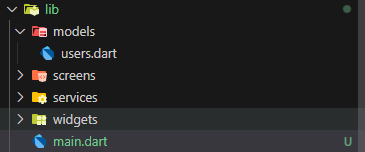
**Desarrollo del Proyecto:**

**Parte I:**

**Creación de los primeros folders con los códigos dados**

****

**Primera parte (así se ve cuando corro el programa)**

**A black background with white text

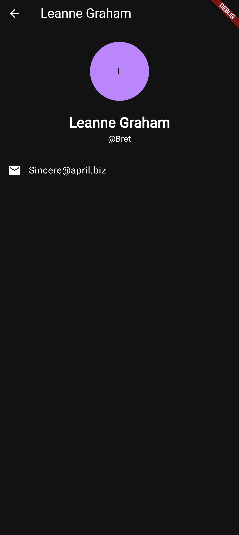
Description automatically generated**

**Parte II y III**

Hubo un cambio en en el código

El termino accentColor por colorScheme.secondary debido a que flutter por su nueva actualización utiliza ese nuevo termino. También se coloque el icono de me gusta y el de compartir (Con un snackbar que hace la simulación de la acción). Por otro lado, colocaron animaciones sencillas y el acceso a los usuarios.

 A screenshot of a computer

Description automatically generated 

(El diseño aplicado)

 Navigator.push(

                context,

                PageRouteBuilder(

                  pageBuilder: (context, animation, secondaryAnimation) =>

                      UsersScreen(),

                  transitionsBuilder:

                      (context, animation, secondaryAnimation, child) {

                    const begin = Offset(1.0, 0.0);

                    const end = Offset.zero;

                    const curve = Curves.easeInOut;

                    var tween = Tween(begin: begin, end: end).chain(

                      CurveTween(curve: curve),

                    );

                    var offsetAnimation = animation.drive(tween);

Se ve de esta manera el cambio del accenColor:

colorScheme: ColorScheme.dark(

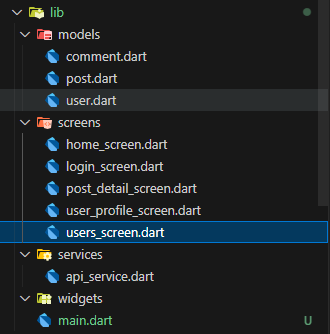
          secondary: Colors.blueAccent,

        ),

**Parte IV:**

En esta parte añadimos el logic\_screen.dart, editamos el home\_screen.dart para añadir el cerrar sección y en el main.dart mantuve mi código anterior debido a que le añadí animaciones entre pantallas. Sin embargo, le hice los cambios que pedía para añadir la conexión con el logic\_screen…

A white background with purple text

Description automatically generated

**Parte V:**

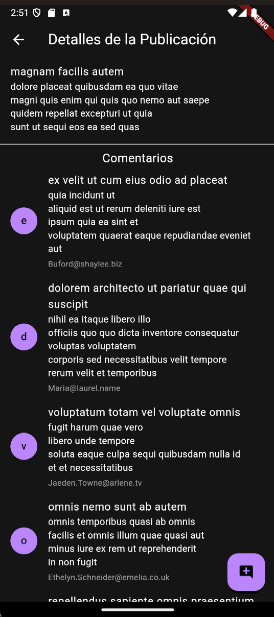
En esta parte seguimos las instrucciones, también edite los colores para una mejor y estuvimos modificando todos los códigos pedidos. En esta parte continue llevando a cabo las instrucciones y logre ejecutar muchas de ellas, añadí el add\_comment.dart y el código. También modificamos el user\_profile\_screen.dart y actualizamos el post\_detail\_screen.dart… Finalmente modifique el main.dart para que lograse ver las publicaciones en primera pantalla para que se vea con mayor realidad y semejanza a una red social verdadera.

A white background with black text

Description automatically generatedA screenshot of a black and white message

Description automatically generated

(Aclaración el color claro u oscuro se aplica a todas solo lo cambie a oscuro por el fondo del Word)

 **Detalles y comentarios**  A screenshot of a phone

Description automatically generated **añadir comentario**

**CAMBIOS DE MAIN.DART:**

import 'package:flutter/material.dart';

import 'package:shared\_preferences/shared\_preferences.dart';

import '../screens/users\_screen.dart';

import 'screens/home\_screen.dart';

import 'screens/login\_screen.dart';

void main() {

  runApp(MyApp());

}

class MyApp extends StatefulWidget {

  // Este widget es la raíz de la aplicación.

  @override

  \_MyAppState createState() => \_MyAppState();

}

class \_MyAppState extends State<MyApp> {

  ThemeMode \_themeMode = ThemeMode.system;

  bool \_isLoggedIn = false;

  @override

  void initState() {

    super.initState();

    \_loadThemePreference();

    \_checkLoginStatus();

  }

  // Cargar la preferencia de tema almacenada

  void \_loadThemePreference() async {

    SharedPreferences prefs = await SharedPreferences.getInstance();

    int? themeIndex = prefs.getInt('themeMode');

    setState(() {

      \_themeMode = ThemeMode.values[themeIndex ?? 0];

    });

  }

  // Verificar si el usuario ya inició sesión

  void \_checkLoginStatus() async {

    SharedPreferences prefs = await SharedPreferences.getInstance();

    bool isLoggedIn = prefs.containsKey('userId');

    setState(() {

      \_isLoggedIn = isLoggedIn;

    });

  }

  // Guardar la preferencia de tema

  void \_saveThemePreference(ThemeMode mode) async {

    SharedPreferences prefs = await SharedPreferences.getInstance();

    await prefs.setInt('themeMode', mode.index);

  }

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      title: 'Red Social Básica',

      theme: ThemeData(

        brightness: Brightness.light,

        primarySwatch: Colors.blue,

        colorScheme: ColorScheme.light(

          secondary: const Color.fromARGB(255, 120, 1, 131),

        ),

        appBarTheme: AppBarTheme(

          color: const Color.fromARGB(229, 104, 17, 17),

          iconTheme:

              IconThemeData(color: const Color.fromARGB(255, 193, 202, 209)),

          titleTextStyle: TextStyle(

              color: const Color.fromARGB(227, 206, 215, 221), fontSize: 20),

        ),

      ),

      darkTheme: ThemeData(

        brightness: Brightness.dark,

        primarySwatch: Colors.blue,

        colorScheme: ColorScheme.dark(

          secondary: const Color.fromARGB(255, 89, 2, 120),

        ),

      ),

      themeMode: \_themeMode, // Modo de tema seleccionado

      home: \_isLoggedIn

          ? HomeScreen(

              onThemeChanged: \_toggleThemeMode,

            )

          : LoginScreen(

              onThemeChanged: \_toggleThemeMode,

            ),

    );

  }

  void \_toggleThemeMode(ThemeMode mode) {

    setState(() {

      \_themeMode = mode;

    });

    \_saveThemePreference(mode);

  }

}

class HomeScreen extends StatelessWidget {

  final Function(ThemeMode) onThemeChanged;

  HomeScreen({required this.onThemeChanged});

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: Text('Red Social Básica'),

        actions: [

          IconButton(

            icon: Icon(Icons.people),

            onPressed: () {

              // Añadir una animación de transición al navegar entre pantallas

              Navigator.push(

                context,

                PageRouteBuilder(

                  pageBuilder: (context, animation, secondaryAnimation) =>

                      UsersScreen(),

                  transitionsBuilder:

                      (context, animation, secondaryAnimation, child) {

                    const begin = Offset(1.0, 0.0);

                    const end = Offset.zero;

                    const curve = Curves.easeInOut;

                    var tween = Tween(begin: begin, end: end).chain(

                      CurveTween(curve: curve),

                    );

                    var offsetAnimation = animation.drive(tween);

                    return SlideTransition(

                        position: offsetAnimation, child: child);

                  },

                ),

              );

            },

          ),

          IconButton(

            icon: Icon(Icons.thumb\_up\_alt\_outlined),

            onPressed: () {

              // Feedback táctil al usuario utilizando InkWell

              ScaffoldMessenger.of(context).showSnackBar(

                SnackBar(content: Text('Funcionalidad de "👍" activada')),

              );

            },

          ),

          IconButton(

            icon: Icon(Icons.share),

            onPressed: () {

              // Feedback táctil al usuario utilizando InkWell

              ScaffoldMessenger.of(context).showSnackBar(

                SnackBar(content: Text('Funcionalidad de "↪" activada')),

              );

            },

          ),

          PopupMenuButton<ThemeMode>(

            onSelected: onThemeChanged,

            itemBuilder: (context) => [

              PopupMenuItem(

                value: ThemeMode.system,

                child: Text('Predeterminado'),

              ),

              PopupMenuItem(

                value: ThemeMode.light,

                child: Text('Modo Claro'),

              ),

              PopupMenuItem(

                value: ThemeMode.dark,

                child: Text('Modo Oscuro'),

              ),

            ],

          ),

        ],

      ),

      body: Center(

        child: InkWell(

          onTap: () {

            ScaffoldMessenger.of(context).showSnackBar(

              SnackBar(

                  content: Text('Bienvenido a la Red Social Básica FAKEBOOK')),

            );

          },

          child: Padding(

            padding: const EdgeInsets.all(16.0),

            child: Text(

              'Bienvenido a la Red Social Básica',

              style: TextStyle(fontSize: 18),

            ),

          ),

        ),

      ),

    );

  }

}

**Imágenes de como se ve el programa con los cambios:**

**A black screen with purple text

Description automatically generated A screenshot of a black phone

Description automatically generated A screenshot of a black phone

Description automatically generated**

**A screenshot of a black screen

Description automatically generated A screenshot of a black screen

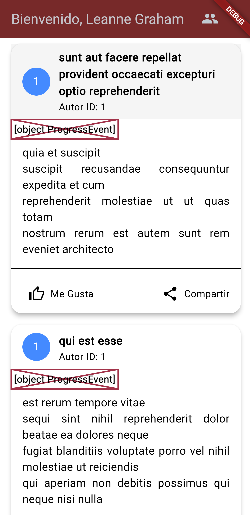
Description automatically generated A screenshot of a phone

Description automatically generated**

**A screenshot of a black screen

Description automatically generated A screenshot of a black screen

Description automatically generated A screenshot of a cell phone

Description automatically generated** **A screenshot of a cell phone

Description automatically generated**

**El folder numero 6 no lo logre aplicar debido a que la laptop no descarga el provider en el pubspec.Yaml y el código se vuelve infuncional el programa.**