

Aula 023 - Prática sobre *Ethereum: Ferramentas de Desenvolvimento e Frameworks*

Prof. Rogério Aparecido Gonçalves *Universidade Tecnológica Federal do Paraná (UTFPR)*

A proposta desta aula prática é instalarmos as ferramentas e *Frameworks* para o desenvolvimento e implantação de Contratos Inteligentes. Utilizaremos como padrão o Compilador para Solidity, o solc e as bibliotecas e *Frameworks* apresentados na aula anterior.

Sumário

1	Leitura do Capítulo 14: <i>Development Tools and Frameworks</i>	1
2	Instalação das Ferramentas	1
3	Próximas Aulas	3
3.1	Próximas Aulas	3
4	Referências	4
4.1	Word Cloud	4
4.2	Referências	4

1 Leitura do Capítulo 14: *Development Tools and Frameworks*

1. Faça a leitura do [Capítulo 14: *Development Tools and Frameworks*](#)

2 Instalação das Ferramentas

2. Instale o Compilador Solidity (solc). O solc converte código de alto nível escrito na linguagem Solidity para *bytecode* da *Ethereum Virtual Machine (EVM)*.

Para distribuições Ubuntu ou derivados do Debian:

```
1 $ sudo apt-get install solc
```

Outras distribuições como o Manjaro Linux, instale o pacote solidity:

```
1 $ sudo pacman -S solidity
```

Feita a instalação, para verificar a versão instalada execute o comando:

```
1 $ solc --version
2 solc, the solidity compiler commandline interface
3 Version: 0.8.17+commit.8df45f5f.Linux.g++
```

Para verificar o funcionamento e algumas funcionalidades vamos criar um contrato simples, com o nome `Addition.sol` e com o seguinte conteúdo:

```

1 pragma solidity ^0.8.17;
2
3 contract Addition {
4     uint8 x;
5
6     function addx(uint8 y, uint8 z ) public {
7         x = y + z;
8     }
9     function retrievex() view public returns (uint8) {
10         return x;
11     }
12 }

```

Verificando o formato binário do contrato:

```

1 $ solc --bin Addition.sol
2 Warning: SPDX license identifier not provided in source file. Before publishing,
   consider adding a comment containing "SPDX-License-Identifier: <SPDX-License>" to
   each source file. Use "SPDX-License-Identifier: UNLICENSED" for non-open-source
   code. Please see https://spdx.org for more information.
3 --> Addition.sol
4
5
6 ===== Addition.sol:Addition =====
7 Binary:
8 608060405234801561001057600080fd5b506101f6806100206000396000f3fe608060405234801561001057
9 600080fd5b50600436106100365760003560e01c806336718d801461003b578063ac04e0a014610057575b60
10 0080fd5b610055600480360381019061005091906100f2565b610075565b005b61005f61009e565b60405161
11 006c9190610141565b60405180910390f35b8082610081919061018b565b6000806101000a81548160ff0219
12 16908360ff1602179055505050565b60008060009054906101000a900460ff16905090565b600080fd5b6000
13 60ff82169050919050565b6100cf816100b9565b81146100da57600080fd5b50565b6000813590506100ec81
14 6100c6565b92915050565b60008060408385031215610109576101086100b4565b5b60006101178582860161
15 00dd565b9250506020610128858286016100dd565b9150509250929050565b61013b816100b9565b82525050
16 565b60006020820190506101566000830184610132565b92915050565b7f4e487b71000000000000000000
17 0000000000000000000000000000000000600052601160045260246000fd5b6000610196826100b9565b91
18 506101a1836100b9565b9250828201905060ff8111156101ba576101b961015c565b5b9291505056fea26469
19 70667358221220e0ec16eaf684603f4f7c74f327a27e4a1a981dfac0cb258479ffe452abda2e4964736f6c63
20 430008110033

```

Estimando o gas. Como uma taxa de gas é cobrada para cada operação que o EVM executa, é uma boa prática estimar gas antes de implantar um contrato em uma rede ativa.

```

1 $ solc --gas Addition.sol
2 ===== Addition.sol:Addition =====
3 Gas estimation:
4 construction:
5     147 + 100400 = 100547
6 external:
7     addx(uint8,uint8): infinite
8     retrievex(): 2479

```

Gerando a ABI do contrato. A *Application Binary Interface (ABI)* é uma forma padrão de interagir com os contratos.

```
1 $ solc --abi Addition.sol
2 ===== Addition.sol:Addition =====
3 Contract JSON ABI
4 [{"inputs":[{"internalType":"uint8","name":"y","type":"uint8"},{"internalType":"uint8",
5 "name":"z","type":"uint8"}],"name":"addx","outputs":[],"stateMutability":"nonpayable",
6 "type":"function"},{"inputs":[],"name":"retrievex","outputs":[{"internalType":"uint8",
7 "name":"","type":"uint8"}],"stateMutability":"view","type":"function"}]
```

Compilando um contrato:

```
1 $ solc --bin --abi -o bin Addition.sol
2 Compiler run successful. Artifact(s) can be found in directory "bin".
```

Se erros ocorrerem serão mostrados no console, caso contrário o compilador irá mostrar uma mensagem de sucesso. Serão gerados os arquivos no diretório bin:

- Addition.abi: Contém a ABI do contrato no formato JSON.
- Addition.bin: Contém a representação binária do código do contrato.

```
1 $ cat bin/Addition.bin
2 608060405234801561001057600080fd5b506101f6806100206000396000f3fe608060405234801561001057
3 600080fd5b50600436106100365760003560e01c806336718d801461003b578063ac04e0a014610057575b60
4 0080fd5b610055600480360381019061005091906100f2565b610075565b005b61005f61009e565b60405161
5 006c9190610141565b60405180910390f35b8082610081919061018b565b6000806101000a81548160ff0219
6 16908360ff1602179055505050565b60008060009054906101000a900460ff16905090565b600080fd5b6000
7 60ff82169050919050565b6100cf816100b9565b81146100da57600080fd5b50565b6000813590506100ec81
8 6100c6565b92915050565b60008060408385031215610109576101086100b4565b5b60006101178582860161
9 00dd565b9250506020610128858286016100dd565b9150509250929050565b61013b816100b9565b82525050
10 565b60006020820190506101566000830184610132565b92915050565b7f4e487b7100000000000000000000
11 00000000000000000000000000000000600052601160045260246000fd5b6000610196826100b9565b91
12 506101a1836100b9565b9250828201905060ff8111156101ba576101b961015c565b5b9291505056fea26469
13 70667358221220e0ec16eaf684603f4f7c74f327a27e4a1a981dfac0cb258479ffe452abda2e4964736f6c63
14 430008110033
15
16 $ cat bin/Addition.abi
17 [{"inputs":[{"internalType":"uint8","name":"y","type":"uint8"},{"internalType":"uint8",
18 "name":"z","type":"uint8"}],"name":"addx","outputs":[],"stateMutability":"nonpayable",
19 "type":"function"},{"inputs":[],"name":"retrievex","outputs":[{"internalType":"uint8",
20 "name":"","type":"uint8"}],"stateMutability":"view","type":"function"}]
```

3. Instale as outras ferramentas: Node.js, Ganache e Ganache-CLI, Truffle, Drizzle, Embark e outras ferramentas indicadas no capítulo.

3 Próximas Aulas

3.1 Próximas Aulas

- Ambientes de Desenvolvimento e Ferramentas.

4 Referências

4.1 Word Cloud



4.2 Referências

Imran, Bashir. 2018. *Mastering Blockchain : Distributed Ledger Technology, Decentralization, and Smart Contracts Explained*, 2nd Edition. Packt Publishing. <https://search.ebscohost.com/login.aspx?direct=true&db=e000xww&AN=1789486&lang=pt-br&site=eds-live&scope=site>.