**Resumen**

Es una aplicación para gestionar una red social entre cooperativas utilizando Ionic/Angular y Firebase.

**Requisitos Previos**

Instalación de Node.js , Angular CLI , Firebase CLI

**Instalación de dependencias**

**Dependencias Angular:**

* **@angular/animations**
* **Descripción: Proporciona soporte para animaciones en aplicaciones Angular.**
* **@angular/common**
* **Descripción: Contiene funcionalidades comunes para todas las aplicaciones Angular.**
* **@angular/compiler**
* **Descripción: Compila los componentes y plantillas de Angular en código JavaScript.**
* **@angular/core**
* **Descripción: Contiene las clases esenciales para construir aplicaciones Angular.**
* **@angular/fire**
* **Descripción: Integración de Angular con Firebase.**
* **@angular/forms**
* **Descripción: Proporciona soporte para construir y validar formularios en Angular.**
* **@angular/platform-browser**
* **Descripción: Contiene las implementaciones comunes para el navegador.**
* **@angular/platform-browser-dynamic**
* **Descripción: Proporciona el código necesario para iniciar la aplicación Angular en un navegador.**
* **@angular/router**
* **Descripción: Implementa la navegación y la administración de rutas para aplicaciones Angular.**
* **rxjs**
* **Descripción: Biblioteca para programación reactiva en JavaScript.**
* **Tslib**
* **Descripción: Biblioteca de utilidades para TypeScript.**
* **zone.js**
* **Descripción: Biblioteca que proporciona zona para Angular.**

**Dependencias Ionic:**

* **@ionic/angular**
* **Descripción: Framework de IU basado en componentes para la construcción de aplicaciones móviles y web con Angular.**
* **@ionic/pwa-elements**
* **Descripción: Proporciona elementos web que se pueden utilizar en aplicaciones progresivas de Ionic.**
* **Ionicons**
* **Descripción: Conjunto de iconos de IU para aplicaciones web y móviles.**

**Dependencias Firebase:**

* **Firebase**
* **Descripción: Plataforma de desarrollo de aplicaciones móviles y web que proporciona servicios en la nube, incluido el backend para aplicaciones.**

**Estructura del Proyecto**

**app**: Aquí se encuentran los componentes, módulos y servicios principales.

**components**: Contiene componentes reutilizables, como custom-input, header, nuevaPublicacion y terminosycondiciones.

* **custom-input**: Componente para entradas personalizadas.
* **header**: Componente para el encabezado de la aplicación.
* **nuevaPublicacion**: Componente para crear nuevas publicaciones.
* **terminosycondiciones**: Componente que muestra los términos y condiciones.

**guards**:

* **auth.guard.ts**: Asegura que solo los usuarios autenticados puedan acceder a ciertas rutas.
* **no-auth.guard.ts**: Asegura que los usuarios no autenticados puedan acceder a ciertas rutas.

**models**:

* Contiene modelos como galeria.model.ts, grupo.model.ts, publicacion.model.ts y user.model.ts que definen las estructuras de datos.

**pages**:

Contiene subcarpetas como auth y main, que agrupan componentes específicos de la página.

**auth**:

* forgot-password: Página para recuperar contraseñas olvidadas.
* login: Página de inicio de sesión.
* register: Página de registro.

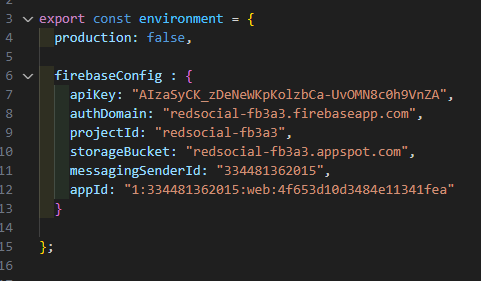
**main**:

* about: Página acerca del proyecto o empresa.
* cooperatives: Página con información sobre cooperativas.
* home: Página principal.
* lista-usuarios: Lista o directorio de usuarios.
* portada: Portada o landing page.
* profile: Perfil del usuario.
* publicar: Herramienta o interfaz para publicar contenido.

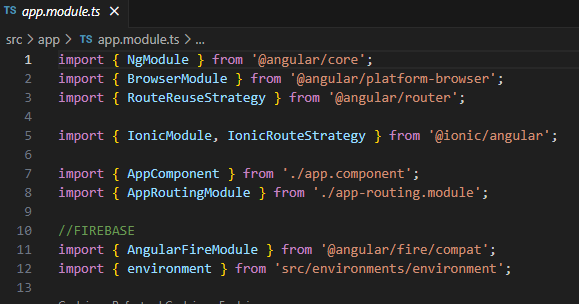
**Configuración Firebase**

Instalamos las dependencias con: npm install firebase

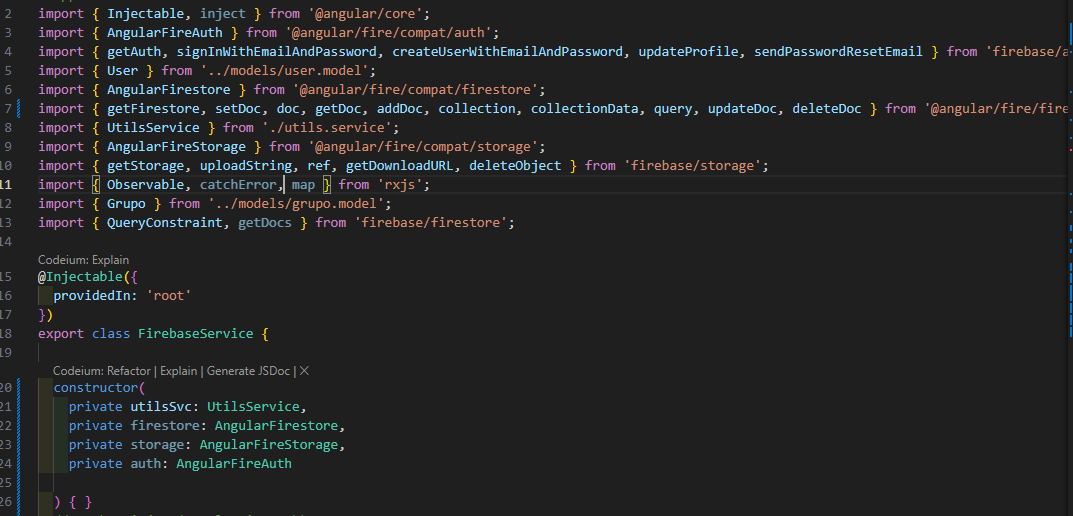
Añadimos la configuración de Firebase en envirotment:



Añadimos las importaciones necesarias:

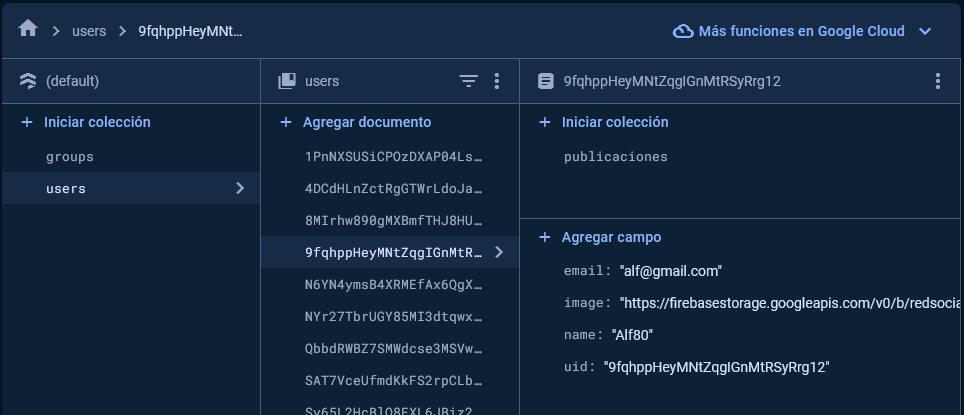


Creamos un Servicio que se encargara de toda la interacción con Firebase:



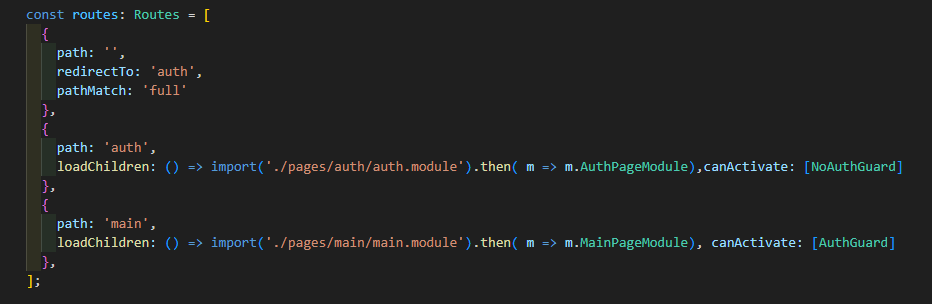
**Estructura de Datos**

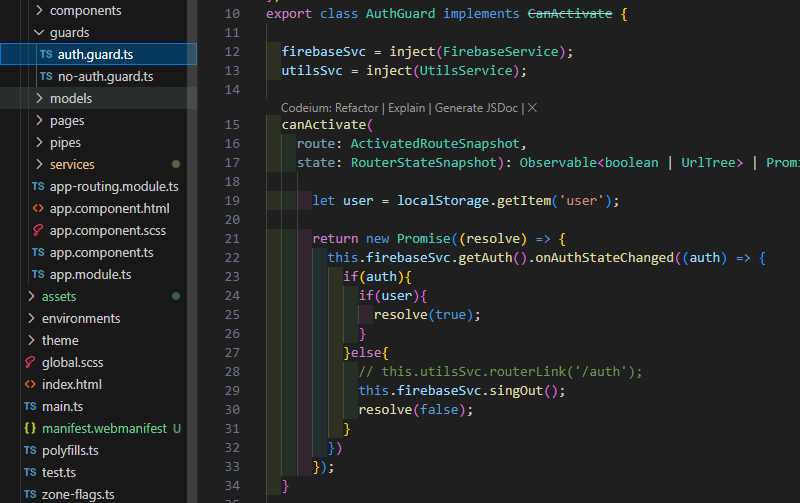
La bbdd consta de 3 colecciones, users, groups y publicaciones.



**Inicio de Sesión y Registro**

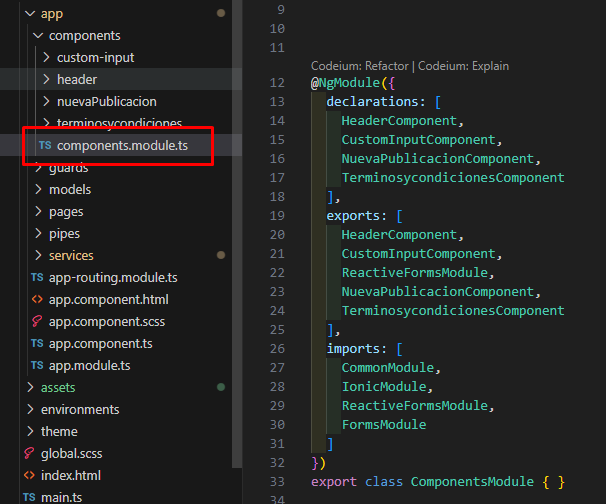
* Para las paginas de Registro (Login, Register y forgot-password) estan dentro de la carpeta Auth para poder evitar que un usuario sin registrar acceda al resto de paginas gracias al uso de guards.



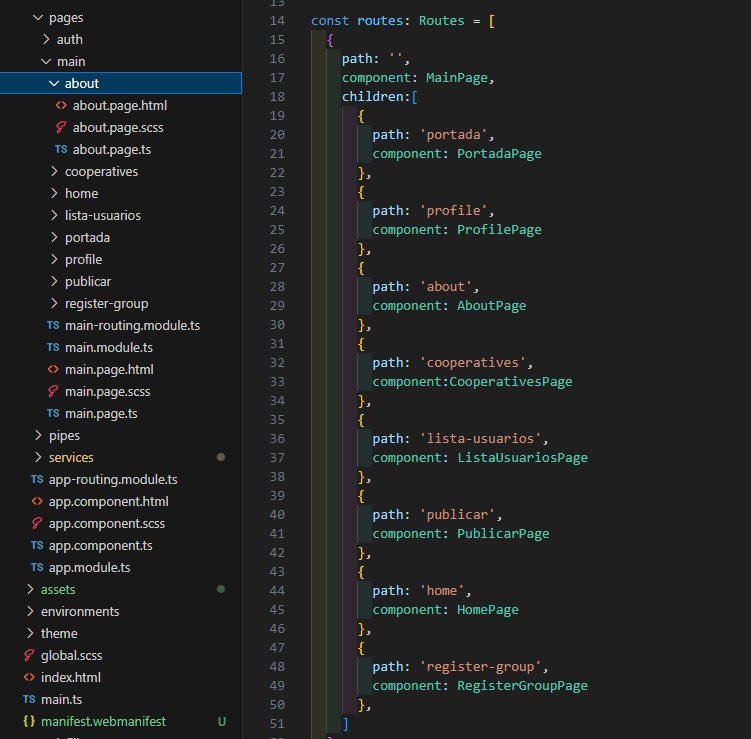


**Paginas y Componentes:**

* Para los componentes he generado un único modulo que se encarga de exportarlos todos.

****

* En cuanto a las paginas he modificado la estructura para tratarlos también como componentes, eliminando los modulos y routings.

****

**Creación y Gestión de Usuarios**

Desde firebaseSvc.ts manejo la interacción de los usuarios con la BBDD:



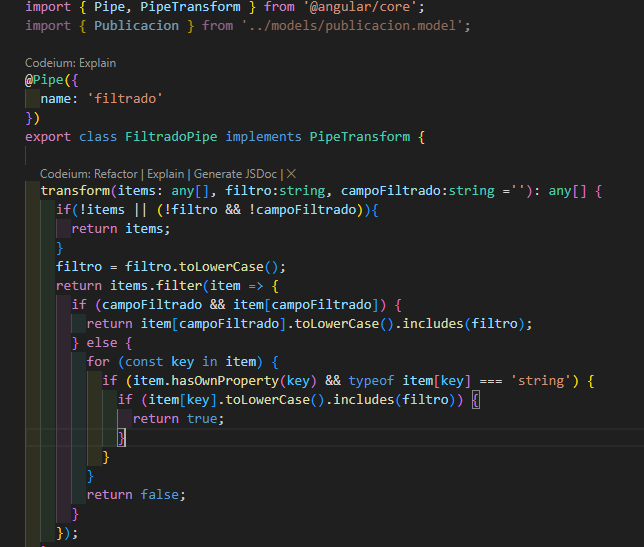
Y desde utilsSvs.service.ts el almacenamiento local y uso de la cámara:



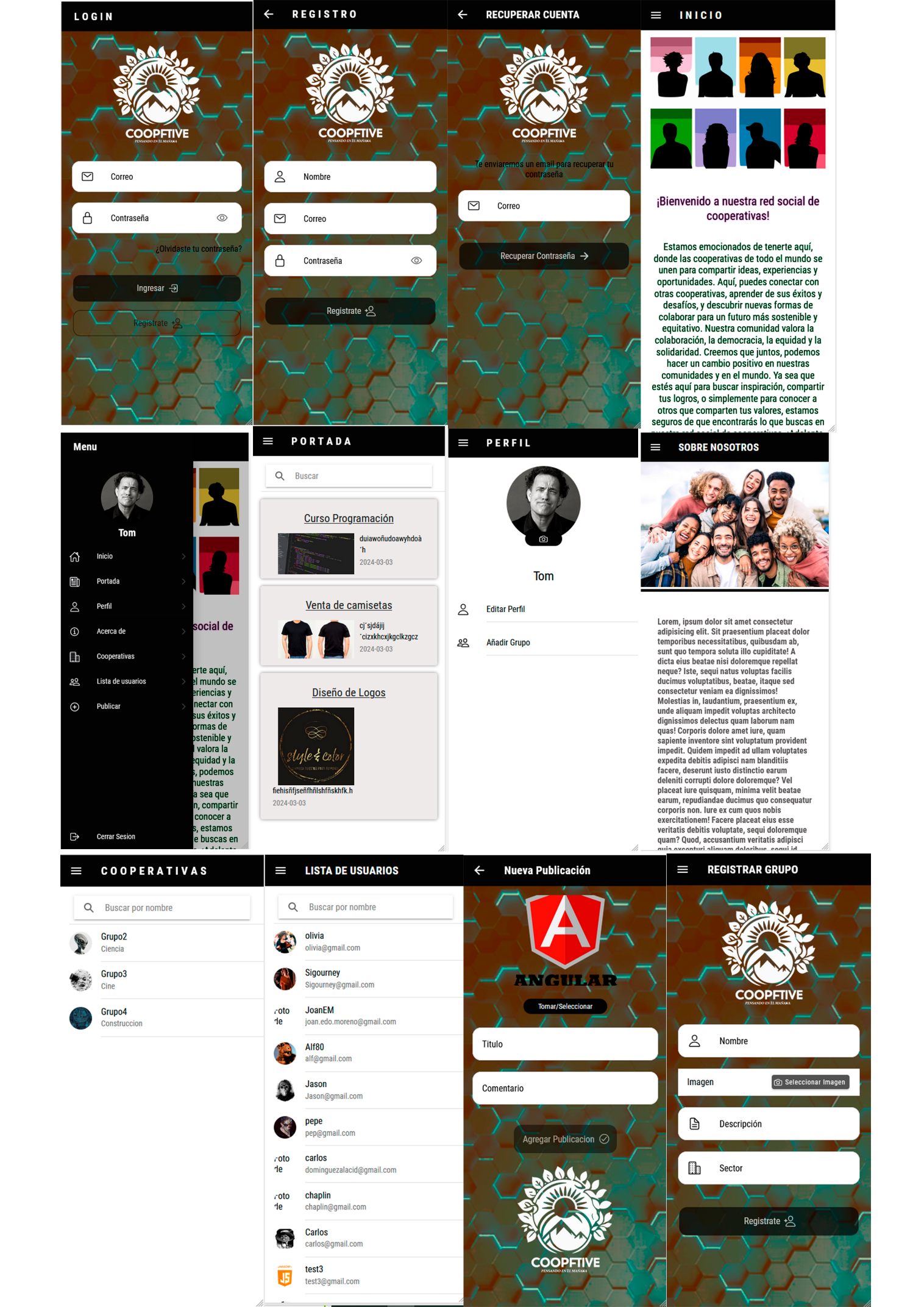
**Creación y Gestión de Grupos y Publicaciones**

* Genero un modelo en donde defino las propiedades de cada uno, (al igual que para usuarios).
* Los nuevos grupos se muestran en formato lista en la página de Cooperativas.
* Las publicaciones pueden verse desde la pagina de portada, se añaden en orden inverso para que siempre salga primero la ultima publicación.

**Uso de pipe para filtrar:**



Utilizo el filtrado tanto para los Usuarios, grupos como publicaciones.



.