



Insper

# **Ativos Digitais e Blockchain**

**Ricardo Rocha**  
**Raul Ikeda**

# Objetivo

- Programando SmartContracts
- Interagindo com Contratos

Atenção: se você ainda não instalou os pacotes da aula passada, faça antes de começar essa aula

# My Own Bank<sup>1</sup>

Na aula passada foi demonstrado um contrato que emula um banco.

O contrato realizava as seguintes operações:

- Depósito para conta interna

- Saque para conta externa

- Transferência entre contas internas

<sup>1</sup> Baseado no Tutorial: <https://www.trufflesuite.com/boxes/drizzle>

# #1 Teste: Usando o truffle

1. Criar uma pasta

2. No prompt de comando apontando para essa nova pasta:

> **truffle init**

Colocar o arquivo Bank.sol na pasta .\contracts\

> **truffle develop**

Ele irá criar uma blockchain de testes com 10 contas:

truffle(develop)> **compile**

Fazendo o deploy e um depósito de 10 weis:

truffle(develop)> **let instance = await Bank.new()**

truffle(develop)> **instance.deposit({from: accounts[0], value:10})**

Verificando o saldo:

truffle(develop)> **instance.balance({from: accounts[0]})**

Não parece muito prático!

## #2 Teste: Usando o truffle+ganache

1. Ainda no prompt do slide anterior:

> **truffle compile**

Abrir o Ganache e criar um Workspace na porta **7545**

Fazendo o deploy do contrato diretamente:

> **truffle console**

Repetir os passos anteriores para implantar o contrato

Podemos ver agora no Ganache que o contrato foi implantado na rede local com sucesso!

Mas ainda não parece uma boa interação

# #3 Teste – Projeto Completo

1. Criar uma nova pasta
2. Manter o Ganache aberto
3. No prompt de comando apontando para essa nova pasta:
  - > **truffle unbox raulikeda/AD-BOX**
  - Café rápido, vai demorar um pouquinho
  - > **truffle compile**
  - > **truffle migrate**
  - > **cd app**
  - > **node node\_modules/react-scripts/scripts/start.js**

# Voilà



## Drizzle Examples

Exemplo de como usar o drizzle!

Baseado no box: <https://www.trufflesuite.com/boxes/drizzle>

### Minha Carteira Ethereum

0x0E78F5B483A3F137e046fb863133B563DEdF6dd3

99.947 Ether

### Saldo da Conta no Banco

Exemplo de como pegar um atributo do contrato com a minha própria chave.

#### Saldo Atual:

- **clientID**  
0x00
- **balance**  
0

### Realizando Operações

Abaixo algumas das interações possíveis com o contrato, baseado nos métodos disponíveis.

#### Deposito (1000 wei)

#### Saque

#### Transferência

# App

Agora teste livremente

Acompanhe também os blocos no Ganache

Modifique o endereço no browser para:

- <http://localhost:3000/?acc=1>



# Explicando

1. Como dito, o Ganache cria uma rede de testes
2. Truffle é um “compilador” de Smart Contracts. Você deve “migrar” para fazer a implantação na rede.
3. O contrato de exemplo está na linguagem Solidity.
4. Dentro do exemplo ainda há uma biblioteca chamada Drizzle que permite uma página web interagir com o contrato. Essa página usa React.
5. Na próxima aula começaremos a desenvolver Smart Contracts passo a passo usando o Vyper

# Projeto

1. Em grupos de 4 ou 5, vocês devem formatar um modelo de negócios usando o Blockchain. No máximo 2 pessoas de um mesmo curso.
2. Não pode ser apenas uma cryptomoeda. É preciso ter alguma atividade alvo atrelada ao projeto.
3. Você tem que apresentar a ideia do projeto e implementar usando uma linguagem de programação. Pode-se usar um framework ou infraestrutura em Cloud.
4. Primeiro Deadline: Proposta **detalhada** do projeto com os integrantes do grupo via Blackboard até 19/Mai. Desconto de meio conceito na perda do deadline.
5. Realização de uma apresentação na data da Prova Final.

# Temas Sugeridos

- Election
- Intellectual Property
- Notary
- Real State
- Digital Identity
- Insurance
- Supply Chain Management
- Repurchase Agreement
- Tokenized Ownership
- Patient Data
- Digital Evidence
- Outro tema de livre escolha

# Cronograma

- Separação por mesas para formação dos grupos
- 20min – Discussão dos temas
- 15min – Rotação e discussão
- 15min – Rotação e discussão
- 10min – Fechamento

# Próxima aula

Desenvolvimento de Contatos passo a passo com Vyper