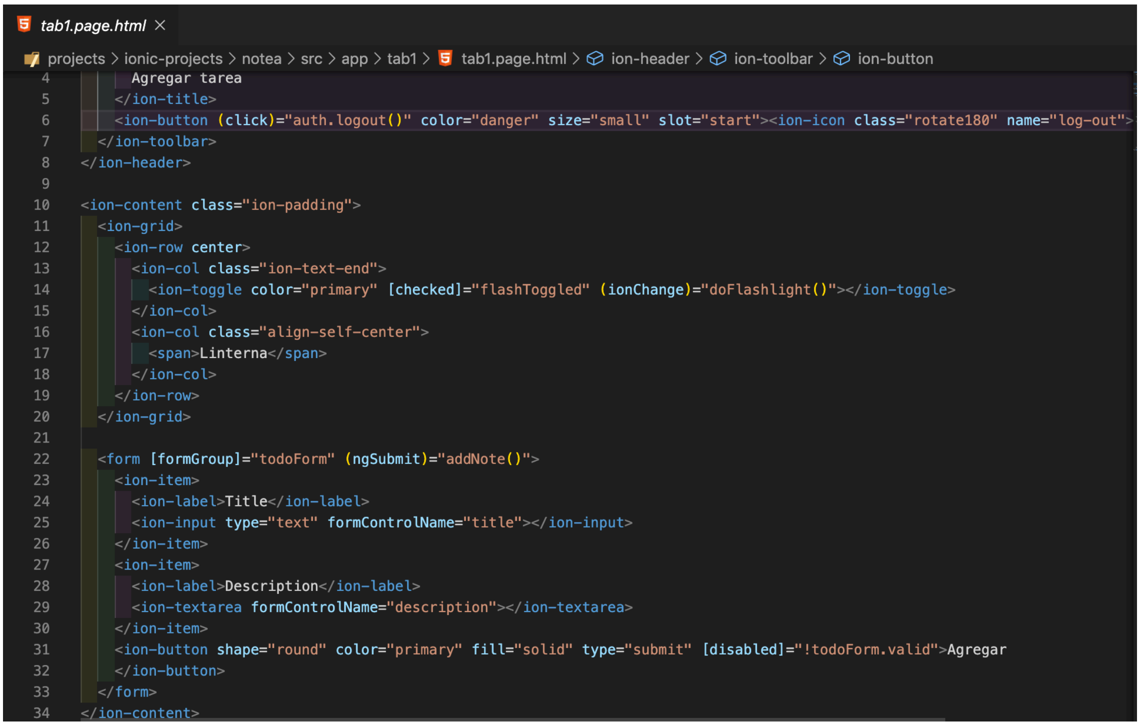
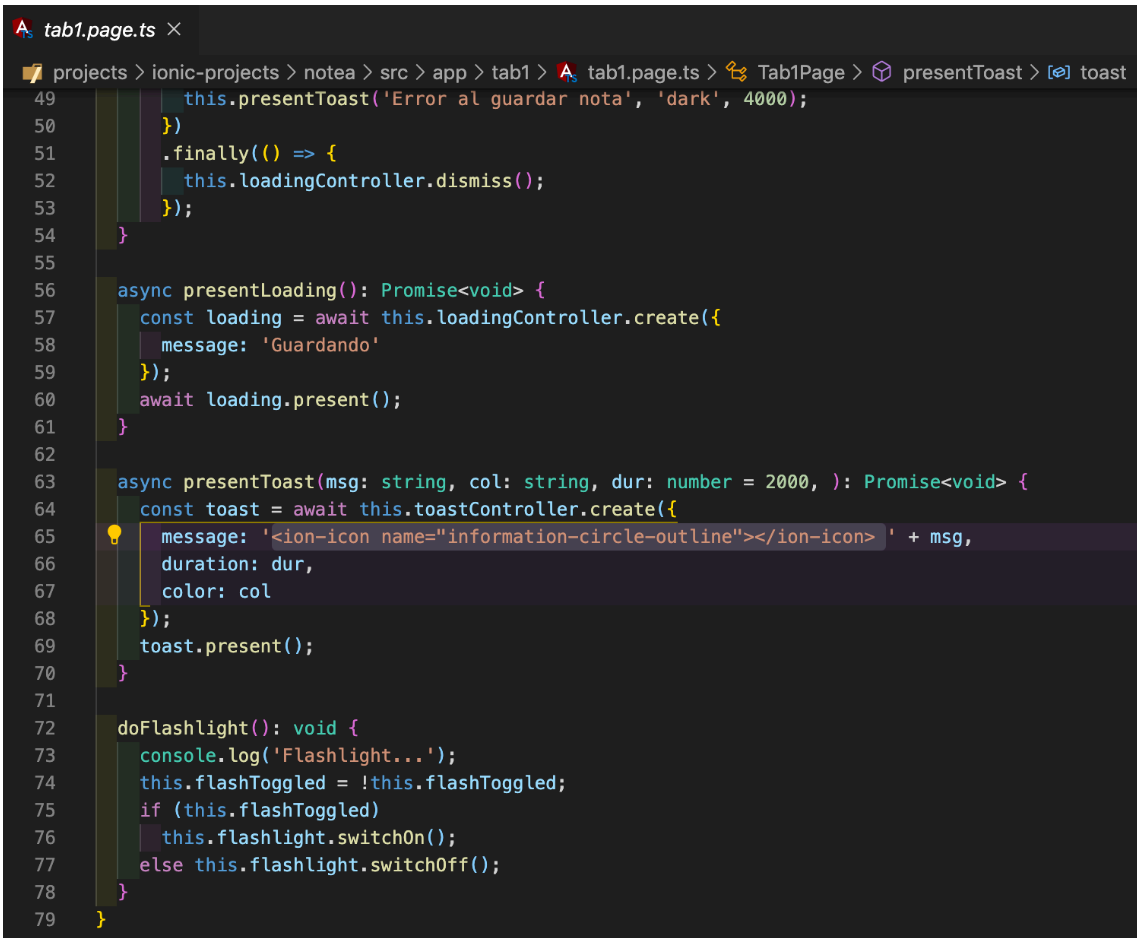
Proceso de desarrollo

App Ionic Notea

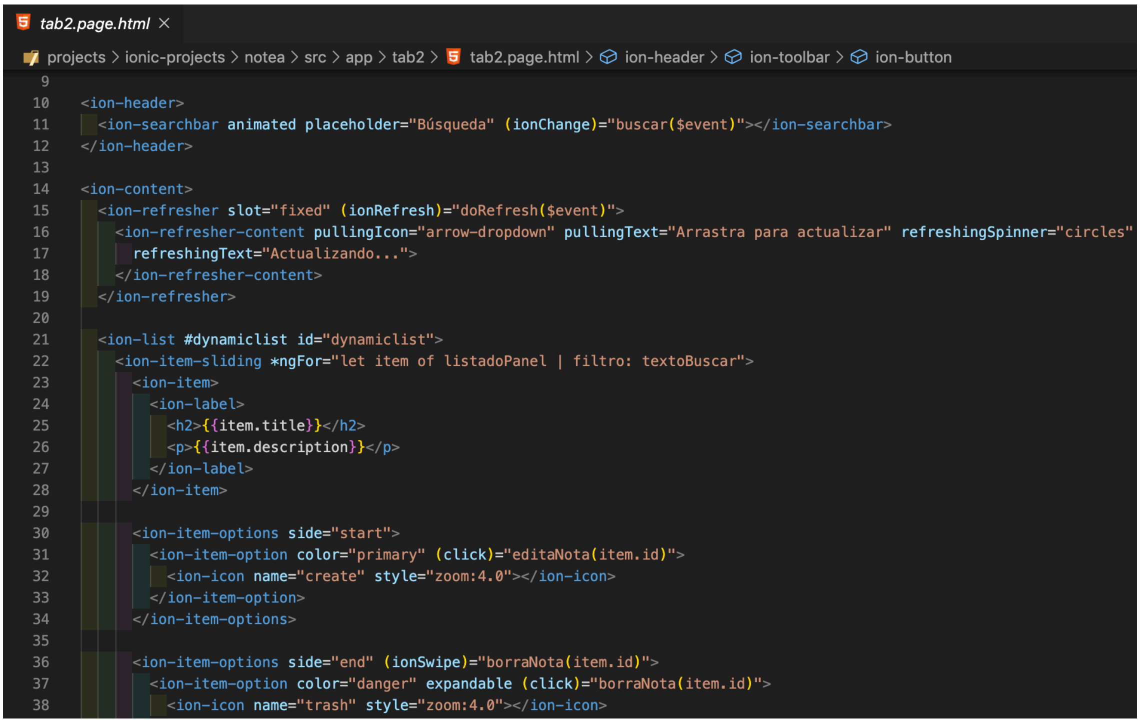

tab1.page.html ×
projects > ionic-projects > notea > src > app > tab1 > tab1.page.html > ion-header > ion-toolbar > ion-button

```
      Agregar tarea
    </ion-title>
    <ion-button (click)="auth.logout()" color="danger" size="small" slot="start"><ion-icon class="rotate180" name="log-out">
  </ion-toolbar>
</ion-header>

<ion-content class="ion-padding">
  <ion-grid>
    <ion-row center>
      <ion-col class="ion-text-end">
        <ion-toggle color="primary" [checked]="flashToggled" (ionChange)="doFlashlight()"></ion-toggle>
      </ion-col>
      <ion-col class="align-self-center">
        <span>Linterna</span>
      </ion-col>
    </ion-row>
  </ion-grid>

  <form [formGroup]="todoForm" (ngSubmit)="addNote()">
    <ion-item>
      <ion-label>Title</ion-label>
      <ion-input type="text" formControlName="title"></ion-input>
    </ion-item>
    <ion-item>
      <ion-label>Description</ion-label>
      <ion-textarea formControlName="description"></ion-textarea>
    </ion-item>
    <ion-button shape="round" color="primary" fill="solid" type="submit" [disabled]="!todoForm.valid">Agregar
    </ion-button>
  </form>
</ion-content>
```

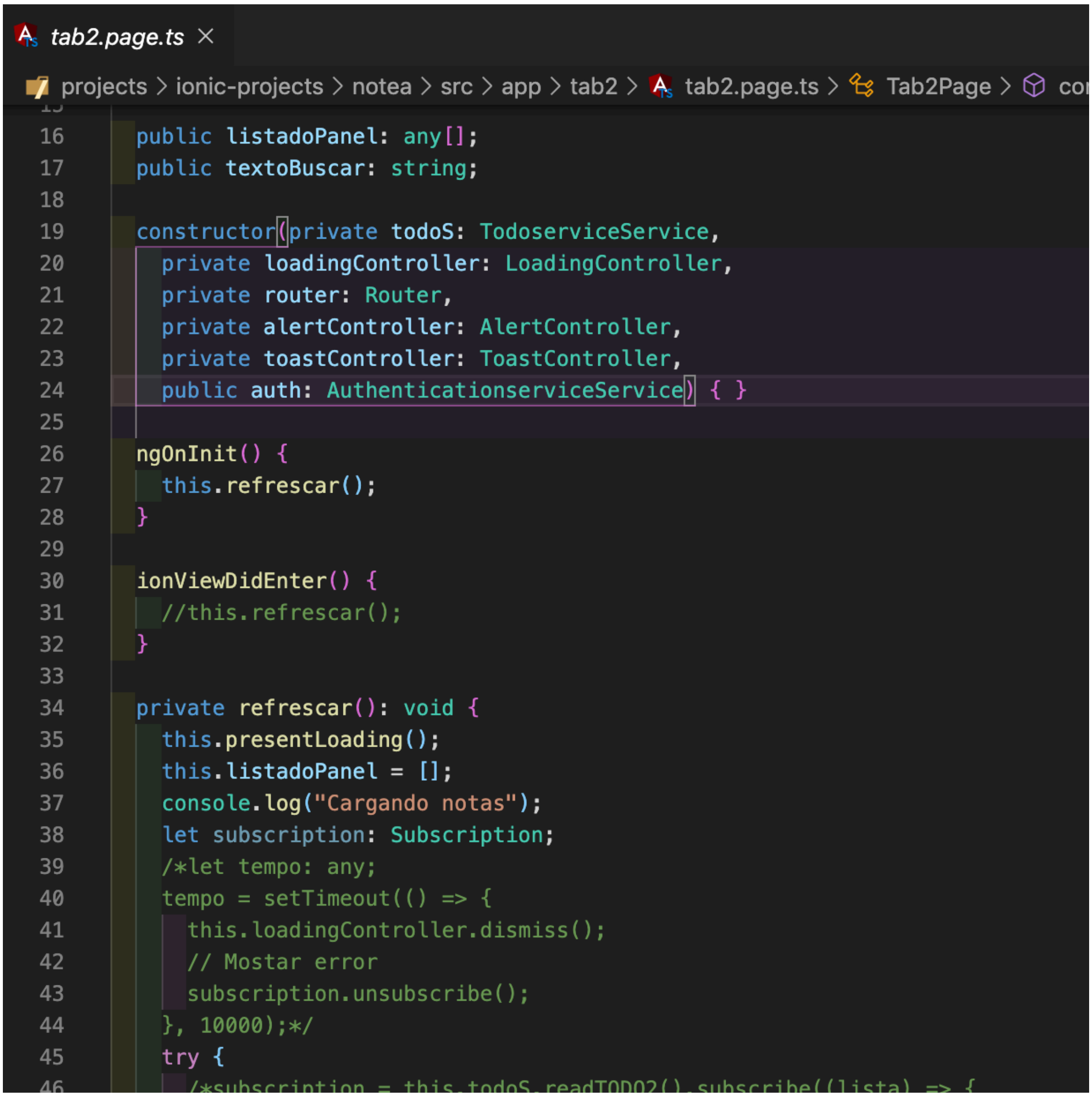
Jesús David Chicano Galindo


tab1.page.ts
projects > ionic-projects > notea > src > app > tab1 > tab1.page.ts > Tab1Page > presentToast > toast

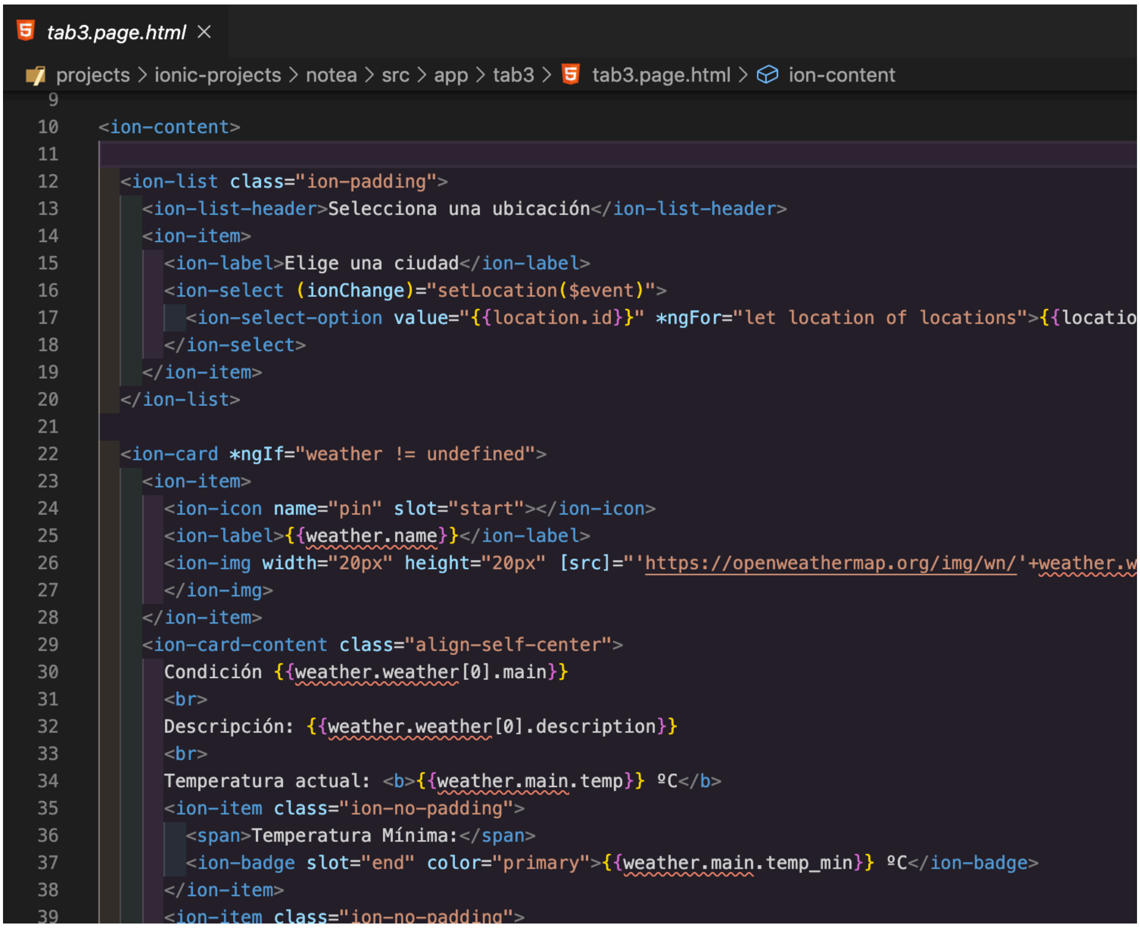
```typescript
        this.presentToast('Error al guardar nota', 'dark', 4000);
      })
      .finally(() => {
        this.loadingController.dismiss();
      });
    }

  async presentLoading(): Promise<void> {
    const loading = await this.loadingController.create({
      message: 'Guardando'
    });
    await loading.present();
  }

  async presentToast(msg: string, col: string, dur: number = 2000, ): Promise<void> {
    const toast = await this.toastController.create({
      message: '<ion-icon name="information-circle-outline"></ion-icon> ' + msg,
      duration: dur,
      color: col
    });
    toast.present();
  }

  doFlashlight(): void {
    console.log('Flashlight...');
    this.flashToggled = !this.flashToggled;
    if (this.flashToggled)
      this.flashlight.switchOn();
    else this.flashlight.switchOff();
  }
}
```


tab2.page.html
projects > ionic-projects > notea > src > app > tab2 > tab2.page.html > ion-header > ion-toolbar > ion-button

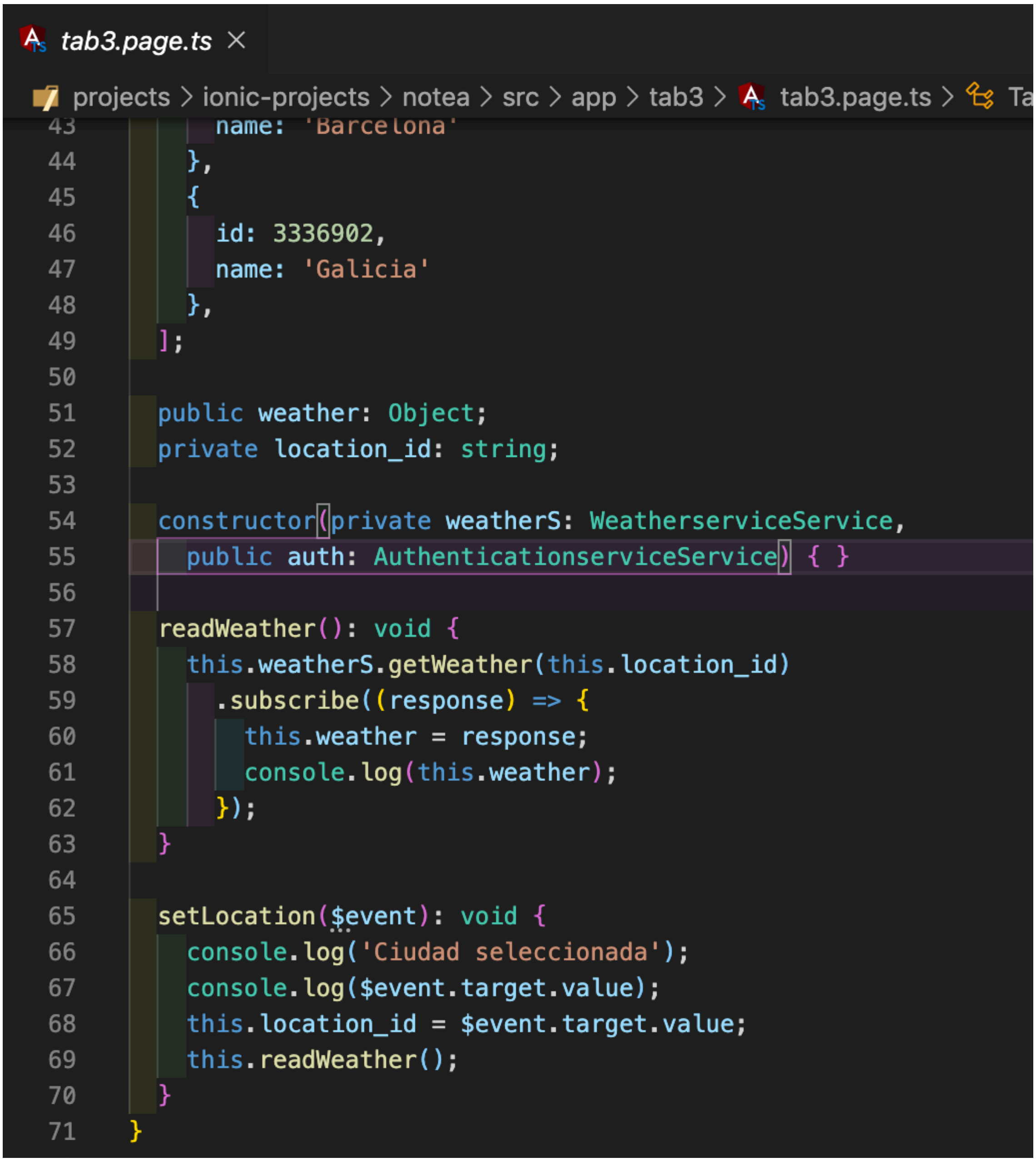
```html

<ion-header>
  <ion-searchbar animated placeholder="Búsqueda" (ionChange)="buscar($event)"></ion-searchbar>
</ion-header>

<ion-content>
  <ion-refresher slot="fixed" (ionRefresh)="doRefresh($event)">
    <ion-refresher-content pullingIcon="arrow-dropdown" pullingText="Arrastra para actualizar" refreshingSpinner="circles"
      refreshingText="Actualizando...">
    </ion-refresher-content>
  </ion-refresher>

  <ion-list #dynamiclist id="dynamiclist">
    <ion-item-sliding *ngFor="let item of listadoPanel | filtro: textoBuscar">
      <ion-item>
        <ion-label>
          <h2>{{item.title}}</h2>
          <p>{{item.description}}</p>
        </ion-label>
      </ion-item>

      <ion-item-options side="start">
        <ion-item-option color="primary" (click)="editaNota(item.id)">
          <ion-icon name="create" style="zoom:4.0"></ion-icon>
        </ion-item-option>
      </ion-item-options>

      <ion-item-options side="end" (ionSwipe)="borraNota(item.id)">
        <ion-item-option color="danger" expandable (click)="borraNota(item.id)">
          <ion-icon name="trash" style="zoom:4.0"></ion-icon>
```


tab2.page.ts
projects > ionic-projects > notea > src > app > tab2 > tab2.page.ts > Tab2Page > co

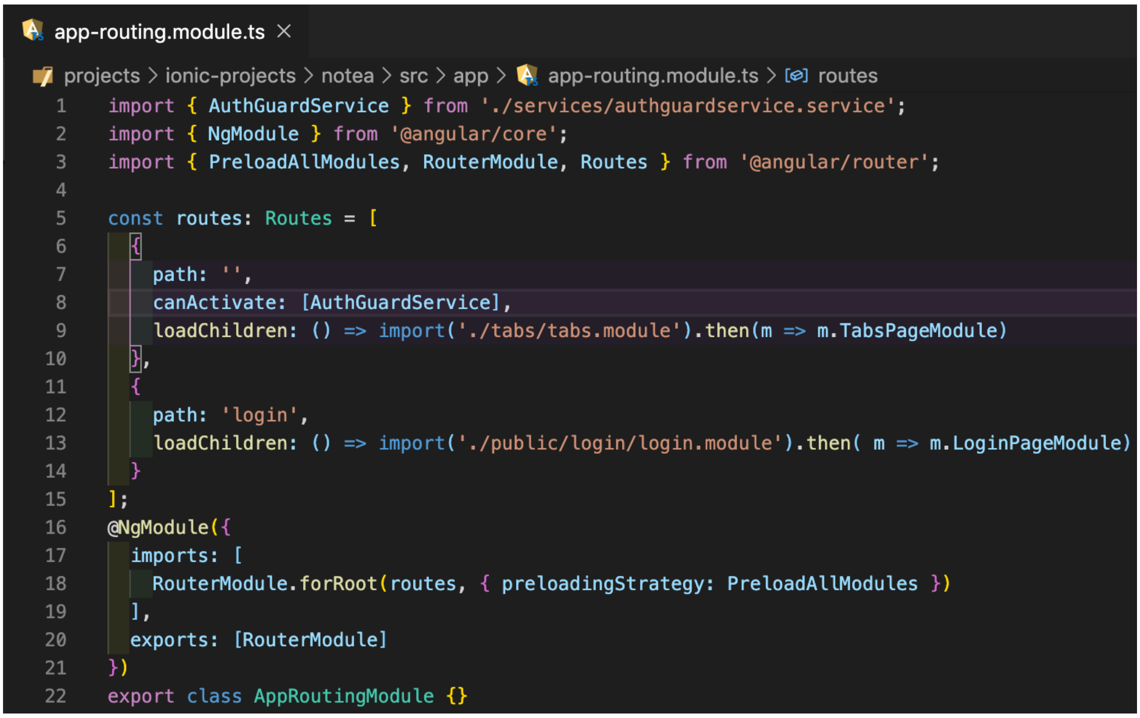
```
  public listadoPanel: any[];
  public textoBuscar: string;

  constructor(private todoS: TodoserviceService,
    private loadingController: LoadingController,
    private router: Router,
    private alertController: AlertController,
    private toastController: ToastController,
    public auth: AuthenticationserviceService) { }

  ngOnInit() {
    this.refrescar();
  }

  ionViewDidEnter() {
    //this.refrescar();
  }

  private refrescar(): void {
    this.presentLoading();
    this.listadoPanel = [];
    console.log("Cargando notas");
    let subscription: Subscription;
    /*let tempo: any;
    tempo = setTimeout(() => {
      this.loadingController.dismiss();
      // Mostar error
      subscription.unsubscribe();
    }, 10000);*/
    try {
      /*subscription = this.todoS.readTODO2().subscribe((lista) => {
```


tab3.page.html ×
projects > ionic-projects > notea > src > app > tab3 > tab3.page.html > ion-content

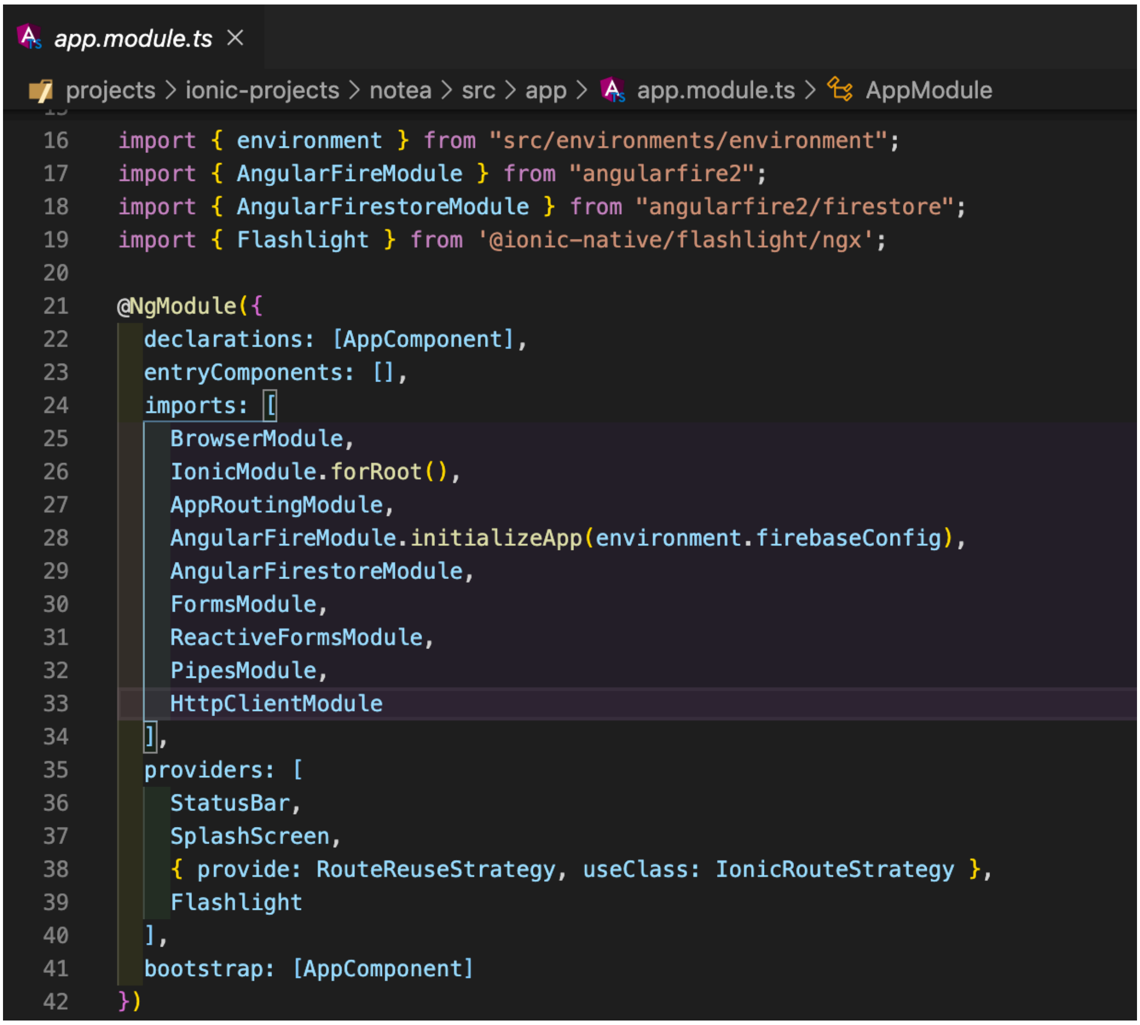
```html

<ion-content>

  <ion-list class="ion-padding">
    <ion-list-header>Selecciona una ubicación</ion-list-header>
    <ion-item>
      <ion-label>Elige una ciudad</ion-label>
      <ion-select (ionChange)="setLocation($event)">
        <ion-select-option value="{{location.id}}" *ngFor="let location of locations">{{locatio
      </ion-select>
    </ion-item>
  </ion-list>

  <ion-card *ngIf="weather != undefined">
    <ion-item>
      <ion-icon name="pin" slot="start"></ion-icon>
      <ion-label>{{weather.name}}</ion-label>
      <ion-img width="20px" height="20px" [src]="'https://openweathermap.org/img/wn/'+weather.w
      </ion-img>
    </ion-item>
    <ion-card-content class="align-self-center">
      Condición {{weather.weather[0].main}}
      <br>
      Descripción: {{weather.weather[0].description}}
      <br>
      Temperatura actual: <b>{{weather.main.temp}} ºC</b>
      <ion-item class="ion-no-padding">
        <span>Temperatura Mínima:</span>
        <ion-badge slot="end" color="primary">{{weather.main.temp_min}} ºC</ion-badge>
      </ion-item>
      <ion-item class="ion-no-padding">
```


tab3.page.ts
projects > ionic-projects > notea > src > app > tab3 > tab3.page.ts > Ta

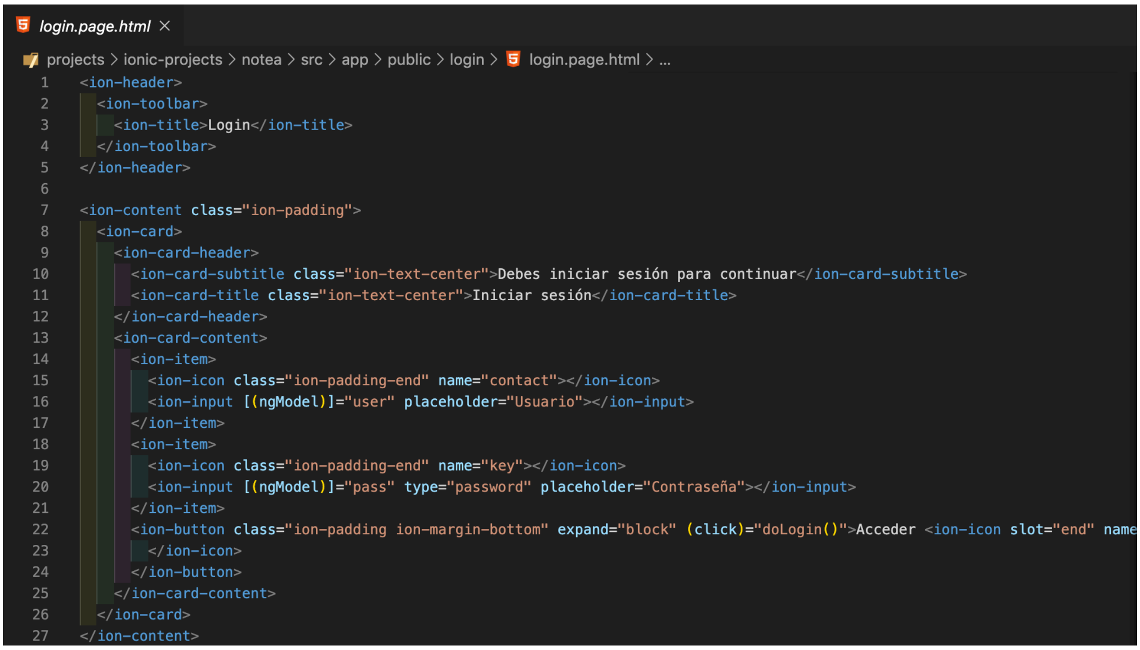
```
      name: 'Barcelona'
    },
    {
      id: 3336902,
      name: 'Galicia'
    },
  ];

  public weather: Object;
  private location_id: string;

  constructor(private weatherS: WeatherserviceService,
    public auth: AuthenticationserviceService) { }

  readWeather(): void {
    this.weatherS.getWeather(this.location_id)
      .subscribe((response) => {
        this.weather = response;
        console.log(this.weather);
      });
  }

  setLocation($event): void {
    console.log('Ciudad seleccionada');
    console.log($event.target.value);
    this.location_id = $event.target.value;
    this.readWeather();
  }
}
```


app-routing.module.ts
projects > ionic-projects > notea > src > app > app-routing.module.ts > routes

```typescript
import { AuthGuardService } from './services/authguardservice.service';
import { NgModule } from '@angular/core';
import { PreloadAllModules, RouterModule, Routes } from '@angular/router';

const routes: Routes = [
  {
    path: '',
    canActivate: [AuthGuardService],
    loadChildren: () => import('./tabs/tabs.module').then(m => m.TabsPageModule)
  },
  {
    path: 'login',
    loadChildren: () => import('./public/login/login.module').then( m => m.LoginPageModule)
  }
];
@NgModule({
  imports: [
    RouterModule.forRoot(routes, { preloadingStrategy: PreloadAllModules })
  ],
  exports: [RouterModule]
})
export class AppRoutingModule {}
```

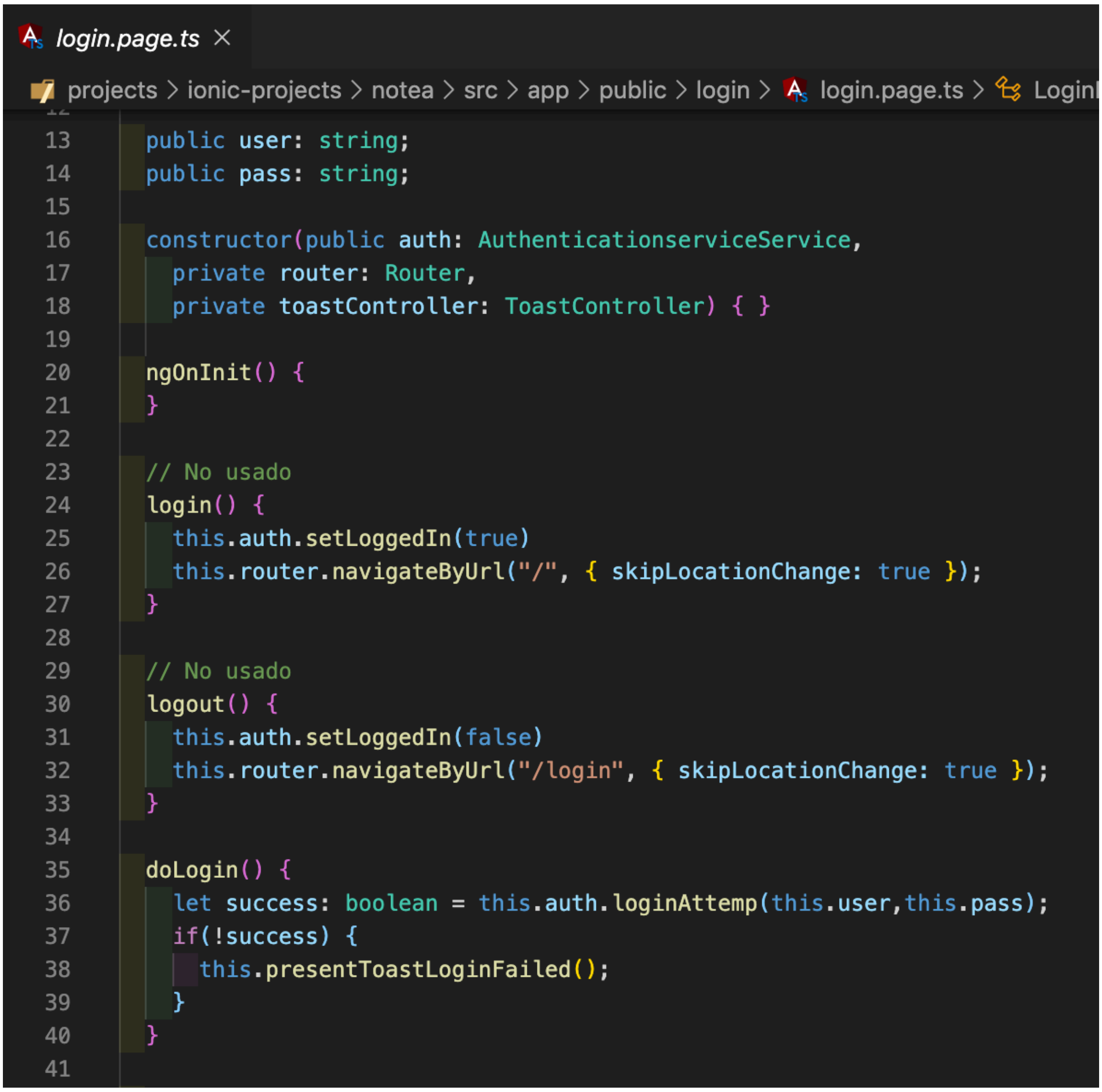

app.module.ts
projects > ionic-projects > notea > src > app > app.module.ts > AppModule

```typescript
import { environment } from "src/environments/environment";
import { AngularFireModule } from "angularfire2";
import { AngularFirestoreModule } from "angularfire2/firestore";
import { Flashlight } from '@ionic-native/flashlight/ngx';

@NgModule({
  declarations: [AppComponent],
  entryComponents: [],
  imports: [
    BrowserModule,
    IonicModule.forRoot(),
    AppRoutingModule,
    AngularFireModule.initializeApp(environment.firebaseConfig),
    AngularFirestoreModule,
    FormsModule,
    ReactiveFormsModule,
    PipesModule,
    HttpClientModule
  ],
  providers: [
    StatusBar,
    SplashScreen,
    { provide: RouteReuseStrategy, useClass: IonicRouteStrategy },
    Flashlight
  ],
  bootstrap: [AppComponent]
})
```



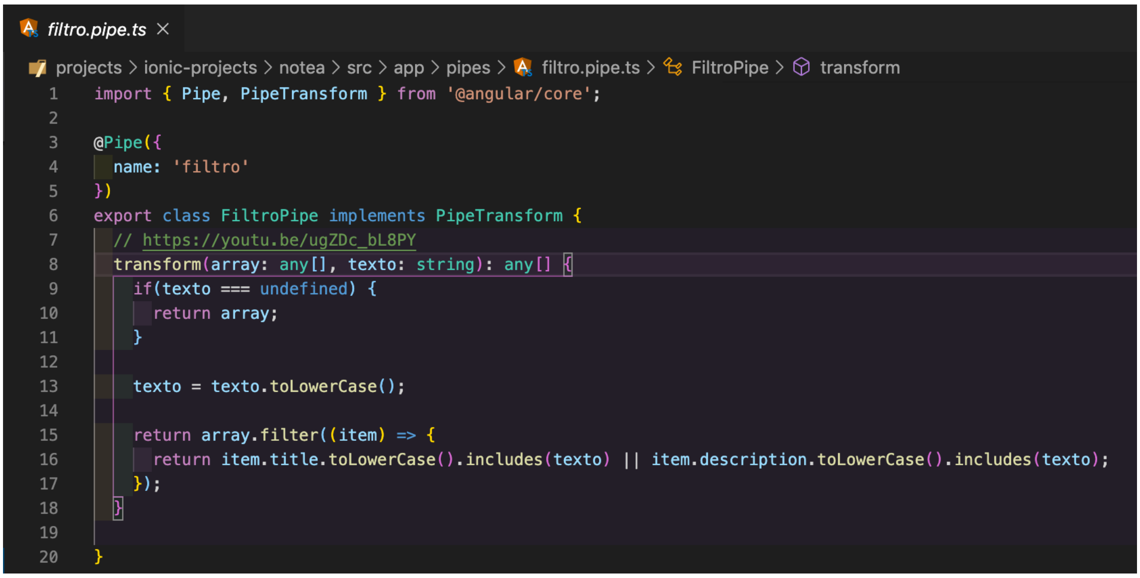
login.page.html
projects > ionic-projects > notea > src > app > public > login > login.page.html > ...

```html
<ion-header>
  <ion-toolbar>
    <ion-title>Login</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content class="ion-padding">
  <ion-card>
    <ion-card-header>
      <ion-card-subtitle class="ion-text-center">Debes iniciar sesión para continuar</ion-card-subtitle>
      <ion-card-title class="ion-text-center">Iniciar sesión</ion-card-title>
    </ion-card-header>
    <ion-card-content>
      <ion-item>
        <ion-icon class="ion-padding-end" name="contact"></ion-icon>
        <ion-input [(ngModel)]="user" placeholder="Usuario"></ion-input>
      </ion-item>
      <ion-item>
        <ion-icon class="ion-padding-end" name="key"></ion-icon>
        <ion-input [(ngModel)]="pass" type="password" placeholder="Contraseña"></ion-input>
      </ion-item>
      <ion-button class="ion-padding ion-margin-bottom" expand="block" (click)="doLogin()">Acceder <ion-icon slot="end" name
        </ion-icon>
      </ion-button>
    </ion-card-content>
  </ion-card>
</ion-content>
```

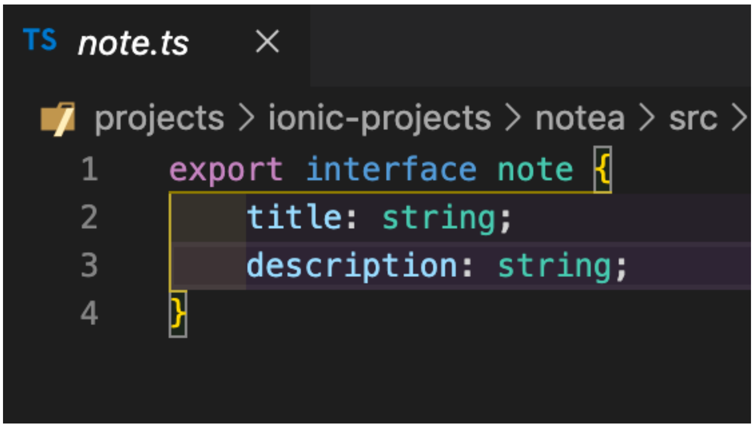


login.page.ts
projects > ionic-projects > notea > src > app > public > login > login.page.ts > Login

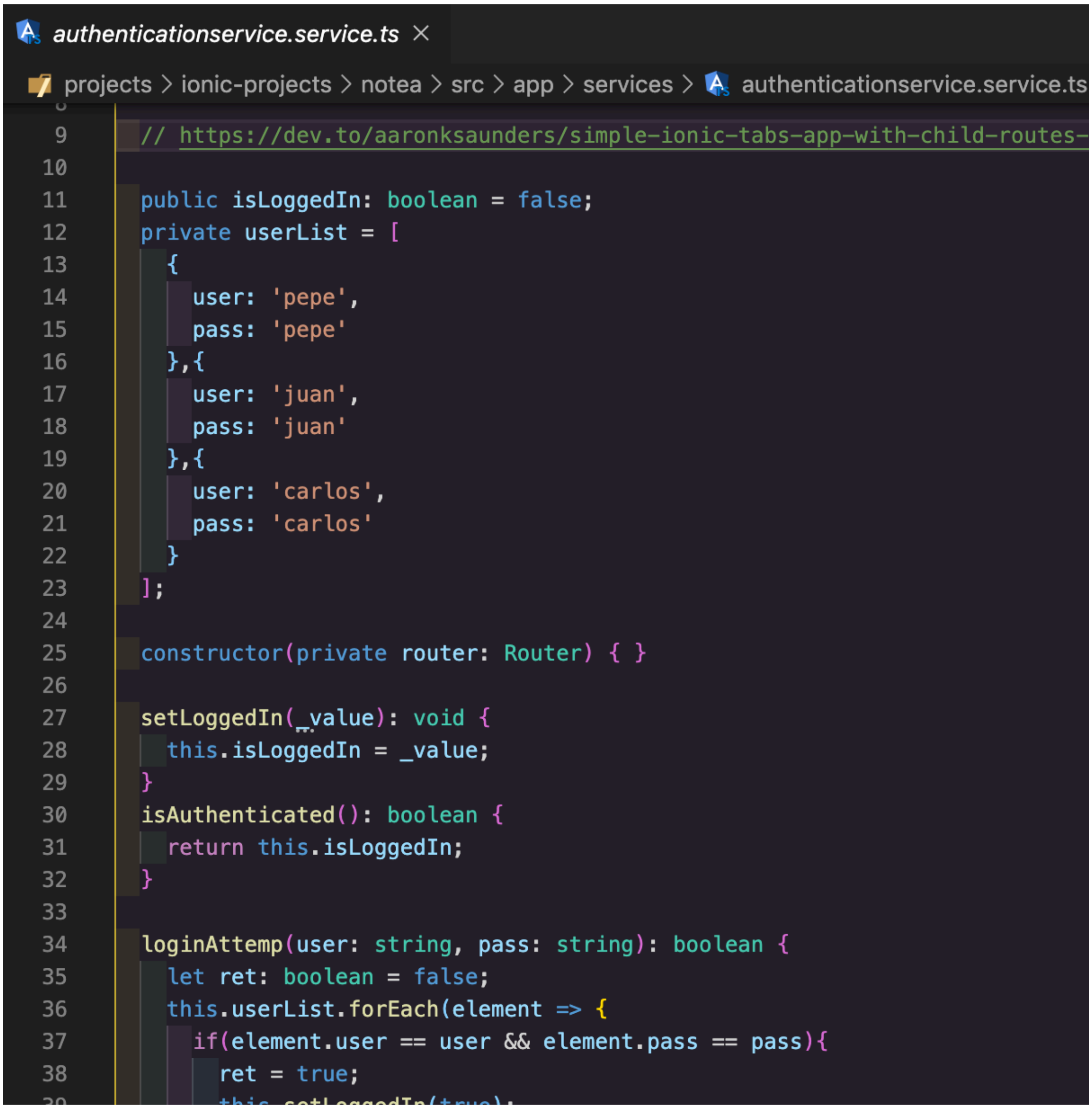
```typescript
public user: string;
public pass: string;

constructor(public auth: AuthenticationserviceService,
  private router: Router,
  private toastController: ToastController) { }

ngOnInit() {
}

// No usado
login() {
  this.auth.setLoggedIn(true)
  this.router.navigateByUrl("/", { skipLocationChange: true });
}

// No usado
logout() {
  this.auth.setLoggedIn(false)
  this.router.navigateByUrl("/login", { skipLocationChange: true });
}

doLogin() {
  let success: boolean = this.auth.loginAttemp(this.user,this.pass);
  if(!success) {
    this.presentToastLoginFailed();
  }
}

```


filtro.pipe.ts
projects > ionic-projects > notea > src > app > pipes > filtro.pipe.ts > FiltroPipe > transform

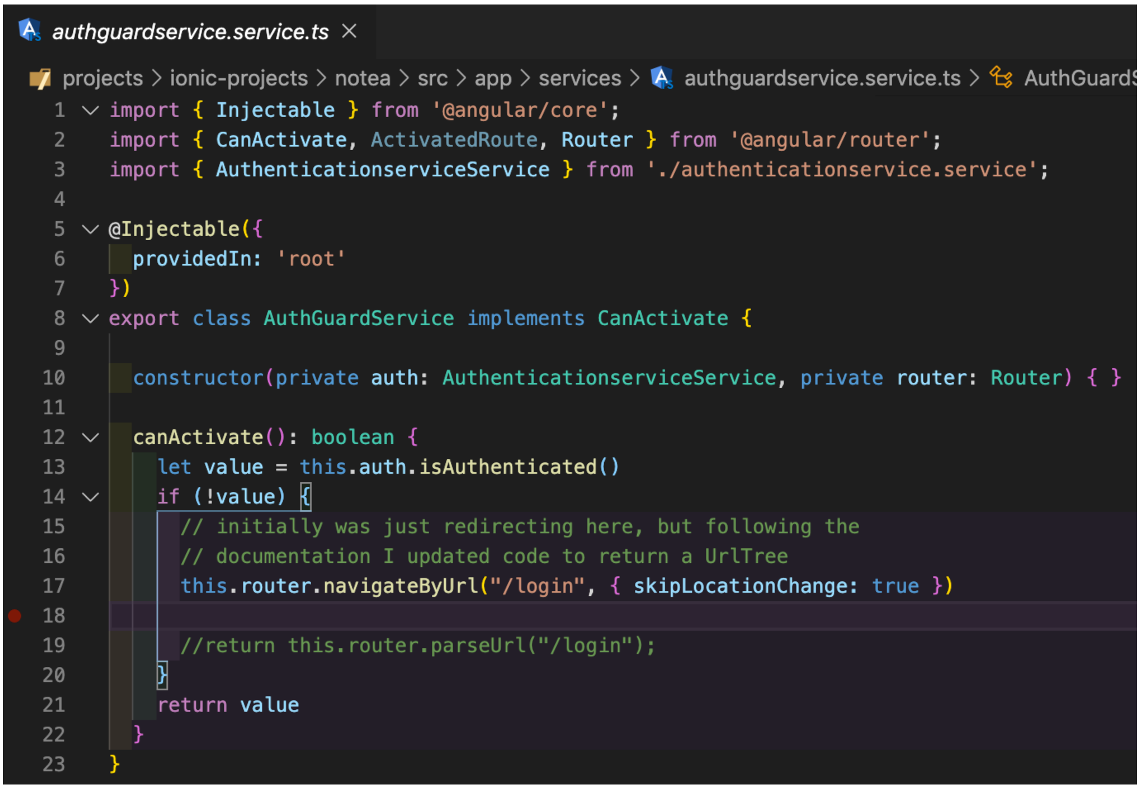
```
import { Pipe, PipeTransform } from '@angular/core';

@Pipe({
  name: 'filtro'
})
export class FiltroPipe implements PipeTransform {
  // https://youtu.be/ugZDc_bL8PY
  transform(array: any[], texto: string): any[] {
    if(texto === undefined) {
      return array;
    }

    texto = texto.toLowerCase();

    return array.filter((item) => {
      return item.title.toLowerCase().includes(texto) || item.description.toLowerCase().includes(texto);
    });
  }

}
```



```
export interface note {
    title: string;
    description: string;
}
```

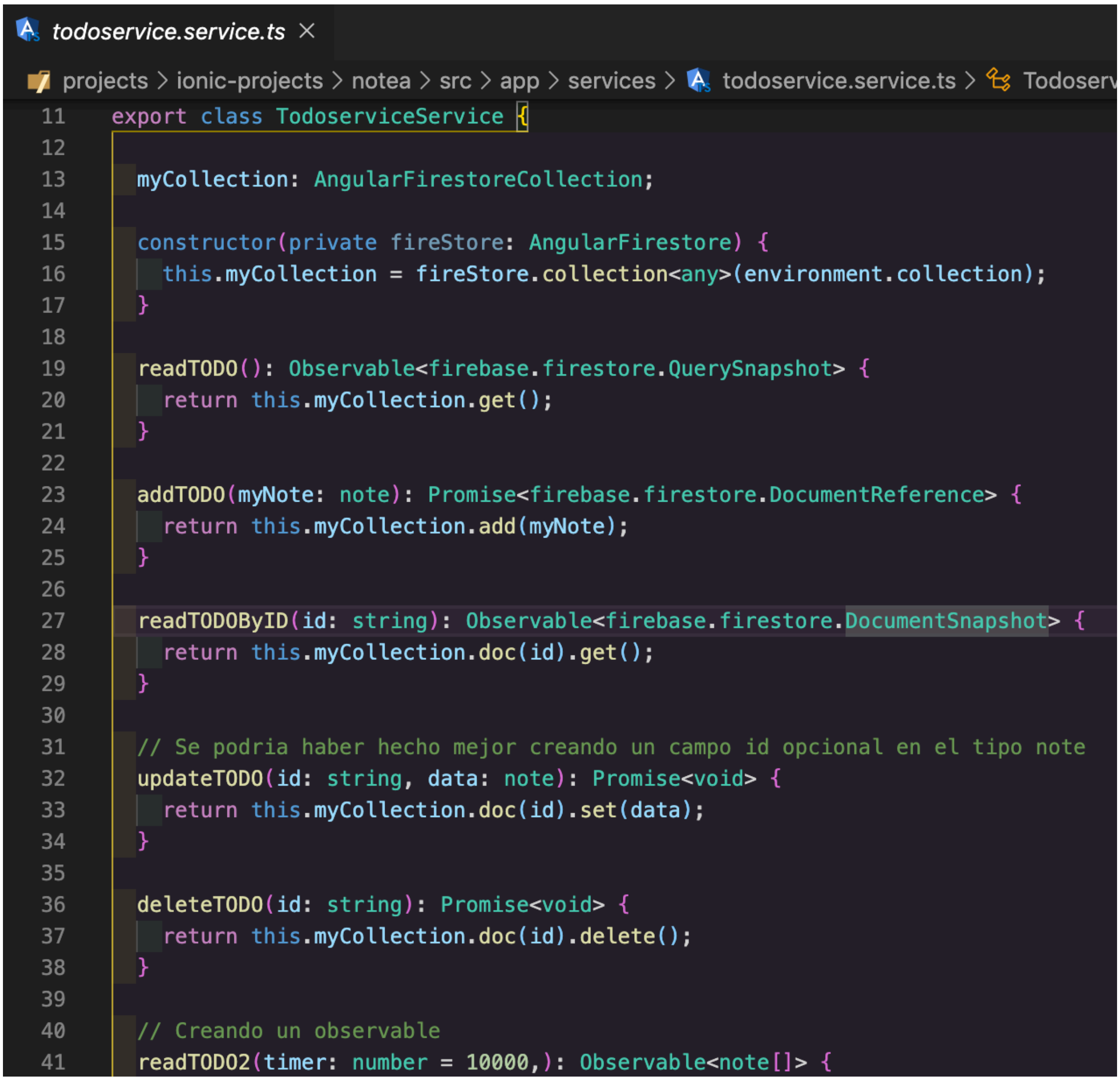

authenticationservice.service.ts
projects > ionic-projects > notea > src > app > services > authenticationservice.service.ts

```
// https://dev.to/aaronksaunders/simple-ionic-tabs-app-with-child-routes-

public isLoggedIn: boolean = false;
private userList = [
  {
    user: 'pepe',
    pass: 'pepe'
  },{
    user: 'juan',
    pass: 'juan'
  },{
    user: 'carlos',
    pass: 'carlos'
  }
];

constructor(private router: Router) { }

setLoggedIn(_value): void {
  this.isLoggedIn = _value;
}
isAuthenticated(): boolean {
  return this.isLoggedIn;
}

loginAttemp(user: string, pass: string): boolean {
  let ret: boolean = false;
  this.userList.forEach(element => {
    if(element.user == user && element.pass == pass){
      ret = true;
      this.setLoggedIn(true);
```

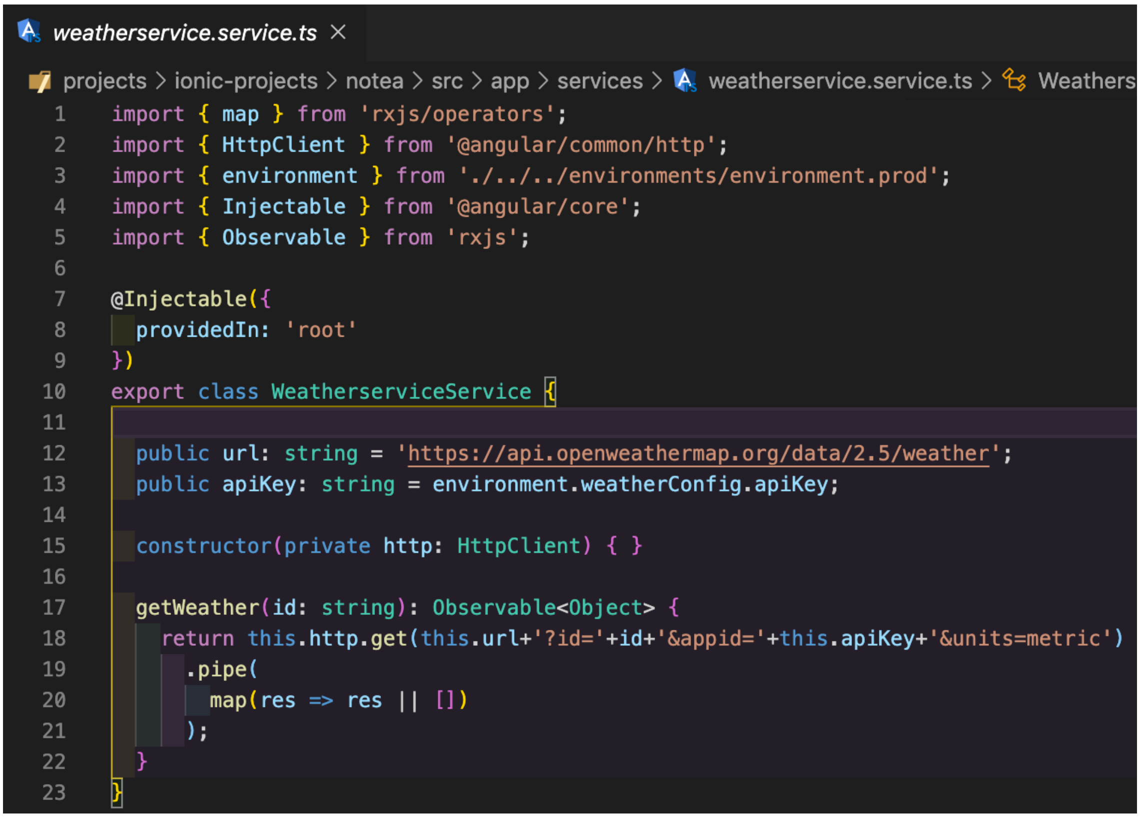


authguardservice.service.ts
projects > ionic-projects > notea > src > app > services > authguardservice.service.ts > AuthGuardS

```typescript
import { Injectable } from '@angular/core';
import { CanActivate, ActivatedRoute, Router } from '@angular/router';
import { AuthenticationserviceService } from './authenticationservice.service';

@Injectable({
  providedIn: 'root'
})
export class AuthGuardService implements CanActivate {

  constructor(private auth: AuthenticationserviceService, private router: Router) { }

  canActivate(): boolean {
    let value = this.auth.isAuthenticated()
    if (!value) {
      // initially was just redirecting here, but following the
      // documentation I updated code to return a UrlTree
      this.router.navigateByUrl("/login", { skipLocationChange: true })

      //return this.router.parseUrl("/login");
    }
    return value
  }
}
```


todoservice.service.ts
projects > ionic-projects > notea > src > app > services > todoservice.service.ts > Todoserv

```
export class TodoserviceService {

  myCollection: AngularFirestoreCollection;

  constructor(private fireStore: AngularFirestore) {
    this.myCollection = fireStore.collection<any>(environment.collection);
  }

  readTODO(): Observable<firebase.firestore.QuerySnapshot> {
    return this.myCollection.get();
  }

  addTODO(myNote: note): Promise<firebase.firestore.DocumentReference> {
    return this.myCollection.add(myNote);
  }

  readTODOByID(id: string): Observable<firebase.firestore.DocumentSnapshot> {
    return this.myCollection.doc(id).get();
  }

  // Se podria haber hecho mejor creando un campo id opcional en el tipo note
  updateTODO(id: string, data: note): Promise<void> {
    return this.myCollection.doc(id).set(data);
  }

  deleteTODO(id: string): Promise<void> {
    return this.myCollection.doc(id).delete();
  }

  // Creando un observable
  readTODO2(timer: number = 10000,): Observable<note[]> {
```


weatherservice.service.ts
projects > ionic-projects > notea > src > app > services > weatherservice.service.ts > Weathers

```typescript
import { map } from 'rxjs/operators';
import { HttpClient } from '@angular/common/http';
import { environment } from './../../environments/environment.prod';
import { Injectable } from '@angular/core';
import { Observable } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class WeatherserviceService {

  public url: string = 'https://api.openweathermap.org/data/2.5/weather';
  public apiKey: string = environment.weatherConfig.apiKey;

  constructor(private http: HttpClient) { }

  getWeather(id: string): Observable<Object> {
    return this.http.get(this.url+'?id='+id+'&appid='+this.apiKey+'&units=metric')
      .pipe(
        map(res => res || [])
      );
  }
}
```