

Comparación SOAP vs RE: x Guia FrontEnd.docx - Doc: x

docs.google.com/document/d/1hSf7nmZ...  
Brayan Mares Buen... Capucha de tie dye... Tu mapa del cielo G... MoonRise UI : Dark...

Solicitar permiso para editar 1

Arch

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

```
15: constructor(private http: 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:   return this.tokenData.asObservable();
23: }
```

el utilizar environments en tu aplicación es con el objetivo de utilizar API KEY para entornos de pruebas y de producción durante el desarrollo de la aplicación.

En la terminal ejecutar el siguiente comando: `ng g environments`

Modificar los archivos environments de la siguiente forma

```
src > environments > TS environments.ts > ...
1 export const environment = {
2   API_URL: "http://localhost:3000/api"
3 };
4
```

auth.service.ts - classroom - Visual Studio Code

```
1 import { BehaviorSubject, Observable, catchError, map, throwError } from '@angular/common/http';
2 import { HttpClient } from '@angular/common/http';
3 import { Router } from '@angular/router';
4 import { MatSnackBar } from '@angular/material/snack-bar';
5
6 export class AuthService {
7
8   private token = new BehaviorSubject<string>("");
9   private tokenData = new BehaviorSubject<any>({});
10
11   constructor(private http: HttpClient, private router: Router, private snackBar: MatSnackBar) { }
12
13   get token$: Observable<string> {
14     return this.token.asObservable();
15   }
16   get tokenData$: Observable<any> {
17     return this.tokenData.asObservable();
18   }
19
20   // never es cuando no regresa nada
21
22 }
```

PROBLEMS DEBUG CONSOLE TERMINAL

s" to "src/environments/environment.development.ts".  
Skipping creation of already existing environment file "src/environments/environment.development.ts".  
Nothing to be done.  
PS C:\CLASEBARRON\classroom

Comparación SOAP vs RE: x Guia FrontEnd.docx - Doc: x

docs.google.com/document/d/1hSf7nmZ...  
Brayan Mares Buen... Capucha de tie dye... Tu mapa del cielo G... MoonRise UI : Dark...

Solicitar permiso para editar 1

Arch

pruebas y de producción durante el desarrollo de la aplicación.

En la terminal ejecutar el siguiente comando: `ng g environments`

Modificar los archivos environments de la siguiente forma

```
src > environments > TS environments.ts > ...
1 export const environment = {
2   API_URL: "http://localhost:3000/api"
3 };
4
```

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

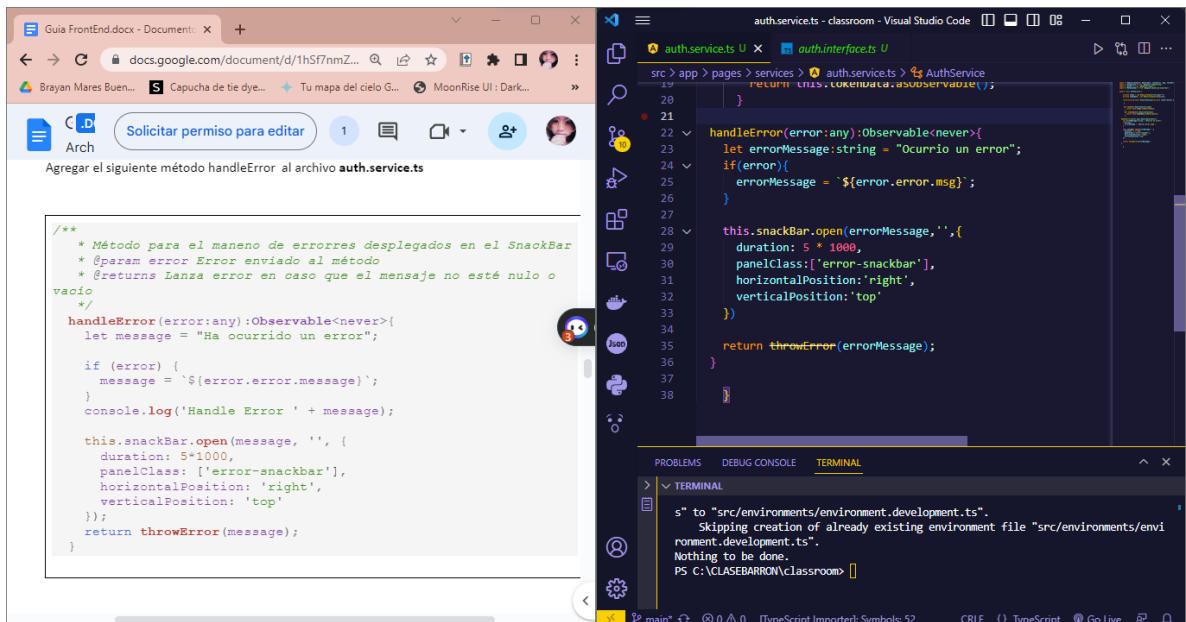
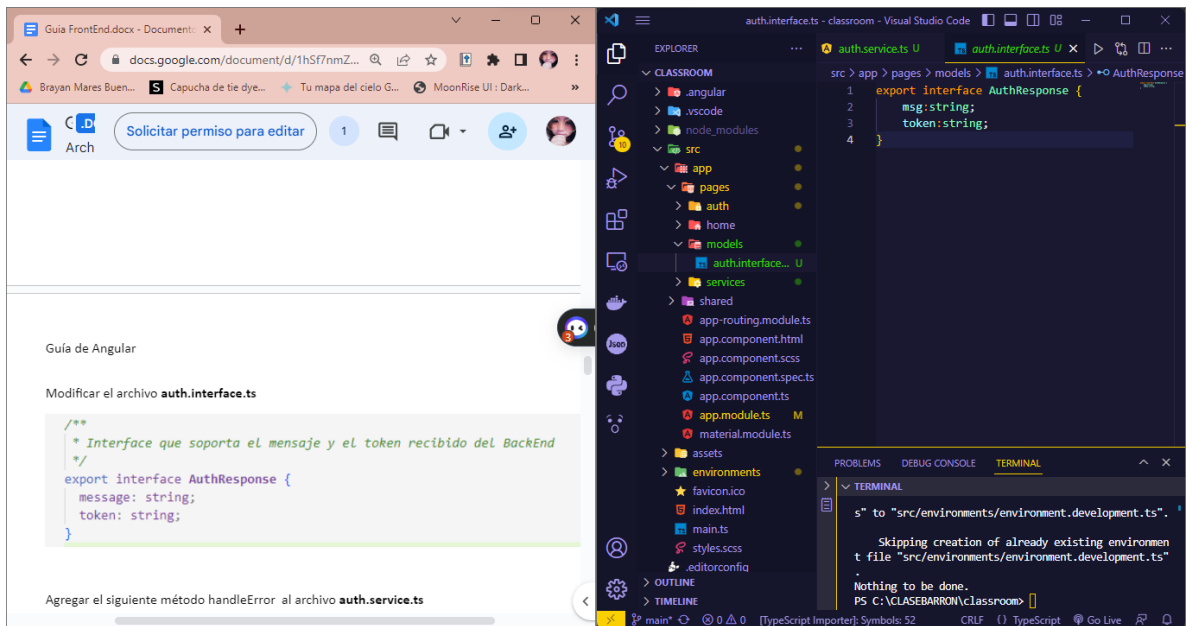
- src
  - app
    - pages
    - auth
    - home
    - models
    - services
  - shared
    - app-routing.module.ts
    - app.component.html
    - app.component.scss
    - app.component.spec.ts
    - app.component.ts
    - app.module.ts
    - material.module.ts
  - assets
    - environments
      - environment.development.ts
      - environment.ts
    - favicon.ico
    - index.html
    - main.ts
    - styles.scss
- authinterface.ts

environments.ts - classroom - Visual Studio Code

```
1 export const environment = {
2
3   API_URL: "http://localhost:3000/api"
4
5 };
6
```

PROBLEMS DEBUG CONSOLE TERMINAL

s" to "src/environments/environment.development.ts".  
Skipping creation of already existing environment file "src/environments/environment.development.ts".  
Nothing to be done.  
PS C:\CLASEBARRON\classroom



Guia FrontEnd.docx - Document: x +

docs.google.com/document/d/1hSf7nmZ...

Brayan Mares Buen... Capucha de tie dye... Tu mapa del cielo G... MoonRise UI : Dark...

Solicitar permiso para editar 1

Arch

Guía de Angular

Guardado de Sesión

Instalar el módulo `@auth0/angular-jwt`

Installation

# installation with npm  
npm install @auth0/angular-jwt

# installation with yarn  
yarn add @auth0/angular-jwt

Importar clase JwtHelperService

import { JwtHelperService } from '@auth0/angular-jwt';

Agregar una variable llamada helper en auth.service.ts

const helper = new JwtHelperService();

auth.service.ts - classroom - Visual Studio Code

auth.service.ts x auth.interface.ts U

src > app > pages > services > auth.service.ts > AuthService

20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38

return this.tokenData.asObservable();

handleError(error:any):Observable<never>{

let errorMessage:string = "Ocurrio un error";

if(error){

errorMessage = \${error.error.msg};

}

this.snackBar.open(errorMessage,'',{

duration: 5 \* 1000,

panelClass:['error-snackbar'],

horizontalPosition:'right',

verticalPosition:'top'

})

return throwError(errorMessage);

}

PROBLEMS DEBUG CONSOLE TERMINAL

TERMINAL

PS C:\CLASEBARRON\classroom> npm install @auth0/angular-jwt

[.....] \ idealTree:Classroom: 511 idealTree buildDeps

Guia FrontEnd.docx - Document: x +

docs.google.com/document/d/1hSf7nmZ...

Brayan Mares Buen... Capucha de tie dye... Tu mapa del cielo G... MoonRise UI : Dark...

Solicitar permiso para editar 1

Arch

# installation with npm  
npm install @auth0/angular-jwt

# installation with yarn  
yarn add @auth0/angular-jwt

Importar clase JwtHelperService

import { JwtHelperService } from '@auth0/angular-jwt';

Agregar una variable llamada helper en auth.service.ts

const helper = new JwtHelperService();

@Injectable({

providedIn: 'root'

})

export class AuthService {

LocalStorage

auth.service.ts - classroom - Visual Studio Code

auth.service.ts x auth.interface.ts U

src > app > pages > services > auth.service.ts > AuthService

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22

import { BehaviorSubject, Observable, catchError, map, throwErr

import { HttpClient } from '@angular/common/http';

import { Router } from '@angular/router';

import { MatSnackBar } from '@angular/material/snack-bar';

import { JwtHelperService } from '@auth0/angular-jwt';

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

const helper = new JwtHelperService();

@Injectable({

providedIn: 'root'

})

export class AuthService {

private token = new BehaviorSubject<string>('');

private tokenData = new BehaviorSubject<any>({});

constructor(private http:HttpClient,private router:Router,;

}

PROBLEMS DEBUG CONSOLE TERMINAL

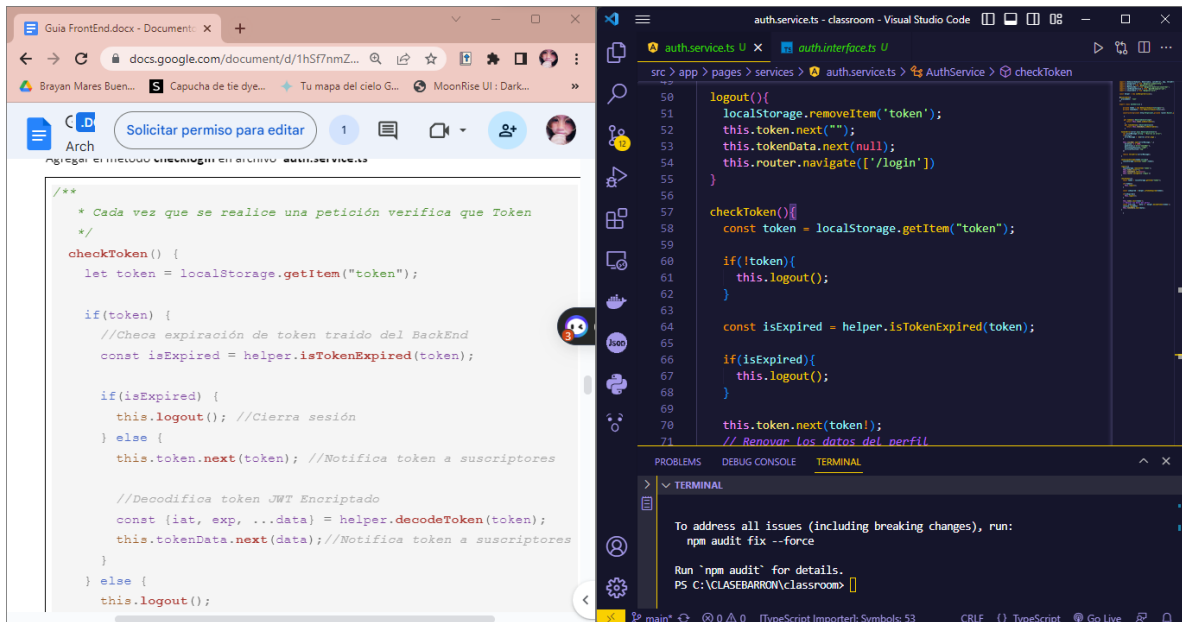
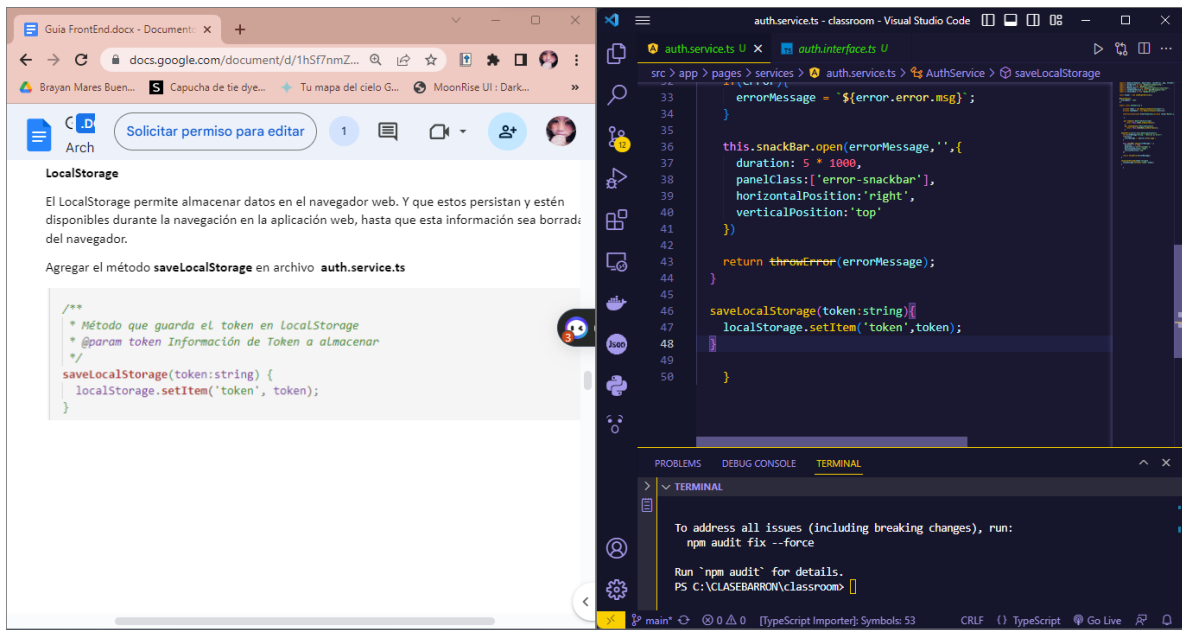
TERMINAL

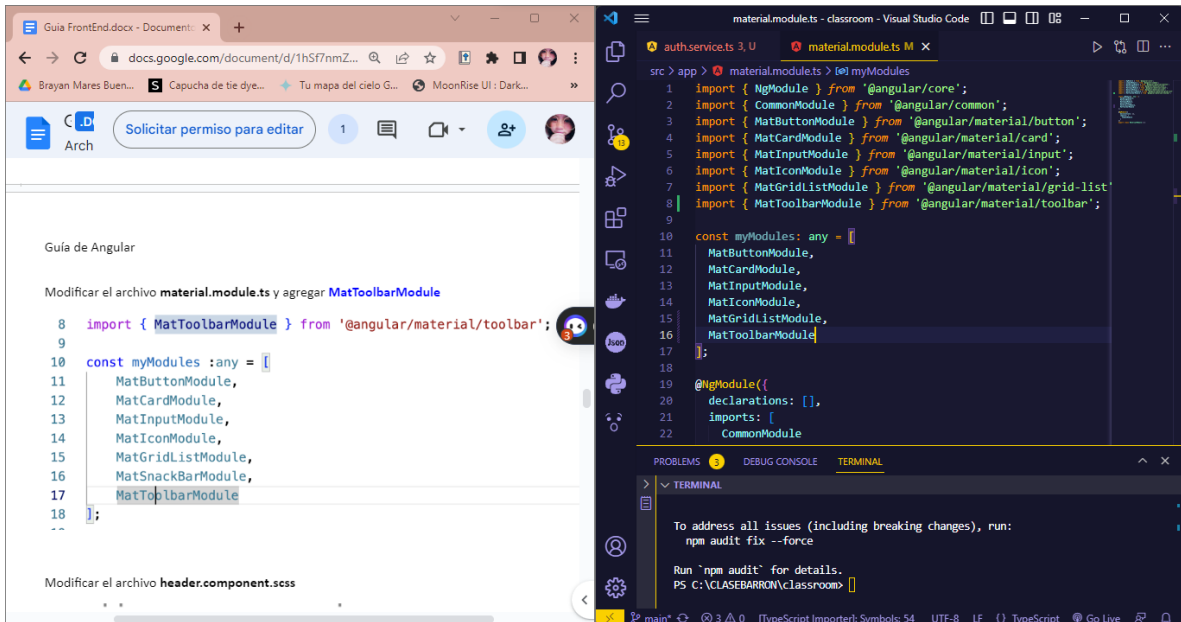
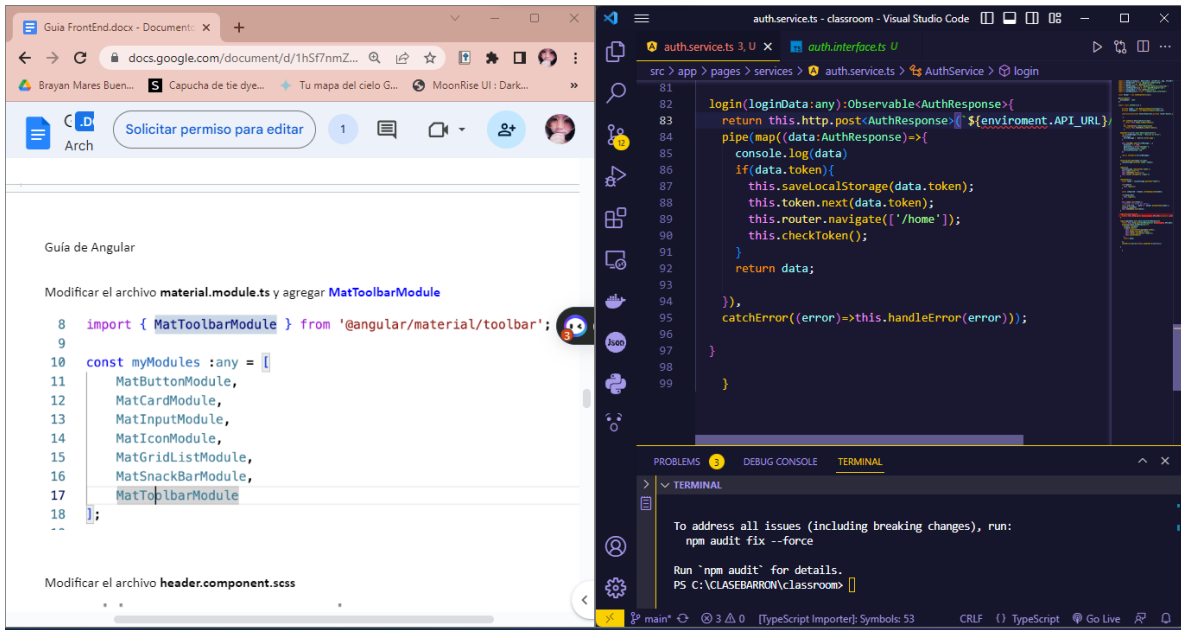
To address all issues (including breaking changes), run:

npm audit fix --force

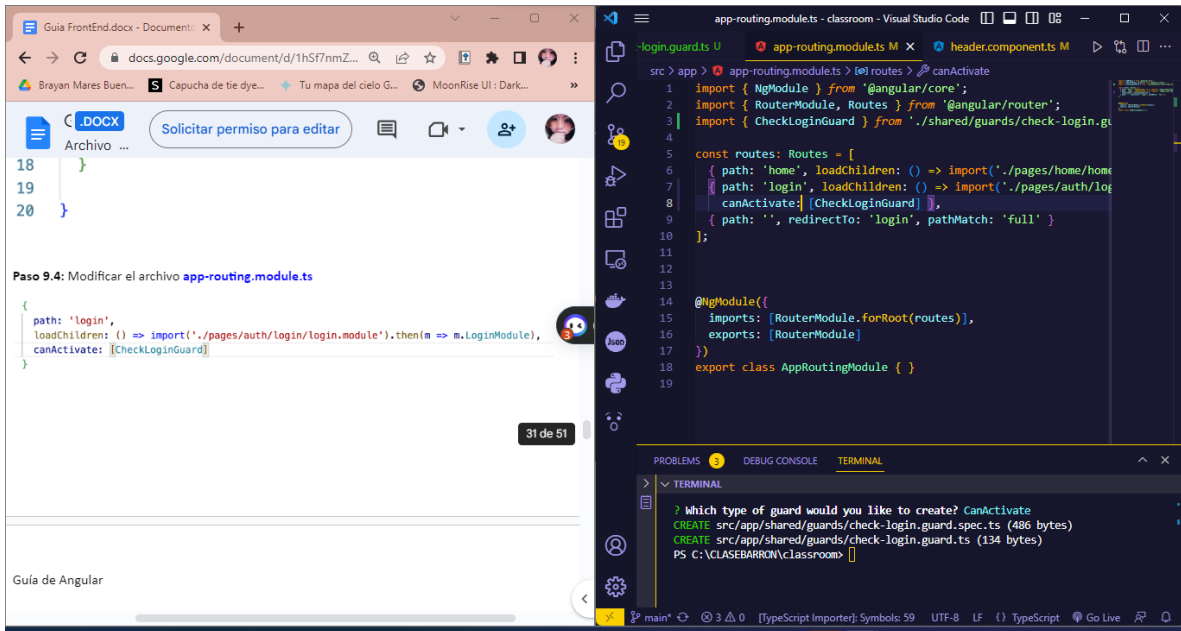
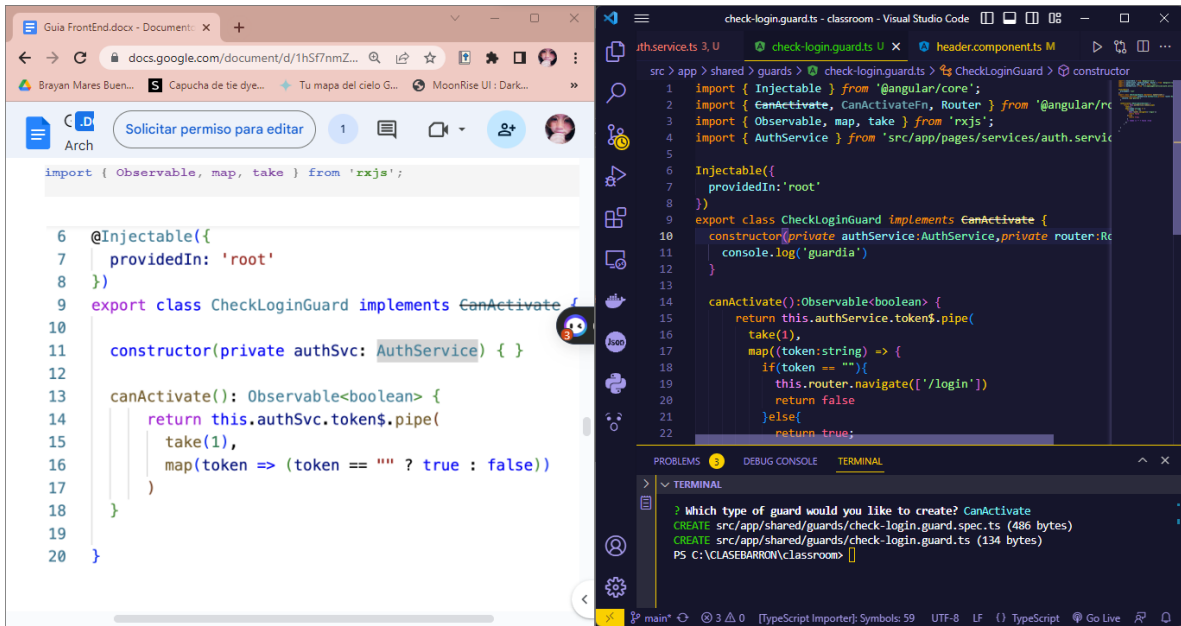
Run 'npm audit' for details.

PS C:\CLASEBARRON\classroom>









login.component.ts - classroom - Visual Studio Code

```
File Edit Selection View Go Run Terminal Help
src > app > pages > auth > login > login.component.ts > LoginComponent > onLogin
31
32 onSubmit() {
33   if (this.loginForm.invalid) {
34     return;
35   }
36   this.router.navigate(['/home']);
37 }
38
39
40 onLogin(){
41   this.recaptchaV3Service.execute('importantAction').subscribe((token:string)=>{
42     this.tokenVisible = true;
43     this.recaptchaToken = `Token [${token}] generated`
44     console.log(this.recaptchaToken)
45   })
46   this.authService.login(this.loginForm.value).subscribe({
47     next:(data:any)=>{
48       console.log(data)
49     },
50     error:(data:any)=>{
51       console.log('error')
52       console.log(data)
53     }
54   })
55 }
```

PROBLEMS 0 DEBUG CONSOLE TERMINAL

TERMINAL

```
> Which type of guard would you like to create? CanActivate
CREATE src/app/shared/guards/check-login.guard.spec.ts (486 bytes)
CREATE src/app/shared/guards/check-login.guard.ts (134 bytes)
PS C:\CLASEBARRON\classroom> []
```

Ln 46, Col 40 Spaces: 2 UTF-8 LF ( ) TypeScript Go Live

Users.ts - classroom - Visual Studio Code

```
File Edit Selection View Go Run Terminal Help
EXPLORER thservices.ts U app.modules.ts M environment.development.ts M login.component.html M environments.ts M Users.ts U x
CLASSROOM src > app > pages > models > Users.ts > User
1 export class User{
2   name!:string;
3   lastname!:string;
4   username!:string;
5   password!:string;
6   role!:string;
7 }
```

PROBLEMS 0 DEBUG CONSOLE TERMINAL

TERMINAL

Initial Chunk Files	Names	Raw Size
main.js	main	30.88 kB
runtime.js	runtime	12.62 kB

5 unchanged chunks

Build at: 2023-07-08T05:59:27.990Z - Hash: e4f5939c6e32db20 - Time: 396ms

✓ Compiled successfully.

Ln 7, Col 2 Spaces: 4 UTF-8 CRLF ( ) TypeScript Go Live



```
<!-- INICIO DE SESIÓN -->
<form [formGroup]="loginForm" (ngSubmit)="onLogin()"
  <div fxLayout="column" fxLayoutAlign="space-around center">
    <mat-form-field class="full-width" appearance="outline">
      <mat-label>Usuario</mat-label>
      <input matInput type="text" mat-form-field="username">
      <mat-error *ngIf="baseForm.isValidField(loginForm.get('username'))">
        {{ baseForm.getErrorMessage(loginForm.get('username'))}}
      </mat-error>
    </mat-form-field>
    <mat-form-field class="full-width" appearance="outline">
      <mat-label>Contraseña</mat-label>
      <input matInput type="password" mat-form-field="password">
      <mat-error *ngIf="baseForm.isValidField(loginForm.get('password'))">
        {{baseForm.getErrorMessage(loginForm.get('password'))}}
      </mat-error>
    </mat-form-field>
  </div>
  <div class="button-row mb-24 text-center">
    <button type="submit" mat-raised-button color="primary" [disabled]="loginForm.invalid">Iniciar
      Sesión</button>
  </div>
</form>
<div class="button-row mb-24 text-center">
  <button (click)="goRegister()" mat-raised-button color="basic">Registrar</button>
</div>
```

