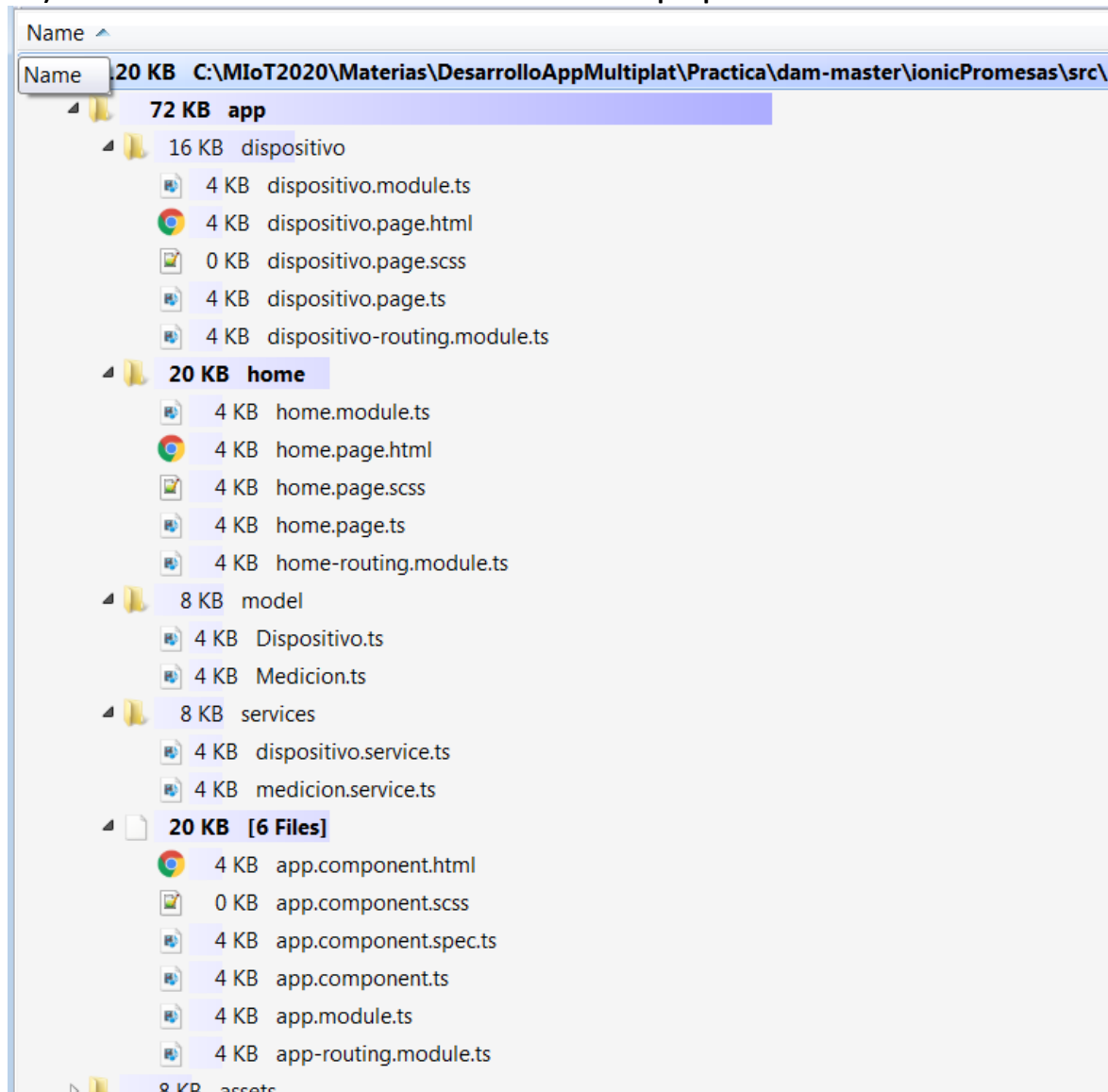


# Ionic Promesas resuelto - R.Oliva 08.2020

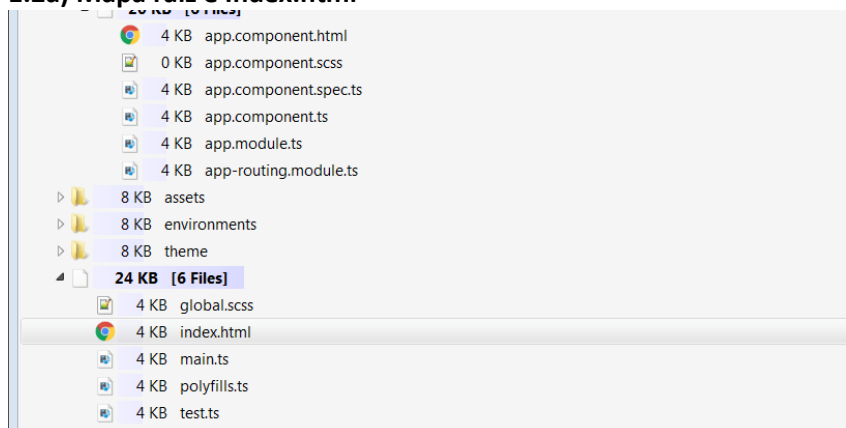
## Resuelto Promesas upd16-08-20. Se usa como base para el TP

### 1) Resuelto Ionic Promesas

#### 1.1) La estructura de este front End es similar al IB descripto previamente.



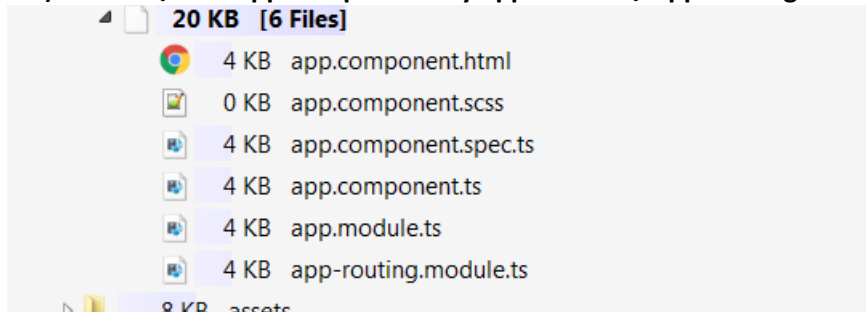
#### 1.2a) Mapa raíz e Index.html -



**1.2b) Aspecto de 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.

```
index.html x app.module.ts x
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,
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>
```

**1.3) Raiz del /src – app.component.\* y app.module / app-routing.module**



**1.3a) app.component.html**

```
app.component.html x index.html x app.module.ts x
1 <ion-app>
2   <ion-router-outlet></ion-router-outlet>
3 </ion-app>
4
```

**1.3b) app.component.ts**

```
app.components x app.module.ts x app.component.html x index.html x
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 }
```

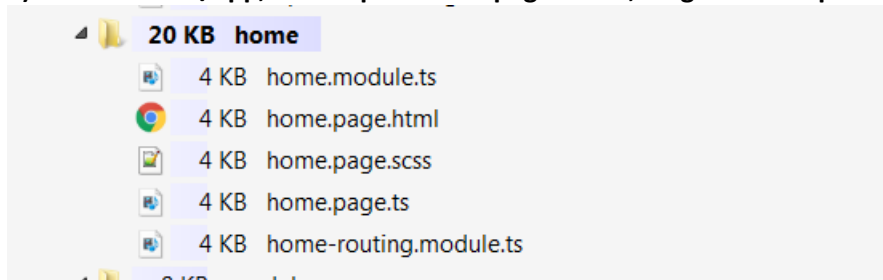
### 1.3c) app.module.ts / respecto a IB, se incluye ahora el HttpClientModule

```
app.module.ts x app.components x app.component.html x index.html 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 import { HttpClientModule } from '@angular/common/http';
12
13 @NgModule({
14   declarations: [AppComponent],
15   entryComponents: [],
16   imports: [BrowserModule, IonicModule.forRoot(), AppRoutingModule, HttpClientModule],
17   providers: [
18     StatusBar,
19     SplashScreen,
20     { provide: RouteReuseStrategy, useClass: IonicRouteStrategy }
21   ],
22   bootstrap: [AppComponent]
23 })
24 export class AppModule {}
```

### 1.3d) app-routing.module.ts (similar a IB)

```
app-routing.module.ts x app.module.ts x app.components x app.component.html x index.html x
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 { }
```

### 2) Ya dentro de /app, vemos primero la page home, luego la de dispositivo:



## 2.1) IoO - home.module.ts

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

## 2.2) IoO - home.page.ts

```
home.page.ts
1 import { Component } from '@angular/core';
2 import { DispositivoService } from '../services/dispositivo.service';
3 import { Dispositivo } from '../model/dispositivo';
4
5 @Component({
6   selector: 'app-home',
7   templateUrl: 'home.page.html',
8   styleUrls: ['home.page.scss'],
9 })
10 export class HomePage {
11
12   listadoDispositivo: Dispositivo[];
13
14   constructor(public dispositivoServ: DispositivoService) {
15     dispositivoServ.getListadoDispositivos().then(lst => {
16       this.listadoDispositivo = lst;
17     })
18   }
19
20 }
21
```

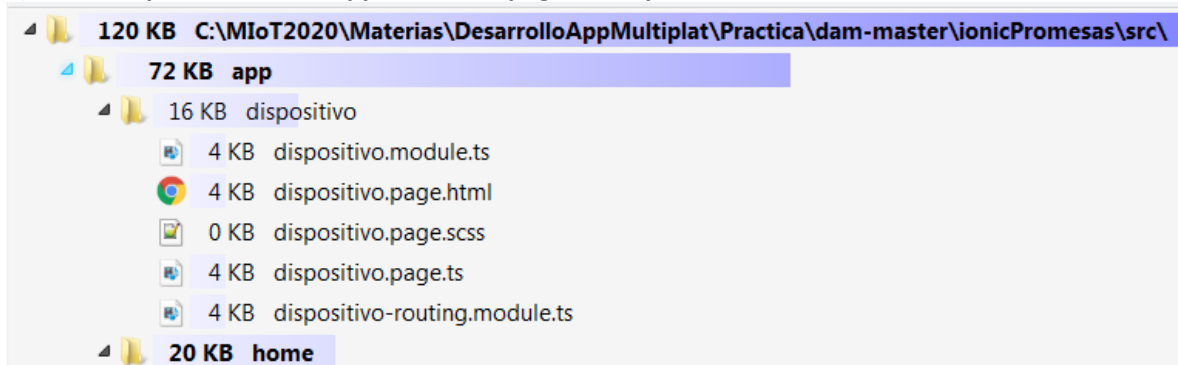
## 2.3) IoO - home.page.html

```
home.page.html
1 <ion-header [translucent]="true">
2   <ion-toolbar>
3     <ion-title>
4       Blank
5     </ion-title>
6   </ion-toolbar>
7 </ion-header>
8
9 <ion-content [fullscreen]="true">
10   <ion-header collapse="condense">
11     <ion-toolbar>
12       <ion-title size="large">Blank</ion-title>
13     </ion-toolbar>
14   </ion-header>
15   <ion-list>
16     <ion-item *ngFor="let dispositivo of this.listadoDispositivo" routerLink="/dispositivo/{{dispositivo.dispositivoId}}" routerDirection="root">
17       <ion-avatar slot="start">
18         <ion-icon size="large" name="thermometer"></ion-icon>
19       </ion-avatar>
20       <ion-label>
21         <h2>Sensor {{dispositivo.dispositivoId}} - {{dispositivo.nombre}}</h2>
22         <h3>
23           <ion-icon name="pin"></ion-icon> {{dispositivo.ubicacion}}</h3>
24       </ion-label>
25     </ion-item>
26   </ion-list>
27 </ion-content>
```

## 2.4) IoO - home-routing.module.ts

```
home-routing.module.ts x home.page.html x home.page.ts x home.module.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
```

## 3) IoO Siempre dentro de /app, vemos la page de dispositivo:



## 3.1) IoO - dispositivo.module.ts

```
dispositivo.module.ts x
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 {}
21
```

### 3.2) loO - dispositivo.page.ts

```
dispositivo.page.ts | dispositivo.module.ts
2 import { ActivatedRoute } from '@angular/router';
3 import { Medicion } from '../model/Medicion';
4 import { MedicionService } from '../services/medicion.service';
5 import * as moment from 'moment';
6
7 @Component({
8   selector: 'app-dispositivo',
9   templateUrl: './dispositivo.page.html',
10  styleUrls: ['./dispositivo.page.scss'],
11 })
12 export class DispositivoPage implements OnInit {
13
14   public medicion: Medicion;
15   public idDispositivo: string;
16   constructor(private router: ActivatedRoute, private mServ: MedicionService) { }
17
18   ngOnInit() {
19
20     this.idDispositivo = this.router.snapshot.paramMap.get('id');
21     this.mServ.getMedicionByIdDispositivo(this.idDispositivo).then((med) => {
22       this.medicion = med;
23     });
24
25     this.mServ.getMedicionesByIdDispositivo(this.idDispositivo).then((med) => {
26       console.log(med)
27     });
28
29     //6
30     // opción 1- utilizar libreria Momentis , haciendo npm install --save moment y luego el import * as moment from 'moment';
31     // donde lo necesitamos.
32     // let a : Medicion= new Medicion(99,moment().format("YYYY-MM-DD hh:mm:ss"),99,1);
33
34     //opción 2, utilizar el objeto Date y hacer el formato necesario a mano.
35     let current_datetime = new Date()
36     let formatted_date = current_datetime.getFullYear() + "-" + (current_datetime.getMonth() + 1) + "-" +
37       current_datetime.getDate() + " " + current_datetime.getHours() + ":" + current_datetime.getMinutes() + ":" + current_datetime.getSeconds()
38     let a : Medicion= new Medicion(99,formatted_date,99,1);
39
40     this.mServ.agregarMedicion(a).then((med) => {
41       console.log(med)
42     });
43
44 }
```

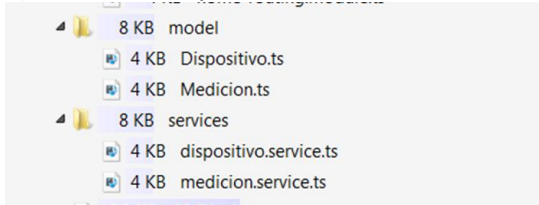
### 3.3) loO - dispositivo.page.html

```
C:\MioT2020\Materias\DesarrolloAppMultiplat\Practica\dam-master\ionicPromesas\src\app\dispositivo\dispositivo.page.html - Notepad++ [Adm
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
dispositivo.page.html | dispositivo.page.ts | dispositivo.module.ts
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 - {{this.idDispositivo}}</ion-title>
9   </ion-toolbar>
10 </ion-header>
11
12 <ion-content>
13   <div *ngIf="this.medicion != null">Medida del sensor: {{this.medicion.valor}}</div>
14 </ion-content>
```

### 3.4) loO - dispositivo-routing.module.ts

```
dispositivo-routing.module.ts | dispositivo.page.html | dispositivo.page.ts | dispositivo.modu
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 { }
```

#### 4) IoO Modelos y Servicios:



##### 4.1) IoO /model/Dispositivo.ts

```
Dispositivo.ts
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 }
```

##### 4.2) IoO /service/dispositivo.service.ts

```
dispositivo.service.ts
1 import { Injectable } from '@angular/core';
2 import { Dispositivo } from '../model/Dispositivo';
3 import { HttpClient } from '@angular/common/http';
4
5
6 @Injectable({
7   providedIn: 'root'
8 })
9 export class DispositivoService {
10   urlApi="http://localhost:3000";
11   // listado:Array<Dispositivo> = new Array<Dispositivo>();
12
13   constructor(private _http: HttpClient ) {
14   }
15
16
17   getListadoDispositivos():Promise<Dispositivo[]>{
18     return this._http.get(this.urlApi+ "/api/dispositivo/").toPromise().then((listado:Dispositivo[])=>{
19       return listado;
20     });
21   }
22
23   getDispositivo(id):Promise<Dispositivo>{
24     return this._http.get(this.urlApi+ "/api/dispositivo/"+id).toPromise().then((dispositivo:Dispositivo)=>{
25       return dispositivo;
26     });
27   };
28
29 }
```

## 4.2) IoO /model/Medicion.ts

```
Medicion.ts | dispositivo.service.ts | Dispositivo.ts
1 export class Medicion{
2     private _medicionId: number;
3     private _fecha: Date;
4     private _valor: number;
5     private _dispositivoId: number;
6
7     constructor(medicion,fecha,valor,dispositivoId){
8         this._medicionId=medicion;
9         this.fecha=fecha;
10        this._valor=valor;
11        this._dispositivoId=dispositivoId;
12    }
13
14    public get medicionId(): number {
15        return this._medicionId;
16    }
17    public set medicionId(value: number) {
18        this._medicionId = value;
19    }
20
21    public get fecha(): Date {
22        return this._fecha;
23    }
24    public set fecha(value: Date) {
25        this._fecha = value;
26    }
27
28    public get valor(): number {
29        return this._valor;
30    }
31    public set valor(value: number) {
32        this._valor = value;
33    }
34
35    public get dispositivoId(): number {
36        return this._dispositivoId;
37    }
38    public set dispositivoId(value: number) {
39        this._dispositivoId = value;
40    }
41 }
```

## 4.4) IoO /service/medicion.service.ts

```
medicion.service.ts | Medicion.ts | dispositivo.service.ts | Dispositivo.ts
1 import { Injectable } from '@angular/core';
2 import { HttpClient } from '@angular/common/http';
3 import { Medicion } from '../model/medicion'
4
5 @Injectable({
6     providedIn: 'root'
7 })
8 export class MedicionService {
9     urlApi="http://localhost:3000";
10
11     constructor(private _http: HttpClient ) {
12     }
13
14     getMedicionByIdDispositivo(id):Promise<Medicion>{
15         return this._http.get(this.urlApi+"/api/medicion/"+id).toPromise().then((med:Medicion)=>{
16             return med;
17         });
18     };
19
20     getMedicionesByIdDispositivo(id):Promise<Medicion[]>{
21         return this._http.get(this.urlApi+"/api/medicion/"+id+"/todas").toPromise().then((mediciones:Medicion[])=>{
22             return mediciones;
23         });
24     };
25
26     agregarMedicion(medicion:Medicion){
27         return this._http.post(this.urlApi+"/api/medicion/agregar",{fecha:medicion.fecha,valor:medicion.valor,dispositivoId:medicion.dispositivoId}).toPromise().then((result)=>{
28             return result;
29         });
30     }
31 }
```