

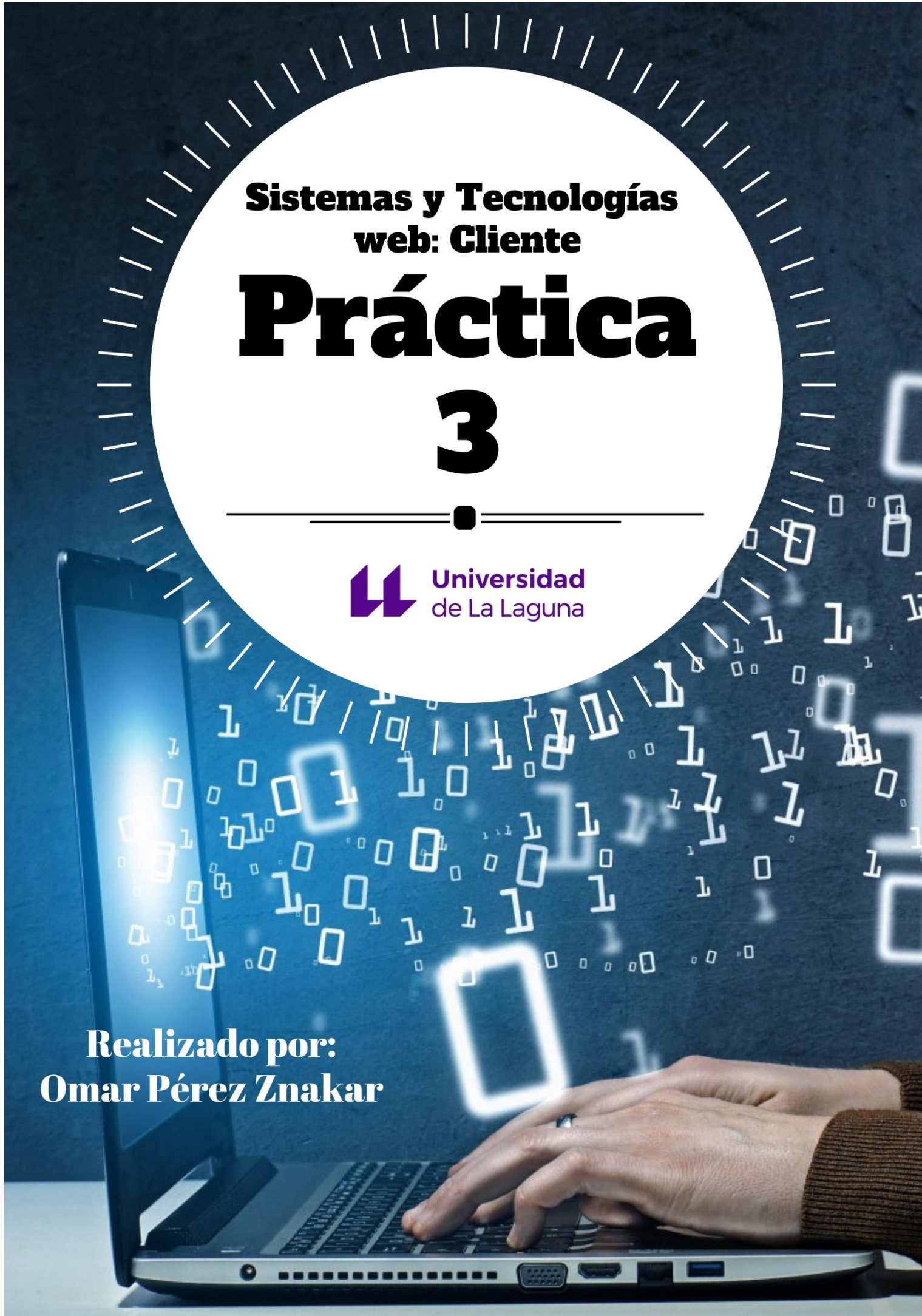
**Sistemas y Tecnologías
web: Cliente**

Práctica 3



**Universidad
de La Laguna**

**Realizado por:
Omar Pérez Znakar**



1. Ejercicios Jekyll

1. Analizaremos el proyecto Jekyll básico en el repositorio: [Proyecto Básico en Jekyll](#).

- **Añade dos páginas adicionales al proyecto, y haz que se muestren junto con la página About.**

Para este paso, se han creado dos páginas adicionales, incorporando los datos necesarios para su correcto funcionamiento. En la siguiente imagen podemos ver un ejemplo del mismo:

Ejemplo de página web adicional añadida.

```
---
layout: page
title: Prueba1
permalink: /Prueba1/
---

<p>
Lorem Ipsum is simply dummy text of the printing and typesetting industry.
Lorem Ipsum has been the industry's standard dummy text ever since the
1500s, when an unknown printer took a galley of type and scrambled it to
make a type specimen book. It has survived not only five centuries, but
also the leap into electronic typesetting, remaining essentially unchanged.
It was popularised in the 1960s with the release of Letraset sheets
containing Lorem Ipsum passages, and more recently with desktop publishing
software like Aldus PageMaker including versions of Lorem Ipsum.
</p>
```

Salida en pantalla

Your awesome title

Prueba1 Prueba2 About

- **Añade otro post al proyecto, y haz que se muestre en la página [index.html](#).**

Para llevarlo a cabo, se ha creado una entrada adicional en la colección de elementos del post. A su vez, para completar esto se ha incorporado un texto de ejemplo extraído de “Lorem Ipsun”. Así, podemos ver un ejemplo de esto en la siguiente imagen:

Post creado

```
---
layout: post
title: "PruebaPost"
date: 2014-10-18 12:58:29
categories: jekyll update
---
```

What is Lorem Ipsum?

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Why do we use it?

It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English. Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for 'lorem ipsum' will uncover many web sites still in their infancy. Various versions have evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like).

Salida en página web

Your awesome title

[Prueba1](#) [Prueba2](#) [About](#)

Posts

Oct 18, 2014

[PruebaPost](#)

Oct 18, 2014

[Welcome to Jekyll!](#)

subscribe [via RSS](#)

2. Identifica objetos, filtros y etiquetas en el siguiente fragmento de código e indica cuál es una finalidad.

```
{% assign episodes = site.episodes | sort: 'weight' %}
`{% for episode in episodes limit: 6 %}
  `<div class="grid__cell grid__cell--33">
    `{% include episode_preview.html episode=episode %}
  </div>
{% endfor %}
```

En esta imagen podemos observar como almacenamos todas los “site.episodes” de nuestra página web, es decir, el contenido de la carpeta “episodes” dentro de una variable. A su vez, podemos ver como ordena estos episodios por el campo “peso”. A continuación, podemos observar como se recorre la lista hasta un máximo de 6 elementos. Por último, vemos como se incluye un documento HTML al que se le paso la estructura de un episodio para su posterior procesamiento.

3. Qué código html tendría como salida:

```
<html>
  <head>
    <title>{{ page.title }} -Ejemplo Jekyll </title>
  </head>
  <body>
    {% include nav.html %}`
    {{ content }}
  </body>
</html>
```

Según lo que podemos ver en la imagen, podríamos decir que tendría lo siguiente.

- En la pestaña de la página aparecería el nombre de la aplicación.
- También podemos ver que se incluye el fichero nav.html (generalmente relacionado con un navbar o menú).
- Po último podemos ver que incluye el contenido de la página en la que nos encontramos actualmente.

4. Especifica los filtros, variables Jekyll, tags, y qué se consigue con el siguiente código.

```
<footer id="footer">
  <p class="small">© Copyright {{ site.time | date: '%Y' }} {{ site.author }}</p>
</footer>
```

Como se puede apreciar en la imagen, se recoge la fecha actual parseando dicho valor únicamente recogiendo el valor del año actual. A su vez, se incorpora el nombre del creador de la página introducido en el fichero de configuración.

5. Indicar qué scripts se están invocando con el siguiente código:

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>
<script src="{{ "/" | relative_url }}assets/js/main.js"></script>
<script src="{{ "/" | relative_url }}assets/js/highlight.js"></script>
```

En la anterior representación, podemos ver que se está llamando al fichero “assets/js/main.js” de forma relativa, es decir, nos resuelve la problemática del nombre la máquina y del proyecto (elementos generalmente variables). Por ejemplo, en un despliegue local, sería: “127.0.0.1:puerto/Nombre Proyecto/assets/js/main.js” o en un despliegue en github <https://usuario.github.io/NombreProyecto/assets/js/main.js>.

6. Indica el resultado que se obtiene con el siguiente código, sabiendo que paginas corresponde la colección de almacenada en _characters

```
<ul class="nav">
  {% assign paginas = site.paginas | sort: 'name' %}
  {% for pagina in paginas %}
    <li class="nav__item">
      <a href="{{ pagina.url }}">{{ pagina.title }}</a>
    </li>
  {% endfor %}
</ul>
```

En la imagen anterior, se representa como obtienen todos los datos recogidos en la carpeta “_character”, en ella, podemos ver como existen diversos datos con una estructura mínima de url de navegación y un título por cada dato. Posteriormente, se extraen y se muestra al usuario. Su uso, puede ser múltiple, desde disponer unos simple datos para mostrar hasta disponer un menú para poder redirigirse al sitio web incorporado en dicha página (entre otras funciones).

7. Dada una colección de documentos con las siguientes variables en el frontmatter.

```
title: Mi Pagina
thumbnail_url: https://XXX
image_url: https://
```

Indicar qué objetivo tiene el siguiente include:

```
<a href="{{ character.url }}" class="character-preview">
  
  <div class="character-preview__label">
    {{ character.title }}
  </div>
</a>
```

En la imagen anterior, se representa como se puede guardar diversos datos dentro de una colección para, posteriormente, mostrarlos al usuario. Esta funcionalidad es muy importante en esta tecnología, ya que de forma simple podemos aumentar el contenido de nuestra página web con tan solo crear un fichero con el contenido necesario

.