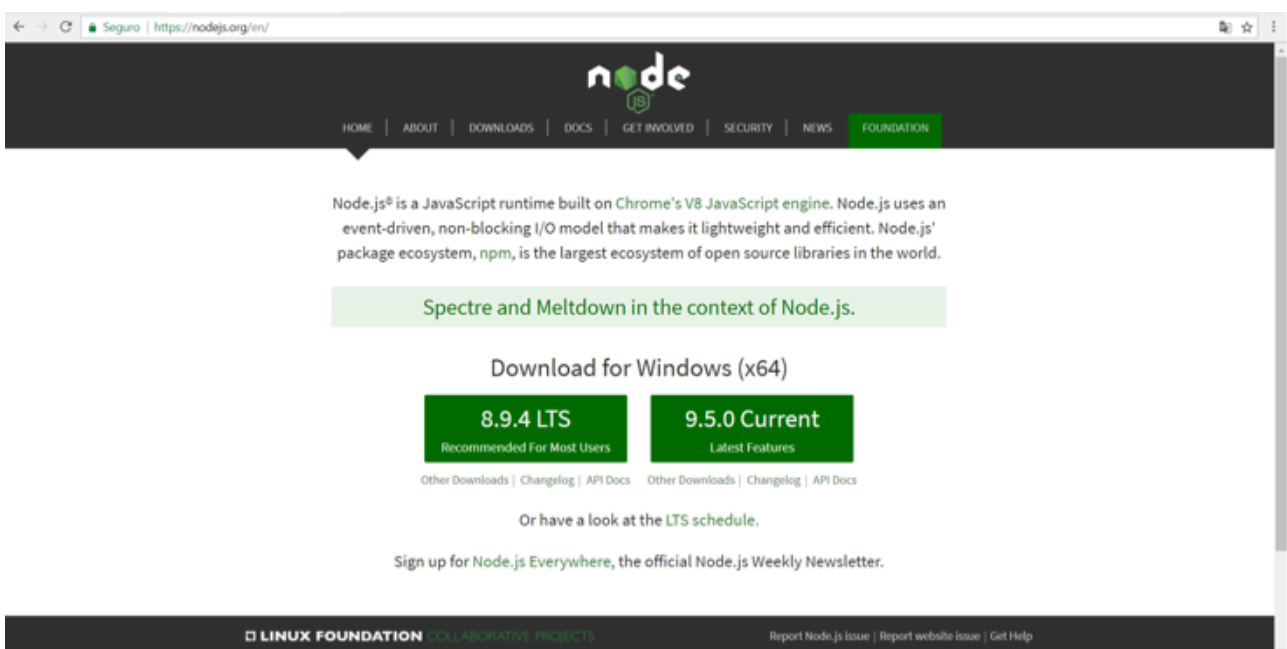
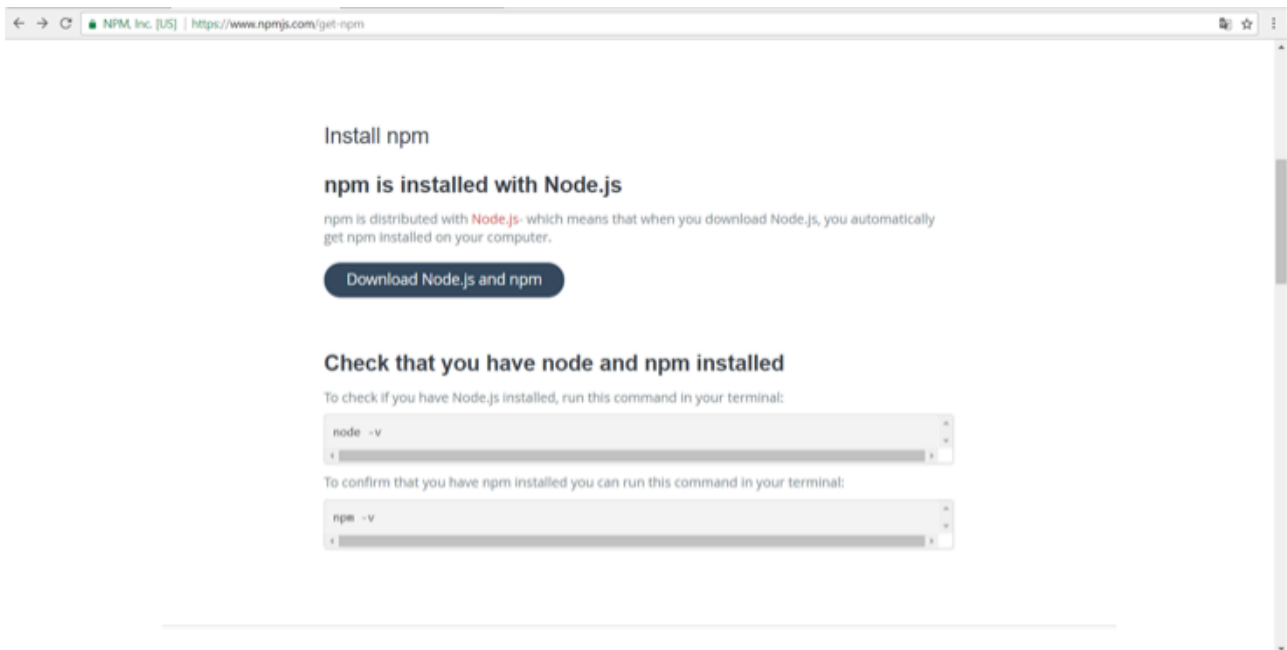
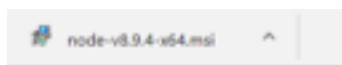


# Guía de Instalación Angular Windows

Para empezar necesitamos tener instalado el comando npm, el cual viene con Node.js, el cual te puedes descargar de la página web oficial <https://www.npmjs.com/get-npm>

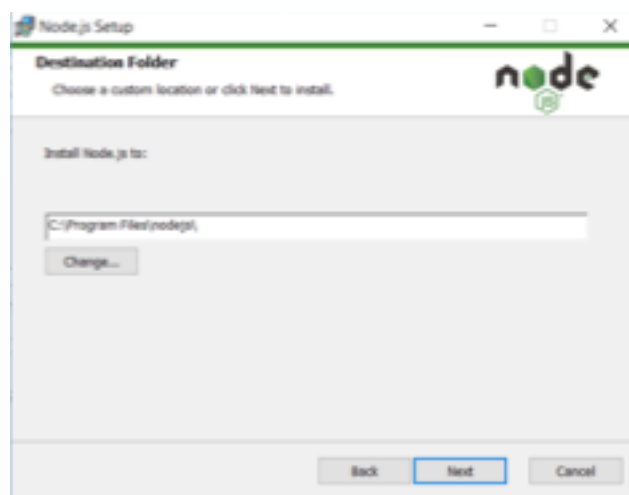
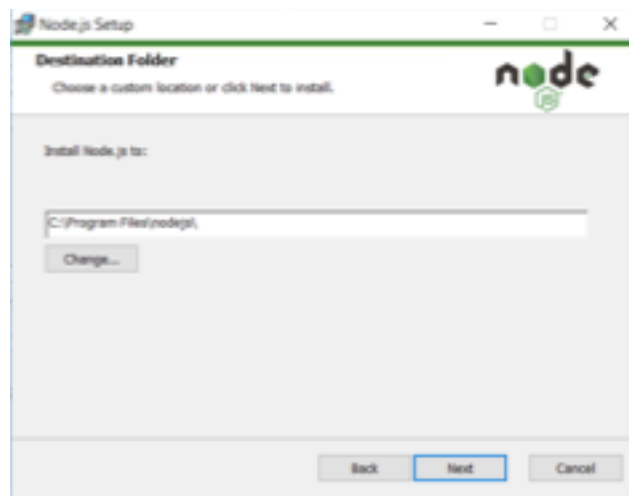


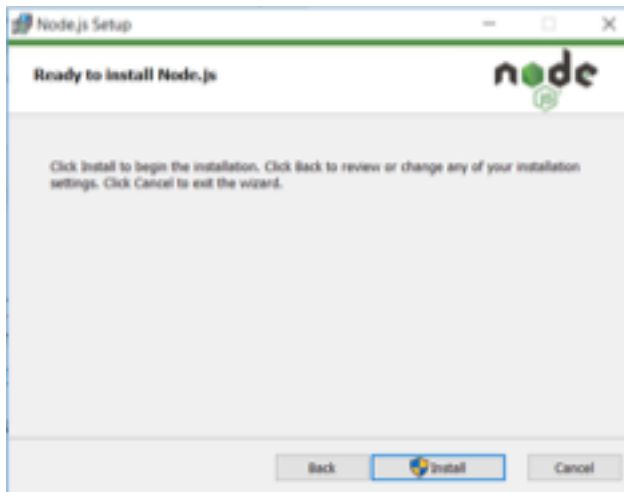
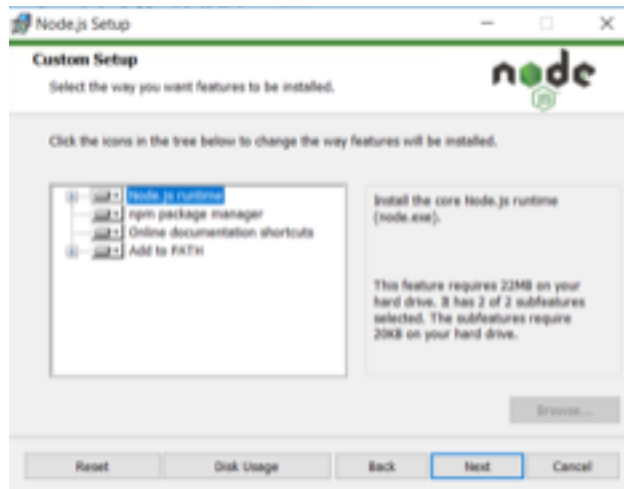
Para efectos de este tutorial me voy a descargar la versión 8.9.4, que sería la más estable. Aunque si deseas, descárgate la versión 9.5.0.

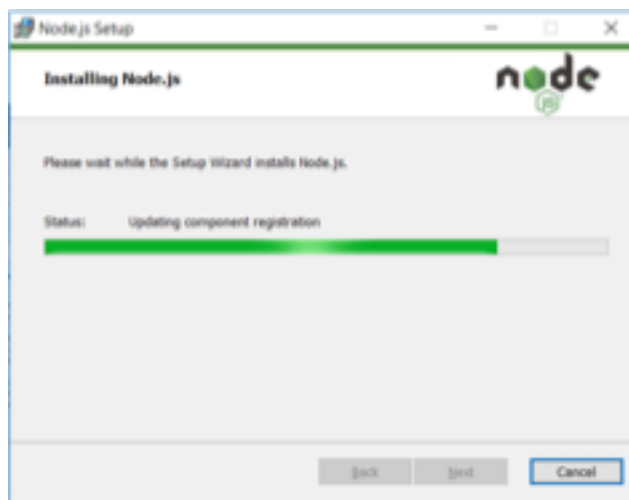


Una vez finalizada la descarga, ejecutamos el instalador. Lo único que tienes que hacer, es darle siguiente, siguiente, siguiente, como es costumbre en Windows. De todas formas te comparto la secuencia de pantallas por las que pase para poder instalar Node.js y npm.









npm se utiliza desde la línea de comandos, aunque Windows tiene su propio cmd (más conocido como Símbolo del Sistema), a mi me gusta utilizar Git Bash, que puedes descargar en el sitio oficial <https://git-scm.com/downloads>

Una vez instalado el npm junto a Node.js, y el Git Bash, nos vamos a este último para verificar que tengamos instalado Node y npm, para ello ejecutaremos dos comandos en la terminal:

```
node -v  
npm -v
```

```
itrjwyss@DESKTOP-RRR7TPU MINGW64 /  
$ node -v  
v8.9.4  
itrjwyss@DESKTOP-RRR7TPU MINGW64 /  
$ npm -v  
5.6.0  
itrjwyss@DESKTOP-RRR7TPU MINGW64 /  
$
```

Todo esto para poder instalar Angular CLI (Angular Command Line Interface), con la cual se crean y ejecutan proyectos Angular. Para la instalación de AngularCLI debemos ejecutar el siguiente comando:

```
npm install -g @angular/cli
```

```
MINGW64:/
itrjwyss@DESKTOP-RRR7TPU MINGW64 /
$ npm install -g @angular/cli
C:\Users\itrjwyss\AppData\Roaming\npm\ng -> C:\Users\itrjwyss\AppData\Roaming\np
m\node_modules\@angular\cli\bin\ng
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.1.3 (node_modules\@an
gular\cli\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@
1.1.3: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"}
)

+ @angular/cli@1.6.7
updated 813 packages in 324.475s

itrjwyss@DESKTOP-RRR7TPU MINGW64 /
$
```

Ten paciencia, puede que esta acción tarde unos cuantos minutos y parecerá que no está ocurriendo nada.

## Hola Mundo en Angular

Ahora empezaremos creando nuestro proyecto HelloWorld, para ello ejecutamos el siguiente comando:

**Nota:** Algo muy importante es que te encuentres dentro de la consola (Git Bash) en un directorio donde tengas permisos de escritura, como Documents (Documentos). También ten paciencia, que la primera vez se descargarán librerías que necesita el framework.

```
ng new HelloWorld
```

Una vez ejecutado el comando, a mi me salió lo siguiente:

```
$ ng new HelloWorld
create HelloWorld/e2e/app.e2e-spec.ts (293 bytes)
create HelloWorld/e2e/app.po.ts (208 bytes)
create HelloWorld/e2e/tsconfig.e2e.json (235 bytes)
create HelloWorld/karma.conf.js (923 bytes)
create HelloWorld/package.json (1295 bytes)
create HelloWorld/protractor.conf.js (722 bytes)
create HelloWorld/README.md (1026 bytes)
create HelloWorld/tsconfig.json (363 bytes)
create HelloWorld/tslint.json (3012 bytes)
create HelloWorld/.angular-cli.json (1246 bytes)
create HelloWorld/.editorconfig (245 bytes)
create HelloWorld/.gitignore (529 bytes)
create HelloWorld/src/assets/.gitkeep (0 bytes)
create HelloWorld/src/environments/environment.prod.ts (51 bytes)
create HelloWorld/src/environments/environment.ts (387 bytes)
create HelloWorld/src/favicon.ico (5430 bytes)
create HelloWorld/src/index.html (297 bytes)
create HelloWorld/src/main.ts (370 bytes)
create HelloWorld/src/polyfills.ts (2405 bytes)
```

```

create HelloWorld/src/styles.css (80 bytes)
create HelloWorld/src/test.ts (642 bytes)
create HelloWorld/src/tsconfig.app.json (211 bytes)
create HelloWorld/src/tsconfig.spec.json (283 bytes)
create HelloWorld/src/typings.d.ts (104 bytes)
create HelloWorld/src/app/app.module.ts (316 bytes)
create HelloWorld/src/app/app.component.html (1141 bytes)
create HelloWorld/src/app/app.component.spec.ts (986 bytes)
create HelloWorld/src/app/app.component.ts (207 bytes)
create HelloWorld/src/app/app.component.css (0 bytes)
Installing packages for tooling via npm.
npm WARN deprecated nodemailer@2.7.2: All versions below 4.0.1 of Nodemailer
are deprecated. See https://
nodemailer.com/status/
npm WARN deprecated node-uuid@1.4.8: Use uuid module instead

> uws@0.14.5 install C:
\Users\itrjwyss\Documents\Angular\HelloWorld\node_modules\uws
> node-gyp rebuild > build_log.txt 2>&1 || exit 0

> node-sass@4.7.2 install C:
\Users\itrjwyss\Documents\Angular\HelloWorld\node_modules\node-sass
> node scripts/install.js

Downloading binary from https://github.com/sass/node-sass/releases/download/
v4.7.2/win32-x64-57_binding.node
Download complete
Binary saved to C:
\Users\itrjwyss\Documents\Angular\HelloWorld\node_modules\node-
sass\vendor\win32-x64-57\binding.node
Caching binary to C:\Users\itrjwyss\AppData\Roaming\npm-cache\node-
sass\4.7.2\win32-x64-57_binding.node

> uglifyjs-webpack-plugin@0.4.6 postinstall C:
\Users\itrjwyss\Documents\Angular\HelloWorld\node_modules\webpack\node_modules\
uglifyjs-webpack-plugin
> node lib/post_install.js

> node-sass@4.7.2 postinstall C:
\Users\itrjwyss\Documents\Angular\HelloWorld\node_modules\node-sass
> node scripts/build.js

Binary found at C:
\Users\itrjwyss\Documents\Angular\HelloWorld\node_modules\node-
sass\vendor\win32-x64-57\binding.node
Testing binary
Binary is fine
npm WARN rollback Rolling back node-pre-gyp@0.6.39 failed (this is probably
harmless): EPERM: operation not permitted, rmdir 'C:
\Users\itrjwyss\Documents\Angular\HelloWorld\node_modules\fsevents\node_modules

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.1.3
(node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for
fsevents@1.1.3: wanted {"os":"darwin","arch":"any"} (current:
{"os":"win32","arch":"x64"})

added 1454 packages in 716.507s
Installed packages for tooling via npm.
Successfully initialized git.
Project 'HelloWorld' successfully created.

```

Ahora vamos a correr la aplicación por primera vez, para eso necesitamos estar dentro del nuevo directorio del proyecto, HelloWorld/

```
MINGW64:/c/Users/itrjwyss/Documents/Angular/HelloWorld

itrjwyss@DESKTOP-RRR7TPU MINGW64 ~/Documents/Angular
$ ls
HelloWorld/

itrjwyss@DESKTOP-RRR7TPU MINGW64 ~/Documents/Angular
$ cd HelloWorld/

itrjwyss@DESKTOP-RRR7TPU MINGW64 ~/Documents/Angular/HelloWorld (master)
$
```

Dentro de nuestro directorio HelloWorld/ ejecutaremos el siguiente comando

`ng serve`

```
MINGW64:/c/Users/itrjwyss/Documents/Angular/HelloWorld

itrjwyss@DESKTOP-RRR7TPU MINGW64 ~/Documents/Angular/HelloWorld (master)
$ ng serve
** NG Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **
Date:
Hash:
Time:    ms
chunk {inline} inline.bundle.js (inline) 5.79 kB [entry] [rendered]
chunk {main} main.bundle.js (main) 19.9 kB [initial] [rendered]
chunk {polyfills} polyfills.bundle.js (polyfills) 556 kB [initial] [rendered]
chunk {styles} styles.bundle.js (styles) 34 kB [initial] [rendered]
chunk {vendor} vendor.bundle.js (vendor) 7.41 MB [initial] [rendered]
webpack: Compiled successfully.
```

Una vez que te indique “webpack: Compiled successfully”, ya podemos probarlo en el navegador consumiendo <http://localhost:4200>



Welcome to app!



Here are some links to help you start:

- [Tour of Heroes](#)
- [CLI Documentation](#)
- [Angular blog](#)





Si lograste llegar invicto hasta este punto, felicidades. Tienes corriendo tu primera aplicación Angular (aunque sea un HelloWorld)