

# Login en Angular – parte 3



## Login en Angular – Parte 3

En esta lección, vamos a agregar la funcionalidad completa de para hacer login y logout de nuestra aplicación, además de modificar la barra de navegación para agregar los elementos de login o logout dependiendo de si el usuario ya ha hecho login o no. Para ello vamos a modificar varios componentes.

### Paso 1: Modificar el servicio de login

En primer lugar, vamos a modificar el servicio de login, para ello agregamos el método de `getAuthState` para saber si un usuario ya ha hecho login o no, además del método `logout` para salir del sistema.

### Código archivo login.service.ts:

```
import { Injectable } from '@angular/core';
import {
  Auth,
  authState,
  signInWithEmailAndPassword,
} from '@angular/fire/auth';
import { Observable } from 'rxjs';

@Injectable({
```

```

providedIn: 'root', // Mantiene el servicio como inyectable en toda la aplicación
})
export class LoginService {
    // Inyectamos el servicio Auth en el constructor
    constructor(private authService: Auth) {}

    login(email: string, password: string) {
        return new Promise((resolve, reject) => {
            signInWithEmailAndPassword(this.authService, email, password)
                .then((datos) => resolve(datos))
                .catch((error) => reject(error));
        });
    }

    // Obtenemos el usuario logueado
    getAuthState(): Observable<any> {
        return authState(this.authService);
    }

    logout() {
        this.authService.signOut();
    }
}

```

## Paso 2: Modificar el componente de cabecera

Modificamos el componente de cabecera para mostrar u ocultar las opciones de la barra de navegación para hacer login o salir del sistema dependiendo del si el usuario ya ha hecho login o no en el sistema. Además definimos algunas variables para poder controlar estos estados.

### Código archivo cabecera.component.ts:

```

import { Component } from '@angular/core';
import { Router, RouterModule } from '@angular/router';
import { LoginService } from '../../servicios/login.service';

@Component({
    selector: 'app-cabecera',
    standalone: true,
    imports: [RouterModule],
    templateUrl: './cabecera.component.html',
    styleUrls: ['./cabecera.component.css']
})

```

```

})
export class CabeceraComponent {
  isLoggedIn: boolean = false;
  loggedInUser: string | null = null;

  constructor(
    private loginService: LoginService,
    private router: Router
  ) { }

  ngOnInit() {
    this.loginService.getAuthState().subscribe( usuario => {
      if(usuario){
        this.isLoggedIn = true;
        this.loggedInUser = usuario.email;
      }
      else{
        this.isLoggedIn = false;
      }
    });
  }

  logout(){
    this.loginService.logout();
    this.isLoggedIn = false;
    this.router.navigate(['/login']);
  }
}

```

Ahora modificamos la plantilla del componente de cabecera para poder validar si un usuario ya ha hecho login o no, y dependiendo de eso mostrar o no las opciones de login o logout.

### Código archivo cabecera.component.html:

```

<nav class="navbar navbar-expand-sm navbar-dark bg-dark p-0">
  <div class="container">
    <a routerLink="/" class="navbar-brand">Control Clientes</a>
    <button
      class="navbar-toggler"
      type="button"
      data-bs-toggle="collapse"
      data-bs-target="#navbarCollapse"
    >

```

```
>
  <span class="navbar-toggler-icon"></span>
</button>

<div class="collapse navbar-collapse" id="navbarCollapse">
  <!-- Menú visible solo si el usuario está autenticado -->
  @if (isLoggedIn) {
    <ul class="navbar-nav me-auto">
      <li class="nav-item px-2">
        <a routerLink="/" class="nav-link">Inicio</a>
      </li>
    </ul>
  }

  <ul class="navbar-nav">
    @if (isLoggedIn) {
      <li class="nav-item px-2">
        <a [routerLink]="/'" class="nav-link">
          <i class="bi bi-person"></i> Bienvenido {{ loggedInUser }}
        </a>
      </li>
    }
  
```

<!-- Opción de login visible si el usuario no está autenticado -->

```
@if (!isLoggedIn) {
  <li class="nav-item px-2">
    <a routerLink="/login" class="nav-link">
      <i class="bi bi-person"></i> Login
    </a>
  </li>
}

<!-- Opción de salir visible si el usuario está autenticado -->
```

```
@if (isLoggedIn) {
  <li class="nav-item px-2">
    <a style="cursor: pointer" (click)="logout()" class="nav-link">
      <i class="bi bi-box-arrow-right"></i> Salir
    </a>
  </li>
}
</ul>
</div>
</div>
</nav>
```

### Paso 3: Modificar el componente de login

Evitamos que se muestre el componente y formulario de login si el usuario ya ha hecho login, enviándolo a la página de inicio en el método ngOnInit():

Código del archivo `login.component.ts`:

```
import { Component } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { Router } from '@angular/router';
import { LoginService } from '../../../../../servicios/login.service';

@Component({
  selector: 'app-login',
  standalone: true,
  imports: [FormsModule],
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.css']
})
export class LoginComponent {
  email: string | null = null;
  password: string | null = null;
  mensaje: string | null = null;

  constructor(
    private router: Router,
    private loginService: LoginService
  ) { }

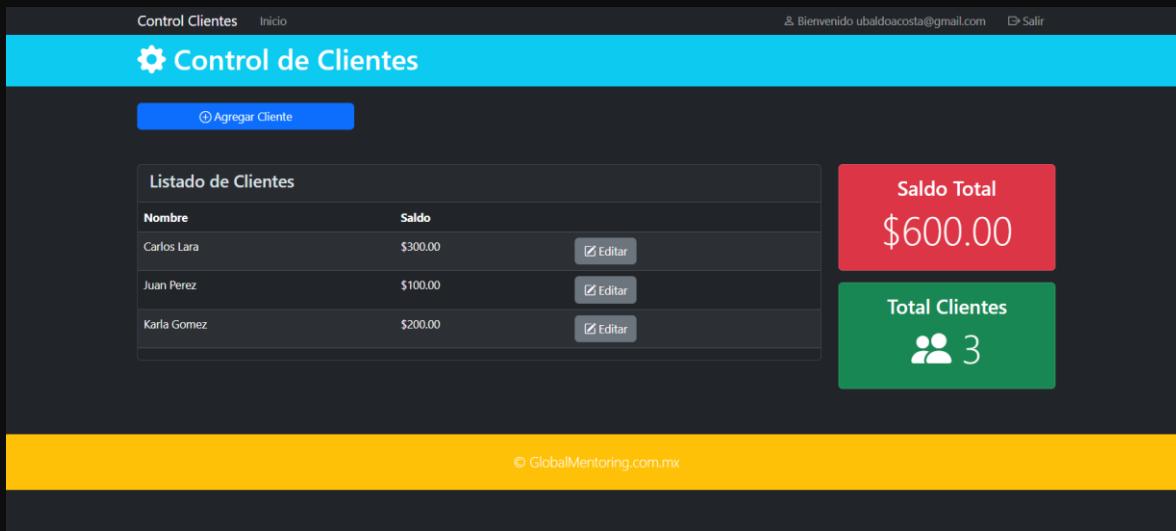
  // Evita que es muestre el formulario de login
  // si el usuario ya esta logueado
  ngOnInit() {
    this.loginService.getAuthState().subscribe(usuario => {
      if(usuario){
        this.router.navigate(['/']);
      }
    })
  }

  login() {
    // Verificación de que email y password no son null
    if (this.email && this.password) {
      this.loginService.login(this.email, this.password)
        .then(() => {
          this.router.navigate(['/']);
        })
    }
  }
}
```

```
        })
        .catch(error => {
            this.mensaje = 'Error al hacer login: ' + error;
        });
    } else {
        this.mensaje = 'Por favor, ingrese un email y una contraseña válidos.';
    }
}
}
```

## Conclusión

Con estos pasos, hemos implementado exitosamente las opciones de login / logout desde nuestra aplicación.



Recuerden que pueden descargar el código actualizado en la sección de recursos de esta lección.

¡Continúa así 🚀, el esfuerzo vale la pena!

Saludos ✌

Ing. Ubaldo Acosta

Fundador de [GlobalMentoring.com.mx](http://www.globalmentoring.com.mx)