**UNIVERSIDAD AUTONO DE SINALOA**

**FACULTAD DE INGENIERIA MOCHIS**

**INGENIERIA DE SOFTWARE**

Una caricatura de una persona

Descripción generada automáticamente con confianza baja

**OPERATIVA III COMPUTO EN LA NUBE**

**TEMA**:

Mini Proyecto 01

**PROFESOR**:

Gibran Uriel López Coronel

**ALUMNO**

Sánchez Guerrero Britney Eduardo

5-02

**CODIGO FUENTE**

1. **bmi\_calculator\_screen.dart**

import 'package:flutter/material.dart';

import 'package:flutter/services.dart';

import 'bmi\_result\_screen.dart';

class BMICalculatorScreen extends StatefulWidget {

  const BMICalculatorScreen({super.key});

  @override

  State<BMICalculatorScreen> createState() => \_BMICalculatorScreenState();

}

class \_BMICalculatorScreenState extends State<BMICalculatorScreen> {

  final \_formKey = GlobalKey<FormState>();

  final \_weightController = TextEditingController();

  final \_heightController = TextEditingController();

  @override

  void dispose() {

    \_weightController.dispose();

    \_heightController.dispose();

    super.dispose();

  }

  void \_calculateBMI() {

    if (\_formKey.currentState!.validate()) {

      final weight = double.parse(\_weightController.text);

      final height = double.parse(\_heightController.text);

      final bmi = weight / (height \* height);

      Navigator.of(context).push(

        MaterialPageRoute(

          builder: (context) => BMIResultScreen(

            bmi: bmi,

            weight: weight,

            height: height,

          ),

        ),

      );

    }

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: const Text('Calculadora de IMC'),

        centerTitle: true,

      ),

      body: SingleChildScrollView(

        padding: const EdgeInsets.all(20.0),

        child: Form(

          key: \_formKey,

          child: Column(

            crossAxisAlignment: CrossAxisAlignment.stretch,

            children: [

              const SizedBox(height: 20),

              // Título principal

              const Text(

                'Calcula tu Índice de Masa Corporal',

                style: TextStyle(

                  fontSize: 24,

                  fontWeight: FontWeight.bold,

                  color: Colors.blue,

                ),

                textAlign: TextAlign.center,

              ),

              const SizedBox(height: 10),

              const Text(

                'Ingresa tu peso en kilogramos y tu estatura en metros',

                style: TextStyle(

                  fontSize: 16,

                  color: Colors.grey,

                ),

                textAlign: TextAlign.center,

              ),

              const SizedBox(height: 40),

              // Fórmula IMC

              Container(

                padding: const EdgeInsets.all(20),

                decoration: BoxDecoration(

                  color: Colors.blue[50],

                  borderRadius: BorderRadius.circular(15),

                  border: Border.all(color: Colors.blue[200]!),

                ),

                child: Column(

                  children: [

                    const Text(

                      'Fórmula del IMC',

                      style: TextStyle(

                        fontSize: 18,

                        fontWeight: FontWeight.bold,

                        color: Colors.blue,

                      ),

                    ),

                    const SizedBox(height: 10),

                    const Text(

                      'IMC = Peso / Estatura²',

                      style: TextStyle(

                        fontSize: 20,

                        fontWeight: FontWeight.bold,

                        color: Colors.black87,

                      ),

                    ),

                  ],

                ),

              ),

              const SizedBox(height: 40),

              // Campo de peso

              TextFormField(

                controller: \_weightController,

                keyboardType: const TextInputType.numberWithOptions(decimal: true),

                inputFormatters: [

                  FilteringTextInputFormatter.allow(RegExp(r'^\d\*\.?\d\*')),

                ],

                decoration: InputDecoration(

                  labelText: 'Peso (kg)',

                  hintText: 'Ej: 70.5',

                  prefixIcon: const Icon(Icons.monitor\_weight, color: Colors.blue),

                  border: OutlineInputBorder(

                    borderRadius: BorderRadius.circular(15),

                  ),

                  focusedBorder: OutlineInputBorder(

                    borderRadius: BorderRadius.circular(15),

                    borderSide: const BorderSide(color: Colors.blue, width: 2),

                  ),

                ),

                validator: (value) {

                  if (value == null || value.isEmpty) {

                    return 'Por favor ingresa tu peso';

                  }

                  final weight = double.tryParse(value);

                  if (weight == null || weight <= 0) {

                    return 'Por favor ingresa un peso válido';

                  }

                  if (weight > 300) {

                    return 'El peso debe ser menor a 300 kg';

                  }

                  return null;

                },

              ),

              const SizedBox(height: 20),

              // Campo de estatura

              TextFormField(

                controller: \_heightController,

                keyboardType: const TextInputType.numberWithOptions(decimal: true),

                inputFormatters: [

                  FilteringTextInputFormatter.allow(RegExp(r'^\d\*\.?\d\*')),

                ],

                decoration: InputDecoration(

                  labelText: 'Estatura (m)',

                  hintText: 'Ej: 1.75',

                  prefixIcon: const Icon(Icons.height, color: Colors.blue),

                  border: OutlineInputBorder(

                    borderRadius: BorderRadius.circular(15),

                  ),

                  focusedBorder: OutlineInputBorder(

                    borderRadius: BorderRadius.circular(15),

                    borderSide: const BorderSide(color: Colors.blue, width: 2),

                  ),

                ),

                validator: (value) {

                  if (value == null || value.isEmpty) {

                    return 'Por favor ingresa tu estatura';

                  }

                  final height = double.tryParse(value);

                  if (height == null || height <= 0) {

                    return 'Por favor ingresa una estatura válida';

                  }

                  if (height > 3) {

                    return 'La estatura debe ser menor a 3 metros';

                  }

                  return null;

                },

              ),

              const SizedBox(height: 40),

              // Botón calcular

              ElevatedButton(

                onPressed: \_calculateBMI,

                style: ElevatedButton.styleFrom(

                  backgroundColor: Colors.blue,

                  foregroundColor: Colors.white,

                  padding: const EdgeInsets.symmetric(vertical: 15),

                  shape: RoundedRectangleBorder(

                    borderRadius: BorderRadius.circular(15),

                  ),

                  elevation: 5,

                ),

                child: const Text(

                  'Calcular IMC',

                  style: TextStyle(

                    fontSize: 18,

                    fontWeight: FontWeight.bold,

                  ),

                ),

              ),

              const SizedBox(height: 30),

              // Información adicional

              Container(

                padding: const EdgeInsets.all(15),

                decoration: BoxDecoration(

                  color: Colors.grey[100],

                  borderRadius: BorderRadius.circular(10),

                ),

                child: const Column(

                  children: [

                    Text(

                      'Información importante:',

                      style: TextStyle(

                        fontWeight: FontWeight.bold,

                        fontSize: 16,

                      ),

                    ),

                    SizedBox(height: 8),

                    Text(

                      '• El IMC es una medida aproximada de la grasa corporal\n'

                      '• No es aplicable para atletas, embarazadas o personas mayores\n'

                      '• Consulta con un profesional de la salud para una evaluación completa',

                      style: TextStyle(fontSize: 14),

                    ),

                  ],

                ),

              ),

            ],

          ),

        ),

      ),

    );

  }

}

1. **bmi\_result\_screen.dart**

import 'package:flutter/material.dart';

import 'splash\_screen.dart';

class BMIResultScreen extends StatelessWidget {

  final double bmi;

  final double weight;

  final double height;

  const BMIResultScreen({

    super.key,

    required this.bmi,

    required this.weight,

    required this.height,

  });

  String \_getBMICategory(double bmi) {

    if (bmi < 18) {

      return 'Peso Bajo';

    } else if (bmi >= 18 && bmi < 25) {

      return 'Normal';

    } else if (bmi >= 25 && bmi < 27) {

      return 'Obesidad';

    } else if (bmi >= 27 && bmi < 30) {

      return 'Obesidad Grado I';

    } else if (bmi >= 30 && bmi < 40) {

      return 'Obesidad Grado II';

    } else {

      return 'Obesidad Grado III';

    }

  }

  String \_getBMIDescription(double bmi) {

    if (bmi < 18) {

      return 'Necesario valorar signos de desnutrición';

    } else if (bmi >= 18 && bmi < 25) {

      return 'Peso saludable';

    } else if (bmi >= 25 && bmi < 27) {

      return 'Riesgo de obesidad';

    } else if (bmi >= 27 && bmi < 30) {

      return 'Riesgo relativo para desarrollar enfermedades cardiovasculares';

    } else if (bmi >= 30 && bmi < 40) {

      return 'Riesgo relativo muy alto para el desarrollo de enfermedades cardiovasculares';

    } else {

      return 'Riesgo relativo extremadamente alto para el desarrollo de enfermedades cardiovasculares';

    }

  }

  Color \_getBMIColor(double bmi) {

    if (bmi < 18) {

      return Colors.orange;

    } else if (bmi >= 18 && bmi < 25) {

      return Colors.green;

    } else if (bmi >= 25 && bmi < 27) {

      return Colors.yellow[700]!;

    } else if (bmi >= 27 && bmi < 30) {

      return Colors.orange;

    } else if (bmi >= 30 && bmi < 40) {

      return Colors.red[600]!;

    } else {

      return Colors.red[800]!;

    }

  }

  IconData \_getBMIIcon(double bmi) {

    if (bmi < 18) {

      return Icons.trending\_down;

    } else if (bmi >= 18 && bmi < 25) {

      return Icons.check\_circle;

    } else if (bmi >= 25 && bmi < 27) {

      return Icons.warning;

    } else if (bmi >= 27 && bmi < 30) {

      return Icons.warning\_amber;

    } else if (bmi >= 30 && bmi < 40) {

      return Icons.error;

    } else {

      return Icons.dangerous;

    }

  }

  @override

  Widget build(BuildContext context) {

    final category = \_getBMICategory(bmi);

    final description = \_getBMIDescription(bmi);

    final color = \_getBMIColor(bmi);

    final icon = \_getBMIIcon(bmi);

    return Scaffold(

      appBar: AppBar(

        title: const Text('Resultado del IMC'),

        centerTitle: true,

        backgroundColor: color,

        foregroundColor: Colors.white,

      ),

      body: SingleChildScrollView(

        padding: const EdgeInsets.all(20.0),

        child: Column(

          crossAxisAlignment: CrossAxisAlignment.stretch,

          children: [

            const SizedBox(height: 20),

            // Tarjeta principal con el resultado

            Card(

              elevation: 8,

              shape: RoundedRectangleBorder(

                borderRadius: BorderRadius.circular(20),

              ),

              child: Container(

                padding: const EdgeInsets.all(30),

                decoration: BoxDecoration(

                  borderRadius: BorderRadius.circular(20),

                  gradient: LinearGradient(

                    colors: [color.withValues(alpha: 0.1), color.withValues(alpha: 0.05)],

                    begin: Alignment.topLeft,

                    end: Alignment.bottomRight,

                  ),

                ),

                child: Column(

                  children: [

                    // Icono del resultado

                    Container(

                      width: 100,

                      height: 100,

                      decoration: BoxDecoration(

                        color: color,

                        shape: BoxShape.circle,

                        boxShadow: [

                        BoxShadow(

                          color: color.withValues(alpha: 0.3),

                          blurRadius: 20,

                          spreadRadius: 5,

                        ),

                        ],

                      ),

                      child: Icon(

                        icon,

                        size: 50,

                        color: Colors.white,

                      ),

                    ),

                    const SizedBox(height: 20),

                    // Valor del IMC

                    Text(

                      bmi.toStringAsFixed(1),

                      style: const TextStyle(

                        fontSize: 48,

                        fontWeight: FontWeight.bold,

                        color: Colors.black87,

                      ),

                    ),

                    const SizedBox(height: 10),

                    // Categoría

                    Text(

                      category,

                      style: TextStyle(

                        fontSize: 24,

                        fontWeight: FontWeight.bold,

                        color: color,

                      ),

                    ),

                    const SizedBox(height: 10),

                    // Descripción

                    Text(

                      description,

                      style: const TextStyle(

                        fontSize: 16,

                        color: Colors.grey,

                      ),

                      textAlign: TextAlign.center,

                    ),

                  ],

                ),

              ),

            ),

            const SizedBox(height: 30),

            // Información detallada

            Card(

              elevation: 4,

              shape: RoundedRectangleBorder(

                borderRadius: BorderRadius.circular(15),

              ),

              child: Padding(

                padding: const EdgeInsets.all(20),

                child: Column(

                  crossAxisAlignment: CrossAxisAlignment.start,

                  children: [

                    const Text(

                      'Información detallada:',

                      style: TextStyle(

                        fontSize: 18,

                        fontWeight: FontWeight.bold,

                        color: Colors.blue,

                      ),

                    ),

                    const SizedBox(height: 15),

                    \_buildInfoRow('Peso:', '${weight.toStringAsFixed(1)} kg'),

                    \_buildInfoRow('Estatura:', '${height.toStringAsFixed(2)} m'),

                    \_buildInfoRow('IMC calculado:', bmi.toStringAsFixed(1)),

                    \_buildInfoRow('Clasificación:', category),

                  ],

                ),

              ),

            ),

            const SizedBox(height: 30),

            // Tabla de rangos de IMC

            Card(

              elevation: 4,

              shape: RoundedRectangleBorder(

                borderRadius: BorderRadius.circular(15),

              ),

              child: Padding(

                padding: const EdgeInsets.all(20),

                child: Column(

                  crossAxisAlignment: CrossAxisAlignment.start,

                  children: [

                    const Text(

                      'Rangos de IMC:',

                      style: TextStyle(

                        fontSize: 18,

                        fontWeight: FontWeight.bold,

                        color: Colors.blue,

                      ),

                    ),

                    const SizedBox(height: 15),

                    \_buildBMIRangeRow('Menor a 18', 'Peso Bajo', Colors.orange),

                    \_buildBMIRangeRow('18 - 24.9', 'Normal', Colors.green),

                    \_buildBMIRangeRow('25 - 26.9', 'Obesidad', Colors.yellow[700]!),

                    \_buildBMIRangeRow('27 - 29.9', 'Obesidad Grado I', Colors.orange),

                    \_buildBMIRangeRow('30 - 39.9', 'Obesidad Grado II', Colors.red[600]!),

                    \_buildBMIRangeRow('Mayor de 40', 'Obesidad Grado III', Colors.red[800]!),

                  ],

                ),

              ),

            ),

            const SizedBox(height: 30),

            // Botones de acción

            Row(

              children: [

                Expanded(

                  child: ElevatedButton.icon(

                    onPressed: () {

                      Navigator.of(context).pop();

                    },

                    icon: const Icon(Icons.calculate),

                    label: const Text('Nuevo Cálculo'),

                    style: ElevatedButton.styleFrom(

                      backgroundColor: Colors.blue,

                      foregroundColor: Colors.white,

                      padding: const EdgeInsets.symmetric(vertical: 15),

                      shape: RoundedRectangleBorder(

                        borderRadius: BorderRadius.circular(15),

                      ),

                    ),

                  ),

                ),

                const SizedBox(width: 15),

                Expanded(

                  child: ElevatedButton.icon(

                    onPressed: () {

                      // Regresar a la pantalla de splash que luego navegará a la calculadora

                      Navigator.of(context).pushAndRemoveUntil(

                        MaterialPageRoute(

                          builder: (context) => const SplashScreen(),

                        ),

                        (route) => false,

                      );

                    },

                    icon: const Icon(Icons.home),

                    label: const Text('Inicio'),

                    style: ElevatedButton.styleFrom(

                      backgroundColor: Colors.grey[600],

                      foregroundColor: Colors.white,

                      padding: const EdgeInsets.symmetric(vertical: 15),

                      shape: RoundedRectangleBorder(

                        borderRadius: BorderRadius.circular(15),

                      ),

                    ),

                  ),

                ),

              ],

            ),

          ],

        ),

      ),

    );

  }

  Widget \_buildInfoRow(String label, String value) {

    return Padding(

      padding: const EdgeInsets.symmetric(vertical: 5),

      child: Row(

        mainAxisAlignment: MainAxisAlignment.spaceBetween,

        children: [

          Text(

            label,

            style: const TextStyle(

              fontSize: 16,

              fontWeight: FontWeight.w500,

            ),

          ),

          Text(

            value,

            style: const TextStyle(

              fontSize: 16,

              color: Colors.black87,

            ),

          ),

        ],

      ),

    );

  }

  Widget \_buildBMIRangeRow(String range, String category, Color color) {

    return Padding(

      padding: const EdgeInsets.symmetric(vertical: 3),

      child: Row(

        children: [

          Container(

            width: 12,

            height: 12,

            decoration: BoxDecoration(

              color: color,

              shape: BoxShape.circle,

            ),

          ),

          const SizedBox(width: 10),

          Expanded(

            flex: 2,

            child: Text(

              range,

              style: const TextStyle(fontSize: 14),

            ),

          ),

          Expanded(

            flex: 3,

            child: Text(

              category,

              style: const TextStyle(fontSize: 14),

            ),

          ),

        ],

      ),

    );

  }

}

1. **splash\_screen.dart**

import 'package:flutter/material.dart';

import 'bmi\_calculator\_screen.dart';

class SplashScreen extends StatefulWidget {

  const SplashScreen({super.key});

  @override

  State<SplashScreen> createState() => \_SplashScreenState();

}

class \_SplashScreenState extends State<SplashScreen>

    with SingleTickerProviderStateMixin {

  late AnimationController \_animationController;

  late Animation<double> \_fadeAnimation;

  late Animation<double> \_scaleAnimation;

  @override

  void initState() {

    super.initState();

    \_animationController = AnimationController(

      duration: const Duration(seconds: 2),

      vsync: this,

    );

    \_fadeAnimation = Tween<double>(begin: 0.0, end: 1.0).animate(

      CurvedAnimation(parent: \_animationController, curve: Curves.easeIn),

    );

    \_scaleAnimation = Tween<double>(begin: 0.5, end: 1.0).animate(

      CurvedAnimation(parent: \_animationController, curve: Curves.elasticOut),

    );

    \_animationController.forward();

    // Navegar a la pantalla principal después de 3 segundos

    Future.delayed(const Duration(seconds: 3), () {

      if (mounted) {

        Navigator.of(context).pushReplacement(

          MaterialPageRoute(builder: (context) => const BMICalculatorScreen()),

        );

      }

    });

  }

  @override

  void dispose() {

    \_animationController.dispose();

    super.dispose();

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      backgroundColor: Colors.blue[50],

      body: Center(

        child: AnimatedBuilder(

          animation: \_animationController,

          builder: (context, child) {

            return FadeTransition(

              opacity: \_fadeAnimation,

              child: ScaleTransition(

                scale: \_scaleAnimation,

                child: Column(

                  mainAxisAlignment: MainAxisAlignment.center,

                  children: [

                    Container(

                      width: 120,

                      height: 120,

                      decoration: BoxDecoration(

                        color: Colors.blue,

                        borderRadius: BorderRadius.circular(60),

                        boxShadow: [

                          BoxShadow(

                            color: Colors.blue.withValues(alpha: 0.3),

                            blurRadius: 20,

                            spreadRadius: 5,

                          ),

                        ],

                      ),

                      child: const Icon(

                        Icons.favorite,

                        size: 60,

                        color: Colors.white,

                      ),

                    ),

                    const SizedBox(height: 30),

                    const Text(

                      'Calculadora de IMC',

                      style: TextStyle(

                        fontSize: 28,

                        fontWeight: FontWeight.bold,

                        color: Colors.blue,

                      ),

                    ),

                    const SizedBox(height: 10),

                    const Text(

                      'Calcula tu Índice de Masa Corporal',

                      style: TextStyle(fontSize: 16, color: Colors.grey),

                    ),

                    const SizedBox(height: 50),

                    const CircularProgressIndicator(

                      valueColor: AlwaysStoppedAnimation<Color>(Colors.blue),

                    ),

                  ],

                ),

              ),

            );

          },

        ),

      ),

    );

  }

}

1. **main.dart**

import 'package:flutter/material.dart';

import 'screens/splash\_screen.dart';

void main() {

  runApp(const BMICalculatorApp());

}

class BMICalculatorApp extends StatelessWidget {

  const BMICalculatorApp({super.key});

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      title: 'Calculadora de IMC',

      theme: ThemeData(

        primarySwatch: Colors.blue,

        visualDensity: VisualDensity.adaptivePlatformDensity,

        appBarTheme: const AppBarTheme(

          backgroundColor: Colors.blue,

          foregroundColor: Colors.white,

          elevation: 0,

        ),

      ),

      home: const SplashScreen(),

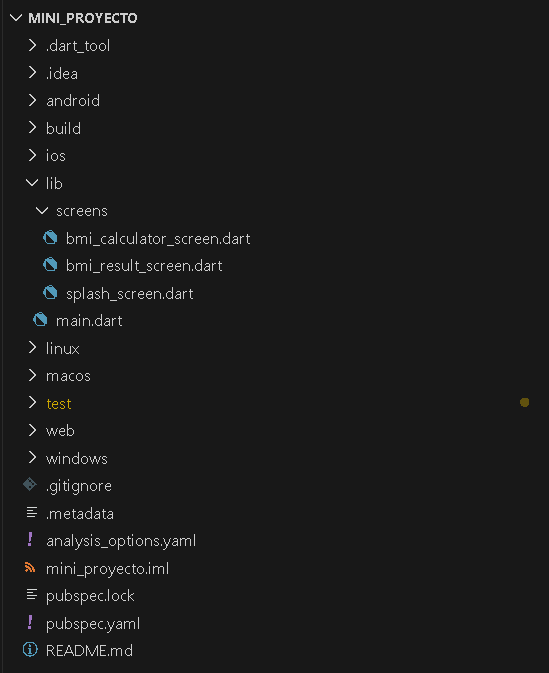
      debugShowCheckedModeBanner: false,

    );

  }

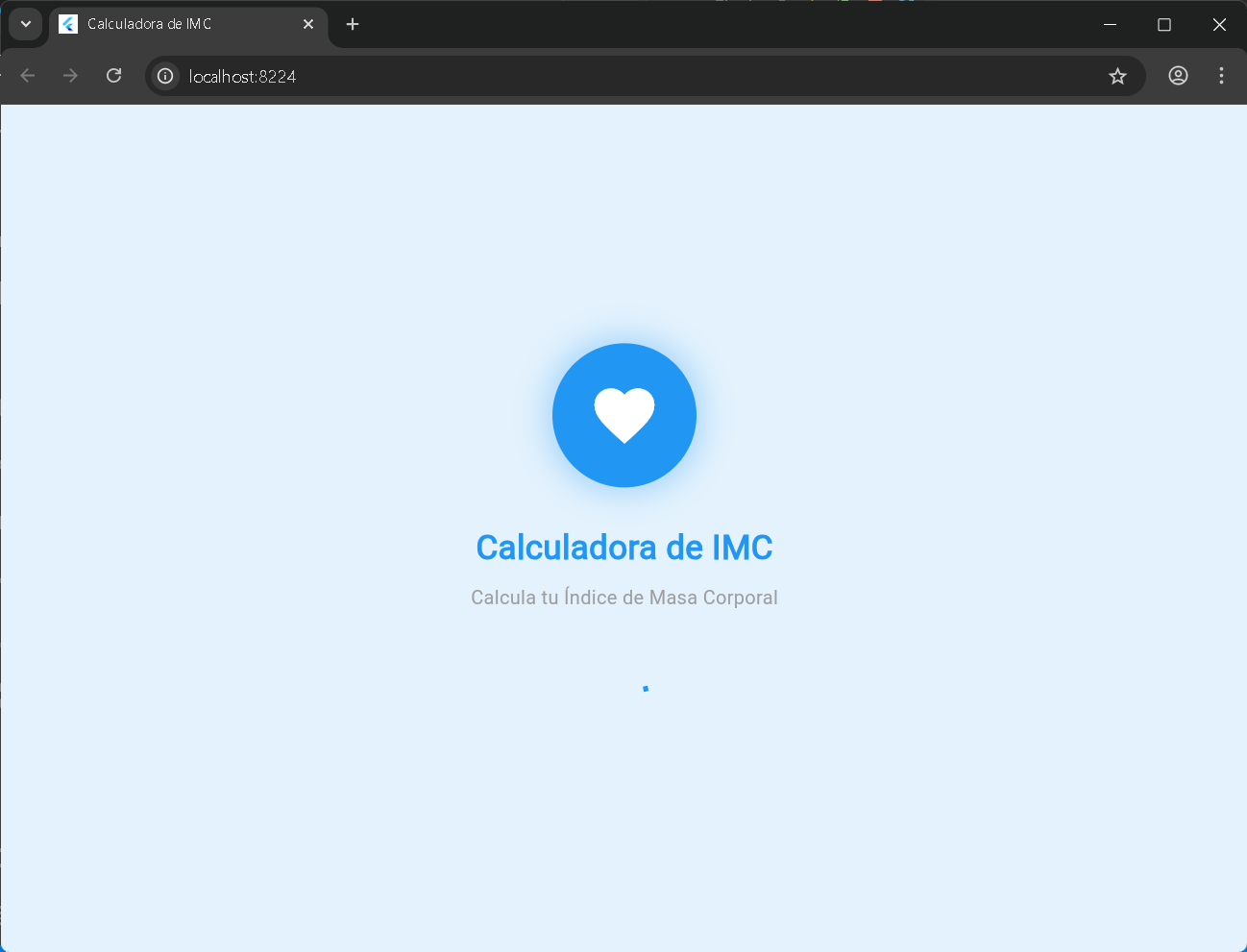
}

**Estructura de las carpetas**

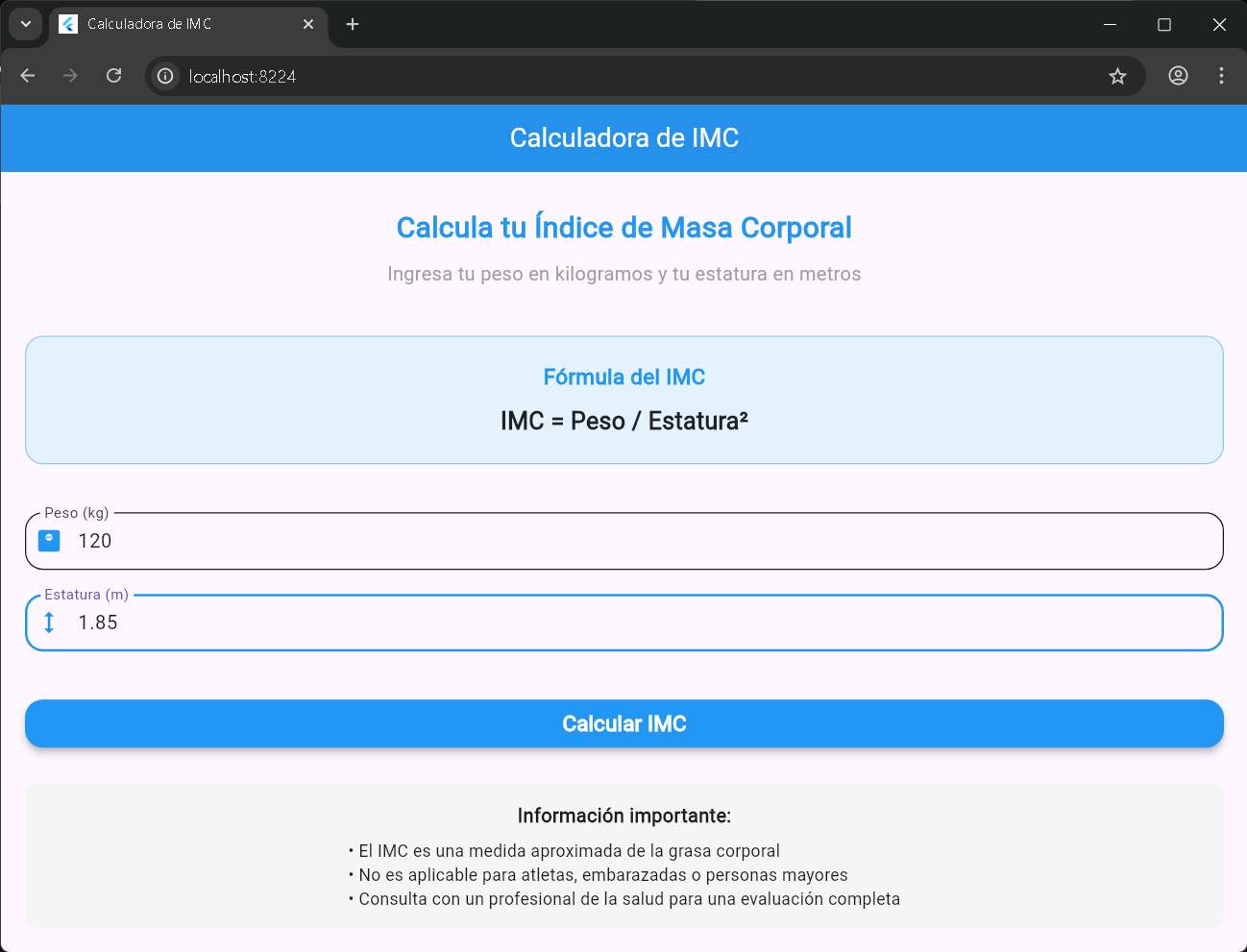
****

1. **Aplicación en funcionamiento**

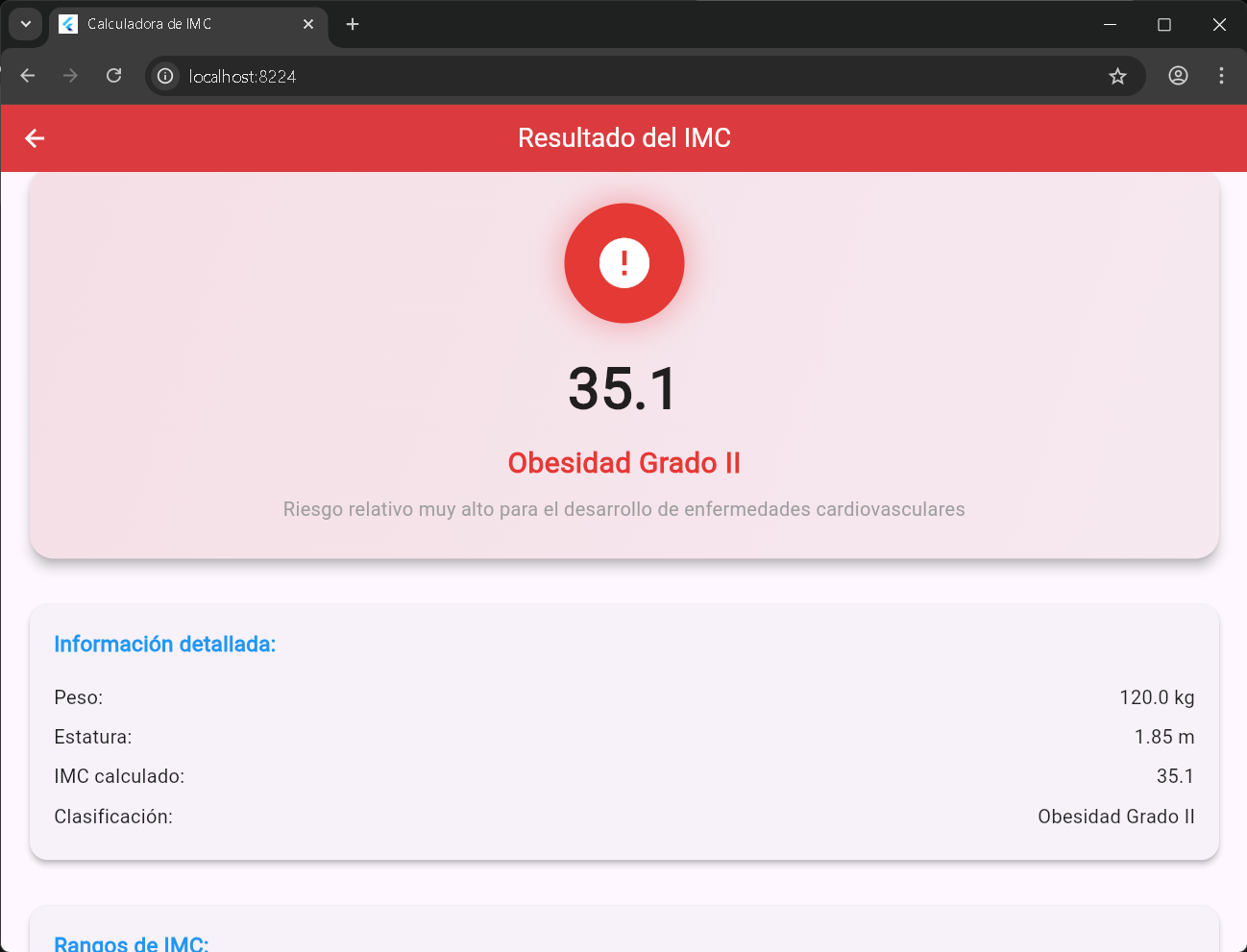
**Inicio**

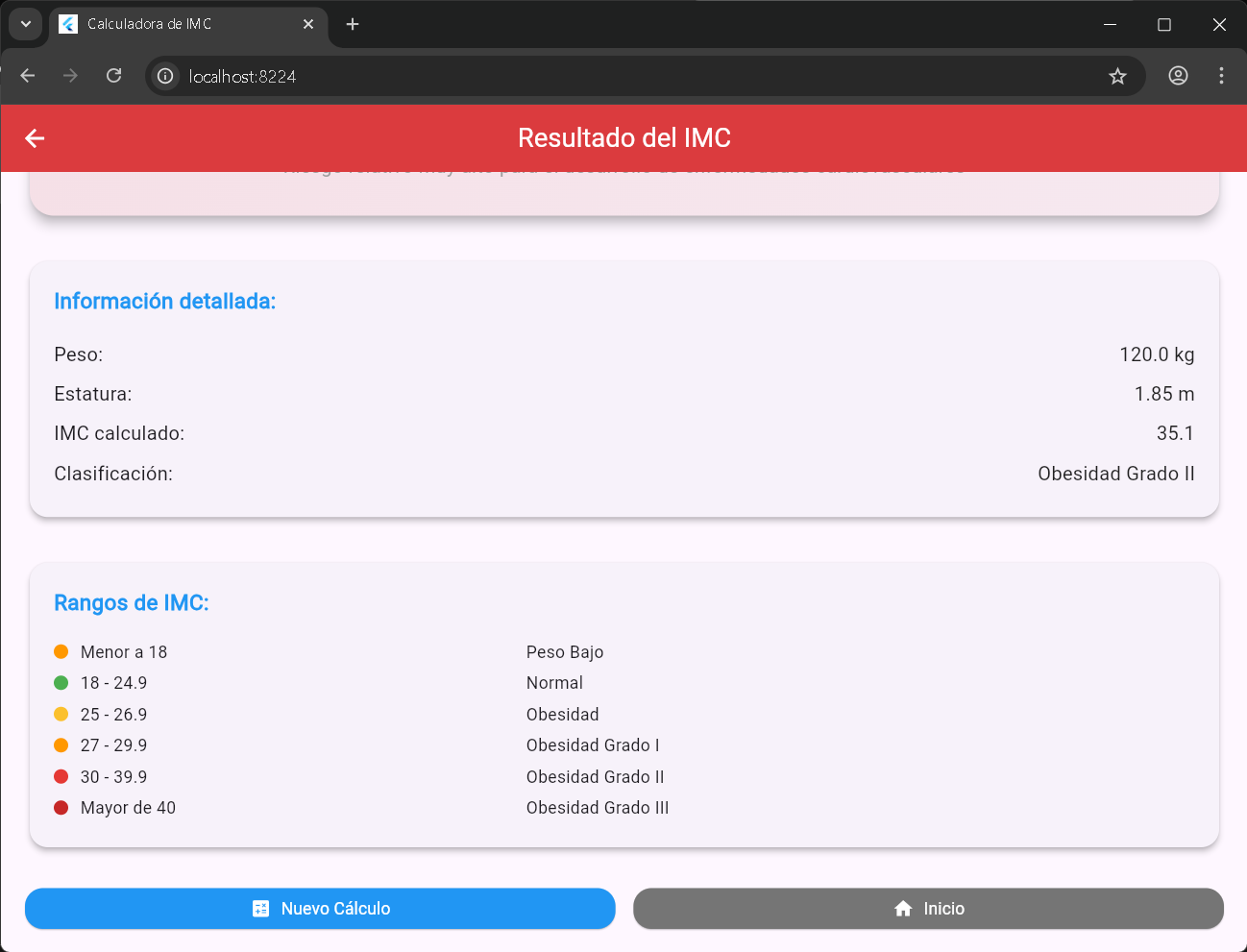
****

**Principal**

****

**Resultados**

****

****