

Lección 1

Creando mi primer proyecto en Angular 8 y PrimefacesNG

Objetivos:

- 1.- Aprender a crear un proyecto angular.
- 2.- Aprender a configurar PrimefacesNG dentro de un proyecto angular.
- 3.- Aprender a crear un componente en angular
- 4.- Crear una ruta en angular.
- 5.- Aprender a configurar los estilos de primefaces.



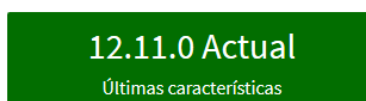
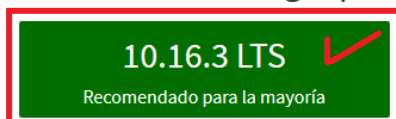
Paso 1

Instale Node 10 en su computadora, baje node e instálelo.



Node.js® es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome.

Descargar para Windows (x64)



[Otras Descargas](#) | [Cambios](#) | [Documentación del API](#) [Otras Descargas](#) | [Cambios](#) | [Documentación del API](#)

Ó revise la [Agenda de LTS](#).

Paso 2

Instale Angular desde la ventana comando de windows

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

Paso 3

Compruebe la instalación de angular desde la ventana comando de windows

`ng --version`

```
C:\PrimeNG>ng --version

Angular CLI
Angular CLI: 8.3.6
Node: 10.16.3
OS: win32 x64
Angular:
...

Package                                  Version
-----
@angular-devkit/architect               0.803.6
@angular-devkit/core                    8.3.6
@angular-devkit/schematics              8.3.6
@schematics/angular                    8.3.6
@schematics/update                      0.803.6
rxjs                                    6.4.0

C:\PrimeNG>
```

Paso 4

Cree un directorio con el nombre su preferencia, en nuestro usaremos

`<C:\PrimeNG>`

Desde la ventana de comando de Windows, ejecute los siguientes pasos:

`C:\`

`mkdir PrimeNG`

Paso 5

Dentro del directorio PrimeNG, cree su primer proyecto angular, desde la ventana comando de windows ejecute el siguiente comando

```
ng new leccion1
```

Confirme la creación del ruteo de los componentes en angular

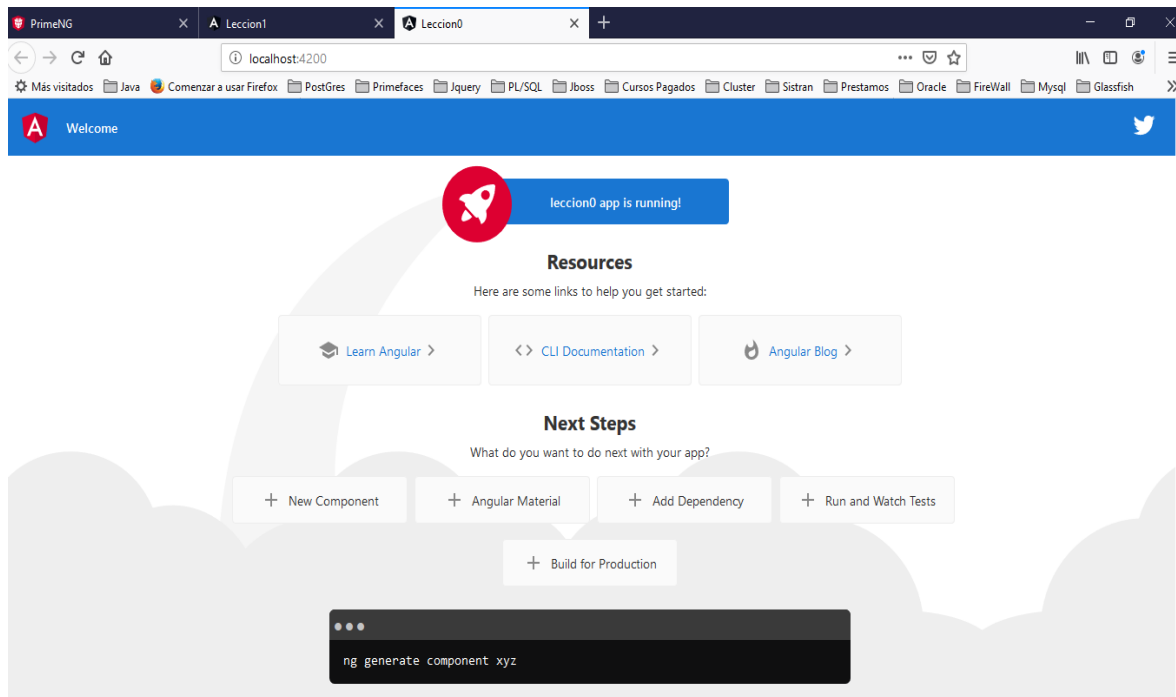
```
C:\PrimeNG>ng new leccion0  
? Would you like to add Angular routing? (y/N)
```

Confirme el tipo de archivo de estilo que usara, use CSS es más recomendado, a menos que sepa Less o Sass, cámbielo.

```
? 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 ]  
Stylus [ http://stylus-lang.com ]
```

Compruebe la creación de la app en angular, desde la ventana de comando de windows

```
ng server -o
```



Paso 6

Cámbiese al directorio leccion1

c:\PrimeNG\leccion1

Paso 7

Cree el componente home dentro del directorio leccion1

```
ng g c components/home
```

Paso 8

Dentro de directorio de lección1, instale los componentes de primefaces

```
npm install primeng -save
npm install primeicons -save
npm install @angular/animations --save
```

Paso 9

Compruebe que la aplicación de angular está corriendo correctamente.

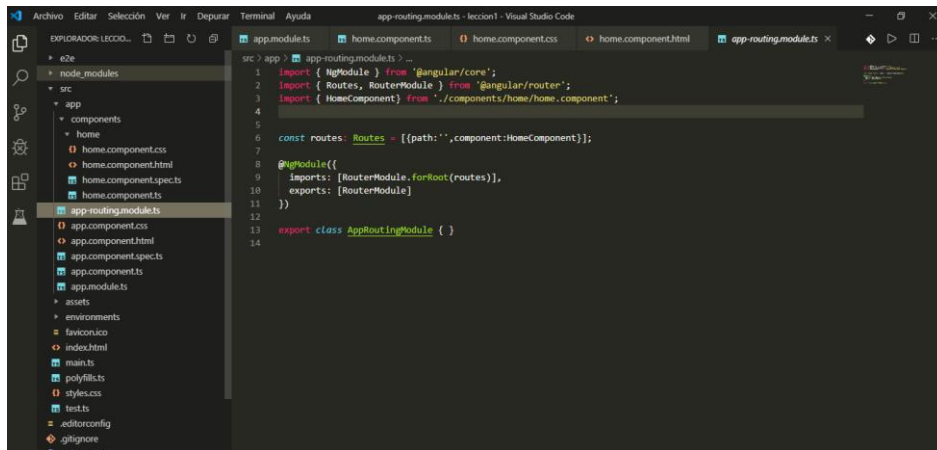
```
ng serve -o
```

Cancelar la ejecución o comandos de node

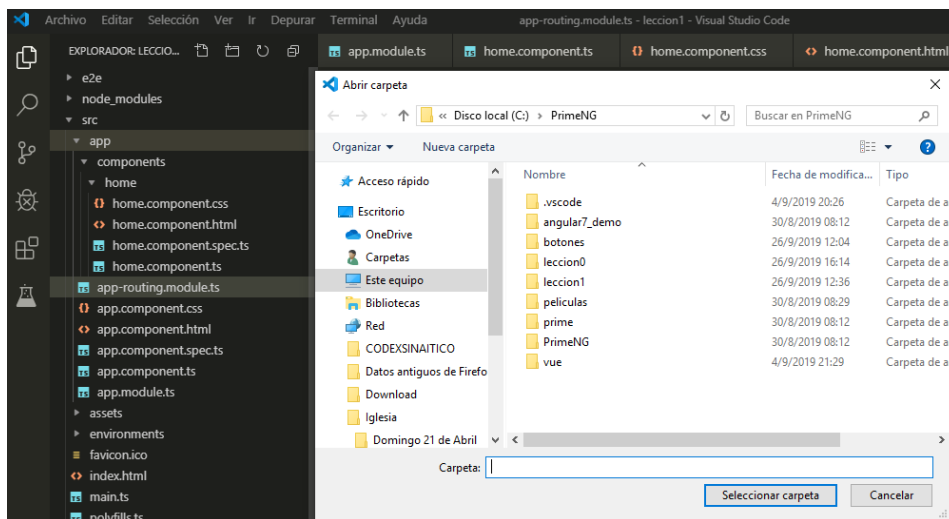
CTRL+C ➔ Cancela o aborta cualquier comando en ejecución de node o angular.

Paso 10

Abra visual Studio code y oprima CTRL+K

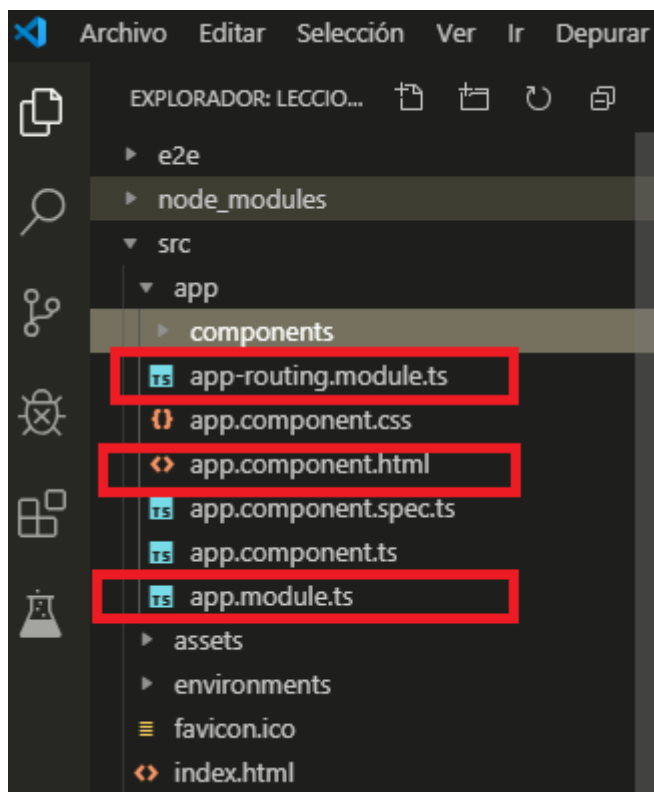


Busque el directorio leccion1 y ábralo o cárguelo



Paso 11

Vamos iniciar a configurar los archivos de inicio de una aplicación angular, vamos a trabajar con 3 archivos básicos o fundamentales:



App-routing.module.ts

(Explicación es el archivo de ruteo de lo componente que se van creando)

App.component.html

(Explicación es la pagina de inicio de la app Angular)

App.module.ts

(Es el archivo donde se realiza toda la importación de componentes o controles, si gustan familiarizarlo con algo)

Paso 12

Configure el archive <app-routing.module.ts> , vamos a configurar la ruta del componente home que creamos en el paso7.

Copie los textos en color naranja y péguelos en el archivo app-routing.module.ts

```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';
import { HomeComponent } from './components/home/home.component';

const routes: Routes = [{path: '', component: HomeComponent}];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
```

```
})  
  
export class AppRoutingModule { }
```

Paso 13

Borramos todo el contenido del archivo <app.component.html>, deberá quedar únicamente con la etiqueta router-outlet

```
<router-outlet></router-outlet>
```

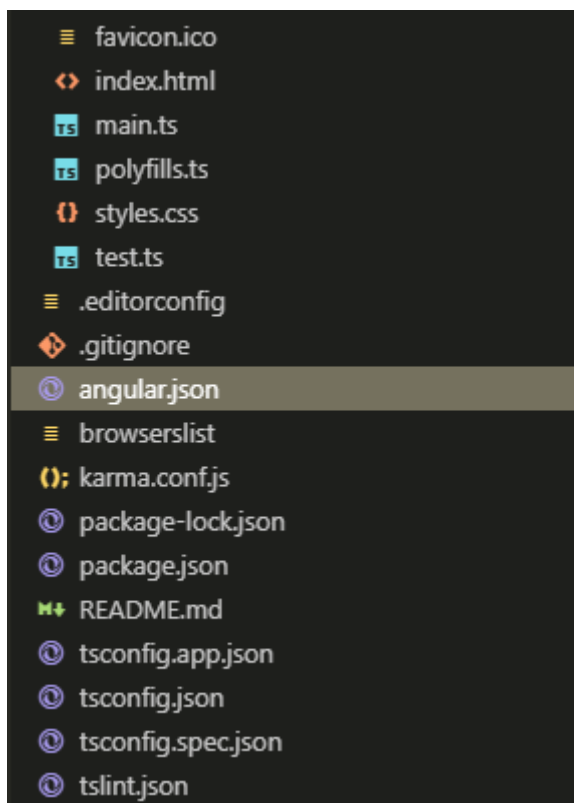
Paso 14

Agregamos los componente o controles de primefaces que vamos a utilizar, copie lo que esta marcado en color naranja y agréguelo a su lección, con excepción de <HomeController>.

```
import { BrowserModule } from '@angular/platform-browser';  
import { NgModule } from '@angular/core';  
import {BrowserAnimationsModule} from '@angular/platform-browser/animations';  
  
import { AppRoutingModule } from './app-routing.module';  
import { AppComponent } from './app.component';  
import {ButtonModule} from 'primeng/button';  
import {PanelModule} from 'primeng/panel';  
import {AccordionModule} from 'primeng/accordion';  
import { HomeComponent } from './components/home/home.component';  
  
@NgModule({  
  declarations: [  
    AppComponent,  
    HomeComponent  
  ],  
  imports: [  
    BrowserModule,  
    AppRoutingModule,  
    BrowserAnimationsModule,  
    ButtonModule,  
    PanelModule,  
    AccordionModule,  
  ],  
  providers: [],  
  bootstrap: [AppComponent]  
})  
export class AppModule { }
```

Paso 16

Configurar los estilos de primefaces



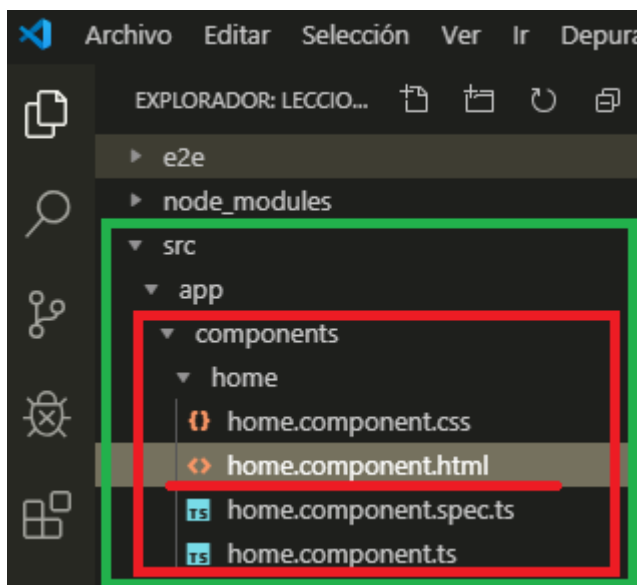
**Modificaremos el archivo
angular.json**

**Modificaremos los estilos
styles**

```
"styles": [  
    "node_modules/primeicons/primeicons.css",  
    "node_modules/primeng/resources/themes/nova-light/theme.css",  
    "node_modules/primeng/resources/primeng.min.css",  
    "src/styles.css"  
],
```

Paso 16

Modificamos el archivo [home.component.html](#) y agregamos el siguiente código:



Código de la lección 1

```
<p-panel header="Lección 1 Botones con Primefaces y Angular ">
  <p> Bienvenidos a la Leccion 1 de Primefaces y Angular 8! </p>
  <p-footer>
    <button pButton type="button" icon="pi pi-plus" label="Nuevo" class="ui-button-
info" style="margin-right: .25em"></button>
    <button pButton type="button" icon="pi pi-search" label="Ver" class="ui-button-
success"></button>
  </p-footer>
</p-panel>
<p-accordion>
  <p-accordionTab header="Componente de Botones">
    <p-button label="Click"></p-button>
    <p-button icon="pi pi-check" label="Click"></p-button>
    <p-button icon="pi pi-check" iconPos="right" label="Click"></p-button>
    <p-button icon="pi pi-check"></p-button>
    <p-button icon="pi pi-check" [disabled]="true" label="Disabled"></p-button>
  </p-accordionTab>
  <p-accordionTab header="Botones de directivas">
    <button pButton type="button" label="Click"></button>
    <button pButton type="button" icon="pi pi-check" label="Click"></button>
    <button pButton type="button" icon="pi pi-
check" iconPos="right" label="Click"></button>
    <button pButton type="button" icon="pi pi-check"></button>
    <button pButton type="button" icon="pi pi-
check" [disabled]="true" label="Disabled"></button>
  </p-accordionTab>
  <p-accordionTab header="Botones de Alerta">
```

```

        <button pButton type="button" label="Primary"></button>
        <button pButton type="button" label="Secondary" class="ui-button-secondary"></button>
        <button pButton type="button" label="Success" class="ui-button-success"></button>
        <button pButton type="button" label="Info" class="ui-button-info"></button>
        <button pButton type="button" label="Warning" class="ui-button-warning"></button>
        <button pButton type="button" label="Danger" class="ui-button-danger"></button>
    </p-accordionTab>
    <p-accordionTab header="Botones de Alerta">
        <button pButton type="button" label="Primary" class="ui-button-raised"></button>
        <button pButton type="button" label="Secondary" class="ui-button-raised ui-button-
secondary"></button>
        <button pButton type="button" label="Success" class="ui-button-raised ui-button-
success"></button>
        <button pButton type="button" label="Info" class="ui-button-raised ui-button-
info"></button>
        <button pButton type="button" label="Warning" class="ui-button-raised ui-button-
warning"></button>
        <button pButton type="button" label="Danger" class="ui-button-raised ui-button-
danger"></button>

    </p-accordionTab>
    <p-accordionTab header="Botones con esquinas redondeadas">
        <button pButton type="button" label="Primary" class="ui-button-rounded"></button>
        <button pButton type="button" label="Secondary" class="ui-button-rounded ui-button-
secondary"></button>
        <button pButton type="button" label="Success" class="ui-button-rounded ui-button-
success"></button>
        <button pButton type="button" label="Info" class="ui-button-rounded ui-button-
info"></button>
        <button pButton type="button" label="Warning" class="ui-button-rounded ui-button-
warning"></button>
        <button pButton type="button" label="Danger" class="ui-button-rounded ui-button-
danger"></button>
    </p-accordionTab>
</p-accordion>

```

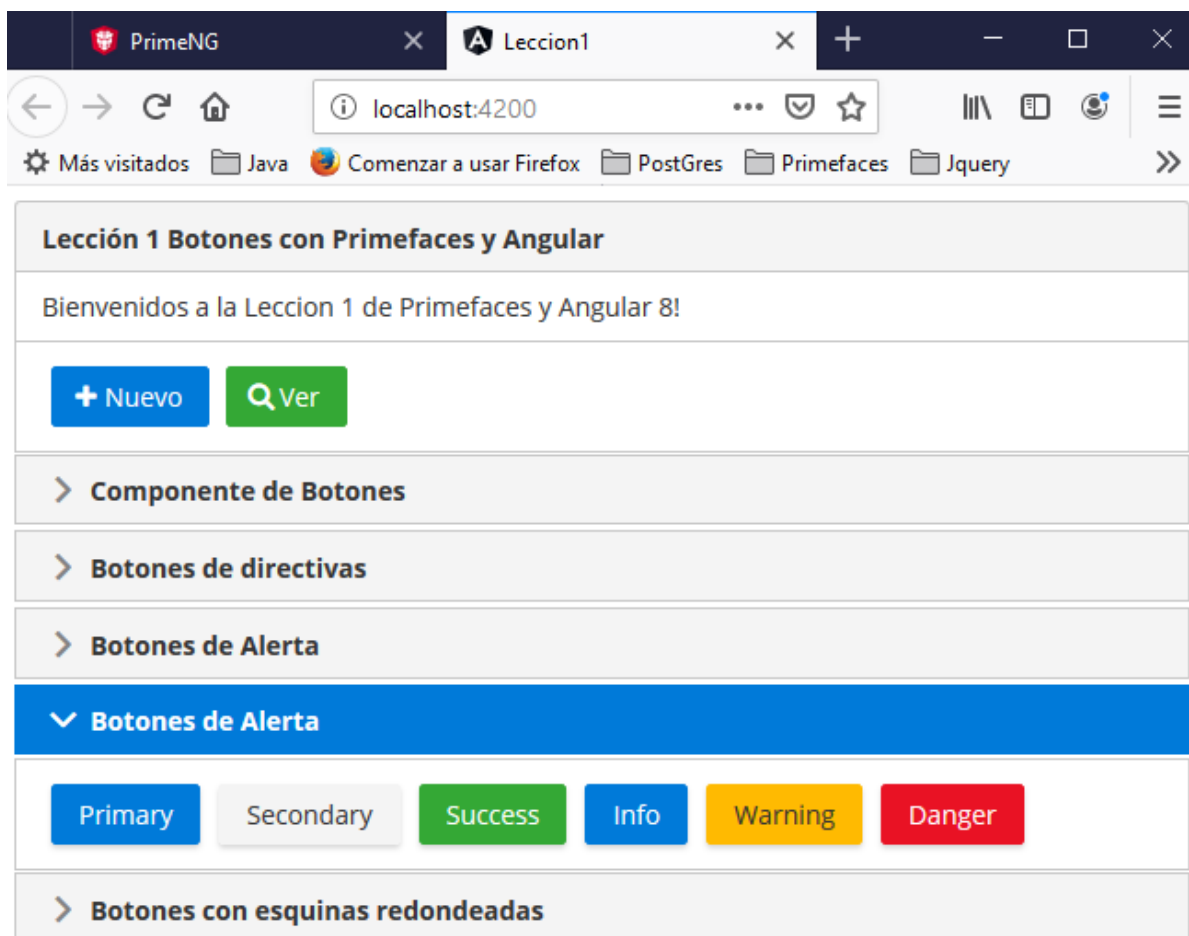
Paso 17

Comprobamos el resultado

ng serve -o

El resultado final deberá verse de la siguiente manera, todos los componentes que mostraremos y enseñamos son responsive, así que

se puede olvidar de crear 2 aplicaciones una para Android y otra para web.



Diseño responsive

El diseño web responsive o adaptativo es una **técnica de diseño web que busca la correcta visualización de una misma página en distintos dispositivos**. Desde ordenadores de escritorio a tablets y móviles.

Hoy en día accedemos a sitios web desde todo tipo de dispositivos; ordenador, tablet, smartphone... por lo que, cada vez más, nos surge la necesidad de que nuestra web se adapte a los diferentes tamaños de los mismos. Pero, ¿qué es esto exactamente?

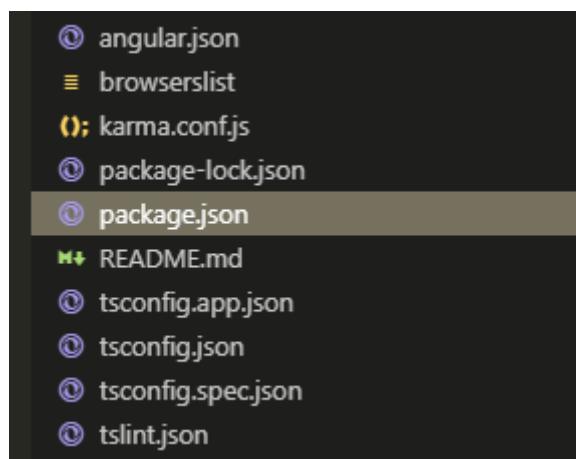
¿En qué consiste el diseño responsive?

Se trata de redimensionar y colocar los elementos de la web de forma que se adapten al ancho de cada dispositivo permitiendo una correcta visualización y una mejor experiencia de usuario. Se caracteriza porque los layouts (contenidos) e imágenes son fluidos y se usa código media-queries de CSS3.

El diseño responsive permite **reducir el tiempo de desarrollo, evita los contenidos duplicados, y aumenta la viralidad** de los contenidos ya que permite compartirlos de una forma mucho más rápida y natural.

TIPS de configuración Angular

En el archivo package.json y agregue la opción -aot -o, para que su compilación se mas rápida



```
{
  "name": "leccion1",
  "version": "0.0.0",
  "scripts": {
    "ng": "ng",
    "start": "ng serve -aot -o",
    "build": "ng build",
    "test": "ng test",
    "lint": "ng lint",
    "e2e": "ng e2e"
  },
}
```

TIPS para crear proyectos angular

```
ng new proyectonormal
```

```
ng new proyectosass --style=scss
```

```
ng new proyectobasico --minimal flag --style=scss
```

```
***** (se recomienda usar esta configuración si se esta aprendiendo) *****
```