PARADIGMA ORIENTADO A CONTRATOS

PROJETO DA DISCIPLINA PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO

O QUE É O PARADIGMA ORIENTADO A CONTRATOS?

SURGIU COMO UMA **ESPECIALIZAÇÃO** DO PARADIGMA ORIENTADO A **OBJETOS,** PARA A CRIAÇÃO DE **CONTRATOS INTELIGENTES** EM **BLOCKCHAINS.**

O QUE É O PARADIGMA ORIENTADO A CONTRATOS?

PARA **ENTENDER O PARADIGMA**, É PRECISO CONHECER O CONCEITO BÁSICO DE **BLOCKCHAIN**.

E O QUE É BLOCKCHAIN?

BLOCKCHAIN PODE SER CONSIDERADA, DE FORMA SIMPLIFICADA, COMO UM **BANCO DE DADOS DESCENTRALIZADO.**

E O QUE É BLOCKCHAIN?

De forma mais específica, uma blockchain funciona como um **livro de registros descentralizado**, onde qualquer **informação** nele incluída **não pode ser apagada ou modificada** no futuro.

QUAL A IMPORTÂNCIA DAS BLOCKCHAINS?

A PRIMEIRA BLOCKCHAIN SURGIU **2009** COMO A TECNOLOGIA POR TRÁS DO FUNCIONAMENTO DO **BITCOIN** E É A **BASE** DE TODO ECOSSISTEMA DE **CRIPTOMOEDAS** E TECNOLOGIAS RELACIONADAS (JOGOS PLAY-TO-EARN, NFTS, DEFI, ETC).

QUAL A IMPORTÂNCIA DAS BLOCKCHAINS?

- O PRINCIPAL OBJETIVO DE UMA BLOCKCHAIN É SER CAPAZ DE ARMAZENAR **INFORMAÇÕES DE VALOR INTRÍNSECO** (NA MAIORIA DOS CASOS, **MONETÁRIO**) DE FORMA INVIOLÁVEL.

QUAL A IMPORTÂNCIA DAS BLOCKCHAINS?

PARA DAR SENTIDO AO ARMAZENAMENTO DE INFORMAÇÕES COM VALOR INTRÍNSECO, AS BLOCKCHAINS REPRESENTAM AS **ENTIDADES DETENTORAS DE ALGUMA INFORMAÇÃO** DE VALOR COMO **ENDEREÇOS.**

QUAL A IMPORTÂNCIA DAS BLOCKCHAINS?

MAIS DO QUE SIMPLESMENTE ARMAZENAR INFORMAÇÕES COM VALOR INTRÍNSECO, AS BLOCKCHAINS REGISTRAM O **HISTÓRICO DE TODAS AS TRANSAÇÕES** ENTRE ENTIDADES DETENTORAS DE INFORMAÇÕES DE VALOR.

QUAL A IMPORTÂNCIA DAS BLOCKCHAINS?

Pessoa 1

Pessoa 1 transfere 20 para Pessoa 2.

Pessoa 2

A blockchain armazena o valor associado a cada entidade, pelo registro das transações.

QUAL A IMPORTÂNCIA DAS BLOCKCHAINS?

AS BLOCKCHAINS FUNCIONAM COMO UM **LIVRO DE REGISTROS DE UM CARTÓRIO,** ONDE A **PROPRIEDADE** DE UM DETERMINADO BEM É **RASTREADA** ATRAVÉS DE SUAS **TRANSAÇÕES.**

QUAL A IMPORTÂNCIA DAS BLOCKCHAINS?

DIFERENTEMENTE, DOS LIVROS-RAZÃO DE CARTÓRIOS CONVENCIONAIS, UMA BLOCKCHAIN **DISPENSA A EXISTÊNCIA DE UMA AUTORIDADE CENTRAL** PARA GARANTIR QUE AS TRANSAÇÕES SÃO REAIS E INVIOLÁVEIS.

QUAL A IMPORTÂNCIA DAS BLOCKCHAINS?

AS BLOCKCHAINS CONSEGUEM GARANTIR A **INVIOLABILIDADE** DE SEUS DADOS ATRAVÉS DA COMBINAÇÃO DE CONCEITOS COMO **CRIPTOGRAFIA** E **PROTOCOLO DE CONSENSO.**

BITCOIN SCRIPT

Quando o Bitcoin foi lançado (juntamente com a sua blockchain), ele trouxe consigo o conceito de **Bitcoin script**, como uma **linguagem de programação baseada em pilha.**

BITCOIN SCRIPT

O BITCOIN SCRIPT FOI PROJETADO PARA PROVER **CONDIÇÕES DE PAGAMENTO** ENTRE ENDEREÇOS DA SUA BLOCKCHAIN, DERIVANDO-SE DAÍ OS CHAMADOS **CONTRATOS INTELIGENTES.**

BITCOIN SCRIPT

Pessoa 1

Pessoa 1 transfere 20 para Pessoa 2,
se a Pessoa 2 atender as seguintes condições...

Pessoa 2

A blockchain armazena o valor associado a cada entidade,
pelo registro das transações.

SOLIDITY

Apesar da revolução trazida pelo Bitcoin, foi o **Ethereum** que popularizou o conceito de **contratos inteligentes.**

SOLIDITY

O Ethereum integrou a sua blockchain com uma máquina virtual capaz de executar uma **linguagem turing completa,** chamada de **Solidity.**

SOLIDITY

A SOLIDITY UTILIZA UMA **ESPECIALIZAÇÃO** DO **PARADIGMA ORIENTADO A OBJETOS,** UMA VEZ QUE ELE NÃO PERMITE A DECLARAÇÃO DE CLASSES GENERICAMENTE.

SOLIDITY

Ao invés de permitir a declaração de classes genericamente, a Solidity **especializou** o conceito de **classes para contratos**.

SOLIDITY

A BLOCKCHAIN DA ETHEREUM SUPORTA **DOIS TIPOS DE ENDEREÇOS: ENTIDADES EXTERNAS** (SIMILAR AO BITCOIN) E **CONTRATOS INTELIGENTES.**

SOLIDITY

UM **CONTRATO INTELIGENTE** PODE TER UMA INFORMAÇÃO DE **VALOR INTRÍNSECO ASSOCIADO** A ELE, ASSIM COMO ACONTECE COM UMA ENTIDADE EXTERNA.

O NOSSO PROJETO

ESTENDER A **LINGUAGEM ORIENTADA A OBJETOS 1** PARA SUPORTAR ALGUMAS DAS PRINCIPAIS CARACTERÍSTICAS DO **PARADIGMA ORIENTADO A CONTRATOS.**

O NOSSO PROJETO

O FOCO SERÁ ADICIONAR FUNCIONALIDADES QUE ASSOCIEM ALGUMA **INFORMAÇÃO DE VALOR INTRÍNSECO** AOS **CONTRATOS INTELIGENTES** E **ENTIDADES EXTERNAS.**

O NOSSO PROJETO

UM **CONTRATO** DEVERÁ SER **DECLARADO** DE FORMA SIMILAR A UMA CLASSE, PORÉM FAZENDO USO DA PALAVRA-CHAVE **CONTRACT.**

O NOSSO PROJETO

TODO **CONTRATO** SERÁ **INSTANCIADO COM UM VALOR ASSOCIADO** A ELE (QUE PODE SER ZERO) VINDO DE ALGUMA **ENTIDADE EXTERNA** (USUÁRIO).

O NOSSO PROJETO

TODO **CONTRATO SERÁ INSTANCIADO** COM A EXIGÊNCIA DE **DOIS PARÂMETROS** (USUÁRIO E VALOR), NA SEGUINTE ORDEM: **NEW** NOME DO CONTRATO (USUÁRIO, VALOR);

O NOSSO PROJETO

UM **USUÁRIO** PODERÁ SER DECLARADO COMO UM **TIPO INTEIRO** DA LINGUAGEM E O SEU VALOR SERÁ O SEU SALDO.

O NOSSO PROJETO

UM **CONTRATO** PODERÁ TER DOIS TIPOS DE **ATRIBUTOS: NATIVOS** E NÃO NATIVOS (**DEFINIDOS PELO USUÁRIO**).

O NOSSO PROJETO

TODO **CONTRATO** TERÁ UM **ATRIBUTO** CHAMADO **BALANCE,** QUE ARMAZENARÁ O SEU SALDO.

O NOSSO PROJETO

UM **CONTRATO** PODERÁ TER DOIS TIPOS DE **MÉTODOS: NATIVOS** E NÃO NATIVOS (**DEFINIDOS PELO USUÁRIO**).

O NOSSO PROJETO

Os **MÉTODOS** (nativos e não nativos) serão representados como **PROCEDIMENTOS GENÉRICOS** e poderão definidos como pagos (**PAYABLE**) ou não pagos (**PROC**).

O NOSSO PROJETO

UM MÉTODO PAGO (**PAYABLE**) HABILITARÁ O **ENVIO** DE ALGUM **VALOR PARA O CONTRATO**, JUNTAMENTE COM A SUA CHAMADA.

O NOSSO PROJETO

TODO **MÉTODO PAGO SERÁ CHAMADO** COM A EXIGÊNCIA DE **DOIS PARÂMETROS** (USUÁRIO E VALOR), NA SEGUINTE ORDEM: NOME DO MÉTODO (USUÁRIO, VALOR);

O NOSSO PROJETO

TODO CONTRATO TERÁ UM **MÉTODO NATIVO NÃO PAGO** CHAMADO **TRANSFER**(USUÁRIO, VALOR).

O NOSSO PROJETO

O **MÉTODO TRANSFER** IRÁ **ENVIAR** UM DETERMINADO **VALOR PARA O USUÁRIO** PASSADO COMO PARÂMETRO (CASO O CONTRATO POSSUA TAL VALOR).

EXEMPLO DE USO DO PROJETO

```
VAR USUARIO = 100;

{ CONTRACT BANCO {

    PAYABLE DEPOSITAR() {

    // AQUI PODERIA SER DECLARADA ALGUMA FUNCIONALIDADE AO CONTRATO.

} } };
```

EXEMPLO DE USO DO PROJETO

```
{ VAR BANCO = NEW BANCO(USUARIO, 25);

BANCO.DEPOSITAR(USUÁRIO, 50);

WRITE(USUARIO);

WRITE(BANCO.BALANCE); }
```

GRAMÁTICA

<Programa> ::= "{" <**Declaracao**> ";" <Comando> "}“

<Declaracao> ::= <DecClasse> | <**DecContrato**>

.

.

.

GRAMÁTICA

<DecClasse> ::= "classe" <Id> "{" <DecVariavel> ";" <DecProcedimento> "}" |
<DecClasse> "," <DecClasse>

GRAMÁTICA

<DecContrato> ::= "contract" <Id> "{" <DecVariavel> ";" <DecProcedimentoGenerico> "}" | <DecContrato> "," <DecContrato>

GRAMÁTICA

<**DecProcedimentoGenerico**> ::= <DecProcedimento> | <**DecPayable**> | <DecProcedimentoGenerico> "," <DecProcedimentoGenerico>

GRAMÁTICA

<DecProcedimento> ::= "proc" <Id> "(" <ListaDeclaracaoParametro> ")" "{" <Comando> "}"

GRAMÁTICA

<**DECPAYABLE**> ::= "**PAYABLE**" <ID> "(" <LISTADECLARACAOPARAMETRO> ")" "{" <COMANDO> "}"