

# PROJETO: CALCULADORAS MULTIFUNCIONAIS

---

## 1. Visão Geral das Demandas

Este projeto consiste em uma aplicação web com diversas calculadoras:

- **Calculadora de Descontos** (base já implementada)
- **Custo de Combustível**
- **Custo por Unidade de Medida**
- **Formulas físicas Força/Aceleração, Distância, Velocidade, Trabalho, Lei de Ohm, Potência, Força Elétrica e Potencial Elétrico**
- **Conversor de Unidades de Temperatura, Comprimento, Capacidade, Massa e Volume**
- **Conversão de Moedas**
- **Cálculo de IMC**

O objetivo é reforçar conceitos de HTML, CSS, JavaScript e fluxo de trabalho colaborativo via Git/GitHub.

---

## 2. Normas de Desenvolvimento

### 2.1 Estrutura de Pastas

```
project-root/
├── src/
│   ├── css/
│   │   └── style.css           # Estilos globais e temas
│   ├── js/
│   │   ├── common.js         # Lógica compartilhada (menu, helpers)
│   │   ├── desconto.js       # Script específico da página desconto
│   │   ├── imc.js            # Script específico da página IMC
│   │   └── ... (scripts por página, nomeados conforme data-page)
│   ├── index.html
│   ├── desconto.html
│   ├── imc.html
│   └── ... (outras calculadoras)
├── README.md                 # Documentação do projeto
└── .gitignore                # Arquivos e pastas a ignorar pelo Git
```

Cada página deve ter seu script específico e deve chamar o script `common.js`

### 2.2 Convenções de Código




- **HTML:** usar tags semânticas (`<header>`, `<main>`, `<footer>`).
- **CSS:** variáveis `--nome-variavel`;
- **JavaScript:** `camelCase` para nome de variáveis e funções nomeadas;




- **Commit messages:** Conventional Commits (**feat**: novo recurso, **fix**: correção de erro, **docs**: documentação ou comentários, **refactor**: refatoração de código ou re-estruturação de código).

## 2.3 Pull Requests (PR)

- Criar Pull Requests (PR) pontuais com descrição clara da feature (demanda) desenvolvida.
- Referenciar issue ou tarefa.
- Marcar líder do grupo como revisor.

## 3. Especificações das Demandas

 Tema	 Descrição	 Requisitos Técnicos Mínimos
<b>Custo de Combustível</b>	Calculadora para saber quanto foi gasto em uma viagem com base no consumo.	<ul style="list-style-type: none"> <li>- Input: distância (km), consumo do veículo (km/l), preço do combustível (R\$).</li> <li>- Output: gasto total em R\$.</li> <li>- Layout com feedback visual do resultado.</li> </ul>
<b>Custo por Unidade de Medida</b>	Calcular o custo de um produto tendo como base o valor de por unidade de medida (metro/km, litros/ml e kg/grama )	<ul style="list-style-type: none"> <li>- Input: preço, quantidade, select box da unidade de medida.</li> <li>- Output: valor total em 'x unidade de medida' .</li> <li>- Interface clara e interativa.</li> </ul>
<b>Fórmulas Físicas</b>	Aplicativo que calcula valores com base nas fórmulas físicas estudadas.	<ul style="list-style-type: none"> <li>- Fórmulas: Força = m.a, Velocidade = d/t, Trabalho = F.d, P = E/t, U = k.q1.q2/d<sup>2</sup>, U = k.q/d, etc.</li> <li>- Campos de entrada numéricos.</li> <li>- Exibição da fórmula usada e resultado.</li> <li>- CSS para destaque visual e organização.</li> </ul>
<b>Conversor de Unidades</b>	Conversor entre unidades de medidas variadas (ex.: cm para m, °C para °F).	<ul style="list-style-type: none"> <li>- Deve conter seletores para categorias: Temperatura, Massa, Volume, Comprimento, Capacidade.</li> <li>- Conversão dinâmica com base na escolha do usuário.</li> <li>- Função JS organizada por tipo de conversão.</li> </ul>

 Tema	 Descrição	 Requisitos Técnicos Mínimos
<b>Conversão de Moedas</b>	Conversor de valores entre moedas diferentes.	<ul style="list-style-type: none"> <li>- Seleção de moeda base e destino (ex.: BRL para USD, EUR para BRL, etc.).</li> <li>- Pode usar valores fixos ou simular taxa de câmbio.</li> <li>- Interface limpa, com flags ou nomes das moedas.</li> </ul>
<b>Cálculo de IMC</b>	Calculadora de Índice de Massa Corporal.	<ul style="list-style-type: none"> <li>- Input: altura (m) e peso (kg).</li> <li>- Output: valor do IMC com faixa/classificação.</li> <li>- Feedback visual (cores, mensagens ou emojis).</li> <li>- Layout com instruções básicas.</li> </ul>

## 4. Fluxo de Trabalho (Git/GitHub)

1. O **repositório principal** (upstream) será mantido pelo [professor](#).
2. **Líder** de cada grupo fará o **fork** (visível no perfil GitHub) e renomeiará para: [repo-GrupoLeo](#), [repo-GrupoJoao](#).
3. Cada integrante dos grupos **clonará** o fork de seu líder.
4. Desenvolver localmente, **commitar** e **push** para o **fork**.
5. Abrir **Pull Request** ao upstream (fork → principal) ou dentro do fork para a branch [main](#) do fork.
6. **Líder** revisa, mergeia e resolve conflitos.
7. Usar **Issues** para reportar bugs e acompanhar progresso.

## 5. Divisão de Grupos e Tarefas

### 5.1 Grupo Leo — Leonardo Augusto Dias Cardoso (Líder)

#### Projeto principal do grupo: Conversor de Unidades

Abrange: Temperatura, Comprimento, Capacidade, Massa e Volume.

Aluno	Subtarefa/Tema
Leonardo Augusto Dias Cardoso	Estrutura geral do HTML e menu de navegação Lógica JS para seleção dinâmica de categorias
Paloma Stefane Lemes Costa Pio	Conversor de Temperatura (Celsius, Fahrenheit, Kelvin)
Carlos Eduardo Moreira Dias	Conversor de Comprimento (m, cm, mm, km, etc.)
Rayssa Vitória da Silva Martins	Conversor de Volume e Capacidade (litros, ml, etc.)
Mayk Leandro Lage Martins	Conversor de Massa (kg, g, mg, tonelada, etc.)
Gleiciele de Souza	Validação dos dados de entrada com JS
Soraia Teodoro Caetano	Estilização CSS geral (cores, fontes, responsividade)

Aluno	Subtarefa/Tema
Camila Júlia Machado dos Santos	Ícones, layout com Flexbox/Grid, organização visual

## 5.2 Grupo João — João Lucas Teixeira Moreira (Líder)

**Projeto principal do grupo: Fórmulas e Cálculos Matemáticos/Físicos**

Aluno	Subtarefa/Tema
João Lucas Teixeira Moreira	Estrutura HTML base com navegação entre fórmulas
Renato José Duarte Monteiro	Cálculo de IMC com classificação e resultado dinâmico
Davi Moraes	Fórmulas de Força e Aceleração
Melissa Jolly Nepomuceno Nascimento	Fórmulas de Velocidade e Distância
Ian Pedro Ferreira Borges	Fórmulas de Trabalho e Potência
Tales Francisco de Moura Martins	Lei de Ohm e Potência Elétrica
Vitor Cezar Moreira	Força Elétrica e Potencial Elétrico
Yuri Luiz Costa	Conversão de moedas (base fixa + API opcional ou simulação local)
Marco Antonio Oliveira Figueiredo	Custo de Combustível (valor/litro e distância rodada)
Lumara Joana Leite Alves	Estilização geral + organização visual (responsividade, feedbacks visuais)

## 6. Demanda Final: Calculo de Custo Benefício em Comparação ao Item (Item A x Item B)

Para a **entrega final**, ambos os grupos devem integrar suas calculadoras de forma colaborativa, comparando dois itens lado a lado:

### 1. Objetivo

- Implementar uma interface que, além de realizar o cálculo de custo-benefício individual de cada item, permita ao usuário selecionar **Item A** e **Item B** para comparação direta.
- Destacar visualmente qual dos dois itens apresenta melhor custo por unidade.

### 2. Requisitos Funcionais

- Exibir uma tabela com pelo menos duas colunas: **Item A** e **Item B**, cada uma contendo **Nome**, **Preço**, **Quantidade**, **Tamanho** e **Custo por Unidade**.
- Botão **Comparar** que, ao ser acionado, realça o item com menor custo por unidade.
- Inclusão de resumo textual abaixo da tabela: "O Item **A/B** apresenta melhor custo-benefício: **[Item X]** com **R\$ Y** por unidade."

### 3. Requisitos Técnicos

- Reutilizar os componentes de formulário e tabela desenvolvidos nas demandas anteriores.
- Lógica de comparação implementada em `beneficio.js` (ou arquivo equivalente), exportando função `compareItems(itemA, itemB)`.
- Estilos de destaque via classe CSS `.highlight-compare`.

#### 4. Fluxo de Trabalho Colaborativo

- Cada grupo desenvolve sua própria versão de comparação em um branch específico:
  - `grupo-leo/feature/comparacao-beneficio`
  - `grupo-joao/feature/comparacao-beneficio`
- Após implementação, abrir Pull Request do branch de comparação para a branch `main` do fork.
- Fazer Code Review cruzado: membros de Grupo Leo revisam **Pull Request** de Grupo João e vice-versa, garantindo qualidade.

#### Observações:

- Cada integrante deve **atualizar** sua branch frequentemente e abrir **Pull Request** para revisão.
- Revisões devem ocorrer **antes de merges** e o líder integrará as partes no upstream.
- Em caso de dúvidas, abrir **issue** direcionada ao líder do grupo.

## 7. Da Avaliação

### 7.1 Critérios das Demandas Iniciais

(Avaliação de cada uma das sete calculadoras individuais: Descontos, Custo de Combustível, Custo por Medida, Força/Aceleração, Conversor de Unidades, Conversão de Moedas e IMC)

Critério	Peso	Descrição
<b>Compleitude Funcional</b>	30%	- Todas as entradas estão presentes e validadas. - Resultados corretos em cenários de teste.
<b>Qualidade de Código</b>	25%	- Estrutura modular e legível. - Aderência às convenções (ES6, semantic HTML/CSS).
<b>Interface &amp; Acessibilidade</b>	20%	- Layout responsivo. - Uso correto de <code>&lt;label&gt;</code> , <code>aria-live</code> , foco e contraste de cores.
<b>Documentação &amp; Commits</b>	15%	- Mensagens de commit claras (Conventional Commits). - README e comentários nos scripts.
<b>Colaboração Git/GitHub</b>	10%	- Branches específicas por tarefa. - PRs revisados, issues abertas e merges bem conduzidos.

### 7.2 Critérios da Demanda Final

(Avaliação da entrega de comparação Item A × Item B)

Critério	Peso	Descrição
----------	------	-----------

Critério	Peso	Descrição
Funcionalidade & Usabilidade	40%	- Interface sem erros integra ambas calculadoras. - Destaque visual e resumo textual corretos ao comparar.
Código & Arquitetura	30%	- Função <code>compareItems(itemA, itemB)</code> clara e reutilizável. - Módulos bem separados (common.js, beneficio.js).
Estilo & Acessibilidade	15%	- Destaque ( <code>.highlight-compare</code> ) com contraste adequado. - Botões e tabelas com <code>aria-label</code> , foco nítido.
Documentação & Processo	15%	- Instruções de teste no README. - Revisão cruzada de PRs conforme fluxo definido.

#### Observação de Peso Final

- As demandas iniciais representam **80%** da nota final.
- A demanda final representa **20%** da nota final.