

Comunicación entre componentes

Tercera Parte

Ph. D. Lenin G. Lemus Zúñiga*

October 6, 2019

Contents

1	Objetivo	2
2	Prerequisitos	2
2.1	Creación y configuración de los componentes guardas e interfaces	2
3	Configuración del módulo Diary	2
3.1	app.module.ts	3
4	Configuración del componente Login	4
4.1	login.component.ts	4
4.2	login.component.html	5
4.3	login.component.scss	5
5	Servicio Auth	6
6	Guarda Auth	6
7	Componentes Header y Footer	7
7.1	Modificaciones al componente header	7
7.1.1	header.component.ts	7
7.1.2	header.component.html	8
7.1.3	header.component.scss	8
7.2	Modificaciones al componente footer	9
7.2.1	footer.component.ts	9
7.2.2	footer.component.html	9
7.2.3	footer.component.scss	9
8	Routing	10
8.1	app-routing.component.ts	10

*lenin.lemus@hotmail.com

1 Objetivo

Agregar la lógica de autenticación a nuestra aplicación en desarrollo.

2 Prerequisitos

Acción recomendada: Realizar las indicaciones del fichero con la practicaNG802.pdf y continuar con las indicaciones de esta práctica.

De forma alternativa, descargue de la página lemus.webs.upv.es. Descomprima el fichero en la carpeta c:

cursoNG8. Verifique que la aplicación está operativa, ejecutando los siguientes comandos:

```
1 c:\> cd c:\cursoNG8\practicaNG802
2 c:\cursoNG8\practicaNG802> npm i
3 c:\cursoNG8\practicaNG802> ng start -o
```

Listing 1: Aplicación operativa

En el navegador por defecto aparecerá la aplicación. Mediante el uso de un editor, se sugiere Visual Studio Code. Abra la carpeta c:

cursoNG8

practicaNG802 y proceda a realizar los cambios que se detallan en las siguientes secciones.

2.1 Creación y configuración de los componentes guardas e interfaces

Se procede a agregar un componente para que muestre un formulario para que los usuarios puedan autenticarse. Una interfaz que indica los campos que se requieren en el login. Y un servicio.

Tanto el servicio como la guarda ya fueron generados en la primera práctica

```
ng generate service services/auth
ng generate guard auth/auth
```

Para crear la interfaz ejecute el siguiente comando:

```
ng generate class auth/userCredentials
ng generate component login
```

Para crear el componente de login ejecute el siguiente comando:

```
ng generate component login
```

3 Configuración del módulo Diary

Modificar los siguientes ficheros:

- app.module.ts

- login.component.ts
- login.component.ts.html
- login.component.ts.scss
- auth.service.ts
- auth.guard
- app-routing.component.ts

3.1 app.module.ts

Verifique que están importados los módulos `ReactiveFormsModule` y `MaterialDesignModule`, en caso de que no lo estén agréguelos. Los módulos fueron agregados desde la primera práctica!

```

1 import { BrowserModule } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3 import { AppRoutingModule } from './app-routing.module';
4 import { AppComponent } from './app.component';
5 import { BrowserAnimationsModule } from '@angular/platform-
  browser/animations';
6 import { HeaderComponent } from './header/header.component';
7 import { FooterComponent } from './footer/footer.component';
8 import { NotFoundComponent } from './not-found/not-found.
  component';
9 import { PrincipalComponent } from './principal/principal.
  component';
10 import { AboutComponent } from './about/about.component';
11 import { ReactiveFormsModule } from
  '@angular/forms'; @*/ (*@import MaterialDesignModule from
  './material-design/material-design.module'; @*/ import
  LoginComponent from
  './login/login.component'; @NgModule (declarations:
  [AppComponent, HeaderComponent, FooterComponent, NotFoundComponent, PrincipalComponent,
  [BrowserModule, AppRoutingModule, BrowserAnimationsModule, (*@ReactiveFormsModule
12   MaterialDesignModule
13 ],
14 providers: [],
15 bootstrap: [AppComponent]
16 })
17 export class AppModule { }
```

4 Configuración del componente Login

4.1 login.component.ts

```
1 import { Component, OnInit } from '@angular/core';
2 import FormGroup, FormBuilder, Validators from
  '@angular/forms';
3 import AuthService from '../services/auth.service';
4
5 @Component({
6   selector: 'app-login',
7   templateUrl: './login.component.html',
8   styleUrls: ['./login.component.scss']
9 })
10 export class LoginComponent implements OnInit {
11
12   form: FormGroup; // 1
13   private formSubmitAttempt: boolean; // 2
14
15   constructor(
16     private fb: FormBuilder, // 3
17     private authService: AuthService // 4
18   ) {}
19
20   ngOnInit() {
21     this.form = this.fb.group({ // 5
22       userName: ['', Validators.required],
23       password: ['', Validators.required]
24     });
25   }
26
27   isFieldInvalid(field: string) { // {6}
28     return (
29       (!this.form.get(field).valid && this.form.get(field).
30         touched) ||
31       (this.form.get(field).untouched && this.
32         formSubmitAttempt)
33     );
34   }
35
36   onSubmit() {
37     if (this.form.valid) {
38       this.authService.login(this.form.value); // {7}
39     }
39     this.formSubmitAttempt = true; // {8}
```

```

39 }
40
41 }

```

4.2 login.component.html

```

1 <div class="signin-content">
2   <mat-card>
3     <mat-card-content>
4       <form [formGroup]="form" (ngSubmit)="onSubmit()">
5         <p>Please login to continue</p>
6         <mat-form-field class="full-width-input">
7           <input matInput placeholder="User"
8             formControlName="userName" required>
9             <mat-error *ngIf="isFieldInvalid('userName
10              ')">
11                Please write your user name
12              </mat-error>
13            </mat-form-field>
14            <mat-form-field class="full-width-input">
15              <input matInput type="password"
16                placeholder="Password" formControlName="password" required
17                >
18                <mat-error *ngIf="isFieldInvalid('userName
19                  ')">
20                  Please write your password
21                </mat-error>
22              </mat-form-field>
23              <button mat-raised-button color="primary">
24                Login</button>
25            </form>
26          </mat-card-content>
27        </mat-card>
28      </div>

```

4.3 login.component.scss

```

1 mat-card {
2   max-width: 400px;
3   margin: 2em auto;
4   text-align: center;
5 }
6 .signin-content {
7   padding: 60px 1rem;
8 }

```

```

9 .full-width-input {
10   width: 100%;
11 }

```

5 Servicio Auth

```

1 import { Injectable } from '@angular/core';
2 import { BehaviorSubject } from 'rxjs';
3 import { Router } from '@angular/router';
4 import { IUserCredentials } from '../auth/user-credentials';
5
6 @Injectable({
7   providedIn: 'root'
8 })
9 export class AuthService {
10
11   private loggedIn = new BehaviorSubject<boolean>(false); // 1
12
13   get isLoggedIn() {
14     console.log(this.loggedIn.getValue());
15     return this.loggedIn.asObservable(); // {2}
16   }
17
18   constructor(private router: Router) { }
19
20   login(user: IUserCredentials) {
21     if (user.userName !== '' && user.password !== '' ) { //
22       {3}
23       this.loggedIn.next(true);
24       this.router.navigate(['/']);
25     }
26
27     logout() { // {4} }
28       this.loggedIn.next(false);
29       this.router.navigate(['/login']);
30     }
31 }

```

6 Guarda Auth

```

1 import { Injectable } from '@angular/core';

```

```

2 import { CanActivate, ActivatedRouteSnapshot,
  RouterStateSnapshot, UrlTree, Router } from '@angular/
  router';
3 import Observable from 'rxjs';
4 import map, take from 'rxjs/operators';
5 import AuthService from '../services/auth.service';
6
7 @Injectable({
8   providedIn: 'root'
9 })
10 export class AuthGuard implements CanActivate {
11   constructor(private authService: AuthService,
12     private router: Router) { }
13
14   canActivate(
15     next: ActivatedRouteSnapshot,
16     state: RouterStateSnapshot): Observable<boolean> {
17     return this.authService.isLoggedIn // 1
18       .pipe(
19         take(1), // 2
20         map((isLoggedIn: boolean) => { // {3}
21           if (!isLoggedIn) { }
22             this.router.navigate(['/login']); // {4}
23             return false;
24           }
25           return true;
26         })
27       );
28   }
29 }

```

7 Componentes Header y Footer

7.1 Modificaciones al componente header

7.1.1 header.component.ts

```

1 import { Component, OnInit } from '@angular/core';
2 import AuthService from '../services/auth.service';
3 import Observable from 'rxjs';
4
5 @Component({
6   selector: 'app-header',
7   templateUrl: './header.component.html',
8   styleUrls: ['./header.component.scss']

```

```

9  })
10 export class HeaderComponent implements OnInit {
11
12     isLoggedIn$: Observable<boolean>; // 1
13
14     constructor(private _authService: AuthService) { }
15
16     ngOnInit() {
17         this.isLoggedIn$ = this._authService.isLoggedIn; // 2
18     }
19
20     onLogout() {
21         this._authService.logout();
22     }
23 }

```

7.1.2 header.component.html

```

1  <mat-toolbar color="primary" *ngIf="isLoggedIn$ | async as
    isLoggedIn">
2      <mat-toolbar-row>
3          <span>
4              
5                  Lenin Lemus' Web Site
6              </span>
7              <span class="spacer"></span>
8              <mat-icon routerLink="/principal" class="icon" aria-
                hidden="false" aria-label="Home">home</mat-icon>
9              <button mat-button routerLink="/diary">Diary</button>
10             <button mat-button routerLink="/studentManagement">
                Students Mgm</button>
11             <button mat-button routerLink="/"
                visitedPlaceManagement">Visited Places Mgm</button>
12             <button mat-button routerLink="/about">About</button>
13             <button mat-button
                (click)="onLogout()">Logout</button>
14         </mat-toolbar-row>
15 </mat-toolbar>

```

7.1.3 header.component.scss

```

1 mat-card {
2     max-width: 400px;
3     margin: 2em auto;
4     text-align: center;

```



```
5 }
6 .signin-content {
7   padding: 60px 1rem;
8 }
9 .full-width-input {
10   width: 100%;
11 }
```

7.2 Modificaciones al componente footer

7.2.1 footer.component.ts

```
1 import { Component, OnInit } from '@angular/core';
2 import { Observable } from 'rxjs';
3 import { AuthService } from '../services/auth.service';
4
5 @Component({
6   selector: 'app-footer',
7   templateUrl: './footer.component.html',
8   styleUrls: ['./footer.component.scss']
9 })
10 export class FooterComponent implements OnInit {
11
12   isLoggedIn$: Observable<boolean>; // 1
13
14   constructor(private _authService: AuthService)
15
16   ngOnInit() {
17     this.isLoggedIn$ = this._authService.isLoggedIn; // 2
18   }
19
20 }
```

7.2.2 footer.component.html

```
1 <mat-toolbar color="primary" *ngIf="isLoggedIn$ | async as
   isLoggedIn">
2   <mat-toolbar-row>
3     <span>&copy; 2019. Lenin Lemus</span>
4     <span class="spacer"></span>
5     <mat-icon class="icon" aria-hidden="false" aria-label
      ="Example heart icon">favorite</mat-icon>
6   </mat-toolbar-row>
7 </mat-toolbar>
```

7.2.3 footer.component.scss

```

1 .icon {
2   padding: 0 14px;
3 }
4
5 .spacer {
6   flex: 1 1 auto;
7 }

```

8 Routing

8.1 app-routing.component.ts

```

1 import { NgModule } from '@angular/core';
2 import { Routes, RouterModule } from '@angular/router';
3 import { PrincipalComponent } from '../principal/principal.
  component';
4 import { NotFoundComponent } from '../not-found/not-found.
  component';
5 import { AboutComponent } from '../about/about.component';
6 import AuthGuard from '../auth/auth.guard';
7 import LoginComponent from '../login/login.component';
8
9 const routes: Routes = [
10   path: '', redirectTo: 'principal', pathMatch: 'full',
      canActivate: [AuthGuard], @*) (*@ path: 'login',
      component: LoginComponent, @*) (*@ path: 'principal',
      component: PrincipalComponent, canActivate: [AuthGuard],
      @*) (*@ path: 'about', component: AboutComponent, @*) (*@
      path: 'diary', loadChildren: () =>
      import('../diary/diary.module').then(m => m.DairyModule),
      canActivate: [AuthGuard], @*) (*@ path:
      'studentManagement', loadChildren:
      '../student/student.module#StudentModule', canActivate:
      [AuthGuard], @*) (*@ path: 'visitedPlaceManagement',
      loadChildren: () =>
      import('../visited-place/visited-place.module').then(m =>
      m.VisitedPlaceModule), canActivate: [AuthGuard], @*)
      path: '**', component:
      NotFoundComponent,];@NgModule(imports:
      [RouterModule.forRoot(routes)], exports:
      [RouterModule])export class AppRoutingModule

```

9 Aplicación en funcionamiento

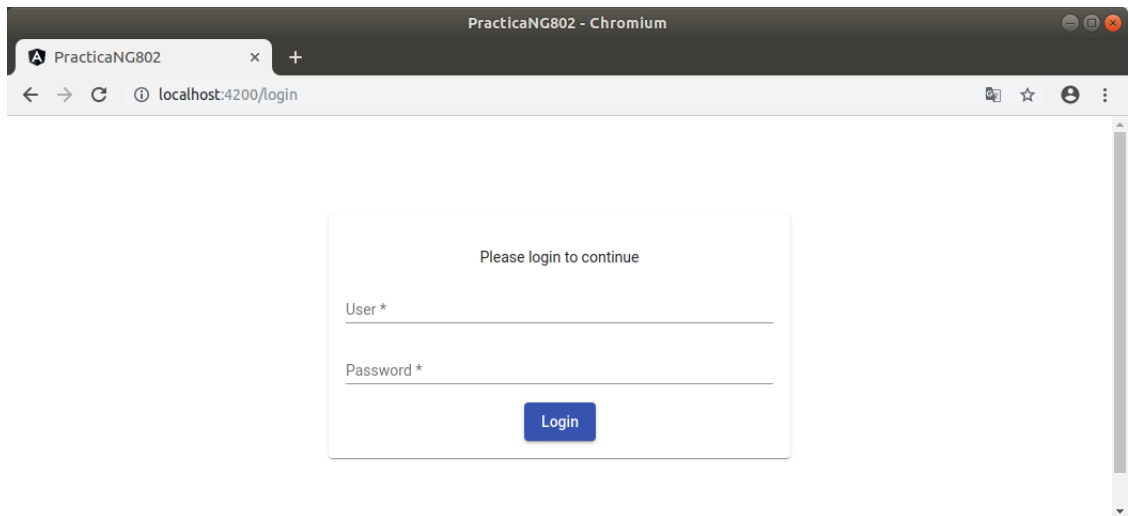


Figure 1: Al iniciar la aplicación se muestra la página de autenticación.

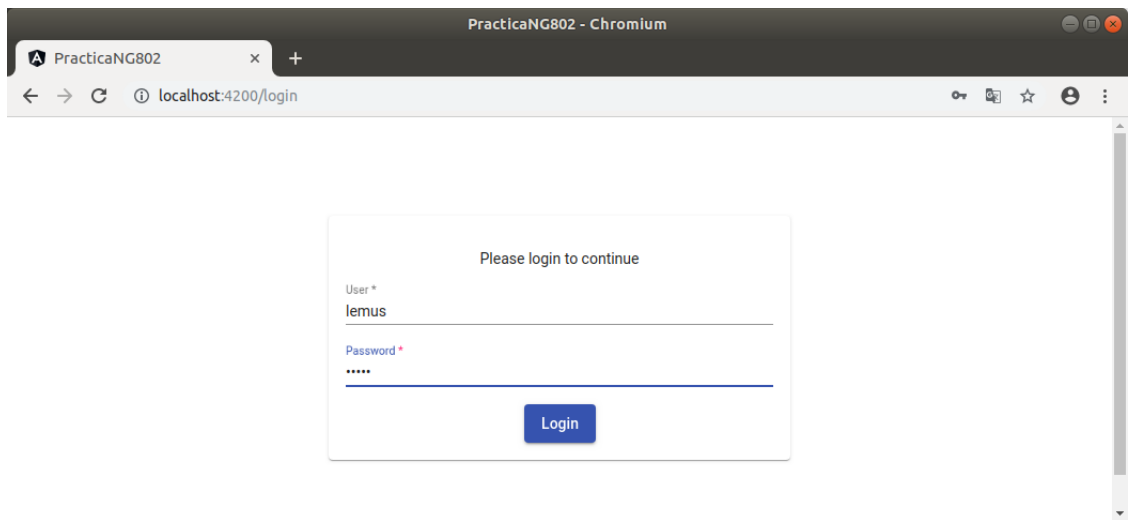


Figure 2: El usuario debe colocar su nombre de ususario y su clave.

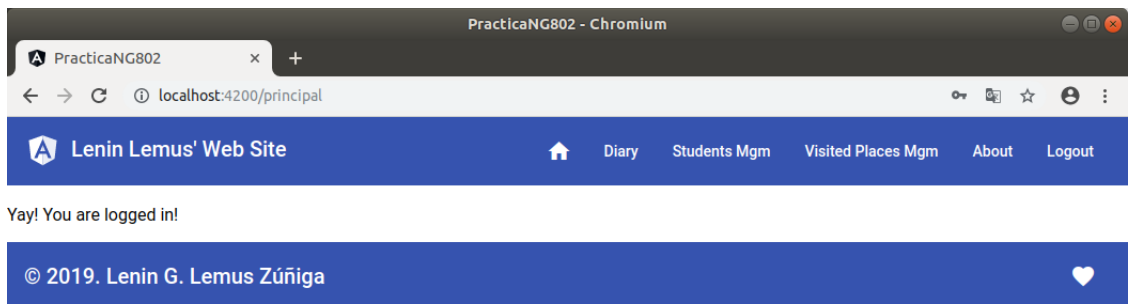


Figure 3: Si las credenciales son correctas se muestra la página principal.

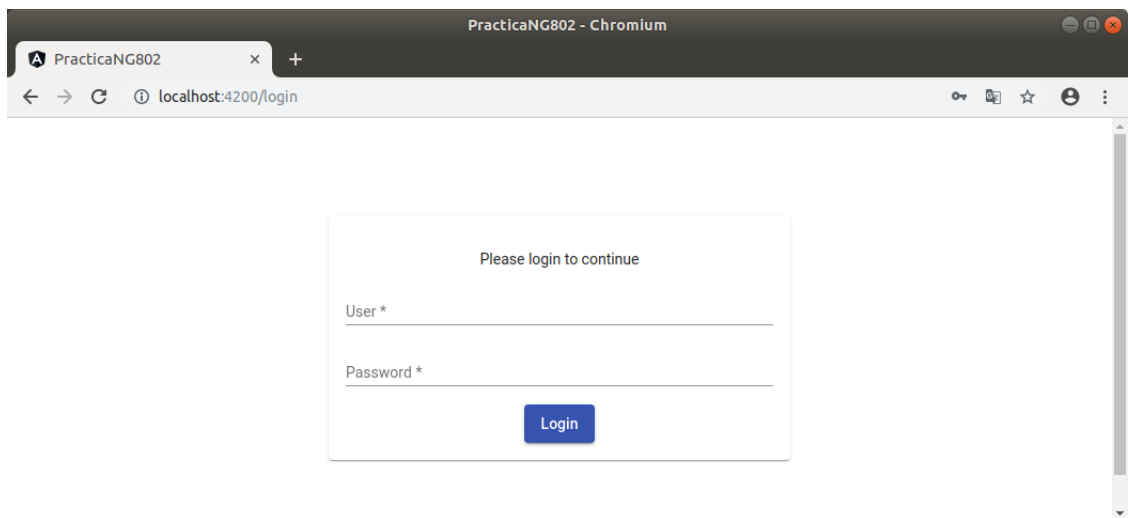


Figure 4: Al pulsar la opción logout, se hace que la variable IsLoggedIn tenga el valor de falso por lo que se vuelve a mostrar la página de login.