

Faculdade Unicesumar

Engenharia de Software

Marcela Staron, Severo Junior, Alan Matheus, Laryssa Bida

My finance

Ponta Grossa

16/06/2025

My Finance



MyFinance \$

Disciplina: Programação Front End

Aluno(s): Marcela Staron, Severo Junior, Laryssa Bida, Alan Matheus Fernandes

1. Introdução

O projeto MyFinance tem como objetivo desenvolver uma aplicação web para controle de gastos pessoais, permitindo que o usuário registre despesas de forma simples e eficiente. A proposta é oferecer uma interface intuitiva para adicionar, visualizar e remover gastos, além de apresentar o total acumulado, auxiliando no gerenciamento financeiro pessoal.

1.1 Funcionalidades

- Adicionar gastos com descrição e valor.
- Visualizar lista de gastos em tabela estilizada.
- Remover gastos individualmente.
- Cálculo automático do total dos gastos.
- Layout fixo com cabeçalho e rodapé para melhor usabilidade.
- Responsividade para diferentes tamanhos de tela.

2. Desenvolvimento

2.1 Tecnologias Utilizadas

- React: Biblioteca JavaScript para construção da interface.
- Tailwind CSS: CSS utilitário para estilização rápida e responsiva.
- JavaScript : Para lógica e manipulação dos dados.
- HTML5 & CSS3: Estrutura e estilos básicos da página.

2.2 Estrutura HTML e Semântica

O arquivo base public/index.html define a estrutura semântica com:

- <header> fixo contendo o título "My Finance".
- <main id="root"> onde o React renderiza o conteúdo dinâmico.
- <footer> fixo com informações de contato e mensagem.

2.3 Lógica de Programação

- Validação: O formulário valida se a descrição não está vazia e se o valor é um número positivo antes de adicionar o gasto.
- Armazenamento: Os gastos são armazenados no estado do componente React usando o hook useState.
- Manipulação: Funções para adicionar e remover gastos atualizam o estado, disparando re-renderização automática da interface.
- Cálculo do total: Soma dinâmica dos valores dos gastos exibida na tabela.

3. Desafios Encontrados e Soluções

- Manter layout fixo com conteúdo dinâmico: foi necessário usar padding e Flexbox para evitar que o conteúdo ficasse oculto atrás do cabeçalho ou rodapé.
- Validação do formulário: implementação de mensagens de erro claras para evitar dados inválidos.
- Ajuste das cores de acordo com a emoção do usuário.

4. Conclusão

Aprendizados

- Uso do React para controlar dados e atualizar a tela.
- Usei Tailwind CSS para deixar o visual rápido e responsivo.
- Organizei o código em componentes para facilitar mudanças futuras.

Sugestões de Melhorias

- Criar login para vários usuários.
- Mostrar gráficos para acompanhar as despesas.
- Permitir filtrar e categorizar os gastos.

Guia de Estilo (Style Guide)

1. Paleta de Cores:

- Primária: #16A34A
- Secundária: #0F1F0F
- Texto: #E6E6E6
- Desabilitado: #555555
- Borda: #2D4B2D
- Foco: #3CB371

2. Tipografia

- Fonte Base: 'Inter', sans-serif
- Tamanhos:
 - Base: 16px
 - Label: 14px
- Pesos:
 - Títulos: 600
 - Texto normal: 400

3. Layout e Espaçamento

- **Padding/Margin:**
 - Extra pequeno (xs): 4px
 - Pequeno (sm): 8px
 - Médio (md): 16px
 - Grande (lg): 24px
- **Sistema de Layout:**
 - Flexbox para organizar as seções em coluna no mobile e linha em telas maiores
 - Layout centralizado com largura máxima de 500px por seção
 - Cabeçalho e rodapé fixos com 64px de altura

4. Documentação dos Componentes

4.1 Button

Props:

- variant: "primary"
- disabled: boolean
- onClick: função de callback
- children: conteúdo textual ou elementos

Estados Visuais:

- Normal: fundo verde , texto branco
- Hover: fundo verde escuro
- Disabled: fundo cinza escuro , cursor not-allowed

4.2 Input

Props:

- label: string
- placeholder: string
- value: string
- onChange: função
- disabled: boolean

Estados Visuais:

- Normal: Borda cinza clara ,texto escuro para boa leitura
- Disabled: Fundo cinza escuro ,texto cinza claro

4.3 Componente Card

Props:

- title: string
- children: ReactNode

4.4 Storybook

- Componentes organizados por categorias: Botões, Inputs, Cards
- Cada componente possui variações de estado (normal, hover, disabled)
- Exemplos de uso e comportamento isolado para facilitar desenvolvimento e testes

5. Principais Interfaces

My Finance

Adicionar Gasto

Descrição

Ex: Supermercado

Valor (R\$)

Ex: 150.50

Adicionar


Controle de Gastos

Limpar Histórico

Nenhum gasto cadastrado.

Histórico de Gastos

Nenhum gasto no histórico.

Desenvolvido com  para seu bolso.

My Finance

Adicionar Gasto

Descrição

Ex: Supermercado

Valor (R\$)


Ex: 150.50

Adicionar

Controle de Gastos

Limpar Histórico

Descrição	Valor (R\$)	Ações
Carro	200.50	Remover
Faculdade	900.00	Remover
Lazer	300.50	Remover
Total	1401.00	

Desenvolvido com  para seu bolso.

My Finance

Adicionar Gasto

Descrição

Ex: Supermercado

Valor (R\$)

Ex: 150.50

Adicionar


Controle de Gastos

Limpar Histórico

Nenhum gasto cadastrado.

Histórico de Gastos

- Lazer: R\$ 300.50
- Faculdade: R\$ 900.00
- Carro : R\$ 200.50

Desenvolvido com  para seu bolso.

My Finance

Adicionar Gasto

Descrição

Ex: Supermercado

Valor (R\$)

Ex: 150.50

Descrição é obrigatória

Adicionar


Controle de Gastos

Limpar Histórico

Nenhum gasto cadastrado.

Histórico de Gastos

Nenhum gasto no histórico.

Desenvolvido com  para seu bolso.