



Accesibilidad con GRUNT

Diseño de interfaces  
Web

**ESTHER CID**

2º DAW



## ÍNDICE

OPTIMIZACIÓN DE INTERFACES CON GRUNT .....	3
INSTALACIÓN DE NODESJS .....	3
INSTALAR NPM .....	6
¿Qué es GRUNT? .....	6
INSTALACIÓN .....	7
¿QUÉ CONTIENE UN PROYECTO DE GRUNT? .....	7

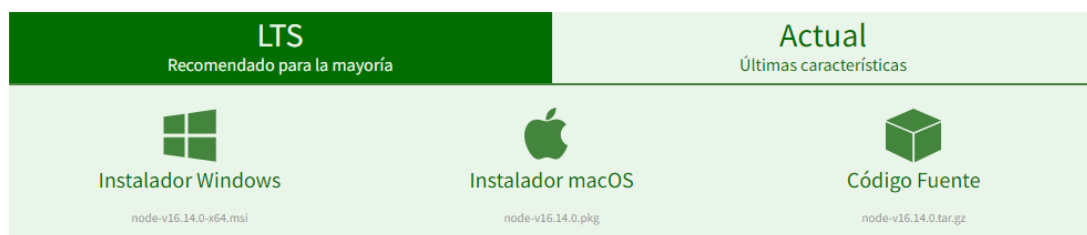
## OPTIMIZACIÓN DE INTERFACES CON GRUNT

Es muy conveniente realizar una optimización de las interfaces que estamos construyendo de forma continuada.

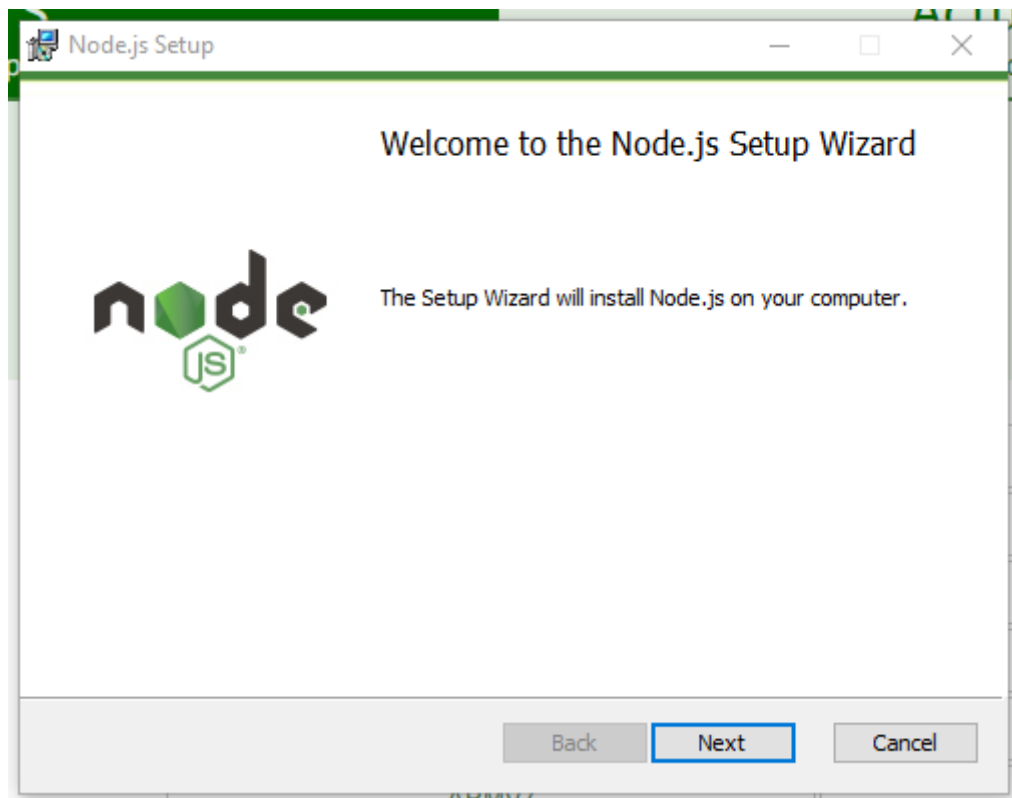
Para poder hacerlo de forma coordinada es necesario utilizar ejecutores de tareas, son ficheros con instrucciones que se configuran para hacer esto de forma automatizada. Algunas de las herramientas que se suelen utilizar se ejecutan con NodeJS y son Grunt y Gulp, aunque nosotros añadiremos una más, que será Selenium.

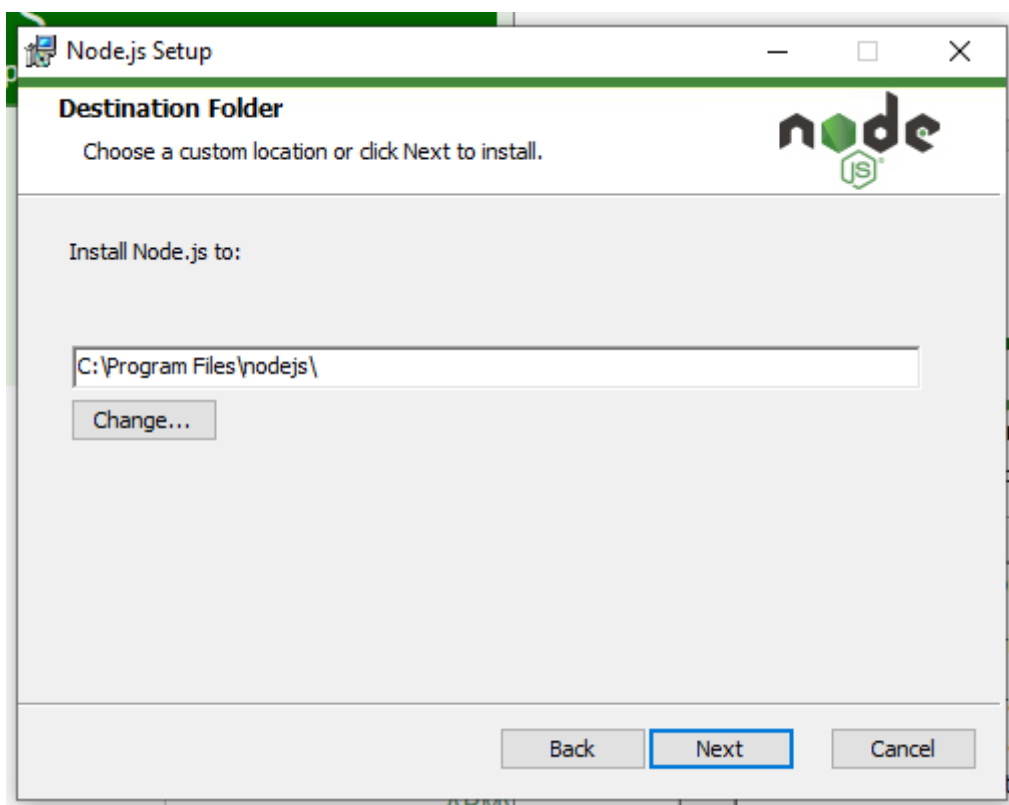
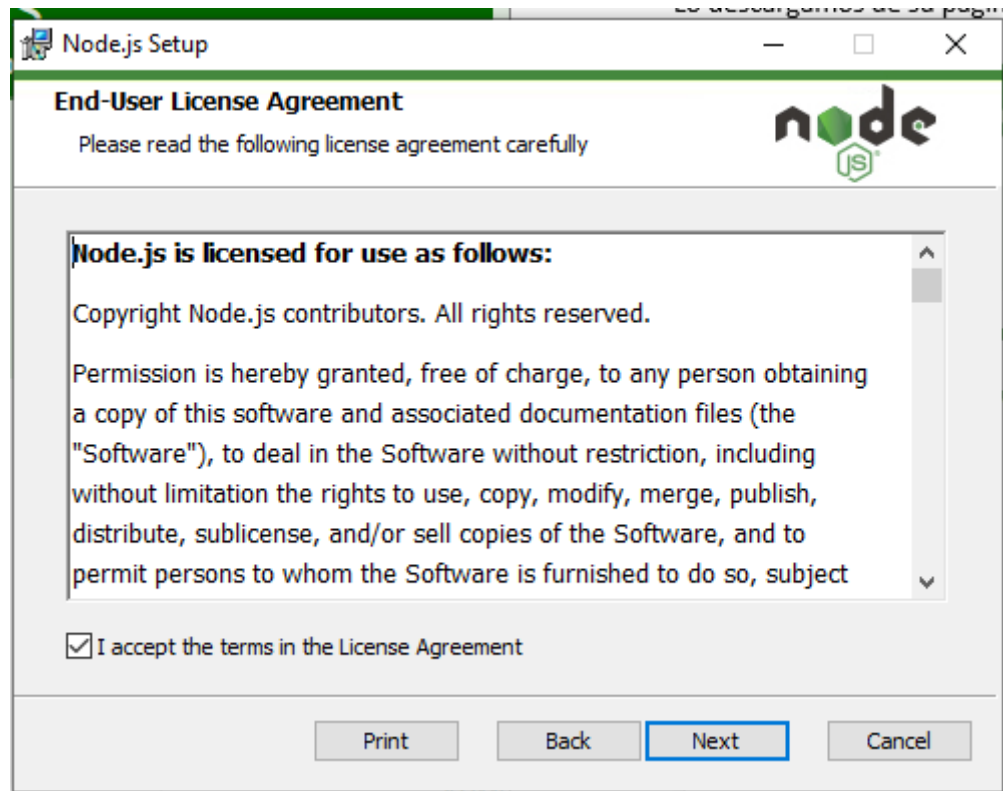
### INSTALACIÓN DE NODEJS

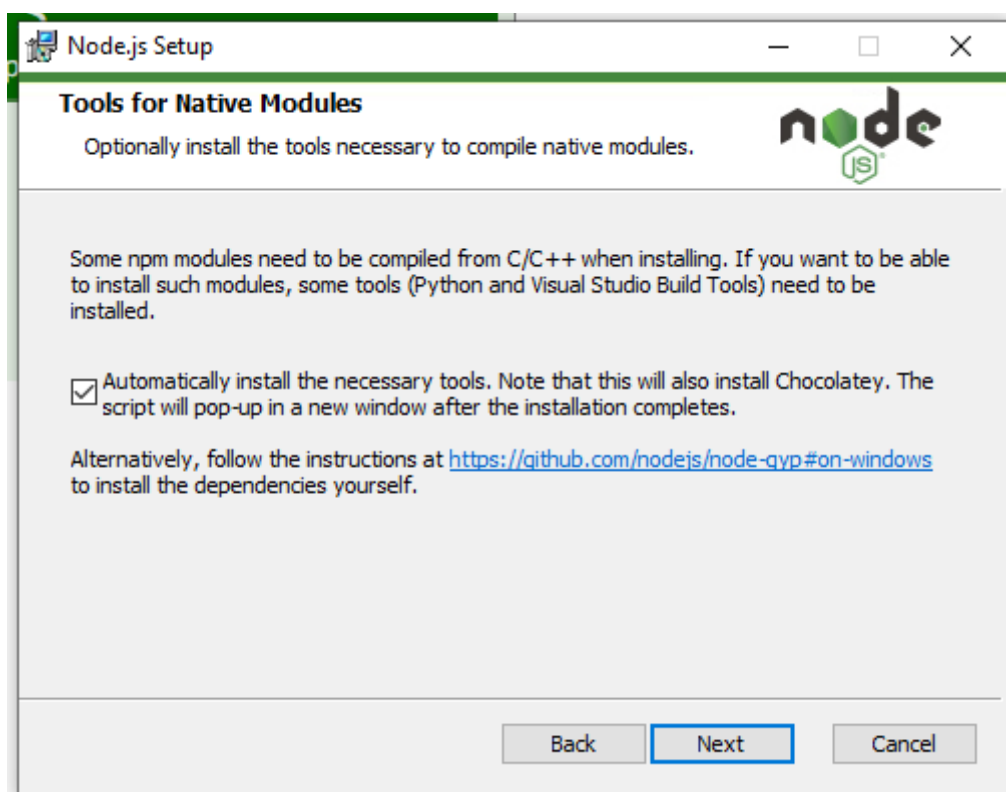
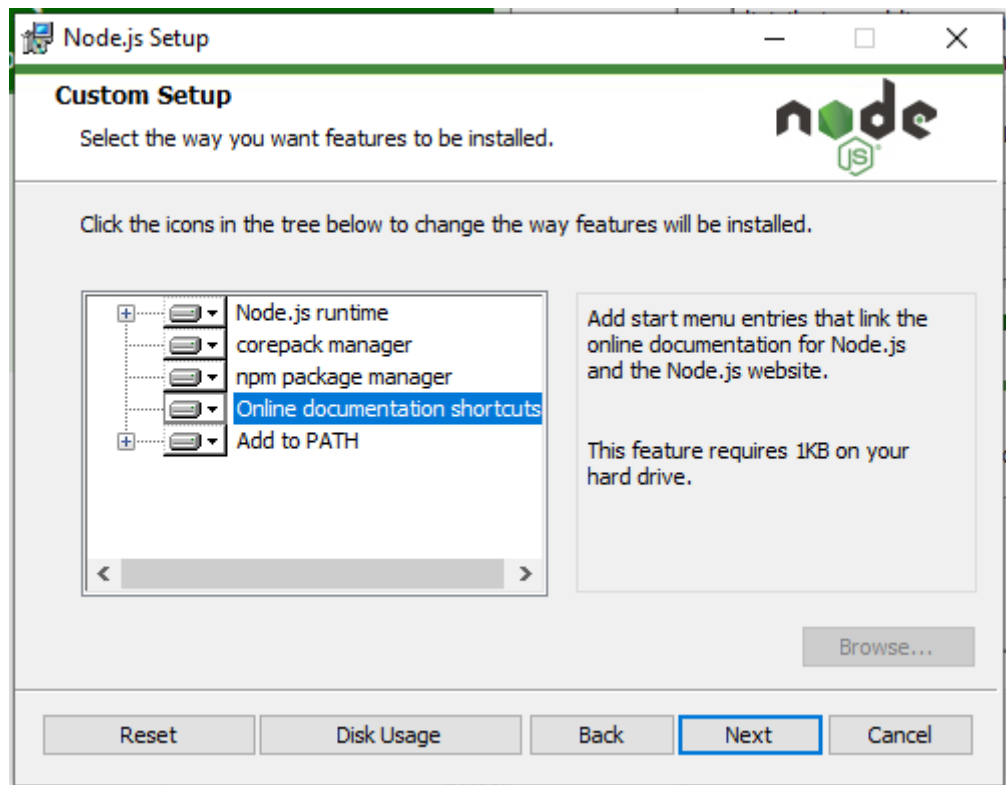
Lo descargamos de su página: <https://nodejs.org/es/download/>

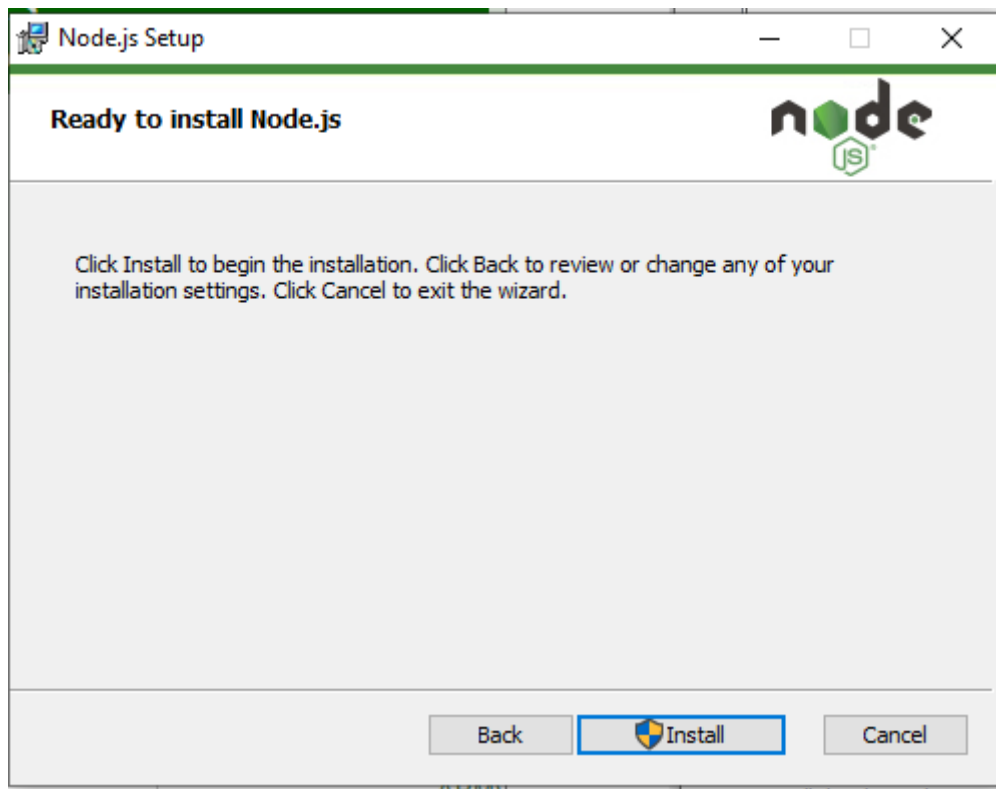


Cuando tenemos el instalador descargado lo abrimos y completamos los apartados requeridos para completar la instalación.









Comprobamos que node se ha instalado con el siguiente comando.

```
C:\Users\Daw>node --version  
v16.13.0
```

## INSTALAR NPM

NPM es el gestor de paquetes de NodeJS, para poder instalarlo tenemos que ir al CMD e introducir el siguiente comando.

```
C:\Users\Daw>npm install -g npm  
[██████████] / reify:npm: timing reify:createSparse Completed in 0ms
```

Comprobamos que se ha instalado bien NPM de la siguiente manera.

```
C:\Users\Daw>npm --version  
8.5.2
```

## ¿Qué es GRUNT?

Es un programa para correr tareas programadas en javascript y fue escrito en Node.js por lo que tener node instalado es necesario.

Podemos realizar varias tareas utilizando GRUNT, como, por ejemplo, minificación tanto de css como de javascript, compilación, testing de unidad, otras.

La principal ventaja que obtenemos utilizando GRUNT es el no tener que hacer tareas repetitivas, minificar y unir el css nos ayuda a que nuestra página cargue más rápido.

## INSTALACIÓN

Para poder instalarlo necesitamos una versión de NodeJS superior o igual a 0.8.0.

Antes de instalar Grunt tenemos que hacerle un update al gestor de paquetes NPM, lo hacemos con el siguiente comando.

```
C:\Users\Daw>npm update -g npm  
[ ] / reify:npm: timing reify:createSparse Completed in 4ms
```

Ahora tenemos que instalar el cliente de Grunt, llamado Grunt-CLI, nosotros vamos a utilizar la versión 0.3.

Después instalamos el grunt-CLI.

```
C:\Users\Daw>npm install -g grunt-cli  
  
changed 59 packages, and audited 60 packages in 1s  
  
4 packages are looking for funding  
  run `npm fund` for details  
  
found 0 vulnerabilities  
  
C:\Users\Daw>
```









Ahora podemos utilizar el comando grunt en el CMD, esto no significa que tengamos instalado el “task-runner”, esto permite que podamos instalar múltiples versiones de grunt.

Ahora tenemos que entrar en la carpeta e intalar dentro de la misma Grunt.

```
C:\Users\Daw\Documents\grunt-optimized-web>npm install grunt-cli  
[ ] - idealTree: sill logfile start cleaning logs, removing 1 files
```

## ¿QUÉ CONTIENE UN PROYECTO DE GRUNT?

Se compone fundamentalmente de dos archivos importantes y de una estructura de carpetas definida.

 dist	03/03/2022 20:15	Carpeta de archivos	
 node_modules	03/03/2022 20:11	Carpeta de archivos	
 src	03/03/2022 20:09	Carpeta de archivos	
 stage	03/03/2022 20:15	Carpeta de archivos	
 .gitignore	16/02/2022 18:28	Archivo de origen ...	1 KB
 Gruntfile	16/02/2022 18:28	Archivo de origen ...	4 KB
 package	03/03/2022 20:11	Archivo de origen ...	2 KB
 README	16/02/2022 18:28	Archivo de origen ...	1 KB

Los archivos imprescindibles son: “package.json” que es donde se encuentran todos los paquetes de NodeJS que hemos instalado y el fichero de Gruntfile.

Dentro del package.json se encuentran todos los paquetes instalados en ese proyecto, entre ellos nos aparece las dependencias concretas de grunt, donde podemos ver todos los paquetes que se han instalado.



```
19   "author": "Pello Altadill",
20   "license": "ISC",
21   "bugs": {
22     "url": "https://github.com/pxai/junefy/issues"
23   },
24   "devDependencies": {
25     "csslint": "^0.10.0",
26     "grunt": "^0.4.5",
27     "grunt-accessibility": "^4.2.0",
28     "grunt-contrib-clean": "^0.7.0",
29     "grunt-contrib-concat": "^0.5.1",
30     "grunt-contrib-csslint": "^0.5.0",
31     "grunt-contrib-cssmin": "^0.14.0",
32     "grunt-contrib-htmlmin": "^1.0.0",
33     "grunt-contrib-jshint": "^0.11.3",
34     "grunt-contrib-uglify": "^0.11.0",
35     "grunt-data-uri": "^0.3.0",
36     "grunt-hashres": "^0.4.1",
37     "grunt-html": "^5.0.1",
38     "grunt-html-validation": "^0.1.18",
39     "grunt-image-embed": "^0.3.3",
40     "grunt-processhtml": "^0.3.13",
41     "grunt-wcag-accessibility": "^0.1.7"
42   },
43   "dependencies": {
44     "grunt-cli": "^1.4.3"
45   }
46 }
47
```

El archivo que vamos a correré siempre que ejecutemos el proyecto es el Gruntfile.

Dentro de éste nos encontramos la función principal que hace que Grunt funcione correctamente, que es el “module export”.

```
Gruntfile.js > .. module export=
1 // Grunt (property) export=: (grunt: any) => void
2 module.exports=function(grunt) {
3
```

Para ejecutar el proyecto tenemos que entrar en la carpeta en la que tenemos el proyecto con el CMD y poner `grunt -force`, para que fuerce la ejecución del proyecto aunque contenga errores.

```

C:\Users\Daw\Documents\grunt-optimized-web>grunt --force
>> Local Npm module "grunt-tenon" not found. Is it installed?

Running "clean:target1" (clean) task
>> 0 paths cleaned.

Running "cssmin:minify" (cssmin) task
>> No files created.

Running "cssmin:combine" (cssmin) task
Warning: The "path" argument must be of type string. Received undefined Used --force, continuing.

Running "concat:dist" (concat) task
File stage/js/app.min.js created.

Running "uglify:report" (uglify) task
>> Destination report not written because src files were empty.
>> No files created.

Running "uglify:target1" (uglify) task
>> 1 file created.

Running "processhtml:dist" (processhtml) task

Running "htmlmin:dist" (htmlmin) task
Minified 1 files

Done, but with warnings.

C:\Users\Daw\Documents\grunt-optimized-web>

```

Para poder realizar solo una tarea de los paquetes que aparecen instalados en el script tenemos que ejecutarlos como se especifique en el apartado específico para la tarea en la web de NPM.

## Using htmlhint

If you'd like to run htmlhint, we have a few options.

This module doesn't provide any interface on its own. It is highly recommended that you use one of the following modules to run the linter:

- `gulp-htmlhint` : a gulp interface for htmlhint
- `grunt-htmlhint` : a grunt plugin for htmlhint
- `htmlhint-cli` : a cli interface for htmlhint (NOTE: not complete at the moment)

Con el siguiente comando nos aparecerá las partes del HTML que necesitan una corrección para que nuestra página pueda ser accesible.

```

C:\Users\Daw\Documents\grunt-optimized-web>grunt htmlhint --force
>> Local Npm module "grunt-tenon" not found. Is it installed?

Running "htmlhint:src" (htmlhint) task
src\index.html [L20:C84] End tag "h1" seen, but there were open elements.
src\index.html [L20:C67] Unclosed element "span".
>> 1 file checked, 2 errors in 1 file
Warning: Task "htmlhint:src" failed. Used --force, continuing.

Done, but with warnings.

```

Podemos utilizar cualquiera de los paquetes que estén instalados en el Gruntfile.js.