



CREANDO SU PRIMERA APLICACIÓN WEB EN ANGULAR

- Angular es un framework de JavaScript moderno y popular que puede ejecutarse en diferentes plataformas adicionales a la web, como computadoras de escritorio y dispositivos móviles. Las aplicaciones están escritas en TypeScript, un superconjunto de JavaScript que proporciona mejoras sintácticas, como técnicas de escritura fuerte y orientadas a objetos.
- Las aplicaciones se crean y desarrollan utilizando una herramienta de línea de comandos creada por el equipo de Angular llamada Angular CLI. Automatiza muchas tareas de desarrollo, como pruebas e implementación, que llevaría mucho tiempo configurar manualmente.
- La popularidad de Angular se refleja considerablemente en su amplio soporte de herramientas. El editor de Visual Studio Code (VSCode) contiene varias extensiones que mejoran la experiencia de desarrollo cuando se trabaja con Angular

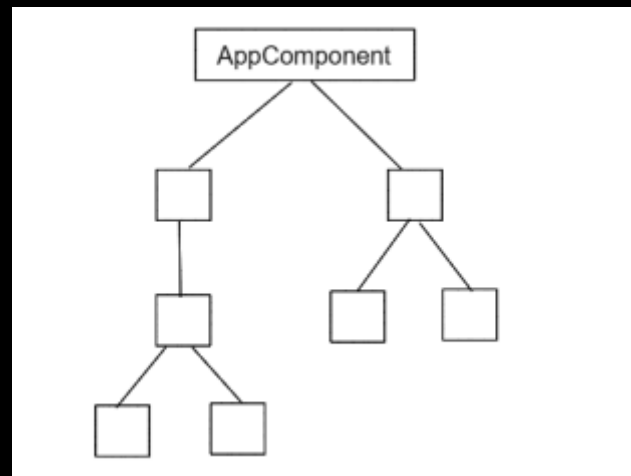
CONTENIDO

- Introducción a Angular
- Introducción a la CLI de Angular
- Explorando el rico ecosistema de herramientas angulares en VSCode
- Cómo crear nuestra primera aplicación Angular
- Cómo usar la consola Nx para automatizar los comandos CLI de Angular

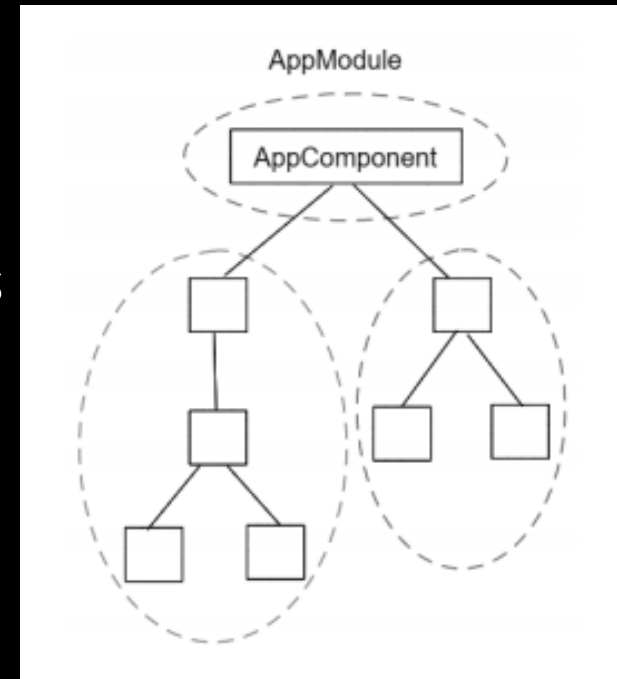
TEORÍA Y CONTEXTO ESENCIALES


- El framework Angular es multiplataforma que puede ejecutarse en una amplia gama de entornos, incluidos la web, los servidores, el dispositivo móvil y el escritorio.
- Consiste en una colección de bibliotecas de JavaScript que podemos usar para crear aplicaciones web escalables y de alto rendimiento.
- La arquitectura de una aplicación Angular se basa en una representación jerárquica de componentes. Los componentes son los bloques de construcción fundamentales de una aplicación Angular. Representan y controlan una parte particular de una página web llamada vista


- Algunos ejemplos de componentes son los siguientes:
 - Una lista de publicaciones de blog
 - Un formulario de notificación de problemas
 - Un widget de visualización del tiempo
- Los componentes de una aplicación Angular se pueden organizar lógicamente como un árbol:



- Una aplicación Angular normalmente tiene un componente principal, llamado AppComponent, por convención. Cada componente del árbol puede comunicarse e interactuar con sus hermanos.
- Una aplicación Angular puede tener muchas características que se denominan módulos. Cada módulo tiene un bloque de funcionalidad única que corresponde a un dominio de aplicación o flujo de trabajo en particular. Los módulos se utilizan para agrupar componentes que comparten una funcionalidad similar:



- 
- En el diagrama anterior, los círculos de línea discontinua representan módulos. Una aplicación Angular normalmente tiene un módulo principal, llamado AppModule. Cada módulo puede importar otros módulos en una aplicación Angular si desea utilizar parte de su funcionalidad.
 - La funcionalidad de un módulo se puede analizar más a fondo en la lógica de presentación y de negocios de una característica. Los componentes solo deben ser responsables de manejar la lógica de presentación y delegar tareas de lógica de negocios a los servicios. Angular proporciona servicios a componentes mediante un mecanismo integrado de inyección de dependencia (DI).

- 
- Angular DI utiliza objetos de propósito especial, llamados inyectores, para ocultar gran parte de la complejidad de proporcionar dependencias a una aplicación Angular. No se requiere que los componentes conozcan la implementación real de un servicio Angular. Solo necesitan pedirlo a un inyector.
 - Un servicio Angular debe seguir el principio de responsabilidad única y no debe cruzar fronteras entre diferentes módulos. Algunos ejemplos de servicios son los siguientes:
 - Acceder a los datos desde una API de backend mediante el protocolo HTTP.
 - Interactuar con el almacenamiento local del navegador.
 - Registro de errores.
 - Transformaciones de datos

INTRODUCCIÓN A ANGULAR CLI

- Angular CLI es una herramienta creada por el equipo de Angular que mejora la experiencia del desarrollador al crear aplicaciones Angular. Oculta gran parte de la complejidad de la organización y la configuración de una aplicación al tiempo que permite al desarrollador concentrarse en lo que mejor saben hacer: **¡codificar!** Antes de que podamos comenzar a usar Angular CLI, debemos configurar los siguientes requisitos previos en nuestro sistema:
 - Node.js: un tiempo de ejecución de JavaScript que se basa en el motor v8 de Chrome. Puede descargar cualquier versión de soporte a largo plazo (LTS) desde <https://nodejs.org/en>.
 - npm: un administrador de paquetes para el tiempo de ejecución de Node.js.


- Luego podemos instalar Angular CLI usando npm desde la línea de comando

```
PS C:\Users\Geovany> npm install -g @angular/cli
```

- Podemos usar la opción -g para instalar Angular CLI de forma global ya que queremos crear aplicaciones desde cualquier ruta de nuestro sistema operativo.

- Para verificar que Angular CLI se haya instalado correctamente, podemos ejecutar lo siguiente desde la línea de comando:


```
PS C:\Users\Geovany> ng version
```



```
Angular CLI: 12.2.0  
Node: 12.18.3  
Package Manager: npm 6.14.6  
OS: win32 x64  
  
Angular:  
...  


| Package                    | Version             |
|----------------------------|---------------------|
| @angular-devkit/architect  | 0.1202.0 (cli-only) |
| @angular-devkit/core       | 12.2.0 (cli-only)   |
| @angular-devkit/schematics | 12.2.0 (cli-only)   |
| @schematics/angular        | 12.2.0 (cli-only)   |

  
PS C:\Users\Geovany> █
```

- 
- El comando anterior informará la versión de Angular CLI que se ha instalado en nuestro sistema. Angular CLI proporciona una interfaz de línea de comandos a través del comando ng, que es el ejecutable binario de Angular CLI. Puede aceptar varias opciones, incluidas las siguientes:
 - serve: crea y ejecuta una aplicación Angular.
 - build: construye una aplicación Angular.
 - test: Ejecuta las pruebas unitarias de una aplicación Angular.
 - generate: genera un nuevo artefacto angular, como un componente o módulo.
 - add: instala una biblioteca de terceros que sea compatible con Angular.
 - new: crea una nueva aplicación angular

HERRAMIENTAS EN VSCODE

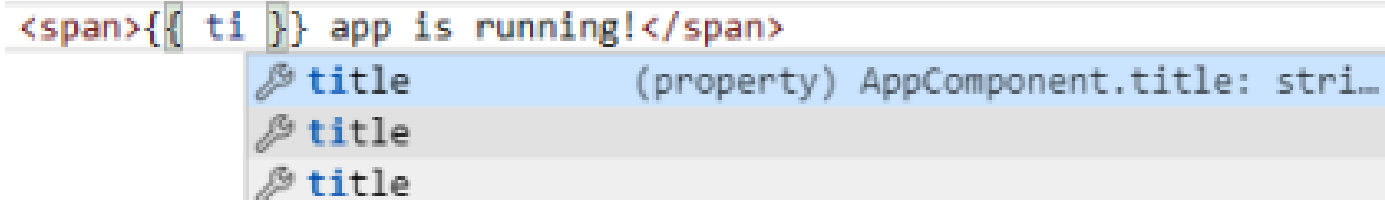
- Hay muchas extensiones disponibles en VSCode Marketplace que mejoran el ecosistema de herramientas de Angular. En esta sección, aprenderemos sobre los más populares que pueden ayudarnos significativamente en el desarrollo de Angular:
 - Nx Console
 - Angular Language Service
 - Angular Snippets
 - Angular Evergreen
 - Material Icon Theme

NX CONSOLE

- La consola Nx es una extensión de VSCode desarrollada por el equipo de Nrwl que proporciona una interfaz gráfica de usuario sobre Angular CLI.
- Contiene la mayoría de los comandos de Angular CLI y la utiliza internamente para ejecutar cada uno. Aprenderemos más sobre esta extensión en la sección un poco más adelante xD.

ANGULAR LANGUAGE SERVICE


- La extensión Angular Language Service proporciona varias mejoras al editar plantillas HTML en una aplicación Angular, incluidas las siguientes:
 - Autocompletar código
 - Compilar mensajes de error
 - Técnicas de definición de referencia
- El autocompletado de código es una función que nos ayuda a encontrar la propiedad o el método adecuados para usar mientras escribimos. Funciona mostrando una lista de sugerencias mientras comenzamos a escribir contenido HTML.



The screenshot shows an IDE with an Angular template. The code is `{{ ti }} app is running!`. The cursor is positioned after the closing curly brace of the interpolation. A dropdown menu of suggestions is visible, showing three entries, each with a key icon and the text `title`. The first entry is highlighted and includes the text `(property) AppComponent.title: stri...` to its right.

```
<span>{{ ti }} app is running!</span>
```

- 🔑 title (property) AppComponent.title: stri...
- 🔑 title
- 🔑 title

- 
- En la imagen anterior, cuando comenzamos a escribir la palabra `ti`, Angular Language Service sugiere la propiedad del componente de título. Tenga en cuenta que la finalización de código solo funciona para las propiedades y métodos públicos de un componente.
 - Uno de los problemas más comunes al desarrollar aplicaciones web es la detección de errores antes de que la aplicación llegue a producción. Este problema se puede resolver parcialmente con el compilador Angular, que se inicia al construir una aplicación Angular para producción. Además, Angular Language Service puede llevar esto más lejos mostrando mensajes de error de compilación mucho antes de que nuestra aplicación llegue al proceso de compilación.

any

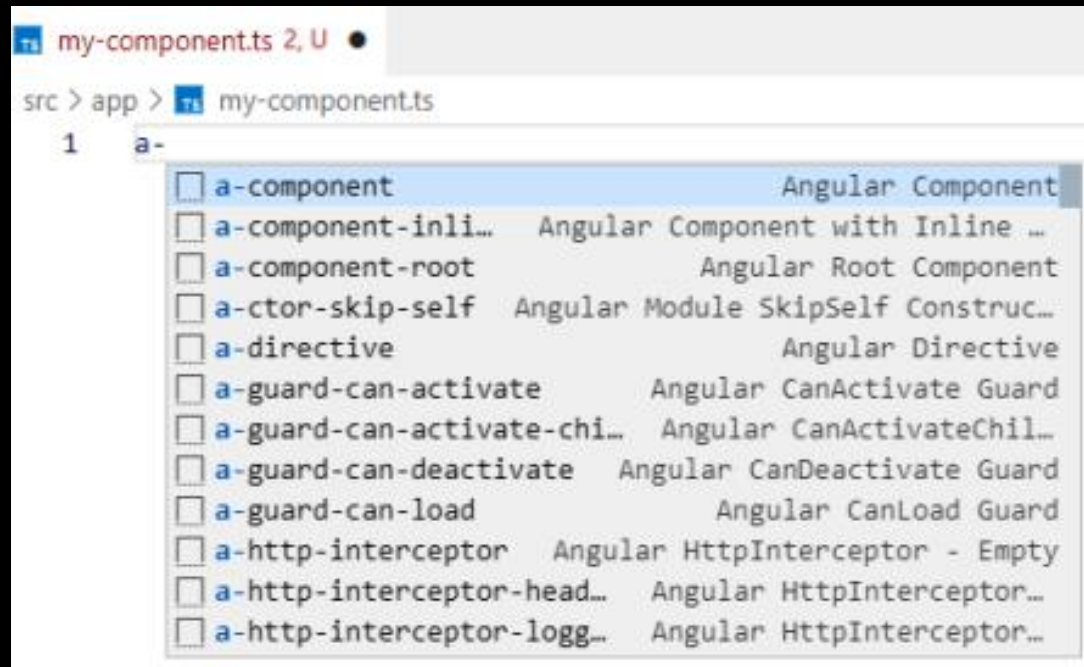
Property 'tile' does not exist on type 'AppComponent'. Did you mean 'title'? ngts(2551)

[View Problem \(Alt+F8\)](#) No quick fixes available

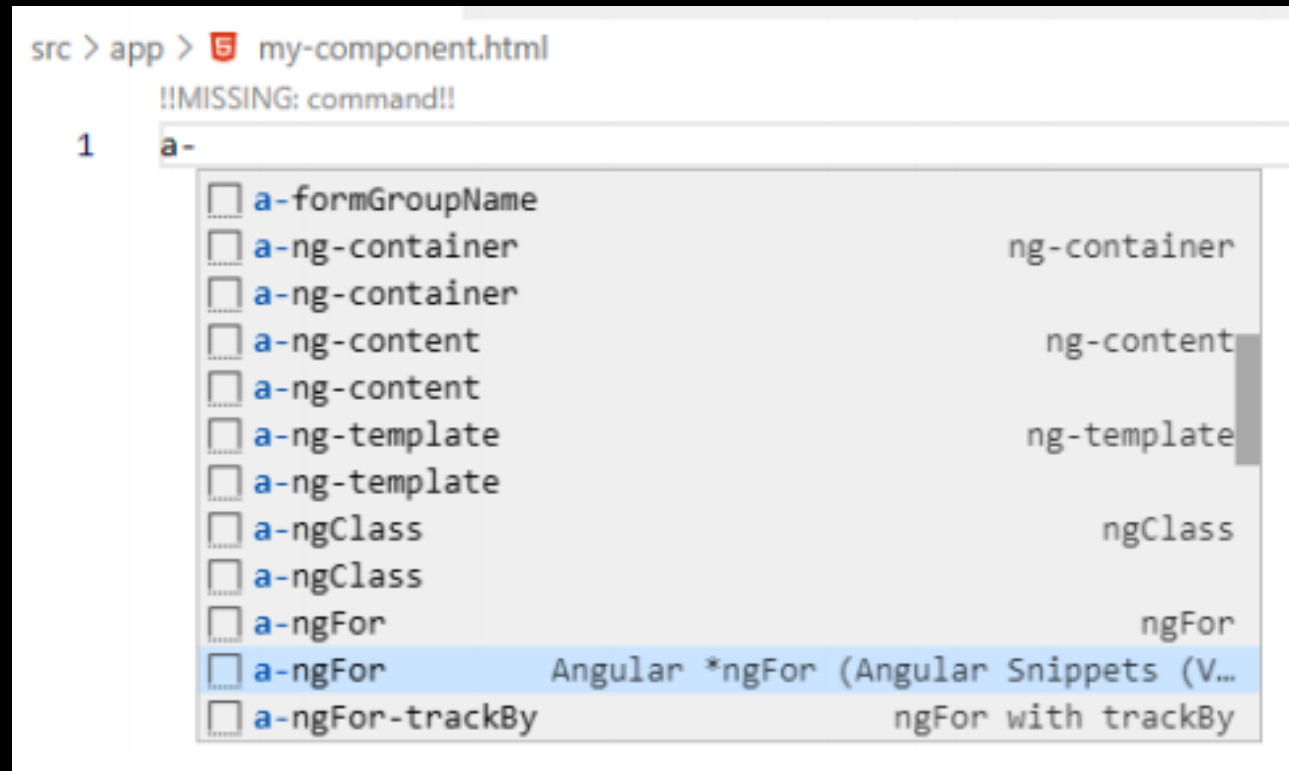
```
<span>{{ tile }} app is running!</span>
```

ANGULAR SNIPPETS

- La extensión Angular Snippets contiene una colección de fragmentos de código Angular para TypeScript y HTML. En TypeScript, podemos usarlo para crear componentes, módulos o servicios en un archivo TypeScript en blanco:




- En una plantilla HTML, podemos usar la extensión para crear elementos útiles, como la directiva * ngFor, para recorrer una lista en HTML:



ANGULAR EVERGREEN

- Uno de los factores principales que hace que Angular sea tan estable es que sigue un ciclo de lanzamiento regular basado en versiones semánticas. Si queremos que nuestras aplicaciones Angular estén empaquetadas con las últimas funciones y correcciones, debemos actualizarlas con regularidad. Pero, ¿cómo podemos mantenernos actualizados de la manera más eficiente? ¡Podemos usar la extensión Angular Evergreen para eso!
- Compara las versiones Angular y Angular CLI de un proyecto Angular CLI con las más recientes y le advierte si necesita actualizarlo:

- 
- Proporciona una interfaz de usuario fácil de usar para ejecutar los siguientes comandos:
 - Actualización de dependencias de Angular a la última versión
 - Actualización de dependencias de Angular a la próxima versión
 - Actualización de todas las dependencias de npm
 - Angular Evergreen es la extensión perfecta para no quedarse nunca desactualizado con sus proyectos Angular.

MATERIAL ICON THEME

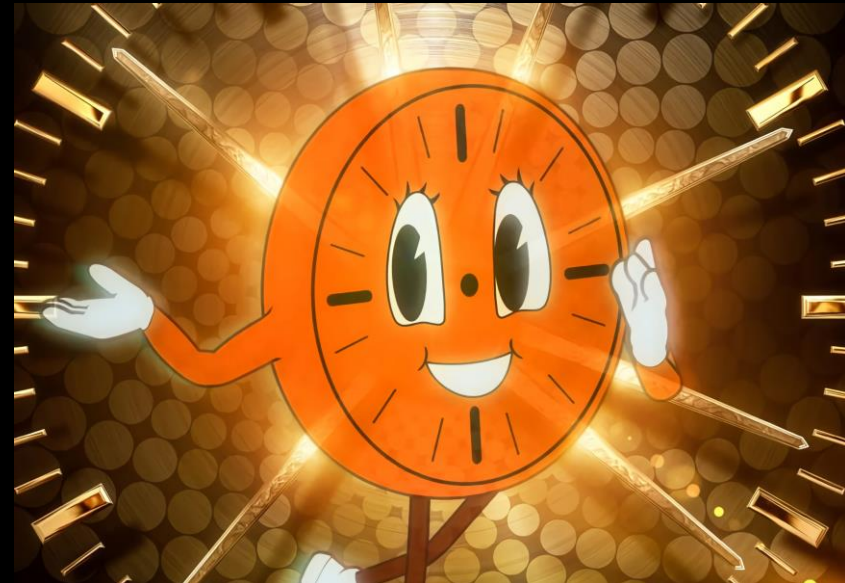
- La última extensión de la lista no agrega mucho valor con respecto a la productividad del desarrollador. En cambio, se centra en la visibilidad y el punto de vista estético modificando el tema del icono de VSCode.
- El tema del icono de material contiene un montón de iconos que se basan en el diseño de materiales. Puede comprender el tipo de cada archivo en su proyecto y mostrar el icono relacionado automáticamente.
- Por ejemplo, los módulos se muestran con un icono rojo, mientras que los componentes se muestran con un icono azul.
- VSCode tiene un tema de icono de archivo predeterminado llamado Seti. Una vez que haya instalado Material Icon Theme, le pedirá que seleccione cuál le gustaría activar.

DESCRIPCIÓN DEL PROYECTO



- En este proyecto, usaremos Angular CLI para crear una nueva aplicación Angular desde cero. Luego, interactuaremos con la funcionalidad principal de Angular para realizar un cambio simple en nuestra aplicación. Finalmente, aprenderemos cómo usar la extensión de la consola Nx para construir y ejecutar nuestra aplicación.

- Tiempo de construcción: 15 minutos.




INICIAMOS

- Se requieren las siguientes herramientas de software para completar este proyecto:
 - Git: un sistema de control de versiones distribuido de código abierto y gratuito. Puedes descargarlo de <https://git-scm.com>.
 - VSCode: un editor de código que puede descargar desde el código <https://visualstudio.com/Download>.
 - Angular CLI: la interfaz de línea de comandos para Angular que vimos anteriormente.

CREANDO NUESTRA PRIMERA APLICACIÓN ANGULAR

- Para crear una nueva aplicación Angular, debemos ejecutar el comando `ng new`, pasando el nombre de la aplicación como un parámetro:

```
PS C:\Users\Geovany\Desktop\Angular> ng new miPrimerAppAngular
```


- 
- El nuevo comando se usa para crear una nueva aplicación Angular o un nuevo espacio de trabajo Angular.
 - Un espacio de trabajo angular es un proyecto CLI angular que contiene una o más aplicaciones angulares, donde algunas de ellas pueden ser bibliotecas. Entonces, cuando ejecutamos el comando `ng new`, creamos un espacio de trabajo Angular con una aplicación Angular por defecto.
 - En el comando anterior, el nombre de nuestra aplicación Angular es `miPrimerAppAngular`. Al ejecutar el comando, CLI hará algunas preguntas para recopilar la mayor cantidad de información posible sobre la naturaleza de la aplicación que queremos crear.
 - Vamos a ver:

- Inicialmente, nos preguntará si queremos habilitar el enrutamiento en nuestra aplicación Angular.

```
P5 C:\Users\Geovany\Desktop\Angular> ng new miPrimerAppAngular  
? Would you like to add Angular routing? (y/N) _
```

- El enrutamiento en Angular se trata de navegar entre los componentes de una aplicación Angular usando una URL. En este proyecto, no nos preocupamos por el enrutamiento, así que presione Enter para aceptar el valor predeterminado.

- Luego, Angular CLI nos pide que seleccionemos el formato de estilo que queremos usar en toda la aplicación Angular:

```
PS C:\Users\Geovany\Desktop\Angular> ng new miPrimerAppAngular
? Would you like to add Angular routing? No
? Which stylesheet format would you like to use? (Use arrow keys)
> CSS
SCSS   [ https://sass-lang.com/documentation/syntax#scss ]
Sass   [ https://sass-lang.com/documentation/syntax#the-indented-syntax ]
Less   [ http://lesscss.org ]
```

- Seleccione un formato de la lista de hojas de estilo disponibles y presione Enter.
- Puede utilizar las teclas de flecha de su teclado para navegar entre las opciones de hoja de estilo disponibles.

- Angular CLI inicia el proceso de creación de su aplicación, que consta de lo siguiente:
 - Generación de la estructura de carpetas necesaria de un proyecto típico
 - Instalar las dependencias npm y los paquetes angular necesarios
 - Inicializando Git en el proyecto CLI de Angular
- Este proceso puede llevar algún tiempo, dependiendo de la velocidad de su red. Una vez que haya terminado, debe tener una nueva carpeta llamada `miPrimerAppAngular` en la ruta donde ejecutó el comando `ng new`

- Ahora, finalmente ha llegado el momento de ejecutar nuestra aplicación Angular y verla en acción:
 1. Abra una ventana de Terminal y navegue hasta la carpeta miPrimerAppAngular.
 2. Ejecute el siguiente comando en Angular CLI:

```
PS C:\Users\Geovany\Desktop\Angular\miPrimerAppAngular> ng serve
```

- Esto construirá la aplicación Angular e iniciará un servidor web integrado que podemos usar para obtener una vista previa. El servidor web se inicia en modo reloj; es decir, reconstruye automáticamente la aplicación Angular cada vez que hacemos un cambio en el código.
- La primera vez que se crea una aplicación Angular, se tarda una cantidad considerable de tiempo en completarse, por lo que debemos ser pacientes. Sabrá cuando el proceso ha finalizado sin errores cuando vea el siguiente mensaje en la ventana de Terminal:

```
Build at: 2021-08-09T18:56:06.446Z - Hash: 0c1f1d679846e28d7952 - Time: 59774ms

** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **

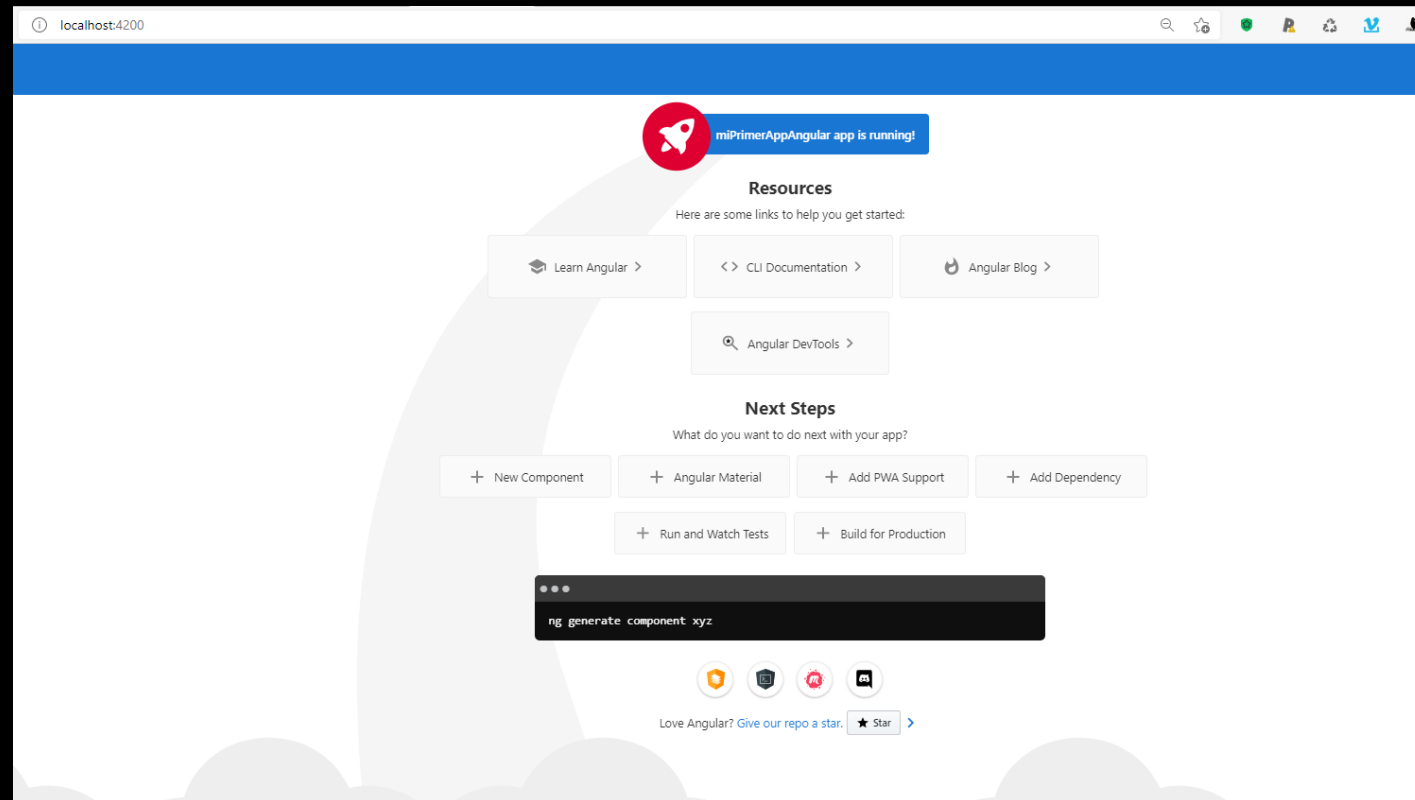
✓ Compiled successfully.
✓ Browser application bundle generation complete.


5 unchanged chunks

Build at: 2021-08-09T18:56:08.180Z - Hash: a97ab4413fbbad3eee08 - Time: 1078ms

✓ Compiled successfully.
```


- Inicie su navegador favorito y navegue hasta <http://localhost:4200> para obtener una vista previa de su nueva aplicación Angular:



- 
- Angular CLI crea una aplicación Angular mínima de forma predeterminada para proporcionarnos un punto de partida para nuestro proyecto Angular.
 - Contiene algunos estilos CSS y contenido HTML listos para usar, que aprenderemos a cambiar de acuerdo con nuestras especificaciones un poco más adelante.

INTERACTUAR CON ANGULAR


- Cuando trabajamos con Angular, la verdadera diversión comienza cuando nos ensuciamos las manos con el framework en sí. Comprender cómo funciona Angular y escribir el código de la aplicación es lo que importa, después de todo.
- El código fuente de la aplicación reside dentro de la carpeta `src\app`, en la raíz de nuestro proyecto.
- Contiene todos los archivos necesarios para construir y probar nuestra aplicación Angular, incluyendo un componente y un módulo. El componente es el componente principal de la aplicación Angular:

 app.component.ts ✕

src > app >  app.component.ts

```
1  import { Component } from '@angular/core';
2
3  @Component({
4    selector: 'app-root',
5    templateUrl: './app.component.html',
6    styleUrls: ['./app.component.css']
7  })
8  export class AppComponent {
9    title = 'miPrimerAppAngular';
10 }
11
```

- Las siguientes propiedades caracterizan un componente angular:
 - selector: un nombre único que se utiliza para identificar y declarar el componente dentro del contenido HTML. Se utiliza como una etiqueta HTML, como cualquier elemento HTML nativo, como `<app-root> </app-root>`.
 - templateUrl: la ruta que apunta a un archivo HTML que contiene el contenido HTML del componente, que se denomina plantilla del componente.
 - styleUrls: una lista de rutas donde cada una apunta a un archivo de hoja de estilo que contiene los estilos CSS del componente.


- 
- Las propiedades anteriores se definen utilizando el decorador `@Component`.
 - Es una función que decora la clase TypeScript del componente y la reconoce como un componente Angular.
 - La propiedad `title` de la clase `AppComponent` es una propiedad pública que contiene un valor de cadena y se puede usar en la plantilla del componente.

app.module.ts X

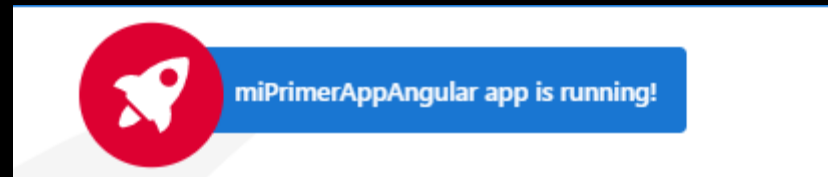
src > app > app.module.ts > ...


```
1  import { NgModule } from '@angular/core';      You, an hour ago • initial commit
2  import { BrowserModule } from '@angular/platform-browser';
3
4  import { AppComponent } from './app.component';
5
6  @NgModule({
7    declarations: [
8      AppComponent
9    ],
10   imports: [
11     BrowserModule
12   ],
13   providers: [],
14   bootstrap: [AppComponent]
15 })
16 export class AppModule { }
17
```

- El módulo principal de nuestra aplicación Angular usa un decorador similar llamado @NgModule para definir sus propiedades:

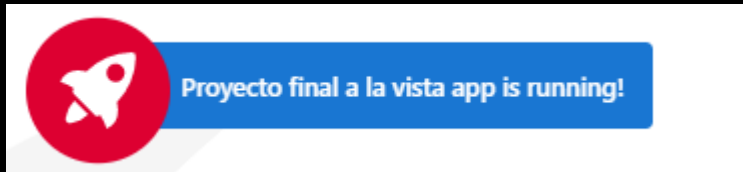
- 
- El decorador de un módulo angular define un conjunto de propiedades que se pueden utilizar para configurar el módulo. Los más comunes son los siguientes:
 - declarations: Define los componentes que forman parte del módulo Angular. Cada componente que existe en el módulo Angular debe agregarse a la matriz de declaraciones.
 - imports: define otros módulos que contienen la funcionalidad que necesita el módulo angular.

- Mojemos nuestros pies ahora modificando el código de nuestra aplicación Angular.
- Cambiaremos el siguiente mensaje de saludo, que se muestra al iniciar la aplicación, por algo más significativo:



- 
- Primero, necesitamos encontrar la ubicación donde se declara el mensaje en la captura de pantalla anterior.
 - El componente principal de una aplicación angular es el componente que se carga al inicio de la aplicación de forma predeterminada

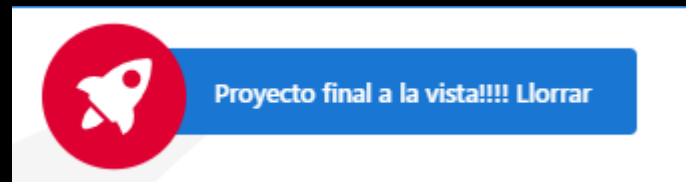
- Por lo tanto, el mensaje debe declararse dentro del archivo `app.component.ts`. Vamos a ver:
 1. Abra el editor de VSCode y seleccione Abrir carpeta ... en la sección Inicio de la pantalla de bienvenida.
 2. Busque la carpeta `miPrimerAppAngular` de la aplicación Angular que creamos y selecciónela.
 3. Navegue a la carpeta `src\app` desde el panel EXPLORER y seleccione el archivo `app.component.ts`.
 4. Busque la propiedad del título en la clase `AppComponent` y cambie su valor a "Proyecto final a la vista":
 5. Ejecute `ng serve` desde la ventana de Terminal



- La propiedad del título está vinculada a la plantilla del componente principal. Si abrimos el archivo `app.component.html` y vamos a la línea 344, veremos el siguiente código HTML:

```
343  
344 <span>{{ title }} app is running!</span>  
345
```

- La sintaxis `{{}}` que rodea la propiedad del título se llama interpolación. Durante la interpolación, angular lee el valor de la propiedad del componente adjunto, lo convierte en texto y lo imprime en la pantalla.
- Angular CLI proporciona una rica colección de comandos que pueden ayudarnos durante nuestra rutina de desarrollo diaria.



CONSTRUYENDO NUESTRA APLICACIÓN CON NX CONSOLE

- Angular CLI es una herramienta de línea de comandos con una variedad de comandos. Cada comando puede aceptar una amplia gama de opciones y parámetros según la tarea que queramos realizar. El proceso de recordar estos comandos y sus opciones de memoria es abrumador y requiere mucho tiempo.
- VSCode Marketplace contiene muchas extensiones útiles que podemos instalar para ayudarnos durante el desarrollo de Angular. Una de estas extensiones es Nx Console, que proporciona una interfaz de usuario a través de Angular CLI. Para instalar la consola Nx en su entorno, siga estos pasos:

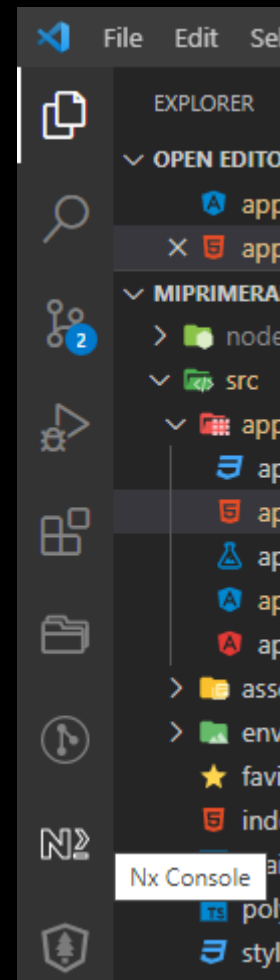
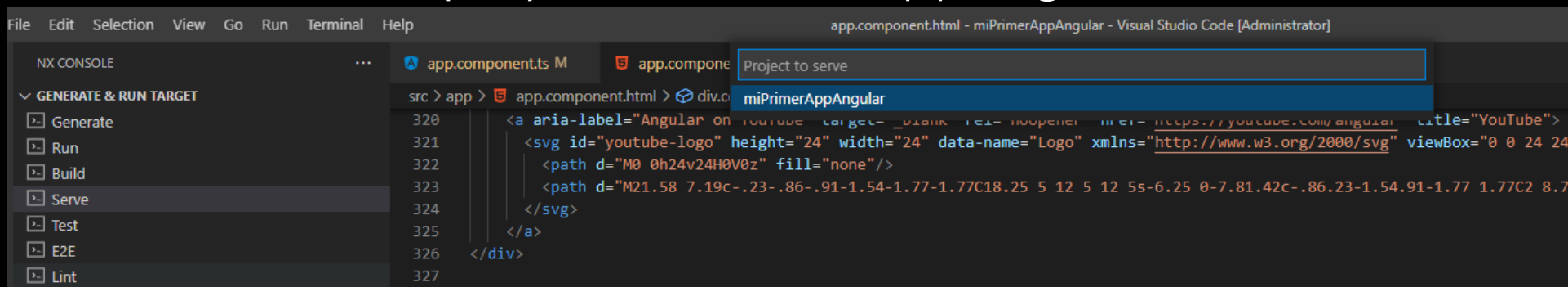
- 1.- Abra VSCode y haga clic en el menú Extensiones en la barra lateral:
- 2.- En el panel EXTENSIONES que aparece, escriba Nx Console.
3. Haga clic en el botón Instalar en el primer elemento para instalar la extensión de la consola Nx.



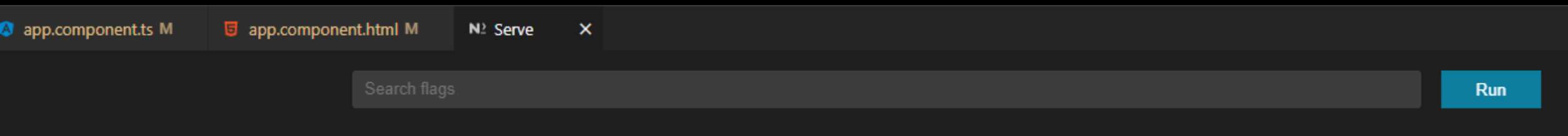
- La extensión Nx Console ahora está instalada globalmente en nuestro entorno, lo que significa que podemos usarla en cualquier proyecto Angular. Es una representación gráfica de los comandos CLI de Angular más comunes. Actualmente, admite los siguientes comandos (el comando CLI Angular relacionado se muestra entre paréntesis):
 - generate: genera nuevos elementos angular, como componentes y módulos
 - run: ejecuta un objetivo como se define en el archivo de configuración angular.json del espacio de trabajo de Angular CLI (ng run).
 - build: construye una aplicación Angular (ng build).
 - serve: cree y sirva una aplicación Angular (ng serve).
 - test: Ejecute las pruebas unitarias de una aplicación Angular (ng test).
 - E2E: Ejecute las pruebas de integración o de un extremo a otro de una aplicación Angular (ng e2e).
 - Change workspace: cargue un espacio de trabajo CLI angular diferente en VSCode


- Todo lo que podemos hacer con Angular CLI casi se puede lograr usando la Consola Nx. El beneficio real es que el desarrollador no necesita recordar todas las opciones de comando de Angular CLI, ya que todas están representadas en una interfaz gráfica. Veamos como:

1. Abra la carpeta miPrimerAppAngular usando VSCode y haga clic en el menú de la consola Nx en la barra lateral:
2. Haga clic en el comando Serve del panel NX CONSOLE
3. Seleccionar el proyecto miPrimerAppAngular



- La consola Nx abrirá la pestaña Serve, donde, opcionalmente, puede configurar opciones adicionales compatibles con el comando de servicio de Angular CLI. Cada opción contiene un título con una breve descripción que explica lo que hace.
- 4. Haga clic en el botón Run para iniciar la aplicación Angular:



- 
- Este es el mismo comando que ejecutamos cuando usamos Angular CLI desde una ventana de Terminal.
 - La consola Nx usa tareas internamente para ejecutar comandos de Angular CLI. Las tareas son un mecanismo integrado de VSCode que nos permite ejecutar scripts o iniciar procesos externos sin interactuar directamente con la línea de comandos.
 - La extensión de la consola Nx hace un trabajo fantástico al eliminar la carga de recordar los comandos de Angular CLI de memoria. VSCode Marketplace contiene muchas más extensiones para desarrolladores de Angular que complementan el trabajo de la Consola Nx

RESUMEN

- Aprendimos sobre los principios básicos del framework Angular y proporcionamos una breve descripción general de la arquitectura.
- Vimos algunas extensiones populares para VSCode que podemos usar para mejorar nuestra experiencia de desarrollo.
- Luego, aprendimos cómo usar Angular CLI, una herramienta para construir una nueva aplicación Angular desde cero. También hicimos nuestra primera interacción con el código Angular modificando el componente Angular de una aplicación Angular CLI. Finalmente, instalamos la extensión de la Consola Nx y aprendimos cómo construir nuestra aplicación con ella.
- Lo siguiente es ver el enrutador y aprenderemos a usarlo para crear un blog personal usando el generador de sitios web estáticos de Scully.

TEST

- 1. ¿Cuál es el componente básico de una aplicación Angular?
- 2. ¿Cómo agrupamos componentes de funcionalidad similar?
- 3. ¿Quién es responsable de manejar las tareas de lógica de negocios en una aplicación Angular?
- 4. ¿Qué comando Angular CLI podemos usar para crear una nueva aplicación Angular?
- 5. ¿Qué comando Angular CLI podemos usar para ejecutar una aplicación Angular?
- 6. ¿Cómo declaramos un componente angular en HTML?
- 7. ¿Cómo declaramos componentes angular en un módulo?
- 8. ¿Qué sintaxis utilizamos para el enlace de texto en plantillas HTML?
- 9. ¿Cuál es el beneficio de usar la consola Nx?
- 10. ¿Qué extensión usamos para realizar análisis estático en nuestro código Angular?

LECTURAS ADICIONALES

- Introducción a los conceptos básicos: <https://angular.io/guide/architecture>
- Interpolación: <https://angular.io/guide/interpolation>
- Nx Console: <https://nx.dev/latest/angular/cli/console>
- Angular Essentials: <https://marketplace.visualstudio.com/items?itemName=johnpapa.angular-essentials>
- Angular Evergreen: <https://expertlysimple.io/get-evergreen>