

En este ejemplo vamos a ver cómo usar nuevos componentes para Angular basados en BootStrap con **Ng-BootStrap** , y el conjunto **PrimeNG** que complementan los de **Angular Material**.

Comenzamos con Ng-BootStrap:

- Creamos un nuevo proyecto `ng new ngBootS` (este ejemplo lo haremos sin módulos -- **standalone**)
- Cambiamos al directorio creado ...
- Añadimos el siguiente paquete relacionado con la internacionalización de la aplicación `ng add @angular/localize` , necesario para algunas versiones de **ng-bootstrap**.
- Instalamos el paquete `npm install --save @ng-bootstrap/ng-bootstrap`
- También instalamos el paquete `npm install bootstrap` . Decir, que no hace falta instalar **jQuery** , ya que aunque es necesario para algunas operaciones con BootStrap , con **Ng-BootStrap** se reemplaza esas dependencias evitando posibles conflictos que causa el uso de jQuery con Angular.
- Añadimos en estilos la línea referente al módulo bootstrap en el archivo de configuración **angular.json** , debe quedar como sigue:

```
"styles": [  
  "node_modules/bootstrap/dist/css/bootstrap.min.css",  
  "src/styles.css"  
],
```

Para usar y probar **Ng-BootStrap**, en **app.component.ts** importamos y añadimos al array **imports** el nombre del módulo necesario.

```
import {NgbAccordionModule} from '@ng-bootstrap/ng-bootstrap';
```

A continuación pegamos el siguiente código (sustituyendo el actual) en **app.component.html**, con el cual probaremos clases de **BootStrap**, y un

componente de **Ng-BootStrap** ([ngbAccordion](https://ng-bootstrap.github.io/#/components/accordion/examples)) (ver enlace: <https://ng-bootstrap.github.io/#/components/accordion/examples>)

```
<div ngbAccordion>

  <div ngbAccordionItem [collapsed]="false">

    <h2 ngbAccordionHeader>

      <button ngbAccordionButton>Simple</button>

    </h2>

    <div ngbAccordionCollapse>

      <div ngbAccordionBody>

        <ng-template>

          Anim pariaturnam cliche reprehenderit, enim eiusmod high life accusamus terry richardson ad squid. 3 wolf moon officia aute, non cupidatat skateboard dolor brunch. Food truck quinoa nesciunt laborum eiusmod. Brunch 3 wolf moon tempor, sunt aliqua put a bird on it squid single-origin coffee nulla assumenda shoreditch et. Nihil anim keffiyeh helvetica, craft beer labore wes anderson cred nesciunt sapiente ea proident. Ad vegan excepteur butcher vice lomo. Leggings occaecat craft beer farm-to-table, raw denim aesthetic synth nesciunt you probably haven't heard of them accusamus labore sustainable VHS.

        </ng-template>

      </div>

    </div>

  </div>

</div>
```

```
<div ngbAccordionItem>

  <h2 ngbAccordionHeader>

    <button ngbAccordionButton>

      <span>#9733; <b>Fancy</b> title &#9733;</span>

    </button>

  </h2>

  <div ngbAccordionCollapse>

    <div ngbAccordionBody>

      <ng-template>

        Anim pariatum cliche reprehenderit, enim eiusmod
high life accusamus terry richardson ad squid. 3 wolf moon
        officia aute, non cupidatat skateboard dolor brunch.
Food truck quinoa nesciunt laborum eiusmod. Brunch 3 wolf
        moon tempor, sunt aliqua put a bird on it squid
single-origin coffee nulla assumenda shoreditch et. Nihil anim
        keffiyeh helvetica, craft beer labore wes anderson
cred nesciunt sapiente ea proident. Ad vegan excepteur
        butcher vice lomo. Leggings occaecat craft beer
farm-to-table, raw denim aesthetic synth nesciunt you probably
        haven't heard of them accusamus labore sustainable
VHS.

      </ng-template>

    </div>

  </div>

</div>

<div ngbAccordionItem [disabled]="true">

  <h2 ngbAccordionHeader>
```

```
<button ngbAccordionButton>Disabled</button>

</h2>

<div ngbAccordionCollapse>

  <div ngbAccordionBody>

    <ng-template>

      Anim pariatum cliche reprehenderit, enim eiusmod
high life accusamus terry richardson ad squid. 3 wolf moon
      officia aute, non cupidatat skateboard dolor brunch.
Food truck quinoa nesciunt laborum eiusmod. Brunch 3 wolf
      moon tempor, sunt aliqua put a bird on it squid
single-origin coffee nulla assumenda shoreditch et. Nihil anim
      keffiyeh helvetica, craft beer labore wes anderson
cred nesciunt sapiente ea proident. Ad vegan excepteur
      butcher vice lomo. Leggings occaecat craft beer
farm-to-table, raw denim aesthetic synth nesciunt you probably
      haven't heard of them accusamus labore sustainable
VHS.

    </ng-template>

  </div>

</div>

</div>

</div>
```

Finalmente, ejecutamos el servidor `ng serve -o`

*Haced pruebas con otros componentes de **Ng-BootStrap** teniendo en cuenta que en algunos casos se usan atributos de clases añadidos en el archivo TS del componente.*

Creamos un nuevo proyecto para usar PrimeNG:

```
ng new ngPrime
```

-

```
npm install primeng @primeng/themes
```

Añadimos al archivo **app.config.ts** los archivos a importar:

```
import { provideAnimationsAsync } from
 '@angular/platform-browser/animations/async';

import { providePrimeNG } from 'primeng/config';

import Aura from '@primeng/themes/aura';

import { routes } from './app.routes';
```

Y como proveedores (debe quedar de algo como ...):

```
export const appConfig: ApplicationConfig = {

  providers: [provideZoneChangeDetection({
eventCoalescing: true }),

    provideRouter(routes),

    provideAnimationsAsync(),

    providePrimeNG({

      theme: {

        preset: Aura

      }

    })

  ]

};
```

Usamos un componente **menú** (<https://primeng.org/menubar>)

En **app.component.ts** importamos un ejemplo de componente menú:

```
import { MenubarModule } from 'primeng/menubar';  
  
import { MenuItem } from 'primeng/api';
```

- Dentro de la clase en el mismo archivo añadimos:

```
items: MenuItem[] | undefined;  
  
ngOnInit() {  
    this.items = [  
        {  
            label: 'Home',  
            icon: 'pi pi-home'  
        },  
        {  
            label: 'Features',  
            icon: 'pi pi-star'  
        },  
        {  
            label: 'Projects',  
            icon: 'pi pi-search',  
            items: [  
                {  
                    label: 'Components',  
                    icon: 'pi pi-bolt'
```

```
    },  
    {  
      label: 'Blocks',  
      icon: 'pi pi-server'  
    },  
    {  
      label: 'UI Kit',  
      icon: 'pi pi-pencil'  
    },  
    {  
      label: 'Templates',  
      icon: 'pi pi-palette',  
      items: [  
        {  
          label: 'Apollo',  
          icon: 'pi pi-palette'  
        },  
        {  
          label: 'Ultima',  
          icon: 'pi pi-palette'  
        }  
      ]  
    }  
  ]  
},  
},
```

```
{
  label: 'Google',
  icon: 'pi pi-envelope'
}
]
}
```

- Y sustituimos el código de **app.component.html** por :

```
<div class="card">
  <p-menubar [model]="items" />
</div>
```

- Finalmente, ejecutamos el servidor `ng serve`

Podríamos enlazar el menú con páginas externas o enrutar internamente.

*Por ejemplo, si añadimos la propiedad **url** (obtenido de la API del propio componente) a una de las opciones que define el menú en la variable **items** , (para **Google** por ejemplo), podemos enlazar a la página de Google de la siguiente forma:*

```
{
  label: 'Google',
  icon: 'pi pi-envelope',
  url: 'https://www.google.com'
}
```

Comprobamos

*Haced pruebas con otros componentes de **PrimeNG** .*