



¿Qué es Angular?

Originalmente surgió como un framework frontend para desarrollo web (lanzado por Google, 2010) que mantenía la filosofía MVC para crear y mantener [aplicaciones web en una sola página](#); hoy en día ya no es un framework sino una plataforma.

Para más información consultar:

<https://docs.angularjs.org/guide>

<https://angular.io/guide/architecture>

Aclaraciones

Angular 2 (versión final), Angular 4 y Angular 5 son similares por lo que puedes trabajar sin preocupación siempre y cuando tengas las últimas actualizaciones.

Configuración inicial (windows)

- 1- Ir a <https://nodejs.org/es/> y descargar la versión más reciente de node.js (por defecto la página detectará nuestro sistema operativo, por lo tanto solo hay que pulsar el **botón verde**).

Seguro | <https://nodejs.org/es/>

node.js

INICIO | ACERCA | DESCARGAS | DOCUMENTACIÓN | PARTICIPA | SEGURIDAD | NOTICIAS | **FUNDACIÓN**

Node.js® es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome. Node.js usa un modelo de operaciones E/S sin bloqueo y orientado a eventos, que lo hace liviano y eficiente. El ecosistema de paquetes de Node.js, npm, es el ecosistema mas grande de librerías de código abierto en el mundo.

Descargar para Windows (x64)

8.11.3 LTS Recomendado para la mayoría	10.6.0 Actual Últimas características
--	---

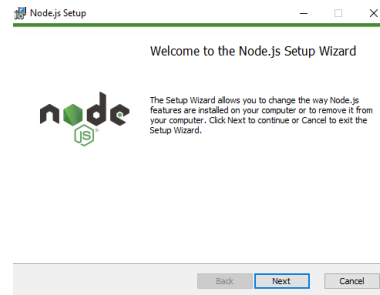
Otras Descargas | Cambios | Documentación del API | Otras Descargas | Cambios | Documentación del API

Ó revise la [Agenda de LTS](#).

- 2- Una vez instalado procederemos a ejecutarlo (solo pulsaremos el botón Next de tal manera que no cambiemos la configuración por default).



node-v8.11.3-x64
.msi



- 3- En caso de no tener instalado XAMPP lo instalas (link de la descarga:
<https://www.apachefriends.org/es/download.html>)
- 4- Buscar PowerShell en la barra de tareas y pulsar el siguiente botón:
- 5- Ir a la carpeta donde se ubica XAMPP (C:\xampp\htdocs) y copiar esa dirección
- 6- Regresando a PowerShell lo que tenemos que hacer es escribir: `cd C:\xampp\htdocs` (nota: en mis ejemplos no tomé en cuenta este paso).
- 7- En caso de tener vestigios de angular escribir los siguientes comandos en la terminal (de lo contrario pasar al punto 8):

```
npm uninstall -g angular-cli
npm uninstall -g @angular/cli
npm cache clean o sino npm cache clean --force
```

Ejemplo:

```
PS C:\WINDOWS\system32> npm uninstall -g angular-cli
up to date in 0.06s
PS C:\WINDOWS\system32> npm uninstall -g @angular/cli
removed 246 packages in 7.337s
PS C:\WINDOWS\system32> npm cache clean
```

- 8- Para tener la última versión de angular escribiremos el siguiente comando:
`npm install -g @angular/cli@latest`

Ejemplo:

```
PS C:\WINDOWS\system32> npm install -g @angular/cli@latest
C:\Users\Erick.LAPTOP-3QCCTC07\AppData\Roaming\npm\ng -> C:\Users\Erick.LAPTOP-3QCCTC07\AppData\Roaming\npm\node_modules\@angular\cli\bin\ng

> @angular/cli@6.0.8 postinstall C:\Users\Erick.LAPTOP-3QCCTC07\AppData\Roaming\npm\node_modules\@angular\cli
> node ./bin/ng-update-message.js

npm WARN SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.4 (node_modules\@angular\cli\node_modules\fsevents):
npm WARN SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.4: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

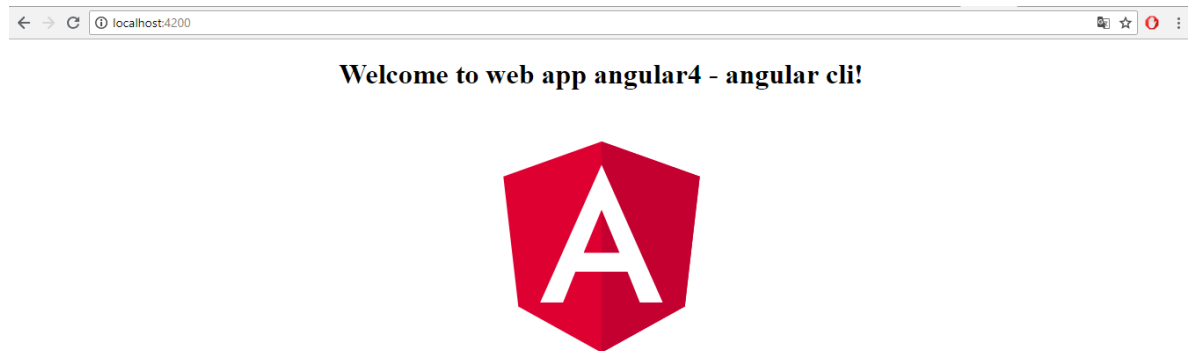
+ @angular/cli@6.0.8
added 246 packages in 50.342s
```

- 9- Creamos un nuevo proyecto con el comando: `ng new escribeElNombreDeTuProyecto` (aquí vamos a tener una versión parecida a la de este repositorio
<https://github.com/angular/quickstart>)

Ejemplo:

```
PS C:\xampp\htdocs> ng new CursoAngular4
```

- 10- Finalmente para arrancar tu proyecto teclea: **npm start** y escribe la siguiente dirección en tu navegador **http://localhost:4200/** (te deberá aparecer algo parecido, en mi caso lo he editado).



Here are some links to help you start:

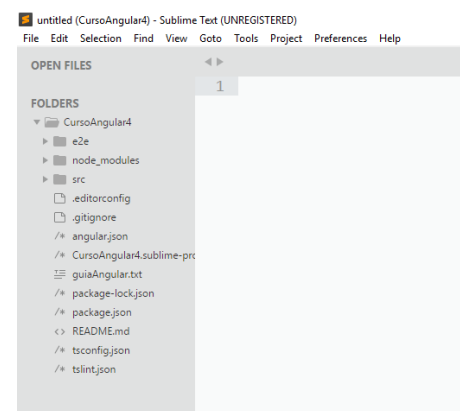
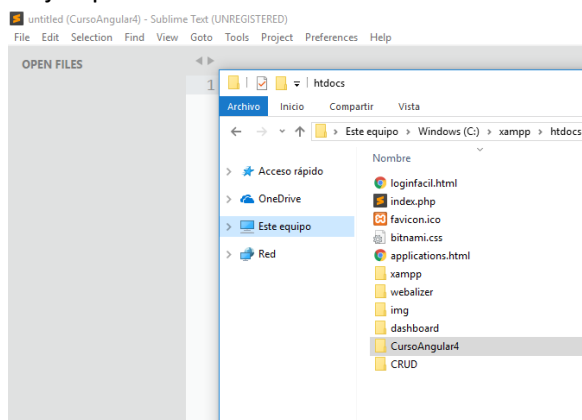
- [Tour of Heroes](#)
- [CLI Documentation](#)
- [Angular blog](#)

- 11- Abre SublimeText3 (y sino tienes instalado package control lo instalas link del manual:) <https://goo.gl/nvngcT>). Una vez abierto, teclea **ctrl + Shift + p** y selecciona la siguiente opción: **Package Control: Install Package**

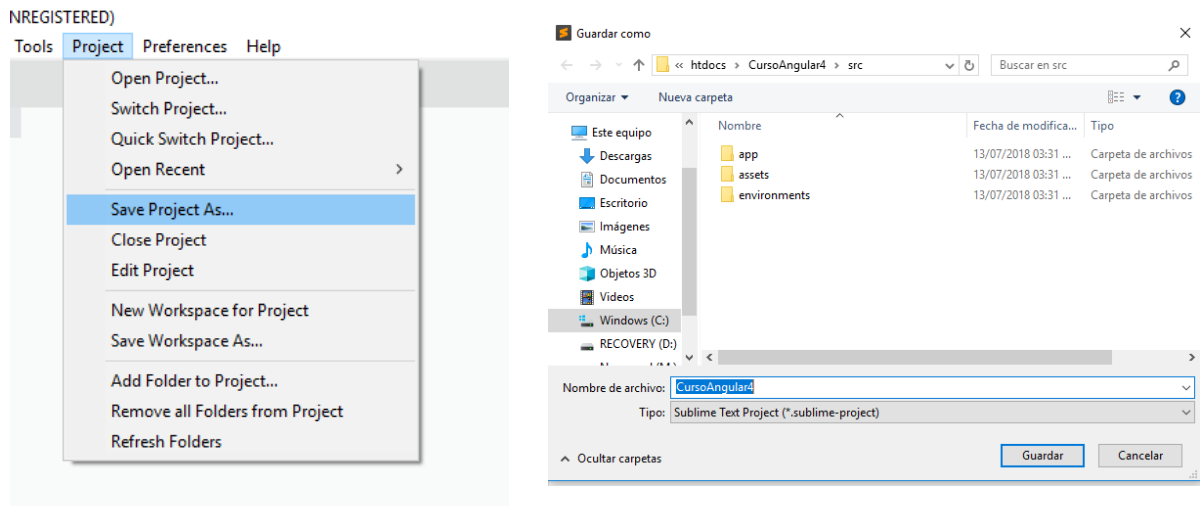
- 12- Escribe **TypeScript** y luego pulsa sobre la opción que te sugiere para instalarlo.

- 13- Dirigete a **C:\xampp\htdocs** y localiza tu proyecto para arrastrarlo a SublimeText3.

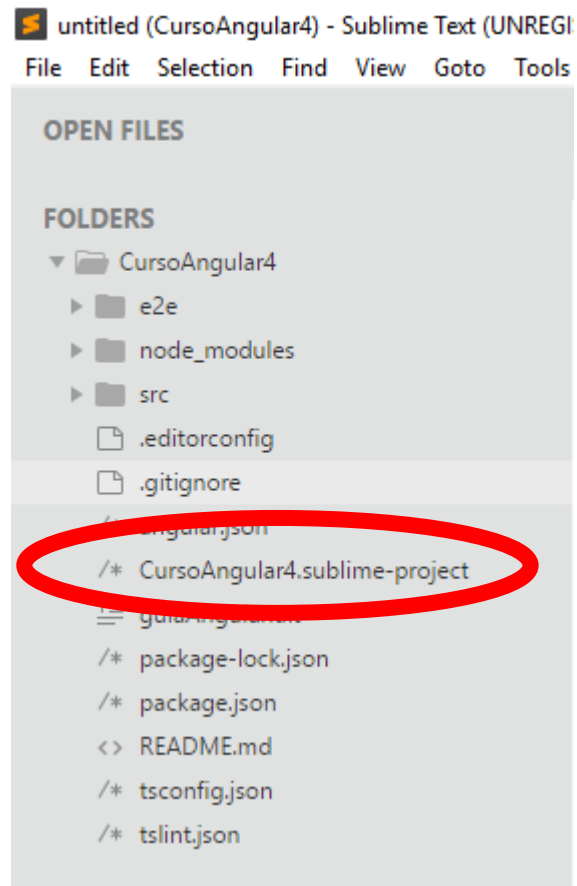
Ejemplo:



14- Finalmente guarda tu proyecto de la siguiente manera:



Revisa que tengas el siguiente archivo:



Definiciones para entender Angular

COMPONENTES

Un componente al final va a controlar un trozo de pantalla o de la vista.

Todo lo que se puede ver en pantalla es controlado y gestionado por este tipo de elementos.

La lógica de un componente dentro una clase en Angular 2 es que da soporte a una vista interactuando con ella a través de un API con propiedades y métodos.

El componente hace de mediador entre la vista a través de la plantilla y la lógica de la app donde se incluirá el modelo de datos, es decir una especie de controlador.

PLANTILLAS

Las plantillas van a definir la vista de los componentes.

Son htmls y tienen sintaxis especial de Angular. Trabajando con el databinding y las directivas.

DECORADORES Y METADATOS

Con los decoradores(patrn de diseo) vamos a configurar dinamicamente atributos/metadatos de las clases y componentes.

Los metadatos van a describir a las clases pero también describen relaciones, por ejemplo si tenemos un componente y una plantilla el metadato se va a encargar de decirle a Angular que ese componente y esa plantilla van juntos, entre otras muchas cosas.

SERVICIOS

Son clases con un objetivo claro, facilita la reutilización, son un tipo de componente dentro de la arquitectura de Angular 2 y mediante la inyección de dependencias los podemos usar en otros componentes principales digamos.

PROVIDERS

Son servicios que nos proveen de datos o funcionalidades mediante sus métodos. Existen providers/servicios propios de Angular o creados por nosotros mismos.

DIRECTIVAS

Son funcionalidades aplicables al DOM y a los elementos HTML en las plantillas de un componente. Por ejemplo una directiva puede servir para controlar que un div se muestre o no o recorrer un array en la vista (directivas estructurales, estructuras condicionales y de control) o incluso también puede servir para darle una un estilo u otro a un elemento del HTML o también para interactuar con el modelo de datos del componente.

Básicamente son nuevos atributos para aplicarle a cualquier cosa en nuestra plantilla/vista.