import 'package:flutter/material.dart';

import 'package:provider/provider.dart';

void main() {

runApp(

ChangeNotifierProvider(

create: (context) => FinanceData(),

child: const MyApp(),

),

);

}

class FinanceData extends ChangeNotifier {

double balance = 900.00;

List<Transaction> incomeTransactions = [

Transaction(description: "SALARIO", amount: 900.00),

];

List<Transaction> expenseTransactions = [];

void addIncome(String description, double amount) {

incomeTransactions.add(Transaction(description: description, amount: amount));

balance += amount;

notifyListeners();

}

void addExpense(String description, double amount) {

expenseTransactions.add(Transaction(description: description, amount: amount));

balance -= amount;

notifyListeners();

}

}

class Transaction {

String description;

double amount;

Transaction({required this.description, required this.amount});

}

class MyApp extends StatelessWidget {

const MyApp({super.key});

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Control de Finanzas',

theme: ThemeData(

colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),

useMaterial3: true,

),

home: const MyHomePage(),

);

}

}

class MyHomePage extends StatefulWidget {

const MyHomePage({super.key});

@override

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

}

class \_MyHomePageState extends State<MyHomePage> {

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

backgroundColor: Theme.of(context).colorScheme.inversePrimary,

title: const Text('Control de Finanzas'),

),

body: Center(

child: Padding(

padding: const EdgeInsets.all(16.0),

child: Column(

mainAxisAlignment: MainAxisAlignment.start,

crossAxisAlignment: CrossAxisAlignment.center,

children: <Widget>[

const SizedBox(height: 20),

Text(

'Balance:',

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

),

Text(

'\$${context.watch<FinanceData>().balance.toStringAsFixed(2)}',

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

),

const SizedBox(height: 20),

Row(

mainAxisAlignment: MainAxisAlignment.center,

children: [

ElevatedButton(

onPressed: () {

\_showAddIncomeDialog(context);

},

child: const Text('Añadir Ingreso'),

),

const SizedBox(width: 20),

ElevatedButton(

onPressed: () {

\_showAddExpenseDialog(context);

},

child: const Text('Añadir Gasto'),

),

],

),

const SizedBox(height: 30),

Text(

'Transacciones de Ingreso:',

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

),

Expanded(

child: ListView.builder(

itemCount: context.watch<FinanceData>().incomeTransactions.length,

itemBuilder: (context, index) {

final transaction = context.watch<FinanceData>().incomeTransactions[index];

return Card(

child: Padding(

padding: const EdgeInsets.all(8.0),

child: ListTile(

title: Text(transaction.description),

subtitle: Text('\$${transaction.amount.toStringAsFixed(2)}'),

),

),

);

},

),

),

const SizedBox(height: 20),

Text(

'Transacciones de Gasto:',

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

),

Expanded(

child: ListView.builder(

itemCount: context.watch<FinanceData>().expenseTransactions.length,

itemBuilder: (context, index) {

final transaction = context.watch<FinanceData>().expenseTransactions[index];

return Card(

child: Padding(

padding: const EdgeInsets.all(8.0),

child: ListTile(

title: Text(transaction.description),

subtitle: Text('\$${transaction.amount.toStringAsFixed(2)}'),

),

),

);

},

),

),

],

),

),

),

);

}

void \_showAddIncomeDialog(BuildContext context) {

String description = '';

double amount = 0.0;

showDialog(

context: context,

builder: (BuildContext context) {

return AlertDialog(

title: const Text('Añadir Ingreso'),

content: Column(

mainAxisSize: MainAxisSize.min,

children: [

TextField(

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

onChanged: (value) {

description = value;

},

),

TextField(

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

keyboardType: TextInputType.number,

onChanged: (value) {

amount = double.tryParse(value) ?? 0.0;

},

),

],

),

actions: [

TextButton(

onPressed: () {

Navigator.of(context).pop();

},

child: const Text('Cancelar'),

),

TextButton(

onPressed: () {

Provider.of<FinanceData>(context, listen: false).addIncome(description, amount);

Navigator.of(context).pop();

},

child: const Text('Añadir'),

),

],

);

},

);

}

void \_showAddExpenseDialog(BuildContext context) {

String description = '';

double amount = 0.0;

showDialog(

context: context,

builder: (BuildContext context) {

return AlertDialog(

title: const Text('Añadir Gasto'),

content: Column(

mainAxisSize: MainAxisSize.min,

children: [

TextField(

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

onChanged: (value) {

description = value;

},

),

TextField(

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

keyboardType: TextInputType.number,

onChanged: (value) {

amount = double.tryParse(value) ?? 0.0;

},

),

],

),

actions: [

TextButton(

onPressed: () {

Navigator.of(context).pop();

},

child: const Text('Cancelar'),

),

TextButton(

onPressed: () {

Provider.of<FinanceData>(context, listen: false).addExpense(description, amount);

Navigator.of(context).pop();

},

child: const Text('Añadir'),

),

],

);

},

);

}

}