1. [UNIDAD 3: Introducción](https://learn.nextu.com/mod/page/view.php?id=3601&pid=P_WEBDEV_V2)

**Bienvenido a Herramientas, Build Systems y tecnologías emergentes**

En esta Unidad aprenderás qué es Angular 2, un framework JavaScript de código abierto promovido por Google y la comunidad de desarrollo libre. Enfocado en la filosofía mobile-first, es uno de los frameworks para el desarrollo de SPAs con mayor renombre en el mundo.

**Puntos de aprendizaje**

**Unidad 3:  Angular 2**

**Los objetivos del Programa que corresponden a esta Unidad son:**

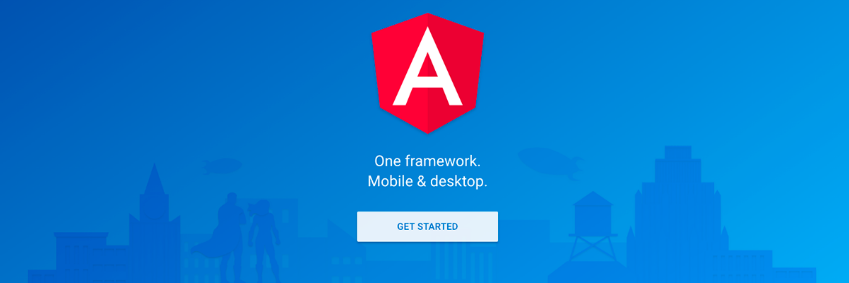
* Crear ambientes de programación y a gestionar recursos.
* Implementar la construcción de sitios web a partir de componentes.

**Los objetivos del Curso 8 que corresponden a esta Unidad son:**

* Identificar y apropiar tendencias web.
* Aplicar el concepto ambiente web.
* Implementar web components.

1. Lección 1: ¿Qué es Angular y cómo funciona?
   1. [Lectura: ¿Qué es Angular 2?](https://learn.nextu.com/mod/lesson/view.php?id=3602&pid=P_WEBDEV_V2)

¿Qué es Angular 2?

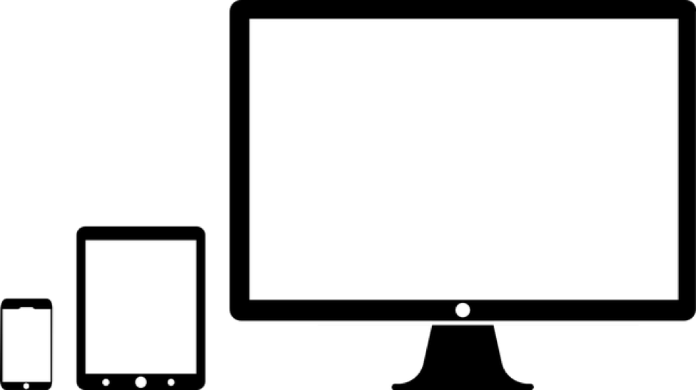


Angular 2 es un framework JavaScript de código abierto promovido por Google y la comunidad de desarrollo libre. Enfocado en la filosofía mobile-first, es uno de los frameworks para el desarrollo de SPAs con mayor renombre en el mundo.   
Fue lanzado en su versión estable en Mayo de 2016 y desde entonces, su comunidad de desarrolladores ha crecido exponencialmente.

Ventajas de Angular 2:

A continuación, se presentan algunas de las principales ventajas y características de Angular 2:

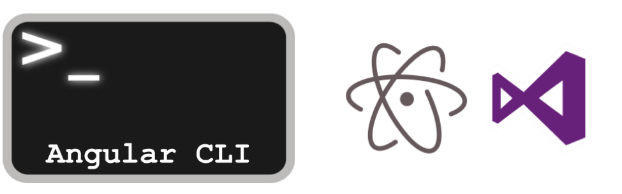
* **Multiplataforma:** con Angular 2 es posible desarrollar páginas Web tipo SPA de alto rendimiento. Sin embargo, es posible, por medio de la integración con otras herramientas y frameworks, crear aplicaciones móviles nativas o híbridas, así como aplicaciones de escritorio en múltiples sistemas operativos.



* **Rapidez y desempeño:** mediante la generación de plantillas y el uso de componentes para la reutilización de código, la rapidez en la que se desarrollan las aplicaciones es mucho más alta que el desarrollo Web tradicional. Además, gracias al enrutador de componentes (Component Router), los usuarios sólo cargan el código necesario para la vista que desean acceder en ese instante, mejorando notoriamente el rendimiento de la aplicación y contribuyendo a una experiencia de usuarios fluida y veloz.



* **Productividad:** Angular CLI como herramienta de ayuda desde la línea de comandos, permite crear proyectos, elementos de la aplicación y mucho más, con gran facilidad y rapidez. Además, existe una gran variedad de IDEs con funcionalidades para autocompletar código, detección de errores, entre muchas más.



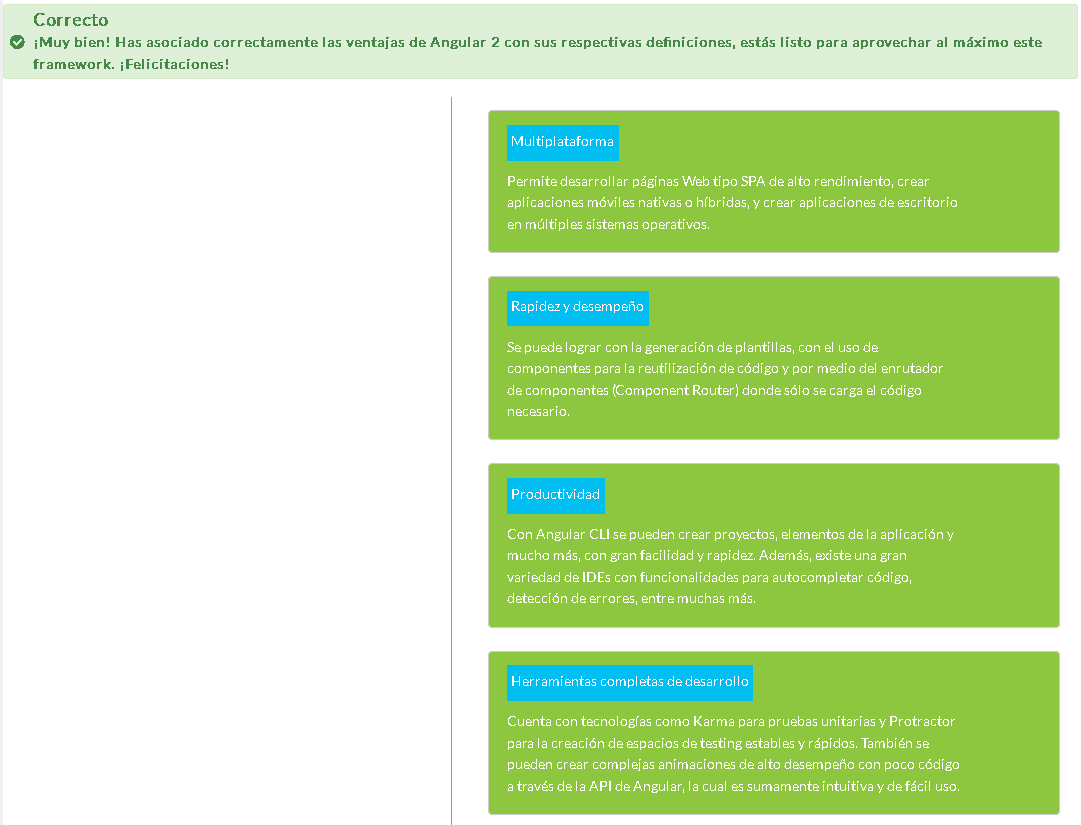
* **Herramientas completas de desarrollo:** Angular 2 incluye tecnologías como Karma para pruebas unitarias y Protractor para la creación de espacios de testing estables y rápidos. Además, es posible crear complejas animaciones de alto desempeño con poco código a través de la API de Angular, la cual es sumamente intuitiva y de fácil uso.



* 1. [Actividad 1](https://learn.nextu.com/mod/lesson/view.php?id=3602&pageid=14478&pid=P_WEBDEV_V2)

Tiempo de pensar

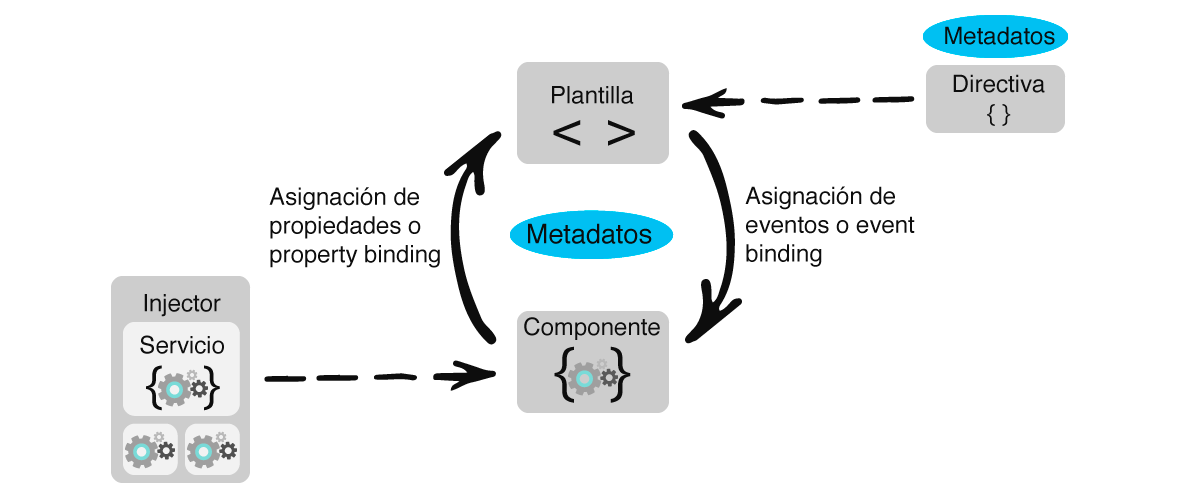
Ahora que sabes qué es Angular 2 y cuáles son sus principales ventajas y características, te invitamos a que realices la siguiente actividad que te ayudará a reforzar tu conocimiento.  
  
Instrucciones:  
Lee el enunciado con atención y realiza la asociación que consideres correcta.  
  
Enunciado:  
Angular 2 es un framework JavaScript de código abierto promovido por Google y la comunidad de desarrollo libre, éste cuenta con unas ventajas principales que tú ya conoces. A continuación, aparecerán listadas y al frente verás las definiciones, las cuales deberás asociar según corresponda. ¡Adelante!



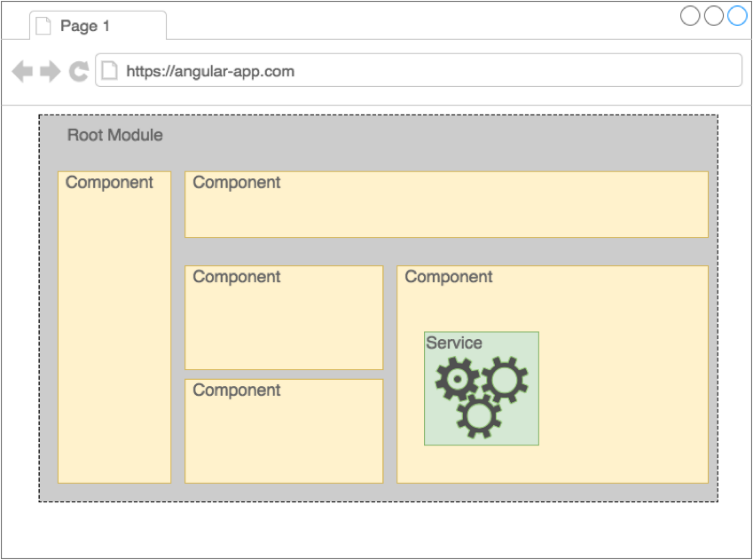
* 1. Lectura: Arquitectura

Arquitectura

Angular 2 es un framework que se encarga de crear aplicaciones del lado del cliente, usando diferentes elementos propios de su estructura para obtener un resultado óptimo en cuanto a eficiencia, desempeño y usabilidad.   
  
La arquitectura de este framework se resume en el siguiente gráfico:



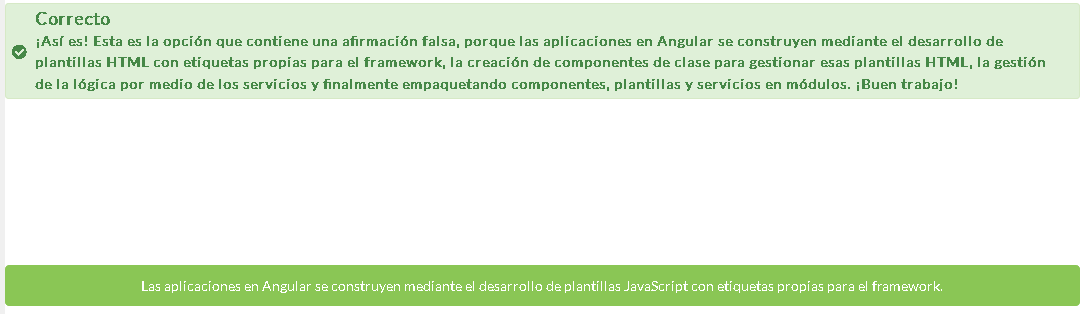
Cada uno de los elementos que hacen parte de la arquitectura de Angular 2 serán estudiados con mayor profundidad más adelante, sin embargo es importante anotar que este framework se considera como modular, es decir, que utiliza módulos reutilizables para la construcción de la aplicación.  
  
Los módulos están integrados por componentes que contienen la vista a través de las plantillas y las directivas, la lógica a través de los servicios y las asignaciones, tanto de eventos, como propiedades para las interacciones dinámicas entre las vistas y la lógica.  
  
Las aplicaciones en Angular se construyen mediante el desarrollo de plantillas HTML con etiquetas propias para el framework, la creación de componentes de clase para gestionar esas plantillas HTML, la gestión de la lógica por medio de los servicios y, finalmente, empaquetando componentes, plantillas y servicios en módulos.   
  
Al lanzar la aplicación, se realiza un empaquetado general de todos los elementos y se muestra el módulo raíz, o root module, de donde parte toda la secuencia de la aplicación.



* 1. Actividad 2

Tiempo de pensar

Ahora que conoces la arquitectura de Angular 2, te invitamos a poner a prueba tus conocimientos con la siguiente actividad.  
  
Instrucciones:  
Lee el enunciado con atención y elige la opción que consideres correcta.  
  
Enunciado:  
A continuación encontrarás varias afirmaciones referentes a la arquitectura de Angular 2 que has visto hasta ahora. Tu misión es encontrar la opción que contiene información falsa sobre el tema. ¡Adelante!



* 1. Configuración del ambiente

Angular CLI 🡪 Interfaz de línea de comandos asistente y sirve para:

* Crear un nuevo proyecto
* Crear elementos del framework
* Crear bloques de código
* Correr una aplicación

**Requerimientos:**

* Node versión 4 o superior
* NPM versión 3 o superior

C:\Users\jrodrigue253>node -v

v12.19.0

C:\Users\jrodrigue253>npm -v

6.14.8

C:\Users\jrodrigue253>

**Se debe proceder con la instalación de angular-cli**

C:\Users\jrodrigue253>npm install -g angular-cli

**Se crea un nuevo proyecto**

C:\...\ejercicio1-Ang2>***ng new primer-proyecto***

? Would you like to add Angular routing? No

? Which stylesheet format would you like to use? CSS

CREATE primer-proyecto/angular.json (3638 bytes)

CREATE primer-proyecto/package.json (1291 bytes)

CREATE primer-proyecto/README.md (1031 bytes)

CREATE primer-proyecto/tsconfig.json (489 bytes)

CREATE primer-proyecto/tslint.json (3125 bytes)

CREATE primer-proyecto/.editorconfig (274 bytes)

CREATE primer-proyecto/.gitignore (631 bytes)

CREATE primer-proyecto/browserslist (429 bytes)

CREATE primer-proyecto/karma.conf.js (1027 bytes)

CREATE primer-proyecto/tsconfig.app.json (210 bytes)

CREATE primer-proyecto/tsconfig.spec.json (270 bytes)

CREATE primer-proyecto/src/favicon.ico (948 bytes)

CREATE primer-proyecto/src/index.html (300 bytes)

CREATE primer-proyecto/src/main.ts (372 bytes)

CREATE primer-proyecto/src/polyfills.ts (2835 bytes)

CREATE primer-proyecto/src/styles.css (80 bytes)

CREATE primer-proyecto/src/test.ts (753 bytes)

CREATE primer-proyecto/src/assets/.gitkeep (0 bytes)

CREATE primer-proyecto/src/environments/environment.prod.ts (51 bytes)

CREATE primer-proyecto/src/environments/environment.ts (662 bytes)

CREATE primer-proyecto/src/app/app.module.ts (314 bytes)

CREATE primer-proyecto/src/app/app.component.html (25725 bytes)

CREATE primer-proyecto/src/app/app.component.spec.ts (969 bytes)

CREATE primer-proyecto/src/app/app.component.ts (219 bytes)

CREATE primer-proyecto/src/app/app.component.css (0 bytes)

CREATE primer-proyecto/e2e/protractor.conf.js (808 bytes)

CREATE primer-proyecto/e2e/tsconfig.json (214 bytes)

CREATE primer-proyecto/e2e/src/app.e2e-spec.ts (648 bytes)

CREATE primer-proyecto/e2e/src/app.po.ts (301 bytes)

√ Packages installed successfully.

Directory is already under version control. Skipping initialization of git.

**Se corre el proyecto con el comando:**

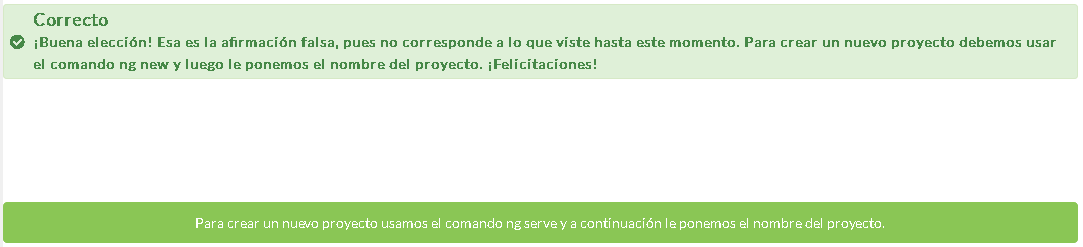
ng serve 🡪 Con este comando se inicia el nuevo proyecto creado, además la app web queda corriendo en el puerto por defecto 4200

* 1. Actividad 3

Tiempo de pensar

En esta oportunidad pudimos configurar el ambiente de Angular 2, un framework para la creación de aplicaciones Web para comenzar a realizar nuestras primeras aplicaciones. Ahora queremos invitarte a validar tus conocimientos a través de la siguiente actividad.  
  
Instrucciones:  
Lee el enunciado con atención y elige la opción que consideres correcta.

Enunciado:  
A continuación encontrarás varias afirmaciones referentes a la configuración del ambiente de Angular 2. Tu misión es encontrar la opción que contiene información falsa sobre el tema. ¡Adelante!



* 1. Lectura: Introducción a TypeScript

Introducción a TypeScript



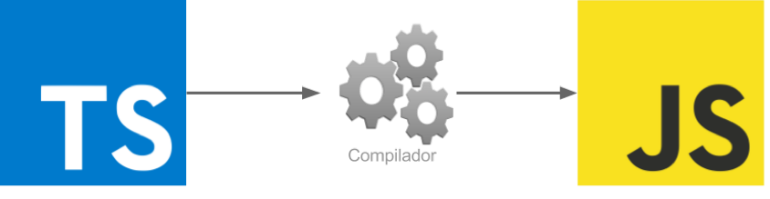
A medida que empieces a conocer un poco más sobre Angular 2, e iniciemos la codificación de algunos elementos, te darás cuenta que existen algunos elementos en el lenguaje diferentes a lo que habíamos trabajado en JavaScript. La explicación para esto es que el lenguaje que usamos para codificar aplicaciones en Angular 2 técnicamente no es JavaScript sino TypeScript. ¿Sabes qué es?

**¿Qué es TypeScript?**

TypeScript es un lenguaje de programación creado por Microsoft y de código abierto, que comprende un set de características adicionales a JavaScript para la creación de scripts Web. En resumen, podríamos decir que es un complemento de JavaScript que añade unas cuantas herramientas adicionales de gran uso para los desarrolladores.

**Compilación**

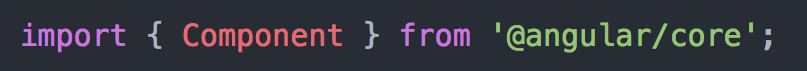
Tal vez te haya surgido la siguiente pregunta: si el único lenguaje que es interpretado por todos los navegadores es JavaScript, ¿cómo hace TypeScript para funcionar? La respuesta está en la misma pregunta, es decir, los archivos TypeScript se compilan a JavaScript plano para después ser interpretados por los navegadores. El compilador de TypeScript convierte el código propio de este lenguaje en un equivalente de JavaScript ECMAScript 3 o superior, esto quiere decir que todo código JavaScript funciona correctamente en un archivo TypeScript.



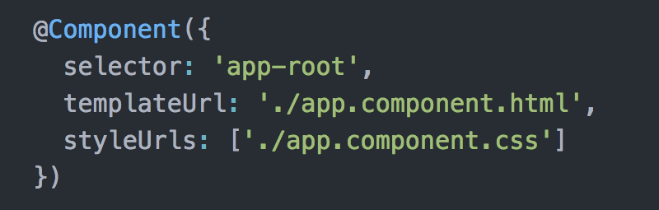
**Diferencias entre TypeScript y JavaScript plano**

Como ya lo mencionamos, TypeScript corresponde a un set adicional de herramientas para JavaScript. Dentro de las principales novedades que se encuentran en este lenguaje están:

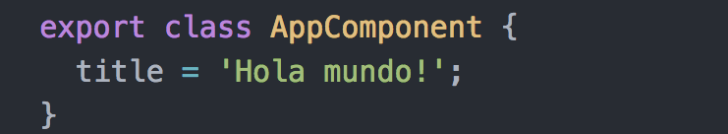
* Imports: la manera de importar código desde otro módulo es muy sencilla, haciendo que la modularidad en los proyectos sea fluida y eficiente.



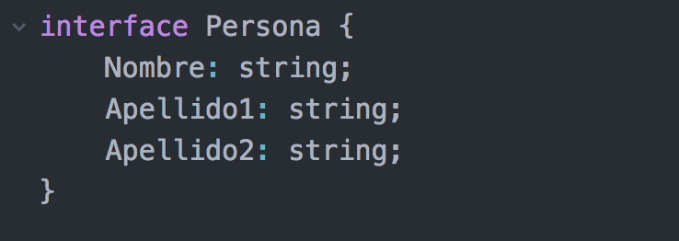
* Decoradores: los decoradores son funciones adjuntas a un objeto que realizan acciones en segundo plano y modifican el objeto al que están unidos para un fin específico.



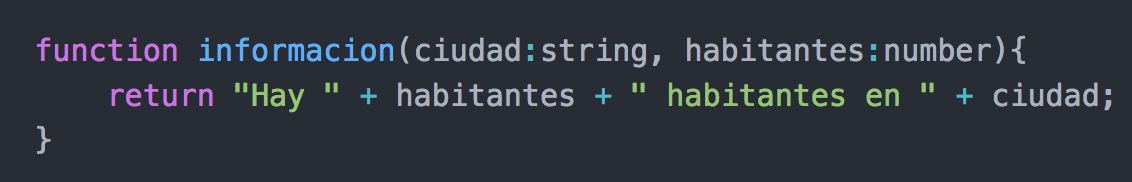
* Clases: la forma en que TypeScript define las clases permite que la programación orientada a objetos sea muy sencilla y correcta.



* Interfaces: elemento que no existe en JavaScript, sirve para forzar la implementación de atributos y métodos en una clase determinada.



* Tipos: como bien sabes, en JavaScript no se definen los tipos de datos para las variables creadas, y por lo tanto, éstas pueden tener cualquier tipo de valor. TypeScript añade un nivel adicional de seguridad en el código mediante la asignación de tipos a las variables o atributos de un objeto. De allí viene su nombre, ya que Type en inglés es Tipo.



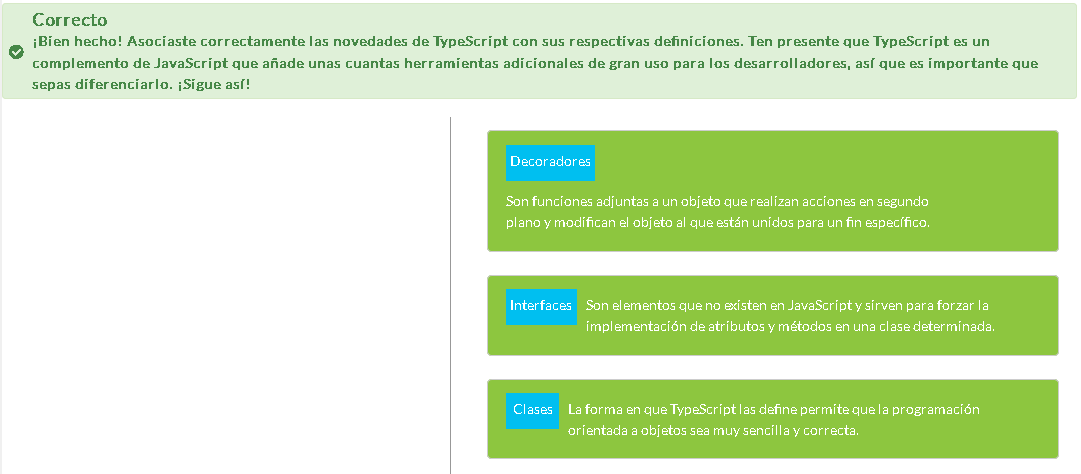
¿Por qué usar TypeScript?

Si el código TyeScript es compilado en JavaScript plano, ¿por qué no hacerlo todo en JavaScript desde el principio? En realidad hay muchas razones por las cuales se usa TypeScript en el desarrollo de aplicaciones con Angular 2:

* 1. Lenguaje fuertemente tipado: TypeScript añade tipos de datos a variables y a atributos de objetos, esto aumenta el nivel de seguridad en el código evitando que valores inapropiados sean asignados y permitiendo fácilmente la detección de errores en la compilación, o que incluso el mismo IDE pueda identificar errores y alertas que te permitan, como desarrollador, detectar estas situaciones con gran facilidad.
  2. Características de nueva generación: las herramientas de TypeScript mejoran en gran medida algunos elementos de JavaScript, lo que aumenta la calidad del código escrito y las posibilidades que se pueden alcanzar con él.
  3. Nuevas características: TypeScript, además de mejorar lo existente en JavaScript, también añade nuevos elementos como las interfaces y los genéricos, ampliando la gama de herramientas para que el desarrollador cree scripts de alto rendimiento y funcionalidad.
  4. Lenguaje principal de Angular 2: los creadores de Angular 2 eligieron a TypeScript como el lenguaje principal del framework, lo que significa que crear aplicaciones en Angular 2 usando TypeScript produce una fluidez y compatibilidad óptimas en la codificación y la compilación.
  5. Documentación y soporte: al ser TypeScript el lenguaje principal para Angular 2, la gran mayoría de documentación y soporte que encontrarás por parte de la comunidad de desarrolladores, estará en dicho lenguaje; permitiendo encontrar ayuda de los expertos con gran facilidad.
  6. Actividad 4

Tiempo de pensar

Ahora que sabes qué es Angular 2, cuáles son las diferencias que tiene frente a JavaScript y por qué usarlo, es hora de afianzar tus conocimientos por medio de la siguiente actividad.  
  
Instrucciones:  
Lee el enunciado con atención y realiza la asociación que consideres correcta.  
  
Enunciado:  
Como ya sabes, TypeScript corresponde a un set adicional de herramientas para JavaScript, algunas de las principales novedades se encuentran a continuación y tu misión será asociar las definiciones a cada una de ellas. ¡Adelante!



* 1. Laboratorio

LABORATORIO DE  
Instalación y Configuración

Durante el desarrollo de una aplicación en Angular 2, puedes encontrar que hay elementos que siempre usarás y que cuentan con una estructura básica similar.   
Sin embargo, la estructuración de dichos elementos y su creación, puede tornarse dispendiosa o aburrida después de un tiempo.   
Adicionalmente, la configuración inicial de un proyecto para crear una estructura de carpetas, instalar las dependencias y programas necesarios, y hasta correr la aplicación, son labores que para el desarrollador pueden tomar un buen tiempo.   
Gracias a Angular CLI, todo esto deja de ser un problema para ti como desarrollador, ya que se encarga de realizar todas estas tareas mencionadas a partir de sencillos comandos en una terminal.

Contenido:

Para hacer uso del asistente AngularCLI debemos primero instalarlo y, para hacerlo, es importante que dispongamos la máquina que vayamos a usar para que sea apta al programa a instalar. ¿Estás preparado?

Instrucciones

* 1. Verifica si tienes Node instalado.  
     a. Ejecuta en la terminal el siguiente comando.

user$ node -v

b. Si el resultado del comando anterior es un mensaje de error indicando que no existe dicha sentencia o no se reconoce, quiere decir que no está instalado node en la máquina.  
En este caso debes dirigirte a la página oficial de Node.js y descargar el archivo de instalación, haciendo click en LTS para la última versión estable. Posteriormente debes seguir el proceso de instalación.

https://nodejs.org/es/



c. Si el resultado del comando en a. es un mensaje que indica la versión de node y ésta inicia con un número menor a 4, debes realizar los mismos pasos en b. para instalar una versión más reciente.  
  
d. Si el resultado del comando en a. es un mensaje que indica la versión de node y ésta inicia con un número mayor o igual a 4, cuentas con la versión correcta de node.

* 1. Verifica que tienes instalado npm con la versión correcta  
     a. Ejecuta en la terminal el siguiente comando.

user$ npm -v

b. Si el resultado del comando anterior es un mensaje de error indicando que no existe dicha sentencia o no se reconoce, quiere decir que no está instalado npm en la máquina. Para instalarlo debes ejecutar el siguiente comando:

user$ npm install -g

c. Si el resultado del comando en a. es un mensaje que indica la versión de node y ésta inicia con un número menor a 3, debes ejecutar el siguiente comando:

user$ npm install npm@latest -g

d. Si el resultado del comando en a. es un mensaje que indica la versión de node y ésta inicia con un número mayor o igual a 3, cuentas con la versión correcta de npm.

* 1. Instala el asistente Angular CLI  
     a. Ejecuta en la terminal el siguiente comando

user$ npm install -g angular-cli  
b. Espera unos cuantos minutos, puesto que la instalación de AngularCLI toma algún tiempo.  
c. Si durante la instalación observas un error que se presente, lee bien la descripción del mismo. Generalmente los errores que se presentan son relacionados con versiones incorrectas de node o npm.  
d. Al finalizar la instalación de manera exitosa se mostrará un mensaje y se muestra nuevamente el indicador de un nuevo comando en la terminal.

* 1. Comprueba la correcta instalación de AngularCLI creando un nuevo proyecto  
     a. El nuevo proyecto llevará el nombre “tareas” e indicamos un prefijo para todos los componentes generados a través de AngularCLI. Este prefijo se usa para evitar duplicidad en los selectores únicos de componentes para que no coincidan con elementos propios de HTML. Ejecuta el siguiente comando:

user$ ng new tareas --prefix t

Buenas Prácticas

Cada vez que realices actualizaciones o instalaciones a través de npm, es importante no hacerlo desde un super usuario (sudo) ya que pueden presentarse problemas que comprometan la seguridad del sistema en general. Si se presentan errores de instalación lee con atención las causas de los mismos para dar una solución adecuada.

1. Lección 2: Componentes y Directivas

[3.1. Componentes](https://learn.nextu.com/mod/lesson/view.php?id=3603&pid=P_WEBDEV_V2)

Componentes – una pieza de un rompecabezas que puede ser utilizado cuantas veces sean necesarias

Cada componente debe tener:

* Vista
* Controlador

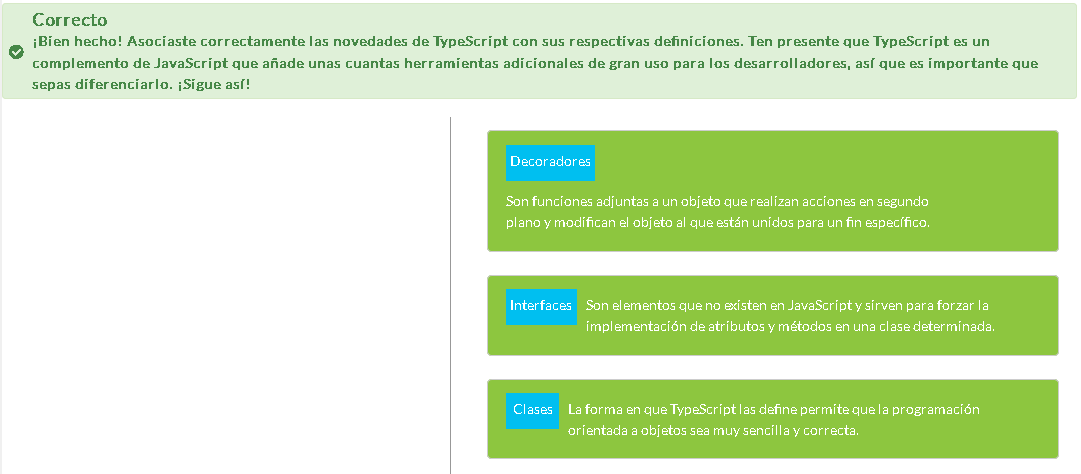
.ts 🡪 typescript

.spec.ts 🡪 Función son usados para labores de testing

[3.2. Actividad 1](https://learn.nextu.com/mod/lesson/view.php?id=3603&pageid=14488&pid=P_WEBDEV_V2)

Tiempo de pensar

Ahora que conoces cada una de las herramientas de Angular para la creación de aplicaciones y sus componentes, te invitamos a realizar la siguiente actividad para que afiances tus conocimientos.  
  
Instrucciones:  
Lee el enunciado con atención y realiza la asociación que consideres correcta.  
  
Enunciado:  
Los decoradores en los componentes sirven para ejecutar acciones en segundo plano que le indican a Angular que a lo que están unidos es un componente, y así poder usarlos. A continuación, aparecerán listados los elementos del decorador, al frente verás las definiciones, las cuales deberás asociar según corresponda. ¡Adelante!



3.3. Creación de componentes

Ejemplos:

Angular 2

Ng g component barra-superior

Angular 9

C:\Users\jrodrigue253\Github\NextUExercises\DesarrolloWeb\Modulo\_8-ToolsBuildSystems\Unidad\_3-Ang2\ejercicio1-Ang2\primer-proyecto>ng generate component barra-superior

CREATE src/app/barra-superior/barra-superior.component.html (29 bytes)

CREATE src/app/barra-superior/barra-superior.component.spec.ts (678 bytes)

CREATE src/app/barra-superior/barra-superior.component.ts (306 bytes)

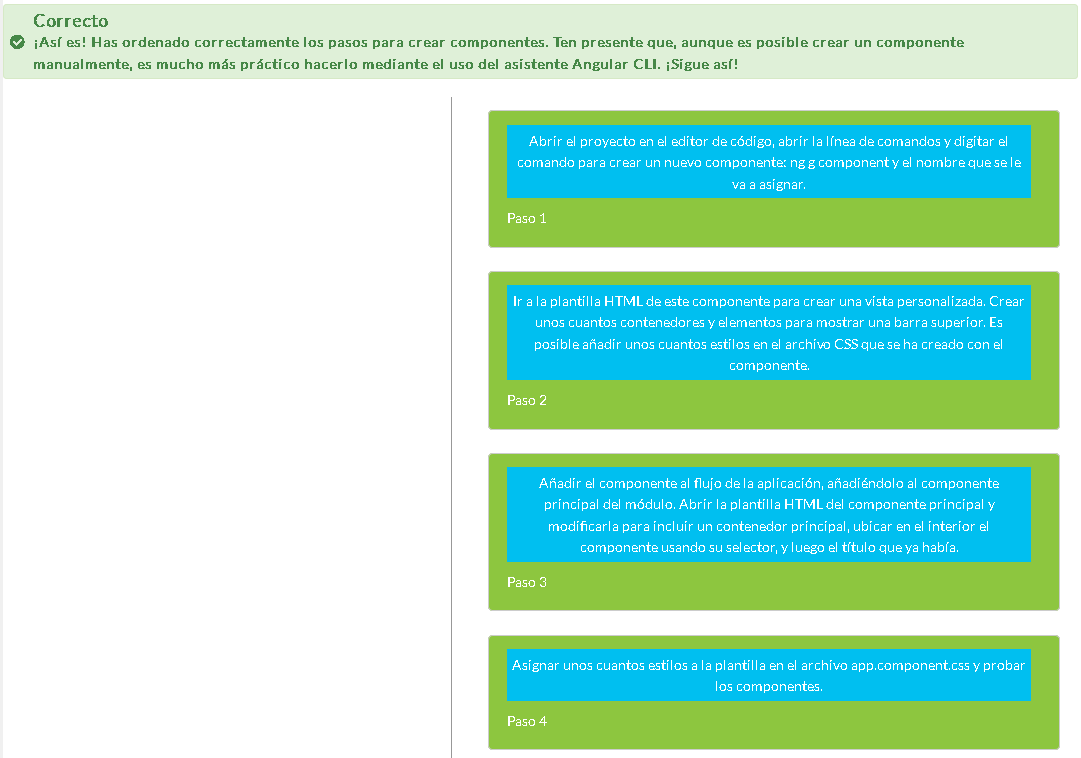
CREATE src/app/barra-superior/barra-superior.component.css (0 bytes)

UPDATE src/app/app.module.ts (426 bytes)

3.4. Actividad 2

Tiempo de pensar

Ya que exploraste cómo crear tus propios componentes y verlos en acción, te invitamos a que pongas a prueba tus conocimientos con esta actividad.  
  
Instrucciones:  
Lee el enunciado con atención y realiza la asociación que consideres correcta.  
  
Enunciado:  
Como pudiste ver, un componente corresponde a un conjunto de archivos y secciones determinadas, y aunque es posible crear uno manualmente, es mucho más práctico hacer uso del asistente Angular CLI para esta tarea. A continuación aparecerán listados los pasos para la creación de componentes, tu misión será ordenarlos de acuerdo a lo que has visto hasta ahora. ¡Adelante!



3.5. Lectura: Databinding

3.6. Actividad 3

3.7. Directivas

3.8. Actividad 4

3.9 Ejercicio

1. Lección 3: Pipes, Servicios y Routing
2. Lección 4: Comunicación con el Back-end
3. Lección 5: Formularios, Animaciones y Deployment
4. UNIDAD 3: Prueba