

REQUERIMIENTOS

Instalar

- Node JS - <https://nodejs.org/es/>
- npm i -g @ionic/cli - <https://ionicframework.com/>

Después de instalar podemos comprobar versiones

Ionic -v

Node -v

Creamos un nuevo proyecto

- **ionic start ejApp sidemenu**

[Starter Templates](#)

[Command List](#)

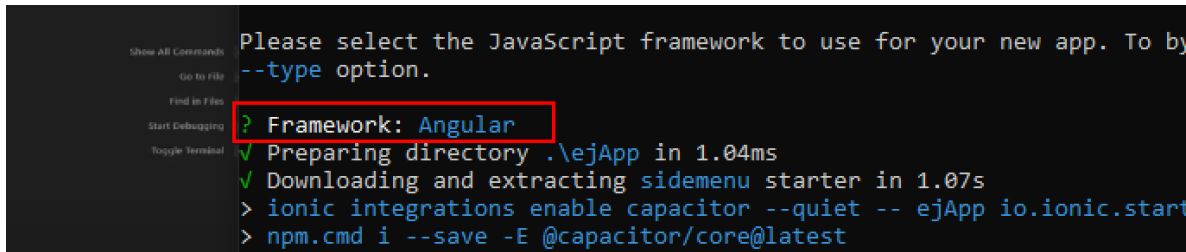
[Configuring](#)

Starter Types

Ionic Angular

Starter	Description
tabs	A starting project with a simple tabbed interface
blank	A blank starter project
sidemenu	A starting project with a side menu with navigation in the content area
super	A starting project complete with pre-built pages, providers and best practices for Ionic development.
conference	A project that demonstrates a realworld application
tutorial	A tutorial based project that goes along with the Ionic documentation
aws	AWS Mobile Hub Starter

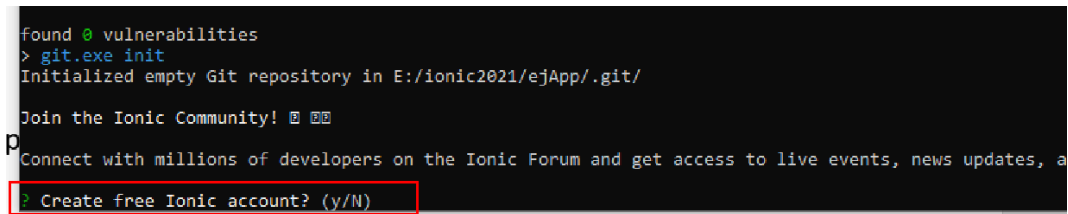
Al crea la aplicación nos pregunta por el framework a utilizar en este caso seleccionamos Angular.



```
Show All Commands
Go to File
Find in Files
Start Debugging
Toggle Terminal

Please select the JavaScript framework to use for your new app. To by
--type option.
? Framework: Angular
✓ Preparing directory .\ejApp in 1.04ms
✓ Downloading and extracting sidemenu starter in 1.07s
> ionic integrations enable capacitor --quiet -- ejApp io.ionic.start
> npm.cmd i --save -E @capacitor/core@latest
```

Luego nos pregunta si queremos crear una cuenta en ionic.



```
found 0 vulnerabilities
> git.exe init
Initialized empty Git repository in E:/ionic2021/ejApp/.git/

Join the Ionic Community! [?]
p Connect with millions of developers on the Ionic Forum and get access to live events, news updates, a
? Create free Ionic account? (y/N)
```

Ejecutamos nuestro proyecto.

ionic serve

Para crear nuestra paginas para nuestra aplicación

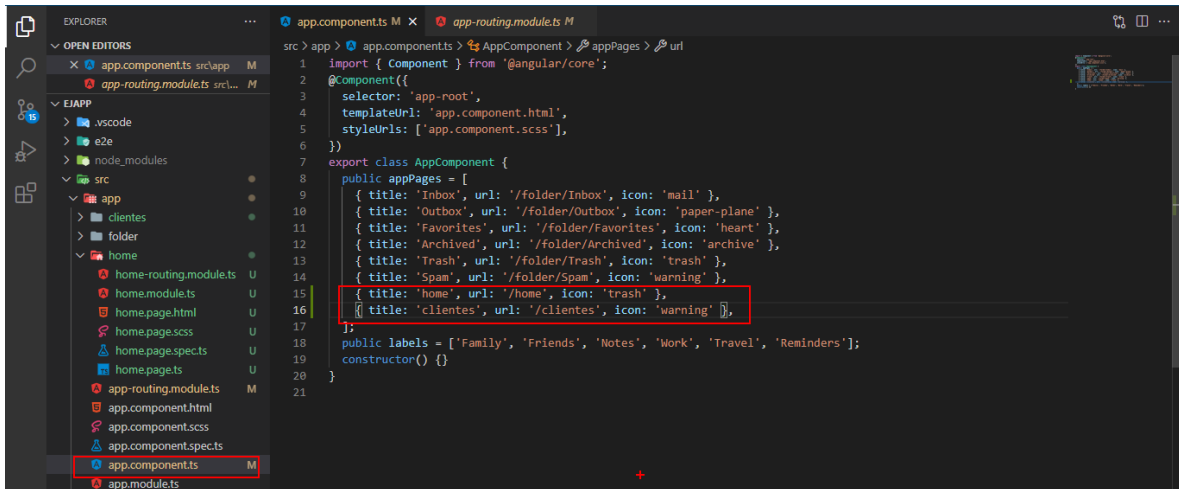
ionic generate page home

o también

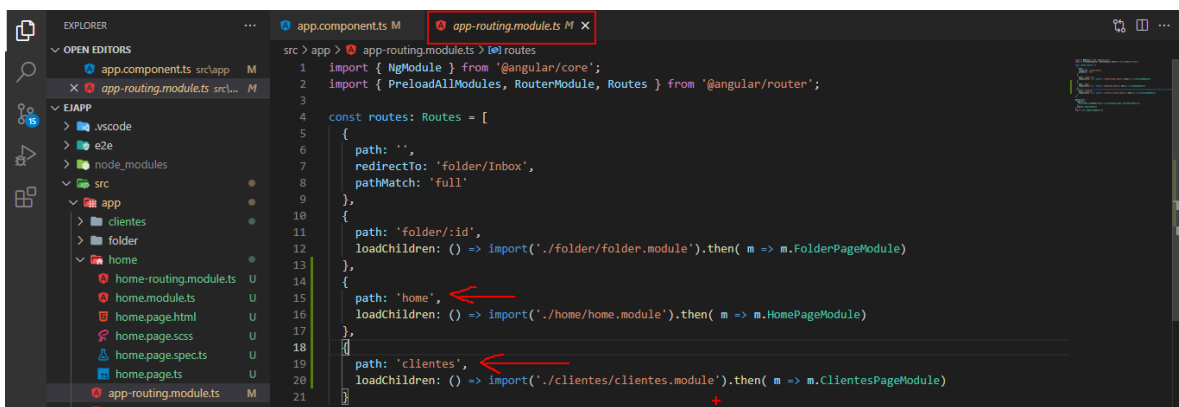
ionic g page home

ionic g page clientes

Ubicamos nuestro archivo app.component.ts y agregamos las opciones para llamar a nuestras páginas creadas.

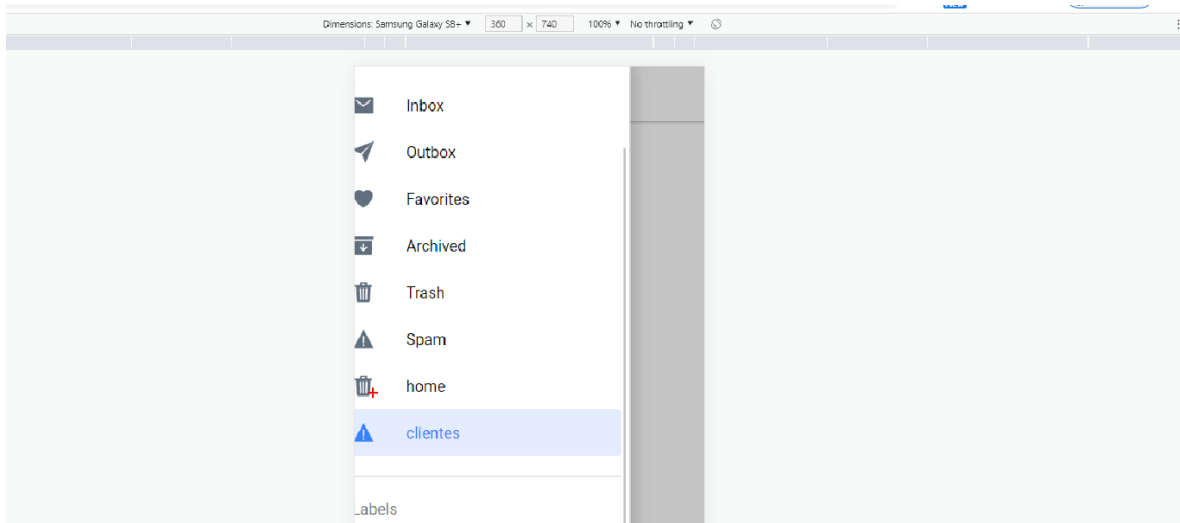


Debemos verificar el nombre de la rutas que se crearon en el app-routing.module.ts.

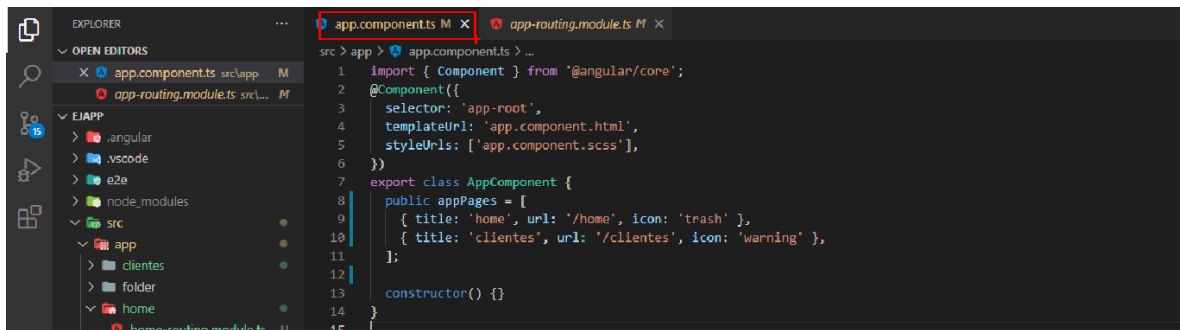


Podemos ejecutar y aparece en nuestro navegador la aplicación, presionando F12 podemos seleccionar un dispositivo móvil para ver cómo funciona nuestra aplicación.

Ionic serve



Podemos sacar las opciones que nos aparecen para lo cual nos vamos al archivo `app.components.ts` y borramos en el arreglo de `appPages` y `labels`.

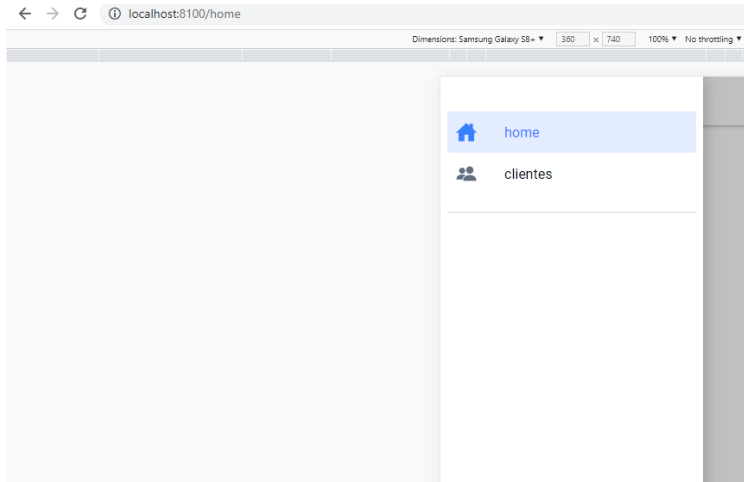


También debemos modificar en el archivo `app.component.html` borrando lo señalado.



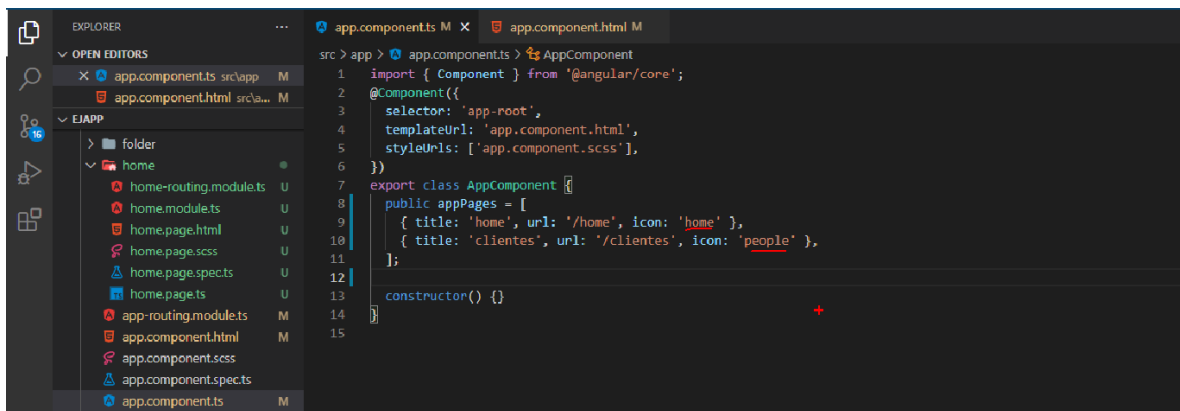
Ejecutamos

Ionic serve



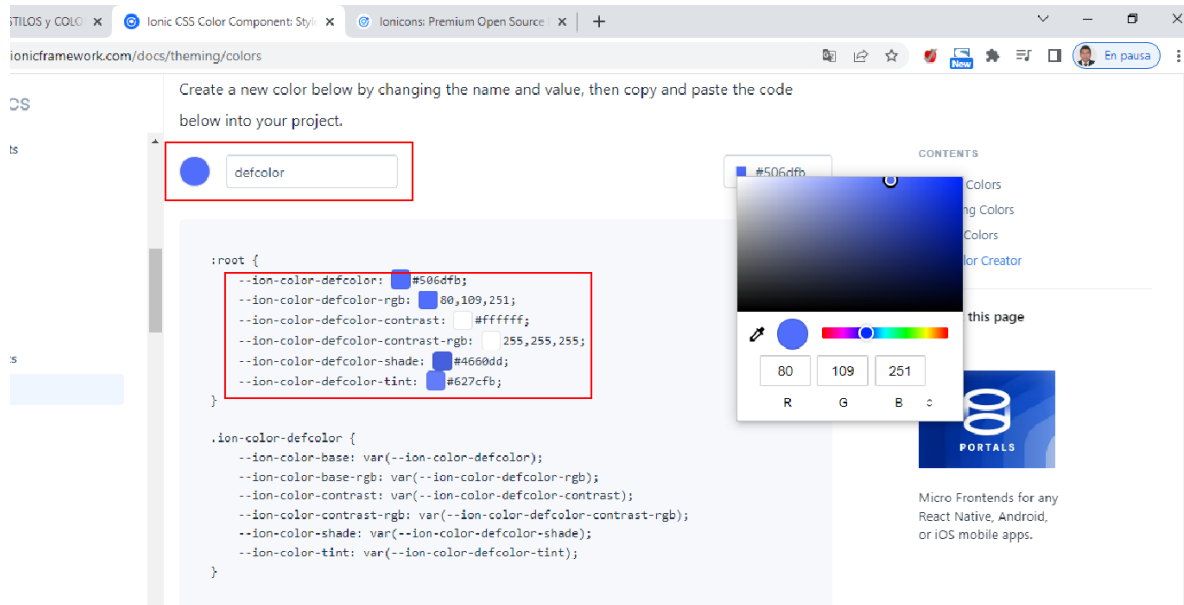
Podemos cambiar los iconos para lo cual buscamos iconos en la página web.

<https://ionic.io/ionicons>

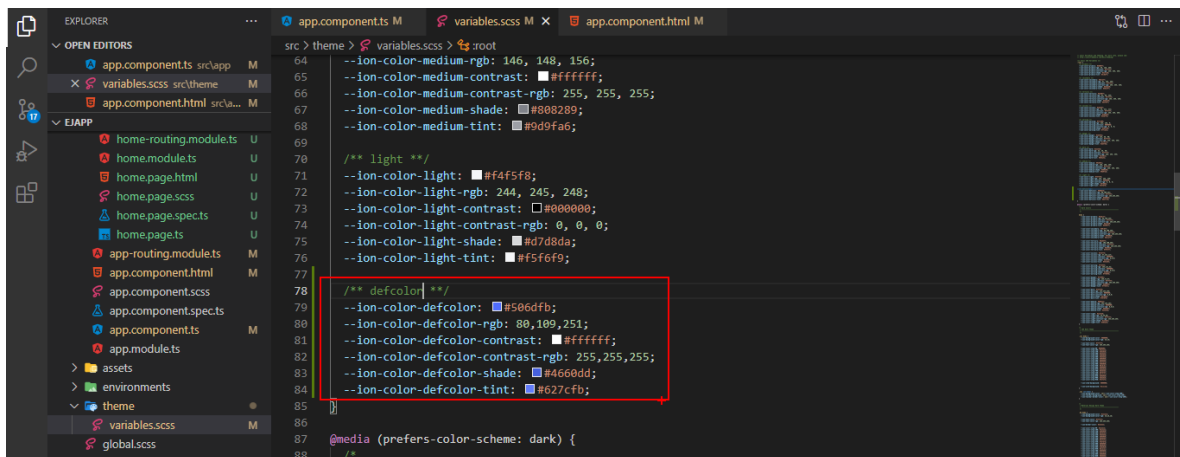


Podemos definir colores para nuestra aplicación ingresamos a la página generamos un color que luego lo utilizaremos en nuestra aplicación.

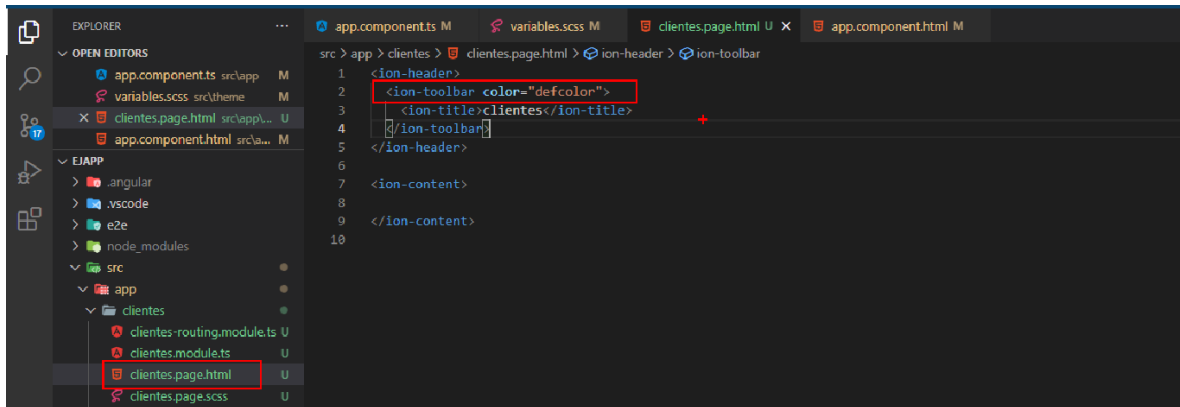
<https://ionicframework.com/docs/theming/colors>



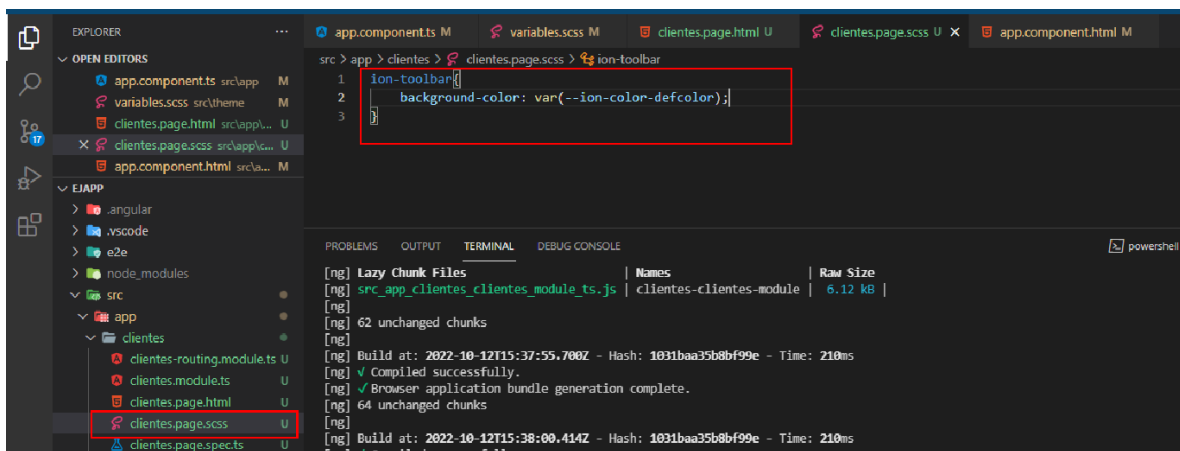
Definidos los colores ubicamos el archivo varbles.css y agregamos la configuración de color.



Luego ubicamos nuestra página clientes.page.html y referenciamos el color a utilizar en el código.

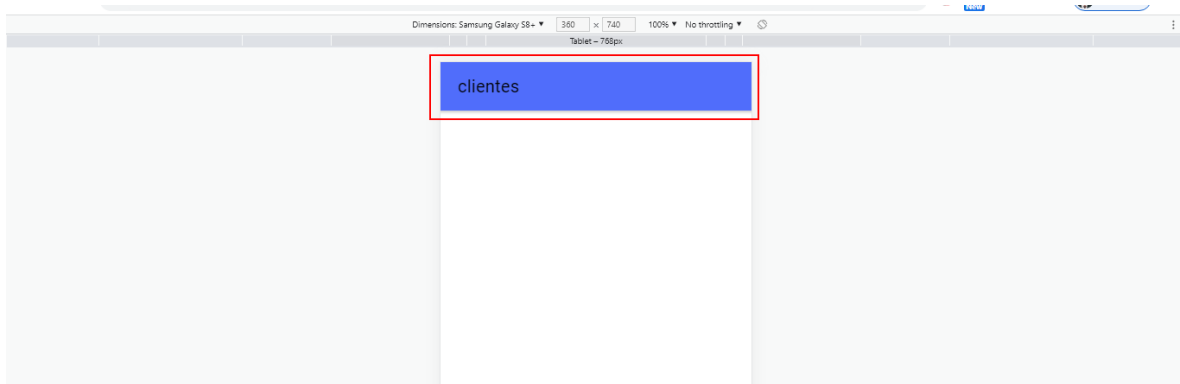


Luego ubicamos el archivo `clientes.page.scss`

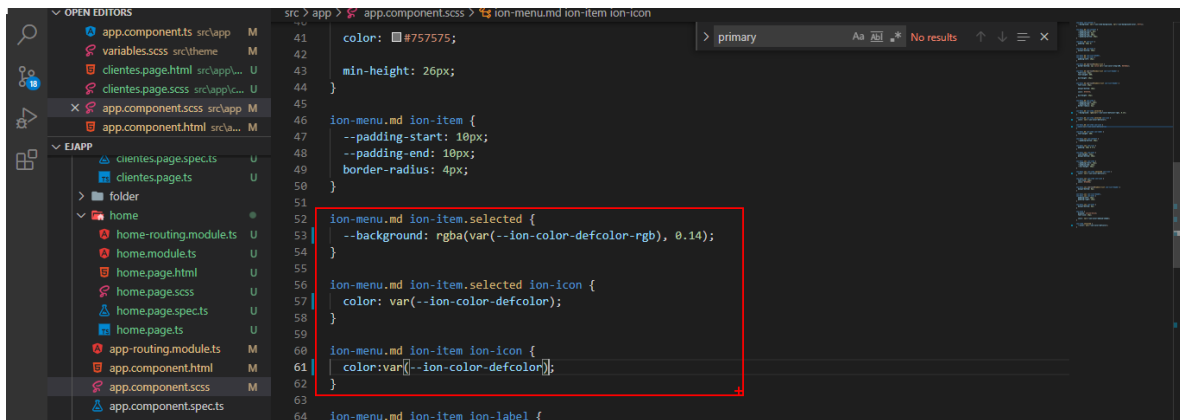


Ejecutamos y podemos ver qué cambio el color de la página clientes.

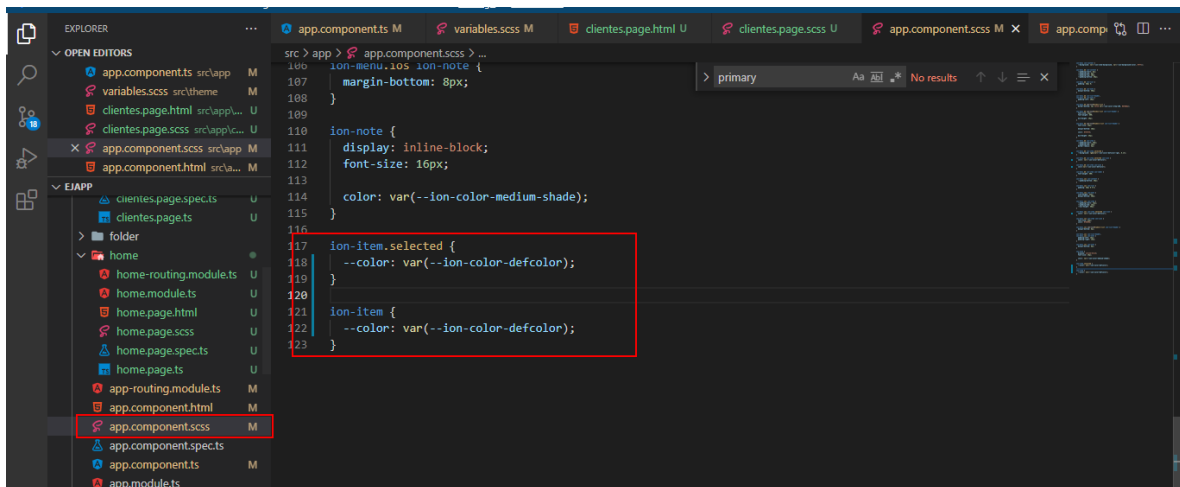
Ionic serve



Ahora ubicamos el archivo `app.component.scss` y modificamos los colores del ítem seleccionado y de los iconos.



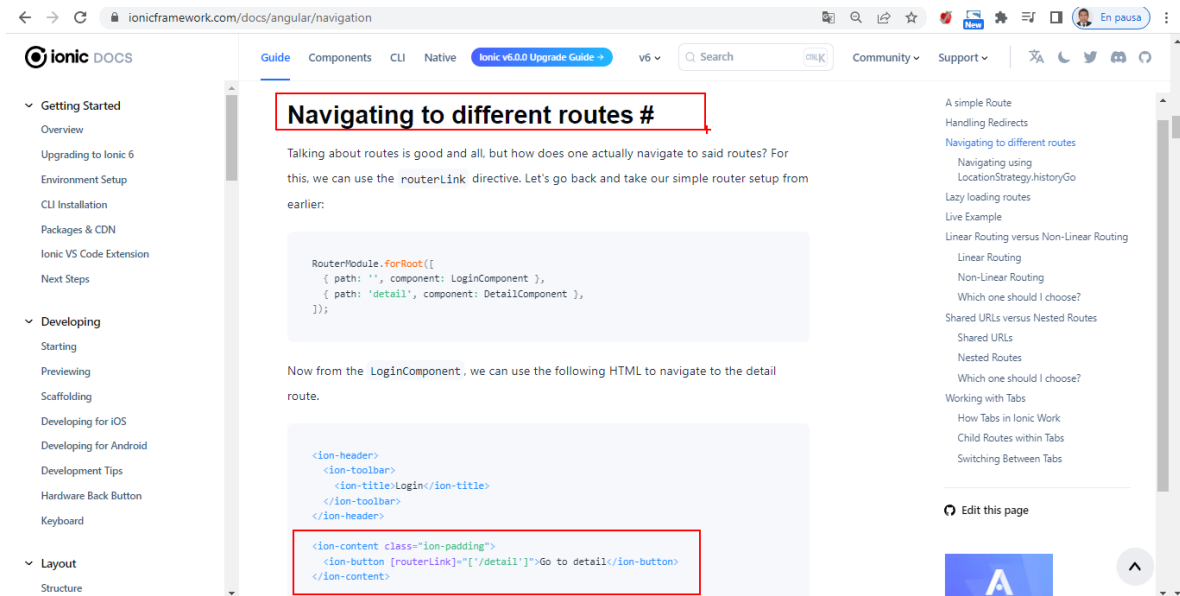
Corregimos los colores para el ítem seleccionado y no seleccionado del menú de la aplicación.



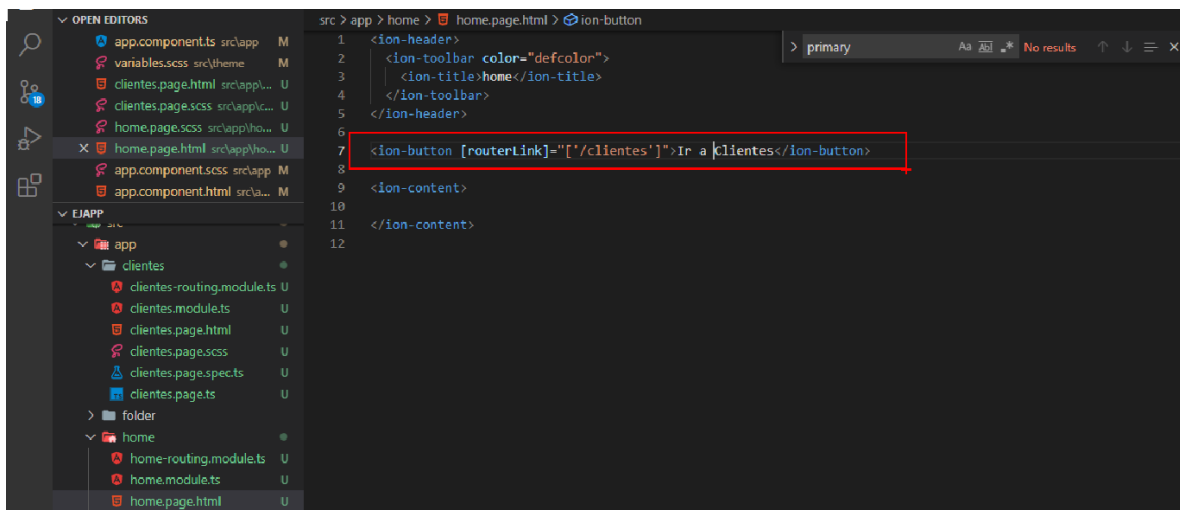
NAVEGACIÓN ENTRE PAGINAS

Ingresamos a la página de documentación con angular

<https://ionicframework.com/docs/angular/navigation>

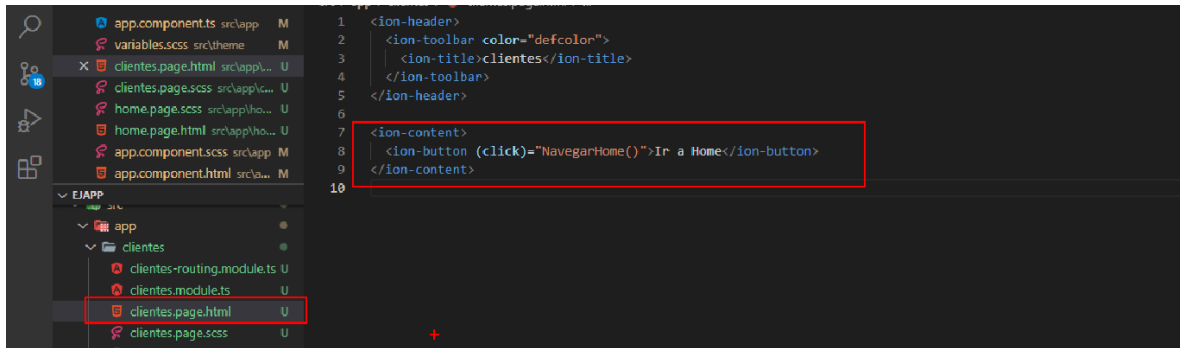


Ubicamos el [home.page.html](#) y agregamos el código modificando el nombre de la ruta.

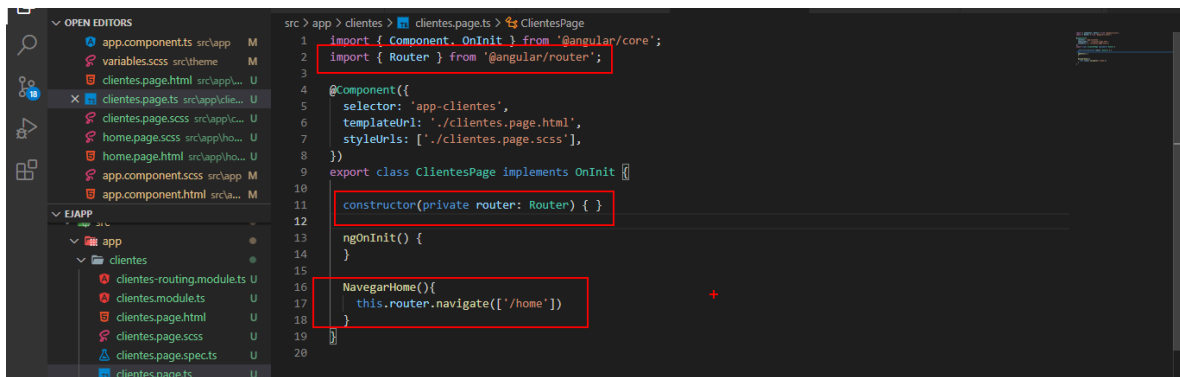


Otra forma es hacerlo desde código typescript

Primero debemos ubicar nuestro archivo `clientes.page.html` y programar creando un botón y llamando a una función en el evento click `NavegarHome()`

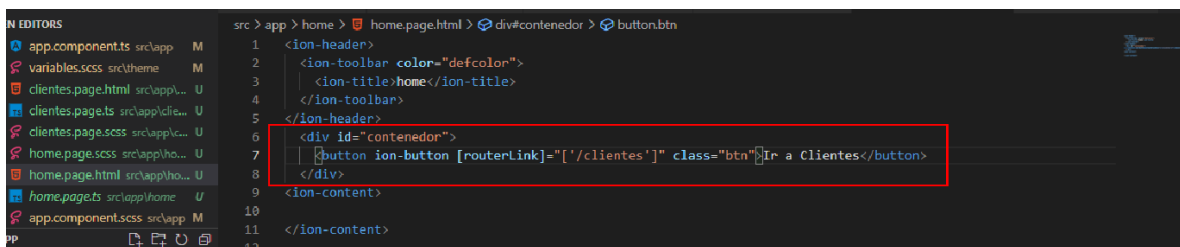


Luego ubicamos el archivo `clientes.page.ts` importamos el Router y definimos la función `NavegarHome`.

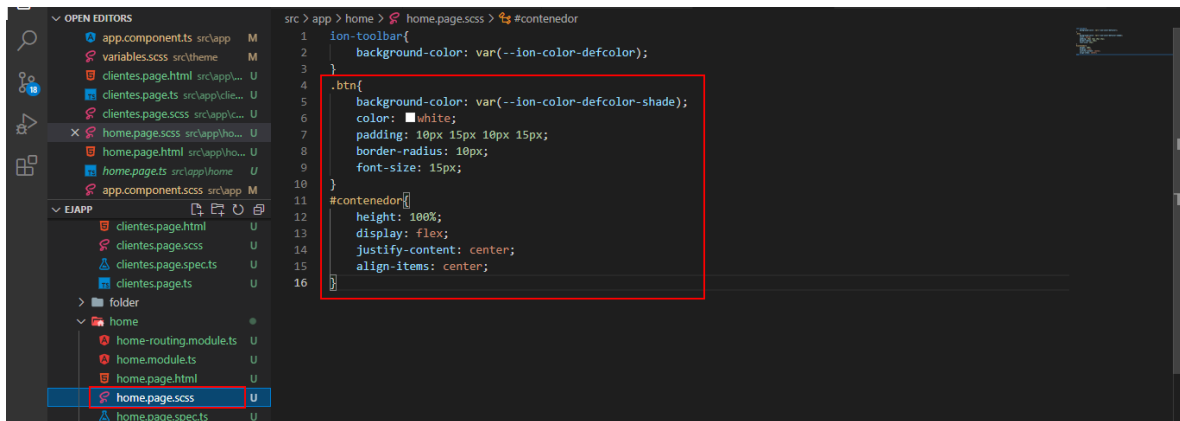


DEFINIENDO ESTILOS AL BOTON

Podemos modificar el código del botón por la etiqueta `button` y definir una clase `btn` e indicar un id contenedor.



Ubicamos el archivo [home.page.css](#) y agregamos código definiendo los estilos.



TRABAJANDO GRID (ION-GRID)

En el archivo [home.page.html](#) trabajamos con ion-grid, ion-row y ion-col.

```
<ion-header>
  <ion-toolbar color="defcolor">
    <ion-title>home</ion-title>
  </ion-toolbar>
</ion-header>
<ion-content>
  <ion-grid>
    <ion-row>
      <ion-col sizeLg="8" sizeMd="4" sizeXs="3" size-sm><div class="texto">Texto 1</div></ion-col>
      <ion-col sizeLg="4" sizeMd="8" sizeXs="9" size-sm><div class="texto">Texto 2</div></ion-col>
    </ion-row>
    <ion-row>
      <ion-col><div class="texto">Texto 3</div></ion-col>
      <ion-col><div class="texto">Texto 4</div></ion-col>
    </ion-row>
  </ion-grid>
</ion-content>
```

En el archivo [home.page.scss](#) creamos una clase texto.

```
.texto{
  background-color:yellowgreen;
  text-align: center;
  padding-top: 20px;
  padding-bottom: 20px;
}
```

```
}
```

TRABAJANDO LISTAS (ION-LIST)

Podemos trabajar con el ion-list agregamos código a nuestro archivo clientes.pages.html

```
<ion-list>
  <ion-list-header> Recent Conversations </ion-list-header>

  <ion-item>
    <ion-avatar slot="start">
      
    </ion-avatar>
    <ion-label>
      <h2>Finn</h2>
      <h3>I'm a big deal</h3>
      <p>Listen, I've had a pretty messed up day...</p>
    </ion-label>
  </ion-item>

  <ion-item>
    <ion-avatar slot="start">
      
    </ion-avatar>
    <ion-label>
      <h2>Han</h2>
      <h3>Look, kid...</h3>
      <p>I've got enough on my plate as it is, and I...</p>
    </ion-label>
  </ion-item>

  <ion-item>
    <ion-avatar slot="start">
      
    </ion-avatar>
    <ion-label>
      <h2>Rey</h2>
      <h3>I can handle myself</h3>
      <p>You will remove these restraints and leave...</p>
    </ion-label>
  </ion-item>
</ion-list>
```

```
</ion-item>

<ion-item>
  <ion-avatar slot="start">
    
  </ion-avatar>
  <ion-label>
    <h2>Luke</h2>
    <h3>Your thoughts betray you</h3>
    <p>I feel the good in you, the conflict...</p>
  </ion-label>
</ion-item>
</ion-list>
```

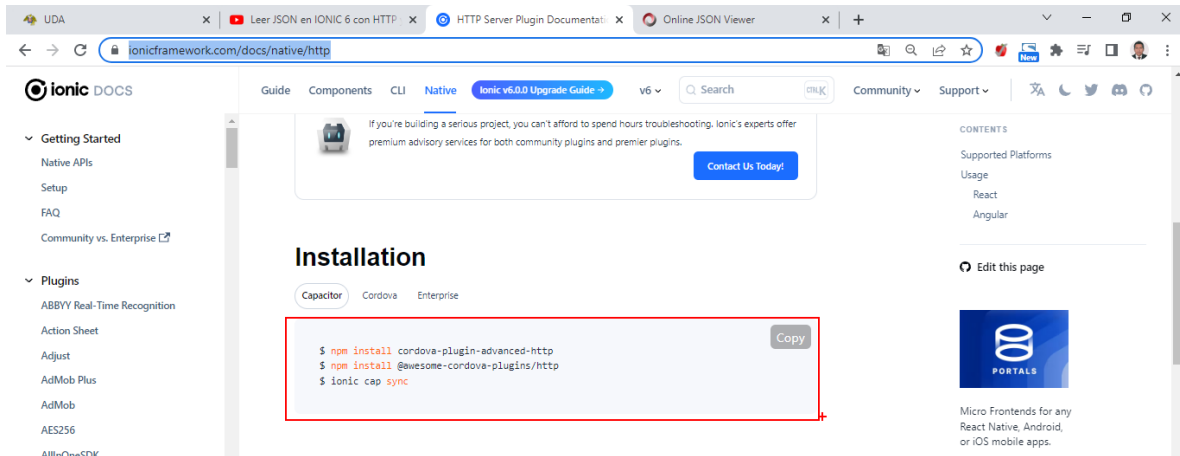
LLENANDO UN ION-LIST DESDE UN JSON

Creemos un archivo clientes.json y guardamos en la carpeta assets y un folder archivos.

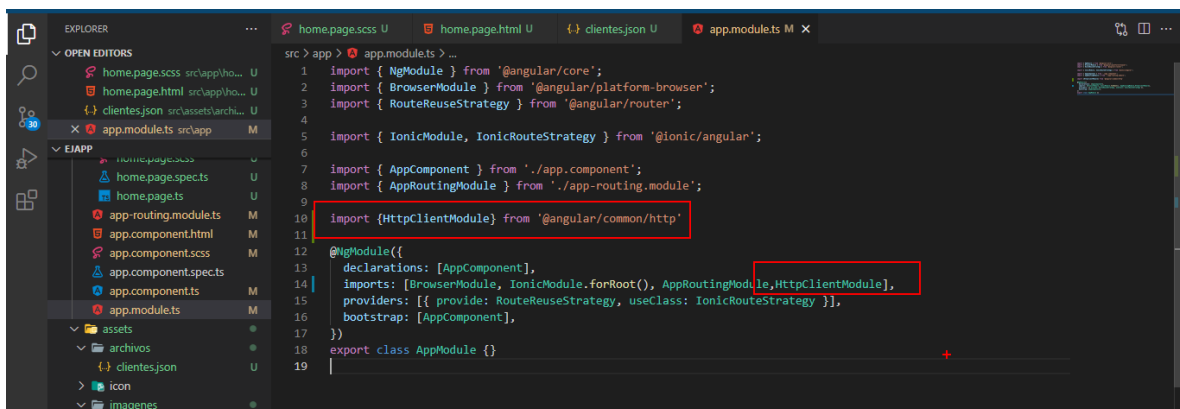
```
{
  "data" : [
    {
      "nombre": "Finn",
      "texto": "I'm a big deal",
      "descripcion": "Listen, I've had a pretty messed up day",
      "foto": "avatar-finn.png"
    },
    {
      "nombre": "Han",
      "texto": "Look, kid",
      "descripcion": "I've got enough on my plate as it is, and I",
      "foto": "avatar-han.png"
    },
    {
      "nombre": "Luke",
      "texto": "Your thoughts betray you",
      "descripcion": "I feel the good in you, the conflict",
      "foto": "avatar-luke.png"
    }
  ]
}
```

Vamos instalamos el plugin de http vamos a la pagina

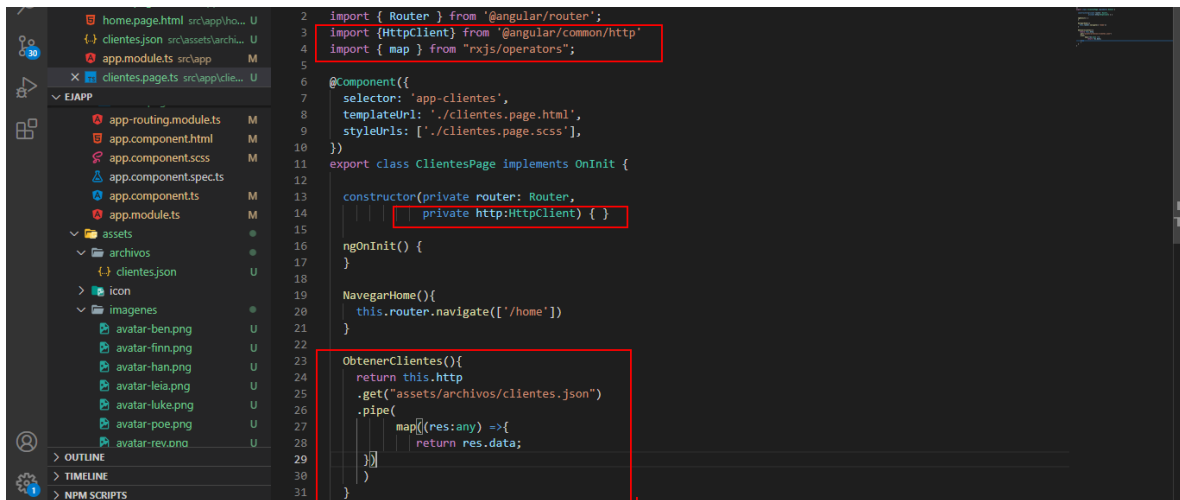
<https://ionicframework.com/docs/native/http>



Luego de instalar el plugin vamos al archivo `app.modules.ts` e importamos el `http`.



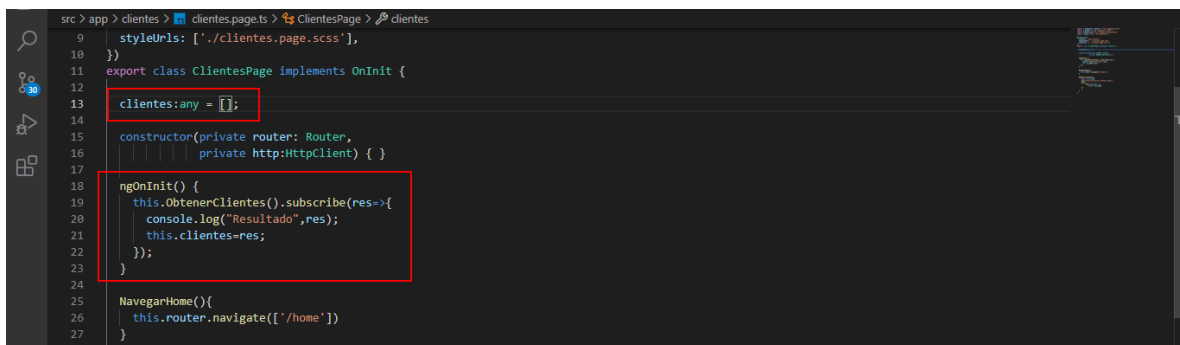
En el archivo `clientes.page.ts` agregamos las importaciones, el código en el constructor y una función `ObtenerClientes`.



Para probar programamos en el archivo **clientes.page.ts** y codificamos dentro de ngOnInit para verificar que muestra los datos en la consola.



Creamos una variable e igualamos el this.clientes=res



En el código en clientes.page.html agregamos el *ngFor para recorrer los datos y mostramos con {{cliente.nombre}}

```
<ion-header>
<ion-toolbar color="defcolor">
```

```
<ion-title>clientes</ion-title>
</ion-toolbar>
</ion-header>
<ion-content>
  <ion-list>
    <ion-item *ngFor="let cliente of clientes">
      <ion-avatar slot="end">
        
      </ion-avatar>
      <ion-label>
        <h2 class="titulo">{{cliente.nombre}}</h2>
        <h3>{{cliente.texto}}</h3>
        <p>{{cliente.descripcion}}</p>
      </ion-label>
    </ion-item>
  </ion-list>
</ion-content>
```

CONDICION ngIf

Para utilizar condiciones agregaremos estado en nuestro archive json

```
{
  "data" : [
    {
      "nombre": "Finn",
      "texto": "I'm a big deal",
      "descripcion": "Listen, I've had a pretty messed up day",
      "foto": "avatar-finn.png",
      "estado": true
    },
    {
      "nombre": "Han",
      "texto": "Look, kid",
      "descripcion": "I've got enough on my plate as it is, and I",
      "foto": "avatar-han.png",
      "estado": true
    },
    {
      "nombre": "Luke",
```



```
        "texto": "Your thoughts betray you",
        "descripcion": "I feel the good in you, the conflict",
        "foto": "avatar-luke.png",
        "estado": false
    }
  ]
}
```

En nuestro archivo clientes.page.html

```
<ion-header>
  <ion-toolbar color="defcolor">
    <ion-title>clientes</ion-title>
  </ion-toolbar>
</ion-header>
<ion-content>
  <ion-list>
    <ion-item *ngFor="let cliente of clientes">
      <ion-avatar slot="end">
        
      </ion-avatar>
      <ion-label>
        <h2 class="titulo">{{cliente.nombre}}</h2>
        <h3>{{cliente.texto}}</h3>
        <p>{{cliente.descripcion}}</p>
      </ion-label>
      <ion-label *ngIf=cliente.estado>
        Activo
      </ion-label>
      <ion-label *ngIf=!cliente.estado>
        Desactivo
      </ion-label>
    </ion-item>
  </ion-list>
</ion-content>
```

```
    </ion-item>
  </ion-list>

</ion-content>
```

Otra forma

```
<ion-header>
  <ion-toolbar color="defcolor">
    <ion-title>clientes</ion-title>
  </ion-toolbar>
</ion-header>
<ion-content>
  <ion-list>
    <ion-item *ngFor="let cliente of clientes">
      <ion-avatar slot="end">
        
      </ion-avatar>
      <ion-label>
        <h2 class="titulo">{{cliente.nombre}}</h2>
        <h3>{{cliente.texto}}</h3>
        <p>{{cliente.descripcion}}</p>
      </ion-label>
      <ion-label *ngIf="cliente.estado; else desac">
        Activo
      </ion-label>
      <ng-template #desac><ion-label>Desactivado</ion-label></ng-template>
    </ion-item>
  </ion-list>
</ion-content>
```