

# ¿Qué es GULP?

Es una build-tool o Task-runner que nos permite generar nuestra página web mediante la especificación de un workflow o sucesión de tareas que nosotros definiremos

## Requisitos

Nodejs-> entorno de ejecución JavaScript

Npm-> gestor de paquetes para node

Npx-> para ejecución de paquetes node

## Instalación

Npm install gulp -g //instalación de gulp

## Comprobar

Gulp --version

## Crear un proyecto gulp

//1º Crear un proyecto con

Npm init

//2º Instalar gulp en mi proyecto, es decir, mi proyecto depende de gulp

// para hacer esto, gulp debe estar instalado en el sistema

Npm install gulp --save-dev

# El fichero gulpfile.js. Tareas Gulp

Una vez creado el proyecto debemos crear el fichero gulpfile.js: un fichero javascript que se carga automáticamente al ejecutar gulp y que contiene las tareas que vamos a ejecutar en nuestro workflow o gestor de tareas.

## ¿Cómo se estructura el fichero gulpfile.js?



## Para ejecutar tareas

Gulp -> Ejecuta la tarea por defecto

Gulp --tasks -> lista las tareas que contiene el fichero

Gulp nombre\_de\_la\_tarea -> ejecuta la tarea nombre\_de\_la\_tarea

## Tareas en serie y en paralelo

Gulp permite ejecutar tareas en serie (serie) y en paralelo (parallel)

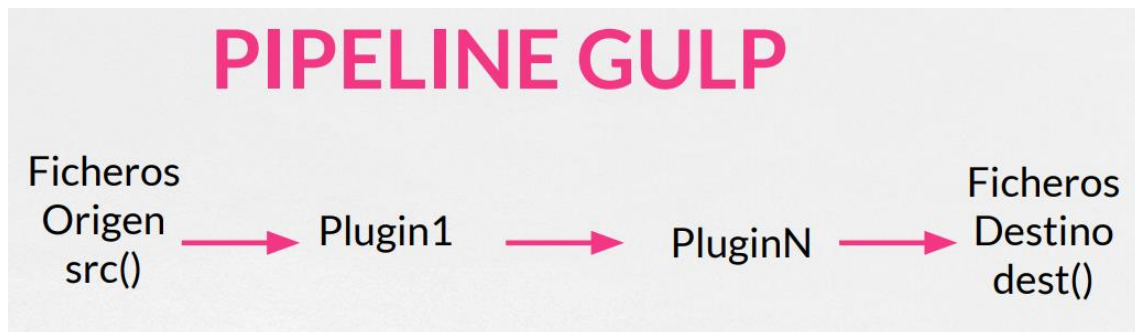
```
const {series, parallel} = require("gulp");  
  
function holamundo(cb){  
  console.log("Hola Mundo");  
  cb()  
}  
  
function adiosmundo(cb){  
  console.log("Adios Mundo");  
  cb();  
}  
  
exports.holamundo = holamundo;  
exports.adiosmundo = adiosmundo;  
exports.default = holamundo;  
exports.serie = series(adiosmundo, holamundo);  
exports.paralelo=parallel(holamundo, adiosmundo);
```

## Pipeline. Métodos src() y dest()

Con los métodos src() y dest() podemos procesar ficheros y construir flujos de trabajo donde usaremos plugins de transformación de esos ficheros.

Src()->aquí definimos los archivos a procesar en la tarea (origen)

Dest()->aquí definimos la carpeta donde se guardarán los archivos resultado(destino)



## Watch. Modo vigilante

### Ejemplo:

Cuando se produce un cambio en alguno de los ficheros scss de la carpeta scss se llamará a la tarea compilar\_scss

```
const { watch } = require('gulp');

exports.default = function() {
  watch('scss/*.scss', compilar_scss);
};
```

## Plugins de gulp

Los podemos encontrar en la página oficial de GULP

<https://gulpjs.com/>

gulp instalación

//Instalación global

Npm install -g nombre-plugin

//Instalación en el proyecto (package.json)

Npm install --save-dev nombre-plugin

## Algunos plugins

- **gulp-concat:** Concatenación de ficheros
- **gulp-if:** Ejecución condicional de tareas en Gulp
- **gulp-rename:** Renombrado de ficheros
- **gulp-dart-sass:** Compilación de Dart-Sass
- **gulp-processhtml:** Procesar y modificar ficheros HTML
- **gulp-plumber:** Para gestionar errores de las tareas de un pipeline

- **del:** Borrado de ficheros
- **path:** Trabajo con rutas
- **gulp-image-optimize:** Para optimizar el tamaño de los ficheros de imágenes
- **gulp-please:** Para el postproceso de ficheros CSS
- **sassdoc:** Para documentar mis ficheros Sass con SassDoc
- **gulp-ssh:** Para conexiones sftp y ssh

## **¿Cómo puede ser mi workflow?**

1. Generar las hojas de estilo a partir de ficheros scss
2. Generar la documentación de nuestro framework
3. Mover todos los ficheros a la carpeta destino
4. Borrar el contenido de la carpeta destino antes de la generación
5. Subir la carpeta destino al servidor