import 'package:flutter/material.dart';

import 'package:intl/intl.dart'; // Asegúrate que la dependencia esté en pubspec.yaml

void main() {

runApp(const MyApp());

}

class MyApp extends StatelessWidget {

const MyApp({super.key});

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'My Finance',

theme: ThemeData(

// Usar colorSchemeSeed es más moderno que primarySwatch

colorSchemeSeed: Colors.teal,

useMaterial3: true, // Opcional: Habilitar Material 3

),

home: const MyHomePage(),

);

}

}

class MyHomePage extends StatefulWidget {

const MyHomePage({super.key});

@override

State<MyHomePage> createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

double ingresos = 0.00;

double gastos = 0.00;

// Balance se calcula, no necesita ser una variable de estado separada si siempre es ingresos - gastos

// double balance = 0.00;

List<Map<String, dynamic>> registroOperaciones = [];

final TextEditingController \_montoController = TextEditingController();

final TextEditingController \_nombreController = TextEditingController();

// Usar un enum o constantes podría ser más robusto que strings para el tipo

String \_tipoTransaccionSeleccionado = ''; // Estado para saber qué botón (Ingreso/Gasto) está "activo"

// Getter calculado para el balance

double get balance => ingresos - gastos;

@override

void dispose() {

// Limpiar controllers al destruir el widget

\_montoController.dispose();

\_nombreController.dispose();

super.dispose();

}

void \_agregarTransaccion(String tipo) {

final String nombre = \_nombreController.text.trim();

final double? monto = double.tryParse(\_montoController.text.trim());

if (nombre.isEmpty || monto == null || monto <= 0) {

// Mostrar un mensaje de error al usuario

showDialog(

context: context,

builder: (context) => AlertDialog(

title: const Text('Error'),

content: const Text('Por favor, ingrese un nombre y un monto válido.'),

actions: [

TextButton(

onPressed: () => Navigator.pop(context),

child: const Text('OK'),

),

],

),

);

return;

}

setState(() {

registroOperaciones.add({

'nombre': nombre,

'monto': monto,

'tipo': tipo,

'fecha': DateTime.now(),

});

if (tipo == 'Ingreso') {

ingresos += monto;

} else {

gastos += monto;

}

// Limpiar los campos después de agregar la transacción

\_nombreController.clear();

\_montoController.clear();

\_tipoTransaccionSeleccionado = ''; // Deselecciona los botones

});

}

@override

Widget build(BuildContext context) {

final NumberFormat currencyFormat = NumberFormat.currency(locale: 'es\_CO', symbol: '\$');

return Scaffold(

appBar: AppBar(

title: const Text('My Finance'),

),

body: Padding(

padding: const EdgeInsets.all(16.0),

child: Column(

crossAxisAlignment: CrossAxisAlignment.stretch,

children: [

Text(

'Balance: ${currencyFormat.format(balance)}',

style: Theme.of(context).textTheme.headlineSmall,

textAlign: TextAlign.center,

),

const SizedBox(height: 20),

Row(

mainAxisAlignment: MainAxisAlignment.spaceEvenly,

children: [

ElevatedButton(

onPressed: () {

setState(() {

\_tipoTransaccionSeleccionado = 'Ingreso';

});

},

style: ElevatedButton.styleFrom(

backgroundColor: \_tipoTransaccionSeleccionado == 'Ingreso' ? Colors.green : null,

),

child: const Text('Ingreso'),

),

ElevatedButton(

onPressed: () {

setState(() {

\_tipoTransaccionSeleccionado = 'Gasto';

});

},

style: ElevatedButton.styleFrom(

backgroundColor: \_tipoTransaccionSeleccionado == 'Gasto' ? Colors.red : null,

),

child: const Text('Gasto'),

),

],

),

const SizedBox(height: 20),

TextField(

controller: \_nombreController,

decoration: const InputDecoration(labelText: 'Nombre de la transacción'),

),

TextField(

controller: \_montoController,

keyboardType: TextInputType.number,

decoration: const InputDecoration(labelText: 'Monto'),

),

const SizedBox(height: 20),

ElevatedButton(

onPressed: \_tipoTransaccionSeleccionado.isNotEmpty ? () => \_agregarTransaccion(\_tipoTransaccionSeleccionado) : null,

child: const Text('Agregar Transacción'),

),

const SizedBox(height: 20),

Expanded(

child: ListView.builder(

itemCount: registroOperaciones.length,

itemBuilder: (context, index) {

final operacion = registroOperaciones[index];

final tipo = operacion['tipo'];

final nombre = operacion['nombre'];

final monto = operacion['monto'];

final fecha = operacion['fecha'];

return Card(

child: ListTile(

leading: Icon(tipo == 'Ingreso' ? Icons.add : Icons.remove, color: tipo == 'Ingreso' ? Colors.green : Colors.red),

title: Text(nombre),

subtitle: Text(DateFormat('yyyy-MM-dd – kk:mm').format(fecha)),

trailing: Text(currencyFormat.format(monto), style: TextStyle(color: tipo == 'Ingreso' ? Colors.green : Colors.red)),

),

);

},

),

),

],

),

),

);

}

}