

Guia de Teste da Aplicação de Tokenização com Contratos Foundry

Este guia fornece instruções detalhadas para testar a aplicação de tokenização utilizando os novos contratos desenvolvidos com Foundry. Siga os passos abaixo para configurar o ambiente de desenvolvimento local e interagir com as principais funcionalidades da aplicação.

Índice

1. Iniciando o Ambiente de Desenvolvimento Local (Anvil)
2. Implantando os Contratos no Ambiente Local
3. Configurando a Aplicação Next.js
4. Interagindo com os Contratos pela Interface
5. Testando as Principais Funcionalidades

1. Iniciando o Ambiente de Desenvolvimento Local (Anvil)

O Anvil é uma blockchain Ethereum local que vem com o Foundry e permite testar contratos inteligentes em um ambiente controlado.

Instalação do Foundry (caso ainda não tenha instalado)

```
curl -L https://foundry.paradigm.xyz | bash
foundryup
```

Iniciando o Anvil

Para iniciar o Anvil com configurações padrão:

```
anvil
```

Você verá uma saída semelhante a esta:

```

      ( _ | |
    / _ \ | | ' _ \ \ \ \ / / | | | |
    | ( _ | | | | | \ \ v / | | | |
    \ _ , | | | | \ \ / | | | |

```

Iniciando HTTP e WebSocket JSON-RPC server em 127.0.0.1:8545

Contas disponíveis

=====

- (0) 0xf39fd6e51aad88f6f4ce6ab8827279cfff92266 (10000 ETH)
- (1) 0x70997970c51812dc3a010c7d01b50e0d17dc79c8 (10000 ETH)

...

Por padrão, o Anvil: - Executa em `http://127.0.0.1:8545` - Fornece 10 contas pré-financiadas com 10.000 ETH cada - Usa o chain ID 31337

Para configurações personalizadas, você pode usar flags adicionais:

```
# Exemplo com configurações personalizadas
anvil --chain-id 1337 --block-time 5
```

Nota: Mantenha esta janela do terminal aberta enquanto estiver testando. Abra um novo terminal para os próximos passos.

Anvil em execução

2. Implantando os Contratos no Ambiente Local

Agora vamos implantar os contratos inteligentes no ambiente Anvil local.

Navegando até o Diretório do Projeto

```
cd /caminho/para/seu/projeto-tokenizacao
```

Compilando os Contratos

```
forge build
```

Implantando os Contratos

Vamos usar o script de implantação para implantar todos os contratos necessários:

```
forge script script/Deploy.s.sol --rpc-url http://127.0.0.1:8545 --broadcast --private-key 0
```

Nota: A chave privada acima corresponde à primeira conta fornecida pelo Anvil. Não use esta chave em ambientes de produção!

Você verá uma saída detalhando os contratos implantados e seus endereços:

Script executado com sucesso.

Contratos implantados:

TokenFactory: 0x5FbDB2315678afecb367f032d93F642f64180aa3

Marketplace: 0xe7f1725E7734CE288F8367e1Bb143E90bb3F0512

WaitingList: 0x9fE46736679d2D9a65F0992F2272dE9f3c7fa6e0

...

Importante: Anote os endereços dos contratos implantados, pois você precisará deles para configurar a aplicação Next.js.

Implantação dos contratos

3. Configurando a Aplicação Next.js para Usar os Contratos Locais

Agora vamos configurar a aplicação Next.js para se conectar aos contratos implantados localmente.

Navegando até o Diretório da Aplicação

```
cd /caminho/para/sua/aplicacao-nextjs
```

Configurando as Variáveis de Ambiente

Crie ou edite o arquivo `.env.local` na raiz do projeto:

```
cp .env.example .env.local  
nano .env.local
```

Atualize as seguintes variáveis com os endereços dos contratos que você anotou anteriormente:

```
# Ambiente de desenvolvimento local (Anvil)  
NEXT_PUBLIC_CHAIN_ID=31337  
NEXT_PUBLIC_RPC_URL=http://127.0.0.1:8545  
NEXT_PUBLIC_TOKEN_FACTORY_ADDRESS=0x5FbDB2315678afecb367f032d93F642f64180aa3  
NEXT_PUBLIC_MARKETPLACE_ADDRESS=0xe7f1725E7734CE288F8367e1Bb143E90bb3F0512  
NEXT_PUBLIC_WAITING_LIST_ADDRESS=0x9fE46736679d2D9a65F0992F2272dE9f3c7fa6e0
```

Iniciando a Aplicação Next.js

```
npm install  
npm run dev
```

A aplicação estará disponível em `http://localhost:3000`.

Aplicação Next.js em execução

4. Interagindo com os Contratos Através da Interface da Aplicação

Agora que a aplicação está configurada e em execução, vamos conectar uma carteira e interagir com os contratos.

Conectando sua Carteira

1. Abra a aplicação em `http://localhost:3000`
2. Clique no botão “Conectar Carteira” no canto superior direito
3. Selecione “MetaMask” ou outra carteira compatível

Configurando a MetaMask para o Anvil Local Se você estiver usando a MetaMask, precisará adicionar a rede Anvil:

1. Abra a extensão MetaMask
2. Clique no seletor de rede no topo
3. Selecione “Adicionar Rede”
4. Clique em “Adicionar Rede Manualmente”
5. Preencha os seguintes detalhes:
 - Nome da Rede: Anvil Local
 - URL RPC: `http://127.0.0.1:8545`
 - ID da Cadeia: 31337
 - Símbolo da Moeda: ETH
6. Clique em “Salvar”

Importando uma Conta do Anvil para a MetaMask Para usar uma das contas pré-financiadas do Anvil:

1. Copie a chave privada de uma das contas listadas quando você iniciou o Anvil
2. Na MetaMask, clique no ícone de conta no canto superior direito
3. Selecione “Importar Conta”
4. Cole a chave privada e clique em “Importar”

Agora você deve estar conectado à aplicação com uma conta que possui ETH suficiente para testar todas as funcionalidades.

Carteira conectada

5. Testando as Principais Funcionalidades

Vamos testar as principais funcionalidades da aplicação de tokenização.

5.1. Tokenização de Ativos

Criando um Novo Token

1. Navegue até a seção “Criar Token” ou “Tokenizar Ativo”
2. Preencha o formulário com os detalhes do ativo:
 - Nome do Token: “Imóvel Teste”
 - Símbolo do Token: “IMOV”
 - Quantidade Total: 1000
 - Preço por Token: 0.1 ETH
 - Descrição: “Tokenização de um imóvel para teste”
 - Imagem do Ativo: Faça upload de uma imagem
3. Clique em “Criar Token”
4. Confirme a transação na sua carteira

Criação de token

Verificando o Token Criado

1. Navegue até a seção “Meus Tokens” ou “Portfólio”
2. Você deverá ver o token recém-criado na lista
3. Clique no token para ver seus detalhes

Detalhes do token

5.2. Marketplace

Listando um Token para Venda

1. Na página de detalhes do token, clique em “Listar para Venda”
2. Defina a quantidade e o preço por token
3. Clique em “Confirmar Listagem”
4. Aprove a transação na sua carteira

Listagem no marketplace

Comprando um Token Para testar a compra, você precisará usar uma conta diferente:

1. Desconecte a conta atual
2. Conecte uma conta diferente seguindo os passos da seção 4
3. Navegue até o “Marketplace”
4. Encontre o token listado e clique em “Comprar”
5. Defina a quantidade que deseja comprar
6. Clique em “Confirmar Compra”
7. Aprove a transação na sua carteira

Compra de token

5.3. Lista de Espera

Entrando na Lista de Espera

1. Navegue até a seção “Lista de Espera” ou “Próximos Lançamentos”
2. Encontre um token que ainda não foi lançado
3. Clique em “Entrar na Lista de Espera”
4. Preencha as informações solicitadas (se houver)
5. Confirme a transação na sua carteira

Lista de espera

Verificando sua Posição na Lista

1. Navegue até “Minha Conta” ou “Perfil”
2. Vá para a seção “Minhas Listas de Espera”
3. Verifique sua posição nas listas em que você se inscreveu

Posição na lista de espera

Solução de Problemas Comuns

Erro de Conexão com o Anvil

Se você encontrar erros de conexão:

1. Verifique se o Anvil está em execução
2. Confirme se a URL RPC está correta nas configurações da aplicação
3. Reinicie o Anvil se necessário

Transações Falhando

Se as transações estiverem falhando:

1. Verifique se você tem ETH suficiente na sua conta
2. Confirme se os endereços dos contratos estão corretos no arquivo `.env.local`
3. Verifique os logs do Anvil para identificar o erro específico

Reiniciando o Ambiente

Se você precisar reiniciar todo o ambiente de teste:

1. Encerre o Anvil (Ctrl+C no terminal onde ele está em execução)
2. Inicie o Anvil novamente
3. Reimplante os contratos
4. Atualize os endereços dos contratos no arquivo `.env.local`
5. Reinicie a aplicação Next.js

Conclusão

Parabéns! Você configurou com sucesso o ambiente de desenvolvimento local e testou as principais funcionalidades da aplicação de tokenização com os novos contratos Foundry.

Este ambiente de teste local permite que você experimente e desenvolva novos recursos sem gastar ETH real ou interagir com redes de teste públicas.

Para qualquer dúvida adicional ou problemas, consulte a documentação do Foundry ou entre em contato com a equipe de desenvolvimento.