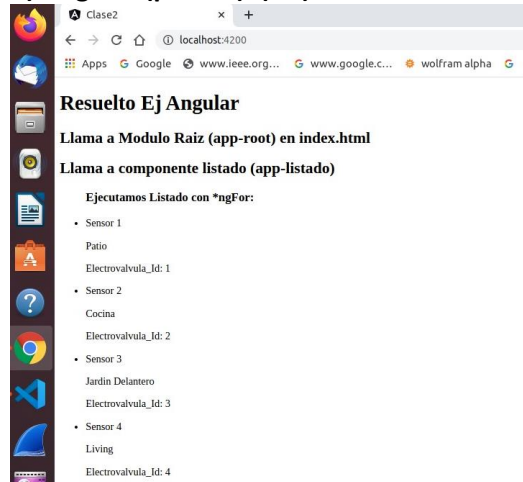


# Ionic4 + Angular resuelto detalles - R.Oliva 12.08.2020

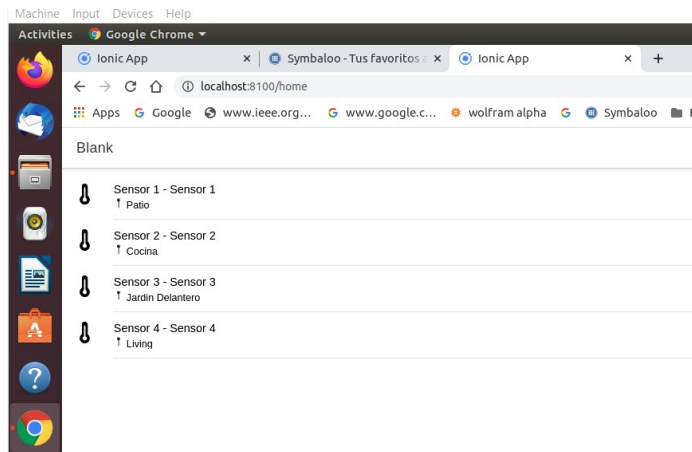
## Uso de Angular + Ionic / Resueltos Listado upd15-08-20. Se usan como base para el TP

1) Angular y ionic resueltos → Se comparan los casos de Listado con Angular Pelado (AP), Listado con Ionic Base (IB) [se puede construir la app Web y dp simular con ionic lab móviles Android / IOS y Ionic Conference (IC):

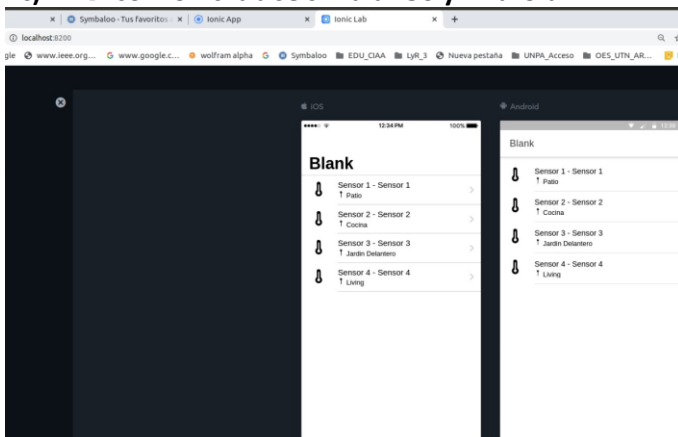
### 1a) Angular (pelado) (AP)



### 1b) Ionic Base (IB), modo Web

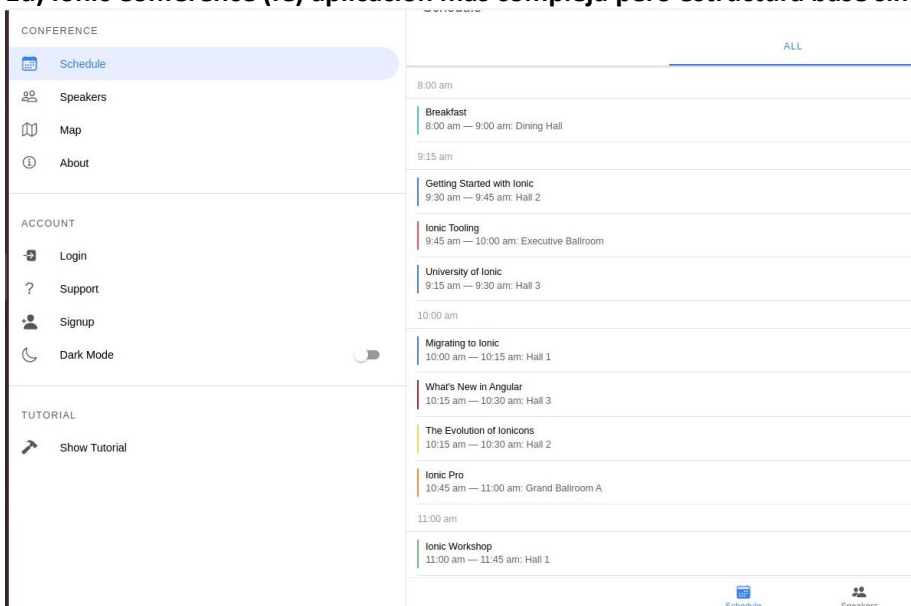


### 1c) IB → con ionic lab se simula IOS y Android



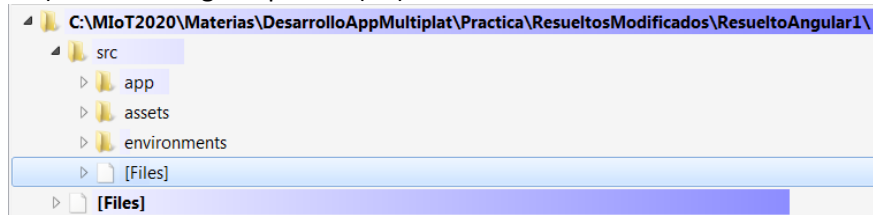
Todos son aplicaciones desarrolladas sobre el framework Angular, en los últimos 2 se agregan capacidades gráficas a través de Ionic Framework corriendo sobre Angular.

### 1d) Ionic Conference (IC) aplicación mas compleja pero estructura base similar.

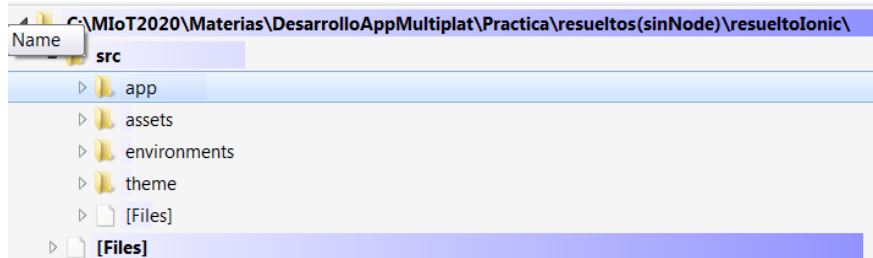


## 2) File Tree BASICO

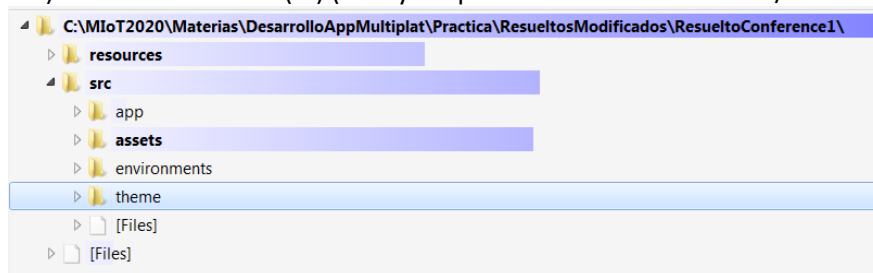
### 2.1) Listado - Angular pelado (AP):



### 2.2) Listado – Ionic Base (IB)

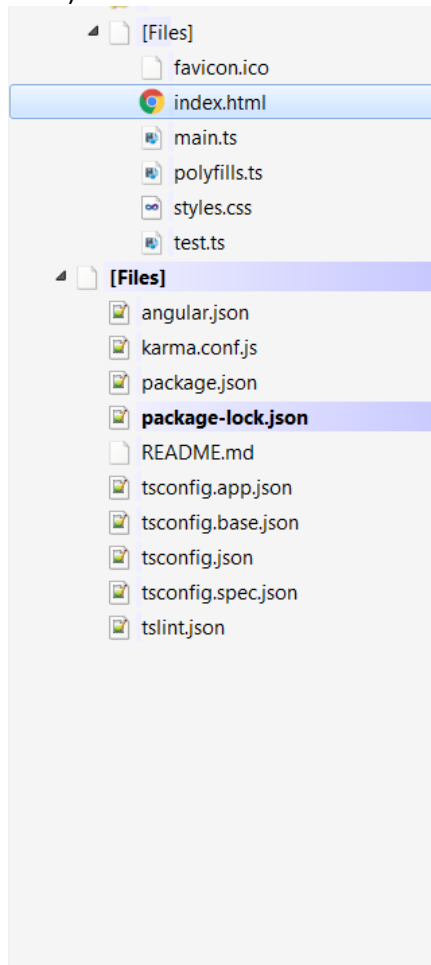


### 2.3) Conference –Ionic (IC) (incluye específicos Android e IOS en /resources)

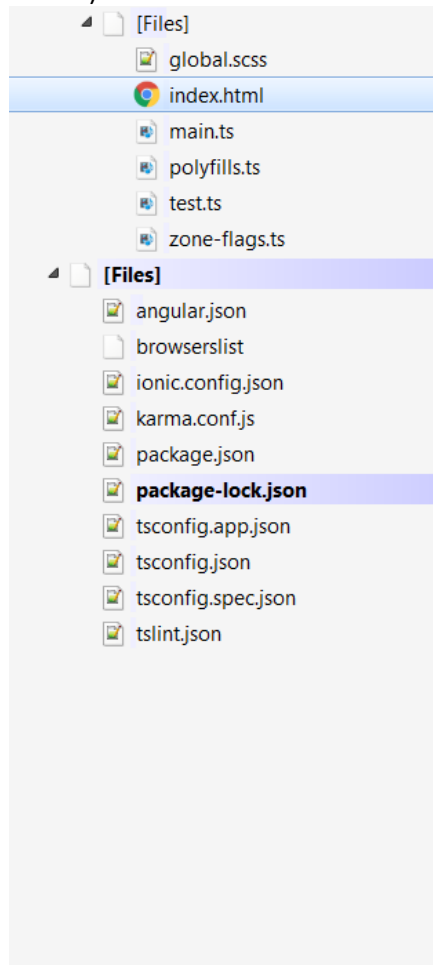


### 2.4) Contenido de los /Files y src/Files:

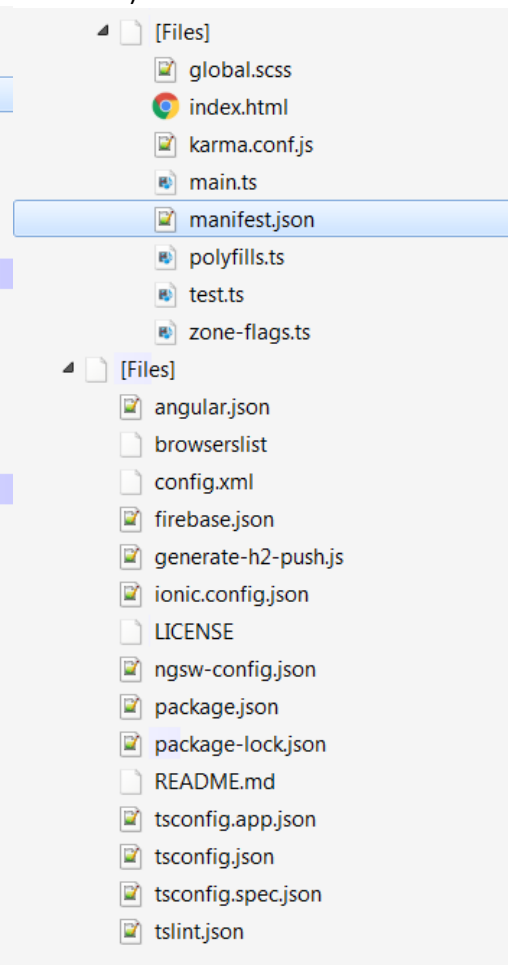
#### 2.4.1) AP



#### 2.4.2) IB



#### 2.4.3) IC



3) Index.html Base en todos: <head> ...</head> define cuestiones de formato globales, y entre<body>..</body> se llama a <app-root> </app-root>, definidos en los archivos de control de Angular.

3.1) index.html –(AP) , con poco formato “por defecto”, y los dos primeros renglones de título:

```
1 <!doctype html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <title>Clase2</title>
6   <base href="/">
7   <meta name="viewport" content="width=device-width, initial-scale=1">
8   <link rel="icon" type="image/x-icon" href="favicon.ico">
9 </head>
10 <body>
11   <h1>Resuelto Ej Angular</h1>
12   <h2>Llama a Modulo Raiz (app-root) en index.html</h2>
13   <app-root></app-root>
14 </body>
15 </html>
```

3.2) index.html - (IB), algo más:

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="utf-8" />
6   <title>Ionic App</title>
7
8   <base href="/" />
9
10  <meta name="color-scheme" content="light dark" />
11  <meta name="viewport" content="viewport-fit=cover, width=device-width, initial-scale=1.0, minin
12  <meta name="format-detection" content="telephone=no" />
13  <meta name="msapplication-tap-highlight" content="no" />
14
15  <link rel="icon" type="image/png" href="assets/icon/favicon.png" />
16
17  <!-- add to homescreen for ios -->
18  <meta name="apple-mobile-web-app-capable" content="yes" />
19  <meta name="apple-mobile-web-app-status-bar-style" content="black" />
20 </head>
21
22 <body>
23   <app-root></app-root>
24 </body>
25
26 </html>
```

3.3) index.html – (IC) Conference:

```
1 <!DOCTYPE html>
2 <html lang="en" dir="ltr">
3
4 <head>
5   <meta charset="UTF-8">
6   <title>Ionic Conference App</title>
7
8   <meta name="viewport" content="viewport-fit=cover, width=device-width, initial-scale=1.0, minimum-scale=1.0, maximum-scale=1.0, user-s
9   <meta name="format-detection" content="telephone=no">
10  <meta name="msapplication-tap-highlight" content="no">
11
12  <meta name="theme-color" content="#387ef5">
13
14  <meta name="apple-mobile-web-app-capable" content="yes">
15  <meta name="apple-mobile-web-app-status-bar-style" content="black-translucent">
16
17  <base href="/">
18
19  <link rel="icon" type="image/x-icon" href="assets/img/appicon.png">
20
21  <link rel="apple-touch-icon" href="assets/img/appicon.png">
22
23  <link rel="manifest" href="manifest.json">
24 </head>
25
26 <body>
27
28   <app-root></app-root>
29
30   <!-- Replace the API key with your own, see:
31   <a href="https://developers.google.com/maps/documentation/javascript/get-api-key">https://developers.google.com/maps/documentation/javascript/get-api-key</a> -->
32   <!-- <script async="" defer="" src="https://maps.googleapis.com/maps/api/js?key=AIzaSyB8pf6ZdFQj5qw7rc_HSGrhUwQKfIe9ICw"></script> -->
33
34   <noscript>Please enable JavaScript to continue using this application.</noscript>
35 </body>
36
37 </html>
```

4) main.ts Base –Idéntico en los 3 (AP, IB, IC) ya que están todos basados en Angular.

```
main.ts
1 import { enableProdMode } from '@angular/core';
2 import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';
3
4 import { AppModule } from './app/app.module';
5 import { environment } from './environments/environment';
6
7 if (environment.production) {
8   enableProdMode();
9 }
10
11 platformBrowserDynamic().bootstrapModule(AppModule)
12   .catch(err => console.log(err));
13
```

5) Estilos, manifest: global css, scss File – vacío en AP. En los Ionic, hay archivos por defecto. En el IC que se muestra se agregan los dos últimos renglones, aunque el app.scss está vacío.

### 5.1) SCSS

```
global.scss
1 /*
2  * App Global CSS
3  * Put style rules here that you want to apply globally. These styles are for
4  * the entire app and not just one component. Additionally, this file can be
5  * used as an entry point to import other CSS/Sass files to be included in the
6  * output CSS.
7  * For more information on global stylesheets, visit the documentation:
8  * https://ionicframework.com/docs/layout/global-stylesheets
9  */
10
11 /* Core CSS required for Ionic components to work properly */
12 @import "~@ionic/angular/css/core.css";
13
14 /* Basic CSS for apps built with Ionic */
15 @import "~@ionic/angular/css/normalize.css";
16 @import "~@ionic/angular/css/structure.css";
17 @import "~@ionic/angular/css/typography.css";
18
19 /* Optional CSS utils that can be commented out */
20 @import "~@ionic/angular/css/padding.css";
21 @import "~@ionic/angular/css/float-elements.css";
22 @import "~@ionic/angular/css/text-alignment.css";
23 @import "~@ionic/angular/css/text-transformation.css";
24 @import "~@ionic/angular/css/flex-utils.css";
25
26
27
28 /*
29  * App CSS
30  * Imports a file that can contain Sass/CSS that should be used throughout
31  * the entire app.
32  */
33
34 @import "./app/app.scss";
35
36
```

### 5.2) Los IB, IC tienen un ionic.config.json

#### 5.2.1) IB)

```
manifest.json ionic.config.json ionic.config.json
1 {
2   "name": "resueltoIonic",
3   "integrations": {},
4   "type": "angular"
5 }
6
```

#### 5.2.1) IC)

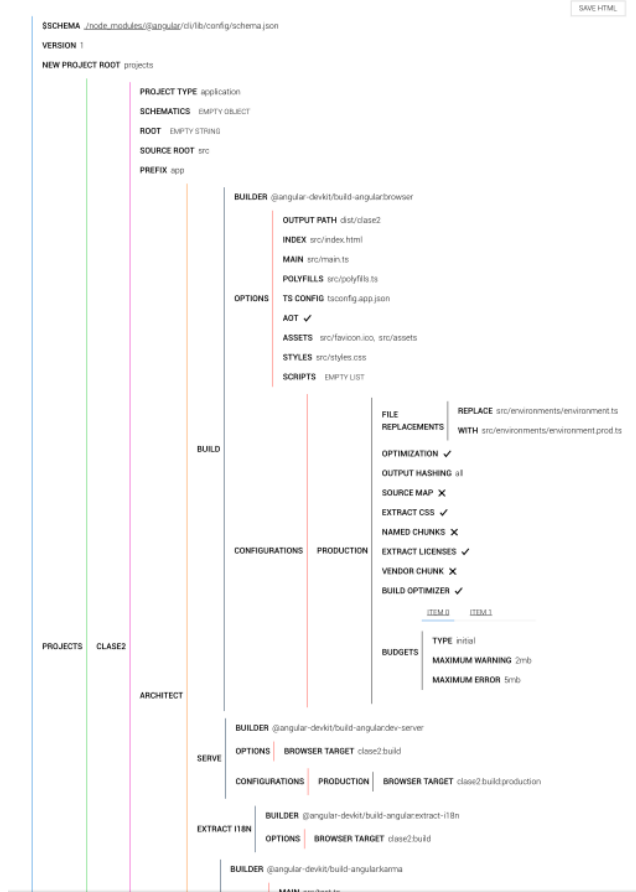
```
manifest.json ionic.config.json ionic.config.json
1 {
2   "name": "ionic-conference-app",
3   "type": "angular",
4   "integrations": {
5     "cordova": {}
6   }
7 }
8
```

### 5.2.2) Manifest en IC, sólo para build Mobile (además tiene un XML, ver mas adelante).

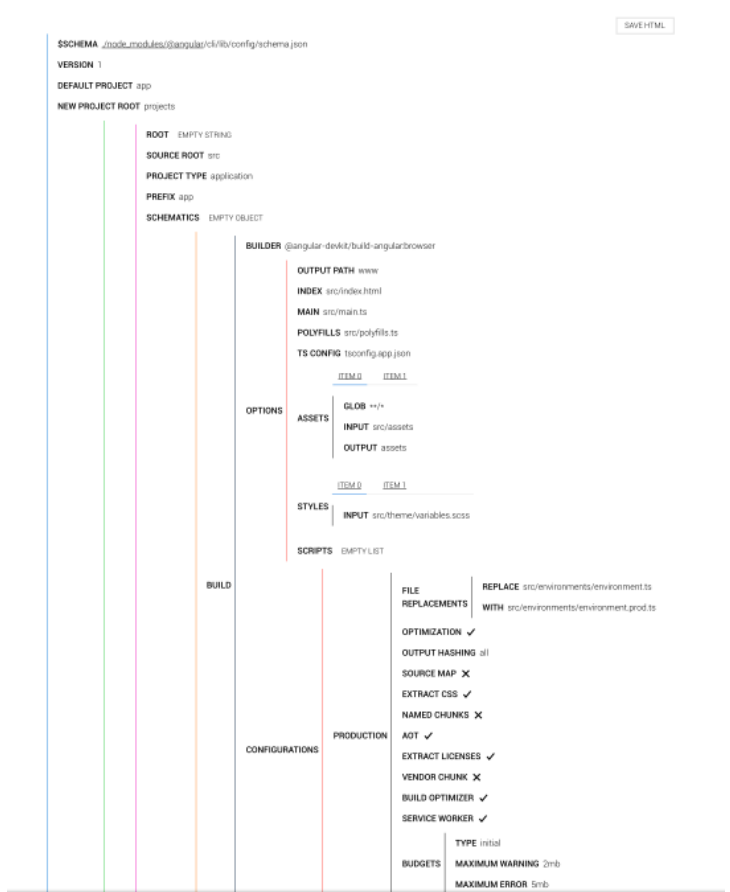
```
manifest.json ionic.config.json ionic.config.json
1 {
2   "name": "Ionic Conference",
3   "short_name": "Ionic Conference",
4   "start_url": "index.html",
5   "display": "standalone",
6   "icons": [
7     {
8       "src": "assets/img/appicon.png",
9       "sizes": "512x512",
10      "type": "image/png"
11    }
12  ],
13   "background_color": "#387ef5",
14   "theme_color": "#387ef5"
15 }
```

## 6) Angular.json (reducido en AP, un poco mas complejo en los 2 IB, IC que son iguales, agregan CORDOVA al final)

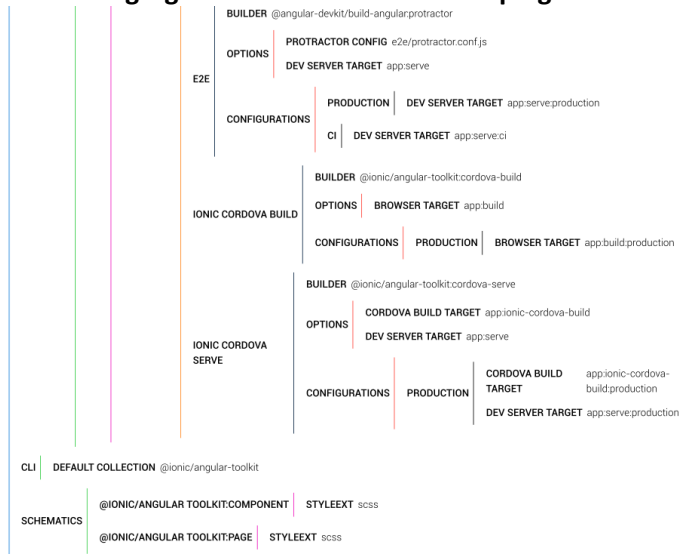
### 6.1) AP parcial inicio



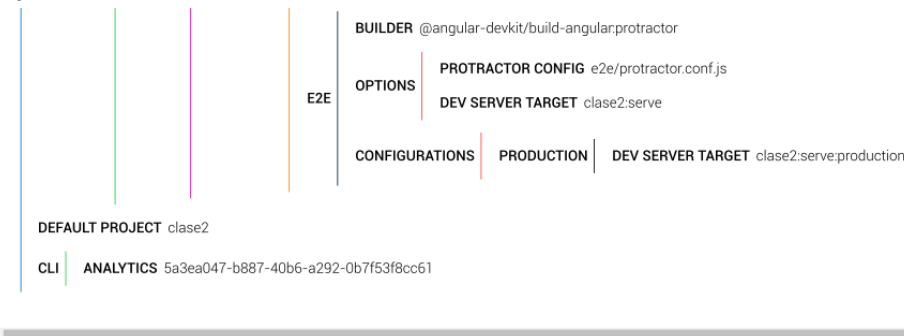
### 6.2a) IB, IC parcial inicio (iguales, no tienen "nombre")



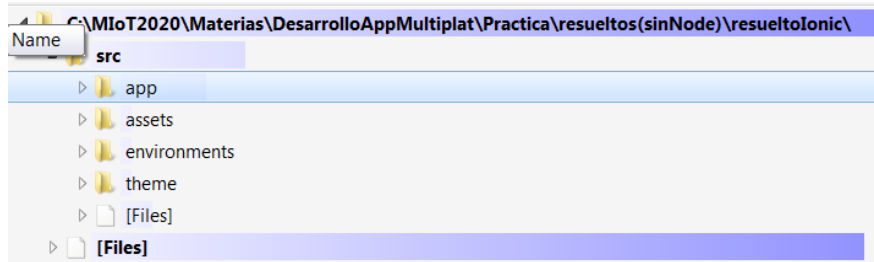
### 6.2c) IB, IC al final agregan referencia a CORDOVA plugins



### 6.3) AP final



**7) Aplicación.** Como se vió en 2.1) a 2.3) las tres aplicaciones tienen dentro de /src los directorios /app, /assets y /environments, y en el caso de los IB, IC incluyen un directorio /theme (ej IB):



### 7.1) File Tree de /app en cada caso:

Dentro de /app están los archivos propios de la aplicación. Se realiza una comparación de AP, IB, IC.

**7.1a) AP (component en vez de pages, pero son similares).** Se crea con `ng new nombreProy` (crea los `app.component.*` dentro de /app, junto con `app.module.ts` y `app-routing.module.ts`). Luego se crea /e2e, /node\_modules y los archivos raíz. Aquí se agregó luego una única clase /model/Dispositivo.ts. Dentro del componente listado creado con `ng create component listado` aparecen los siguientes: .css de estilo, vista .html

|                                                                                             |          |
|---------------------------------------------------------------------------------------------|----------|
| C:\MioT2020\Materias\DesarrolloAppMultiplat\Practica\ResueltosModificados\ResueltoAngular1\ | 530,0 KB |
| src                                                                                         | 10,6 KB  |
| app                                                                                         | 4,7 KB   |
| listado                                                                                     | 0,9 KB   |
| listado.component.css                                                                       | 0,0 KB   |
| listado.component.html                                                                      | 0,3 KB   |
| listado.component.ts                                                                        | 0,4 KB   |
| listado.module.ts                                                                           | 0,3 KB   |
| model                                                                                       | 1,0 KB   |
| Dispositivo.ts                                                                              | 1,0 KB   |
| services                                                                                    | 0,6 KB   |
| dispositivo.service.ts                                                                      | 0,6 KB   |
| [Files]                                                                                     | 2,0 KB   |
| app.component.css                                                                           | 0,0 KB   |
| app.component.html                                                                          | 0,1 KB   |
| app.component.spec.ts                                                                       | 1,0 KB   |
| app.component.ts                                                                            | 0,2 KB   |
| app.module.ts                                                                               | 0,5 KB   |
| app-routing.module.ts                                                                       | 0,2 KB   |
| assets                                                                                      | 0,0 KB   |
| environments                                                                                | 0,7 KB   |
| [Files]                                                                                     | 5,2 KB   |
| [Files]                                                                                     | 519,4 KB |

### 7.1b) IB Ionic: Aquí 2 pages (orientado a mobile): home y dispositivo, y comparte /model/Dispositivo.ts

|                                                                                        |          |
|----------------------------------------------------------------------------------------|----------|
| C:\MioT2020\Materias\DesarrolloAppMultiplat\Practica\resueltos(sinNode)\resueltoIonic\ | 555,5 KB |
| src                                                                                    | 24,9 KB  |
| app                                                                                    | 9,7 KB   |
| dispositivo                                                                            | 2,0 KB   |
| dispositivo.module.ts                                                                  | 0,5 KB   |
| dispositivo.page.html                                                                  | 0,5 KB   |
| dispositivo.page.scss                                                                  | 0,0 KB   |
| dispositivo.page.ts                                                                    | 0,7 KB   |
| dispositivo-routing.module.ts                                                          | 0,4 KB   |
| home                                                                                   | 2,3 KB   |
| home.module.ts                                                                         | 0,4 KB   |
| home.page.html                                                                         | 0,9 KB   |
| home.page.scss                                                                         | 0,3 KB   |
| home.page.ts                                                                           | 0,3 KB   |
| home-routing.module.ts                                                                 | 0,3 KB   |
| model                                                                                  | 1,0 KB   |
| Dispositivo.ts                                                                         | 1,0 KB   |
| services                                                                               | 0,8 KB   |
| dispositivo.service.ts                                                                 | 0,8 KB   |
| [Files]                                                                                | 3,6 KB   |
| app.component.html                                                                     | 0,1 KB   |
| app.component.scss                                                                     | 0,0 KB   |
| app.component.spec.ts                                                                  | 1,5 KB   |
| app.component.ts                                                                       | 0,6 KB   |
| app.module.ts                                                                          | 0,7 KB   |
| app-routing.module.ts                                                                  | 0,6 KB   |
| assets                                                                                 | 2,1 KB   |
| environments                                                                           | 0,7 KB   |
| theme                                                                                  | 6,7 KB   |
| [Files]                                                                                | 5,7 KB   |
| [Files]                                                                                | 530,6 KB |

**7.1c) IC Ionic: App Conference.** Tiene múltiples pages, pero siguen la distribución similar a las 2 anteriores:

| Name                  | Size       |
|-----------------------|------------|
| src                   | 8.906,6 KB |
| app                   | 84,4 KB    |
| interfaces            | 0,1 KB     |
| user-options.ts       | 0,1 KB     |
| pages                 | 65,2 KB    |
| about                 | 5,7 KB     |
| about-popover         | 1,1 KB     |
| account               | 3,2 KB     |
| login                 | 3,2 KB     |
| map                   | 6,5 KB     |
| schedule              | 9,7 KB     |
| schedule-filter       | 3,2 KB     |
| session-detail        | 5,5 KB     |
| signup                | 3,0 KB     |
| speaker-detail        | 6,2 KB     |
| speaker-list          | 5,0 KB     |
| support               | 3,7 KB     |
| tabs-page             | 4,1 KB     |
| tutorial              | 5,3 KB     |
| providers             | 6,6 KB     |
| [Files]               | 12,6 KB    |
| app.component.html    | 3,7 KB     |
| app.component.scss    | 1,4 KB     |
| app.component.spec.ts | 2,3 KB     |
| app.component.ts      | 2,9 KB     |
| app.module.ts         | 1,1 KB     |
| app.scss              | 0,0 KB     |
| app-routing.module.ts | 1,1 KB     |
| assets                | 8.801,3 KB |
| environments          | 0,8 KB     |
| theme                 | 12,7 KB    |
| [Files]               | 7,4 KB     |

En el caso de Ionic (IB, IC) se crean con:

**7.1d) (IB)** ionic start nombre\_proyecto blank (crea los app.component.\* dentro de /app, junto con app.module.ts y app-routing.module.ts). Luego se crea /e2e, /node\_modules y además /theme en ionic. y los archivos raíz. Se migra desde AP lo relativo a Listado Después agregamos con ionic generate page dispositivo todo lo relativo a Dispositivos.

Hay que seleccionar Angular como FW base, y además ahora pide compatibilidad con Capacitor (solución interna de Ionic que evita los plugins externos Cordova, todavía no lo usamos)

**7.1e) (IC)** ionic start -list y allí se elige de la lista Conference para Angular. Esto crea la estructura 7.1c) que es un proyecto completo.



## 8) Archivos app.module.ts de cada caso:

### 8.1) AP

```
1 import { BrowserModule } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3
4 import { AppRoutingModule } from './app-routing.module';
5 import { AppComponent } from './app.component';
6 import { ListadoModule } from './listado/listado.module';
7
8 @NgModule({
9   declarations: [
10     AppComponent
11   ],
12   imports: [
13     BrowserModule,
14     AppRoutingModule,
15     ListadoModule
16   ],
17   providers: [],
18   bootstrap: [AppComponent]
19 })
20 export class AppModule { }
21
```

### 8.2) IB

```
1 import { NgModule } from '@angular/core';
2 import { BrowserModule } from '@angular/platform-browser';
3 import { RouteReuseStrategy } from '@angular/router';
4
5 import { IonicModule, IonicRouteStrategy } from '@ionic/angular';
6 import { SplashScreen } from '@ionic-native/splash-screen/ngx';
7 import { StatusBar } from '@ionic-native/status-bar/ngx';
8
9 import { AppComponent } from './app.component';
10 import { AppRoutingModule } from './app-routing.module';
11
12 @NgModule({
13   declarations: [AppComponent],
14   entryComponents: [],
15   imports: [BrowserModule, IonicModule.forRoot(), AppRoutingModule],
16   providers: [
17     StatusBar,
18     SplashScreen,
19     { provide: RouteReuseStrategy, useClass: IonicRouteStrategy }
20   ],
21   bootstrap: [AppComponent]
22 })
23 export class AppModule { }
24
```

### 8.3) IC

```
1 import { HttpClientModule } from '@angular/common/http';
2 import { NgModule } from '@angular/core';
3 import { BrowserModule } from '@angular/platform-browser';
4 import { InAppBrowser } from '@ionic-native/in-app-browser/ngx';
5 import { SplashScreen } from '@ionic-native/splash-screen/ngx';
6 import { StatusBar } from '@ionic-native/status-bar/ngx';
7 import { IonicModule } from '@ionic/angular';
8 import { IonicStorageModule } from '@ionic/storage';
9
10 import { AppRoutingModule } from './app-routing.module';
11 import { AppComponent } from './app.component';
12 import { ServiceWorkerModule } from '@angular/service-worker';
13 import { environment } from '../environments/environment';
14 import { FormsModule } from '@angular/forms';
15
16 @NgModule({
17   imports: [
18     BrowserModule,
19     AppRoutingModule,
20     HttpClientModule,
21     FormsModule,
22     IonicModule.forRoot(),
23     IonicStorageModule.forRoot(),
24     ServiceWorkerModule.register('ngsw-worker.js', {
25       enabled: environment.production
26     })
27   ],
28   declarations: [AppComponent],
29   providers: [InAppBrowser, SplashScreen, StatusBar],
30   bootstrap: [AppComponent]
31 })
32 export class AppModule { }
33
```

En los tres casos se comparte el `@NgModule` que define fundamentalmente los **declarations**, **bootstrap** (siempre `[AppComponent]`) **imports** [varia] y los **imports** dan lugar a los “statements” de importación comunes a todos al principio:

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
```

Además se exporta la clase `AppModule`. En el caso de AP Listado se agregó. En el caso de IB, IC se incluye la importación de `IonicModule` ( y variantes, `Storage` para IC, `IonicRoute` para IB) y los `Providers` `StatusBar` y `SplashScreen`. Solo en IC se incluye `InAppBrowser` y se importa el cliente requerido con :

```
import { HttpClientModule } from '@angular/common/http';
```



## 9) Archivos app.component.ts de cada caso:

### 9.1) AP

```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-root',
5   templateUrl: './app.component.html',
6   styleUrls: ['./app.component.css']
7 })
8 export class AppComponent {
9   title = 'clase2';
10 }
11
```

### 9.2) IB

```
1 import { Component } from '@angular/core';
2
3 import { Platform } from '@ionic/angular';
4 import { SplashScreen } from '@ionic-native/splash-screen/ngx';
5 import { StatusBar } from '@ionic-native/status-bar/ngx';
6
7 @Component({
8   selector: 'app-root',
9   templateUrl: 'app.component.html',
10   styleUrls: ['app.component.scss']
11 })
12 export class AppComponent {
13   constructor(
14     private platform: Platform,
15     private splashScreen: SplashScreen,
16     private statusBar: StatusBar
17   ) {
18     this.initializeApp();
19   }
20
21   initializeApp() {
22     this.platform.ready().then(() => {
23       this.statusBar.styleDefault();
24       this.splashScreen.hide();
25     });
26   }
27 }
28
```

### 9.3) IC (primera parte)

```
1 import { Component, OnInit, ViewEncapsulation } from '@angular/core';
2 import { Router } from '@angular/router';
3 import { SwUpdate } from '@angular/service-worker';
4 import { MenuController, Platform, ToastController } from '@ionic/angular';
5 import { SplashScreen } from '@ionic-native/splash-screen/ngx';
6 import { StatusBar } from '@ionic-native/status-bar/ngx';
7 import { Storage } from '@ionic/storage';
8 import { UserData } from './providers/user-data';
9
10 @Component({
11   selector: 'app-root',
12   templateUrl: './app.component.html',
13   styleUrls: ['./app.component.scss'],
14   encapsulation: ViewEncapsulation.None
15 })
16 export class AppComponent implements OnInit {
17   appPages = [
18     {
19       title: 'Schedule',
20       url: '/app/tabs/schedule',
21       icon: 'calendar'
22     },
23     {
24       title: 'Speakers',
25       url: '/app/tabs/speakers',
26       icon: 'people'
27     },
28     {
29       title: 'Map',
30       url: '/app/tabs/map',
31       icon: 'map'
32     },
33     {
34       title: 'About',
35       url: '/app/tabs/about',
36       icon: 'information-circle'
37     }
38   ];
39   loggedIn = false;
40   dark = false;
41
42   constructor(
43     private menu: MenuController,
44     private platform: Platform,
45     private router: Router,
46     private splashScreen: SplashScreen,
47     private statusBar: StatusBar,
48     private storage: Storage,
49     private userData: UserData,
50     private swUpdate: SwUpdate,
51     private toastCtrl: ToastController,
52   ) {
53     this.initializeApp();
54   }
55
56   async ngOnInit() {
```

(la 2º parte mas adelante)

En los tres casos se “importan” al principio los módulos requeridos, y luego las tres comparten el `@NgComponent` que define fundamentalmente el `selector: 'app-root'` que se llama desde `index.html`, y el template HTML: `templateUrl: './app.component.html'`, con su archivo de estilo `styleUrls: ['./app.component.scss']`. Después se da lugar a alguna variante (en Ionic es más compleja) de:

```
export class AppComponent {
}
```

La continuación de app.component.ts de Conference es:

```

56 async ngOnInit() {
57   this.checkLoginStatus();
58   this.listenForLoginEvents();
59
60   this.swUpdate.available.subscribe(async res => {
61     const toast = await this.toastCtrl.create({
62       message: 'Update available!',
63       position: 'bottom',
64       buttons: [
65         {
66           role: 'cancel',
67           text: 'Reload'
68         }
69       ]
70     });
71     await toast.present();
72
73     toast
74       .onDidDismiss()
75       .then(() => this.swUpdate.activateUpdate())
76       .then(() => window.location.reload());
77   });
78 }
79
80 initializeApp() {
81   this.platform.ready().then(() => {
82     this.statusBar.styleDefault();
83     this.splashScreen.hide();
84   });
85 }
86
87 checkLoginStatus() {
88   return this.userData.isLoggedIn().then(loggedIn => {
89     return this.updateLoggedInStatus(loggedIn);
90   });
91 }
92
93 updateLoggedInStatus(loggedIn: boolean) {
94   setTimeout(() => {
95     this.loggedIn = loggedIn;
96   }, 300);
97 }
98
99 listenForLoginEvents() {
100   window.addEventListener('user:login', () => {
101     this.updateLoggedInStatus(true);
102   });
103
104   window.addEventListener('user:signup', () => {
105     this.updateLoggedInStatus(true);
106   });
107
108   window.addEventListener('user:logout', () => {
109     this.updateLoggedInStatus(false);
110   });
111 }
112
113 logout() {
114   this.userData.logout().then(() => {
115     return this.router.navigateByUrl('/app/tabs/schedule');
116   });
117 }
118
119 openTutorial() {
120   this.menu.enable(false);
121   this.storage.set('ion_did_tutorial', false);
122   this.router.navigateByUrl('/tutorial');
123 }
124
125 }
126

```

## 10) Archivos app.component.html de cada caso:

### 10.1) AP

```

app.component.html
1 <h2>Llama a componente listado (app-listado)</h2>
2 <app-listado></app-listado>
3
4 <router-outlet></router-outlet>

```

### 10.2) IB

```

app.component.html
1 <ion-app>
2   <ion-router-outlet></ion-router-outlet>
3 </ion-app>
4

```

10.2a) IC: Este es significativamente más complejo, ya que realiza una distribución de los componentes que utiliza la aplicación de conferencia. Usa un “Split-pane” y el content del menu depnde de si está el usuario “loggeado” o no:

```

app.component.html
1 <ion-app [class.dark-theme]="dark">
2   <ion-split-pane contentId="main-content">
3
4     <ion-menu contentId="main-content">
5       <ion-content>
6         <ion-list lines="none">
20
21           <ion-list *ngIf="loggedIn" lines="none">
54
55           <ion-list *ngIf="!loggedIn" lines="none">
95
96           <ion-list lines="none">
107         </ion-content>
108       </ion-menu>
109
110       <ion-router-outlet id="main-content"></ion-router-outlet>
111     </ion-split-pane>
112   </ion-app>
113
114
115

```

## 10.2b) IC: El listado completo se aprecia aquí:

```
1. <ion-app [class.dark-theme]="dark">
2.   <ion-split-pane contentId="main-content">
3.
4.     <ion-menu contentId="main-content">
5.       <ion-content>
6.         <ion-list lines="none">
7.           <ion-list-header>
8.             Conference
9.           </ion-list-header>
10.          <ion-menu-toggle autoHide="false" *ngFor="let p of appPages; let i = index">
11.            <ion-
12.              item [routerLink]="p.url" routerLinkActive="selected" routerDirection="root" detail="false">
13.                <ion-icon slot="start" [name]="p.icon + '-outline'"></ion-icon>
14.                <ion-label>
15.                  {{p.title}}
16.                </ion-label>
17.              </ion-item>
18.            </ion-menu-toggle>
19.          </ion-list>
20.
21.          <ion-list *ngIf="loggedIn" lines="none">
22.            <ion-list-header>
23.              Account
24.            </ion-list-header>
25.
26.            <ion-menu-toggle autoHide="false">
27.              <ion-item routerLink="/account" routerLinkActive="active" routerDirection="root" detail="false">
28.                <ion-icon slot="start" name="person"></ion-icon>
29.                <ion-label>
30.                  Account
31.                </ion-label>
32.              </ion-item>
33.            </ion-menu-toggle>
34.
35.            <ion-menu-toggle autoHide="false">
36.              <ion-item routerLink="/support" routerLinkActive="active" routerDirection="root" detail="false">
37.                <ion-icon slot="start" name="help"></ion-icon>
38.                <ion-label>
39.                  Support
40.                </ion-label>
41.              </ion-item>
42.            </ion-menu-toggle>
43.
44.            <ion-menu-toggle autoHide="false">
45.              <ion-item button (click)="logout()" detail="false">
46.                <ion-icon slot="start" name="log-out"></ion-icon>
47.                <ion-label>
48.                  Logout
49.                </ion-label>
50.              </ion-item>
51.            </ion-menu-toggle>
52.
53.          </ion-list>
54.
55.          <ion-list *ngIf="!loggedIn" lines="none">
56.            <ion-list-header>
57.              Account
58.            </ion-list-header>
59.
60.            <ion-menu-toggle autoHide="false">
61.              <ion-item routerLink="/login" routerLinkActive="active" routerDirection="root" detail="false">
62.                <ion-icon slot="start" name="log-in"></ion-icon>
63.                <ion-label>
64.                  Login
65.                </ion-label>
66.              </ion-item>
67.            </ion-menu-toggle>
```

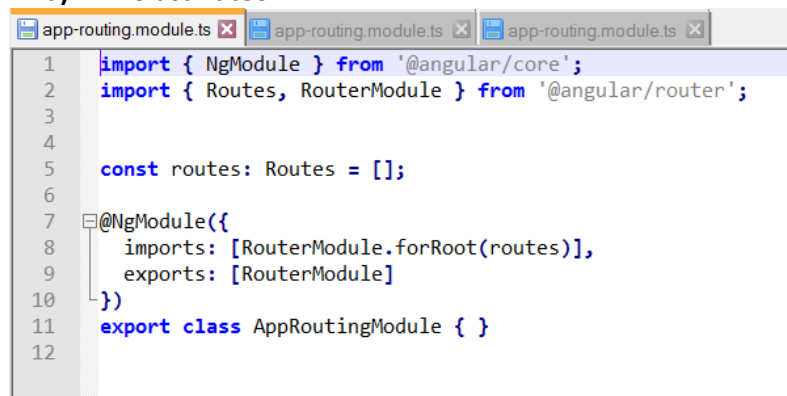
```

68.
69.     <ion-menu-toggle autoHide="false">
70.         <ion-item routerLink="/support" routerLinkActive="active" routerDirection="root" detail="false">
71.             <ion-icon slot="start" name="help"></ion-icon>
72.             <ion-label>
73.                 Support
74.             </ion-label>
75.         </ion-item>
76.     </ion-menu-toggle>
77.
78.     <ion-menu-toggle autoHide="false">
79.         <ion-item routerLink="/signup" routerLinkActive="active" routerDirection="root" detail="false">
80.             <ion-icon slot="start" name="person-add"></ion-icon>
81.             <ion-label>
82.                 Signup
83.             </ion-label>
84.         </ion-item>
85.     </ion-menu-toggle>
86.
87.     <ion-item>
88.         <ion-icon slot="start" name="moon-outline"></ion-icon>
89.         <ion-label>
90.             Dark Mode
91.         </ion-label>
92.         <ion-toggle [(ngModel)]="dark"></ion-toggle>
93.     </ion-item>
94. </ion-list>
95.
96.     <ion-list lines="none">
97.         <ion-list-header>
98.             Tutorial
99.         </ion-list-header>
100.        <ion-menu-toggle autoHide="false">
101.            <ion-item button (click)="openTutorial()" detail="false">
102.                <ion-icon slot="start" name="hammer"></ion-icon>
103.                <ion-label>Show Tutorial</ion-label>
104.            </ion-item>
105.        </ion-menu-toggle>
106.    </ion-list>
107. </ion-content>
108. </ion-menu>
109.
110. <ion-router-outlet id="main-content"></ion-router-outlet>
111.
112. </ion-split-pane>
113.
114. </ion-app>

```

## 11) Archivos app-routing.module.ts de cada caso:

### 11.a) AP no usa ruteo:



```

1  import { NgModule } from '@angular/core';
2  import { Routes, RouterModule } from '@angular/router';
3
4
5  const routes: Routes = [];
6
7  @NgModule({
8      imports: [RouterModule.forRoot(routes)],
9      exports: [RouterModule]
10 })
11 export class AppRoutingModule { }
12

```

### 11.b) IB usa ruteo a los dos módulos → home y dispositivos:

```
app-routing.module.ts app-routing.module.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: 'home',
7     loadChildren: () => import('./home/home.module').then( m => m.HomePageModule)
8   },
9   {
10    path: '',
11    redirectTo: 'home',
12    pathMatch: 'full'
13  },
14  {
15    path: 'dispositivo/:id',
16    loadChildren: () => import('./dispositivo/dispositivo.module').then( m => m.DispositivoPageModule)
17  },
18 ];
19
20 @NgModule({
21   imports: [
22     RouterModule.forRoot(routes, { preloadingStrategy: PreloadAllModules })
23   ],
24   exports: [RouterModule]
25 })
26 export class AppRoutingModule { }
27
```

### 11.c) IC usa ruteo a varios modulos:

```
app-routing.module.ts app-routing.module.ts app-routing.module.ts
1 import { NgModule } from '@angular/core';
2 import { RouterModule, Routes } from '@angular/router';
3 import { CheckTutorial } from './providers/check-tutorial.service';
4
5 const routes: Routes = [
6   {
7     path: '',
8     redirectTo: '/tutorial',
9     pathMatch: 'full'
10  },
11  {
12    path: 'account',
13    loadChildren: () => import('./pages/account/account.module').then(m => m.AccountModule)
14  },
15  {
16    path: 'support',
17    loadChildren: () => import('./pages/support/support.module').then(m => m.SupportModule)
18  },
19  {
20    path: 'login',
21    loadChildren: () => import('./pages/login/login.module').then(m => m.LoginModule)
22  },
23  {
24    path: 'signup',
25    loadChildren: () => import('./pages/signup/signup.module').then(m => m.SignUpModule)
26  },
27  {
28    path: 'app',
29    loadChildren: () => import('./pages/tabs-page/tabs-page.module').then(m => m.TabsModule)
30  },
31  {
32    path: 'tutorial',
33    loadChildren: () => import('./pages/tutorial/tutorial.module').then(m => m.TutorialModule),
34    canLoad: [CheckTutorial]
35  }
36 ];
37
38 @NgModule({
39   imports: [RouterModule.forRoot(routes)],
40   exports: [RouterModule]
41 })
42 export class AppRoutingModule { }
43
```

## 12) Services y modelo asociado

12.1) – En AP e IB, es un service ( función disponible a toda la aplicación con @Injectable) sencillo, alojado en /services. Sólo difieren en que el de IB incorpora las líneas 21 a 23, para crear la función getDispositivo(id):

```
dispositivo.service.ts | dispositivo.service.ts
1  import { Injectable } from '@angular/core';
2  import { Dispositivo } from '../model/Dispositivo';
3
4  @Injectable({
5    providedIn: 'root'
6  })
7  export class DispositivoService {
8    listado:Array<Dispositivo> = new Array<Dispositivo>();
9
10   constructor() {
11     var disp:Dispositivo= new Dispositivo(1,"Sensor 1","Patio",1);
12     var disp2:Dispositivo= new Dispositivo(2,"Sensor 2","Cocina",2);
13     var disp3:Dispositivo= new Dispositivo(3,"Sensor 3","Jardin Delantero",3);
14     var disp4:Dispositivo= new Dispositivo(4,"Sensor 4","Living",4);
15     this.listado.push(disp);
16     this.listado.push(disp2);
17     this.listado.push(disp3);
18     this.listado.push(disp4);
19   }
20
21   getDispositivo(id):Dispositivo{
22     return this.listado.filter(dispositivo=> dispositivo.dispositivoId==id)[0];
23   }
24 }
25
```

12.1b) En /model/Dispositivo.ts se tiene definida la clase Dispositivo:

```
dispositivo.service.ts | dispositivo.service.ts | Dispositivo.ts | user-data.ts | conference-da
1  export class Dispositivo{
2    private _dispositivoId: number;
3    private _nombre: string;
4    private _ubicacion: string;
5    private _electrovalvulaId: number;
6
7    constructor(dispositivo,nombre,ubicacion,electrovalvulaId){
8      this._dispositivoId=dispositivo;
9      this._nombre=nombre;
10     this._ubicacion=ubicacion;
11     this._electrovalvulaId=electrovalvulaId;
12   }
13
14   public get dispositivoId(): number {
15     return this._dispositivoId;
16   }
17   public set dispositivoId(value: number) {
18     this._dispositivoId = value;
19   }
20
21   public get nombre(): string {
22     return this._nombre;
23   }
24   public set nombre(value: string) {
25     this._nombre = value;
26   }
27
28   public get ubicacion(): string {
29     return this._ubicacion;
30   }
31   public set ubicacion(value: string) {
32     this._ubicacion = value;
33   }
34
35   public get electrovalvulaId(): number {
36     return this._electrovalvulaId;
37   }
38   public set electrovalvulaId(value: number) {
39     this._electrovalvulaId = value;
40   }
41 }
```

12.1c) NOTA - Esta clase es la que se usa en el TP, sólo que el service debería acceder a la Base de datos DAM. Está definida y por ahora lo que tenemos de eso es que la estructura y el dump inicial de prueba es:



## 12.1d) Base DAM (TP)

```
-- Table `DAM`.`Electrovalvulas`
DROP TABLE IF EXISTS `DAM`.`Electrovalvulas` ;

CREATE TABLE IF NOT EXISTS `DAM`.`Electrovalvulas` (
  `electrovalvulaId` INT NOT NULL AUTO_INCREMENT,
  `nombre` VARCHAR(45) NULL,
  PRIMARY KEY (`electrovalvulaId`))
ENGINE = InnoDB;

-- Table `DAM`.`Dispositivos`
DROP TABLE IF EXISTS `DAM`.`Dispositivos` ;

CREATE TABLE IF NOT EXISTS `DAM`.`Dispositivos` (
  `dispositivoId` INT NOT NULL AUTO_INCREMENT,
  `nombre` VARCHAR(200) NULL,
  `ubicacion` VARCHAR(200) NULL,
  `electrovalvulaId` INT NOT NULL,
  PRIMARY KEY (`dispositivoId`, `electrovalvulaId`),
  INDEX `fk_Dispositivos_Electrovalvulas1_idx` (`electrovalvulaId` ASC),
  CONSTRAINT `fk_Dispositivos_Electrovalvulas1`
    FOREIGN KEY (`electrovalvulaId`)
      REFERENCES `DAM`.`Electrovalvulas` (`electrovalvulaId`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
1 INSERT INTO DAM.Electrovalvulas (nombre) VALUES ('elPatio');
2 INSERT INTO DAM.Electrovalvulas (nombre) VALUES ('elCocina');
3 INSERT INTO DAM.Electrovalvulas (nombre) VALUES ('elJardinDelantero');
4 INSERT INTO DAM.Electrovalvulas (nombre) VALUES ('elLiving');
5 INSERT INTO DAM.Electrovalvulas (nombre) VALUES ('elHabitacion1');
6 INSERT INTO DAM.Electrovalvulas (nombre) VALUES ('elHabitacion2');
7
8
9 INSERT INTO DAM.Dispositivos (nombre,ubicacion,electrovalvulaId) VALUES ('Sensor 1', 'Patio',1);
10 INSERT INTO DAM.Dispositivos (nombre,ubicacion,electrovalvulaId) VALUES ('Sensor 2', 'Cocina',2);
11 INSERT INTO DAM.Dispositivos (nombre,ubicacion,electrovalvulaId) VALUES ('Sensor 3', 'Jardin Delantero',3);
12 INSERT INTO DAM.Dispositivos (nombre,ubicacion,electrovalvulaId) VALUES ('Sensor 4', 'Living',4);
13 INSERT INTO DAM.Dispositivos (nombre,ubicacion,electrovalvulaId) VALUES ('Sensor 5', 'Habitacion 1',5);
14 INSERT INTO DAM.Dispositivos (nombre,ubicacion,electrovalvulaId) VALUES ('Sensor 6', 'Habitacion 2',6);
15
16
17 INSERT INTO DAM.Mediciones (fecha,valor,dispositivoId) VALUES (current_timestamp(),60,1 );
18 INSERT INTO DAM.Mediciones (fecha,valor,dispositivoId) VALUES (current_timestamp(),40,1 );
19 INSERT INTO DAM.Mediciones (fecha,valor,dispositivoId) VALUES (current_timestamp(),30,2 );
20 INSERT INTO DAM.Mediciones (fecha,valor,dispositivoId) VALUES (current_timestamp(),50,3 );
21 INSERT INTO DAM.Mediciones (fecha,valor,dispositivoId) VALUES (current_timestamp(),33,5 );
22 INSERT INTO DAM.Mediciones (fecha,valor,dispositivoId) VALUES (current_timestamp(),17,4 );
23 INSERT INTO DAM.Mediciones (fecha,valor,dispositivoId) VALUES (current_timestamp(),29,6 );
24 INSERT INTO DAM.Mediciones (fecha,valor,dispositivoId) VALUES (current_timestamp(),20,1 );
25 INSERT INTO DAM.Mediciones (fecha,valor,dispositivoId) VALUES (current_timestamp(),44,4 );
26 INSERT INTO DAM.Mediciones (fecha,valor,dispositivoId) VALUES (current_timestamp(),61,5 );
27 INSERT INTO DAM.Mediciones (fecha,valor,dispositivoId) VALUES (current_timestamp(),12,2 );
28
```

La DAM.Mediciones tienen esta estructura:

```
-- Table `DAM`.`Mediciones`
DROP TABLE IF EXISTS `DAM`.`Mediciones` ;

CREATE TABLE IF NOT EXISTS `DAM`.`Mediciones` (
  `medicionId` INT NOT NULL AUTO_INCREMENT,
  `fecha` DATETIME NULL,
  `valor` VARCHAR(100) NULL,
  `dispositivoId` INT NOT NULL,
  PRIMARY KEY (`medicionId`, `dispositivoId`),
  INDEX `fk_Mediciones_Dispositivos_idx` (`dispositivoId` ASC),
  CONSTRAINT `fk_Mediciones_Dispositivos`
    FOREIGN KEY (`dispositivoId`)
      REFERENCES `DAM`.`Dispositivos` (`dispositivoId`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

La de DAM.Log\_Riegos:

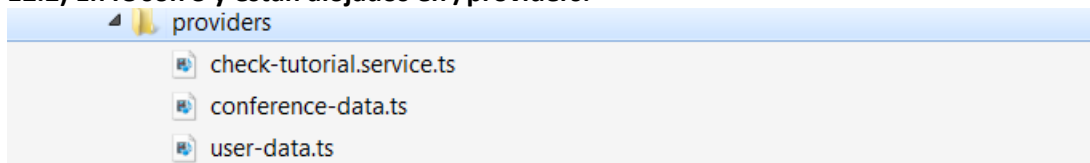
```
-- Table `DAM`.`Log_Riegos`
DROP TABLE IF EXISTS `DAM`.`Log_Riegos` ;

CREATE TABLE IF NOT EXISTS `DAM`.`Log_Riegos` (
  `logRiegoId` INT NOT NULL AUTO_INCREMENT,
  `apertura` TINYINT NULL,
  `fecha` DATETIME NULL,
  `electrovalvulaId` INT NOT NULL,
  PRIMARY KEY (`logRiegoId`, `electrovalvulaId`),
  INDEX `fk_Log_Riegos_Electrovalvulas1_idx` (`electrovalvulaId` ASC),
  CONSTRAINT `fk_Log_Riegos_Electrovalvulas1`
    FOREIGN KEY (`electrovalvulaId`)
      REFERENCES `DAM`.`Electrovalvulas` (`electrovalvulaId`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

Se hizo un Dump ampliado, para ver la inyección en la última tabla (verificar si así va):

```
27 INSERT INTO DAM.Mediciones (fecha,valor,dispositivoId) VALUES (current_timestamp(),12,2 );
28
29
30 INSERT INTO DAM.Log_Riegos (apertura, fecha, electrovalvulaId) VALUES (1, current_timestamp(),6);
31 INSERT INTO DAM.Log_Riegos (apertura, fecha, electrovalvulaId) VALUES (0, current_timestamp(),4);
32 INSERT INTO DAM.Log_Riegos (apertura, fecha, electrovalvulaId) VALUES (1, current_timestamp(),2);
```

## 12.2) En IC son 3 y están alojados en /providers:



Si bien son todos Servicios, La estructura de estos es bastante más compleja, por ahora sólo mostramos la del mas sencillo que es check-tutorial.ts:



```

check-tutorial.service.ts x DumpDatosTp(ampliado).sql x Dispositivo.ts x user-data.ts
1 import { Injectable } from '@angular/core';
2 import { CanLoad, Router } from '@angular/router';
3 import { Storage } from '@ionic/storage';
4 @Injectable({
5   providedIn: 'root'
6 })
7 export class CheckTutorial implements CanLoad {
8   constructor(private storage: Storage, private router: Router) {}
9
10  canLoad() {
11    return this.storage.get('ion_did_tutorial').then(res => {
12      if (res) {
13        this.router.navigate(['/app', 'tabs', 'schedule']);
14        return false;
15      } else {
16        return true;
17      }
18    });
19  }
20 }

```

## 13) Componentes / Angular y 14) Pages en Ionic

13.1) Cada componente consta de cuatro archivos, por ejemplo listado en AP – en realidad primero se genera el módulo que contiene al componente, en este caso se llaman listado:

|                        |       |
|------------------------|-------|
| listado                | 0,9 K |
| listado.component.css  | 0,0 K |
| listado.component.html | 0,3 K |
| listado.component.ts   | 0,4 K |
| listado.module.ts      | 0,3 K |
| model                  | 1.0 K |

El módulo “importa” el componente listado creado, aquí se llama ListadoComponent, que en AP es muy sencillo porque hace uso del único Service (Dispositivo) en su constructor.

```

check-tutorial.service.ts x listado.module.ts x
1 import { NgModule } from '@angular/core';
2 import { CommonModule } from '@angular/common';
3 import { ListadoComponent } from './listado.component';
4
5 @NgModule({
6   declarations: [ListadoComponent],
7   imports: [
8     CommonModule
9   ],
10  exports:[ListadoComponent]
11 })
12 export class ListadoModule {}

```

```

listado.module.ts x listado.components.ts x
1 import { Component, OnInit } from '@angular/core';
2 import { DispositivoService } from '../services/dispositivo.service';
3
4 @Component({
5   selector: 'app-listado',
6   templateUrl: './listado.component.html',
7   styleUrls: ['./listado.component.css']
8 })
9 export class ListadoComponent implements OnInit {
10
11   constructor(public dispositivoServ:DispositivoService) {
12
13   }
14
15   ngOnInit(): void {
16
17   }
18
19 }

```

(recordar de 8.1, que la App raíz es también un @NgModule, que a su vez importa a ListadoModule, como se muestra en la figura siguiente). Finalmente, listado.component.html es la estructura o template en HTML controlado por el archivo typescript listado.module.ts, haciendo a través del \*ngFor la iteración de los dispositivos, como en la Figura 1a) que se repite en la página siguiente. El .css está vacío:

De 8.1) app.module.ts

```

app.module.ts x app.module.ts x app.module.ts x
1 import { BrowserModule } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3
4 import { AppRoutingModule } from './app-routing.module';
5 import { AppComponent } from './app.component';
6 import { ListadoModule } from './listado/listado.module';
7
8 @NgModule({
9   declarations: [
10     AppComponent
11   ],
12   imports: [
13     BrowserModule,
14     AppRoutingModule,
15     ListadoModule
16   ],
17   providers: [],
18   bootstrap: [AppComponent]
19 })
20 export class AppModule {}

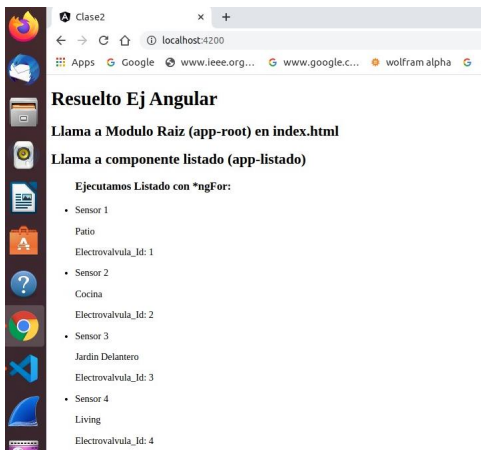
```

listado.component.html

```

listado.module.ts x listado.components.ts x listado.component.html x
1 <ul>
2   <h3>Ejecutamos listado con *ngFor:</h3>
3   <li *ngFor="let dispositivo of dispositivoServ.listado">
4     <p>{{dispositivo.nombre}}</p>
5     <p>{{dispositivo.ubicacion}}</p>
6     <p>Electrovalvula_Id: {{dispositivo.electrovalvulaId}}</p>
7   </li>
8 </ul>

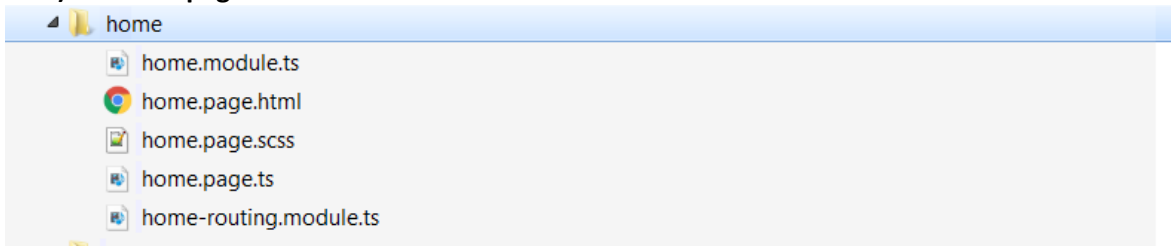
```



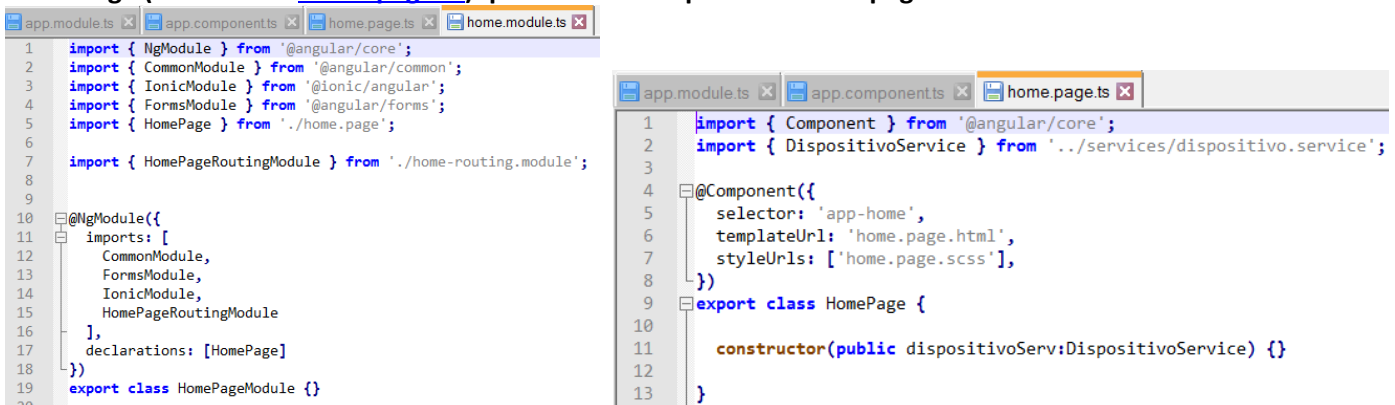
Listado de dispositivos, a través del \*ngFor en el html del comp.

14) En ionic, se usan las pages, pero son similares a los componentes. Vamos a ver el caso de IB que es el que usamos como base para el TP. Hay una home-page y una dispositivos page.

14.1) La home-page tiene 5 archivos:



14.1.1) Ahora el módulo se llama HomePageModule (definido en [home.module.ts](#)), y en como componente tengo HomePage (definido en [home.page.ts](#)) que tiene su template en home.page.html :



14.1.2) Ya el template es mas complejo que en el caso de AP, y además tiene un .css no vacío

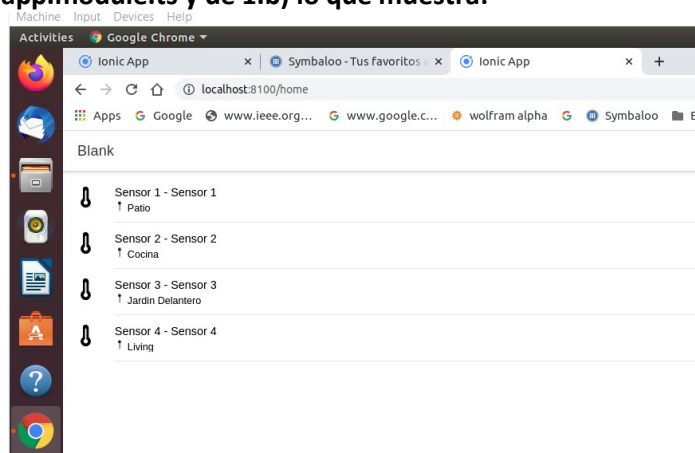


### 14.1.3) Hay un archivo adicional de ruteo:

```
home-routing.module.ts x home.page.scss x app.module.ts x app.component.ts x
1 import { NgModule } from '@angular/core';
2 import { RouterModule, Routes } from '@angular/router';
3 import { HomePage } from './home.page';
4
5 const routes: Routes = [
6   {
7     path: '',
8     component: HomePage,
9   }
10 ];
11
12 @NgModule({
13   imports: [RouterModule.forChild(routes)],
14   exports: [RouterModule]
15 })
16 export class HomePageRoutingModule {}
17
```

### 14.1.4) De 8.2) recordamos para IB (app raíz) el app.module.ts y de 1.b) lo que muestra:

```
app.module.ts x app.module.ts x app.module.ts x
1 import { NgModule } from '@angular/core';
2 import { BrowserModule } from '@angular/platform-browser';
3 import { RouteReuseStrategy } from '@angular/router';
4
5 import { IonicModule, IonicRouteStrategy } from '@ionic/angular';
6 import { SplashScreen } from '@ionic-native/splash-screen/ngx';
7 import { StatusBar } from '@ionic-native/status-bar/ngx';
8
9 import { AppComponent } from './app.component';
10 import { AppRoutingModule } from './app-routing.module';
11
12 @NgModule({
13   declarations: [AppComponent],
14   entryComponents: [],
15   imports: [BrowserModule, IonicModule.forRoot(), AppRoutingModule],
16   providers: [
17     StatusBar,
18     SplashScreen,
19     { provide: RouteReuseStrategy, useClass: IonicRouteStrategy }
20   ],
21   bootstrap: [AppComponent]
22 })
23 export class AppModule {}
24
```



## 14.2) La dispositivo-page tiene 5 archivos:

```
dispositivo
├── dispositivo.module.ts
├── dispositivo.page.html
├── dispositivo.page.scss
├── dispositivo.page.ts
└── dispositivo-routing.module.ts
```

14.2.1) Ahora el módulo se llama DispositivoPageModule (definido en [dispositivo.module.ts](#)), y en como componente tengo DispositivoPage (definido en [dispositivo.page.ts](#)) que tiene su template en dispositivo.page.html (el .css está vacío) :

a) Module page en dispositivo.module.ts

```
dispositivo.module.ts
1 import { NgModule } from '@angular/core';
2 import { CommonModule } from '@angular/common';
3 import { FormsModule } from '@angular/forms';
4
5 import { IonicModule } from '@ionic/angular';
6
7 import { DispositivoPageRoutingModule } from './dispositivo-routing.module';
8
9 import { DispositivoPage } from './dispositivo.page';
10
11 @NgModule({
12   imports: [
13     CommonModule,
14     FormsModule,
15     IonicModule,
16     DispositivoPageRoutingModule
17   ],
18   declarations: [DispositivoPage]
19 })
20 export class DispositivoPageModule {}
```

b) Componente = page en dispositivo.page.ts

```
dispositivo.module.ts | dispositivo.page.ts
1 import { Component, OnInit } from '@angular/core';
2 import { ActivatedRoute } from '@angular/router';
3 import { Dispositivo } from '../model/Dispositivo';
4 import { DispositivoService } from '../services/dispositivo.service';
5
6 @Component({
7   selector: 'app-dispositivo',
8   templateUrl: './dispositivo.page.html',
9   styleUrls: ['./dispositivo.page.scss'],
10 })
11 export class DispositivoPage implements OnInit {
12
13   public dispositivo: Dispositivo;
14   constructor(private router: ActivatedRoute, private dServ: DispositivoService) {}
15
16   ngOnInit() {
17     let idDispositivo = this.router.snapshot.paramMap.get('id');
18     this.dispositivo = this.dServ.getDispositivo(idDispositivo);
19     console.log(this.dispositivo);
20   }
21
22   ionViewWillEnter(){
23
24   }
25
26 }
```

c) Template dispositivo.page.html

```
dispositivo.module.ts | dispositivo.page.ts | dispositivo.page.html
1 <ion-header>
2   <ion-toolbar>
3     <ion-buttons slot="start">
4       <ion-button routerLink="/">
5         <ion-icon slot="icon-only" name="arrow-back"></ion-icon>
6       </ion-button>
7     </ion-buttons>
8     <ion-title>Dispositivo </ion-title>
9   </ion-toolbar>
10 </ion-header>
11
12 <ion-content>
13   <div>Acá debería traer mas información del dispositivo y su medida del sensor</div>
14   <div>
15     {{dispositivo.ubicacion}}
16   </div>
17 </ion-content>
```

### 14.2.2) El dispositivo-routing.module.ts

```
dispositivo-routing.module.ts | dispositivo.module.ts | dispositivo.page.ts | dispositivo.page.html
1 import { NgModule } from '@angular/core';
2 import { Routes, RouterModule } from '@angular/router';
3
4 import { DispositivoPage } from './dispositivo.page';
5
6 const routes: Routes = [
7   {
8     path: '',
9     component: DispositivoPage
10  }
11 ];
12
13 @NgModule({
14   imports: [RouterModule.forChild(routes)],
15   exports: [RouterModule],
16 })
17 export class DispositivoPageRoutingModule {}
18
```

Muy similar al del home-routing.module.ts visto en el punto 13,

```
home-routing.module.ts | home.page.scss | app.module.ts | app.component.ts
1 import { NgModule } from '@angular/core';
2 import { RouterModule, Routes } from '@angular/router';
3 import { HomePage } from './home.page';
4
5 const routes: Routes = [
6   {
7     path: '',
8     component: HomePage,
9   }
10 ];
11
12 @NgModule({
13   imports: [RouterModule.forRoot(routes)],
14   exports: [RouterModule]
15 })
16 export class HomePageRoutingModule {}
17
```

En 11.2) se mostró el módulo de ruteo de la app raíz IB, que contiene a HomePageModule definido en 14.1.1) y al DispositivoPageModule definido en 14.2.1).

```
app-routing.module.ts | app-routing.module.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: 'home',
7     loadChildren: () => import('./home/home.module').then( m => m.HomePageModule)
8   },
9   {
10    path: '',
11    redirectTo: 'home',
12    pathMatch: 'full'
13  },
14  {
15    path: 'dispositivo/:id',
16    loadChildren: () => import('./dispositivo/dispositivo.module').then( m => m.DispositivoPageModule)
17  },
18 ];
19
20 @NgModule({
21   imports: [
22     RouterModule.forRoot(routes, { preloadingStrategy: PreloadAllModules })
23   ],
24   exports: [RouterModule]
25 })
26 export class AppRoutingModule {}
27
```

## 15) Aplicación IC Conference:

Aquí se ve un detalle de componentes de cada page. A los nombres se les sacó la terminación .page (no sabemos si automáticamente).

|                           |         |
|---------------------------|---------|
| pages                     | 65,2 KB |
| about                     | 5,7 KB  |
| about.html                | 2,9 KB  |
| about.module.ts           | 0,6 KB  |
| about.scss                | 1,3 KB  |
| about.ts                  | 0,6 KB  |
| about-routing.module.ts   | 0,3 KB  |
| about-popover             | 1,1 KB  |
| account                   | 3,2 KB  |
| account.html              | 0,7 KB  |
| account.module.ts         | 0,4 KB  |
| account.scss              | 0,0 KB  |
| account.ts                | 1,6 KB  |
| account-routing.module.ts | 0,3 KB  |
| login                     | 3,2 KB  |
| login.html                | 1,5 KB  |
| login.module.ts           | 0,5 KB  |
| login.scss                | 0,1 KB  |
| login.ts                  | 0,8 KB  |
| login-routing.module.ts   | 0,3 KB  |
| map                       | 6,5 KB  |
| map.html                  | 0,3 KB  |
| map.module.ts             | 0,4 KB  |
| map.scss                  | 0,2 KB  |
| map.ts                    | 2,9 KB  |
| map-dark-style.js         | 2,4 KB  |
| map-routing.module.ts     | 0,3 KB  |
| schedule                  | 9,7 KB  |

### 15.1) Page login → detalle: La login-page tiene 5 archivos:

|                         |
|-------------------------|
| login                   |
| login.html              |
| login.module.ts         |
| login.scss              |
| login.ts                |
| login-routing.module.ts |
| map                     |

15.1.1) Ahora el módulo se llama LoginModule (definido en [login.module.ts](#)), y en como componente tengo Login (definido en [login.ts](#)) que tiene su template en login.html (el .scss es breve) :

a) Module page en login.module.ts

```
login.module.ts x login.html x login-routing.module.ts x login.ts x login.scss
1 import { NgModule } from '@angular/core';
2 import { CommonModule } from '@angular/common';
3 import { FormsModule } from '@angular/forms';
4 import { IonicModule } from '@ionic/angular';
5
6 import { LoginPage } from './login';
7 import { LoginPageRoutingModule } from './login-routing.module';
8
9 @NgModule({
10   imports: [
11     CommonModule,
12     FormsModule,
13     IonicModule,
14     LoginPageRoutingModule
15   ],
16   declarations: [
17     LoginPage,
18   ]
19 })
20 export class LoginModule { }
```

## b) Componente = page en login.ts, y login.scss (muy breve)

login.module.ts
login.html
login-routing.module.ts
login.ts
login.scss

```

1  import { Component } from '@angular/core';
2  import { NgForm } from '@angular/forms';
3  import { Router } from '@angular/router';
4  import { UserData } from '../providers/user-data';
5  import { UserOptions } from '../interfaces/user-options';
6
7  @Component({
8    selector: 'page-login',
9    templateUrl: 'login.html',
10   styleUrls: ['./login.scss'],
11 })
12 export class LoginPage {
13   login: UserOptions = { username: '', password: '' };
14   submitted = false;
15
16   constructor(
17     public userData: UserData,
18     public router: Router
19   ) { }
20
21   onLogin(form: NgForm) {
22     this.submitted = true;
23
24     if (form.valid) {
25       this.userData.login(this.login.username);
26       this.router.navigateByUrl('/app/tabs/schedule');
27     }
28   }
29
30   onSignup() {
31     this.router.navigateByUrl('/signup');
32   }
33 }

```

login.scss
login.module.ts

```

1  .login-logo {
2    padding: 20px 0;
3    min-height: 200px;
4    text-align: center;
5  }
6
7  .login-logo img {
8    max-width: 150px;
9  }
10
11  .list {
12    margin-bottom: 0;
13  }
14

```

## c) Template login.page.html

login.module.ts
login.html
login-routing.module.ts
login.ts
login.scss

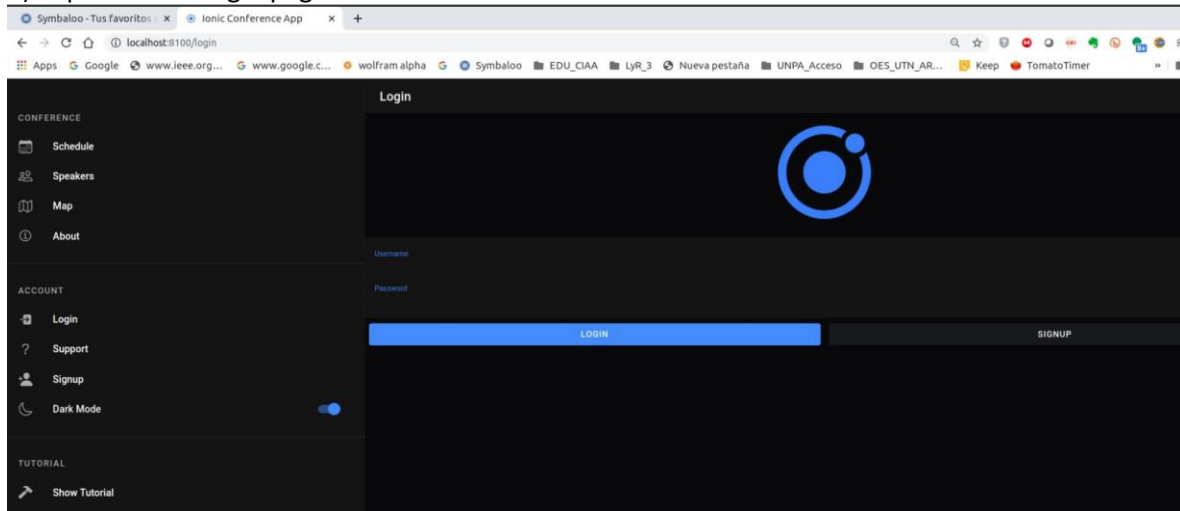
```

1  <ion-header>
2    <ion-toolbar>
3      <ion-buttons slot="start">
4        <ion-menu-button></ion-menu-button>
5      </ion-buttons>
6
7      <ion-title>Login</ion-title>
8    </ion-toolbar>
9  </ion-header>
10
11  <ion-content>
12    <div class="login-logo">
13      
14    </div>
15
16    <form #loginForm="ngForm" novalidate>
17      <ion-list>
18        <ion-item>
19          <ion-label position="stacked" color="primary">Username</ion-label>
20          <ion-input [(ngModel)]="login.username" name="username" type="text" #username="ngModel" spellcheck="false" autocapitalize="off"
21            required>
22          </ion-input>
23        </ion-item>
24
25        <ion-text color="danger">
26          <p [hidden]="username.valid || submitted == false" class="ion-padding-start">
27            Username is required
28          </p>
29        </ion-text>
30
31        <ion-item>
32          <ion-label position="stacked" color="primary">Password</ion-label>
33          <ion-input [(ngModel)]="login.password" name="password" type="password" #password="ngModel" required>
34          </ion-input>
35        </ion-item>
36
37        <ion-text color="danger">
38          <p [hidden]="password.valid || submitted == false" class="ion-padding-start">
39            Password is required
40          </p>
41        </ion-text>
42      </ion-list>
43
44      <ion-row>
45        <ion-col>
46          <ion-button (click)="onLogin(loginForm)" type="submit" expand="block">Login</ion-button>
47        </ion-col>
48        <ion-col>
49          <ion-button (click)="onSignup()" color="light" expand="block">Signup</ion-button>
50        </ion-col>
51      </ion-row>
52    </form>
53  </ion-content>

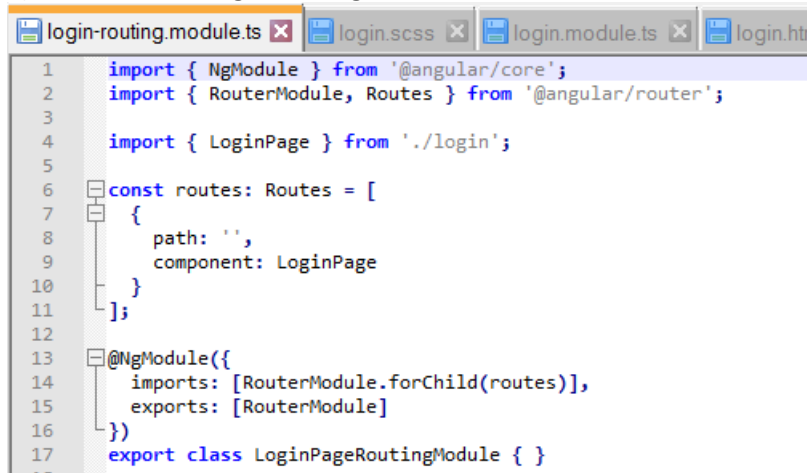
```



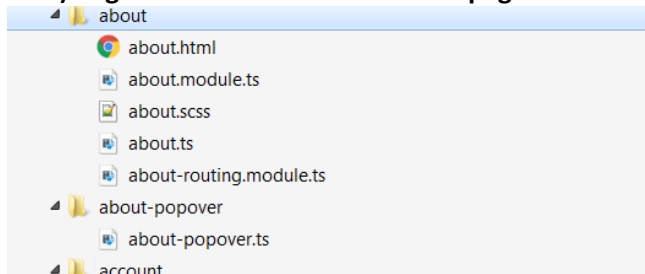
#### d) Aspecto de la login page:



#### e) Modulo ruteo login-routing.module.ts

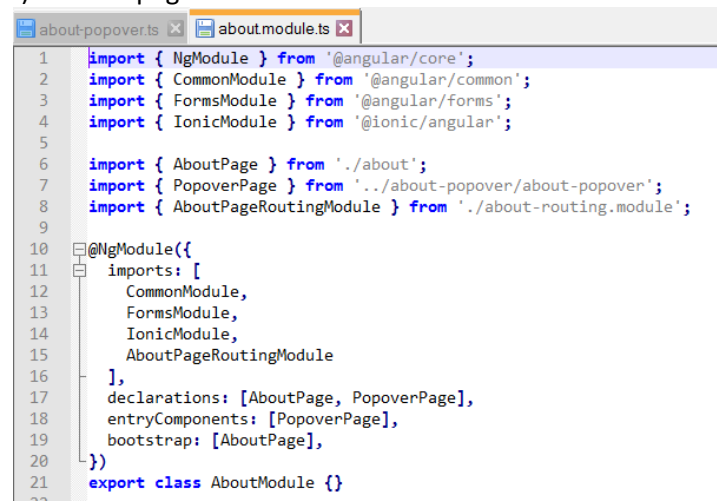


#### 15.2) Page about → detalle: La about-page tiene 5 archivos, y un directorio anexo “popover”:



#### 15.2.1) Ahora el módulo se llama AboutModule (definido en [about.module.ts](#)), y como componente tengo About (definido en [about.ts](#)) que tiene su template en about.html (el .scss es breve) :

##### a) Module page en about.module.ts



b) Componente = page en about.ts, y about.scss (muy breve)

```
about-popover.ts | aboutmodule.ts | aboutts
1  import { Component } from '@angular/core';
2
3  import { PopoverController } from '@ionic/angular';
4
5  import { PopoverPage } from '../about-popover/about-popover'
6
7  @Component({
8    selector: 'page-about',
9    templateUrl: 'about.html',
10   styleUrls: ['./about.scss'],
11 })
12 export class AboutPage {
13   location = 'madison';
14   conferenceDate = '2047-05-17';
15
16   selectOptions = {
17     header: 'Select a Location'
18   };
19
20   constructor(public popoverCtrl: PopoverController) { }
21
22   async presentPopover(event: Event) {
23     const popover = await this.popoverCtrl.create({
24       component: PopoverPage,
25       event
26     });
27     await popover.present();
28   }
29 }
30
```

15.2.3) c) Template about.page.html

c.1) Imagen y About

```
about.html | about-popover.ts | aboutmodule.ts | aboutts
1  <ion-content>
2    <ion-header class="ion-no-border">
3      <ion-toolbar>
4        <ion-buttons slot="start">
5          <ion-menu-button></ion-menu-button>
6        </ion-buttons>
7        <ion-buttons slot="end">
8          <ion-button (click)="presentPopover($event)">
9            <ion-icon slot="icon-only" ios="ellipsis-horizontal" md="ellipsis-vertical"></ion-icon>
10          </ion-button>
11        </ion-buttons>
12      </ion-toolbar>
13    </ion-header>
14
15    <div class="about-header">
16      <!-- Instead of loading an image each time the select changes, use opacity to transition them -->
17      <div class="about-image madison" [ngStyle]="location === 'madison' && {'opacity': '1'}"></div>
18      <div class="about-image austin" [ngStyle]="location === 'austin' && {'opacity': '1'}"></div>
19      <div class="about-image chicago" [ngStyle]="location === 'chicago' && {'opacity': '1'}"></div>
20      <div class="about-image seattle" [ngStyle]="location === 'seattle' && {'opacity': '1'}"></div>
21    </div>
22
23    <div class="about-info">
24      <h3 class="ion-padding-top ion-padding-start">About</h3>
25
26      <p class="ion-padding-start ion-padding-end">
27        The Ionic Conference is a one-day conference on {{ conferenceDate | date: 'mediumDate' }} featuring talks
28        from the Ionic team. It is focused on Ionic applications being built with Ionic Framework. This includes
29        migrating apps to the latest version of the framework, Angular concepts, Webpack, Sass, and many other
30        technologies used in Ionic 2. Tickets are completely sold out, and we're expecting more than
31        1000 developers - making this the largest Ionic conference ever!
32      </p>
33    </div>
34  </ion-content>
```

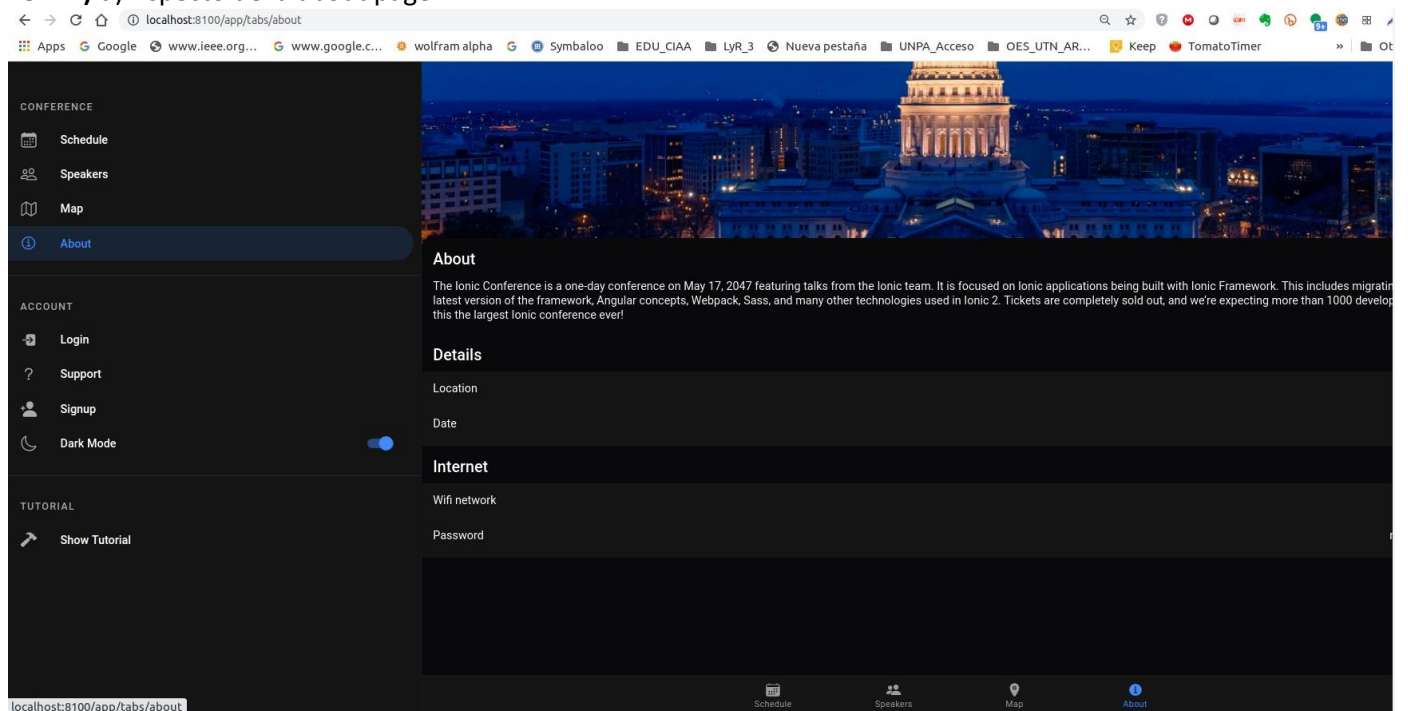
## c.2) Details e Internet

```

33
34 <h3 class="ion-padding-top ion-padding-start">Details</h3>
35
36 <ion-list lines="none">
37   <ion-item>
38     <ion-label>
39       Location
40     </ion-label>
41     <ion-select [(ngModel)]="location" [interfaceOptions]="selectOptions">
42       <ion-select-option value="madison">Madison, WI</ion-select-option>
43       <ion-select-option value="austin">Austin, TX</ion-select-option>
44       <ion-select-option value="chicago">Chicago, IL</ion-select-option>
45       <ion-select-option value="seattle">Seattle, WA</ion-select-option>
46     </ion-select>
47   </ion-item>
48   <ion-item>
49     <ion-label>
50       Date
51     </ion-label>
52     <ion-datetime
53       displayFormat="MMM DD, YYYY"
54       max="2056"
55       [(ngModel)]="conferenceDate">
56     </ion-datetime>
57   </ion-item>
58 </ion-list>
59
60 <h3 class="ion-padding-top ion-padding-start">Internet</h3>
61
62 <ion-list lines="none">
63   <ion-item>
64     <ion-label>
65       Wifi network
66     </ion-label>
67     <ion-label class="ion-text-end">
68       {{conferenceDate | date: 'y'}}
69     </ion-label>
70   </ion-item>
71   <ion-item>
72     <ion-label>
73       Password
74     </ion-label>
75     <ion-label class="ion-text-end">
76       makegoodthings
77     </ion-label>
78   </ion-item>
79 </ion-list>
80 </div>
81 </ion-content>

```

## 15.2.4) d) Aspecto de la about page:



### 15.2.5) e) Modulo ruteo about-routing.module.ts

```
about-routing.module.ts | about.html | about-popover.ts | about.module.ts
1 import { NgModule } from '@angular/core';
2 import { RouterModule, Routes } from '@angular/router';
3
4 import { AboutPage } from './about';
5
6 const routes: Routes = [
7   {
8     path: '',
9     component: AboutPage
10  }
11 ];
12
13 @NgModule({
14   imports: [RouterModule.forChild(routes)],
15   exports: [RouterModule]
16 })
17 export class AboutPageRoutingModule { }
18
```

### f) About popover:

```
about-popover.ts
1 import { Component } from '@angular/core';
2
3 import { PopoverController } from '@ionic/angular';
4
5 @Component({
6   template: `
7     <ion-list>
8       <ion-item button (click)="close('https://ionicframework.com/getting-started')">
9         <ion-label>Learn Ionic</ion-label>
10      </ion-item>
11       <ion-item button (click)="close('https://ionicframework.com/docs/')">
12         <ion-label>Documentation</ion-label>
13      </ion-item>
14       <ion-item button (click)="close('https://showcase.ionicframework.com')">
15         <ion-label>Showcase</ion-label>
16      </ion-item>
17       <ion-item button (click)="close('https://github.com/ionic-team/ionic')">
18         <ion-label>GitHub Repo</ion-label>
19      </ion-item>
20       <ion-item button (click)="support()">
21         <ion-label>Support</ion-label>
22      </ion-item>
23     </ion-list>
24   `
25 })
26 export class PopoverPage {
27   constructor(public popoverCtrl: PopoverController) {}
28
29   support() {
30     // this.app.getRootNavs()[0].push('/support');
31     this.popoverCtrl.dismiss();
32   }
33
34   close(url: string) {
35     window.open(url, '_blank');
36     this.popoverCtrl.dismiss();
37   }
38 }
39
```

## ANEXO APLICACIÓN MUSIC

1) Aplicación ejemplo de Ionic, pero librerías antiguas (no anduvo)

2) Detalle Assets y files dentro de /app:

|                     |          |    |
|---------------------|----------|----|
| app                 | 4,7 KB   |    |
| assets              | 0,0 KB   |    |
| .gitkeep            | 0,0 KB   |    |
| environments        | 0,7 KB   |    |
| environment.prod.ts | 0,0 KB   |    |
| environment.ts      | 0,6 KB   |    |
| [Files]             | 5,2 KB   |    |
| favicon.ico         | 0,9 KB   |    |
| index.html          | 0,4 KB   |    |
| main.ts             | 0,4 KB   |    |
| polyfills.ts        | 2,8 KB   |    |
| styles.css          | 0,1 KB   |    |
| test.ts             | 0,7 KB   |    |
| [Files]             | 519,4 KB | 54 |
| angular.json        | 3,6 KB   |    |
| karma.conf.js       | 1,0 KB   |    |
| package.json        | 1,2 KB   |    |
| package-lock.json   | 507,9 KB | 50 |
| README.md           | 1,0 KB   |    |
| tsconfig.app.json   | 0,3 KB   |    |
| tsconfig.base.json  | 0,4 KB   |    |
| tsconfig.json       | 0,5 KB   |    |
| tsconfig.spec.json  | 0,3 KB   |    |
| tslint.json         | 3,1 KB   |    |

3) Detalle directorio raíz de proyecto ionic:

|                    |          |
|--------------------|----------|
| [Files]            | 488,5 KB |
| .angulardoc.json   | 0,1 KB   |
| .gitignore         | 0,4 KB   |
| angular.json       | 5,1 KB   |
| browserslist       | 0,4 KB   |
| config.xml         | 7,6 KB   |
| ionic.config.json  | 0,1 KB   |
| karma.conf.js      | 1,0 KB   |
| package.json       | 2,7 KB   |
| package-lock.json  | 467,8 KB |
| README.md          | 0,5 KB   |
| tsconfig.app.json  | 0,2 KB   |
| tsconfig.json      | 0,5 KB   |
| tsconfig.spec.json | 0,3 KB   |
| tslint.json        | 1,8 KB   |

4) Aspecto:

5) Primera parte de Config.xml (después siguen cosas específicas de Android e IOS)

```

1 <?xml version='1.0' encoding='utf-8'?>
2 <widget id="io.ionic.starter" version="0.0.1" xmlns="http://www.w3.org/ns/widgets" xmlns:cdv="http://cordova.apache.org/ns/1.0">
3   <name>MyApp</name>
4   <description>An awesome Ionic/Cordova app.</description>
5   <author email="hi@ionicframework.com" href="http://ionicframework.com/">Ionic Framework Team</author>
6   <content src="index.html" />
7   <access origin="*" />
8   <allow-intent href="http://*/*" />
9   <allow-intent href="https://*/*" />
10  <allow-intent href="tel:*" />
11  <allow-intent href="sms:*" />
12  <allow-intent href="mailto:*" />
13  <allow-intent href="geo:*" />
14  <preference name="ScrollEnabled" value="false" />
15  <preference name="android-minSdkVersion" value="19" />
16  <preference name="BackupWebStorage" value="none" />
17  <preference name="SplashMaintainAspectRatio" value="true" />
18  <preference name="FadeSplashScreenDuration" value="300" />
19  <preference name="SplashShowOnlyFirstTime" value="false" />
20  <preference name="SplashScreen" value="screen" />
21  <preference name="SplashScreenDelay" value="3000" />
22  <platform name="android">

```

## 6) Primera parte de packages.json

```

1 {
2   "name": "ionic-sqlite-app",
3   "version": "0.0.1",
4   "author": "Ionic Framework",
5   "homepage": "https://ionicframework.com/",
6   "scripts": {
7     "ng": "ng",
8     "start": "ng serve",
9     "build": "ng build",
10    "test": "ng test",
11    "lint": "ng lint",
12    "e2e": "ng e2e"
13  },
14  "private": true,
15  "dependencies": {
16    "@angular/common": "~8.1.2",
17    "@angular/core": "~8.1.2",
18    "@angular/forms": "~8.1.2",
19    "@angular/platform-browser": "~8.1.2",
20    "@angular/platform-browser-dynamic": "~8.1.2",
21    "@angular/router": "~8.1.2",
22    "@ionic-native/core": "^5.23.0",
23    "@ionic-native/splash-screen": "^5.23.0",
24    "@ionic-native/sqlite": "^5.23.0",
25    "@ionic-native/sqlite-porter": "^5.23.0",
26    "@ionic-native/status-bar": "^5.23.0",
27    "@ionic/angular": "^5.0.5",
28    "cordova-android": "8.1.0",
29    "cordova-browser": "^6.0.0",
30    "cordova-sqlite-storage": "^4.0.0",
31    "core-js": "^2.5.4",
32    "rxjs": "~6.5.1",
33    "tslib": "^1.11.1",
34    "uk.co.workingedge.cordova.plugin.sqliteporter": "^1.1.1",
35    "zone.js": "~0.9.1"
36  },

```

## 7) ionic.config.json

```

1 {
2   "name": "ionic-sqlite-app",
3   "integrations": {
4     "cordova": {}
5   },
6   "type": "angular"
7 }
8

```

## 8) Primera parte de angular.json

```

1 {
2   "$schema": "./node_modules/@angular/cli/lib/config/schema.json",
3   "version": 1,
4   "defaultProject": "app",
5   "newProjectRoot": "projects",
6   "projects": {
7     "app": {
8       "root": "",
9       "sourceRoot": "src",
10      "projectType": "application",
11      "prefix": "app",
12      "schematics": {},
13      "architect": {
14        "build": {
15          "builder": "@angular-devkit/build-angular:browser",
16          "options": {
17            "outputPath": "www",
18            "index": "src/index.html",
19            "main": "src/main.ts",
20            "polyfills": "src/polyfills.ts",
21            "tsConfig": "tsconfig.app.json",
22            "assets": [
23              {
24                "glob": "**/*",
25                "input": "src/assets",
26                "output": "assets"
27              }
28            ]

```

## 9) index.html , donde en el body sólo se llama a <app-root>

```

1 <html lang="en">
2
3
4 <head>
5   <meta charset="utf-8" />
6   <title>Ionic App</title>
7
8   <base href="/" />
9
10  <meta name="viewport" content="viewport-fit=cover, width=device-width, initia
11  <meta name="format-detection" content="telephone=no" />
12  <meta name="msapplication-tap-highlight" content="no" />
13
14  <link rel="icon" type="image/png" href="assets/icon/favicon.png" />
15
16  <!-- add to homescreen for ios -->
17  <meta name="apple-mobile-web-app-capable" content="yes" />
18  <meta name="apple-mobile-web-app-status-bar-style" content="black" />
19 </head>
20
21 <body>
22   <app-root></app-root>
23 </body>
24
25 </html>

```

10) main.ts es el controlador base → básicamente solo llama a AppModule dentro de /app

```

1 import { enableProdMode } from '@angular/core';
2 import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';
3
4 import { AppModule } from './app/app.module';
5 import { environment } from './environments/environment';
6
7 if (environment.production) {
8   enableProdMode();
9 }
10
11 platformBrowserDynamic().bootstrapModule(AppModule)
12   .catch(err => console.log(err));

```

11) Global.scss es el global stylesheet, para toda la aplicacion:

```

1 /*
2  * App Global CSS
3  * -----
4  * Put style rules here that you want to apply globally. These styles are for
5  * the entire app and not just one component. Additionally, this file can be
6  * used as an entry point to import other CSS/Sass files to be included in the
7  * output CSS.
8  * For more information on global stylesheets, visit the documentation:
9  * https://ionicframework.com/docs/layout/global-stylesheets
10  */
11
12 /* Core CSS required for Ionic components to work properly */
13 @import "~@ionic/angular/css/core.css";
14
15 /* Basic CSS for apps built with Ionic */
16 @import "~@ionic/angular/css/normalize.css";
17 @import "~@ionic/angular/css/structure.css";
18 @import "~@ionic/angular/css/typography.css";
19 @import "~@ionic/angular/css/display.css";
20
21 /* Optional CSS utils that can be commented out */
22 @import "~@ionic/angular/css/padding.css";
23 @import "~@ionic/angular/css/float-elements.css";
24 @import "~@ionic/angular/css/text-alignment.css";
25 @import "~@ionic/angular/css/text-transformation.css";
26 @import "~@ionic/angular/css/flex-utils.css";

```

12) Los dos archivos de /environments se refieren a la versión de producción:



```
DumpDatosTp.sql x .gitignore x global.scss x environment.ts x environment.prod.ts x
1 export const environment = {
2   production: true
3 };
4
```

13) Dentro de /themes, está el variables.scss de Ionic:

```
DumpDatosTp.sql x .gitignore x global.scss x environment.ts x environment.prod.ts x dump.sql x variables.scss x
1 // Ionic Variables and Theming. For more info, please see:
2 // http://ionicframework.com/docs/theming/
3
4 /** Ionic CSS Variables */
5 :root {
6   /** primary */
7   --ion-color-primary: #3880ff;
8   --ion-color-primary-rgb: 56, 128, 255;
9   --ion-color-primary-contrast: #ffffff;
10  --ion-color-primary-contrast-rgb: 255, 255, 255;
11  --ion-color-primary-shade: #3171e0;
12  --ion-color-primary-tint: #4c8dff;
13
14  /** secondary */
15  --ion-color-secondary: #0cd1e8;
16  --ion-color-secondary-rgb: 12, 209, 232;
17  --ion-color-secondary-contrast: #ffffff;
18  --ion-color-secondary-contrast-rgb: 255, 255, 255;
```

14) Dentro de /assets, hay imágenes, y lo mas relevante es la base de datos SQLite con la inserción de 8 canciones:

```
DumpDatosTp.sql x .gitignore x global.scss x environment.ts x environment.prod.ts x dump.sql x
1 CREATE TABLE IF NOT EXISTS songtable(
2   id INTEGER PRIMARY KEY AUTOINCREMENT,
3   artist_name TEXT,
4   song_name TEXT
5 );
6
7 INSERT or IGNORE INTO songtable(id, artist_name, song_name) VALUES (1, 'Justin Bieber', 'Yummy');
8 INSERT or IGNORE INTO songtable(id, artist_name, song_name) VALUES (2, 'Jonas Brothers', 'What A Man Gotta Do');
9 INSERT or IGNORE INTO songtable(id, artist_name, song_name) VALUES (3, 'Life Is Good', 'Future');
10 INSERT or IGNORE INTO songtable(id, artist_name, song_name) VALUES (4, 'Lauv', 'Tattoos Together');
11 INSERT or IGNORE INTO songtable(id, artist_name, song_name) VALUES (5, 'Heavy Steppers', 'Whateva');
12 INSERT or IGNORE INTO songtable(id, artist_name, song_name) VALUES (6, 'DigDat 2020', 'Eight Mile');
13 INSERT or IGNORE INTO songtable(id, artist_name, song_name) VALUES (7, 'Blackbear', 'me & ur ghost');
14 INSERT or IGNORE INTO songtable(id, artist_name, song_name) VALUES (8, 'Hailee Steinfeld', 'Wrong Direction');
15
```

-0-