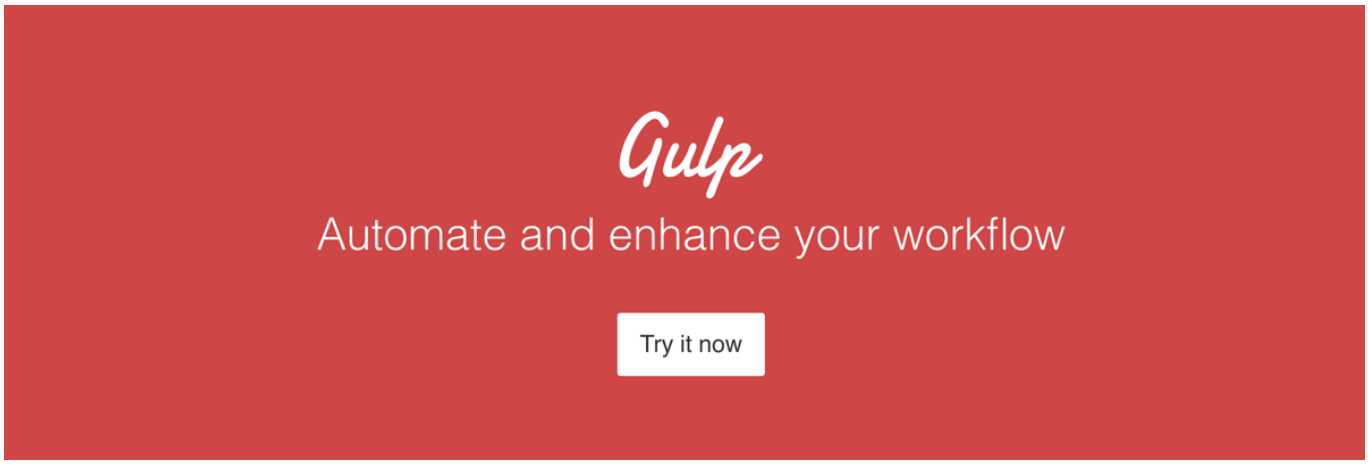
Lección 2: Gulp.JS

Lectura: Qué es Gulp.JS y cómo lo implementamos

¿Qué es Gulp.JS y cómo lo implementamos?



Es una herramienta en forma de script, escrita en Nodejs, que ayuda a automatizar tareas comunes en el desarrollo de aplicaciones. Muchos programadores front-end solemos realizar tareas muy repetitivas. Esta herramienta fue creada para facilitar el proceso de desarrollo, gastando menos tiempo en el proceso de escribir código, realizar validaciones y pruebas de nuestro código.

Para iniciar un nuevo proyecto con esta herramienta tan sólo debemos ejecutar unos pocos comandos para tener nuestros paquetes listos para su uso. Son los siguientes:

```
~ sudo npm install gulp-cli
```

Después de instalar nuestro paquete de forma global, nos ubicamos en nuestra base de nuestro proyecto, donde encontramos nuestro archivo package.json y instalamos el siguiente paquete:

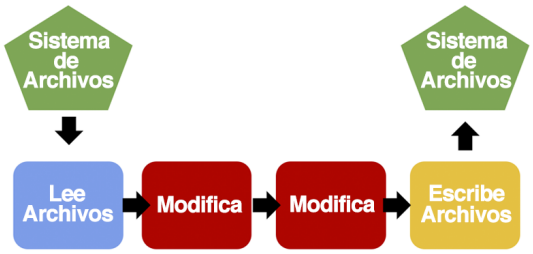
```
~ npm install gulp --save-dev
```

Este comando indica que vamos a instalar un paquete o librería con el nombre gulp y que deseamos que se guarde como un librería de dependencia, únicamente de desarrollo, que no la vamos a necesitar obligatoriamente en producción, sino que únicamente la usaremos en un entorno de desarrollo donde hacemos normalmente nuestras pruebas locales, mucho antes de realizar un proceso de lanzamiento en nuestro sitio oficial.

Enseguida, después de realizar este proceso, creamos en la base de nuestro proyecto un nuevo archivo con el nombre de **gulpfile.js** donde definiremos nuestras tareas o procesos a automatizar, en la base de nuestro proyecto deberíamos de tener la siguiente estructura:

```
.
..
gulpfile.js
node_modules
package.json
```

Para ver realmente el funcionamiento de esta herramienta, debemos conocer la forma en que trabaja, en la siguiente imagen veremos la forma en que trabaja Gulp.



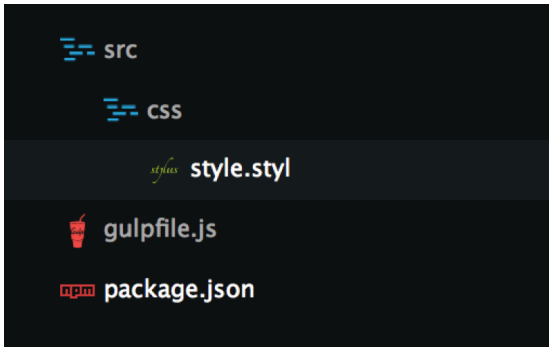
Las tareas que normalmente realizamos en el desarrollo de aplicaciones, suelen ser tareas como preprocesadores de estilos, ya sean realizadas con Stylus, Sass, L
siendo archivos físicos con la sintaxis determinada por la herram

archivos físicos con la sintaxis determinada por la herramienta que se haya usado. Estos archivos físicos no son válidos o no podemos usarlos directamente en nuestro navegador, sino que debemos hacer una tarea de procesar ese archivo para obtener nuestro archivo válido o que nuestro navegador pueda entender o interpretar.

En este caso sería pasar un archivo .styl a .css. Aquí veríamos el flujo real de gulp, pues tomamos los archivos que necesiten ser procesados y luego modificados, procesados o concatenados y reescritos en nuevos archivos, que serán los archivos que agregaremos a nuestro aplicativo o desarrollo que estemos realizando.

Veamos un ejemplo sencillo de procesar un archivo stylus a css:

En la base de nuestro proyecto creamos la siguiente estructura de carpetas **"src/css/".**
Dentro de esta estructura creamos un archivo **estilo.styl** que viene siendo archivo que vamos a procesar por medio de gulp. Deberíamos tener, hasta este punto, la siguiente estructura y archivos:



Ya que contamos con estos archivos y nuestra estructura de carpeta deseada, programaremos nuestra tarea en el archivo **gulpfile.js,** que procese nuestro archivo estilo.styl y como resultado, crearemos la siguiente estructura de carpetas **"build/css/estilos.css"** desde la base de nuestro proyecto.

```
gulp.task('style', function(){
    gulp.src('./src/css/*.styl')
    .pipe(stylus())
    .pipe(gulp.dest('./build/css'));
});
```

Como vemos, registramos una nueva tarea de gulp con la siguiente función **gulp.task('', '')** que es la función que utilizamos para registrar nuevas tareas en gulp.
Registramos nuestra tarea con el nombre de "style" y a continuación pasamos una función anónima.
La función que encontramos dentro de la función anónima, osea,

búsqueda del archivo o archivos que vamos a procesar.
A continuación encontramos la función **.pipe(),** dentro de esta función. Ejecutamos una función que va a ser la encargada de procesar o convertir nuestro archivo o archivos según lo que hemos asignado en el **gulp.src('').**
La función que se ejecuta en este caso es **stylus(),** pero te preguntarás de dónde proviene esta función. Pues bien, la definimos en un lugar previo, antes de definir nuestras tareas, así:

```
var gulp = require('gulp');
var stylus = require('gulp-stylus');
```

Para que realmente todo funcione correctamente, debemos instalar primero nuestros paquetes necesarios y luego definirlos dentro de nuestro archivo gulpfile.js
En este caso tenemos dos paquetes que debemos instalar:

- El que ya hemos mencionado y es con el siguiente comando: **"npm install gulp --save-dev"**
- Y una nueva librería, esta es la encargada de procesar los archivos .styl y la instalamos con el siguiente comando **"npm install gulp-stylus --save-dev"**

De esta forma ya tendríamos todo finalizado y listo para ejecutar nuestra primera tarea automatizada en nuestros flujos de trabajo diario.

Este sería nuestro resultado a nivel de archivo **gulpfile.js.**

```
var gulp = require('gulp');
var stylus = require('gulp-stylus');

gulp.task('default', function(){
    console.log('Hola este es mi primer task con gulp');
});

gulp.task('style', function(){
    gulp.src('./src/css/*.styl')
    .pipe(stylus())
    .pipe(gulp.dest('./build/css'));
});
```

Y la ejecución de esta tarea daría el siguiente resultado: