

## Desarrollo Web Integral

Fatima Natalia Ruiz Rivera

1220100350

GDGS0391

DOCENTE:

**Gabriel Barrón Rodriguez**

**06/julio/2023**

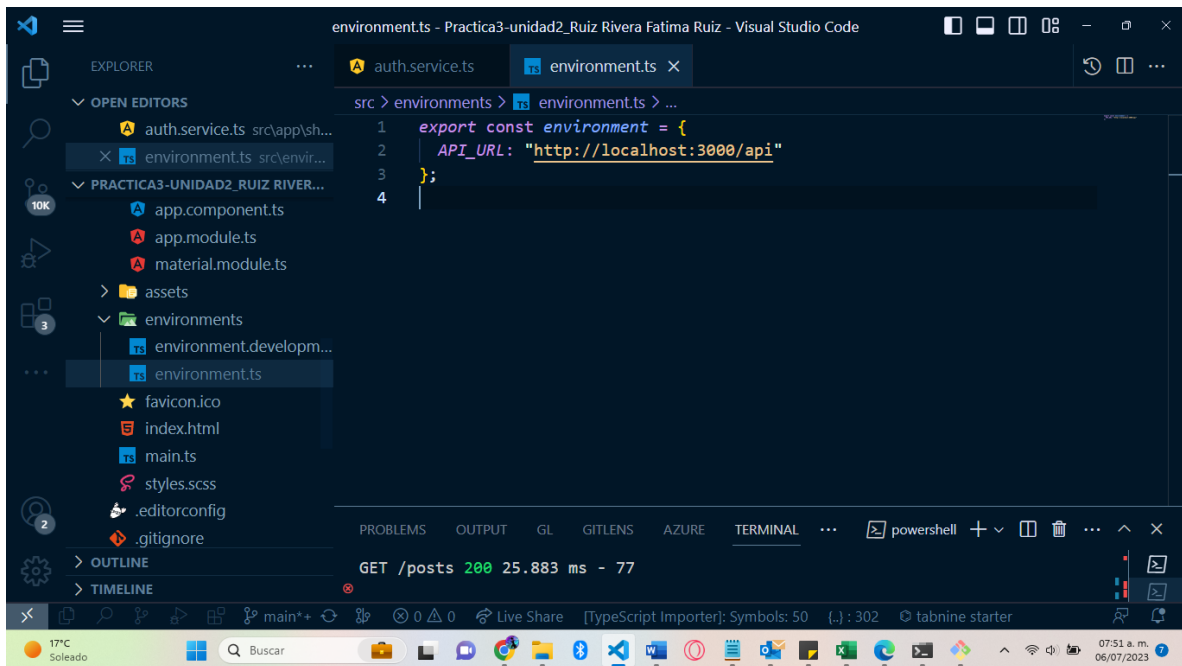
Se genera el servicio de autorización

The screenshot shows the Visual Studio Code interface with the Explorer panel on the left displaying the project structure. The file `auth.service.ts` is selected under the path `src > app > shared > services`. The main editor shows the content of `auth.service.ts`, which includes imports for `HttpClient`, `Injectable`, `Router`, `BehaviorSubject`, `Observable`, `catchError`, `map`, `throwError`, and `MatSnackBar`. The `Terminal` panel at the bottom shows the command `ng g s shared/services/auth` being executed, resulting in the file being created. The status bar at the bottom indicates the file is 302 lines long.

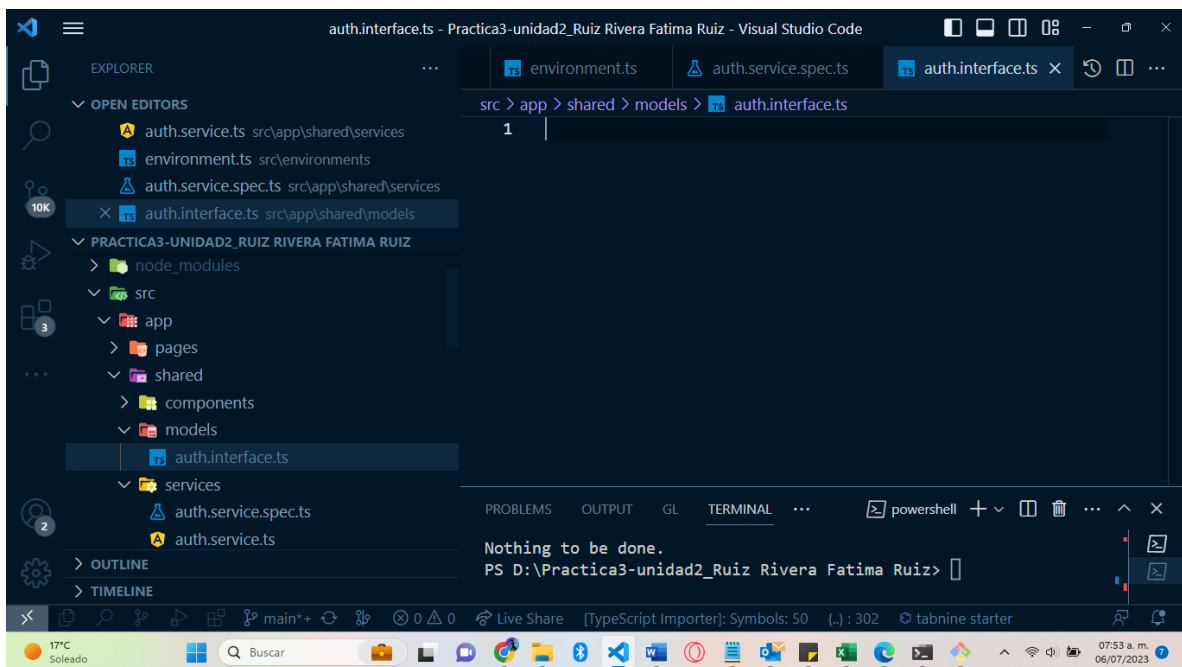
Se agregan las instrucciones correspondientes en el auth.service.ts

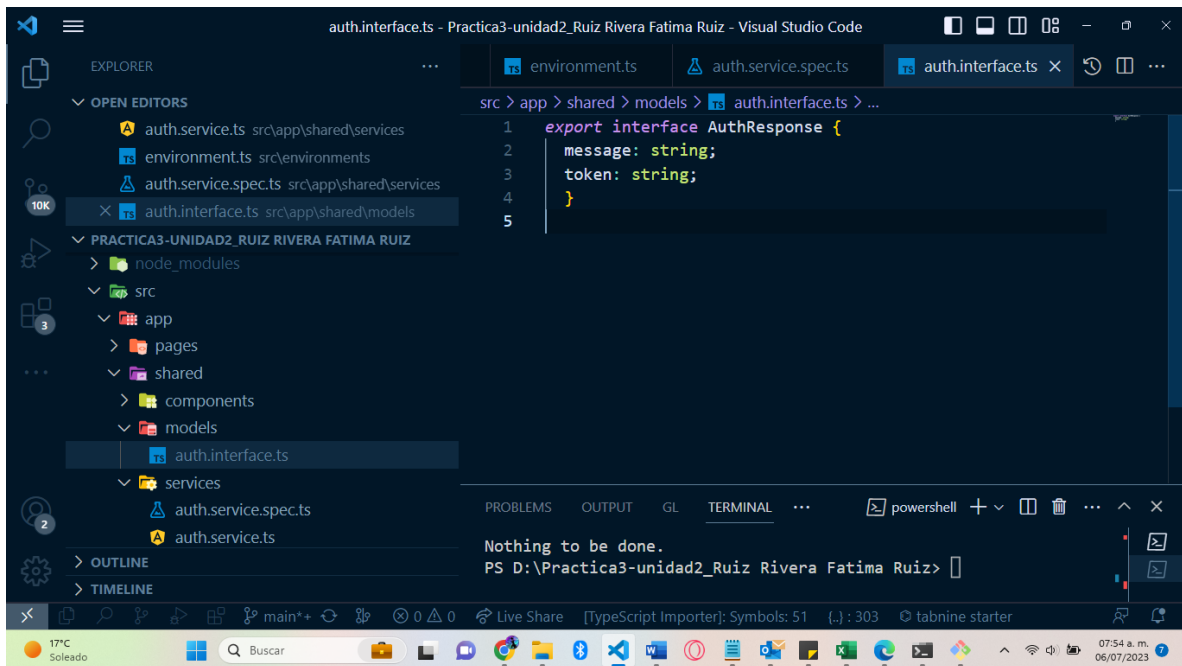
The screenshot shows the Visual Studio Code interface with the `auth.service.ts` file open. The main editor displays the implementation of the `AuthService` class. The class is decorated with `@Injectable` and has a constructor that takes `HttpClient` and `Router` as arguments. It includes two methods: `get tokens(): Observable<string>` and `get tokenData$(): Observable<any>`, both returning `Observable` instances. The `Terminal` panel at the bottom shows the command `GET /posts 200 25.883 ms - 77` being executed. The status bar at the bottom indicates the file is 302 lines long.

Se genera y modifica el environment

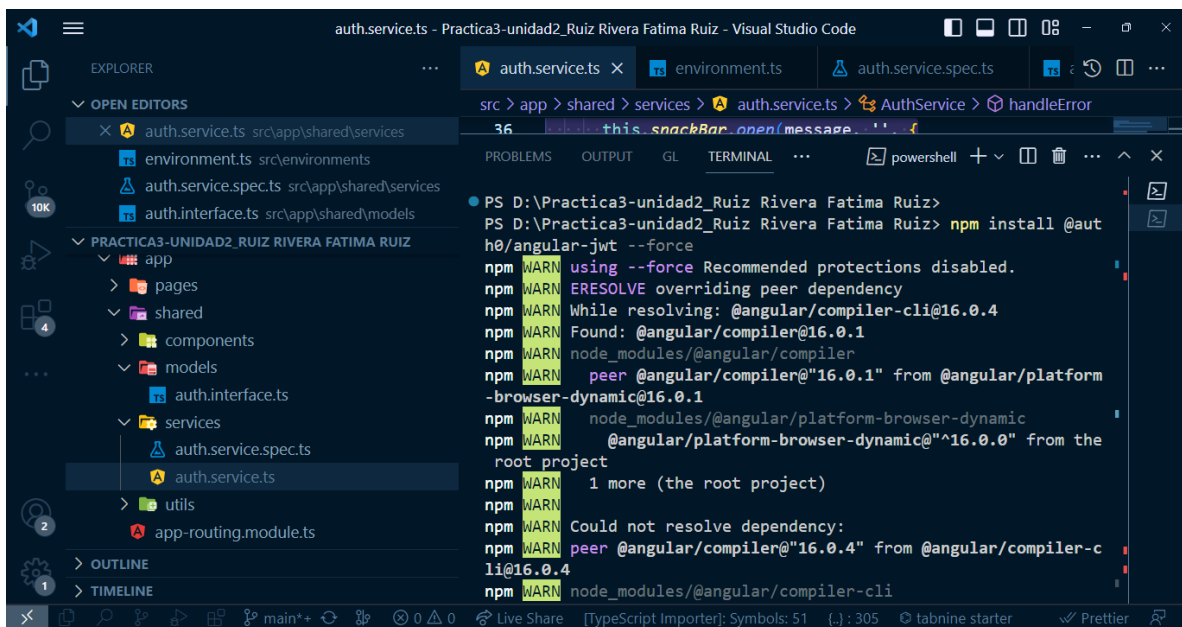


Se crea el archivo **auth.interface.ts**





Se instalan las librerías



Se agregan al cosntruktor

```
header.component.ts - Practica3-unidad2_Ruiz Rivera Fatima Ruiz - Visual Studio Code

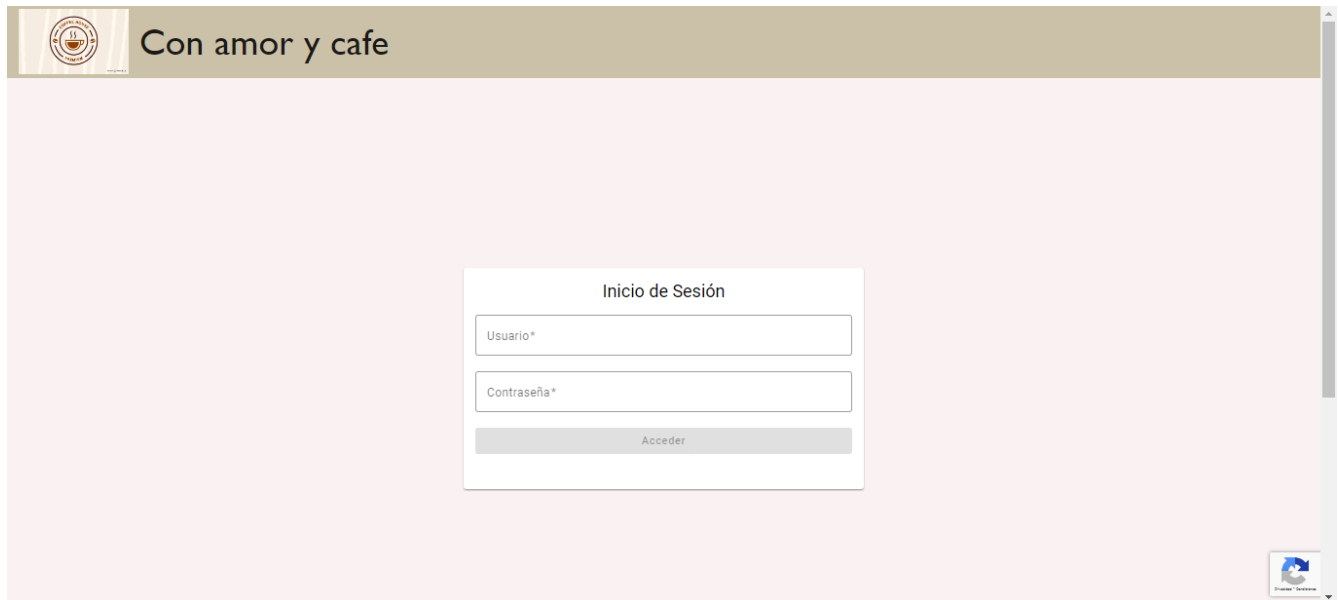
EXPLORER
src > app > shared > components > header > header.component.ts > HeaderComponent
13 constructor(private authService: AuthService) {}
14
15 //Se suscribe para notificacion cuando cambie el token de
16 ngOnInit(): void {
17   this.authService.getTokenData$.subscribe((data: any)
18     | this.data=data;
19   });
20
21 onLogout() {
22   this.authService.logout();
23   this.data = null;
24 }
25
26

PROBLEMS 1 OUTPUT GL TERMINAL ... powershell + - x
npm WARN peer @angular/compiler@"16.0.1" from @angular/platform-browser-dynamic@16.0.1
```

```
environment.ts - Practica3-unidad2_Ruiz Rivera Fatima Ruiz - Visual Studio Code

EXPLORER
src > environments > environment.ts > ...
1 export const environment = {
2   production: false,
3
4   recaptcha:{
5     sitekey: '6LdHkYQmAAAAADPrPhI9H7Sjve6VhiHYbZhtUCb'
6   },
7   API_URL: "http://localhost:4200/api",
8 };
9

PROBLEMS 1 OUTPUT GL TERMINAL ... powershell + - x
npm WARN peer @angular/compiler@"16.0.1" from @angular/platform-browser-dynamic@16.0.1
```



The image shows a web browser window displaying a login page. The browser's address bar shows a URL starting with 'http://'. The page has a light beige header with a circular logo on the left and the text 'Con amor y cafe' on the right. The main content area is a light pink color. In the center, there is a white rectangular box titled 'Inicio de Sesión'. Inside this box, there are two input fields: 'Usuario\*' and 'Contraseña\*'. Below these fields is a grey button labeled 'Acceder'. In the bottom right corner of the browser window, there is a small icon for 'Internet Explorer'.

Con amor y cafe

Inicio de Sesión

Usuario\*

Contraseña\*

Acceder