

# Projeto Prático: Simulador de Preço de Gas na Polkadot

## Objetivo:

Desenvolver uma aplicação web interativa que permita ao usuário calcular o custo de "gas" (taxa de transação) para diferentes valores e níveis de complexidade de transações na rede Polkadot. A interface será simples e intuitiva, composta por um formulário onde o usuário poderá inserir os dados da transação e obter o custo estimado do gas em tempo real. Além disso, sua missão será implementar os **desafios** e **melhorias** propostos, elevando a funcionalidade da aplicação e tornando-a ainda mais completa e dinâmica.

---

## Passo a Passo:

### 1. Criação da Estrutura HTML

#### Instruções:

- Crie um arquivo chamado index.html.
- No arquivo HTML, crie a estrutura básica da página, incluindo um formulário com os seguintes campos:
  - Valor da transação (em DOT)
  - Complexidade da transação (Escolha entre "baixa", "média" e "alta")
  - Botão para calcular o preço do gas
  - Um campo para exibir o valor estimado do gas

#### Estrutura básica do HTML:

```
<!DOCTYPE html>
```

```
<html lang="pt-BR">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title>Simulador de Preço de Gas - Polkadot</title>
```

```
<link rel="stylesheet" href="styles.css">
```

```
</head>
```

```
<body>
```

```
<h1>Simulador de Preço de Gas - Polkadot</h1>
```

```
<form id="gas-form">
```

```
<label for="valor-transacao">Valor da Transação (DOT):</label>

<input type="number" id="valor-transacao" required>

<label for="complexidade">Complexidade da Transação:</label>

<select id="complexidade" required>

  <option value="baixa">Baixa</option>

  <option value="media">Média</option>

  <option value="alta">Alta</option>

</select>

<button type="button" id="calcular">Calcular Preço do Gas</button>

</form>

<div id="resultado"></div>

<script src="script.js"></script>

</body>

</html>
```

---

## 2. Criação do Estilo com CSS

Vamos adicionar um estilo básico para melhorar a aparência do formulário.

### Instruções:

- Crie um arquivo chamado styles.css.
- Aplique estilos para centralizar o formulário e adicionar uma aparência moderna.

### Estilo básico:

```
body {

  font-family: Arial, sans-serif;

  background-color: #f0f0f0;

  text-align: center;

  padding: 20px;

}
```

```
h1 {  
  color: #333;  
}
```

```
form {  
  background-color: #fff;  
  padding: 20px;  
  border-radius: 10px;  
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
  max-width: 400px;  
  margin: 0 auto;  
}
```

```
label {  
  display: block;  
  margin-bottom: 8px;  
  text-align: left;  
}
```

```
input, select {  
  width: 100%;  
  padding: 10px;  
  margin-bottom: 20px;  
  border: 1px solid #ddd;  
  border-radius: 5px;  
}
```

```
button {  
  background-color: #007BFF;  
  color: white;
```

```
padding: 10px 20px;

border: none;

border-radius: 5px;

cursor: pointer;
}

button:hover {

  background-color: #0056b3;
}

#resultado {

  margin-top: 20px;

  font-size: 18px;

  font-weight: bold;
}
```

---

### 3. Adicionando Interatividade com JavaScript

Agora, vamos adicionar a lógica para calcular o custo estimado do gas com base no valor da transação e na complexidade selecionada.

#### Instruções:

- Crie um arquivo chamado script.js.
- O JavaScript vai capturar os dados do formulário e calcular o preço do gas com base em uma fórmula simples. A complexidade influencia diretamente no preço final.

#### Código JavaScript:

```
document.getElementById('calcular').addEventListener('click', function() {

  // Captura os valores inseridos pelo usuário

  let valorTransacao = parseFloat(document.getElementById('valor-transacao').value);

  let complexidade = document.getElementById('complexidade').value;

  let precoGas;

  // Definição de preço do gas por complexidade

  switch (complexidade) {
```

```
case 'baixa':
    precoGas = 0.01;
    break;
case 'media':
    precoGas = 0.05;
    break;
case 'alta':
    precoGas = 0.1;
    break;
}

// Calcula o custo total do gas
let custoGas = valorTransacao * precoGas;

// Exibe o resultado
if (valorTransacao > 0) {
    document.getElementById('resultado').innerHTML = `
        <p>Valor da Transação: ${valorTransacao} DOT</p>
        <p>Complexidade: ${complexidade.charAt(0).toUpperCase() +
complexidade.slice(1)}</p>
        <p><strong>Custo Estimado do Gas: ${custoGas.toFixed(2)} DOT</strong></p>
    `;
} else {
    document.getElementById('resultado').innerHTML = '<p style="color:red;">Por favor, insira
um valor de transação válido.</p>';
}
});
```

---

#### Funcionalidades do Projeto:

- O formulário permite ao usuário inserir o valor da transação em DOT e selecionar a complexidade da transação.

- O sistema calcula o preço estimado do gas com base em uma regra simples: transações de baixa complexidade têm uma taxa menor, enquanto transações de alta complexidade têm um custo de gas maior.
  - O resultado é exibido na tela com o valor da transação, a complexidade escolhida e o custo total do gas.
- 

#### 4. Testando o Projeto

- Execute o projeto no seu navegador, abrindo o arquivo index.html com o Visual Studio Code.
  - Teste diferentes valores de transação e complexidade para ver o cálculo do gas.
  - Certifique-se de que o layout e as funcionalidades estejam corretos.
- 

### 5. Melhorias e Desafios :

#### 1. Adicionar Validação para Valores de Transação:

Adicione uma validação que garanta que o valor da transação seja maior que zero. Caso contrário, exiba uma mensagem de erro ao usuário.

#### 2. Adicionar Conversão de Moedas:

Permita que o usuário selecione uma moeda de conversão (por exemplo, USD ou EUR) e veja o valor estimado do gas na moeda escolhida, além de DOT.

#### 3. Adicionar um Contador de Transações:

Implemente um contador que exiba quantas transações o usuário já simulou durante a sessão. A cada simulação, o contador deve ser incrementado e exibido no topo da página.

#### 4. Adicionar Temporizador Para Limpar o Histórico de Transações:

Implemente um temporizador que limpa o histórico de transações automaticamente após um determinado período (por exemplo, 1 minuto), simulando o comportamento de expiração de transações temporárias em redes blockchain.

#### 5. Adicionar Histórico de Transações Simuladas:

Mantenha um registro de todas as transações simuladas e exiba o histórico na página, mostrando informações como valor da transação, complexidade e custo do gas.

### ATENÇÃO!!!

O caminho do aprendizado nem sempre é fácil, mas é exatamente esse tipo de desafio que transforma você em um desenvolvedor de verdade!



A cada linha de código que você escreve, a cada problema que resolve, você está mais perto de dominar as habilidades que vão te destacar no mercado Web3.

**Lembre-se:** Praticar é o que faz a diferença! Ao construir essa aplicação e enfrentar os desafios propostos, você não está apenas seguindo um exercício, está criando algo que reflete o futuro da tecnologia. Cada melhoria que você implementar é uma demonstração do seu crescimento e da sua capacidade de inovar.

Seja persistente, explore novas possibilidades e, acima de tudo, divirta-se no processo. Aprender é uma jornada, e você está no caminho certo.

**Acredite no seu potencial e siga em frente, porque o sucesso está logo à frente!**

Estamos aqui para te apoiar, então vamos lá — **mãos no código e rumo ao próximo nível!**

**Vamos JUNTOS!!!**

**Prof: Rodrigo Moreira e equipe CÓDIGO BRAZUCA**