
Table of Contents

| | |
|--------------|-------|
| Introduction | 1.1 |
| Herramientas | 1.2 |
| Gitbook | 1.2.1 |
| Github | 1.2.2 |
| Gulp | 1.2.3 |
| Apuntes | 1.3 |

Apuntes SYTW

Este libro está dedicado a llevar un constante seguimiento de la asignatura de *Sistemas y Tecnologías Web*.

El contenido se ha dividido en diversas secciones, orientadas hacia la práctica y hacia la teoría de la asignatura.

Herramientas

Este primer capítulo se centra en la determinación de las herramientas que se utilizarán a lo largo de la asignatura así como un acercamiento y profundación acerca del funcionamiento de cada una de ellas.

Gitbook

Se trata de una herramienta para **crear documentación de proyectos y libros técnicos usando Markdown y Git/Github**. Entre las numerosas características que nos ofrece podemos destacar:

- Publicar y mantener fácilmente libros online utilizando *gitbook.com*.
- Ofrece la posibilidad de trabajar desde una aplicación propia.
- Permite **incluir ejemplos y ejercicios interactivos**.
- Mediante la utilización de **Markdown** podemos "maquetar" nuestros documentos y crearlos con diferentes formatos: **PDF, ebook o web**.

Gitbook está implementado usando node.js, podemos instalarlo usando **NPM**:

```
npm install gitbook -g
```

Una vez instalado podemos comenzar a trabajar sobre nuestros archivos en formato Markdown y posteriormente servir dichos archivos cara un repositorio, por ejemplo:

```
gitbook serve txt ./repositorio
```

En este ejemplo, **txt** es el directorio en el que tenemos nuestros archivos Markdown y **./repositorio** es donde nos dejará los html.

Una opción muy interesante es servir nuestro libro bajo gh-pages de forma que contenemos nuestro libro exportado en un sitio web fácilmente accesible.

Para profundizar un poco mas adjuntamos el siguiente enlace:

Github

GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git. El código se almacena de forma pública, aunque también se puede hacer de forma privada, creando una cuenta de pago.

Que es un issue?

En GitHub, un **issue** es la unidad de trabajo designada para realizar una mejora en un Sistema informático. Un **issue** puede ser el arreglo de un fallo, una característica pedida, una tarea, una solicitud de Documentación en específico y todo tipo de solicitud al equipo de desarrollo. Con las **issues** se puede asignar una tarea a un colaborador de tu repositorio o proyecto, lo cual permite una mejor organización en proyectos que se pueden realizar desde casa siempre y cuando la organización tenga su repositorio en la nube.

Gulp

Se trata de un **automatizador de tareas** que corre bajo **node.js**. A grosso modo, **Gulp** nos permite definir tareas bajo un mismo **nombre** de tal forma que al ejecutar nuestra tarea, **gulp** de forma automática realice las tareas que hemos indicado. Las tareas que podemos crear pueden englobar todo aquello que queramos **automatizar**, ejemplo de los usos *más comunes* de gulp son:

- Minimificación y concatenación
- Uso en sistemas de control de versiones
- Compilación de archivos SASS
- Agregar prefijos
- Realizar pruebas en php
- Crear controladores
- Etc

Proceso de instalación de Gulp

Para poder llevar a cabo la instalación es necesario tener Node.js instalado previamente. A continuación, en línea de comandos escribiremos la siguiente línea que iniciará el proceso de instalación de Gulp.js:

```
npm install -g gulp
```

Una vez finalizada la instalación podemos verificar si se instaló de forma correcta comprobando la versión de la siguiente manera:

```
gulp -v
```

Si la instalación ha sido correcta aparecerá algo como **CLI versión 3.9.0**.

Utilización de Gulp.js

Para comprender correctamente la utilización de **Gulp** describiremos los pasos a realizar para un supuesto proyecto. Primeramente deberemos crear una carpeta con el nombre que deseemos. En ella crearemos un archivo **gulpfile.js**, este archivo tiene **especial importancia** ya que es el que Gulp.js necesita para saber que tareas realizará. Pasaremos a configurarlo más adelante, por ahora inicializaremos nuestro directorio mediante:

```
npm init
```

En el proceso de inicialización, se nos formularán diversas preguntas como el nombre, una descripción, licencia... *En caso de tener ya un directorio de trabajo inicializado en el que únicamente queramos incorporar el funcionamiento de Gulp, nos saltaremos ese paso previo.*

Antes de comenzar a trabajar sobre nuestros archivos necesitamos **agregar las dependencias de desarrollo** a nuestro proyecto, para **Gulp**:

```
npm install --save-dev gulp
```

En este momento se deberían reflejar dichas dependencias en el **package.json** de nuestro proyecto. *Dejando de lado la configuración de carpetas que tengamos en nuestro proyecto, el gulpfile.js deberá contener algo similar a lo que se muestra a continuación:*

```
/*
 * Dependencias
 */
var gulp = require('gulp');
/*
 * Configuración de la tarea
 */
gulp.task('nombre de la tarea', function () {
  ...
});
```

Con el método `gulp.task` **definimos una tarea**. Dicho método recibe **3** argumentos, el nombre de nuestra tarea (*podemos utilizar "default" como nombre para declarar una tarea por defecto que invocaremos desde línea de comandos introduciendo simplemente "gulp"*), la o las tareas que dependen de esta tarea y por último la función que llama a esta tarea. También es posible **definir** si queremos que una tarea se realice **si** una tarea anterior ha terminado, por ejemplo:

```
gulp.task('js', ['css'], function(){
  ...
});
```

De esta manera, la tarea `js` se ejecutará sólo cuando la tarea `css` **haya terminado**. Existen multitud de plugins para Gulp, ejemplo de los más comúnmente utilizados son **Gulp-uglify** para minificar y **Gulp-concat** para concatenar.

Apuntes clases teóricas

La presente sección se destina a llevar un registro de los apuntes de la asignatura de **Sistemas y Tecnologías Web**.