Projeto de Tokens Personalizados

- Script de interação para testar funcionalidades
- Configuração para redes locais e testnets
- Sistema de logging e monitoramento
- Relatórios de gas e cobertura

🚀 Como usar:

1. Configurar o ambiente:

```bash
npm install
cp .env.example .env
# Editar .env com suas configurações
```

2. Testar localmente:

```bash
npm run compile
npm test
npm run node  # Em um terminal
npm run deploy:localhost  # Em outro terminal
```

Copiar

3. Interagir com os contratos:

```bash
npx hardhat run scripts/interact.js --network localhost info
npx hardhat run scripts/interact.js --network localhost full-tes
```

📊 Principais funcionalidades implementadas:

- **Sistema de Taxa**: 1% deduzido automaticamente e enviado para tax receiver
- **Controle de Propriedade**: Owner pode mintar, pausar e configurar
- **Segurança**: Pausable para emergências, queima controlada
- **Otimização**:

Esta resposta foi pausada porque Claude atingiu seu limite máximo de caracteres para uma mensagem. Clique em continuar para que Claude prossiga.

Continuar

Tentar novamente

O Claude pode cometer erros. Confira sempre as respostas.

Você está sem mensagens gratuitas até 15:00

Fazer upgrade do plano

OBRIGADO.!

Claude Sonnet 4

Copiar

Publicar

```
const { ethers } = require("hardhat");

async function main() {
  console.log("🔧 Script de Interação com os Tokens SPB e BPS\n");

  // Endereços dos contratos (substitua pelos endereços reais após depl
  const SPB_ADDRESS = "0x..."; // Substitua pelo endereço real do SPB
  const BPS_ADDRESS = "0x..."; // Substitua pelo endereço real do BPS

  if (SPB_ADDRESS === "0x..." || BPS_ADDRESS === "0x...") {
    console.log("❌ Por favor, atualize os endereços dos contratos no s
    console.log("Execute primeiro: npm run deploy");
    return;
  }

  // Obter contas
  const [owner, user1, user2, taxReceiver] = await ethers.getSigners();

  console.log("📋 Contas disponíveis:");
  console.log("Owner:", owner.address);
  console.log("User1:", user1.address);
  console.log("User2:", user2.address);
  console.log("Tax Receiver:", taxReceiver.address);
  console.log();

  // Conectar aos contratos
  const SPBToken = await ethers.getContractFactory("SPBToken");
  const BPSToken = await ethers.getContractFactory("BPSToken");

  const spbToken = SPBToken.attach(SPB_ADDRESS);
  const bpsToken = BPSToken.attach(BPS_ADDRESS);

  try {
    // Menu de operações
    const operation = process.argv[2];

    switch (operation) {
      case "info":
        await showTokenInfo(spbToken, bpsToken, owner);
        break;

      case "transfer":
        await testTransfer(spbToken, owner, user1, taxReceiver);
        break;

      case "mint":
        await testMint(spbToken, owner, user1);
        break;

      case "pause":
        await testPause(spbToken, owner);
        break;

      case "burn":
        await testBurn(spbToken, owner);
```