

```
import React, { useState } from 'react';
import { View, Text, TextInput, Button, FlatList, StyleSheet } from
'react-native';

// Interface para representar uma transação
interface Transaction {
  id: string;
  description: string;
  amount: number;
  type: 'income' | 'expense'; // 'income' para receita, 'expense' para
despesa
}

export default function App() {
  // Estado para armazenar as transações na memória
  const [transactions, setTransactions] = useState<Transaction[]>([]);
  const [description, setDescription] = useState('');
  const [amount, setAmount] = useState('');

  // Função para adicionar uma nova transação
  const addTransaction = (type: 'income' | 'expense') => {
    const amountValue = parseFloat(amount);

    if (!description || isNaN(amountValue)) {
      alert('Por favor, preencha todos os campos corretamente.');
```

return;

```
    }

    const newTransaction: Transaction = {
      id: Date.now().toString(), // Gera um ID único baseado no
timestamp
      description,
      amount: amountValue,
      type,
    };

    // Adiciona a nova transação à lista
    setTransactions((prevTransactions) => [...prevTransactions,
newTransaction]);

    // Limpa os campos após adicionar
```

```

    setDescription('');
    setAmount('');
  };

  // Calcula o saldo total (receitas - despesas)
  const calculateBalance = () => {
    const totalIncome = transactions
      .filter((t) => t.type === 'income')
      .reduce((sum, t) => sum + t.amount, 0);

    const totalExpense = transactions
      .filter((t) => t.type === 'expense')
      .reduce((sum, t) => sum + t.amount, 0);

    return totalIncome - totalExpense;
  };

  return (
    <View style={styles.container}>
      {/* Título do Aplicativo */}
      <Text style={styles.title}>Controle Financeiro</Text>

      {/* Formulário para Adicionar Transações */}
      <View style={styles.form}>
        <TextInput
          placeholder="Descrição"
          value={description}
          onChangeText={setDescription}
          style={styles.input}
        />
        <TextInput
          placeholder="Valor"
          value={amount}
          onChangeText={setAmount}
          keyboardType="numeric"
          style={styles.input}
        />
        <View style={styles.buttonsContainer}>
          <Button title="Adicionar Receita" onPress={() =>
addTransaction('income')} />

```

```

        <Button title="Adicionar Despesa" onPress={() =>
addTransaction('expense')} />
    </View>
</View>

    {/* Saldo Total */}
    <Text style={styles.balance}>Saldo: R$
{calculateBalance().toFixed(2)}</Text>

    {/* Lista de Transações */}
    <FlatList
        data={transactions}
        keyExtractor={({item}) => item.id}
        renderItem={({ item }) => (
            <View style={styles.transactionItem}>
                <Text>{item.description}</Text>
                <Text style={item.type === 'income' ? styles.income :
styles.expense}>
                    {item.type === 'income' ? '+R$' : '-R$'}
{item.amount.toFixed(2)}
                </Text>
            </View>
        )}
        ListEmptyComponent={<Text style={styles.emptyList}>Nenhuma
transação registrada.</Text>}
    />
</View>
);
}

// Estilos do Componente
const styles = StyleSheet.create({
    container: {
        flex: 1,
        padding: 20,
        backgroundColor: '#f5f5f5',
    },
    title: {
        fontSize: 24,
        fontWeight: 'bold',
        textAlign: 'center',
    },

```

```
    marginVertical: 20,
  },
  form: {
    marginBottom: 20,
  },
  input: {
    borderWidth: 1,
    borderColor: '#ccc',
    borderRadius: 5,
    padding: 10,
    marginBottom: 10,
    backgroundColor: '#fff',
  },
  buttonsContainer: {
    flexDirection: 'row',
    justifyContent: 'space-between',
  },
  balance: {
    fontSize: 18,
    fontWeight: 'bold',
    textAlign: 'center',
    marginVertical: 10,
  },
  transactionItem: {
    flexDirection: 'row',
    justifyContent: 'space-between',
    padding: 10,
    marginVertical: 5,
    backgroundColor: '#fff',
    borderRadius: 5,
  },
  income: {
    color: 'green',
  },
  expense: {
    color: 'red',
  },
  emptyList: {
    textAlign: 'center',
    marginTop: 20,
    color: '#666',
  },
}
```

```
    },  
});
```