



Reclame su perfil de buscador de empleo gratuito de DZone, llev.

**Reclama ahora ▶**

# Aplicaciones de construcción con material angular

por **Brian Fernandes** MVB · Sep. 07, 17 · Zona Web Dev

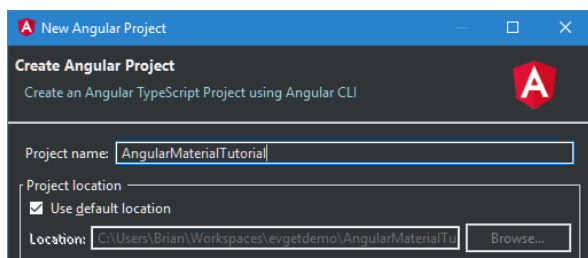
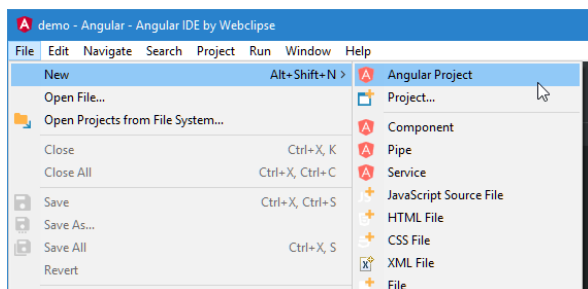
Empiece a codificar hoy para experimentar el potente motor que impulsa el desarrollo de la aplicación de datos, que se le ofrece en asociación con Qlik .

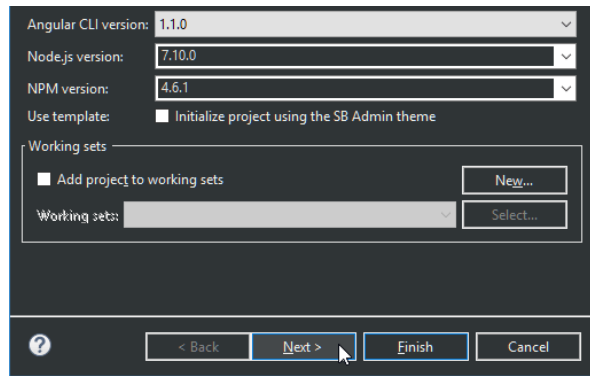
Material angular es un conjunto de componentes de interfaz de usuario de alta calidad desarrollado por el equipo de Angular, basado en la especificación de diseño de material de Google. Estos componentes nos ayudan a crear aplicaciones con una interfaz de usuario atractiva y familiar, que ofrece a los usuarios una experiencia consistente en todos los dispositivos.

En este tutorial Angular, aprenderá cómo configurar el diseño de material en su aplicación Angular. En el curso de este tutorial, vamos a utilizar Angular IDE .

Empecemos por disparar nuestro IDE Angular y crear un nuevo proyecto Angular llamado

`AngularMaterialTutorial` .



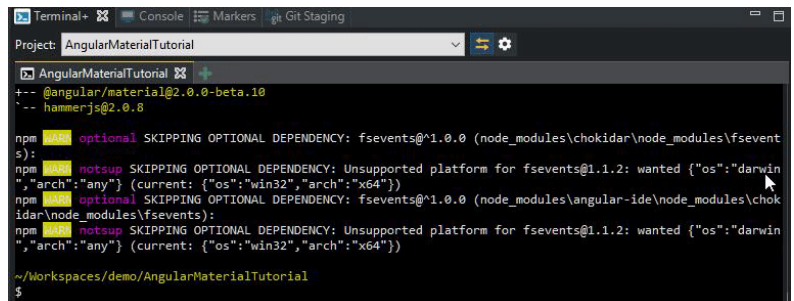


## Instalar material angular y sus dependencias

Una vez creado el proyecto, podemos instalar módulos de nodo adicionales en nuestro proyecto. Angular IDE tiene una vista de terminal desde la que podemos ejecutar comandos, por lo que en el menú de selección Window, a continuación Show view, haga clic en Terminal+. En el terminal abierto, escriba los siguientes comandos para instalar HammerJS, Angular Material y Angular CDK.

```
npm install --save hammerjs @angular / material
```

1



Instalamos HammerJS porque algunos de los componentes del Material Angular dependen de HammerJS para los gestos. Los proyectos de Modern Angular ya incluyen el módulo de animaciones Angular ( @angular/animations ), pero si usted está siguiendo una versión más antigua de la CLI o Angular, por favor instálela también.

## Uso de temas de materiales angulares

Antes de pasar a utilizar los widgets, vamos a configurar nuestros estilos.

El material incluye algunos temas pre-construidos que

usted puede encontrar a

`node_modules/@angular/material/prebuilt-themes`. Si abre esa ubicación, probablemente notará los siguientes temas:

- deeppurple-amber
- añil rosa
- rosa-azulverde
- verde púrpura

En nuestro tutorial, vamos a usar la primera opción - deeppurple-amber - añadiendo la siguiente línea a nuestro archivo de estilos global en `/src/styles.css`

```
1  @import '~@angular/material/prebuilt-themes/dee
```

También vamos a usar ligaduras de fuente, para lo cual necesitamos agregar la siguiente línea a la sección principal de nuestro `/src/index.html` archivo, para traer la fuente de los iconos de material.

```
1  < link href = "https://fonts.googleapis.com/icon
```

## Uso de componentes de material angular

En este tutorial, vamos a utilizar algunos botones de material, un *snackbar* y un menú. Comenzaremos importando una serie de estos módulos en el `app.module.ts` archivo; por ahora, puede reemplazar la fuente del módulo por el siguiente fragmento, pero recuerde que, cuando trabaja con la fuente, puede utilizar la asistencia de contenido para que la importación se realice automáticamente añadido para usted, como se puede ver a continuación.

```
TS *app.module.ts X
1  import { BrowserModule } from '@angular/platform-browser';
2  import { NgModule } from '@angular/core';
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 })
```

```

17 })
18 export class AppModule { }

```

/src/app/app.module.ts

```

import { BrowserModule } desde '@ angular / plat
1  import { NgModule } de '@ angular / core' ;
2
3  import { AppComponent } de './app.component' ;
4  import { MdButtonModule , MdButtonToggleModule ,
5  import { BrowserAnimationsModule } desde '@ angu
6
7
8  @ NgModule ({
9    declaraciones : [
10     AppComponent
11   ],
12   importaciones : [
13     BrowserModule ,
14     MdButtonModule ,
15     MdButtonToggleModule ,
16     MdSnackBarModule ,
17     MdMenuModule ,
18     MdIconModule ,
19     BrowserAnimationsModule
20   ],
21   proveedores : [],
22   bootstrap : [ AppComponent ]
23 })
24
25  clase de exportación AppModule {}

```

Reemplace la fuente del app.components.ts archivo por la fuente que se encuentra debajo, este es realmente el componente con un openSnackBar método adicional que usaremos en nuestra plantilla.

/src/app/app.component.ts

```

import { Component } de '@ angular / core' ;
1  import { MdSnackBar } de '@ angular / material'
2
3
4  @ Componente ({
5    selector : 'app-root' ,
6    templateUrl : './app.component.html' ,
7    styleUrls : [ './app.component.css' ]
8  })
9
10

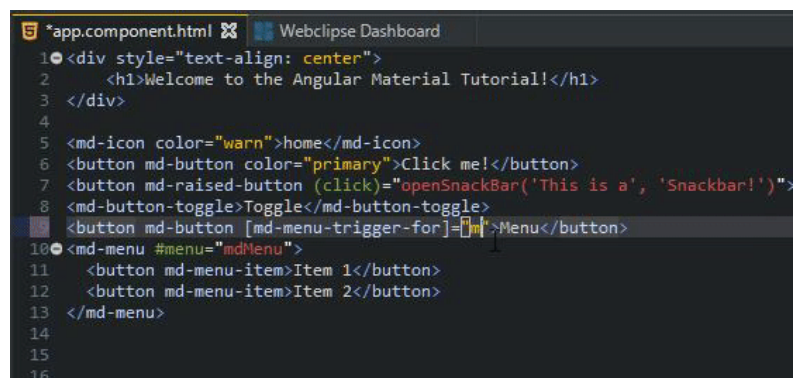
```

```

8    })
9    clase de exportación AppComponent {
10
11        constructor ( snackBar público : MdSnackBar )
12
13        openSnackBar ( mensaje : cadena , acción : cad
14
15            esto . SNACKBAR . abierto ( mensaje , acción
16
17            duración : 2000 ,
18
19        });
20    }
21
22    }

```

Finalmente, reemplace el `app.component.html` contenido del archivo con el siguiente fragmento. Una vez más, recuerde que en Angular IDE , Asistencia de contenido angular está disponible incluso dentro de plantillas HTML.



```

1 <div style="text-align: center">
2   <h1>Welcome to the Angular Material Tutorial!</h1>
3 </div>
4
5 <md-icon color="warn">home</md-icon>
6 <button md-button color="primary">Click me!</button>
7 <button md-raised-button (click)="openSnackBar('This is a', 'Snackbar!')">
8 <md-button-toggle>Toggle</md-button-toggle>
9 <button md-button [md-menu-trigger-for]="mdMenu">Menu</button>
10 <md-menu #menu="mdMenu">
11   <button md-menu-item>Item 1</button>
12   <button md-menu-item>Item 2</button>
13 </md-menu>
14
15
16

```

/src/app/app.component.html

```

1 < div style = "text-align: center" >
2   < h1 > Bienvenido al Tutorial de Material Ang
3 </ div >
4
5 < md-icon color = "warn" > inicio </ md-icon >
6 < button md-button color = "primary" > Haga cli
7 < botón md-raise-button (click) = "openSnackBar
8 < md-button-toggle > Alternar </ md-button-toggle
9 < md-menu #menu = "mdMenu" >
10   < botón md-menu-item > Elemento 1 </ button >
11   < botón md-menu-item > Elemento 2 </ button >

```

```

12 </ md-menu >
    < botón md-botón [md-menu-trigger-for] = "menú"
13

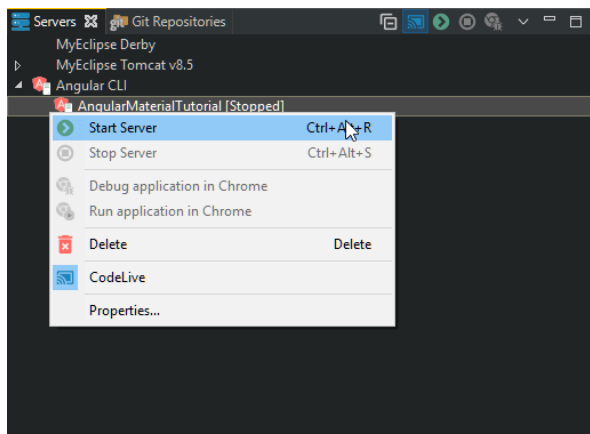
```

### Notas:

- Hemos utilizado ligaduras de fuente , que convierten "home" dentro del elemento md-icon en una imagen de inicio en lugar de mostrar el texto.
- El atributo de color nos permite establecer el color de los iconos, botones y otros widgets, especificando valores como primario, acento o advertencia. El color real utilizado variaría de un tema a otro.
- El elemento md-menu no hace nada, este menú debe ser activado desde otro widget, como lo hemos hecho usando un botón.
- A menos que se importen los módulos correspondientes a estos widgets, lo que hicimos en app.module.ts, verás errores en la plantilla, la autocompletar no funcionará y, por supuesto, los widgets mismos estarán difuntos en tu aplicación.

## Implementación de la aplicación

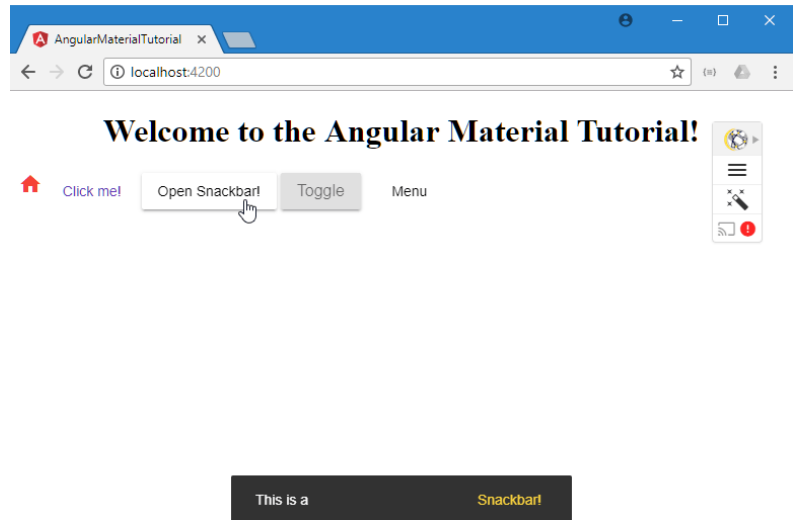
¡Bien! Es hora de ver nuestra aplicación en acción. En la vista Servidores, expanda el Angular CLI nodo y seleccione AngularMaterialTutorial . En la barra de herramientas, elija Start Server .



Utilice las acciones del menú contextual Ejecutar o depurar para abrir automáticamente la aplicación en Chrome o abrirla manualmente

http://127.0.0.1:4200 en su navegador para ver la aplicación en acción

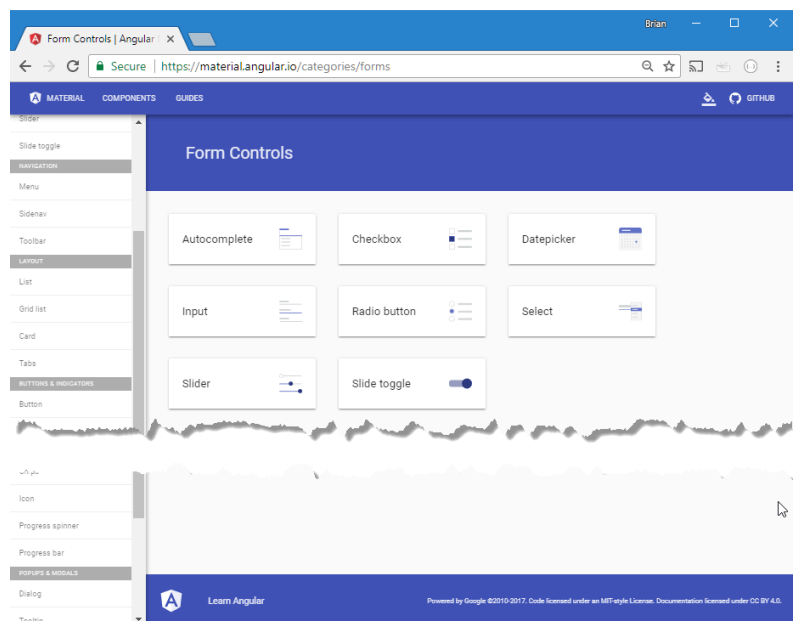
Nuestra aplicación debería tener el siguiente aspecto:



## Conclusión


El material angular está perfectamente ajustado para el rendimiento y probado completamente en navegadores y dispositivos modernos. Ya sea que esté navegando en un escritorio, portátil, tablet o teléfono, la aplicación se ve genial y proporciona a los usuarios una experiencia de navegación sólida y capaz.

Hemos explorado sólo algunos de los widgets Angular en este tutorial, pero hay docenas más - no debería tener problemas al utilizar estos pasos similares.



Cree aplicaciones basadas en datos en el entorno de codificación libre y fácil de usar de Qlik, presentado en asociación con Qlik .

Temas: MATERIAL ANGULAR, DEV  
WEB, DESARROLLO DE APLICACIONES WEB, ANGULAR

Publicado en DZone con permiso de Brian Fernandes ,  
DZone MVB . [Vea el artículo original aquí.](#)   
Las opiniones expresadas por los contribuyentes de  
DZone son propias.

# Obtenga lo mejor de Web Dev en su bandeja de entrada.

Manténgase actualizado con el boletín bimensual Web Dev  
de DZone. [VER UN EJEMPLO](#)

SUSCRIBIR

## Recursos para partners de Web Dev

Una guía para el desarrollo web de Java moderno con Crafter  
CMS

Crafter Software



Aprenda a experimentar el lado del servidor con Optimizely Full  
Stack. Vea una demostración de seis minutos hoy.

Optimizely



See how top engineering teams are experimenting in any  
application. Watch a webinar on Optimizely Full Stack.

Optimizely



Getting Started with Node.js, Docker, and Kubernetes  
NodeSource





