



Reactor

S ã O P A U L O

Construindo Contratos Inteligentes
em Ethereum usando Solidity

Código do evento: #7945

Solange Gueiros

Blockchain developer



Especialista em Blockchain (arquitetura e desenvolvimento), com foco em Bitcoin, Ethereum e Smart Contracts, faz palestras, cursos e consultoria.

Trabalha com tecnologia há mais de 20 anos.

Palestrante em diversas conferências no Brasil e no mundo.

Ganhou prêmios nos hackathons de New York, Berlim e Denver.

Em 2020 está na lista top 50 do Cointelegraph Brazil.

<https://www.linkedin.com/in/solangegueiros/>

Blog: <https://solange.dev/>



meetup.com/Microsoft-Reactor-Sao-Paulo/


Agenda

15:30	Solidity fundamentals	Fundamentos de Solidity
16:00	TaskManager project: Learn Solidity building the smart contract	Projeto TaskManager: Aprenda Solidity construindo o smart contract
16:45	Q&A	Perguntas e respostas
17:00	Event end	Fim do evento

Overview

1. Instalação de pré requisitos;
2. Configurar um projeto utilizando Truffle;
3. Arquitetura do projeto;
4. Criar um smart contract;
5. Conceitos de Solidity;
6. Compilar e publicar um smart contract;
7. Interagir com o smart contract através do Truffle console.
8. Próximos passos - upgrades;
9. Publicando em Blockchain;
10. Considerações finais.

Requirements Pré-requisitos


 meetup.com/Microsoft-Reactor-Sao-Paulo/

Pré-requisitos

- Git
 - <https://gitforwindows.org/>
- Node.js e NPM (Node Package Manager)
 - <https://nodejs.org/en/>
- Visual Studio Code (VSCode)
 - <https://code.visualstudio.com/>
- Extensão VSCode para a linguagem Solidity
 - Menu View -> Extensions -> extensão Solidity do Juan Blanco
- Truffle
 - <https://www.trufflesuite.com/truffle>

Solidity Fundamentals

Fundamentos de Solidity

 meetup.com/Microsoft-Reactor-Sao-Paulo/

Fundamentos de Solidity

- Declaração de variáveis de estado
- Tipos de dados: Enum
- Tipos de dados: Struct
- Arrays
- Mappings

Fundamentos de Solidity

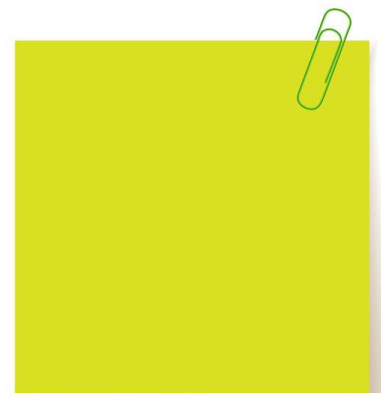
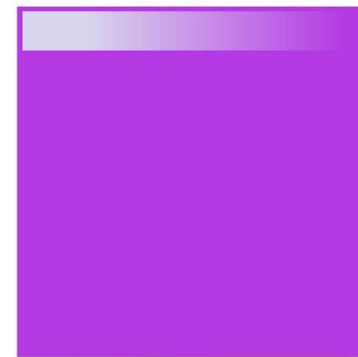
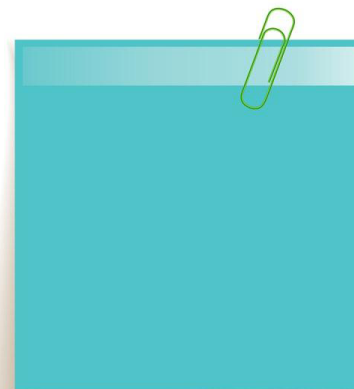
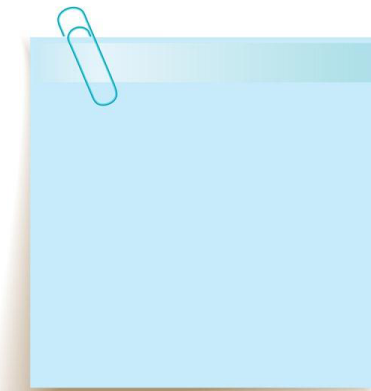
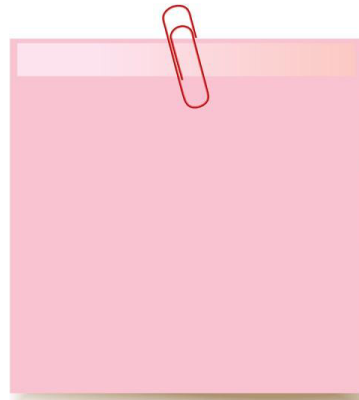
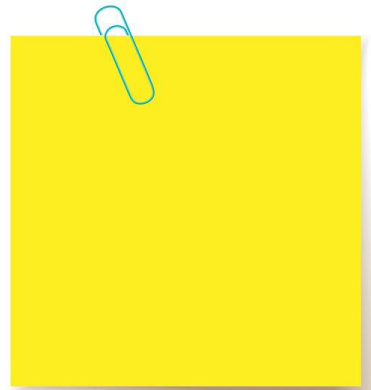
- Versão do compilador
- Definição de um smart contract
- Eventos
- Modificadores
- Construtor
- Funções
- Tratamento de erros

Project: Task Manager

Projeto: Gerenciador de tarefas

TaskManager

Gerenciador de tarefas



Uma tarefa

- dono da tarefa, representado pelo seu endereço
- name
- phase
- priority

Fases de uma tarefa

- ToDo,
- InProgress,
- Done,
- Blocked,
- Review,
- Postponed,
- Canceled


Regras de negócio

- Cada pessoa, representada por sua conta / endereço Ethereum, pode ter várias tarefas.
- Quando uma tarefa for adicionada, um evento será emitido, ou seja qualquer sistema que esteja monitorando o Blockchain pode acompanhar a criação de tarefas.
- Dado um índice da tarefa, queremos saber seus detalhes: dono, nome, fase e prioridade.

Regras de negócio

- Queremos ter uma lista com os índices de todas as nossas tarefas.
- Cada pessoa só pode ver a sua lista de tarefas.
- Qualquer pessoa pode adicionar uma tarefa para si mesma, mas não pode adicionar para outra pessoa.
- Depois que uma tarefa for criada, a única coisa que pode ser alterada é a sua fase.

Solidity no Projeto Task Manager

 meetup.com/Microsoft-Reactor-Sao-Paulo/

Versão do compilador Solidity

// Qualquer versão a partir da 0.7 (= 0.7.x)

```
pragma solidity ^0.7.0;
```

// Maior que a versão 0.5.5, menor que a versão 0.6.4

```
pragma solidity >0.5.5 <0.6.4;
```

// Apenas a versão 0.5.4

```
pragma solidity 0.5.4;
```

Definindo o smart contract

```
//Define / declara um smart contract  
contract TaskManager {  
  
}
```

Declarando variáveis

```
// Variável do tipo inteiro sem sinal, pública  
uint public nTasks;
```

```
// Variável privada para armazenar um endereço Ethereum  
address private owner;
```

```
// Variável para armazenar qualquer valor ou um texto  
string name;
```

Tipos enumerados

```
//enum TaskPhase
```

```
//ToDo = 0, InProgress = 1, Done = 2, ...
```

```
enum TaskPhase {ToDo, InProgress, Done, Blocked, Review,  
Postponed, Canceled}
```

Estruturas

//contém variáveis de diversos tipos

```
struct TaskStruct {
```

```
    address owner;
```

```
    string name;
```

```
    TaskPhase phase;
```

```
    // Priority 1-5: 1 higher, 5 less important
```

```
    uint priority;
```

```
}
```



meetup.com/Microsoft-Reactor-Sao-Paulo/

Array

```
//array de endereços  
address[] addressesList;
```

```
//array de structs
```

```
//cada posição do array contém
```

```
//uma struct com as informações definidas na struct
```

```
TaskStruct[] private tasks;
```

Mapping

//estrutura do tipo chave => valor

// Cada número (exemplo: CPF), aponta para seu nome
mapping (unit => string) names;

//cada endereço aponta para sua lista de inteiros
mapping (address => uint[]) private myTasks;

Event

//Definição

```
event TaskAdded(address owner, string name,  
                TaskPhase phase, uint priority);
```

//Chamada

```
emit TaskAdded (msg.sender, _name, _phase, _priority);
```


Modifier

//Definição

```
modifier onlyOwner (uint _taskIndex) {  
    if (tasks[_taskIndex].owner == msg.sender) {  
        _;  
    }  
}
```

//Utilização

```
function updatePhase() public onlyOwner(_taskIndex) { }
```

Constructor

//função chamada apenas na publicação do smart contract

//utilizada para inicialização de valores, por exemplo.

```
constructor() public {  
    nTasks = 0;  
    addTask ("Create Task Manager", TaskPhase.Done, 1);  
    addTask ("Create Your first task", TaskPhase.ToDo, 1);  
    addTask ("Clean your house", TaskPhase.ToDo, 5);  
}
```

Function

//Declaração de uma função

// Retorna uma lista de inteiros

```
function listMyTasks() public view returns (uint[] memory)
{
    return myTasks[msg.sender];
}
```

Function

//Recebe um parâmetro de entrada e retorna diversos campos

```
function getTask(uint _taskIndex) public view  
    returns (address owner, string memory name, TaskPhase phase,  
        uint priority) {  
    owner = tasks[_taskIndex].owner;  
    name = tasks[_taskIndex].name;  
    phase = tasks[_taskIndex].phase;  
    priority = tasks[_taskIndex].priority;  
}
```



meetup.com/Microsoft-Reactor-Sao-Paulo/

Function

//Recebe diversos parâmetros na entrada

//Uso de modificador

```
function updatePhase(uint _taskIndex, TaskPhase _phase)
    public onlyOwner(_taskIndex) {

    tasks[_taskIndex].phase = _phase;

}
```

Function

```
function addTask(string memory _name, TaskPhase _phase, uint _priority) public returns (uint index) {  
    require ((_priority >= 1 && _priority <=5), "priority must be between 1 and 5");  
    TaskStruct memory taskAux = TaskStruct ({  
        owner: msg.sender,  
        name: _name,  
        phase: _phase,  
        priority: _priority  
    });  
    index = tasks.push (taskAux) - 1;  
    nTasks ++;  
    myTasks[msg.sender].push(index);  
    emit TaskAdded (msg.sender, _name, _phase, _priority);  
}
```



Tratamento de Erros em Solidity

Assert


Require

Revert

Em AddTask:

```
require ((_priority >= 1 && _priority <=5),  
        "priority must be between 1 and 5");
```

Perguntas e Respostas

 meetup.com/Microsoft-Reactor-Sao-Paulo/



Reactor

S ã O P A U L O

Estamos constantemente nos esforçando para criar excelentes conteúdos e agradeceríamos se você pudesse responder esta rápida pesquisa.

Link pesquisa: <https://aka.ms/Reactor/Survey>

Use o código do evento 7945 no início da pesquisa.

Próximos passos

- Mais informações na estrutura de uma tarefa
 - Tipo de tarefa: pessoal, família, casa, trabalho, escola, etc
 - Data limite para execução da tarefa
- Classificação por cores
- Lista de tarefas por prioridade
- Tarefas para grupos

Material

- <https://github.com/microsoft/ReactorSaoPaulo/tree/main/Workshops/Blockchain>

Junte-se a nós



[meetup.com/Microsoft-Reactor-Sao-Paulo/](https://www.meetup.com/Microsoft-Reactor-Sao-Paulo/)



@MSFTReactor



<http://www.youtube.com/c/MicrosoftReactor>



reactorsaopaulo@microsoft.com



[meetup.com/Microsoft-Reactor-Sao-Paulo/](https://www.meetup.com/Microsoft-Reactor-Sao-Paulo/)



Microsoft Reactor at Distrito AdTech Hub,
Rua Estados Unidos, 1570, Sao Paulo,
Sao Paulo 01412-100

Questions? reactorsaopaulo@microsoft.com

Solange Gueiros



Blog: <https://solange.dev/>

<https://www.linkedin.com/in/solangegueiros/>

Twitter, telegram, facebook, instagram, youtube:
solangegueiros



Reactor

S ã O P A U L O

Muito obrigada!



meetup.com/Microsoft-Reactor-Sao-Paulo/



[@MSFTReactor](https://twitter.com/MSFTReactor)



<http://www.youtube.com/c/MicrosoftReactor>



aka.ms/ReactorEmailSignUp



meetup.com/Microsoft-Reactor-Sao-Paulo/