

MANUAL TECNICO

DISEÑO DESARROLLO E IMPLEMENTACION DE APLICACIÓN
PARA MOVILES YOUAPP

VICTOR ALEJANDRO ALEJO GALVEZ

DENYS ENRIQUE CRUZ INESTROZA

MATTHEW EMILIO GAITAN RAMOS

MARCO ANTONIO HERNANDEZ HERNANDEZ

ALVARO ANIBAL VELASQUEZ RIVAS

UNIVERSIDAD DON BOSCO

FACULTAD DE INGENIERIA

INGENIERIA DE SISTEMAS

2021

Indice

Presentación	3
Objetivo	4
Procesos	5
Procesos de entrada.....	5
Procesos de salida	5
Herramientas utilizadas para el desarrollo	6
Android Studio.....	6
IntelliJ IDEA.....	6
JDK	6
MySQL	6
NPM.....	6
React native.....	6
Visual Studio Code	6
Docker.....	7
MinIO.....	7
Estructura	8
Estructura del proyecto backend	8
Módulo de componentes	9
Docker-images	10
Services-support.....	10
Services-instances	10
AuthenticationApplication.java	11
Estructura de proyecto frontend.....	12
src.....	12
Enums	13
Environment.....	13
Models.....	14
Modules.....	14
Assets	15
Styles.....	15
Index.js.....	16

Presentación

El siguiente manual guiara a los usuarios que harán soporte a la aplicación YouApp, el cual les dará a conocer los requerimientos y la estructura para la construcción de dicha aplicación, en el desarrollo de la aplicación móvil conectada mediante un backend desarrollado en Java, a una base de datos, el cual muestra las herramientas necesarias para la construcción y la funcionalidad del sistema.

Objetivo

Informar y especificar al usuario la estructura y conformación del sistema con el fin de que puedan hacer soporte y modificaciones o actualizaciones al sistema en general.

Procesos

Procesos de entrada

Ingresar a la aplicación

Ingresar datos de logeo

Ingresar datos para registro de usuarios

Ingresar datos para recuperación de contraseña

Procesos de salida

Correo para verificación de cuenta

Correo para recuperación de contraseña

Mostrar pantalla home

Herramientas utilizadas para el desarrollo

Android Studio

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de apps para Android y está basado en IntelliJ IDEA.

IntelliJ IDEA

IntelliJ IDEA es un entorno de desarrollo integrado (IDE) para el desarrollo de programas informáticos. Es desarrollado por JetBrains (anteriormente conocido como IntelliJ), y está disponible en dos ediciones: edición para la comunidad y edición comercial.

JDK

Java Development Kit (JDK) es un software para los desarrolladores de Java. Incluye el intérprete Java, clases Java y herramientas de desarrollo Java (JDT): compilador, depurador, desensamblador, visor de applets, generador de archivos de apéndice y generador de documentación.

MySQL

MySQL es el sistema de gestión de bases de datos relacional más extendido en la actualidad al estar basada en código abierto.

NPM

De sus siglas **NPM (Node Package Manager)** es un gestor de paquetes desarrollado en su totalidad bajo el lenguaje JavaScript por Isaac Schlueter, a través del cual podemos obtener cualquier librería con tan solo una sencilla línea de código.

React native

React Native es un **framework JavaScript** para crear aplicaciones reales nativas para iOS y Android, basado en la librería de JavaScript React para la creación de componentes visuales, cambiando el propósito de los mismos para, en lugar de ser ejecutados en navegador, **correr directamente sobre las plataformas móviles nativas**, en este caso iOS y Android.

Visual Studio Code

Es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.

Docker

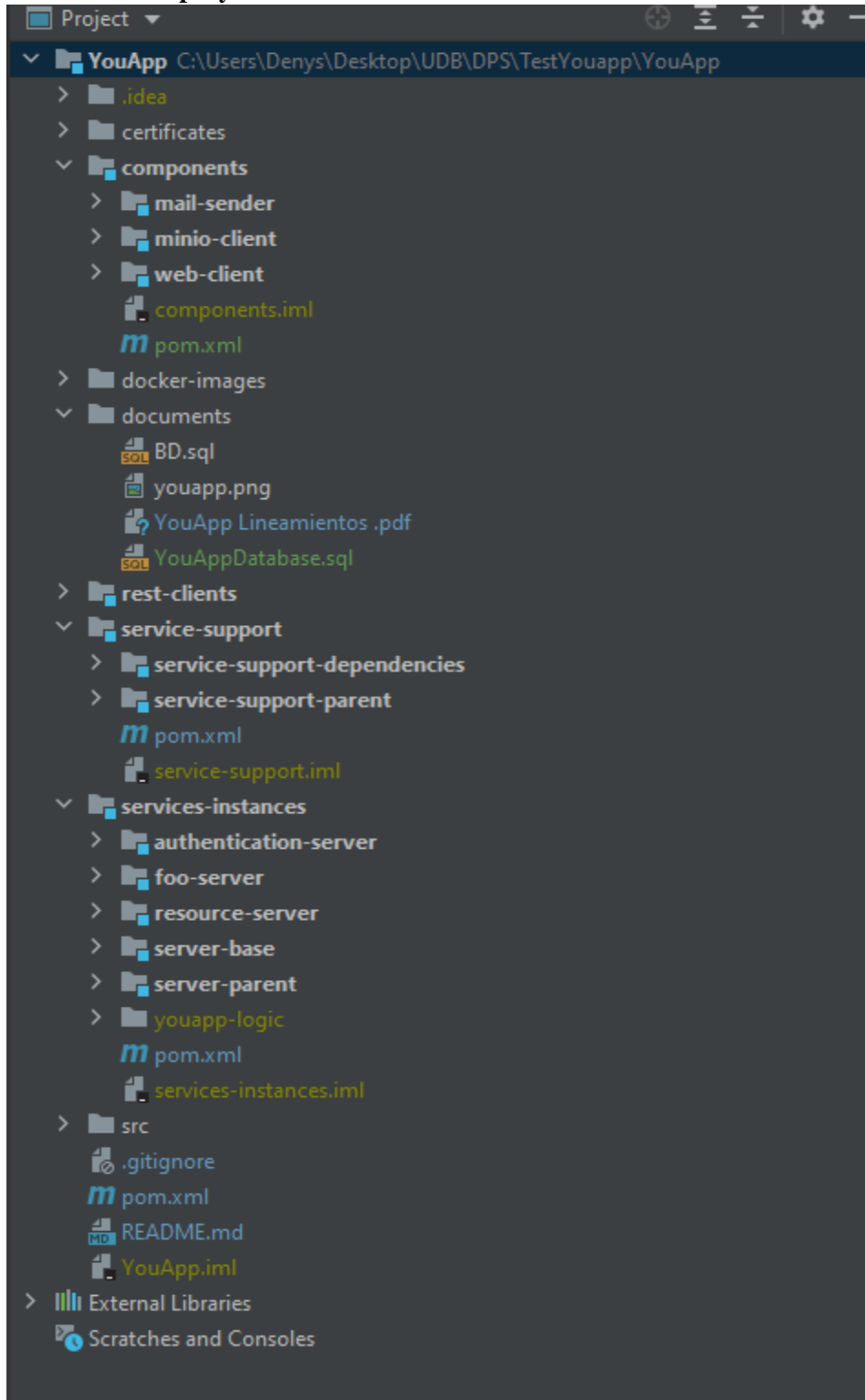
TI, es una tecnología de creación de contenedores que permite la creación y el uso de contenedores de Linux

MinIO

Es un servidor de almacenamiento en la nube compatible con Amazon S3, liberado bajo Licencia Apache v2

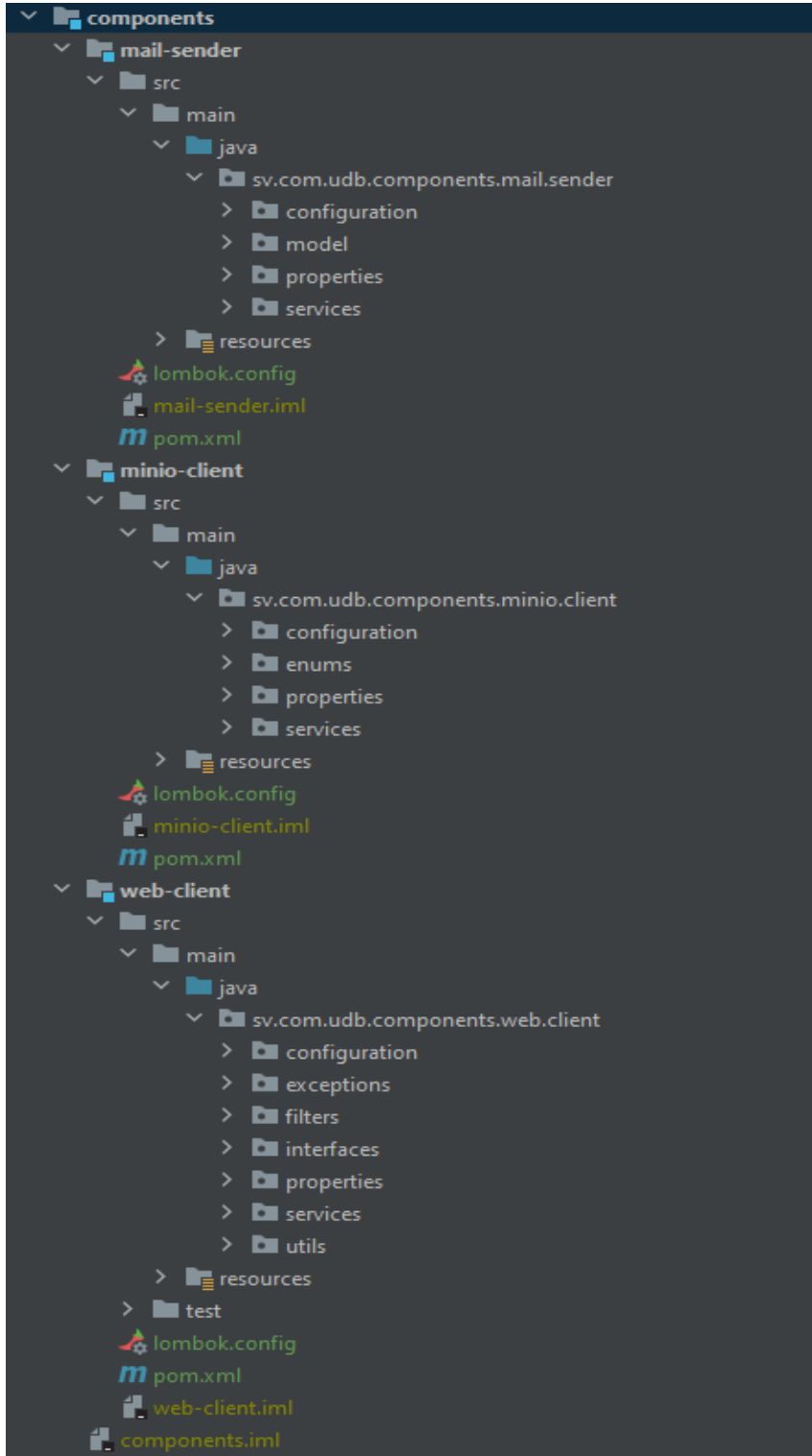
Estructura

Estructura del proyecto backend



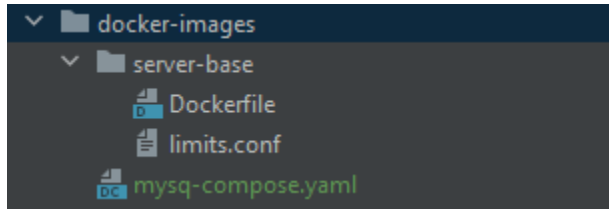
Módulo de componentes

Este módulo contiene los submódulos de mail-sender que es el encargado del envío de correos, minio-client el cual se comunica con nuestro servidor de archivos (minio) y web-client que es el que realiza peticiones externas.



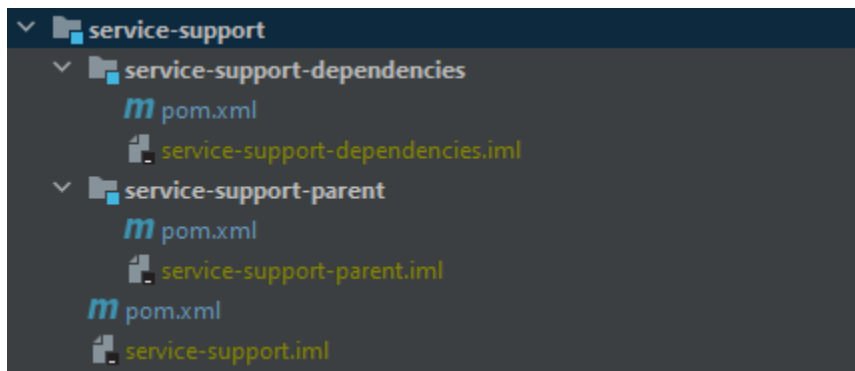
Docker-images

Contiene la configuración necesaria para poder crear contenedores Docker de cualquier modulo gracias a la ayuda del plugin yib



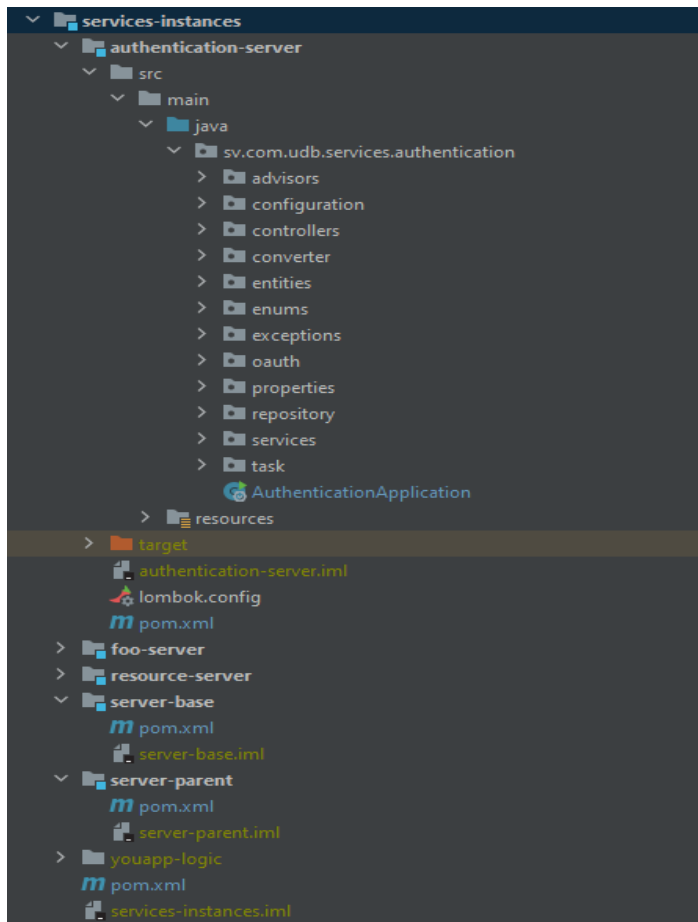
Services-support

Aquí almacenamos los módulos service-support-dependencies y service-support-parent los que en pocas palabras podemos decir que contienen nuestros bom



Services-instances

Podemos definir este modulo como el contenedor de nuestros módulos finales, los cuales son: authentication-server, el cual se encarga del logeo usando OAuth2; foo-server y resource-server, los cuales contienen configuraciones de la seguridad; server-base y server-parent, que son los que contienen los pom.xml bases para los módulos funcionales tales como authentication-server y youapp-logic; youapp-logic, aquí se encuentra la solución a los requisitos funcionales de nuestra aplicación.



AuthenticationApplication.java

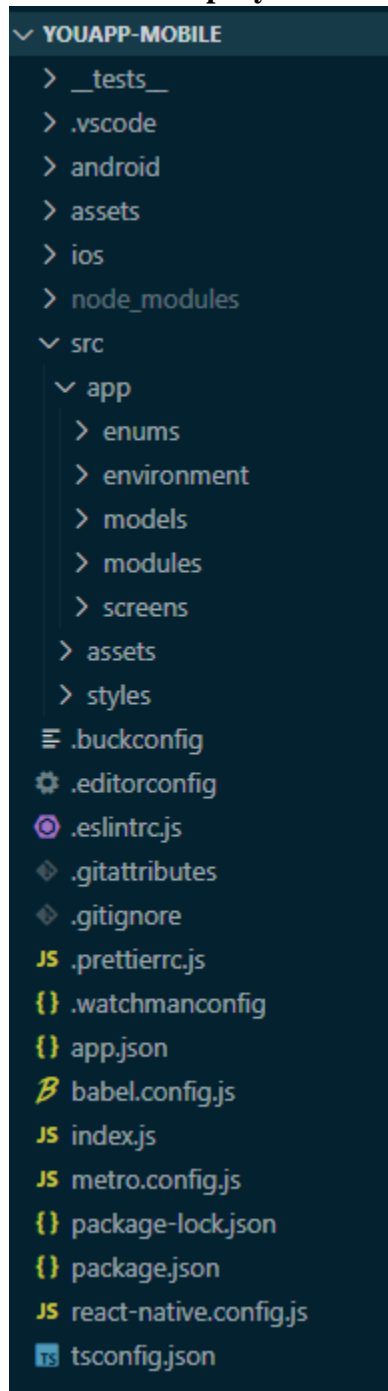
Es la clase que contiene el main para poder correr nuestro modulo authentication-server

```
package sv.com.udb.services.authentication;

import ...

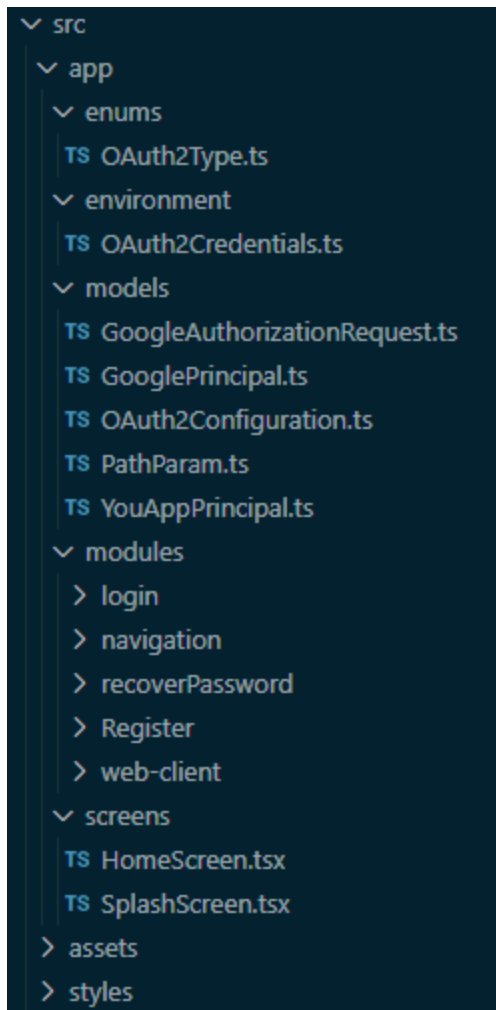
@SpringBootApplication(scanBasePackages = "sv.com")
public class AuthenticationApplication {
    public static void main(String[] args) {
        new SpringApplicationBuilder(AuthenticationApplication.class)
            .registerShutdownHook(true).run(args);
    }
}
```

Estructura de proyecto frontend



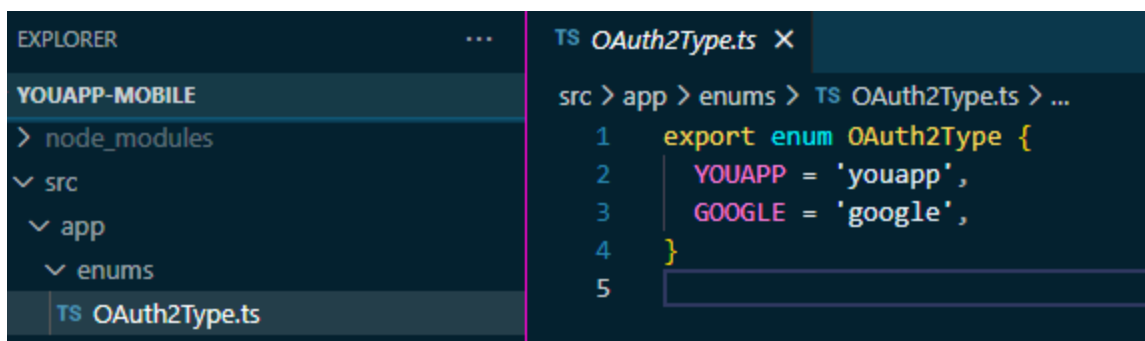
src

Contiene los archivos fuentes para el funcionamiento de nuestra aplicación



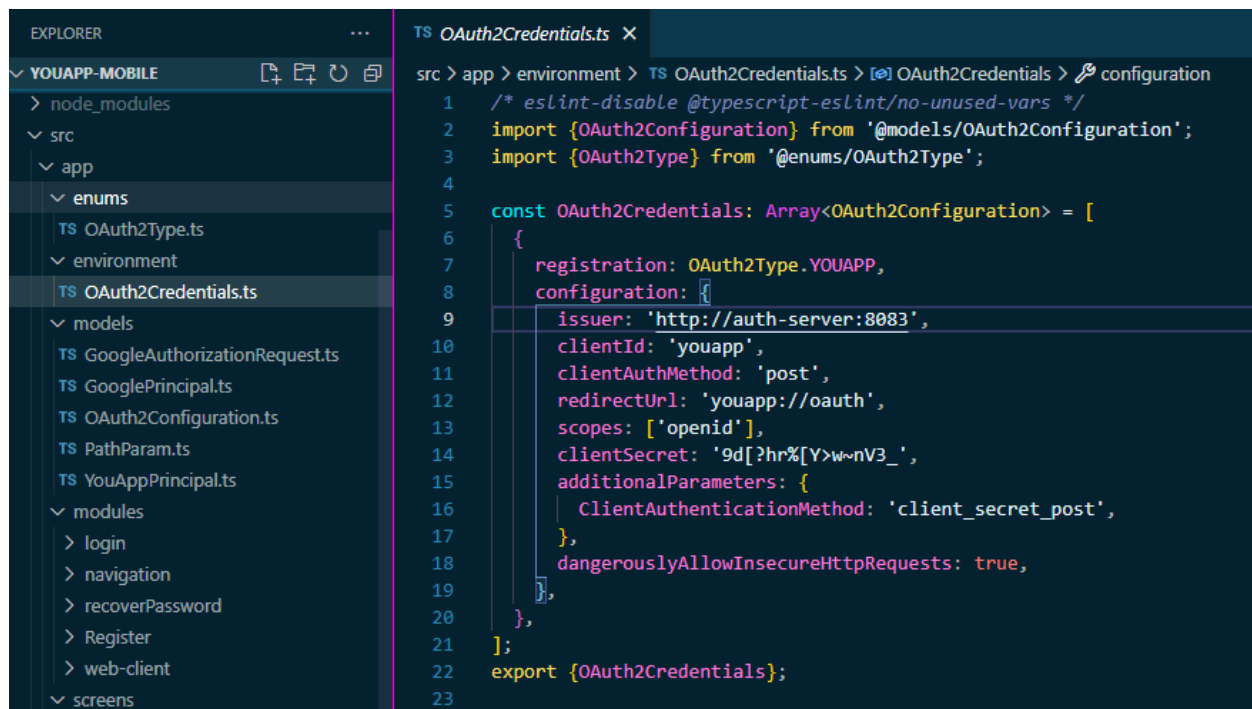
Enums

Contiene un archivo de utilidad tipo enum con las opciones de autenticación que posee YouApp



Environment

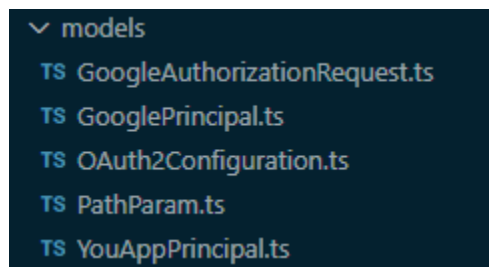
Como su nombre lo indica aquí tenemos las variables de nuestro entorno, en este caso la constante que expone la configuración necesaria para la seguridad con OAuth2.



```
src > app > environment > TS OAuth2Credentials.ts > OAuth2Credentials > configuration
1  /* eslint-disable @typescript-eslint/no-unused-vars */
2  import {OAuth2Configuration} from '@models/OAuth2Configuration';
3  import {OAuth2Type} from '@enums/OAuth2Type';
4
5  const OAuth2Credentials: Array<OAuth2Configuration> = [
6    {
7      registration: OAuth2Type.YOUAPP,
8      configuration: {
9        issuer: 'http://auth-server:8083',
10       clientId: 'youapp',
11       clientAuthMethod: 'post',
12       redirectUrl: 'youapp://oauth',
13       scopes: ['openid'],
14       clientSecret: '9d[?hr%[Y>w~nV3_',
15       additionalParameters: {
16         ClientAuthenticationMethod: 'client_secret_post',
17       },
18       dangerouslyAllowInsecureHttpRequests: true,
19     },
20   ],
21 ];
22 export {OAuth2Credentials};
23
```

Models

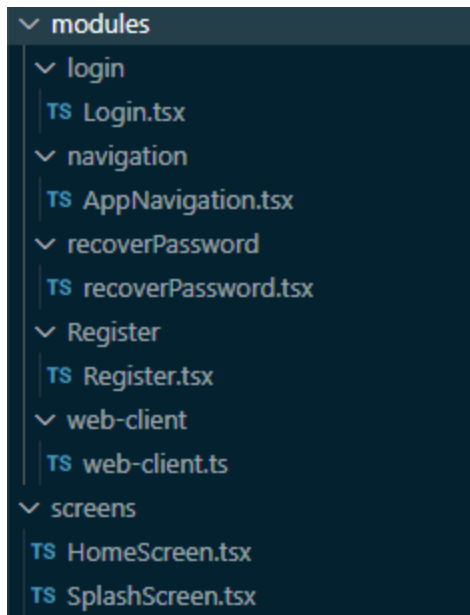
Contiene los entidades a usar para el desarrollo de la aplicación



```
models
  TS GoogleAuthorizationRequest.ts
  TS GooglePrincipal.ts
  TS OAuth2Configuration.ts
  TS PathParam.ts
  TS YouAppPrincipal.ts
```

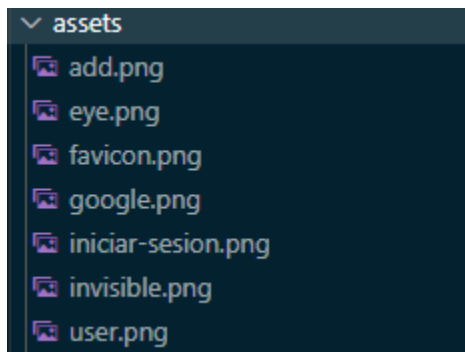
Modules

En esta carpeta tenemos los módulos que componen las pantallas de nuestra aplicación y también el módulo web-client el cual expone una instancia a WebClient para realizar peticiones



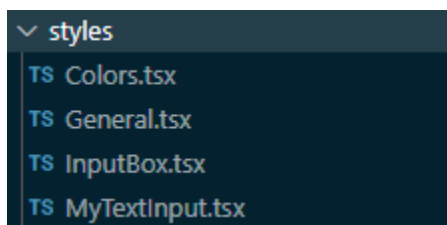
Assets

En esta carpeta se almacenan los archivos para estilizar la aplicación, tales como fuentes, imágenes, iconos, entre otros.



Styles

Aquí tenemos nuestros estilos y componentes estilizados para poder reutilizarlos.



Index.js

Esta es la clase principal para correr el proyecto react, la cual solo llama al componente que controla la navegación (El que gestiona las pantallas) de la aplicación.

```
JS index.js  X
JS index.js > ...
1  import AppNavigation from '@modules/navigation/AppNavigation';
2  import {AppRegistry} from 'react-native';
3  import {name as appName} from './app.json';
4
5  AppRegistry.registerComponent(appName, () => AppNavigation);
6
```