

# Sass

Los preprocesadores CSS son herramientas para los desarrolladores de sitios web, que permiten traducir un código de hojas de estilo no estándar, específico del preprocesador en cuestión, a un código CSS estándar, entendible por los navegadores.

Los preprocesadores básicamente nos ofrecen diversas utilidades que a día de hoy no se encuentran en el lenguaje CSS, o bien no son compatibles con todos los navegadores. Ejemplos pueden ser variables, anidación de selectores, funciones (denominadas mixins), etc.

Al desarrollar con un preprocesador se consigue principalmente un ahorro de tiempo, ya que tenemos que escribir menos código para hacer las cosas. Pero también conseguimos una mayor facilidad de mantenimiento del código, gracias a una mejor organización del código y la posibilidad de editar una vez ciertos valores y que afecten a decenas, o cientos, de lugares del código CSS generado.

# Sass

## ¿Qué es sass?

Lo primero que debes conocer de Sass es que existen dos tipos de sintaxis para escribir su código:

Sintaxis Sass: esta sintaxis es un poco diferente de la sintaxis de CSS estándar. No difiere mucho. Por ejemplo, te evita colocar puntos y coma al final de los valores de propiedades. Además, las llaves no se usan y en su lugar se realizan indentados.

Sintaxis SCSS: Es una sintaxis bastante similar a la sintaxis del propio CSS. De hecho, el código CSS es código SCSS válido. Podríamos decir que SCSS es código CSS con algunas cosas extra.

# Sass

Crear carpeta: sass

Cambiarse a la carpeta: cd sass

Crear las siguientes carpetas dentro de sass:

css

sass

src

npm init -y                      genera un archivo json

npm install -g node-sass

Instalar:

Live Server

Live Sass Compiler

Agregar el archivo index.html

# Sass

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link rel="stylesheet" href="./index.css"
</head>
<body>
  <p>
    Esta es una linea de prueba
  </p>
  <div class="container">
    <nav>
      <ul>
        <li>CSS</li>
        <li>HTML</li>
        <li>CSS</li>
        <li>Angular</li>
      </ul>
      <a href="#">Hola</a>
    </nav>
  </div>
  <div class="transform">

  </div>
  <div class="element">

  </div>
</body>
</html>
```

# Sass

Agregar los archivos  
index.sass  
index.scss

# Ionic

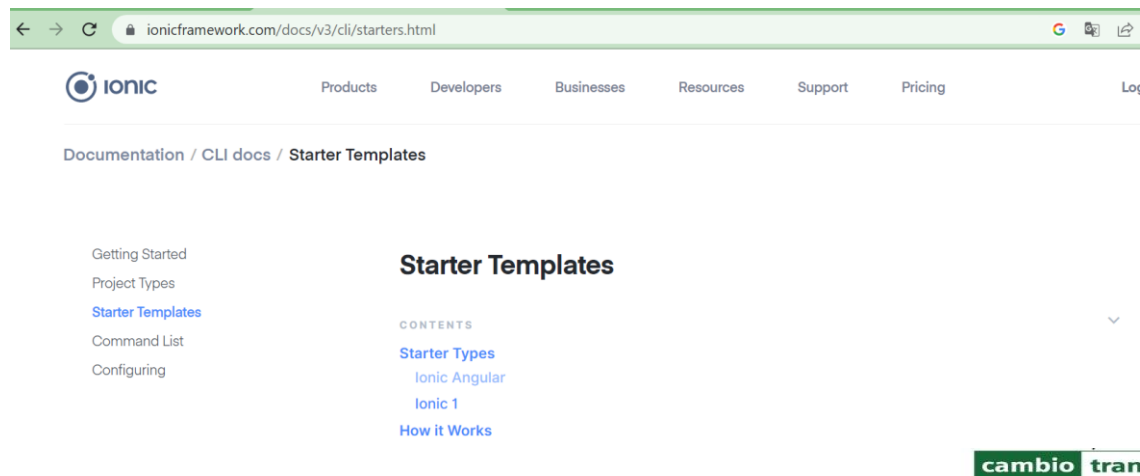
**Ionic** Framework es un SDK **de** frontend **de** código abierto **para** desarrollar aplicaciones híbridas basado en tecnologías web (HTML, CSS y JS). Es decir, un framework que nos permite desarrollar aplicaciones **para** iOS nativo, **Android** y la web, desde una única base **de** código. La plataforma **Ionic** proporciona herramientas y servicios para el desarrollo **de** aplicaciones móviles híbridas, aplicaciones **de** escritorio y PWA (Aplicaciones Web Progresivas) basadas en las más modernas tecnologías y prácticas **de** desarrollo web

# Ionic

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

```
ionic -v
```

```
ionic start templates
```



VIVE LA FIME

# Ionic

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

```
ionic -v
```

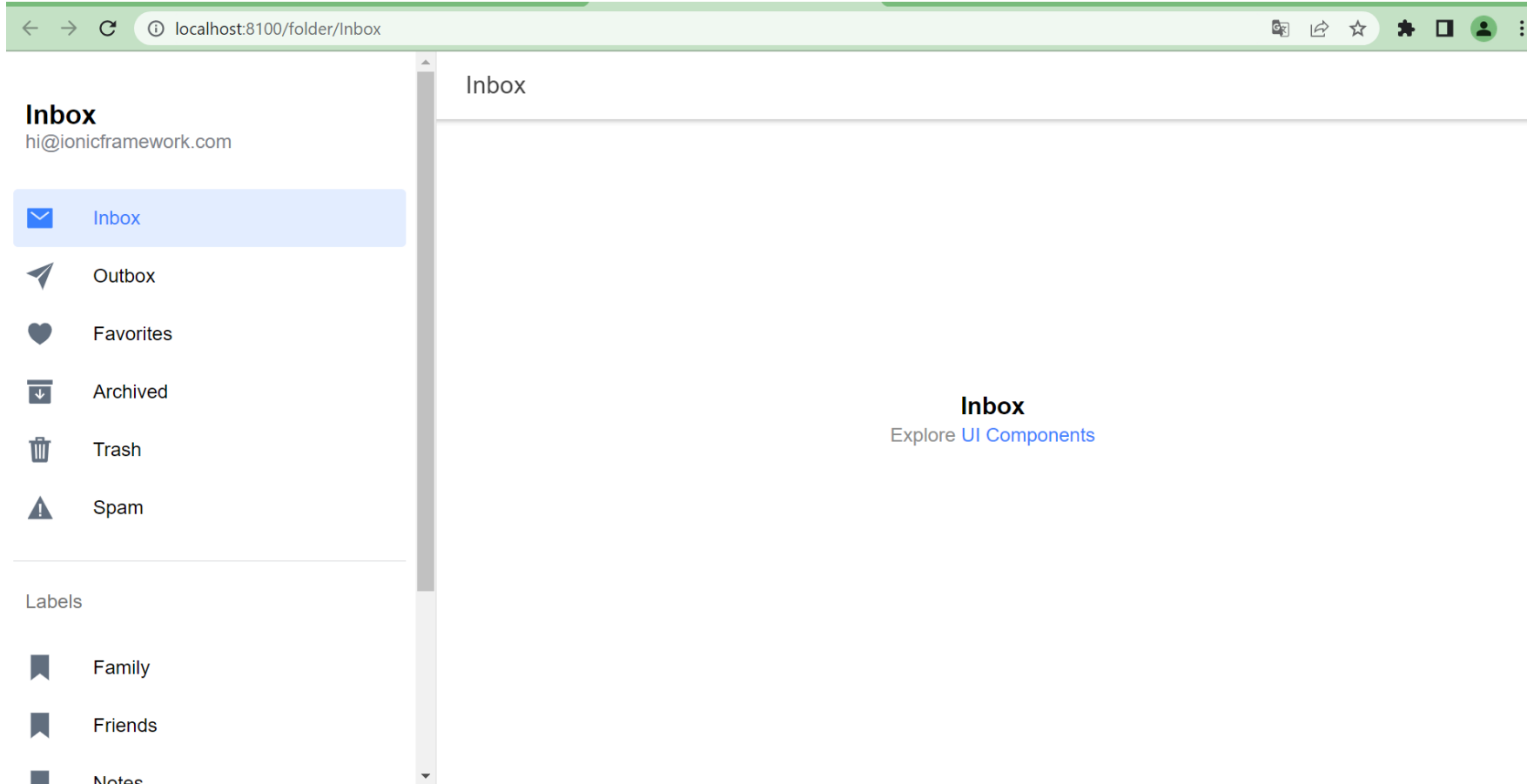
```
ionic start myTestApp sidemenu
```

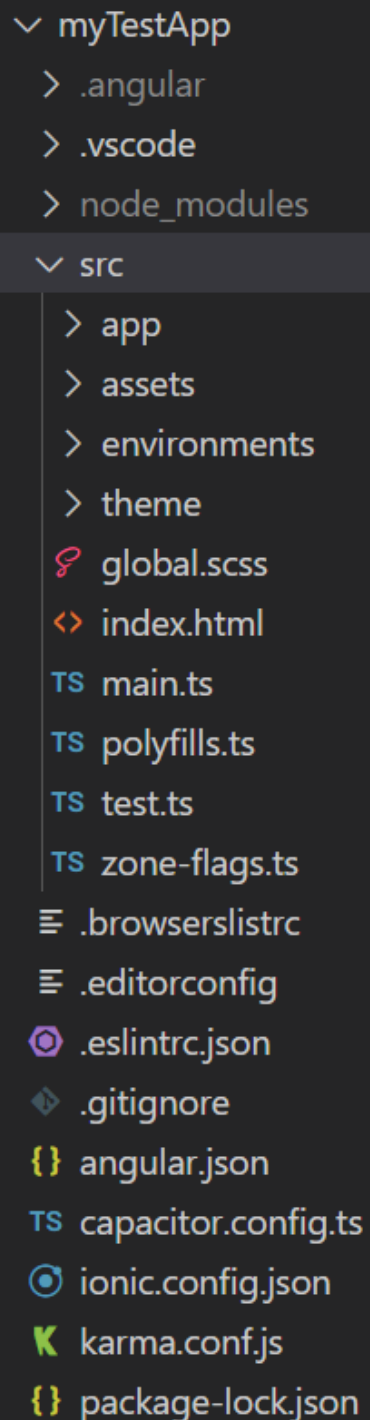
```
cd myTestApp
```

```
ionic serve
```



# Ionic





File explorer view of an Ionic project:

- myTestApp
  - .angular
  - .vscode
  - node\_modules
- src
  - app
  - assets
  - environments
  - theme
  - global.scss
  - index.html
  - main.ts
  - polyfills.ts
  - test.ts
  - zone-flags.ts
- .browserslistrc
- .editorconfig
- .eslinttrc.json
- .gitignore
- angular.json
- capacitor.config.ts
- ionic.config.json
- karma.conf.js
- package-lock.json

# Ionic

**node\_modules**: módulos preinstalados al momento de crear la aplicación, es donde se van a agregar los nuevos módulos que se van instalando

**src**: área donde vamos a trabajar

**app**: área donde se agregan las páginas que se van a ir creando

**app-routing.module**: área donde se van a generar las rutas de las páginas

**app-component.html**: área donde se agregan los componentes de la aplicación

# Ionic

**app.component.scss**: área de estilos de la aplicación

**app.component.ts**: archivo de typescript de la aplicación

**app.module.ts**: área donde se agregarán los nuevos módulos que se van utilizando, se tienen que importar

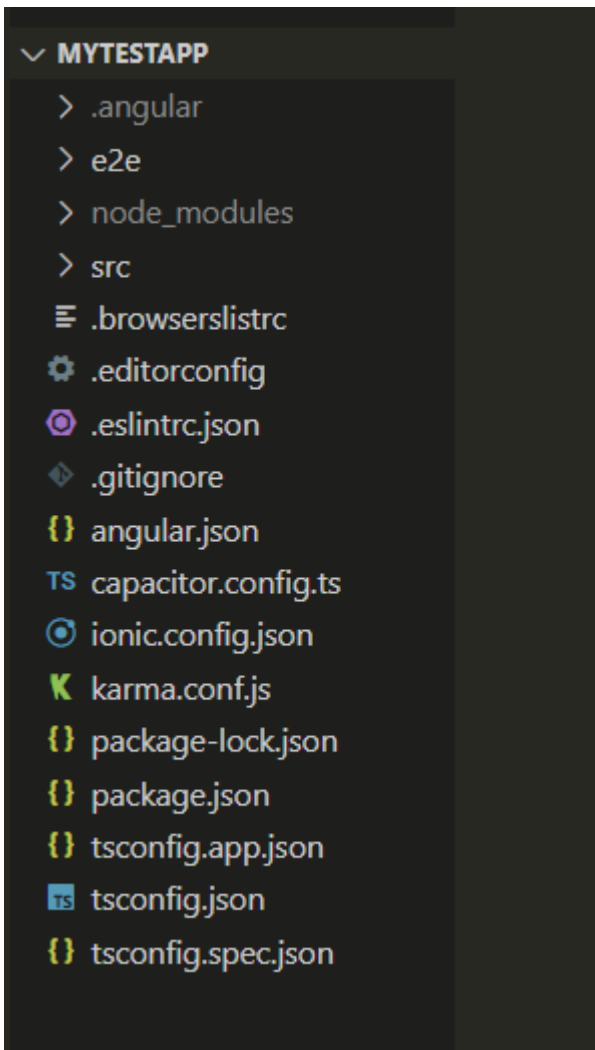
**assets**: es la carpeta de los recursos de la aplicación como iconos e imágenes

**enviroments**: se utiliza si queremos trabajar con distintos entornos

**theme**: contiene todas las variables de colores

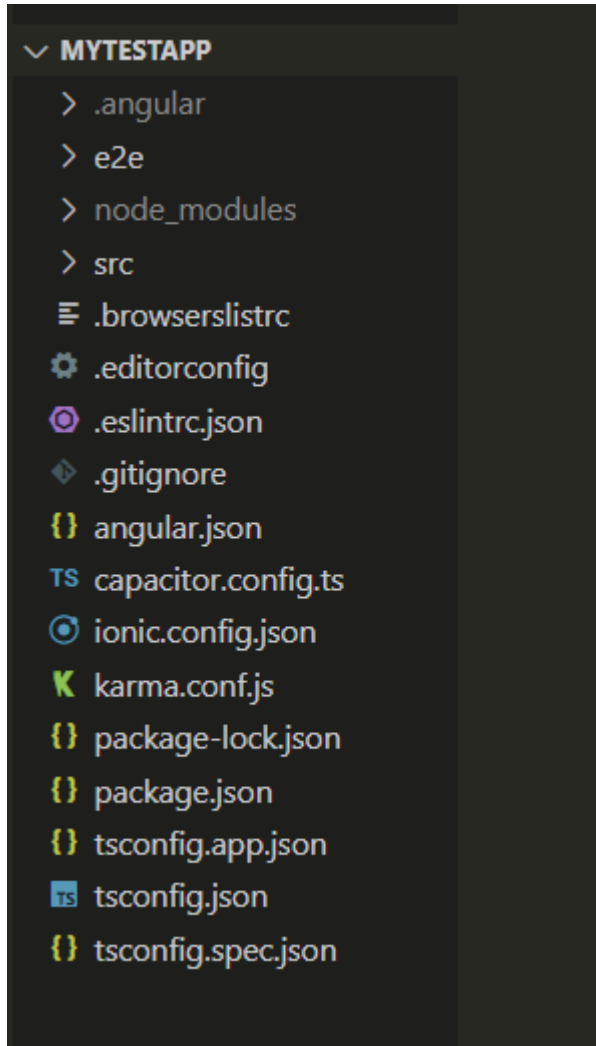
**global.scss**: contiene los estilos generales para todas las paginas

**index**: pagina web principal



# Ionic

**package-lock.json y package.json:** contiene las versiones de todo lo que se instala y sus dependencias dentro de la app



# Creacion de paginas

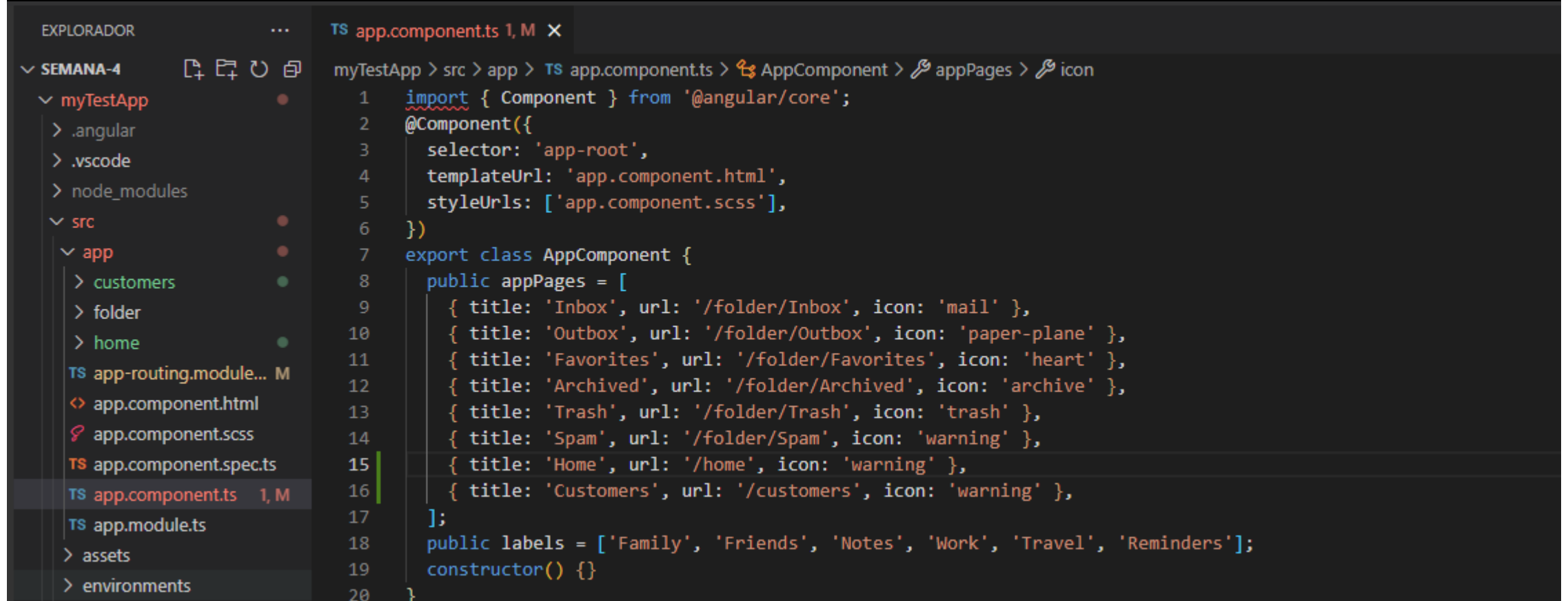
**ionic** generate page home

**ionic** generate page customer

```
EXPLORADOR
...
TS app-routing.module.ts 1, M
myTestApp > src > app > TS app-routing.module.ts > ...
1 import { NgModule } from '@angular/core';
2 import { PreloadAllModules, RouterModule, Routes } from '@angular/router';
3
4 const routes: Routes = [
5   {
6     path: '',
7     redirectTo: 'folder/Inbox',
8     pathMatch: 'full'
9   },
10  {
11    path: 'folder/:id',
12    loadChildren: () => import('./folder/folder.module').then( m => m.FolderPageModule)
13  },
14  {
15    path: 'home',
16    loadChildren: () => import('./home/home.module').then( m => m.HomePageModule)
17  },
18  {
19    path: 'customers',
20    loadChildren: () => import('./customers/customers.module').then( m => m.CustomersPageModule)
21  }
22 ]
23
24 @NgModule({
25   imports: [RouterModule.forRoot()],
26   declarations: [],
27   providers: [],
28   bootstrap: []
29 })
30 export class AppRoutingModule {}
```

# Ionic

## Modificar app.component.ts para agregar opciones al menu

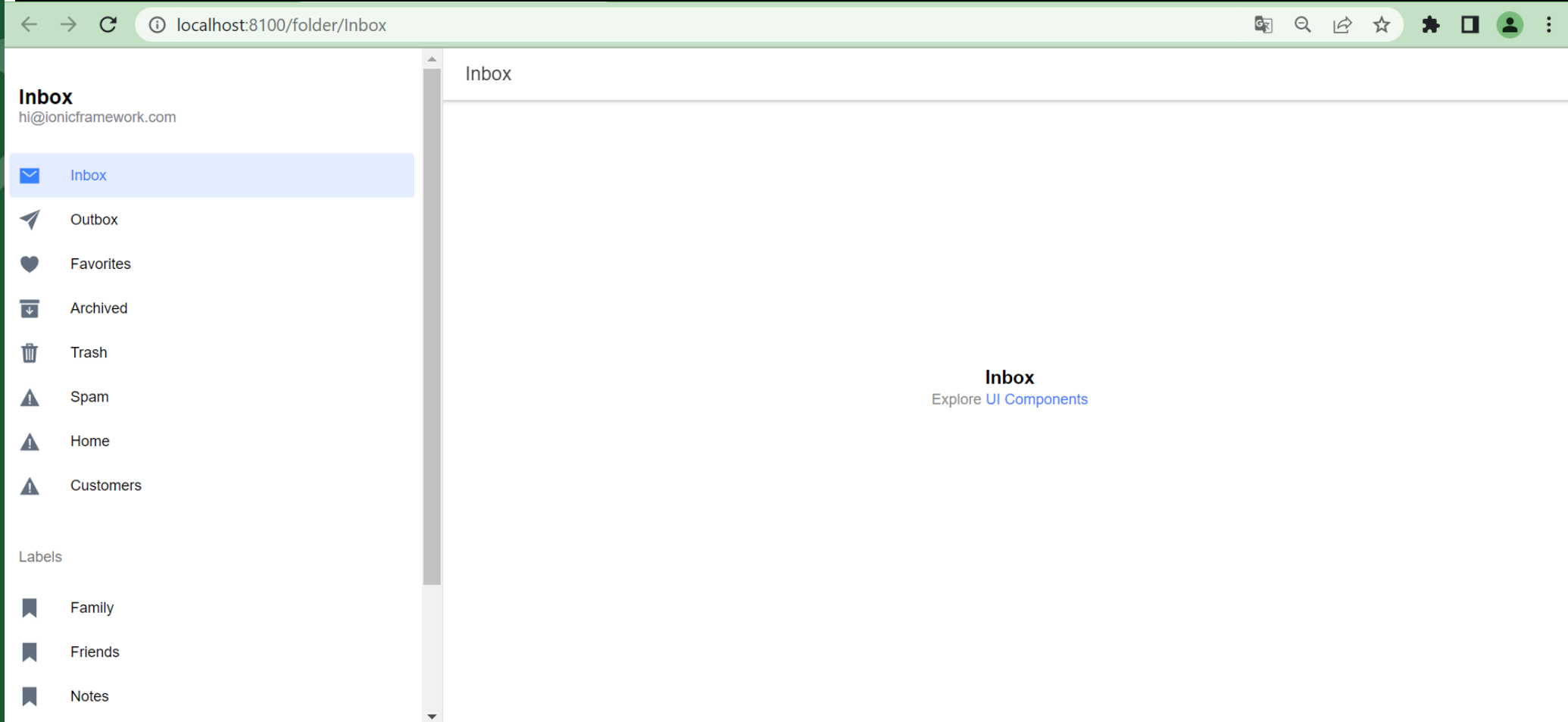


The screenshot shows the Visual Studio Code editor interface. On the left, the Explorer sidebar displays the project structure for 'SEMANA-4', including folders like 'myTestApp', 'src', and 'app'. The file 'app.component.ts' is selected. The main editor area shows the TypeScript code for the AppComponent class. The breadcrumb navigation at the top indicates the path: 'myTestApp > src > app > TS app.component.ts > AppComponent > appPages > icon'. The code defines the AppComponent class with a selector 'app-root', templateUrl 'app.component.html', and styleUrls ['app.component.scss']. The appPages array lists several pages with their titles, URLs, and icons. The labels array contains 'Family', 'Friends', 'Notes', 'Work', 'Travel', and 'Reminders'. The constructor is currently empty.

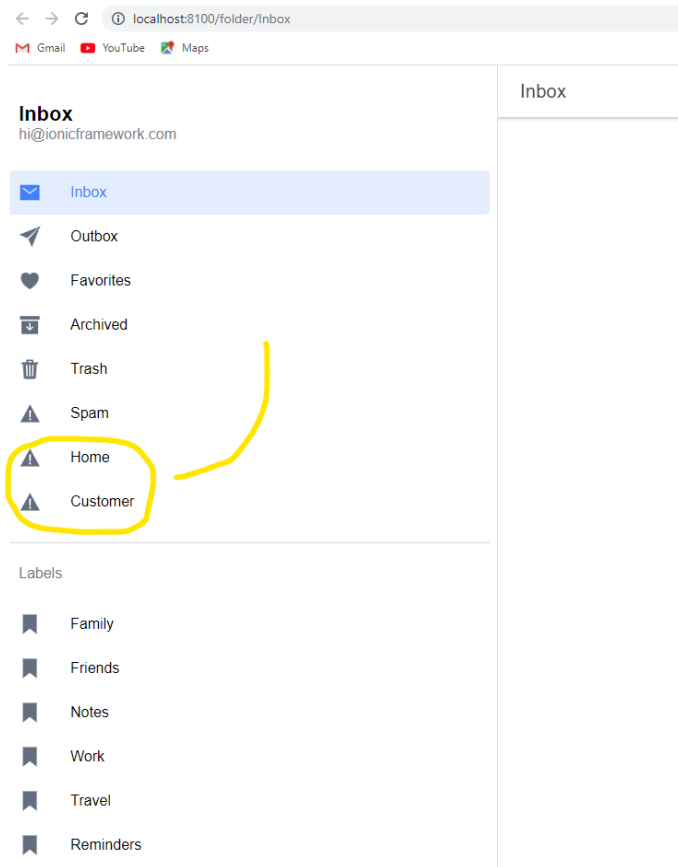
```
1 import { Component } from '@angular/core';
2 @Component({
3   selector: 'app-root',
4   templateUrl: 'app.component.html',
5   styleUrls: ['app.component.scss'],
6 })
7 export class AppComponent {
8   public appPages = [
9     { title: 'Inbox', url: '/folder/Inbox', icon: 'mail' },
10    { title: 'Outbox', url: '/folder/Outbox', icon: 'paper-plane' },
11    { title: 'Favorites', url: '/folder/Favorites', icon: 'heart' },
12    { title: 'Archived', url: '/folder/Archived', icon: 'archive' },
13    { title: 'Trash', url: '/folder/Trash', icon: 'trash' },
14    { title: 'Spam', url: '/folder/Spam', icon: 'warning' },
15    { title: 'Home', url: '/home', icon: 'warning' },
16    { title: 'Customers', url: '/customers', icon: 'warning' },
17  ];
18  public labels = ['Family', 'Friends', 'Notes', 'Work', 'Travel', 'Reminders'];
19  constructor() {}
20 }
```

# Ionic

Modificar app.component.ts para agregar opciones al menu



# Ionic



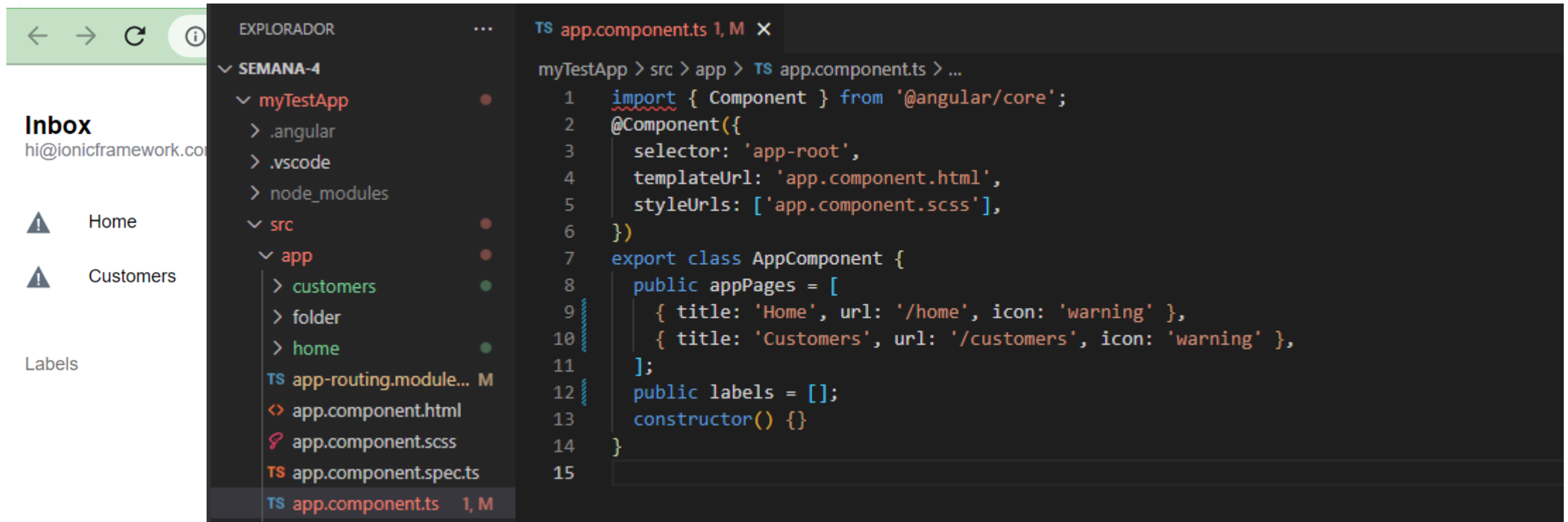
**unidos**  
*por un*  
**cambio transformador**

**VIVE LA FIME**



# Ionic

Modificar app.component.ts para dejar solo las paginas creadas

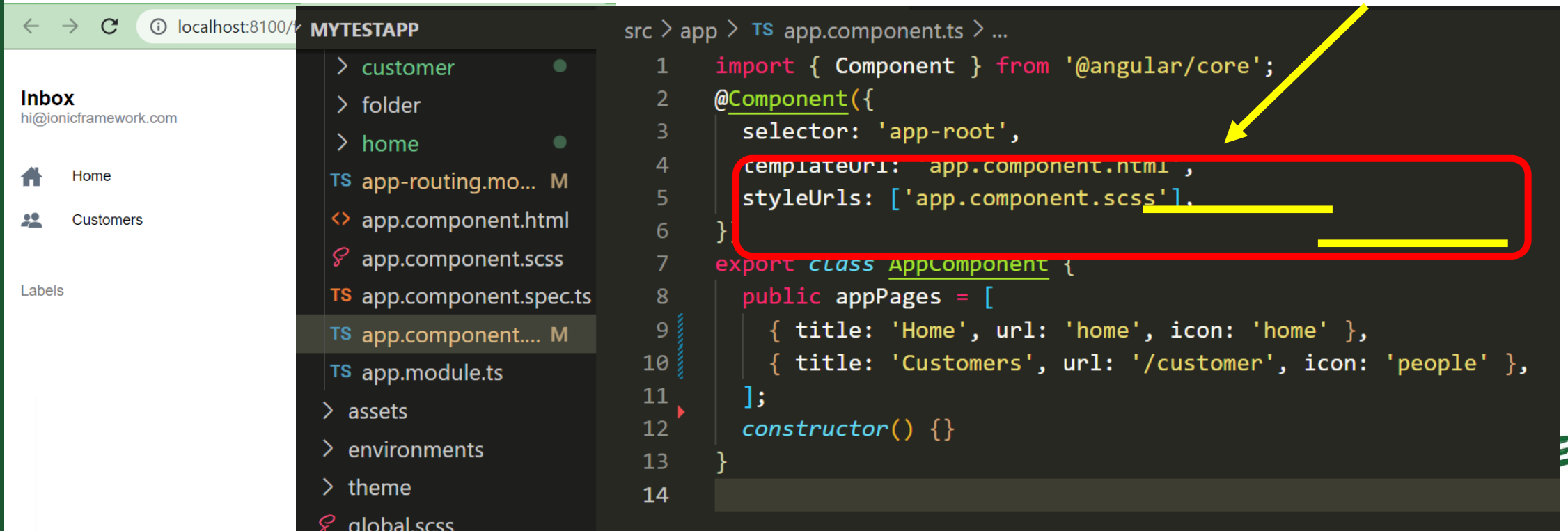


The screenshot shows the Visual Studio Code interface with the Explorer, Search, and Run and Debug sidebar on the left. The Explorer sidebar shows a project structure with a folder named 'myTestApp' containing subfolders '.angular', '.vscode', and 'node\_modules'. Inside 'myTestApp' is a folder 'src' containing a folder 'app'. The 'app' folder contains files 'customers', 'folder', 'home', 'app-routing.module.ts', 'app.component.html', 'app.component.scss', 'app.component.spec.ts', and 'app.component.ts'. The 'app.component.ts' file is selected and its content is displayed in the main editor area. The code in 'app.component.ts' is as follows:

```
1 import { Component } from '@angular/core';
2 @Component({
3   selector: 'app-root',
4   templateUrl: 'app.component.html',
5   styleUrls: ['app.component.scss'],
6 })
7 export class AppComponent {
8   public appPages = [
9     { title: 'Home', url: '/home', icon: 'warning' },
10    { title: 'Customers', url: '/customers', icon: 'warning' },
11  ];
12  public labels = [];
13  constructor() {}
14 }
15
```

# Ionic

Cambiar iconos: [ionicons.com](https://ionicons.com)



The image shows a web browser on the left and a code editor on the right. The browser displays an Ionic application with a sidebar menu. The code editor shows the `src > app > TS app.component.ts` file. A red rectangle highlights the `templateUrl` and `styleUrls` properties in the `@Component` decorator. A yellow arrow points from the top right towards the `templateUrl` property.

**Browser View:**

- Address bar: `localhost:8100/`
- Page Title: **MYTESTAPP**
- Content: **Inbox** (hi@ionicframework.com)
- Menu Items: Home, Customers
- Labels: (empty)

**Code Editor View:**

```
src > app > TS app.component.ts > ...
1  import { Component } from '@angular/core';
2  @Component({
3    selector: 'app-root',
4    templateUrl: 'app.component.html',
5    styleUrls: ['app.component.scss'],
6  })
7  export class AppComponent {
8    public appPages = [
9      { title: 'Home', url: 'home', icon: 'home' },
10     { title: 'Customers', url: '/customer', icon: 'people' },
11   ];
12   constructor() {}
13 }
14
```

# Ionic

## Modificar app.component.html para quitar Textos

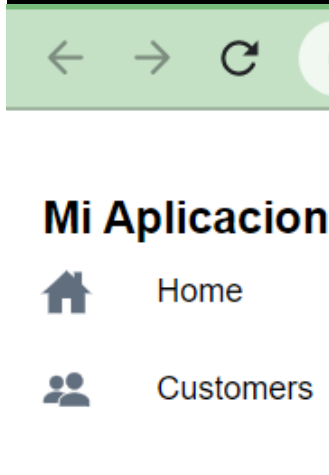
```
<ion-app>
  <ion-split-pane contentId="main-content">
    <ion-menu contentId="main-content" type="overlay">
      <ion-content>
        <ion-list id="inbox-list">
          <ion-list-header>Inbox</ion-list-header>
          <ion-note>hi@ionicframework.com</ion-note>
          <ion-menu-toggle auto-hide="false" *ngFor="let p of appPages; let i = index">
            <ion-item routerDirection="root" [routerLink]="[p.url]" lines="none" detail="false" routerLinkActive="selected">
              <ion-icon slot="start" [ios]="p.icon + '-outline'" [md]="p.icon + '-sharp'"></ion-icon>
              <ion-label>{{ p.title }}</ion-label>
            </ion-item>
          </ion-menu-toggle>
        </ion-list>

        <ion-list id="labels-list">
          <ion-list-header>Labels</ion-list-header>

          <ion-item *ngFor="let label of labels" lines="none">
            <ion-icon slot="start" ios="bookmark-outline" md="bookmark-sharp"></ion-icon>
            <ion-label>{{ label }}</ion-label>
          </ion-item>
        </ion-list>
      </ion-content>
    </ion-menu>
    <ion-router-outlet id="main-content"></ion-router-outlet>
  </ion-split-pane>
</ion-app>
```

# Ionic

## Modificar app.component.html para quitar Textos



```
<ion-app>
  <ion-split-pane contentId="main-content">
    <ion-menu contentId="main-content" type="overlay">
      <ion-content>
        <ion-list id="inbox-list">
          <ion-list-header>Mi Aplicacion</ion-list-header>
          <ion-menu-toggle auto-hide="false" *ngFor="let p of appPages; let i = index">
            <ion-item routerDirection="root" [routerLink]="[p.url]" lines="none" detail="false" routerLinkActive="selected">
              <ion-icon slot="start" [ios]="p.icon + '-outline'" [md]="p.icon + '-sharp'"></ion-icon>
              <ion-label>{{ p.title }}</ion-label>
            </ion-item>
          </ion-menu-toggle>
        </ion-list>
      </ion-content>
    </ion-menu>
    <ion-router-outlet id="main-content"></ion-router-outlet>
  </ion-split-pane>
</ion-app>
```

# Ionic

<https://ionicframework.com/docs/theming/color-generator>

## Modificar Estilos y Colores: crear variable de colores en variables.scss

### New Color Creator

Create a new color below by changing the name and value, then copy and paste the code below into your project.



mycolor

```
:root {  
  --ion-color-mycolor: #177d19;  
  --ion-color-mycolor-rgb: 23,125,25;  
  --ion-color-mycolor-contrast: #ffffff;  
  --ion-color-mycolor-contrast-rgb: 255,255,255;  
  --ion-color-mycolor-shade: #146e16;  
  --ion-color-mycolor-tint: #2e8a30;  
}
```

Layered Color:

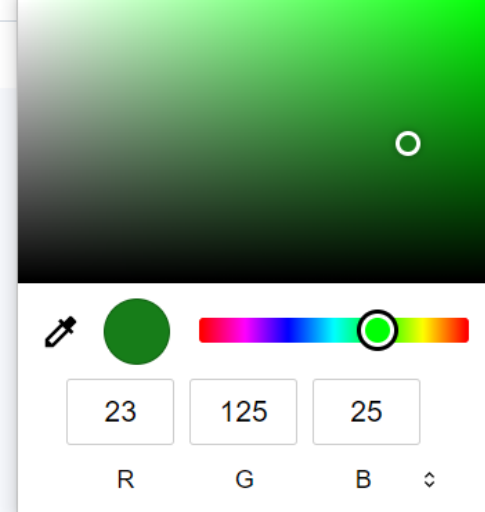
Modifying Col

Adding Colors

[New Color Cre](#)

[Edit this p](#)

#177d19



por un  
cambio transformador

VIVE LA FIME

# Ionic

<https://ionicframework.com/docs/theming/color-generator>

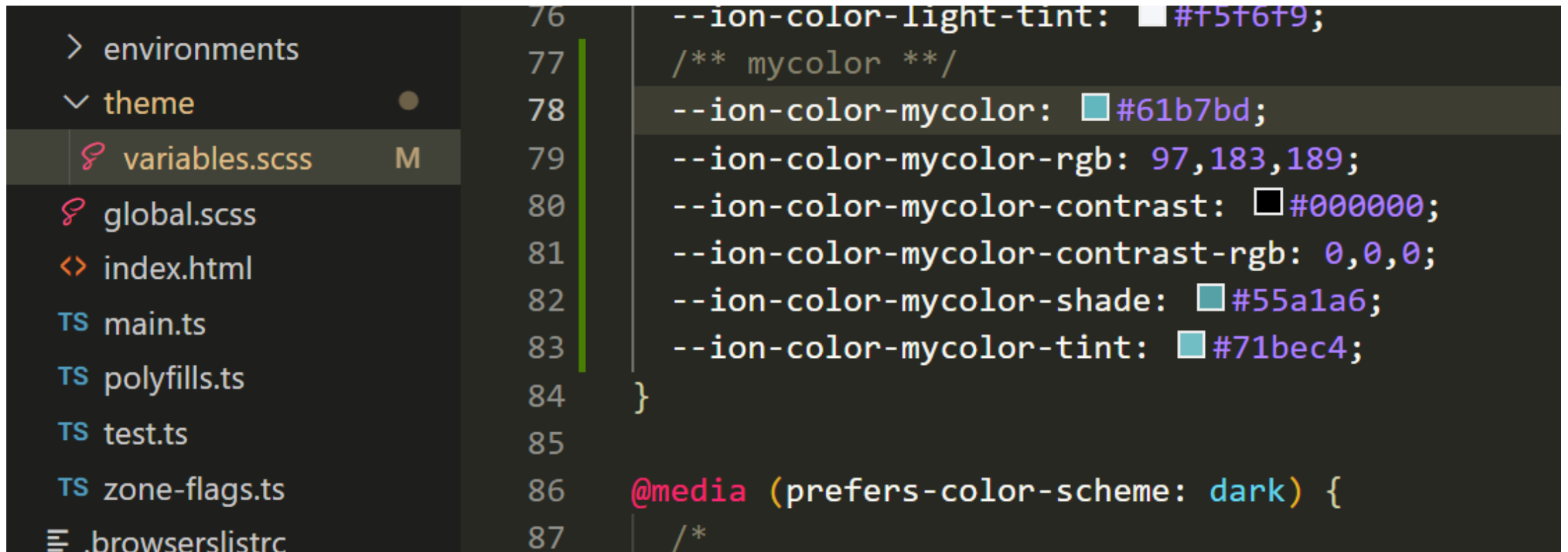
## Modificar Estilos y Colores: crear variable de colores en variables.scss

- ✓ app
  - > customers
  - > folder
  - > home
  - TS app-routing.module... M
  - <> app.component.html M
  - 🔗 app.component.scss
  - TS app.component.spec.ts
  - TS app.component.ts M
  - TS app.module.ts
  - > assets
  - > environments
- ✓ theme
  - 🔗 variables.scss

```
/** light **/  
--ion-color-light: #f4f5f8;  
--ion-color-light-rgb: 244, 245, 248;  
--ion-color-light-contrast: #000000;  
--ion-color-light-contrast-rgb: 0, 0, 0;  
--ion-color-light-shade: #d7d8da;  
--ion-color-light-tint: #f5f6f9;  
  
/** mycolor **/  
--ion-color-mycolor: #177d19;  
--ion-color-mycolor-rgb: 23,125,25;  
--ion-color-mycolor-contrast: #ffffff;  
--ion-color-mycolor-contrast-rgb: 255,255,255;  
--ion-color-mycolor-shade: #146e16;  
--ion-color-mycolor-tint: #2e8a30;  
}
```

# Ionic

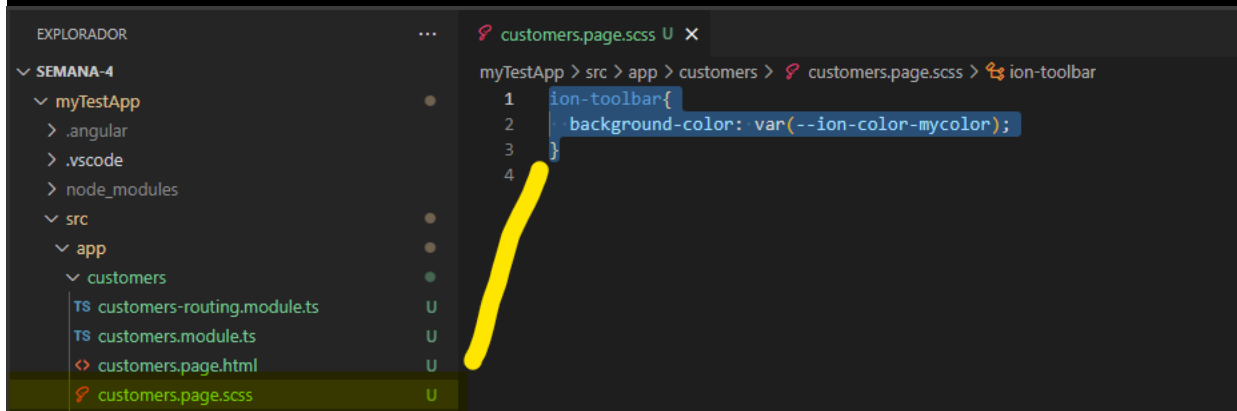
Editar customer.page.html y



```
76 --ion-color-light-tint: #f5f6f9;
77 /** mycolor **/
78 --ion-color-mycolor: #61b7bd;
79 --ion-color-mycolor-rgb: 97,183,189;
80 --ion-color-mycolor-contrast: #000000;
81 --ion-color-mycolor-contrast-rgb: 0,0,0;
82 --ion-color-mycolor-shade: #55a1a6;
83 --ion-color-mycolor-tint: #71bec4;
84 }
85
86 @media (prefers-color-scheme: dark) {
87 /*
```

# Ionic

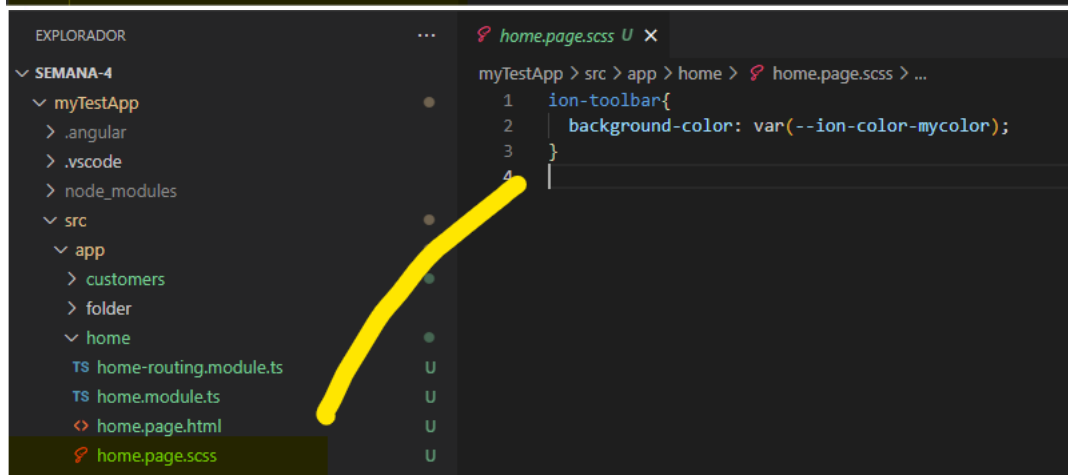
## Editar customer.page.html y customer.page.scss



The screenshot shows the VS Code interface. On the left, the Explorer sidebar is open, showing a project structure under 'SEMANA-4' with a subfolder 'myTestApp'. Inside 'myTestApp', there are folders '.angular', '.vscode', and 'node\_modules', and a 'src' folder. The 'src' folder contains an 'app' folder, which in turn contains a 'customers' folder. The 'customers' folder contains three files: 'customers-routing.module.ts', 'customers.module.ts', and 'customers.page.html'. The 'customers.page.scss' file is selected and highlighted in the Explorer. The main editor area shows the 'customers.page.scss' file with the following content:

```
1 ion-toolbar{  
2   background-color: var(--ion-color-mycolor);  
3 }  
4
```

A yellow arrow points from the 'customers.page.scss' file in the Explorer to the editor window.



The screenshot shows the VS Code interface. On the left, the Explorer sidebar is open, showing a project structure under 'SEMANA-4' with a subfolder 'myTestApp'. Inside 'myTestApp', there are folders '.angular', '.vscode', and 'node\_modules', and a 'src' folder. The 'src' folder contains an 'app' folder, which in turn contains a 'home' folder. The 'home' folder contains three files: 'home-routing.module.ts', 'home.module.ts', and 'home.page.html'. The 'home.page.scss' file is selected and highlighted in the Explorer. The main editor area shows the 'home.page.scss' file with the following content:

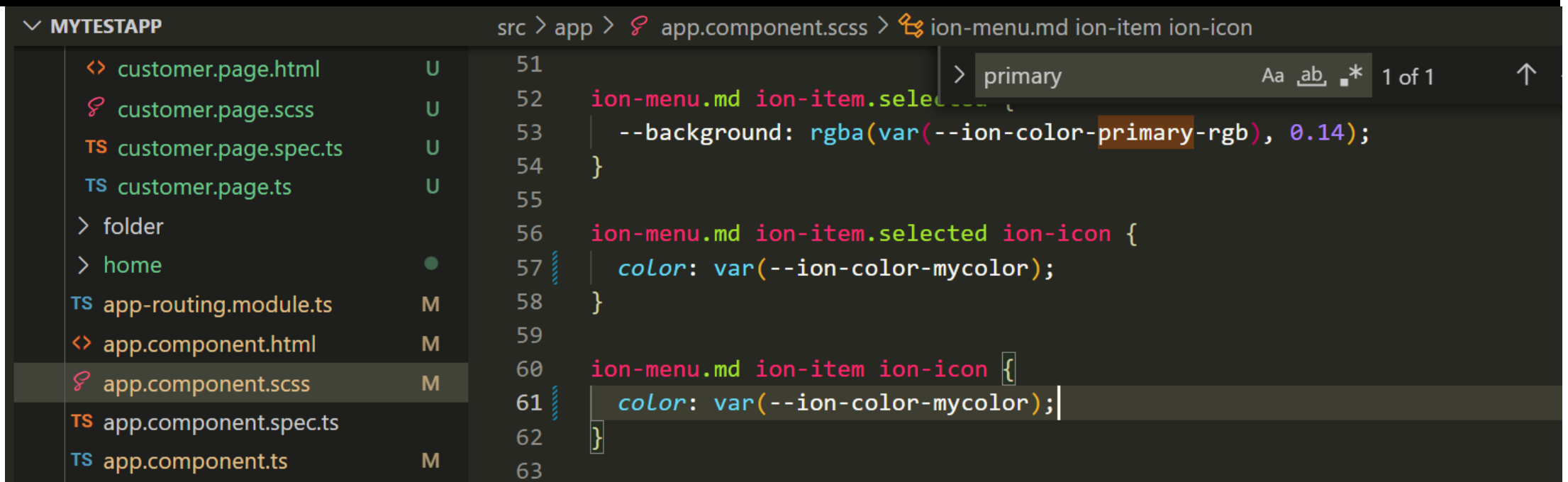
```
1 ion-toolbar{  
2   background-color: var(--ion-color-mycolor);  
3 }  
4
```

A yellow arrow points from the 'home.page.scss' file in the Explorer to the editor window.



# Ionic

## Cambiar color de texto y color de iconos, modificar app.components.scss



```
MYTESTAPP src > app > app.component.scss > ion-menu.md ion-item ion-icon
customer.page.html U 51
customer.page.scss U 52
customer.page.spec.ts U 53
customer.page.ts U 54
folder 55
home 56
app-routing.module.ts M 57
app.component.html M 58
app.component.scss M 59
app.component.spec.ts 60
app.component.ts M 61
62
63
```

```
ion-menu.md ion-item.selected {
  --background: rgba(var(--ion-color-primary-rgb), 0.14);
}

ion-menu.md ion-item.selected ion-icon {
  color: var(--ion-color-mycolor);
}

ion-menu.md ion-item ion-icon {
  color: var(--ion-color-mycolor);
}
```

primary Aa .ab \* 1 of 1 ↑

# Ionic

Cambiar color de texto y color de iconos, modificar `app.components.scss`



VIVE LA FIME

# Ionic

## Cambiar color background de menu

```
ion-menu.md ion-item.selected {  
  --background: rgba(var(--ion-color-primary-rgb), 0.14);  
}
```



```
ion-menu.md ion-item.selected {  
  --background: rgba(var(--ion-color-mycolor-rgb), 0.14);  
}
```

# Ionic

```
ion-menu.md ion-item.selected ion-icon {  
  color: var(--ion-color-primary);  
}
```



```
ion-menu.md ion-item.selected ion-icon {  
  color: var(--ion-color-mycolor);  
}
```

**Cambiar color iconos de menu**

```
ion-item.selected {  
  --color: var(--ion-color-primary);  
}
```

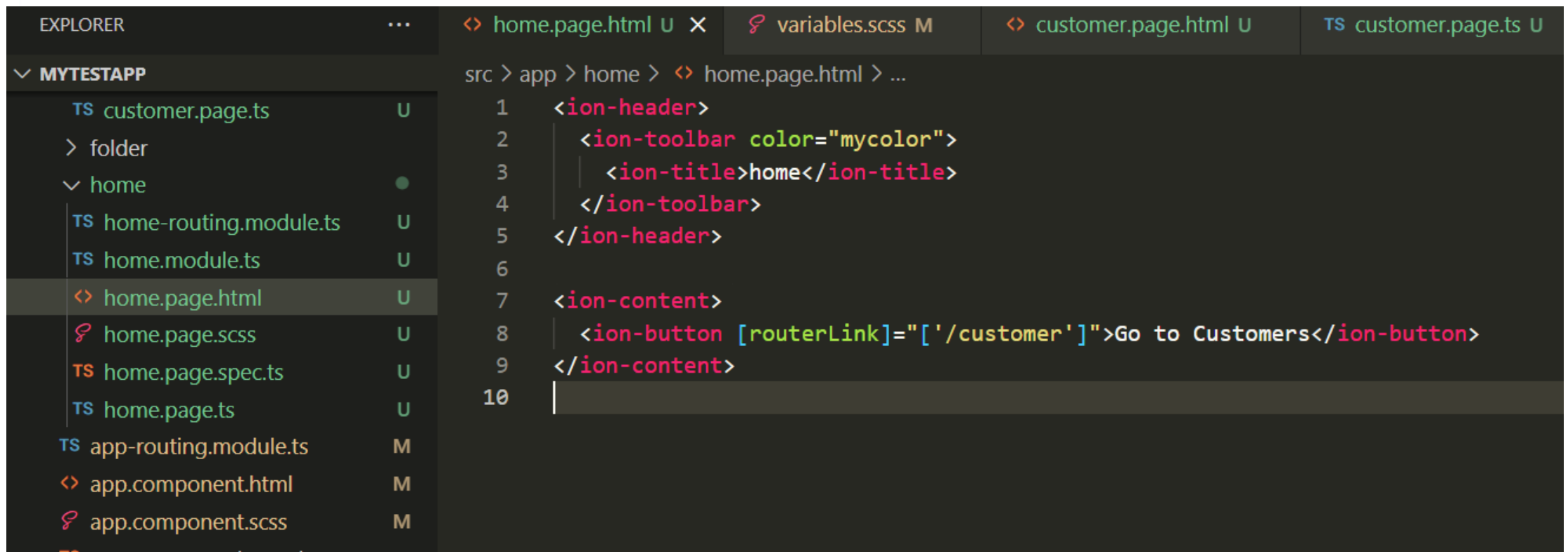


```
ion-item.selected {  
  --color: var(--ion-color-mycolor);  
}
```

**Cambiar color texto de menu**

# Ionic

## Navegación entre páginas: actualización a través de HTML



EXPLORER

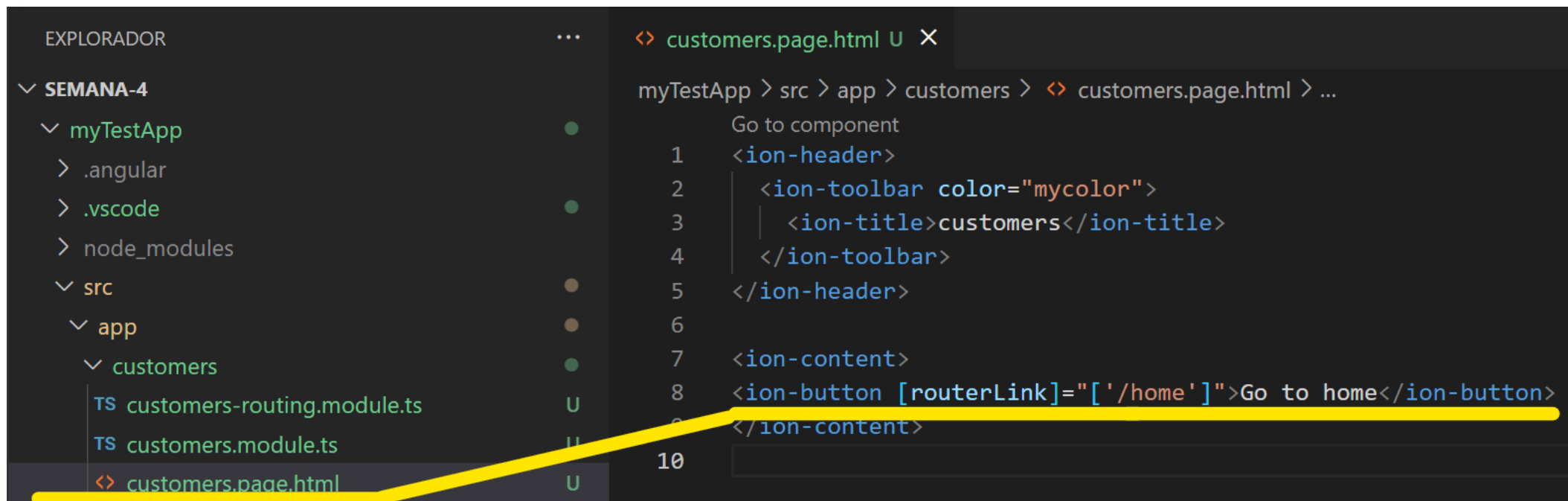
- MYTESTAPP
  - customer.page.ts U
  - folder
  - home
    - home-routing.module.ts U
    - home.module.ts U
    - home.page.html U
    - home.page.scss U
    - home.page.spec.ts U
    - home.page.ts U
  - app-routing.module.ts M
  - app.component.html M
  - app.component.scss M

src > app > home > home.page.html > ...

```
1 <ion-header>
2   <ion-toolbar color="mycolor">
3     <ion-title>home</ion-title>
4   </ion-toolbar>
5 </ion-header>
6
7 <ion-content>
8   <ion-button [routerLink]="['/customer']">Go to Customers</ion-button>
9 </ion-content>
10
```

# Ionic

## Navegación entre páginas: actualización a través de HTML



EXPLORADOR

- SEMANA-4
  - myTestApp
    - .angular
    - .vscode
    - node\_modules
    - src
      - app
        - customers
          - customers-routing.module.ts
          - customers.module.ts
          - customers.page.html

customers.page.html

```
myTestApp > src > app > customers > customers.page.html > ...  
Go to component  
1 <ion-header>  
2   <ion-toolbar color="mycolor">  
3     <ion-title>customers</ion-title>  
4   </ion-toolbar>  
5 </ion-header>  
6  
7 <ion-content>  
8 <ion-button [routerLink]="['/home']">Go to home</ion-button>  
9 </ion-content>  
10
```

# Ionic

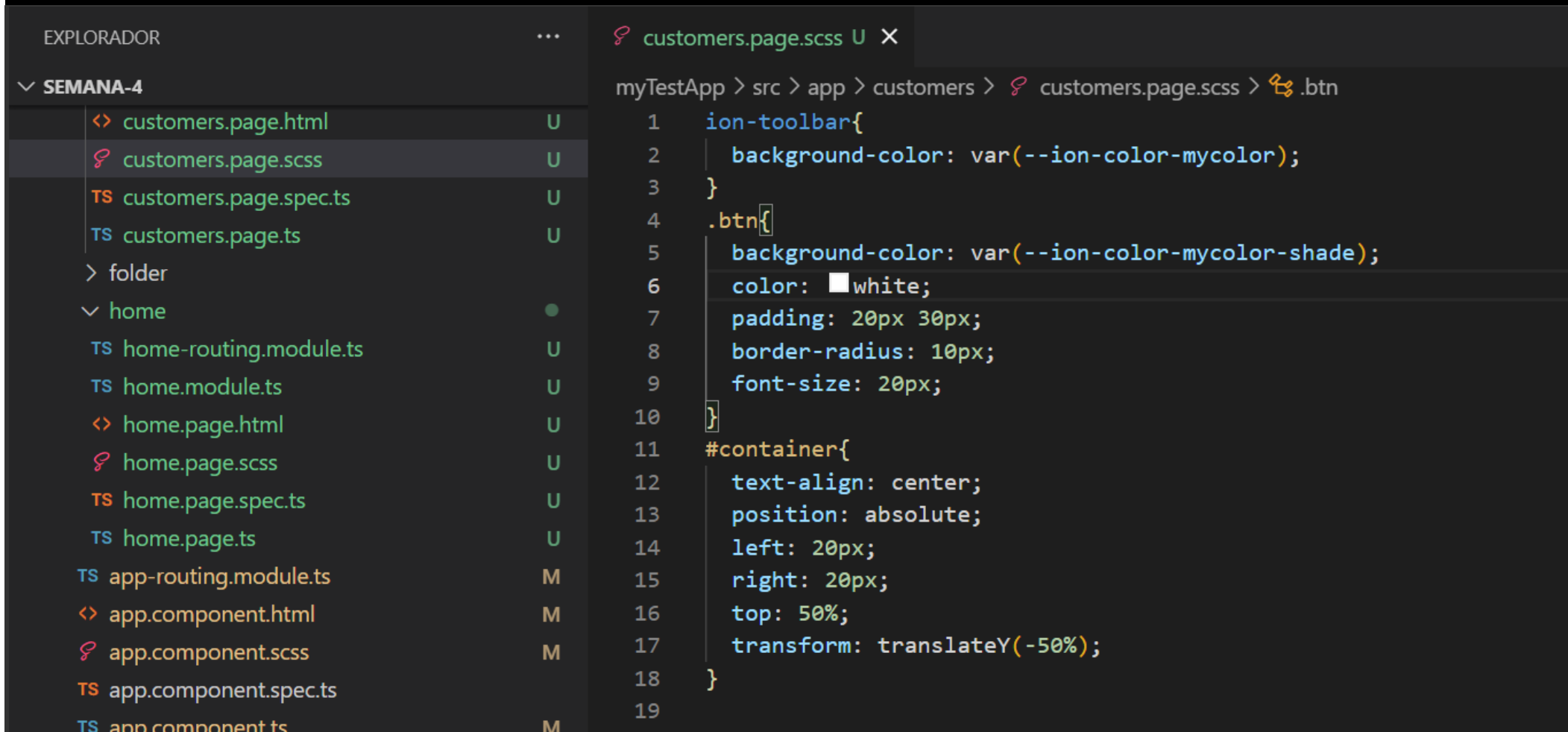
## Navegación entre páginas: Aplicación de estilos a botones customer.page.html

```
<ion-content>  
<ion-button [routerLink]="['/home']">Go to home</ion-button>  
</ion-content>
```

```
<ion-content>  
<button ion-button [routerLink]="['/home']" class="btn" >Go to home</button>  
</ion-content>
```

# Ionic

## Navegación entre páginas: Aplicación de estilos a botones customer.page.scss



The screenshot shows an IDE interface with a file explorer on the left and a code editor on the right. The file explorer, titled 'EXPLORADOR', shows a project structure under 'SEMANA-4'. The code editor, titled 'customers.page.scss', shows the following SCSS code:

```
1 ion-toolbar{
2   background-color: var(--ion-color-mycolor);
3 }
4 .btn{
5   background-color: var(--ion-color-mycolor-shade);
6   color: white;
7   padding: 20px 30px;
8   border-radius: 10px;
9   font-size: 20px;
10 }
11 #container{
12   text-align: center;
13   position: absolute;
14   left: 20px;
15   right: 20px;
16   top: 50%;
17   transform: translateY(-50%);
18 }
19
```

TIME

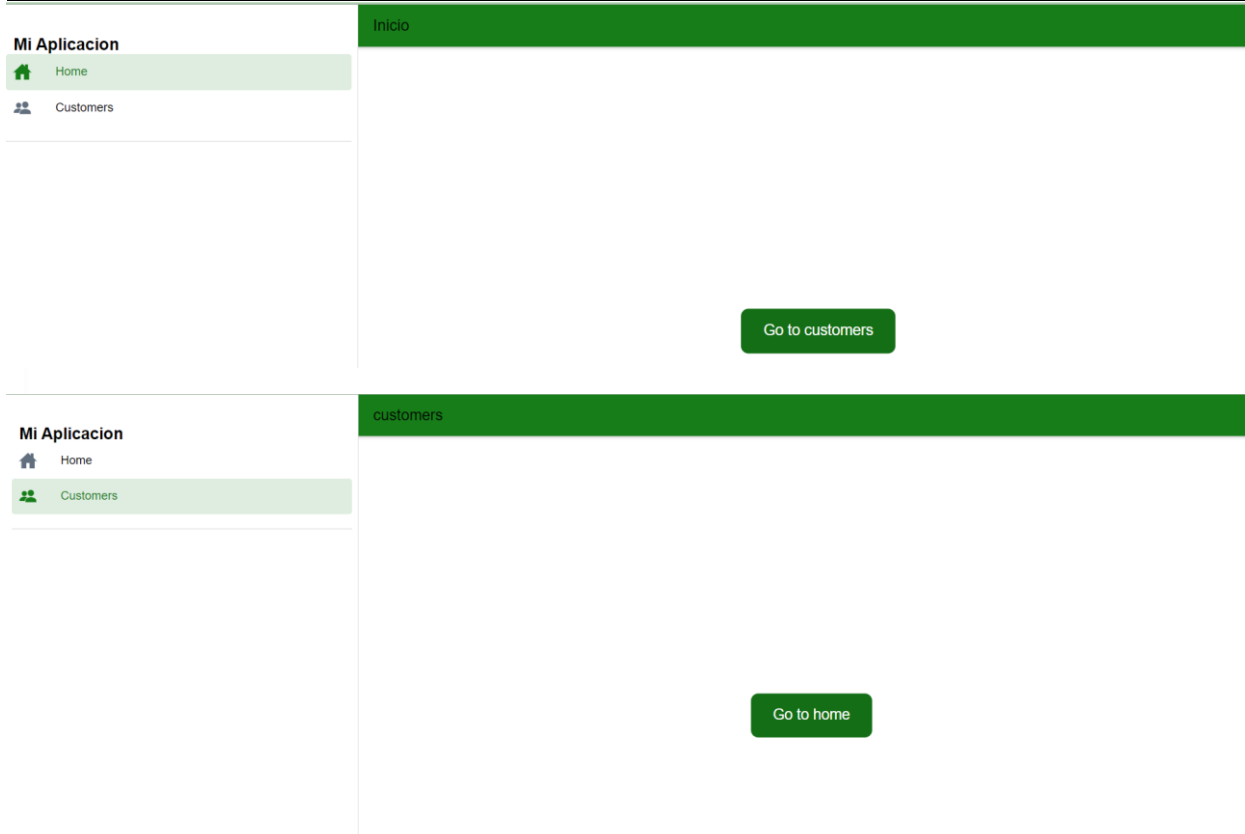


# Ionic

**Navegación entre páginas: Aplicación de estilos aplicar lo mismo a la otra pagina**

# Ionic

**Navegación entre páginas: Aplicación de estilos aplicar lo mismo a la otra pagina**



# Ionic

Listas con Avatar sustituir código de customers.page.html  
(Código lista: <https://ionicframework.com/docs/api/list>)

Modificar el contenido de la pagina customer.page.html  
Agregar imágenes png en asset/images

# Listas Ionic

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

    <ion-item>
      <ion-avatar slot="start">
        
      </ion-avatar>
      <ion-avatar slot="end">
        <ion-icon name="search-outline"></ion-icon>
      </ion-avatar>
      <ion-label>
        <h2 class="colorMyColor">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>
```

# Listas Ionic

```
<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-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>
```

# Listas Ionic

```
<ion-list>
  <ion-list-header> Online </ion-list-header>

  <ion-item>
    <ion-avatar slot="start">
      
    </ion-avatar>
    <ion-label>
      <h2>Poe</h2>
      <h3>New Ride</h3>
      <p>I just upgraded my X-Wing. Next time...</p>
    </ion-label>
  </ion-item>

  <ion-item>
    <ion-avatar slot="start">
      
    </ion-avatar>
    <ion-label>
      <h2>Ben</h2>
      <h3>Move Along</h3>
      <p>These aren't the droids you're looking for...</p>
    </ion-label>
  </ion-item>
```

# Listas Ionic

```
<ion-item>
  <ion-avatar slot="start">
    
  </ion-avatar>
  <ion-label>
    <h2>Leia</h2>
    <h3>You're My Only Hope</h3>
    <p>I've placed information vital to the survival...</p>
  </ion-label>
</ion-item>

<ion-item>
  <ion-avatar slot="start">
    
  </ion-avatar>
  <ion-label>
    <h2>Yoda</h2>
    <h3>Size matters not</h3>
    <p>Do or do not. There is no try...</p>
  </ion-label>
</ion-item>
</ion-list>
</ion-content>
```

# Grid Ionic

## Mi Aplicacion



Home



Customers



Grid

grid

Texto 1

Texto 2

Texto 3

Texto 4

grid

Texto 1

Texto 2

Texto 3

Texto 4




# Grid Ionic

<> grid.page.html U X

```
<ion-header>
  <ion-toolbar>
    <ion-title>grid</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>
  <ion-grid>
    <ion-row>
      <ion-col sizeLg="4" sizeMd="4" sizeXs="6">
        <div class="texto">Texto 1</div>
      </ion-col>
      <ion-col sizeLg="4" sizeMd="4" sizeXs="6">
        <div class="texto">Texto 2</div>
      </ion-col>
    </ion-row>
    <ion-row>
      <ion-col sizeLg="4" sizeMd="4" sizeXs="9">
        <div class="texto">Texto 3</div>
      </ion-col>
      <ion-col sizeLg="4" sizeMd="4" sizeXs="3">
        <div class="texto">Texto 4</div>
      </ion-col>
    </ion-row>
  </ion-grid>
</ion-content>
```

# Grid Ionic

 grid.page.scss U X

```
ion-toolbar{
  background-color: var(--ion-color-mycolor);
}
.texto{
  background-color: cadetblue;
  text-align: center;
  padding-top: 20px;
  padding-bottom: 20px;
}
```