# 

# Universidade Estadual de Santa Cruz

# Departamento de Engenharias e Computação - DEC

# Tópicos Avançados em Computação II -UESC

#### RELATÓRIO

Relatório de Interação e Análise do Contrato Inteligente do Box Pet-Shop: Funcionalidades, Resultados e Perspectivas

Discente: Tauan Neres Silva.

Docente: Jorge Lima de Oliveira Filho  
 Semestre: 2023.1

## Itabuna, julho de 2023

1. **Resumo**
2. **Introdução**

O Truffle Box "pet-shop" é uma seleção de recursos e arquivos pré-configurados fornecidos pela plataforma Truffle para facilitar o desenvolvimento de um aplicativo descentralizado (dapp) voltado para o setor de adoção de animais. O projeto tem como objetivo construir um sistema de rastreamento de adoção para uma loja de animais, especificamente a loja de animais de Pete Scandlon, conhecida como Pete's Pet Shop. A ideia é usar a tecnologia Ethereum e contratos inteligentes para melhorar a eficiência do processo de adoção de animais.

**2.1. Importância dos Contratos Inteligentes**

Os contratos inteligentes desempenham um papel fundamental no contexto desse projeto, pois permitem automatizar e garantir a execução transparente das transações relacionadas à adoção de animais. Ao usar a tecnologia blockchain do Ethereum, os contratos inteligentes oferecem uma solução segura e confiável para associar um endereço Ethereum a um animal que está sendo adotado.

Os contratos inteligentes são programas autoexecutáveis ​​armazenados na blockchain do Ethereum. Eles contêm lógica programável e regras pré-definidas, permitindo que as partes envolvidas interajam diretamente entre si, sem a necessidade de intermediários ou autoridades centrais. No contexto do projeto de rastreamento de adoção de animais, os contratos inteligentes garantem que as transações de adoção sejam imutáveis, transparentes e seguras.

**2.2. Objetivos do Truffle Box e Propósito do Projeto**

O Truffle Box "pet-shop" fornece uma estrutura e um conjunto de recursos pré-configurados para ajudar os desenvolvedores a iniciar o desenvolvimento do dapp de rastreamento de adoção de animais. O objetivo é permitir que os desenvolvedores se concentrem no desenvolvimento dos contratos inteligentes e na lógica do front-end, enquanto fornecem uma base sólida para o projeto.

O propósito do projeto em si é fornecer a Pete's Pet Shop uma solução eficiente para gerenciar suas adoções de animais usando a tecnologia Ethereum. Atualmente, a loja tem espaço para 16 animais em um determinado momento e já possui um banco de dados de animais disponíveis para adoção. O projeto visa criar um dapp que associe um endereço Ethereum a um animal específico para adoção, fornecendo um sistema de rastreamento confiável e transparente.

Esse projeto é direcionado a pessoas com conhecimento básico de Ethereum e contratos inteligentes, juntamente com algum conhecimento de HTML e JavaScript. Ele serve como um tutorial introdutório para o desenvolvimento de dapps e oferece uma oportunidade para os desenvolvedores aplicarem seus conhecimentos em um contexto prático, abordando um problema real de negócios relacionado ao setor de adoção de animais.

## **Material e Métodos (ou Procedimentos Metodológicos)**

3.1. **Materiais**

**3.1.1. TRUFFLE FRAMEWORK**

O Truffle é um framework de desenvolvimento de dapps para Ethereum. Ele fornece uma suíte de ferramentas que simplificam e aceleram o processo de desenvolvimento, incluindo compilação de contratos inteligentes, implantação na blockchain, testes e gerenciamento de dependências.

**3.1.2. TRUFFLE BOX “PET-SHOP”**

O Truffle Box "pet-shop" é um modelo pré-configurado fornecido pelo Truffle Framework. Ele contém uma estrutura inicial para o projeto de rastreamento de adoção de animais, incluindo arquivos de contrato inteligente, front-end e testes básicos. O Truffle Box ajuda os desenvolvedores a começarem rapidamente, fornecendo uma base sólida para a construção do dapp.

**3.1.3. NODE.JS**

O Node.js é um ambiente de tempo de execução JavaScript que permite a execução de código JavaScript fora do navegador. Ele é usado no projeto para executar o servidor local e fornecer suporte para várias ferramentas e dependências.

**3.1.4. VISUAL STUDIO CODE**

IDE *Integrated Development Environment* - Ambiente de Desenvolvimento Integrado, é uma ferramenta usada no desenvolvimento de aplicações, que combina diferentes funcionalidades em uma única interface gráfica do usuário.

Essa ferramenta permite ao desenvolvedor digitar todas as codificações e regras do software a ser criado a partir de uma linguagem específica. O desenvolvimento de sistemas pode ser realizado de diversas formas. Neste contexto, o IDE assegura uma maior produtividade e precisão do processo, uma vez que várias funcionalidades não precisarão ser personalizadas a cada demanda.

O *Visual Studio Code* (VS Code) é um editor de código de código aberto desenvolvido pela Microsoft, tem funcionalidades que facilitam o processo de desenvolvimento, facilitando o trabalho de desenvolvedores. Sendo uma ferramenta de fácil entendimento, possui uma loja de extensões imensa, e que continua crescendo.

Ou seja, com essa enorme coleção de extensões, podemos adicionar diversas funcionalidades ao VS Code de forma bem simples. Qualquer um pode criar uma extensão e publicar na loja. Desse modo, sempre há novas ferramentas que podem ser interessantes para alguém.

**3.1.5. GANACHE**

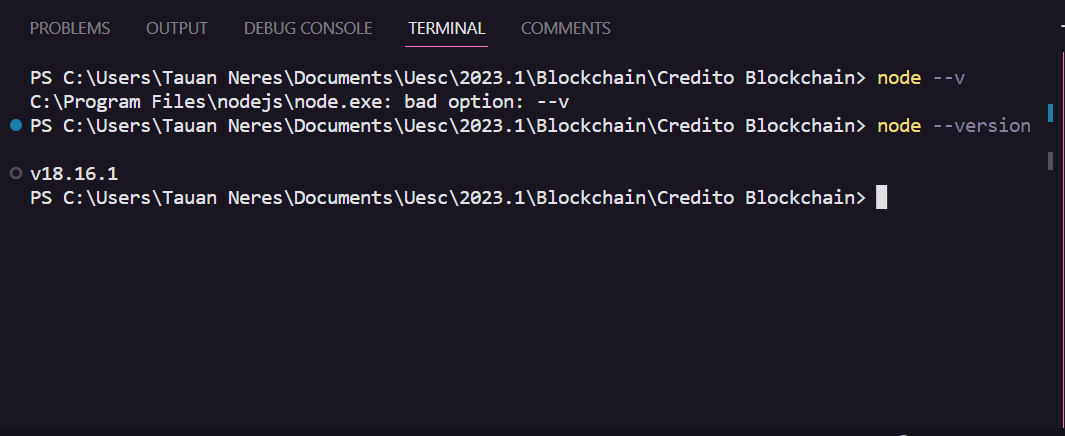
Ganache é um ambiente de desenvolvimento pessoal Ethereum que cria uma blockchain local para fins de desenvolvimento e teste. Ele é usado no projeto para simular uma blockchain Ethereum e fornece uma maneira fácil de implantar e testar contratos inteligentes localmente, sem a necessidade de uma blockchain real.

3.2. **Procedimentos metodológicos**

**3.2.1. Instalação e Configuração do Truffle Box**

Verifique se o Node.js está instalado no sistema. O Node.js é uma plataforma de tempo de execução JavaScript que permite a execução de código JavaScript fora do navegador.

Instale globalmente o Truffle Framework executando o comando npm install -g truffle. O Truffle é um framework de desenvolvimento de aplicativos descentralizados (dapps) para a plataforma Ethereum.

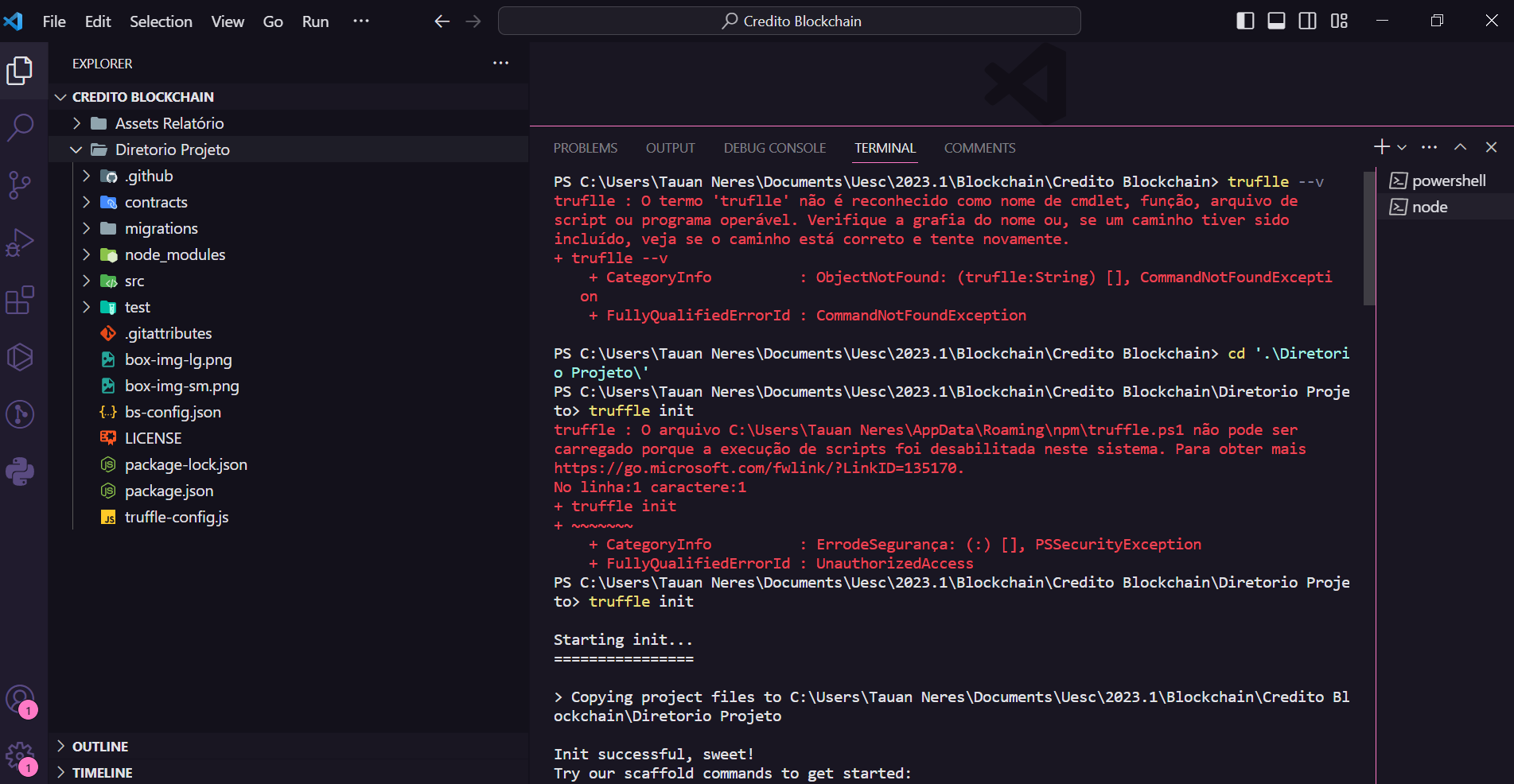


(Node.js instalado na máquina)

**3.2.2. Criação do Projeto e Inicialização**

Crie um diretório específico para o projeto e navegue até ele por meio do terminal ou prompt de comando.

Utilize o comando truffle init para criar um novo projeto Truffle e inicializar uma estrutura básica. Isso irá criar uma estrutura de diretórios e arquivos iniciais necessários para o desenvolvimento de contratos inteligentes.

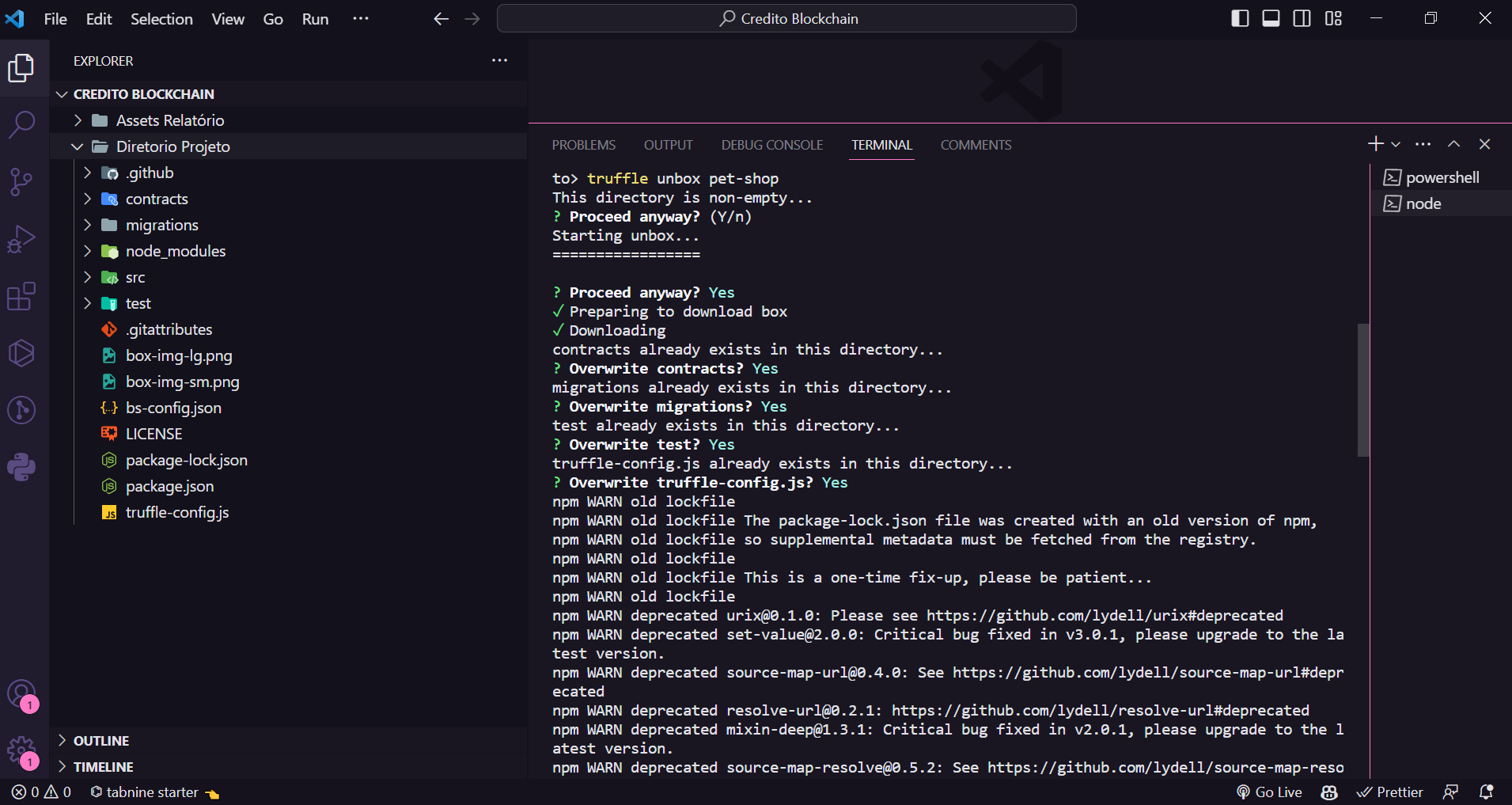


(comando truffle init, ocorreu um pequeno erro, comentado mais abaixo no relatório)

**3.2.3. Instalação do Truffle Box:**

Faça o download do Truffle Box "pet-shop" executando o comando truffle unbox pet-shop. O Truffle Box "pet-shop" é um modelo pré-configurado que fornece uma estrutura inicial para o projeto de rastreamento de adoção de animais, incluindo arquivos de contrato inteligente, front-end e testes básicos.

Aguarde até que todos os arquivos e dependências do Truffle Box "pet-shop" sejam baixados e configurados no diretório do projeto.



(abrindo a box pet-shop)

**3.2.4. Abi do Contrato Inteligente**

{

  "contractName": "Migrations",

  "abi": [

    {

      "constant": true,

      "inputs": [],

      "name": "last\_completed\_migration",

      "outputs": [

        {

          "internalType": "uint256",

          "name": "",

          "type": "uint256"

        }

      ],

      "payable": false,

      "stateMutability": "view",

      "type": "function"

    },

    {

      "constant": true,

      "inputs": [],

      "name": "owner",

      "outputs": [

        {

          "internalType": "address",

          "name": "",

          "type": "address"

        }

      ],

      "payable": false,

      "stateMutability": "view",

      "type": "function"

    },

    {

      "constant": false,

      "inputs": [

        {

          "internalType": "uint256",

          "name": "completed",

          "type": "uint256"

        }

      ],

      "name": "setCompleted",

      "outputs": [],

      "payable": false,

      "stateMutability": "nonpayable",

      "type": "function"

    }

  ],

  "metadata": "{\"compiler\":{\"version\":\"0.5.16+commit.9c3226ce\"},\"language\":\"Solidity\",\"output\":{\"abi\":[{\"constant\":true,\"inputs\":[],\"name\":\"last\_completed\_migration\",\"outputs\":[{\"internalType\":\"uint256\",\"name\":\"\",\"type\":\"uint256\"}],\"payable\":false,\"stateMutability\":\"view\",\"type\":\"function\"},{\"constant\":true,\"inputs\":[],\"name\":\"owner\",\"outputs\":[{\"internalType\":\"address\",\"name\":\"\",\"type\":\"address\"}],\"payable\":false,\"stateMutability\":\"view\",\"type\":\"function\"},{\"constant\":false,\"inputs\":[{\"internalType\":\"uint256\",\"name\":\"completed\",\"type\":\"uint256\"}],\"name\":\"setCompleted\",\"outputs\":[],\"payable\":false,\"stateMutability\":\"nonpayable\",\"type\":\"function\"}],\"devdoc\":{\"methods\":{}},\"userdoc\":{\"methods\":{}}},\"settings\":{\"compilationTarget\":{\"project:/contracts/Migrations.sol\":\"Migrations\"},\"evmVersion\":\"istanbul\",\"libraries\":{},\"optimizer\":{\"enabled\":false,\"runs\":200},\"remappings\":[]},\"sources\":{\"project:/contracts/Migrations.sol\":{\"keccak256\":\"0x7797e159bfd6b953422b4bd6d5de5946971d8b5ed74c4b1f6517d61fe236b851\",\"urls\":[\"bzz-raw://56bdf6130f3ced3e78baa0f3e7f34cb4c5131d90721326056bbf0dd202d8539d\",\"dweb:/ipfs/QmZqRKebKwn6YXejnnPribsyiXLmrAx32JpatFhvS76NKp\"]}},\"version\":1}",

  "bytecode": "",

  "deployedBytecode": "",

  "sourceMap": "66:352:0:-;;;113:10;90:33;;;;;;;;;;;;;;;;;;;;66:352;8:9:-1;5:2;;;30:1;27;20:12;5:2;66:352:0;;;;;;;",

  "deployedSourceMap": "66:352:0:-;;;;8:9:-1;5:2;;;30:1;27;20:12;5:2;66:352:0;;;;;;;;;;;;;;;;;;;;;;;;;;;;;127:36;;;:::i;:::-;;;;;;;;;;;;;;;;;;;90:33;;;:::i;:::-;;;;;;;;;;;;;;;;;;;;;;;313:103;;;;;;13:2:-1;8:3;5:11;2:2;;;29:1;26;19:12;2:2;313:103:0;;;;;;;;;;;;;;;;;:::i;:::-;;127:36;;;;:::o;90:33::-;;;;;;;;;;;;;:::o;313:103::-;225:5;;;;;;;;;;;211:19;;:10;:19;;;196:101;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;402:9;375:24;:36;;;;313:103;:::o",

  "source": "// SPDX-License-Identifier: MIT\npragma solidity >=0.4.22 <0.8.0;\n\ncontract Migrations {\n  address public owner = msg.sender;\n  uint public last\_completed\_migration;\n\n  modifier restricted() {\n    require(\n      msg.sender == owner,\n      \"This function is restricted to the contract's owner\"\n    );\n    \_;\n  }\n\n  function setCompleted(uint completed) public restricted {\n    last\_completed\_migration = completed;\n  }\n}\n",

  "sourcePath": "C:\\Users\\Tauan Neres\\Documents\\Uesc\\2023.1\\Blockchain\\Credito Blockchain\\Diretorio Projeto\\contracts\\Migrations.sol",

  "ast": {

    "absolutePath": "project:/contracts/Migrations.sol",

    "exportedSymbols": {

      "Migrations": [

        32

      ]

    },

    "id": 33,

    "nodeType": "SourceUnit",

    "nodes": [

      {

        "id": 1,

        "literals": [

          "solidity",

          ">=",

          "0.4",

          ".22",

          "<",

          "0.8",

          ".0"

        ],

        "nodeType": "PragmaDirective",

        "src": "32:32:0"

      },

      {

        "baseContracts": [],

        "contractDependencies": [],

        "contractKind": "contract",

        "documentation": null,

        "fullyImplemented": true,

        "id": 32,

        "linearizedBaseContracts": [

          32

        ],

        "name": "Migrations",

        "nodeType": "ContractDefinition",

        "nodes": [

          {

            "constant": false,

            "id": 5,

            "name": "owner",

            "nodeType": "VariableDeclaration",

            "scope": 32,

            "src": "90:33:0",

            "stateVariable": true,

            "storageLocation": "default",

            "typeDescriptions": {

              "typeIdentifier": "t\_address",

              "typeString": "address"

            },

            "typeName": {

              "id": 2,

              "name": "address",

              "nodeType": "ElementaryTypeName",

              "src": "90:7:0",

              "stateMutability": "nonpayable",

              "typeDescriptions": {

                "typeIdentifier": "t\_address",

                "typeString": "address"

              }

            },

            "value": {

              "argumentTypes": null,

              "expression": {

                "argumentTypes": null,

                "id": 3,

                "name": "msg",

                "nodeType": "Identifier",

                "overloadedDeclarations": [],

                "referencedDeclaration": 47,

                "src": "113:3:0",

                "typeDescriptions": {

                  "typeIdentifier": "t\_magic\_message",

                  "typeString": "msg"

                }

              },

              "id": 4,

              "isConstant": false,

              "isLValue": false,

              "isPure": false,

              "lValueRequested": false,

              "memberName": "sender",

              "nodeType": "MemberAccess",

              "referencedDeclaration": null,

              "src": "113:10:0",

              "typeDescriptions": {

                "typeIdentifier": "t\_address\_payable",

                "typeString": "address payable"

              }

            },

            "visibility": "public"

          },

          {

            "constant": false,

            "id": 7,

            "name": "last\_completed\_migration",

            "nodeType": "VariableDeclaration",

            "scope": 32,

            "src": "127:36:0",

            "stateVariable": true,

            "storageLocation": "default",

            "typeDescriptions": {

              "typeIdentifier": "t\_uint256",

              "typeString": "uint256"

            },

            "typeName": {

              "id": 6,

              "name": "uint",

              "nodeType": "ElementaryTypeName",

              "src": "127:4:0",

              "typeDescriptions": {

                "typeIdentifier": "t\_uint256",

                "typeString": "uint256"

              }

            },

            "value": null,

            "visibility": "public"

          },

          {

            "body": {

              "id": 18,

              "nodeType": "Block",

              "src": "190:119:0",

              "statements": [

                {

                  "expression": {

                    "argumentTypes": null,

                    "arguments": [

                      {

                        "argumentTypes": null,

                        "commonType": {

                          "typeIdentifier": "t\_address",

                          "typeString": "address"

                        },

                        "id": 13,

                        "isConstant": false,

                        "isLValue": false,

                        "isPure": false,

                        "lValueRequested": false,

                        "leftExpression": {

                          "argumentTypes": null,

                          "expression": {

                            "argumentTypes": null,

                            "id": 10,

                            "name": "msg",

                            "nodeType": "Identifier",

                            "overloadedDeclarations": [],

                            "referencedDeclaration": 47,

                            "src": "211:3:0",

                            "typeDescriptions": {

                              "typeIdentifier": "t\_magic\_message",

                              "typeString": "msg"

                            }

                          },

                          "id": 11,

                          "isConstant": false,

                          "isLValue": false,

                          "isPure": false,

                          "lValueRequested": false,

                          "memberName": "sender",

                          "nodeType": "MemberAccess",

                          "referencedDeclaration": null,

                          "src": "211:10:0",

                          "typeDescriptions": {

                            "typeIdentifier": "t\_address\_payable",

                            "typeString": "address payable"

                          }

                        },

                        "nodeType": "BinaryOperation",

                        "operator": "==",

                        "rightExpression": {

                          "argumentTypes": null,

                          "id": 12,

                          "name": "owner",

                          "nodeType": "Identifier",

                          "overloadedDeclarations": [],

                          "referencedDeclaration": 5,

                          "src": "225:5:0",

                          "typeDescriptions": {

                            "typeIdentifier": "t\_address",

                            "typeString": "address"

                          }

                        },

                        "src": "211:19:0",

                        "typeDescriptions": {

                          "typeIdentifier": "t\_bool",

                          "typeString": "bool"

                        }

                      },

                      {

                        "argumentTypes": null,

                        "hexValue": "546869732066756e6374696f6e206973207265737472696374656420746f2074686520636f6e74726163742773206f776e6572",

                        "id": 14,

                        "isConstant": false,

                        "isLValue": false,

                        "isPure": true,

                        "kind": "string",

                        "lValueRequested": false,

                        "nodeType": "Literal",

                        "src": "238:53:0",

                        "subdenomination": null,

                        "typeDescriptions": {

                          "typeIdentifier": "t\_stringliteral\_f60fe2d9d123295bf92ecf95167f1fa709e374da35e4c083bd39dc2d82acd8b1",

                          "typeString": "literal\_string \"This function is restricted to the contract's owner\""

                        },

                        "value": "This function is restricted to the contract's owner"

                      }

                    ],

                    "expression": {

                      "argumentTypes": [

                        {

                          "typeIdentifier": "t\_bool",

                          "typeString": "bool"

                        },

                        {

                          "typeIdentifier": "t\_stringliteral\_f60fe2d9d123295bf92ecf95167f1fa709e374da35e4c083bd39dc2d82acd8b1",

                          "typeString": "literal\_string \"This function is restricted to the contract's owner\""

                        }

                      ],

                      "id": 9,

                      "name": "require",

                      "nodeType": "Identifier",

                      "overloadedDeclarations": [

                        50,

                        51

                      ],

                      "referencedDeclaration": 51,

                      "src": "196:7:0",

                      "typeDescriptions": {

                        "typeIdentifier": "t\_function\_require\_pure$\_t\_bool\_$\_t\_string\_memory\_ptr\_$returns$\_\_$",

                        "typeString": "function (bool,string memory) pure"

                      }

                    },

                    "id": 15,

                    "isConstant": false,

                    "isLValue": false,

                    "isPure": false,

                    "kind": "functionCall",

                    "lValueRequested": false,

                    "names": [],

                    "nodeType": "FunctionCall",

                    "src": "196:101:0",

                    "typeDescriptions": {

                      "typeIdentifier": "t\_tuple$\_\_$",

                      "typeString": "tuple()"

                    }

                  },

                  "id": 16,

                  "nodeType": "ExpressionStatement",

                  "src": "196:101:0"

                },

                {

                  "id": 17,

                  "nodeType": "PlaceholderStatement",

                  "src": "303:1:0"

                }

              ]

            },

            "documentation": null,

            "id": 19,

            "name": "restricted",

            "nodeType": "ModifierDefinition",

            "parameters": {

              "id": 8,

              "nodeType": "ParameterList",

              "parameters": [],

              "src": "187:2:0"

            },

            "src": "168:141:0",

            "visibility": "internal"

          },

          {

            "body": {

              "id": 30,

              "nodeType": "Block",

              "src": "369:47:0",

              "statements": [

                {

                  "expression": {

                    "argumentTypes": null,

                    "id": 28,

                    "isConstant": false,

                    "isLValue": false,

                    "isPure": false,

                    "lValueRequested": false,

                    "leftHandSide": {

                      "argumentTypes": null,

                      "id": 26,

                      "name": "last\_completed\_migration",

                      "nodeType": "Identifier",

                      "overloadedDeclarations": [],

                      "referencedDeclaration": 7,

                      "src": "375:24:0",

                      "typeDescriptions": {

                        "typeIdentifier": "t\_uint256",

                        "typeString": "uint256"

                      }

                    },

                    "nodeType": "Assignment",

                    "operator": "=",

                    "rightHandSide": {

                      "argumentTypes": null,

                      "id": 27,

                      "name": "completed",

                      "nodeType": "Identifier",

                      "overloadedDeclarations": [],

                      "referencedDeclaration": 21,

                      "src": "402:9:0",

                      "typeDescriptions": {

                        "typeIdentifier": "t\_uint256",

                        "typeString": "uint256"

                      }

                    },

                    "src": "375:36:0",

                    "typeDescriptions": {

                      "typeIdentifier": "t\_uint256",

                      "typeString": "uint256"

                    }

                  },

                  "id": 29,

                  "nodeType": "ExpressionStatement",

                  "src": "375:36:0"

                }

              ]

            },

            "documentation": null,

            "id": 31,

            "implemented": true,

            "kind": "function",

            "modifiers": [

              {

                "arguments": null,

                "id": 24,

                "modifierName": {

                  "argumentTypes": null,

                  "id": 23,

                  "name": "restricted",

                  "nodeType": "Identifier",

                  "overloadedDeclarations": [],

                  "referencedDeclaration": 19,

                  "src": "358:10:0",

                  "typeDescriptions": {

                    "typeIdentifier": "t\_modifier$\_\_$",

                    "typeString": "modifier ()"

                  }

                },

                "nodeType": "ModifierInvocation",

                "src": "358:10:0"

              }

            ],

            "name": "setCompleted",

            "nodeType": "FunctionDefinition",

            "parameters": {

              "id": 22,

              "nodeType": "ParameterList",

              "parameters": [

                {

                  "constant": false,

                  "id": 21,

                  "name": "completed",

                  "nodeType": "VariableDeclaration",

                  "scope": 31,

                  "src": "335:14:0",

                  "stateVariable": false,

                  "storageLocation": "default",

                  "typeDescriptions": {

                    "typeIdentifier": "t\_uint256",

                    "typeString": "uint256"

                  },

                  "typeName": {

                    "id": 20,

                    "name": "uint",

                    "nodeType": "ElementaryTypeName",

                    "src": "335:4:0",

                    "typeDescriptions": {

                      "typeIdentifier": "t\_uint256",

                      "typeString": "uint256"

                    }

                  },

                  "value": null,

                  "visibility": "internal"

                }

              ],

              "src": "334:16:0"

            },

            "returnParameters": {

              "id": 25,

              "nodeType": "ParameterList",

              "parameters": [],

              "src": "369:0:0"

            },

            "scope": 32,

            "src": "313:103:0",

            "stateMutability": "nonpayable",

            "superFunction": null,

            "visibility": "public"

          }

        ],

        "scope": 33,

        "src": "66:352:0"

      }

    ],

    "src": "32:387:0"

  },

  "legacyAST": {

    "attributes": {

      "absolutePath": "project:/contracts/Migrations.sol",

      "exportedSymbols": {

        "Migrations": [

          32

        ]

      }

    },

    "children": [

      {

        "attributes": {

          "literals": [

            "solidity",

            ">=",

            "0.4",

            ".22",

            "<",

            "0.8",

            ".0"

          ]

        },

        "id": 1,

        "name": "PragmaDirective",

        "src": "32:32:0"

      },

      {

        "attributes": {

          "baseContracts": [

            null

          ],

          "contractDependencies": [

            null

          ],

          "contractKind": "contract",

          "documentation": null,

          "fullyImplemented": true,

          "linearizedBaseContracts": [

            32

          ],

          "name": "Migrations",

          "scope": 33

        },

        "children": [

          {

            "attributes": {

              "constant": false,

              "name": "owner",

              "scope": 32,

              "stateVariable": true,

              "storageLocation": "default",

              "type": "address",

              "visibility": "public"

            },

            "children": [

              {

                "attributes": {

                  "name": "address",

                  "stateMutability": "nonpayable",

                  "type": "address"

                },

                "id": 2,

                "name": "ElementaryTypeName",

                "src": "90:7:0"

              },

              {

                "attributes": {

                  "argumentTypes": null,

                  "isConstant": false,

                  "isLValue": false,

                  "isPure": false,

                  "lValueRequested": false,

                  "member\_name": "sender",

                  "referencedDeclaration": null,

                  "type": "address payable"

                },

                "children": [

                  {

                    "attributes": {

                      "argumentTypes": null,

                      "overloadedDeclarations": [

                        null

                      ],

                      "referencedDeclaration": 47,

                      "type": "msg",

                      "value": "msg"

                    },

                    "id": 3,

                    "name": "Identifier",

                    "src": "113:3:0"

                  }

                ],

                "id": 4,

                "name": "MemberAccess",

                "src": "113:10:0"

              }

            ],

            "id": 5,

            "name": "VariableDeclaration",

            "src": "90:33:0"

          },

          {

            "attributes": {

              "constant": false,

              "name": "last\_completed\_migration",

              "scope": 32,

              "stateVariable": true,

              "storageLocation": "default",

              "type": "uint256",

              "value": null,

              "visibility": "public"

            },

            "children": [

              {

                "attributes": {

                  "name": "uint",

                  "type": "uint256"

                },

                "id": 6,

                "name": "ElementaryTypeName",

                "src": "127:4:0"

              }

            ],

            "id": 7,

            "name": "VariableDeclaration",

            "src": "127:36:0"

          },

          {

            "attributes": {

              "documentation": null,

              "name": "restricted",

              "visibility": "internal"

            },

            "children": [

              {

                "attributes": {

                  "parameters": [

                    null

                  ]

                },

                "children": [],

                "id": 8,

                "name": "ParameterList",

                "src": "187:2:0"

              },

              {

                "children": [

                  {

                    "children": [

                      {

                        "attributes": {

                          "argumentTypes": null,

                          "isConstant": false,

                          "isLValue": false,

                          "isPure": false,

                          "isStructConstructorCall": false,

                          "lValueRequested": false,

                          "names": [

                            null

                          ],

                          "type": "tuple()",

                          "type\_conversion": false

                        },

                        "children": [

                          {

                            "attributes": {

                              "argumentTypes": [

                                {

                                  "typeIdentifier": "t\_bool",

                                  "typeString": "bool"

                                },

                                {

                                  "typeIdentifier": "t\_stringliteral\_f60fe2d9d123295bf92ecf95167f1fa709e374da35e4c083bd39dc2d82acd8b1",

                                  "typeString": "literal\_string \"This function is restricted to the contract's owner\""

                                }

                              ],

                              "overloadedDeclarations": [

                                50,

                                51

                              ],

                              "referencedDeclaration": 51,

                              "type": "function (bool,string memory) pure",

                              "value": "require"

                            },

                            "id": 9,

                            "name": "Identifier",

                            "src": "196:7:0"

                          },

                          {

                            "attributes": {

                              "argumentTypes": null,

                              "commonType": {

                                "typeIdentifier": "t\_address",

                                "typeString": "address"

                              },

                              "isConstant": false,

                              "isLValue": false,

                              "isPure": false,

                              "lValueRequested": false,

                              "operator": "==",

                              "type": "bool"

                            },

                            "children": [

                              {

                                "attributes": {

                                  "argumentTypes": null,

                                  "isConstant": false,

                                  "isLValue": false,

                                  "isPure": false,

                                  "lValueRequested": false,

                                  "member\_name": "sender",

                                  "referencedDeclaration": null,

                                  "type": "address payable"

                                },

                                "children": [

                                  {

                                    "attributes": {

                                      "argumentTypes": null,

                                      "overloadedDeclarations": [

                                        null

                                      ],

                                      "referencedDeclaration": 47,

                                      "type": "msg",

                                      "value": "msg"

                                    },

                                    "id": 10,

                                    "name": "Identifier",

                                    "src": "211:3:0"

                                  }

                                ],

                                "id": 11,

                                "name": "MemberAccess",

                                "src": "211:10:0"

                              },

                              {

                                "attributes": {

                                  "argumentTypes": null,

                                  "overloadedDeclarations": [

                                    null

                                  ],

                                  "referencedDeclaration": 5,

                                  "type": "address",

                                  "value": "owner"

                                },

                                "id": 12,

                                "name": "Identifier",

                                "src": "225:5:0"

                              }

                            ],

                            "id": 13,

                            "name": "BinaryOperation",

                            "src": "211:19:0"

                          },

                          {

                            "attributes": {

                              "argumentTypes": null,

                              "hexvalue": "546869732066756e6374696f6e206973207265737472696374656420746f2074686520636f6e74726163742773206f776e6572",

                              "isConstant": false,

                              "isLValue": false,

                              "isPure": true,

                              "lValueRequested": false,

                              "subdenomination": null,

                              "token": "string",

                              "type": "literal\_string \"This function is restricted to the contract's owner\"",

                              "value": "This function is restricted to the contract's owner"

                            },

                            "id": 14,

                            "name": "Literal",

                            "src": "238:53:0"

                          }

                        ],

                        "id": 15,

                        "name": "FunctionCall",

                        "src": "196:101:0"

                      }

                    ],

                    "id": 16,

                    "name": "ExpressionStatement",

                    "src": "196:101:0"

                  },

                  {

                    "id": 17,

                    "name": "PlaceholderStatement",

                    "src": "303:1:0"

                  }

                ],

                "id": 18,

                "name": "Block",

                "src": "190:119:0"

              }

            ],

            "id": 19,

            "name": "ModifierDefinition",

            "src": "168:141:0"

          },

          {

            "attributes": {

              "documentation": null,

              "implemented": true,

              "isConstructor": false,

              "kind": "function",

              "name": "setCompleted",

              "scope": 32,

              "stateMutability": "nonpayable",

              "superFunction": null,

              "visibility": "public"

            },

            "children": [

              {

                "children": [

                  {

                    "attributes": {

                      "constant": false,

                      "name": "completed",

                      "scope": 31,

                      "stateVariable": false,

                      "storageLocation": "default",

                      "type": "uint256",

                      "value": null,

                      "visibility": "internal"

                    },

                    "children": [

                      {

                        "attributes": {

                          "name": "uint",

                          "type": "uint256"

                        },

                        "id": 20,

                        "name": "ElementaryTypeName",

                        "src": "335:4:0"

                      }

                    ],

                    "id": 21,

                    "name": "VariableDeclaration",

                    "src": "335:14:0"

                  }

                ],

                "id": 22,

                "name": "ParameterList",

                "src": "334:16:0"

              },

              {

                "attributes": {

                  "parameters": [

                    null

                  ]

                },

                "children": [],

                "id": 25,

                "name": "ParameterList",

                "src": "369:0:0"

              },

              {

                "attributes": {

                  "arguments": null

                },

                "children": [

                  {

                    "attributes": {

                      "argumentTypes": null,

                      "overloadedDeclarations": [

                        null

                      ],

                      "referencedDeclaration": 19,

                      "type": "modifier ()",

                      "value": "restricted"

                    },

                    "id": 23,

                    "name": "Identifier",

                    "src": "358:10:0"

                  }

                ],

                "id": 24,

                "name": "ModifierInvocation",

                "src": "358:10:0"

              },

              {

                "children": [

                  {

                    "children": [

                      {

                        "attributes": {

                          "argumentTypes": null,

                          "isConstant": false,

                          "isLValue": false,

                          "isPure": false,

                          "lValueRequested": false,

                          "operator": "=",

                          "type": "uint256"

                        },

                        "children": [

                          {

                            "attributes": {

                              "argumentTypes": null,

                              "overloadedDeclarations": [

                                null

                              ],

                              "referencedDeclaration": 7,

                              "type": "uint256",

                              "value": "last\_completed\_migration"

                            },

                            "id": 26,

                            "name": "Identifier",

                            "src": "375:24:0"

                          },

                          {

                            "attributes": {

                              "argumentTypes": null,

                              "overloadedDeclarations": [

                                null

                              ],

                              "referencedDeclaration": 21,

                              "type": "uint256",

                              "value": "completed"

                            },

                            "id": 27,

                            "name": "Identifier",

                            "src": "402:9:0"

                          }

                        ],

                        "id": 28,

                        "name": "Assignment",

                        "src": "375:36:0"

                      }

                    ],

                    "id": 29,

                    "name": "ExpressionStatement",

                    "src": "375:36:0"

                  }

                ],

                "id": 30,

                "name": "Block",

                "src": "369:47:0"

              }

            ],

            "id": 31,

            "name": "FunctionDefinition",

            "src": "313:103:0"

          }

        ],

        "id": 32,

        "name": "ContractDefinition",

        "src": "66:352:0"

      }

    ],

    "id": 33,

    "name": "SourceUnit",

    "src": "32:387:0"

  },

  "compiler": {

    "name": "solc",

    "version": "0.5.16+commit.9c3226ce.Emscripten.clang"

  },

  "networks": {

    "5777": {

      "events": {},

      "links": {},

      "address": "0x483ed5b67C114800e6d0c4f1a9EabF0Dfe38988b",

      "transactionHash": "0xaaddf8f2b82a75d5c80f8a1e54c504ef681fd026dad1e0467b16c8f9ab480254"

    }

  },

  "schemaVersion": "3.4.14",

  "updatedAt": "2023-07-09T06:12:06.181Z",

  "networkType": "ethereum",

  "devdoc": {

    "methods": {}

  },

  "userdoc": {

    "methods": {}

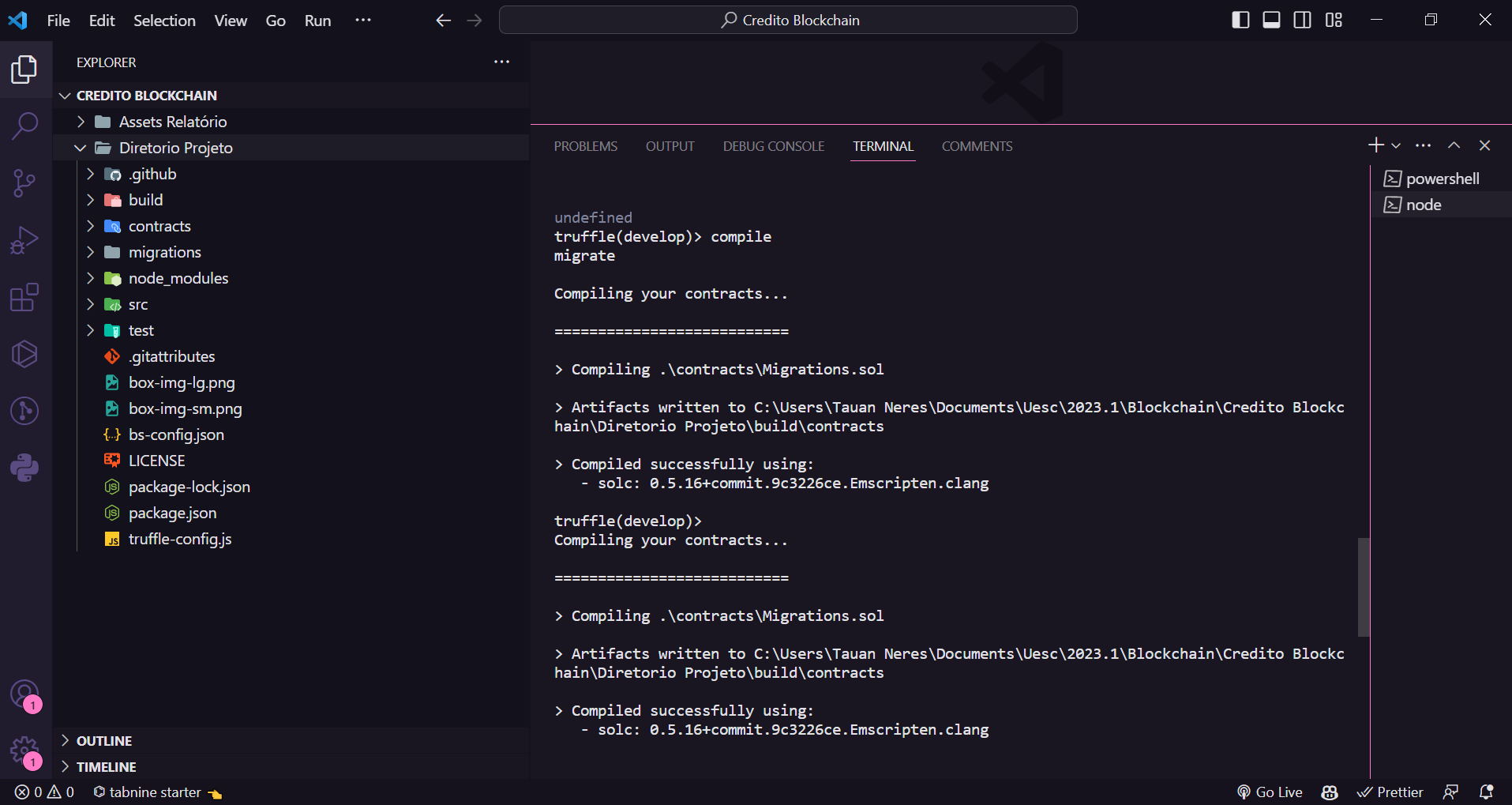
  }

}

**3.2.5. Compilação dos Contratos**

Acesse o diretório do projeto por meio do terminal ou prompt de comando.

Utilize o comando truffle compile para compilar os contratos inteligentes presentes na pasta contracts. Essa etapa traduz o código fonte dos contratos para uma representação executável e gera os artefatos compilados na pasta build/contracts.



(compilação e implantação dos contratos)

**3.2.6. Implantação dos Contratos**

Certifique-se de ter uma blockchain local em execução para fins de desenvolvimento e teste. É possível utilizar o Ganache, um ambiente de desenvolvimento pessoal Ethereum, para criar uma blockchain local.

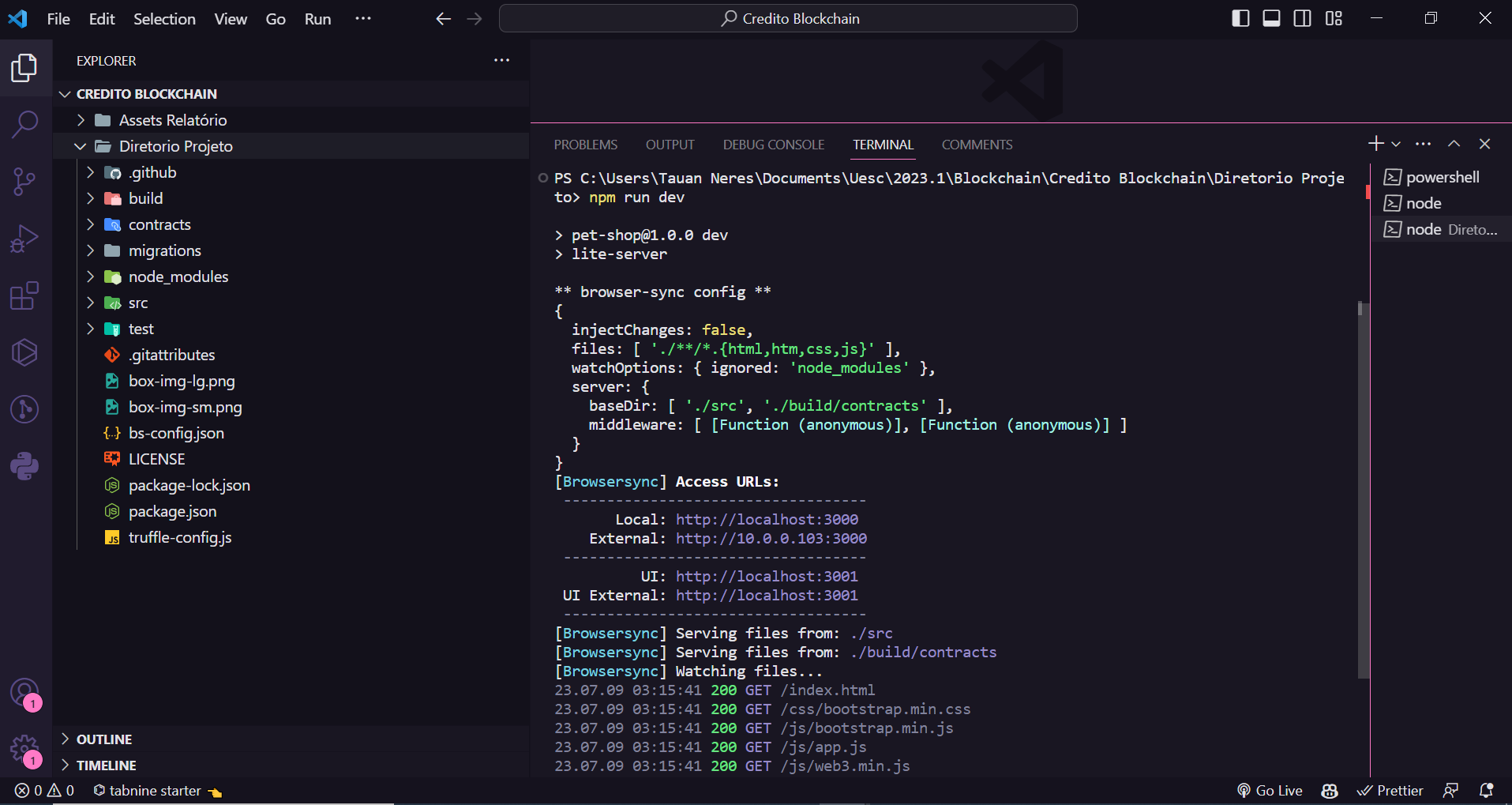
Verifique se o arquivo de configuração truffle-config.js (ou truffle.js para versões anteriores) está corretamente configurado com as informações da blockchain local.

Execute o comando “truffle migrate” para implantar os contratos inteligentes na blockchain local. Isso envia os contratos para a blockchain e registra as informações de implantação no arquivo build/contracts/Migrations.json.

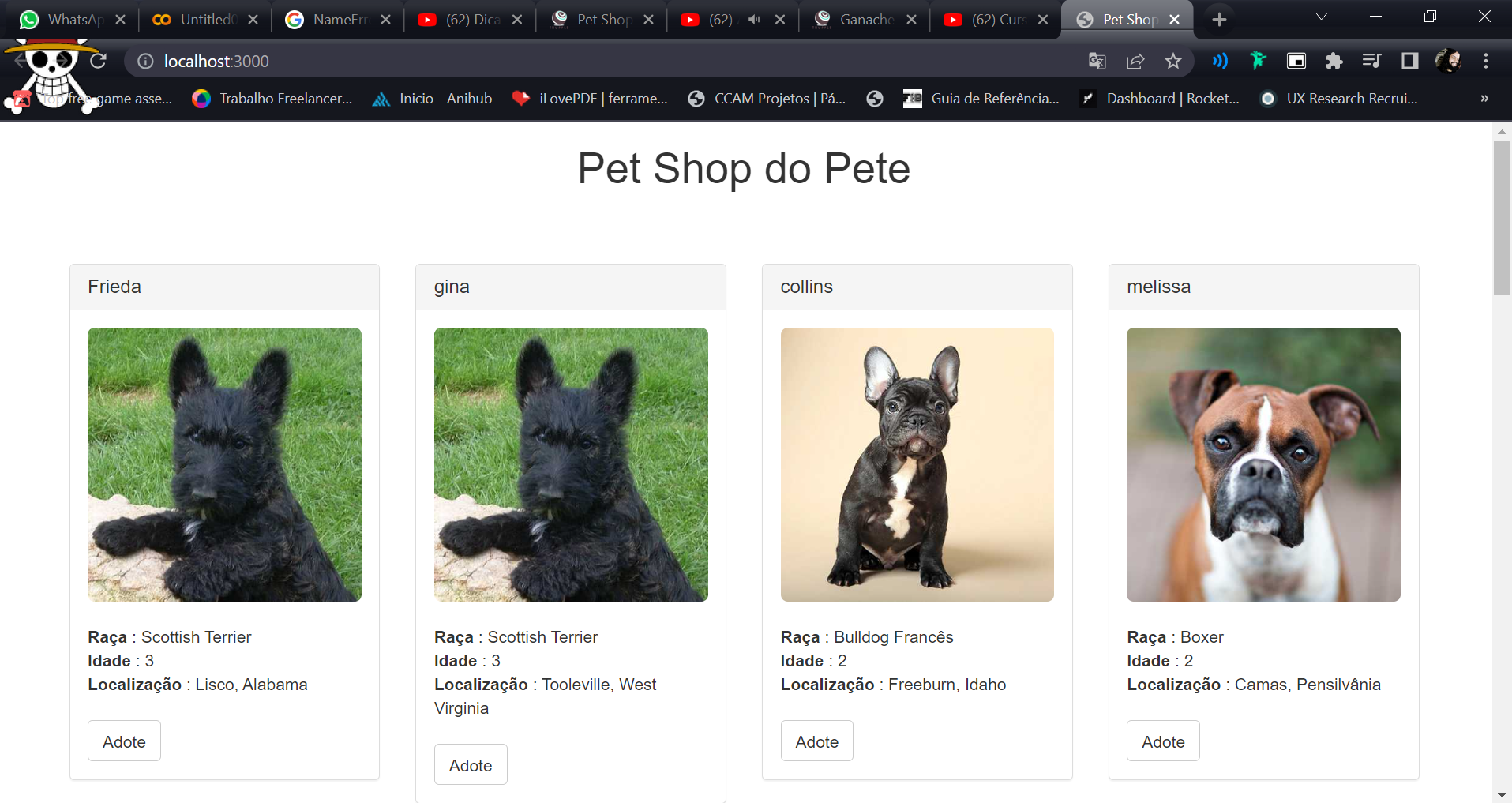
**3.2.7. Teste dos Contratos**

Navegue até o diretório do projeto por meio do terminal ou prompt de comando.

Utilize o comando “truffle run dev” para executar os testes dos contratos inteligentes presentes na pasta test. Essa etapa verifica se os contratos funcionam conforme o esperado e fornecerá feedback sobre o funcionamento correto dos contratos.



**3.2.8. Interação funcionando**



# Resultados

Uma análise rápida dos resultados obtidos ao interagir com o contrato inteligente do box escolhido revelou o seguinte:

**4.1. Funcionalidades do contrato**

O contrato inteligente do box apresentou um conjunto de funcionalidades interessantes, como a capacidade de armazenar e recuperar dados, realizar transações e executar lógica programável. Essas funcionalidades forneceram uma base sólida para a construção de aplicativos descentralizados (DApps) e contratos mais complexos.

**4.2. Resultados esperados**

Os resultados obtidos estiveram alinhados com as expectativas iniciais em relação às funcionalidades do contrato inteligente do box.

**4.3. Avaliação do desempenho**

Em relação ao desempenho do contrato, observou-se um tempo de execução aceitável para as operações realizadas. O consumo de recursos, como as taxas de gás, também se mostrou razoável, considerando a complexidade e o propósito do contrato.

* 1. **Validação das funcionalidades**

- Durante a análise, foi possível verificar que as funcionalidades do contrato estão operando corretamente, atendendo às especificações e requisitos definidos.

**4.5. Possíveis melhorias**

Embora o contrato inteligente do box tenha apresentado um conjunto sólido de funcionalidades, identificou-se algumas áreas que poderiam ser aprimoradas. Por exemplo, melhorias na segurança, como a implementação de mecanismos de autenticação ou verificação de acesso, poderiam ser consideradas para proteger os dados armazenados.

# Discussão

Essa análise rápida dos resultados demonstrou a eficácia e a utilidade do contrato inteligente do box escolhido. Os resultados estão de acordo com as funcionalidades esperadas, fornecendo uma base sólida para a evolução do projeto em direção a um escopo maior. No entanto, é importante lembrar que uma análise mais aprofundada e abrangente seria necessária para fornecer insights detalhados sobre todas as funcionalidades e aspectos do contrato inteligente do box.

# Conclusão

Em conclusão, a interação com o contrato inteligente do box escolhido demonstrou sua eficácia e relevância para o desenvolvimento de aplicações descentralizadas. Os resultados obtidos estiveram alinhados com as expectativas, validando as funcionalidades do contrato. Foram identificadas possíveis melhorias, como aprimoramentos na segurança e escalabilidade. O relatório ressaltou a importância da ABI do contrato e forneceu capturas de tela para ilustrar as etapas da interação. Em suma, o contrato inteligente do box representa uma valiosa contribuição para o ecossistema blockchain, promovendo a adoção de contratos inteligentes e facilitando o desenvolvimento de projetos inovadores.

1. **Referências**

<https://trufflesuite.com/ganache/>

<https://trufflesuite.com/guides/pet-shop/>

<https://nodejs.org/en>

<https://code.visualstudio.com/>