**Registro con servicios de Google**

**Objetivo**: Implementar el registro de usuarios utilizando los servicios de Google en tu aplicación Angular.

**Recursos necesarios:**

* Una cuenta de desarrollador en Google: Debes crear un proyecto en la Consola de Desarrolladores de Google y obtener las credenciales necesarias para utilizar los servicios de Google en tu aplicación.
* Angular CLI: Asegúrate de tener Angular CLI instalado en tu entorno de desarrollo.

**Pasos:**

**1. Configurar el proyecto en la Consola de Desarrolladores de Google:**

* Crea un proyecto en la Consola de Desarrolladores de Google y habilita los servicios necesarios, como Google Sign-In.
* Obtiene las credenciales de cliente para la autenticación con Google. Puedes obtener el ID de cliente y la clave de cliente.

Para más información accede a <https://developers.google.com/identity/protocols/oauth2>

**2. Instalar el paquete angularx-social-login:**

* En la terminal, ejecuta el siguiente comando para instalar el paquete angularx-social-login:

ng add angularx-social-login

**3. Configurar el módulo de autenticación con Google:**

* En el archivo app.module.ts, importa SocialLoginModule y GoogleLoginProvider de angularx-social-login y configura la inicialización del proveedor de autenticación de Google utilizando las credenciales obtenidas en el paso 1.
* Asegúrate de proporcionar el ID de cliente y la clave de cliente en la configuración del proveedor.

En el campo providers agrega el siguiente objeto:

providers: [

{

provide: 'SocialAuthServiceConfig',

useValue: {

autoLogin: true, //keeps the user signed in

providers: [

{

id: GoogleLoginProvider.PROVIDER\_ID,

provider: new GoogleLoginProvider(MyClientId),//your client id

},

],

},

}

]

//MyClientId se obtuvo en el punto 1.

**4. Implementar el botón de registro con Google:**

Dentro del componente RegisterComponent realizar lo siguiente:

* Declara la variable “google”
* Crea una nuevo atributo googleButton: any;
* En el ngOnInit coloca el siguiente código:

ngOnInit(): void {

//google button setup

google.accounts.id.initialize({

client\_id: environment.oAuthClientId,

callback: (response: any) => {

//let payload = Utilities.parseJwt(response.credential);

this.regUserAuthGoogle(response.credential);

},

});

}

* Crea un método para crear un botón oculto de google

crearBotonOcultoGoogle() {

//Crea un nuevo elemento HTML en el documento actual que se está visualizando en el navegador.

const googleLogin: HTMLElement = document.createElement('div');

// Ocultamos el el elmento nuevo

googleLogin.style.display = 'none';

// Agregamos el nuevo elemento,se inserta como un elemento secundario del elemento "body" de la página

document.body.appendChild(googleLogin);

// Pasamos las propiedades a renderButton, el cual creara un boton de google

google?.accounts?.id?.renderButton(googleLogin, {

type: 'icon',

prompt: 'select\_account',

});

//Se adentra dentro de las propidades html del googleLogin y seleciona el div con rol de boton

const googleLoginRoleButton: any =

googleLogin.querySelector('div[role=button]');

//Retornamos una funcion llamada clic () esta ejecuta un evento propio de div[role=button]

return {

click: () => {

googleLoginRoleButton?.click();

},

};

}

* En le hook ngAfterViewInit hacer el llamado al método anteriormente creado:

ngAfterViewInit(): void {

this.googleButton = this.crearBotonOcultoGoogle();

}

* Crear el siguiente método:

iniciarGoogleLogin() {

this.googleButton.click();

}

* Crear un botón en el template con sus estilos personalizados:

<button type="button" class="my-button" (click)="iniciarGoogleLogin()"><img class="icon" src="assets/…"><span class="text ">Iniciar con Google</span></button>

**5. Manejar la respuesta de autenticación:**

En el callback de éxito, obtén la información del usuario autenticado y realiza las acciones necesarias, como el registro en tu backend.

//google button setup

google.accounts.id.initialize({

client\_id: environment.oAuthClientId,

callback: (response: any) => {

//let payload = Utilities.parseJwt(response.credential);

this.regUserAuthGoogle(response.credential); //acciones necesarias para el registro del usuario

},

});

regUserAuthGoogle(){

//acciones necesarias para el registro

}