

Sessão 15 – Módulo 02 – Fundamentos JavaScript – aula 14 – Projeto

Dashboard Financeiro Pessoal (Vanilla JavaScript)

Objetivo Geral

Desenvolver uma aplicação web de controle financeiro pessoal utilizando JavaScript puro (Vanilla JS), aplicando:

Manipulação dinâmica do DOM

Programação modular com ESModules

Persistência de dados com LocalStorage

Métodos funcionais como reduce()

Separação de responsabilidades

O HTML e CSS já estão fornecidos.

O desafio consiste em implementar toda a lógica utilizando apenas JavaScript.

Alterações no HTML para o seu toque pessoal estão permitidas e são estimuladas, o projeto base é só para ajudar na economia de tempo na preparação do layout.

Contexto do Problema

Um utilizador deseja controlar suas finanças pessoais através de um dashboard simples que permita:

Adicionar receitas e despesas

Visualizar histórico de transações

Acompanhar saldo total

Manter os dados mesmo após atualizar a página

Requisitos Funcionais

Formulário de Transações

A aplicação deve permitir:

- Inserir descrição
- Inserir valor numérico positivo
- Selecionar tipo (receita ou despesa)
- Registrar automaticamente a data da transação

Ao clicar em **"Adicionar ao histórico"**:

- Validar os campos
- Criar objeto de transação
- Armazenar no sistema
- Atualizar a interface
- Deverá ainda ser implementado uma solução de retirar uma transação.

Lista de Transações (DOM Dinâmico)

A lista deve ser gerada dinamicamente via JavaScript.

Não deve haver dados fixos no HTML.

Cada item deve refletir:

Descrição

Tipo (com classe visual correspondente)

Data

Valor formatado

A lista deve ser re-renderizada sempre que o estado mudar.

Cálculo de Totais (Obrigatório uso de `reduce()`)

A aplicação deve calcular dinamicamente:

Saldo total

Total de receitas

Total de despesas

É obrigatório utilizar o método `.reduce()` para pelo menos um dos cálculos.

Persistência de Dados

As transações devem ser salvas no `localStorage`.

Ao recarregar a página, os dados devem ser restaurados automaticamente.

Os dados devem ser armazenados como JSON.

Organização em Módulos (ESModules)

A aplicação deve estar organizada em múltiplos arquivos JS