](#)
- [Markdown Cheatsheet, Adam Pritchard](#)
- [Markdown Tutorial \(Interactive\), Garen Torikian](#)
- [The Markdown Guide, Matt Cone](#) Otras referencias
- <https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>
- <https://programminghistorian.org/es/lecciones/introduccion-a-markdown>
-

Este metalenguaje o lenguaje con marcas es muy utilizado en la documentación técnica.

Es importante aprender a utilizarla con cierta flexibilidad

Creando páginas de contenido

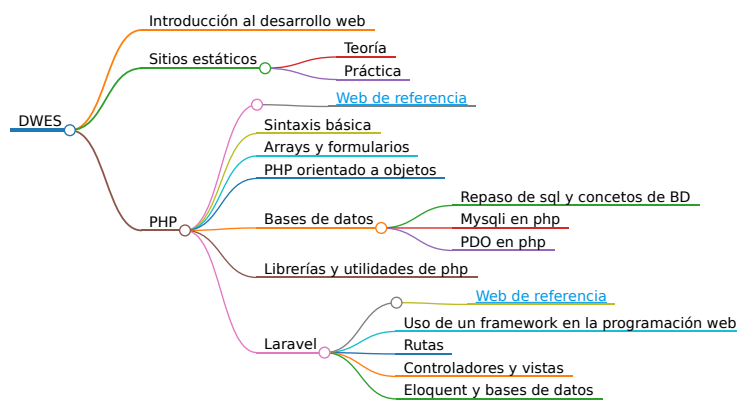
Lo primero establecemos la organización de nuestros contenidos



Observamos cómo usamos diferentes directorios para crear nuestra organización de nuestro contenido.

Cada directorio corresponderá a una **sección**

Dentro de cada sección puede haber un fichero **_index.md** cuyo contenido se visualizará cuando seleccionemos el directorio concreto. Por ejemplo para el contenido del curso podría tener una organización como se muestra en la imagen siguiente:



Creando ficheros y secciones

Usaremos el CLI de hugo (línea de comandos) para crear ficheros de contenido y/o secciones

```
hugo new [seccion]/fichero.md
```

La sección puede ser un directorio o varios (subdirectorios anidados)

Los ficheros se crean en la carpeta **content** del directorio raíz del proyecto

El fichero creado tendrá un contenido según quede especificado en el fichero **patrón** ubicado en la carpeta **archetypes/default.md**

Creando un fichero md

```
hugo new docs/prueba.md
```

Genera un fichero con el siguiente contenido

```
---
title: "Prueba"
date: 2021-10-04T19:39:33+02:00
draft: false
---
```

Las páginas en el directorio content *Las páginas ubicadas directamente en el directorio **content** serán páginas decir globales o bundles pages: contacto, about, datos relevantes. Estas páginas no son secciones y por lo tanto no tendrán contenidos

La forma de organizar las secciones dependerá de nuestra visión del sitio y en cualquier caso deben de ser páginas relacionadas con el tema de la sección: teoría, prácticas ,....

Cada directorio puede contener un fichero llamado **_index.md** que va a contener el índice o información que se visualizará al cargar la sección concreta, sería como el fichero índice de la sección.

Creando ficheros con relearn

En el caso de la plantilla que estamos usando ([ver documentación](#)), clasifica las páginas de dos tipos, las que son cabecera de un tema o sección (*_index.md*) y las que tienen contenido de las mismas.

Para crear las páginas de un tipo u otro, tiene una serie de argumentos

Ejemplo de cómo hacerlo con relearn

- Para crear una página que es cabecera de capítulo (creamos la sección teoría y el fichero *_index.md*)

```
hugo new chapter --kind teoria/_index.md
```

- Para crear un fichero dentro de la sección teoría

```
hugo new chapter --kind teoria/tema1.md
```

Front matter

<https://gohugo.io/content-management/front-matter/> Cuando generamos contenidos, los ficheros creados vemos que en la cabecera o primeras líneas pueden tener **metainformación** Esta información va entre tres líneas y constituye lo que se llama el **front matter** de la página En esta sección daremos valores a variables que luego se pueden utilizar.

De los diferentes valores, nos interesa ver alguno que suele estar en el archtypes y suelen aparecer por defecto:

- **title** será el título que va a tener esta página para referenciarse en los diferentes menús
- **weight** es una de las formas de ordenar las páginas y secciones.
- **draft** Indica si la página es borrador *true* o no. En caso de ser borrador nos e visualizará.
- **description** Es una descripción que aparecerá en la página principal o index de esta sección. Aparecerá para cada página, salvo que se especifique que no lo queramos. visualizará.

La información del fichero creado anteriormente corresponde al **Front matter**, el cual es aportado por el fichero base o patrón ubicado en la carpeta **archetypes**

```
content
├── blog          <-- Sección, porque es el primer nivel bajo content/
│   ├── funny-cats
│   ├── mypost.md
│   └── kittens   <-- Sección porque contiene _index.md
│       └── _index.md
└── tech          <-- Sección porque contiene _index.md
    └── _index.md
```

Repaso

- Lo que precede al contenido
- Sintaxis especial
- Hay unas variables predefinidas pero el usuario puede crear nuevas.
- Se genera según el **archetype** de la sección. Por defecto `_default.md`
- Se pueden pasar valores por defecto a las páginas que descienden de una sección. Por ejemplo si ponemos en `content/blog/_index.md`

```
title = 'Blog'  
[cascade]  
  banner = 'images/typewriter.jpg'
```

Esta página y todas las que descienden de la sección `blog` devuelven `images/typewriter.jpg` en `.Params.banner` a no ser que tengan su propio banner o un ancestro más cercano lo modifique.

Referencias

- <https://gohugo.io/content-management/front-matter/>
- <https://gohugo.io/variables/page/>

Taxonomías

<https://gohugo.io/content-management/taxonomies>

Una taxonomía es una categorización que nos permite clasificar contenido.

Por ejemplo, en una web de Películas podríamos tener actores, directores, estudios, géneros, años, premios ... A continuación se muestra un breve ejemplo, pero esta parte no la vamos a incluir en nuestro ejemplo.

Por defecto Hugo añade las taxonomías **categories** y **tags**. Se pueden añadir más en **config**

```
[taxonomies]  
  category = 'categories'  
  tag = 'tags'
```

Si se quiere deshabilitar:

```
disableKinds = ['taxonomy', 'term']
```

En **frontmatter**

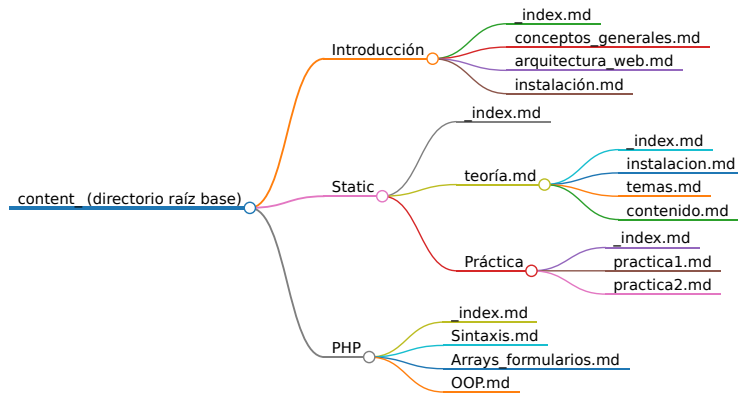
```
categories:  
- Personal  
- Trabajo  
tags:  
- software  
- html
```

Práctica

Ahora practica tú



- Establece un contenido para tu sitio web. Para ello debes de tener clara las secciones y en cada sección los ficheros contenidos. Por ejemplo podría ser parte del contenido de este curso visto anteriormente



Con esto nos aparecerá la siguiente estructura:

contenido ejemplo

Ahora completa los ficheros con contenido (no hace falta que haya muchos, mejor que tenga sentido, que vayas realizando tu portfolio, que sea tu forma de presentarte y mostrar a los demás en qué estás trabajando, puede ser un blog, ...

Recuerda cómo se crean las secciones, ficheros `_index` y ficheros de contenido en este template

También podrías usar la estructura anterior e ir completándola con esquemas, resúmenes de los aspectos que vayas aprendiendo en el curso.

Ahora reordena todo usando el *frontmatter*, recuerda la propiedad **weight**, modifica los títulos según consideres y puedes poner una descripción.

Ve revisando como queda todo al visualizarlo en la web

Comprueba en el servidor como se ha actualizado tu sitio web