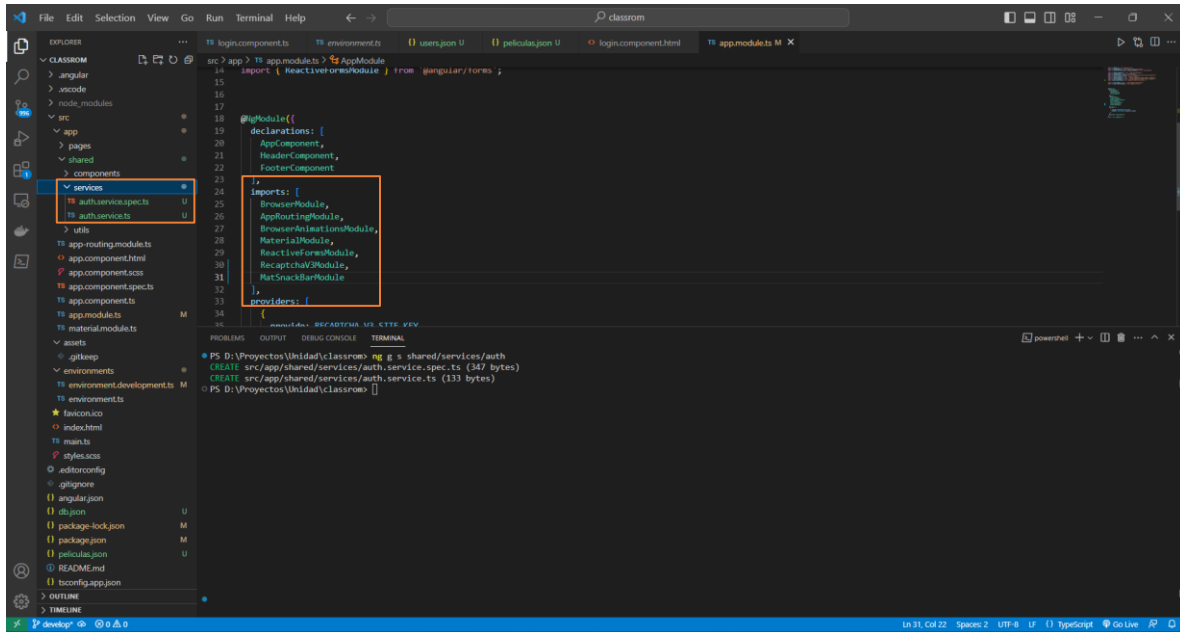


## Crear servicio de Autenticación

Primeramente ,agregar módulo [MatSnackBarModule](#) al archivo **material.module.ts** para usar el servicio de mensajes; modificar el archivo **auth.service.ts**

Abrir terminal y ejecutar el siguiente comando para crear el servicio de autenticación: **ng g s shared/services/auth**



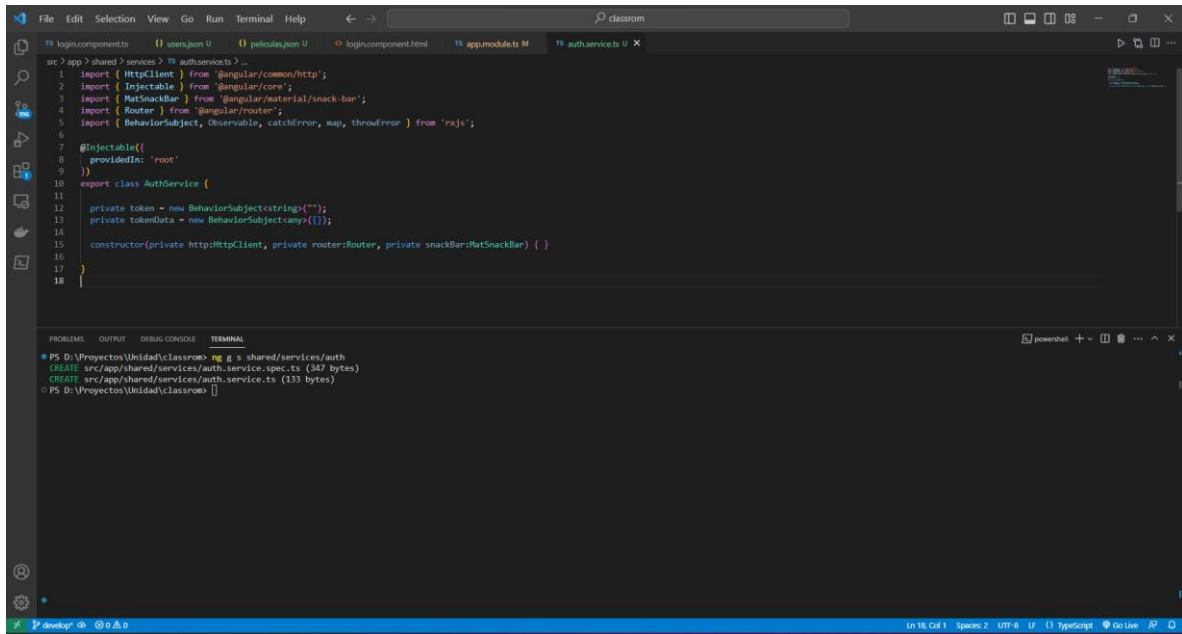
Agregamos la siguiente instrucción para ocupar elementos a utilizar para el manejo del Token y datos de usuario

```
import { BehaviorSubject, Observable, catchError, map, throwError }  
from 'rxjs';
```

Agregar las propiedades a la clase AuthService:

**token** Es el token encriptado

**tokenData** Datos de usuario encriptado qué es el Payload

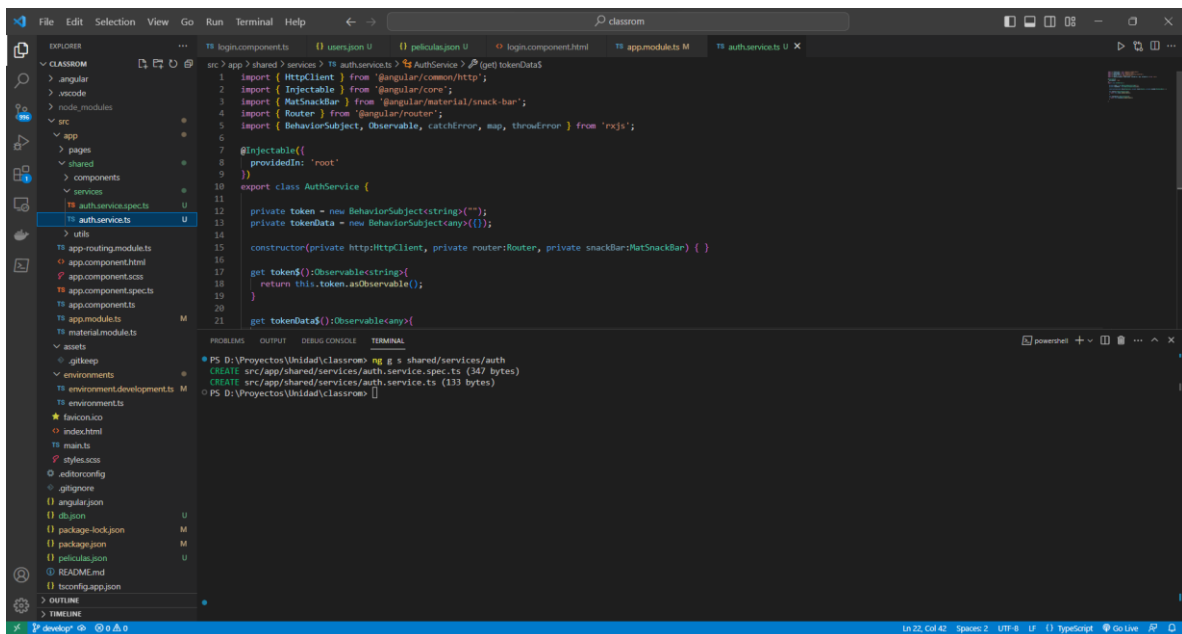


```
src > app > shared > services > ts authservices > ...
1 import { HttpClient } from '@angular/common/http';
2 import { Injectable } from '@angular/core';
3 import { MatSnackBar } from '@angular/material/snack-bar';
4 import { Router } from '@angular/router';
5 import { BehaviorSubject, Observable, catchError, map, throwError } from 'rxjs';
6
7 @Injectable({
8   providedIn: 'root'
9 })
10 export class AuthService {
11
12   private token = new BehaviorSubject<string>('');
13   private tokenData = new BehaviorSubject<any>({});
14
15   constructor(private httpClient:HttpClient, private router:Router, private snackBar:MatSnackBar) {}
16
17 }
18
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\Proyectos\Unidad\classroom> ng g s shared/services/auth
CREATE src/app/shared/services/auth.service.spec.ts (347 bytes)
CREATE src/app/shared/services/auth.service.ts (133 bytes)
PS D:\Proyectos\Unidad\classroom>
```

Agregar las siguientes instrucciones al archivo **auth.service.ts** con el objetivo de definir las propiedades **token** y **tokenData** como Observables y permitir suscripciones y saber de sus valores



```
src > app > shared > services > ts AuthService > ts get tokenData
1 import { HttpClient } from '@angular/common/http';
2 import { Injectable } from '@angular/core';
3 import { MatSnackBar } from '@angular/material/snack-bar';
4 import { Router } from '@angular/router';
5 import { BehaviorSubject, Observable, catchError, map, throwError } from 'rxjs';
6
7 @Injectable({
8   providedIn: 'root'
9 })
10 export class AuthService {
11
12   private token = new BehaviorSubject<string>('');
13   private tokenData = new BehaviorSubject<any>({});
14
15   constructor(private httpClient:HttpClient, private router:Router, private snackBar:MatSnackBar) {}
16
17   get token():Observable<string>{
18     return this.token.asObservable();
19   }
20
21   get tokenData():Observable<any>{
22
23   }
24 }
25
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\Proyectos\Unidad\classroom> ng g s shared/services/auth
CREATE src/app/shared/services/auth.service.spec.ts (347 bytes)
CREATE src/app/shared/services/auth.service.ts (133 bytes)
PS D:\Proyectos\Unidad\classroom>
```

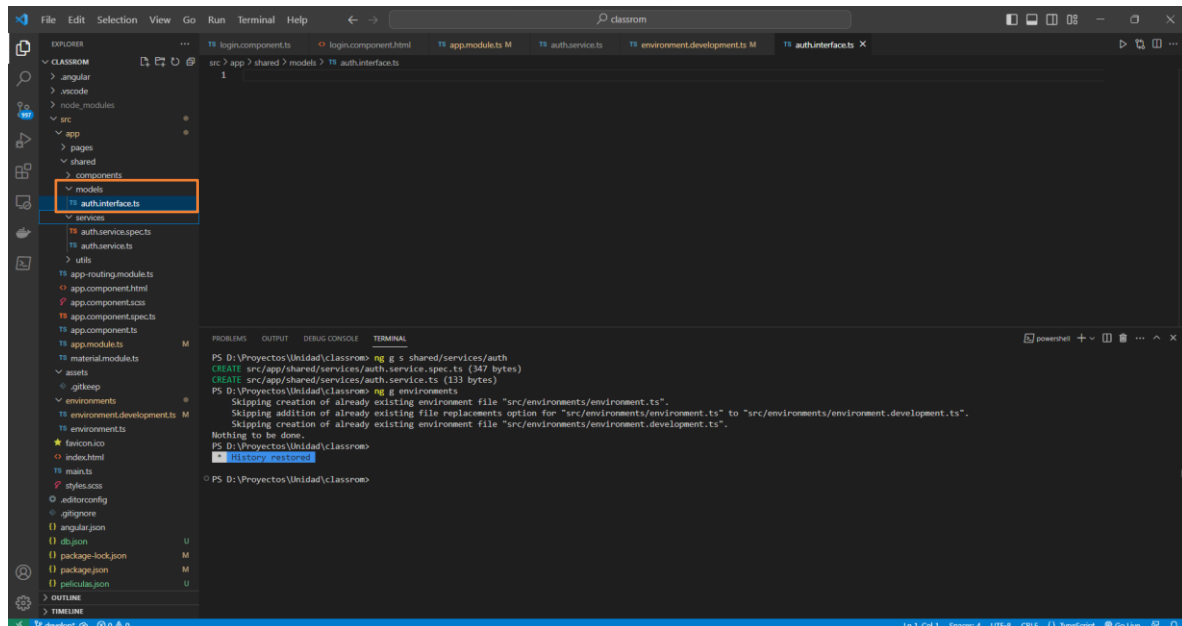
Modificar los archivos environments de la siguiente forma

```
src > environments > TS environment.development.ts > ...
1  export const environment = {
2    production: false,
3    recaptcha:{
4      siteKey:'6LetkYQmAAAAAwFQ8U2tu08mCm0wFGH0vuwS6se',
5    },
6    API_URL : 'http://localhost:3000/',
7  };
8
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\Proyectos\Unidad\classroom> ng g s shared/services/auth
CREATE src/app/shared/services/auth.service.spec.ts (347 bytes)
CREATE src/app/shared/services/auth.service.ts (133 bytes)
PS D:\Proyectos\Unidad\classroom> ng g environments
Skipping creation of already existing environment file "src/environments/environment.ts".
Skipping addition of already existing file replacements option for "src/environments/environment.ts" to "src/environments/environment.development.ts".
Skipping creation of already existing environment file "src/environments/environment.development.ts".
Nothing to be done.
PS D:\Proyectos\Unidad\classroom>
* History restored
PS D:\Proyectos\Unidad\classroom>
```

En carpeta **shared/models** agregar un archivo con el nombre **auth.interface.ts**, en caso que no exista se deberá de crear



## Modificar el archivo `auth.interface.ts`

```
login.component.ts login.component.html TS app.module.ts M TS auth.service.ts TS environment.development.ts M TS auth.interface.ts X
src > app > shared > models > TS auth.interface.ts > AuthResponse > token
1 export interface AuthResponse {
2   message: string;
3   token: string;
4 }
```

## Agregar el siguiente método `handleError` al archivo `auth.service.ts`

```
File Edit Selection View Go Run Terminal Help classrom
src > app > shared > services > TS auth.service.ts > ...
28 * @returns lanza error en caso que el mensaje no esté nulo o vacío
29 */
30 handleError(error: any): Observable<any> {
31   let message = "Ha ocurrido un error";
32
33   if (error) {
34     message = `${error.error.message}`;
35   }
36   console.log('Handle Error ' + message);
37
38   this.snackBar.open(message, '', {
39     duration: 5000,
40     panelClass: ['error-snackbar'],
41     horizontalPosition: 'right',
42     verticalPosition: 'top'
43   });
44   return throwError(message);
45 }
46
47
48
49
50
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
CREATE src/app/shared/services/auth.service.spec.ts (347 bytes)
CREATE src/app/shared/services/auth.service.ts (133 bytes)
PS D:\Proyectos\Unidad\classrom> ng g environments
Skipping creation of already existing environment file "src/environments/environment.ts".
Skipping addition of already existing file replacements option for "src/environments/environment.ts" to "src/environments/environment.development.ts".
Skipping creation of already existing environment file "src/environments/environment.development.ts".
Nothing to be done.
PS D:\Proyectos\Unidad\classrom>
History restored
PS D:\Proyectos\Unidad\classrom>
```

## Instalar el módulo `@auth0/angular-jwt`

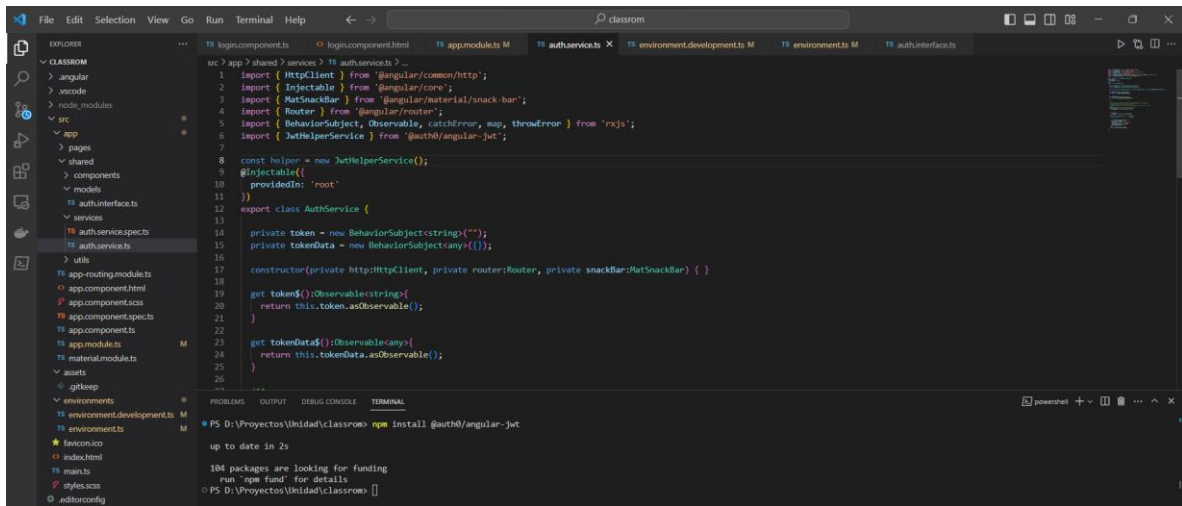
```
PS D:\Proyectos\Unidad\classrom> npm install @auth0/angular-jwt

up to date in 2s

104 packages are looking for funding
run `npm fund` for details
PS D:\Proyectos\Unidad\classrom>
```

Importar clase JwtHelperService

Agregar una variable llamada helper en auth.service.ts

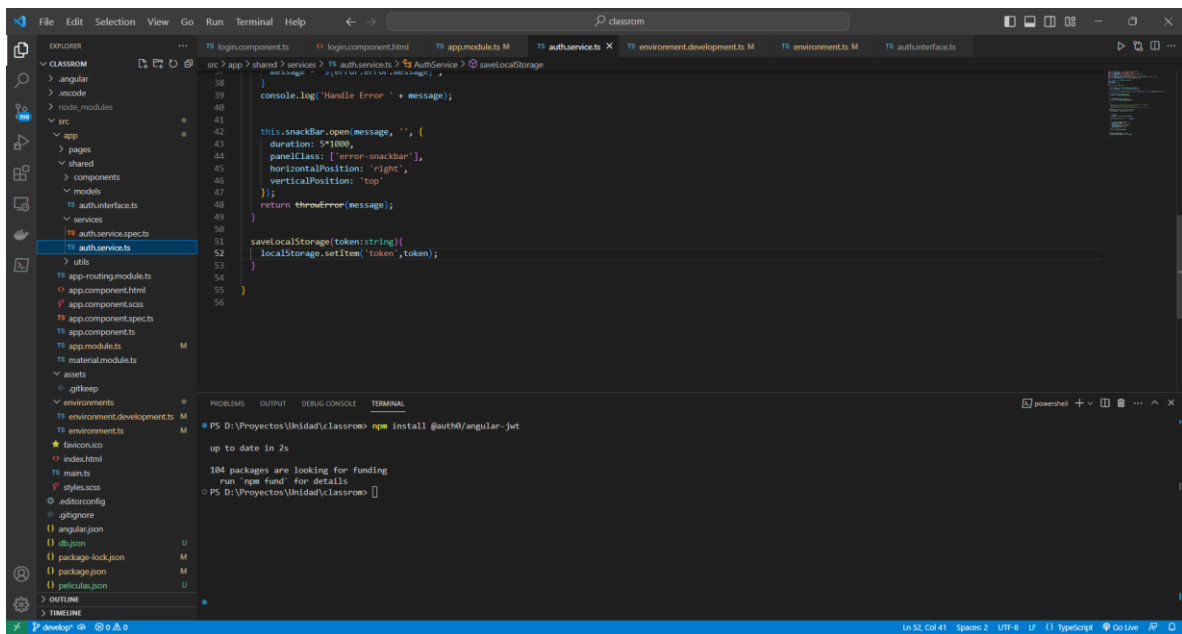


The screenshot shows the Visual Studio Code editor with the file explorer on the left. The file explorer shows the project structure, including the 'src' folder and the 'auth.service.ts' file. The 'auth.service.ts' file is open in the editor, showing the following code:

```
src > app > shared > services > auth.service.ts > ...
1 import { HttpClient } from '@angular/common/http';
2 import { Injectable } from '@angular/core';
3 import { MatSnackBar } from '@angular/material/snack-bar';
4 import { Router } from '@angular/router';
5 import { BehaviorSubject, Observable, catchError, map, throwError } from 'rxjs';
6 import { JwtHelperService } from '@auth0/angular-jwt';
7
8 const helper = new JwtHelperService();
9 @Injectable({
10   providedIn: 'root'
11 })
12 export class AuthService {
13
14   private token = new BehaviorSubject<string>('');
15   private tokenData = new BehaviorSubject<any>({});
16
17   constructor(private http: HttpClient, private router: Router, private snackBar: MatSnackBar) {}
18
19   get token$(): Observable<string> {
20     return this.token.asObservable();
21   }
22
23   get tokenData$(): Observable<any> {
24     return this.tokenData.asObservable();
25   }
26 }
```

The terminal at the bottom shows the command `npm install @auth0/angular-jwt` being executed, with the output indicating that the package is up to date.

Agregar el método **saveLocalStorage** en archivo **auth.service.ts**

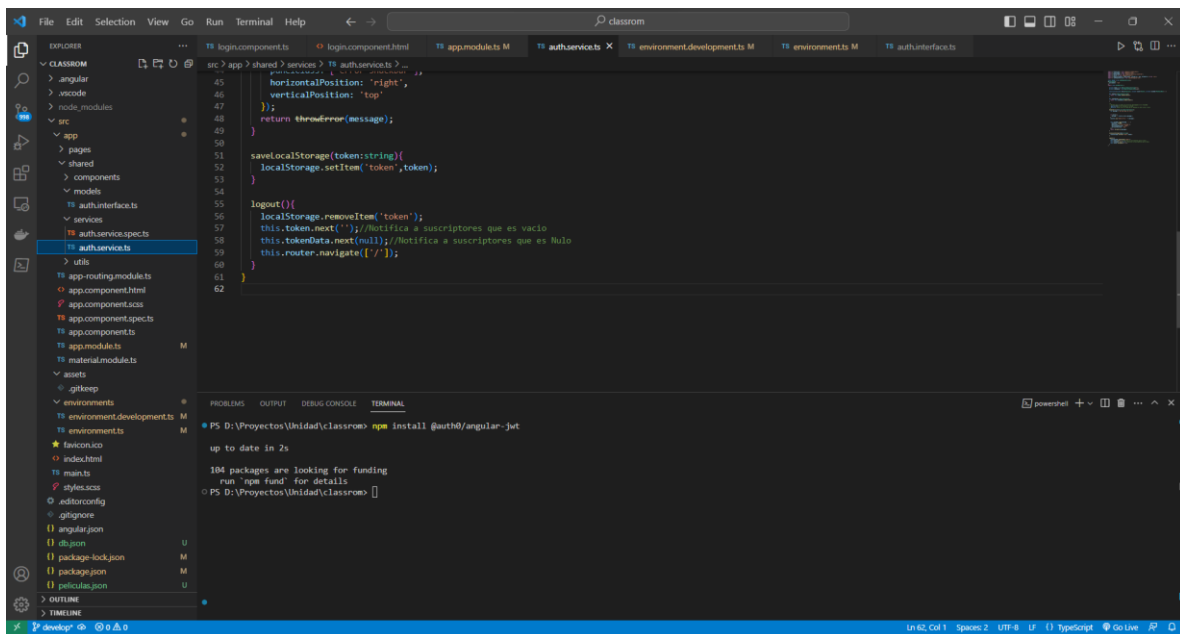


The screenshot shows the Visual Studio Code editor with the file explorer on the left. The file explorer shows the project structure, including the 'src' folder and the 'auth.service.ts' file. The 'auth.service.ts' file is open in the editor, showing the following code:

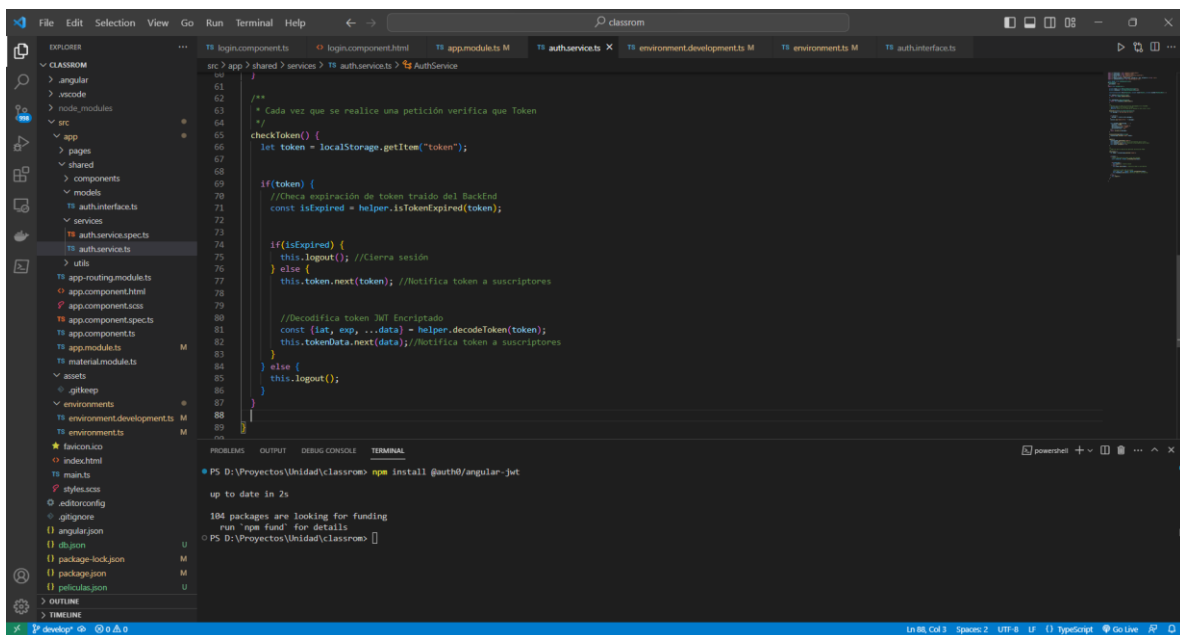
```
src > app > shared > services > auth.service.ts > AuthService > saveLocalStorage
38
39 console.log('Handle Error ' + message);
40
41
42 this.snackBar.open(message, '', {
43   duration: 5 * 1000,
44   panelClass: ['error-snackbar'],
45   horizontalPosition: 'right',
46   verticalPosition: 'top'
47 });
48 return throwError(message);
49 }
50
51 saveLocalStorage(token:string){
52   localStorage.setItem('token',token);
53 }
54
55
56 }
```

The terminal at the bottom shows the command `npm install @auth0/angular-jwt` being executed, with the output indicating that the package is up to date.

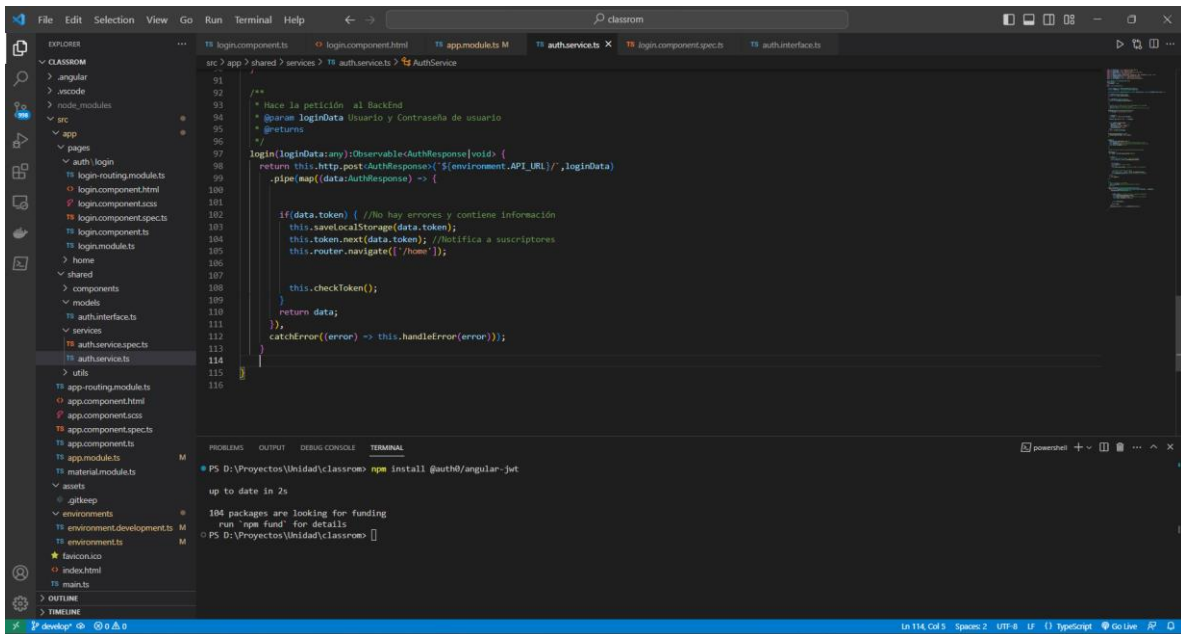
## Agregar el método **logout** en archivo **auth.service.ts**



## Agregar el método **checklogin** en archivo **auth.service.ts**



Modificar método login de la siguiente forma:

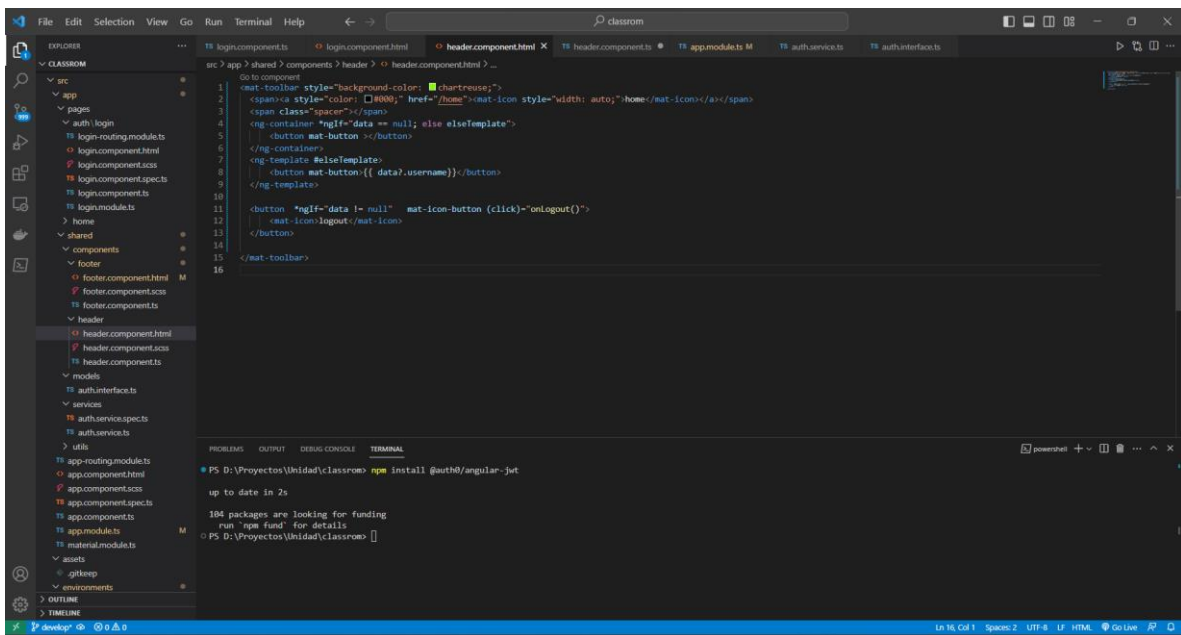


The screenshot shows the Visual Studio Code editor with the file `src/app/shared/services/auth.service.ts` open. The `login` method is implemented as follows:

```
login(loginData:any):Observable<AuthResponse>[void] {  
  return this.http.post<AuthResponse>(`${environment.API_URL}/`, loginData)  
    .pipe(map((data:AuthResponse) => {  
  
      if(data.token) { //No hay errores y contiene información  
        this.saveLocalStorage(data.token);  
        this.token.next(data.token); //Notifica a suscriptores  
        this.router.navigate(['/home']);  
  
        this.checkToken();  
      }  
      return data;  
    })),  
    catchError(error => this.handleError(error));  
}
```

The left sidebar shows the project structure with the `src/app/shared/services/auth.service.ts` file selected. The bottom terminal shows the command `npm install @auth0/angular-jwt` being executed.

Modificar el archivo `header.component.html`

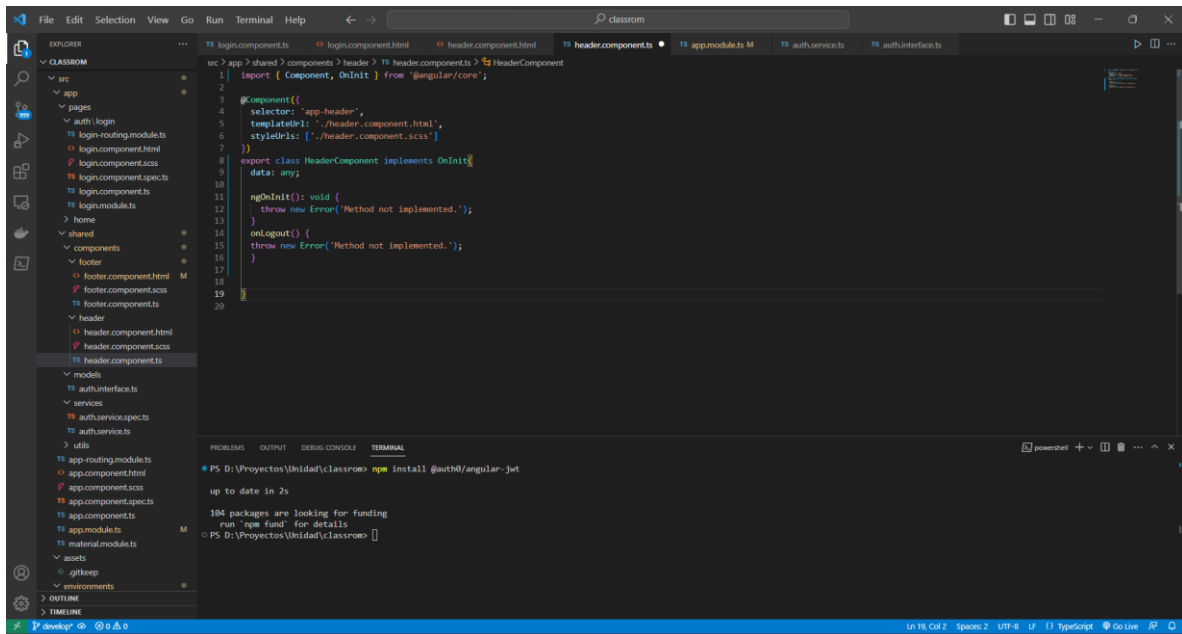


The screenshot shows the Visual Studio Code editor with the file `src/app/shared/components/header/header.component.html` open. The HTML structure is as follows:

```
<mat-toolbar style="background-color: #chartreuse;">  
  <span><a style="color: #000000;" href="/home"><mat-icon style="width: auto;" home"/></a></span>  
  <span class="spacer"></span>  
  <ng-container *ngIf="data == null; else elseTemplate">  
    <button mat-button></button>  
  </ng-container>  
  <ng-template #elseTemplate>  
    <button mat-button>{{ data.username }}</button>  
  </ng-template>  
  <button *ngIf="data != null" mat-icon-button (click)="onLogout()">  
    <mat-icon>logout</mat-icon>  
  </button>  
</mat-toolbar>
```

The left sidebar shows the project structure with the `src/app/shared/components/header/header.component.html` file selected. The bottom terminal shows the command `npm install @auth0/angular-jwt` being executed.

## Modificar el archivo header.component.ts



```
src > app > shared > components > header > TS header.components > TS HeaderComponent
1 import { Component, OnInit } from '@angular/core';
2
3 @Component({
4   selector: 'app-header',
5   templateUrl: './header.component.html',
6   styleUrls: ['./header.component.scss']
7 })
8 export class HeaderComponent implements OnInit {
9   data: any;
10
11   ngOnInit(): void {
12     throw new Error('Method not implemented.');

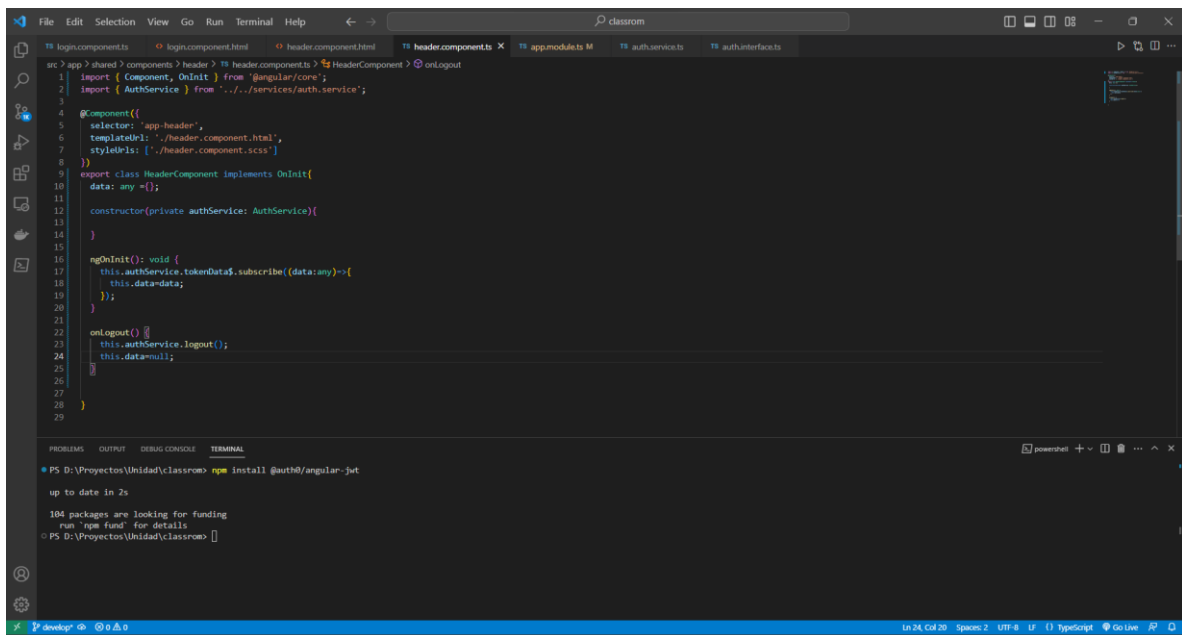
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL



```
PS D:\Proyectos\Unidad\classroom> npm install @auth0/angular-jwt
up to date in 2s
104 packages are looking for funding
  run `npm fund` for details
PS D:\Proyectos\Unidad\classroom>
```


```

## Modificar el archivo header.component.ts



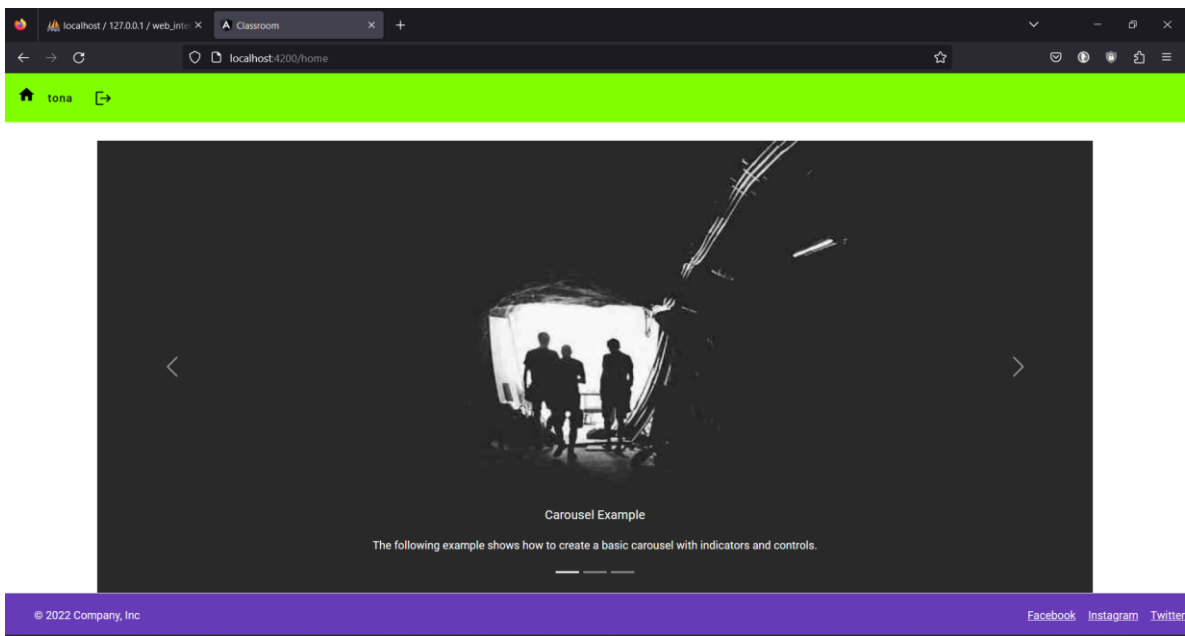
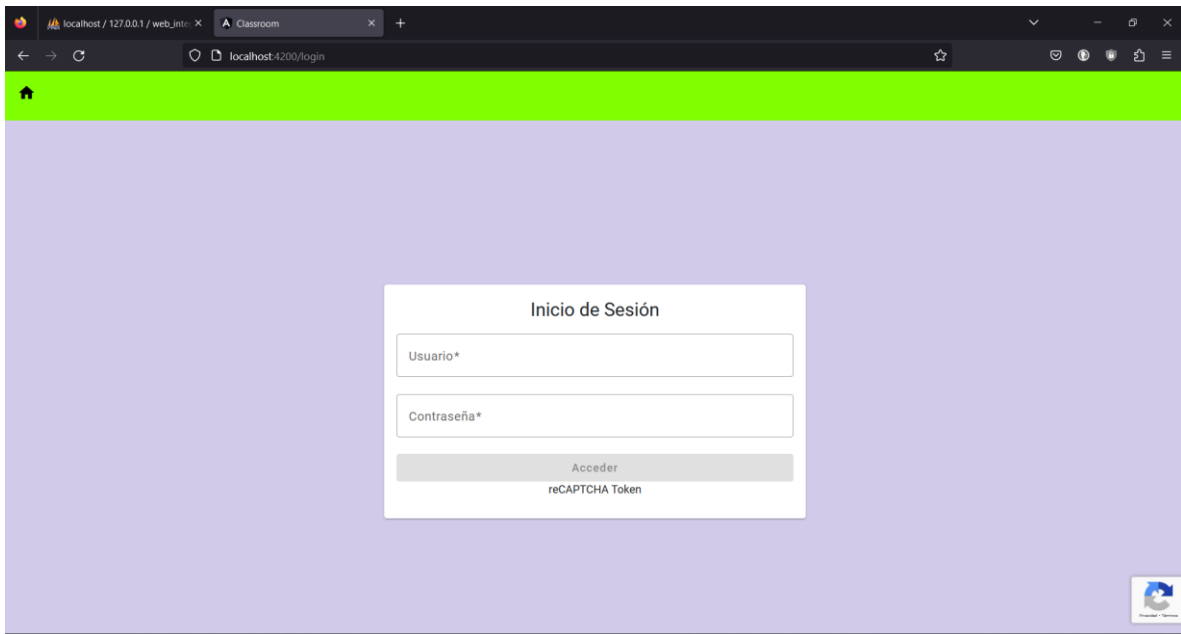
```
src > app > shared > components > header > TS header.components > onLogout
1 import { Component, OnInit } from '@angular/core';
2 import { AuthService } from '../../services/auth.service';
3
4 @Component({
5   selector: 'app-header',
6   templateUrl: './header.component.html',
7   styleUrls: ['./header.component.scss']
8 })
9 export class HeaderComponent implements OnInit {
10   data: any = {};
11
12   constructor(private authService: AuthService) {
13   }
14
15   ngOnInit(): void {
16     this.authService.tokenData$.subscribe((data: any) => {
17       this.data = data;
18     });
19   }
20
21   onLogout() {
22     this.authService.logout();
23     this.data = null;
24   }
25 }
26
27
28
29
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\Proyectos\Unidad\classroom> npm install @auth0/angular-jwt
up to date in 2s
104 packages are looking for funding
  run `npm fund` for details
PS D:\Proyectos\Unidad\classroom>
```

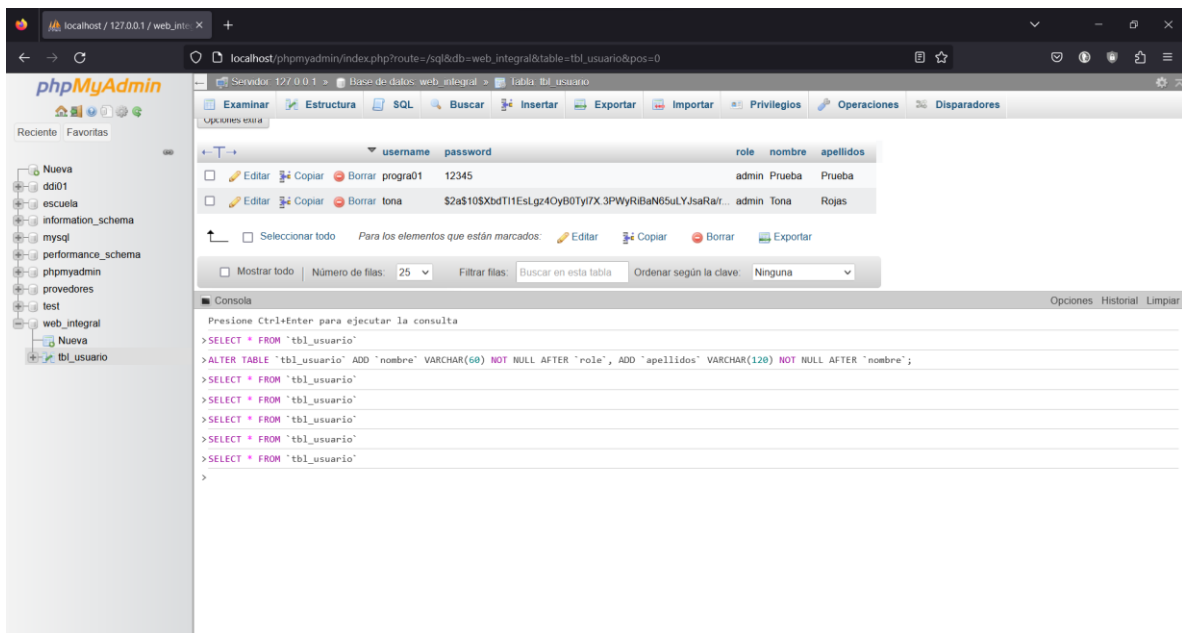


## Funcionamiento del login

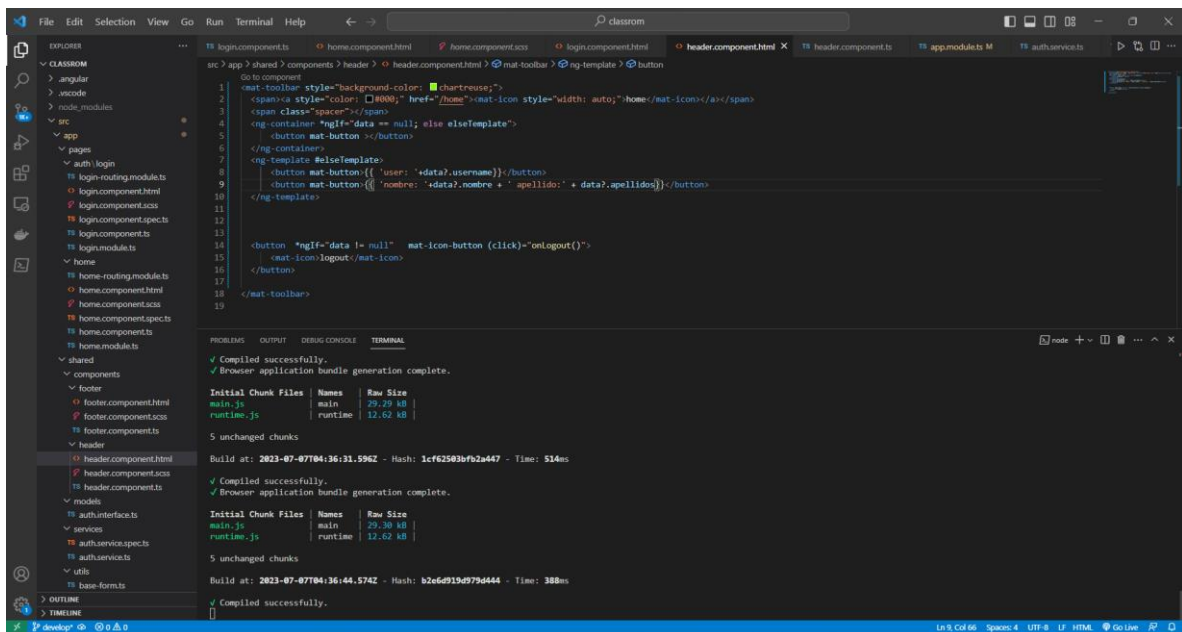


## Desafío

- Agregar más campos a la tabla tbl\_usuario: nombre, apellidos.



Una vez qué está logueado imprimir su nombre completo



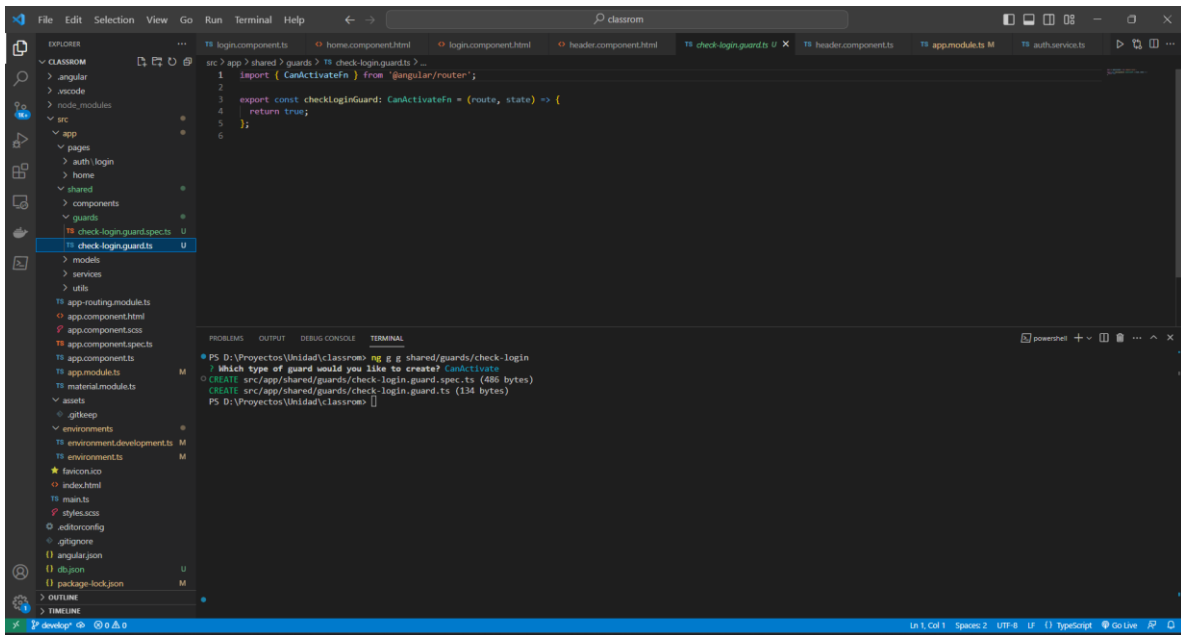


## Protección de rutas

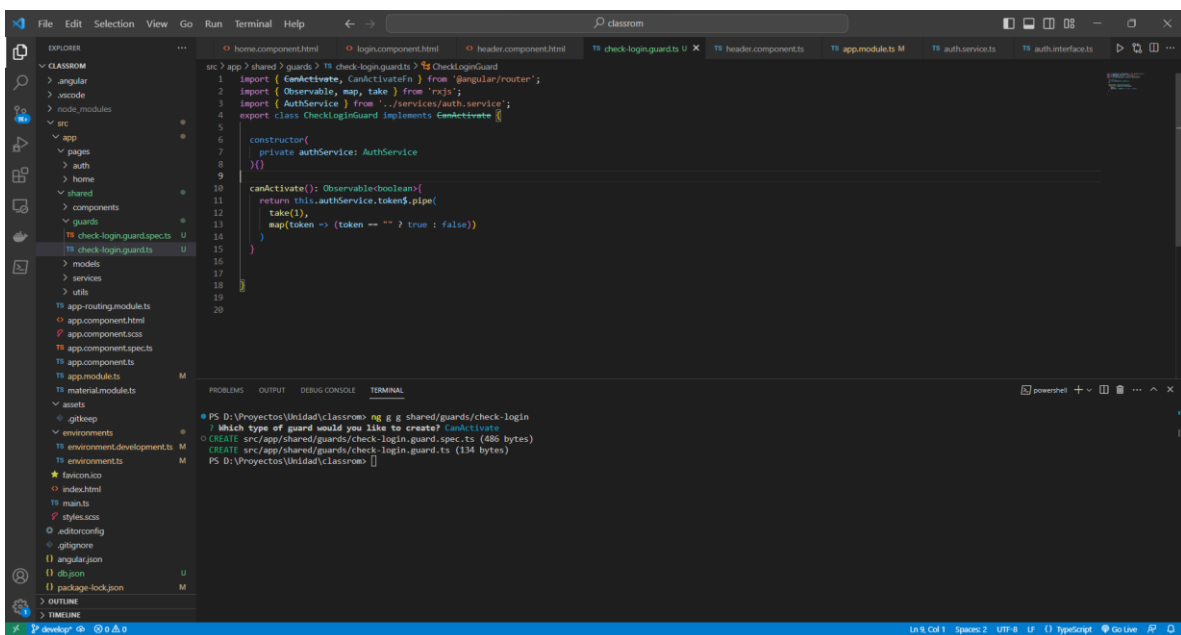
Ejecutar la siguiente línea en la terminal:

**ng g shared/guards/check-login**

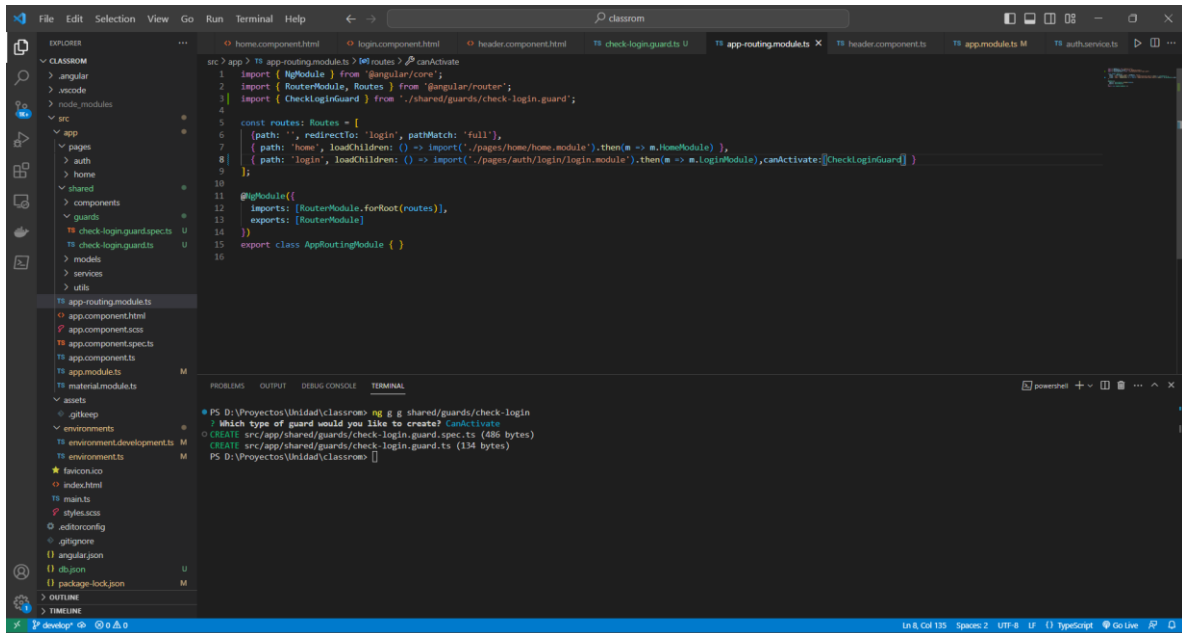
Seleccionar la opción **CanActivate**.



Modificamos el guards



## Modificar el archivo `app-routing.module.ts`



```
src > app > ts app-routing.module.ts > ts routes > ts canActivate
1 import { NgModule } from '@angular/core';
2 import { RouterModule, Routes } from '@angular/router';
3 import { CheckLoginGuard } from '../shared/guards/check-login.guard';
4
5 const routes: Routes = [
6   { path: '', redirectTo: 'login', pathMatch: 'full' },
7   { path: 'home', loadChildren: () => import('../pages/home/home.module').then(m => m.HomeModule) },
8   { path: 'login', loadChildren: () => import('../pages/auth/login/login.module').then(m => m.LoginModule), canActivate: [CheckLoginGuard] },
9 ];
10
11 @NgModule({
12   imports: [RouterModule.forRoot(routes)],
13   exports: [RouterModule]
14 })
15 export class AppRoutingModuleModule {}
16
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\Proyectos\Unidad\classroom> ng g g shared/guards/check-login
? Which type of guard would you like to create? (anActivate)
CREATE src/app/shared/guards/check-login.guard.spec.ts (486 bytes)
CREATE src/app/shared/guards/check-login.guard.ts (134 bytes)
PS D:\Proyectos\Unidad\classroom>
```

