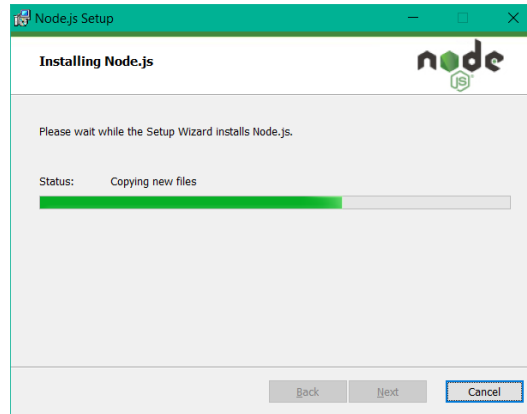


# 1. INSTALACIÓN DE HERRAMIENTAS

Actualización: 4/Nov/2019

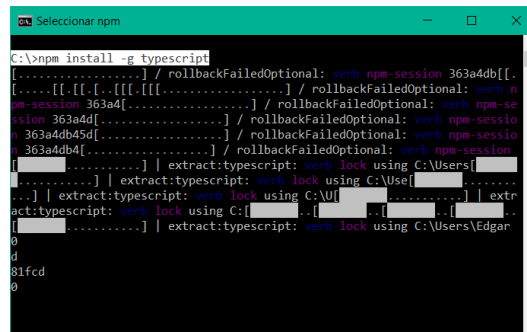
1. Instalar Node JS (12.13.0 LTS). Esto instala el NPM (Gestor de paquetes)

<https://nodejs.org/en/>



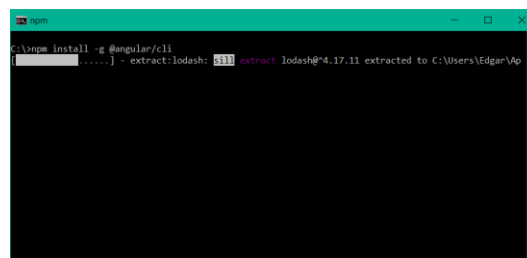
2. Instalar Typescript desde el paquete NPM con línea de comandos, para ello abra el CMD de Windows y teclee el siguiente comando:

`C:\>npm install -g typescript`



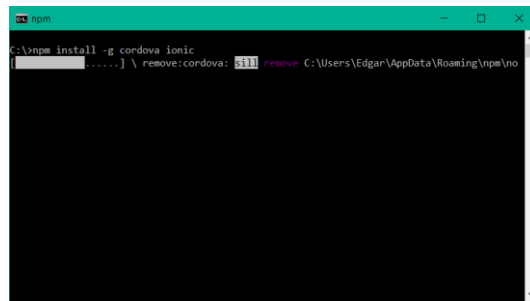
3. Instalar Angular CLI con el gestor NPM desde el paquete NPM con línea de comandos

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



**Nota:** Al final pregunta si desea compartir datos de uso, seleccionar “N” para No.

4. Instalar Ionic 2 con el gestor NPM desde el paquete NPM con línea de comandos  
C:\npm install -g cordova ionic



```
C:\>npm install -g cordova ionic
[.....] \ remove:cordova: 5.1.1 remove C:\Users\Edgar\AppData\Roaming\npm\node_modules\cordova\bin\cordova.cmd
```

5. Descargar e instalar VisualCode, de la página:

<https://code.visualstudio.com/>

## 2. CREAR PROYECTO ANGULAR

1. Crear carpeta donde quedará el Nuevo Proyecto, ejemplo:

`C:\tmp\ProyectoAngular`

2. Navegar hasta la carpeta desde línea de comandos
3. Comando para generar el proyecto

`ng new HolaMundo --skip-git`

4. Va a preguntar si desea agregar el routing, indicar que si:  
? Would you like to add Angular routing? (y/N): y
5. Solicitará una hoja

`CSS`

6. Para lanzar el servidor que proporciona angular, navegar hasta la ruta del proyecto en línea de comandos, `C:\tmp\ProyectoAngular\HolaMundo`, y escribir el siguiente comando:

`ng s`

7. El puerto default es el 4200, para cambiar de puerto, se lanza el siguiente comando

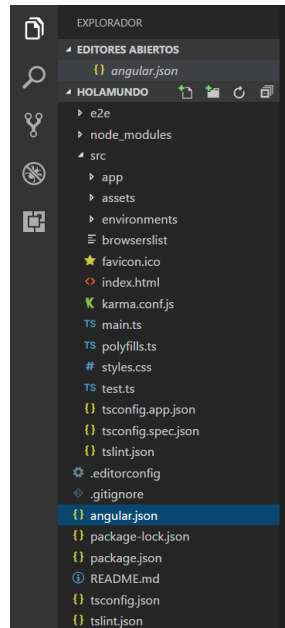
`ng s --port 4201`

8. Abrir un navegador web, y escribir la dirección:  
`localhost:4200`
9. Reiniciar la computadora

**Nota:** Una vez que se genere el proyecto, este puede ser guardado como un proyecto plantilla, para futuros proyectos solo se copia esta carpeta y se omite el proceso de crear un proyecto. La ventaja es que se podría crear un proyecto offline (sin acceso a internet).

### 3. ABRIR PROYECTO EN VISUAL CODE

1. Abrir Visual Code
2. Seleccionar la opción “**Archivo->Abrir carpeta...**”, seleccionar la carpeta del proyecto:  
C:\tmp\ProyectoAngular\HolaMundo
3. Borrar el archivo “package-lock.json”
4. Abrir el archivo “angular.json”



5. Incluir las referencias en la sección “**styles**” para utilizar los estilos de bootstrap en el proyecto (incluir las primeras 5 líneas):

```
25.     "styles": [  
26.         "node_modules/bootstrap/dist/css/bootstrap.min.css",  
27.         "node_modules/primeicons/primeicons.css",  
28.         "node_modules/primeng/resources/themes/saga-blue/theme.css",  
29.         "node_modules/primeng/resources/primeng.min.css",  
30.         "node_modules/font-awesome/css/font-awesome.css",  
31.         "src/styles.css"  
32.     ],
```

6. Agregar las siguientes referencias a la sección de “Scripts”.

```
33.     "scripts": [  
34.         "node_modules/jquery/dist/jquery.slim.js",  
35.         "node_modules/popper.js/dist/umd/popper.js",  
36.         "node_modules/bootstrap/dist/js/bootstrap.min.js"  
37.     ]
```

6. Abrir la terminal dentro de Visual Code, con la combinación “Ctrl+Ñ”, y ejecutar los siguientes comandos:

Instalar primeng:

```
PS C:\tmp\ProyectoAngular\HolaMundo> npm install primeng --save
```

Instalar primeicons

```
PS C:\tmp\ProyectoAngular\HolaMundo> npm install primeicons --save
```

Instalar font-awesome

```
PS C:\tmp\ProyectoAngular\HolaMundo> npm install font-awesome --save
```

Instalar jquery

```
PS C:\tmp\ProyectoAngular\HolaMundo> npm install jquery --save
```

Instalar popper

```
PS C:\tmp\ProyectoAngular\HolaMundo> npm install popper.js --save
```

Instalar bootstrap

```
PS C:\tmp\ProyectoAngular\HolaMundo> npm install bootstrap --save
```

Instalar ts-md5

```
PS C:\tmp\ProyectoAngular\HolaMundo> npm install ts-md5 --save
```

Instalar json-server

```
PS C:\tmp\ProyectoAngular\HolaMundo> npm install json-server --save
```

Instalar jsonwebtoken

```
PS C:\tmp\ProyectoAngular\HolaMundo> npm install jsonwebtoken --save
```

Instalar CdkVirtualScrollViewport

```
PS C:\tmp\ProyectoAngular\HolaMundo> npm install @angular/cdk --save
```

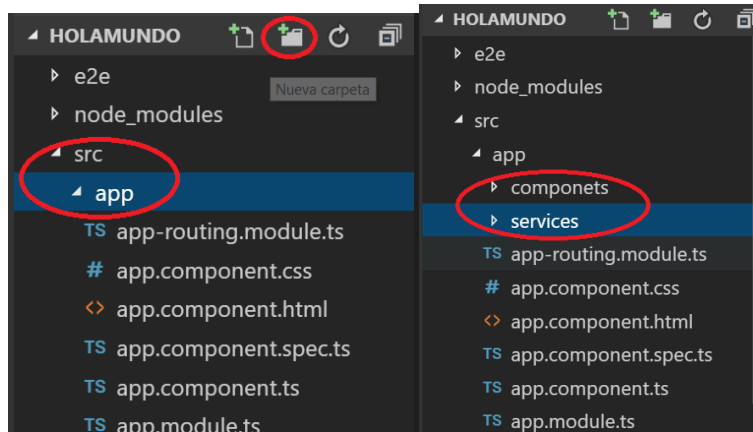
7. Compilar:

```
PS C:\tmp\ProyectoAngular\HolaMundo> ng s
```

**Nota:** En la compilación puede haber errores, lo más común es que se haya escrito mal alguna referencia o no se haya instalado el paquete correspondiente. Revisar los mensajes de error para determinar el origen del error y hacer las correcciones necesarias.

8. Abrir un navegador web, y escribir la dirección:  
localhost:4200

9. Crear las carpetas “Components” y “Services” dentro de la carpeta “src/app” del proyecto:



10. En la consola navegar hasta la carpeta “components”

```
PS C:\tmp\ProyectoAngular\HolaMundo> cd src/app/components
```

11. Crear un componente llamado “inicio”

```
PS C:\tmp\ProyectoAngular\HolaMundo\src\app\components> ng generate component inicio
```

12. Se crea una carpeta llamada “inicio” dentro de la carpeta “componets” con 4 archivos, eliminar el que se llama “inicio.component.spec”
13. Abrir el archivo llamado “app.component.html” que se encuentra dentro de “src/app”
14. Entrar a la página de bootstrap y copar un “NavBar” desde <https://getbootstrap.com/docs/4.1/components/navbar/>
15. Reemplazar el código del archivo “app.component.html” con el código del navbar de bootstrap, manteniendo la última línea de código

```
37. <router-outlet></router-outlet>
```

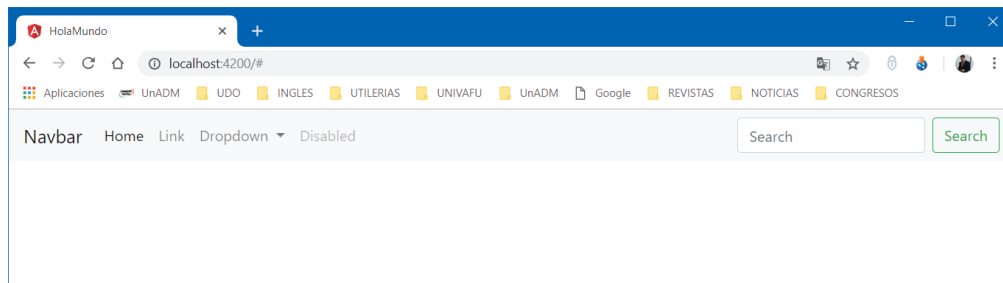
16. Navegar hasta “C:\tmp\ProyectoAngular\HolaMundo ” y Compilar:

```
PS C:\tmp\ProyectoAngular\HolaMundo> ng s
```

17. Abrir un navegador web, y escribir la dirección:

localhost:4200

18. Se visualizará la barra de navegación



19. Navegar hasta la carpeta “components” Crear un segundo componente llamado “fin”

PS C:\tmp\ProyectoAngular\HolaMundo\src\app\components> **ng generate component fin**

20. Entrar a la carpeta “src/app/components/fin” y eliminar el archivo “fin.component.spec.ts”

21. Abrir el archivo “app-routing.module.ts” desde la carpeta raíz “app” para agregar un routing (ruta de navegación).

22. Agregar los siguientes imports:

```
4. import { FinComponent } from "../components/fin/fin.component";
5. import { InicioComponent } from "../components/inicio/inicio.component";
```

23. En el mismo archivo “app-routing.module.ts” agregar las siguientes rutas:

```
8. const routes: Routes = [
9.   {path: "Inicio", component: InicioComponent},
10.  {path: "Fin", component: FinComponent},
11.  {path: "**", pathMatch: "full", redirectTo: "/Inicio"}
12.];
```

24. Abrir el archivo “app.module.ts” desde la carpeta “src/app”, y agregar el siguiente import:

```
25. import { BrowserAnimationsModule } from "@angular/platform-browser/animations";
```

26. Agregar los imports “BrowserAnimationsModule” en “@NgModule” dentro del mismo archive:

```
@NgModule({
  declarations: [
    AppComponent,
    InicioComponent,
    FinComponent
  ],
  imports: [
    BrowserModule,
    BrowserAnimationsModule,
    AppRoutingModule,
  ],
  providers: [],
  bootstrap: [AppComponent]
})
```

```

],
providers: [],
bootstrap: [AppComponent]
})

```

27. Abrir el archivo “app.component.html” y agregar el siguiente “div” fuera de la etiqueta `<nav></nav>`

```

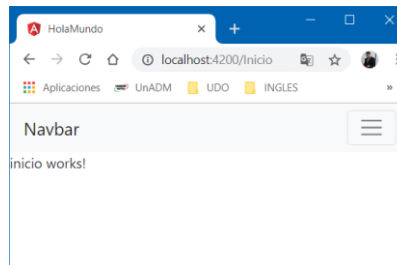
38.<div class="container">
39.  <router-outlet></router-outlet>
40.</div>

```

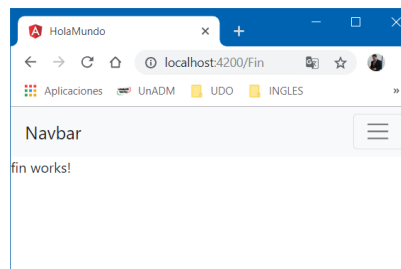
28. Navegar hasta la ruta del proyecto “C:\tmp\ProyectoAngular\HolaMundo” y compilar  
 PS C:\tmp\ProyectoAngular\HolaMundo> **ng s**

29. Abrir un navegador web, y escribir las siguientes direcciones para comprobar que está funcionando el routing:

localhost:400/Inicio



localhost:400/Fin



30. Abrir el archivo “app.component.html” de la carpeta “src/app” para ligar el NavBar al routing:

31. Modificar los dos menús principales para que queden así:

```

<div class="collapse navbar-collapse" id="navbarSupportedContent">
  <ul class="navbar-nav mr-auto">

```



```
<li class="nav-item active">
  <a class="nav-link" routerLink="/Inicio">Inicio <span class="sr-
only">(current)</span></a>
</li>
<li class="nav-item">
  <a class="nav-link" routerLink="/Fin">Fin</a>
</li>
```

## 4. AGREGAR COMPONENTES DE PRIME

1. Ingresar a la documentación de PRIME: <https://www.primefaces.org/primeng>
2. Entrar a la sección de botón y copiar la línea del import
3. Abrir el archivo “app.module.ts” y agregar el import

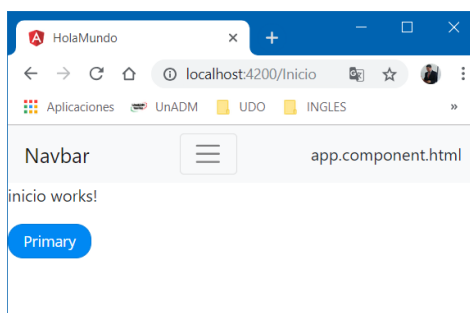
```
import {ButtonModule} from 'primeng/button';
```

4. En el mismo archivo, agregar en la sección “import” “ButtonModule” del “@NgModule”

```
5. @NgModule({  
6.   declarations: [  
7.     AppComponent,  
8.     InicioComponent,  
9.     FinComponent  
10.  ],  
11.  imports: [  
12.    BrowserModule,  
13.    BrowserModule,  
14.    AppRoutingModule,  
15.    ButtonModule  
16.  ],  
17.  providers: [],  
18.  bootstrap: [AppComponent]  
19.})
```

5. Abrir el componente en el que se desea utilizar el botón, por ejemplo componente “Inicio”, abrir el archivo “inicio.component.html” y pegar el código del botón

```
<button pButton type="button" label="Primary" class="ui-button-  
rounded"></button>
```



6. Entrar a la documentación de Prime, y entrar a la sección “Overlay”, copiar el import y pegarla en el archivo “app.module.ts” y agregarlo al import del “NgModules”

```
import {DialogModule} from 'primeng/dialog';
```

```
@NgModule({
  declarations: [
    AppComponent,
    InicioComponent,
    FinComponent
  ],
  imports: [
    BrowserModule,
    BrowserAnimationsModule,
    AppRoutingModule,
    ButtonModule,
    DialogModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
```

7. Copiar el código del componente “Dialogo” de “Prime”, y pegarlo al final del archivo “inicio.component.html” del componente “Inicio”, poner el nombre del header a “Acceso al sistema” y en nombre al formulario como “frmLogin”.
8. Abrir el archivo “inicio.component.ts” e inicializar el nombre del componente “frmLogin” a “false”:

```
9. export class InicioComponent implements OnInit {
10.   frmLogin: boolean = false;
11.
12.   constructor() { }
13.
14.   ngOnInit() {
15.   }
16.
17. }
```

9. En el mismo archivo, agregar una función llamada “MostrarDialog”:

```
mostrarDialog()
{
```

```
this.frmLogin = true;
}
```

10. Abrir el archivo “inicio.component.html” y agregar la directiva “(click)” al botón:

```
<button (click)="mostrarDialog()" pButton type="button" label="Primary" class="ui-  
button-rounded"></button>
```

11. Modificar el evento (click) de los botones para que se cierre el dialog, cambiar la etiqueta “display” por “frmLogin” que es el nombre del dialog. Con esto se afecta el valor del dialogo a false.

```
<p-footer>  
  <button type="button" pButton icon="pi pi-check" (click)="frmLogin=false"  
label="Yes"></button>  
  <button type="button" pButton icon="pi pi-close" (click)="frmLogin=false"  
label="No" class="ui-button-secondary"></button>  
</p-footer>
```

12. Compilar y probar

13. Abrir el archivo “inicio.component.ts” y agregar el import:

```
import { Router } from "@angular/router";
```

14. Modificar el constructor con el siguiente parámetro:

```
constructor(public router: Router)
```

15. En el mismo archivo, agregar una función:

```
iraFin()  
{  
  this.router.navigate(["/Fin"]);  
}
```

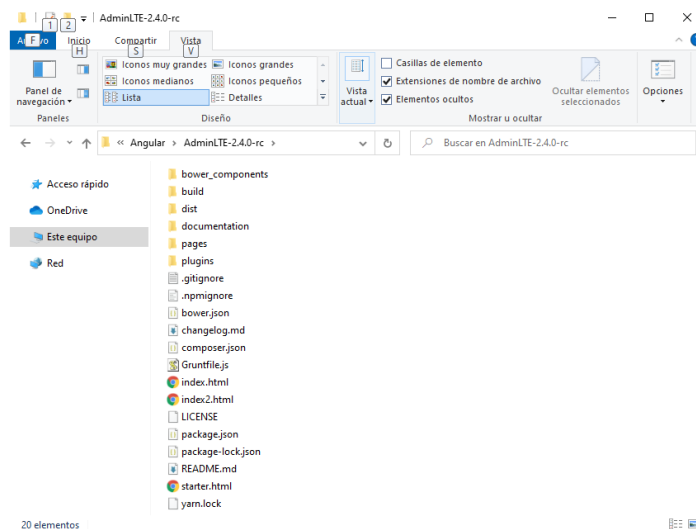
16. Abrir el archivo “inicio.component.html” y agregar la función al botón “Yes” del diálogo

```
<p-footer>
  <button type="button" pButton icon="pi pi-check" (click)="iraFin()"
label="Yes"></button>
  <button type="button" pButton icon="pi pi-close" (click)="frmLogin=false"
label="No" class="ui-button-secondary"></button>
</p-footer>
```

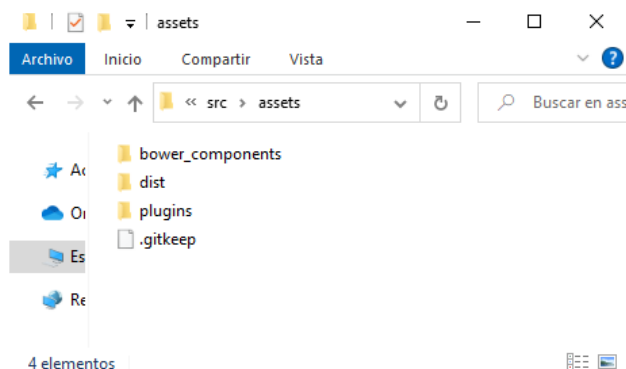
# 5. INTEGRAR TEMPLATE ADMIN-LTE

Siga estos pasos para integrar el Template de Admin-LTE en la Aplicación.

1. Descargar la versión “AdminLTE-2.4.0-rc” desde la siguiente página:  
<https://www.mgtechnologies.co.in/donate/8>
2. Descomprimir el archivo, se generará la carpeta “AdminLTE-2.4.0-rc” con el siguiente contenido:



3. Copiar las subcarpetas “bower\_components”, “dist” y “plugins” de la carpeta “AdminLTE-2.4.0-rc” a la carpeta “src/assets” del proyecto.



4. Abrir el archivo “starter.html” de la carpeta “AdminLTE-2.4.0-rc”, copiar el código **js (en la sección del body)** y **css (en la sección del head)** al archivo “index.html” del proyecto
5. Revisar las rutas, asegurar que apuntan a la carpeta “assets”

6. Crear la carpeta “Layout” dentro de la carpeta “app” del proyecto
7. Crear el componente “LayoutHeader” dentro de la carpeta “Layout”

Código:

```
ng g c --skipTests=true Layout/LayoutHeader
```

Nota:

Copiar el código englobado por “<header class="main-header">” que está en el archivo “starter.html”

8. Cree el componente “LayoutMenu” dentro de la carpeta “Layout”

Código:

```
ng g c --skipTests=true Layout/LayoutMenu
```

Nota:

Copiar el código englobado por <aside class="main-sidebar">” que está en el archivo “starter.html”

9. Cree el componente “LayoutContent” dentro de la carpeta “Layout”

Código:

```
ng g c --skipTests=true Layout/LayoutContent
```

Nota:

Copiar el código englobado por “<div class="content-wrapper">” que está en el archivo “starter.html”

10. Cree el componente “LayoutFooter” dentro de la carpeta “Layout”

Código:

```
ng g c --skipTests=true Layout/LayoutFooter
```

Nota:

Copiar el código englobado por “<footer class="main-footer">” que está en el archivo “starter.html”

11. Cree el componente “LayoutSettings” dentro de la carpeta “Layout”

Código:

```
ng g c --skipTests=true Layout/LayoutSettings
```

Nota:

Copiar el código englobado por "<aside class='control-sidebar control-sidebar-dark'" que está en el archivo "starter.html"

12. Copiar la clase del body del archivo "starter.html" al body del archivo "index.html" de nuestro proyecto:

```
<body class="hold-transition skin-blue sidebar-mini">
```

13. Pega el siguiente código en el archivo "app.component.html" del proyecto

```
<div class="wrapper">
  <app-layout-header></app-layout-header>
  <app-layout-menu></app-layout-menu>
  <app-layout-content></app-layout-content>
  <app-layout-footer></app-layout-footer>
  <app-layout-settings></app-layout-settings>
</div>
```

- 14.

Seguir este link para la versión 2.4.5:

<https://www.youtube.com/watch?v=47UW2U50TDA>

- 1.

Nota: Si falla el TreeView:

1. Instalar:

```
npm install jquery jquery-ui-dist
```

2. Agregar a la sección de Scripts de Angular.json:

```
"scripts": [
  "../node_modules/jquery/dist/jquery.min.js",
  "../node_modules/jquery-ui-dist/jquery-ui.js"
],
```

## ADMIN-LTE 2.4.5



3. Abrir el componente donde se utiliza el TreeView, declarar una variable en la sección de los imports:

declare var \$: any;

Luego escribir el método onInit

```
ngOnInit(): void
{
  $(document).ready(() => {
    const trees: any = $('[data-widget="tree"]');
    if (trees) {
      trees.tree();
    }
  });
}
```

## ADMIN-LTE 3.0.5

1. Abrir el componente donde se utiliza el TreeView, declarar una variable en la sección de los imports:

2. declare var \$: any;

3. import \* as AdminLte from 'admin-lte';

Luego escribir el método onInit

```
ngOnInit(): void {
  // $('[data-widget="treeview"]').Treeview('init');
  $('[data-widget="treeview"]').each(function () {
    AdminLte.Treeview._jQueryInterface.call($(this), 'init');
    AdminLte.Layout._jQueryInterface.call($('body'));
    AdminLte.PushMenu._jQueryInterface.call $('[data-widget="pushmenu"]');
  });
}
```

- 4.

## 6. VARIOS

1. Crear un módulo

```
ng g m {NombreModulo}
```

2. Crear un componente sin archivo de pruebas

```
ng g c --skipTests=true {NombreComponente}
```

Prefijo: Cmp

Ejemplo: CmpLogin

3. Crear un modelo

```
ng g cl --skipTests=true {NombreModelo}
```

Prefijo: Mdl

Ejemplo: MdlUsuarios

4. Crear un Enumerado

```
ng g enum {NombreEnumerado}
```

Prefijo: Enm

Ejemplo: EnmRoles

5. Crear un servicio sin archivo de pruebas

```
ng g s --skipTests=true {NombreServicios}
```

Prefijo: Srv

Ejemplo: SrvLogin

# 7. PÁGINAS DE INTERÉS

---

## PÁGINAS DE INTERÉS

---

Colores Bootstrap:

<https://getbootstrap.com/docs/4.0/utilities/colors/>

w3Schools

<https://www.w3schools.com/default.asp>

---

ACTUALIZAR ANGULAR CLI AND ANGULAR

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

