



# PARADIGMA ORIENTADO A CONTRATOS

**PROJETO DA DISCIPLINA PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO**

EQUIPE: DANILO RAFAEL DE LIMA CABRAL E HENRIQUE FERRAZ ARCOVERDE

# O QUE É O PARADIGMA ORIENTADO A CONTRATOS?

SURTIU 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** NELA INCLUÍDA **NÃO PODE SER APAGADA OU MODIFICADA** NO FUTURO.

# QUAL A IMPORTÂNCIA DAS BLOCKCHAINS?

A PRIMEIRA BLOCKCHAIN SURTIU **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?

○ 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?

EM GERAL, CADA **ENDEREÇO** NUMA BLOCKCHAIN É CRIADO A PARTIR DE UMA **CHAVE PÚBLICA**, A QUAL POR SUA VEZ É GERADA A PARTIR DE UMA **CHAVE PRIVADA**.

# QUAL A IMPORTÂNCIA DAS BLOCKCHAINS?

UMA **CHAVE PÚBLICA** FUNCIONA COMO O **IDENTIFICADOR** DE ALGUÉM NUMA BLOCKCHAIN.

# QUAL A IMPORTÂNCIA DAS BLOCKCHAINS?

CRIADO A PARTIR DE UMA CHAVE PÚBLICA, O **ENDEREÇO** FUNCIONA COMO UM **IDENTIFICADOR MENOR E MAIS AMIGÁVEL** NUMA BLOCKCHAIN.



# QUAL A IMPORTÂNCIA DAS BLOCKCHAINS?

POR OUTRO LADO, A **CHAVE PRIVADA** FUNCIONA COMO UMA **SENHA** PARA DAR ACESSO A QUALQUER VALOR ASSOCIADO A UMA **CHAVE PÚBLICA** GERADA POR ELA (E, POR ISSO, DEVE SER SEMPRE MANTIDA EM SEGREDO).

# QUAL A IMPORTÂNCIA DAS BLOCKCHAINS?

MAIS DO QUE SIMPLEMENTE 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?

Endereço 1



Endereço 1 transfere 20 para Endereço 2

Endereço 2



A blockchain armazena uma transação assinada pelo Endereço 1  
que libera o valor 20 para o Endereço 2



# 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 INVOLÁVEIS.

# QUAL A IMPORTÂNCIA DAS BLOCKCHAINS?

AS BLOCKCHAINS CONSEGUEM GARANTIR A **INVIOLABILIDADE** DE SEUS DADOS ATRAVÉS DA COMBINAÇÃO DE CONCEITOS COMO **CRİPTOGRAFIA** 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

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

# BITCOIN SCRIPT

Endereço 1



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

Endereço 2



A blockchain armazena uma transação assinada pelo Endereço 1  
que libera o valor 20 para o Endereço 2



# BITCOIN SCRIPT

NA PRÁTICA, **TODAS AS TRANSAÇÕES** (MESMO AS MAIS SIMPLES) ENTRE ENDEREÇOS DA REDE DO BITCOIN SÃO REALIZADAS UTILIZANDO O **BITCOIN SCRIPT**.

# BITCOIN SCRIPT

ISSO ACONTECE PORQUE, NO **TIPO DE TRANSAÇÃO MAIS SIMPLES**, O PROPRIETÁRIO DO **ENDEREÇO 1 DEVE PROVAR A SUA PROPRIEDADE** SOBRE O SEU ENDEREÇO, A FIM DE ENVIAR UM VALOR ASSOCIADO A ELE PARA OUTRO ENDEREÇO.

# BITCOIN SCRIPT

DE FORMA SIMILAR, O **PROPRIETÁRIO DO ENDEREÇO 2 DEVERÁ COMPROVAR** QUE TAL ENDEREÇO É DE SUA **PROPRIEDADE**, A FIM DE RESGATAR O SALDO LIBERADO PRA ELE PELO ENDEREÇO 1.



# EXEMPLO DE TRANSAÇÃO COM O BITCOIN SCRIPT

Endereço 1



**Value:** 20

**ScriptPubKey:** OP\_DUP OP\_HASH160 <Hash da Chave Pública 2>  
OP\_EQUALVERIFY OP\_CHECKSIG

Endereço 2



A blockchain armazena uma transação assinada pelo Endereço 1  
que libera o valor 20 para o Endereço 2

# EXEMPLO DE TRANSAÇÃO COM O BITCOIN SCRIPT

PARA RESGATAR O VALOR ENVIADO PELO ENDEREÇO 1, O **PROPRIETÁRIO DO ENDEREÇO 2** DEVERÁ ENVIAR UM SCRIPT DE RESGATE PARA A TRANSAÇÃO ANTERIOR, CONTENDO UMA **ASSINATURA** GERADA POR SUA **CHAVE PRIVADA**.

# EXEMPLO DE TRANSAÇÃO COM O BITCOIN SCRIPT

A FIM DE **VERIFICAR A AUTENTICIDADE DO SCRIPT DE RESGATE** ENVIADO PELO ENDEREÇO 2, A BLOCKCHAIN IRÁ UNÍ-LO COM O **SCRIPTPubKey** ENVIADO PELO ENDEREÇO 1.



# EXEMPLO DE TRANSAÇÃO COM O BITCOIN SCRIPT

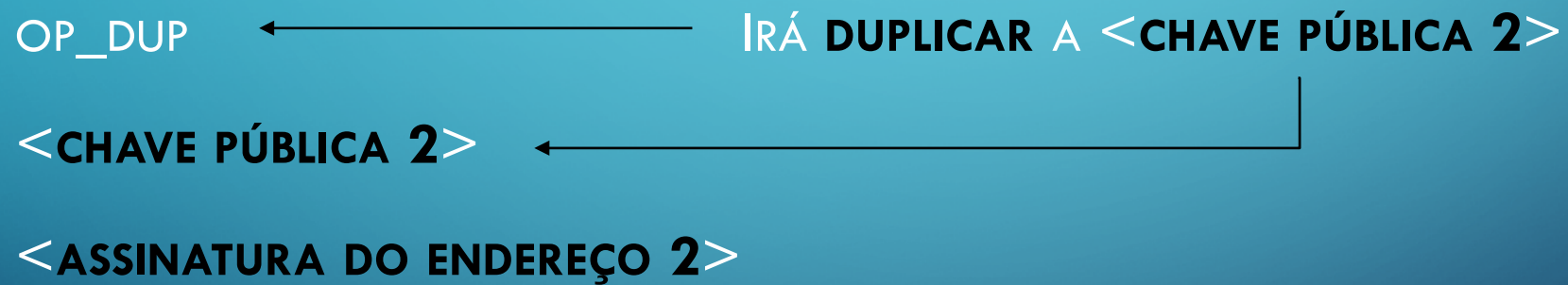
○ SEGUINTE SCRIPT DEVERÁ SER PROCESSADO COMO UMA PILHA: **<ASSINATURA DO ENDEREÇO 2> <CHAVE PÚBLICA 2> OP\_DUP OP\_HASH160 <HASH DA CHAVE PÚBLICA 2> OP\_EQUALVERIFY OP\_CHECKSIG**

# EXEMPLO DE TRANSAÇÃO COM O BITCOIN SCRIPT

OP\_DUP   ←   IRÁ DUPLICAR A <CHAVE PÚBLICA 2>

<CHAVE PÚBLICA 2>   ←

<ASSINATURA DO ENDEREÇO 2>



# EXEMPLO DE TRANSAÇÃO COM O BITCOIN SCRIPT

**<CHAVE PÚBLICA 2>**

**<CHAVE PÚBLICA 2>**

**<ASSINATURA DO ENDEREÇO 2>**



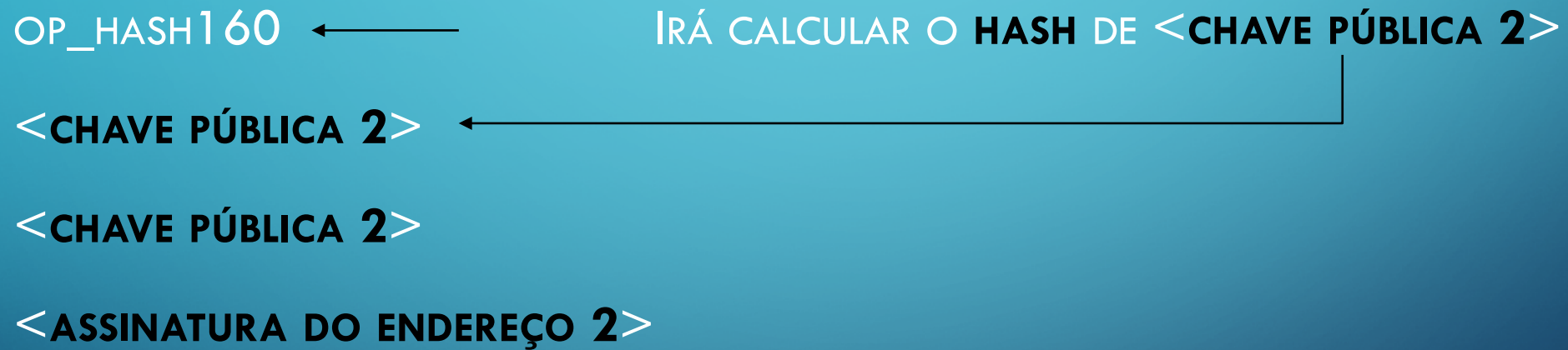
# EXEMPLO DE TRANSAÇÃO COM O BITCOIN SCRIPT

OP\_HASH160 ← IRÁ CALCULAR O HASH DE <CHAVE PÚBLICA 2>

<CHAVE PÚBLICA 2>

<CHAVE PÚBLICA 2>

<ASSINATURA DO ENDEREÇO 2>



# EXEMPLO DE TRANSAÇÃO COM O BITCOIN SCRIPT

**<HASH DA CHAVE PÚBLICA 2>**

**<CHAVE PÚBLICA 2>**

**<ASSINATURA DO ENDEREÇO 2>**

# EXEMPLO DE TRANSAÇÃO COM O BITCOIN SCRIPT

**<HASH DA CHAVE PÚBLICA 2>**

**<HASH DA CHAVE PÚBLICA 2>**

**<CHAVE PÚBLICA 2>**

**<ASSINATURA DO ENDEREÇO 2>**



# EXEMPLO DE TRANSAÇÃO COM O BITCOIN SCRIPT

OP\_EQUALVERIFY ← IRÁ VERIFICAR SE OS **DOIS ELEMENTOS DO TOPO SÃO IGUAIS**

<HASH DA CHAVE PÚBLICA 2>

<HASH DA CHAVE PÚBLICA 2>

<CHAVE PÚBLICA 2>

<ASSINATURA DO ENDEREÇO 2>

# EXEMPLO DE TRANSAÇÃO COM O BITCOIN SCRIPT

OP\_EQUALVERIFY ← SE FOREM **IGUAIS**, SERÃO **REMOVIDOS** DA PILHA

<HASH DA CHAVE PÚBLICA 2>

<HASH DA CHAVE PÚBLICA 2>

<CHAVE PÚBLICA 2>

<ASSINATURA DO ENDEREÇO 2>

# EXEMPLO DE TRANSAÇÃO COM O BITCOIN SCRIPT

OP\_EQUALVERIFY ← SE FOREM **DIFERENTES**, O SCRIPT É **INVALIDADO**

<HASH DA CHAVE PÚBLICA 2>

<HASH DA CHAVE PÚBLICA 2>

<CHAVE PÚBLICA 2>

<ASSINATURA DO ENDEREÇO 2>



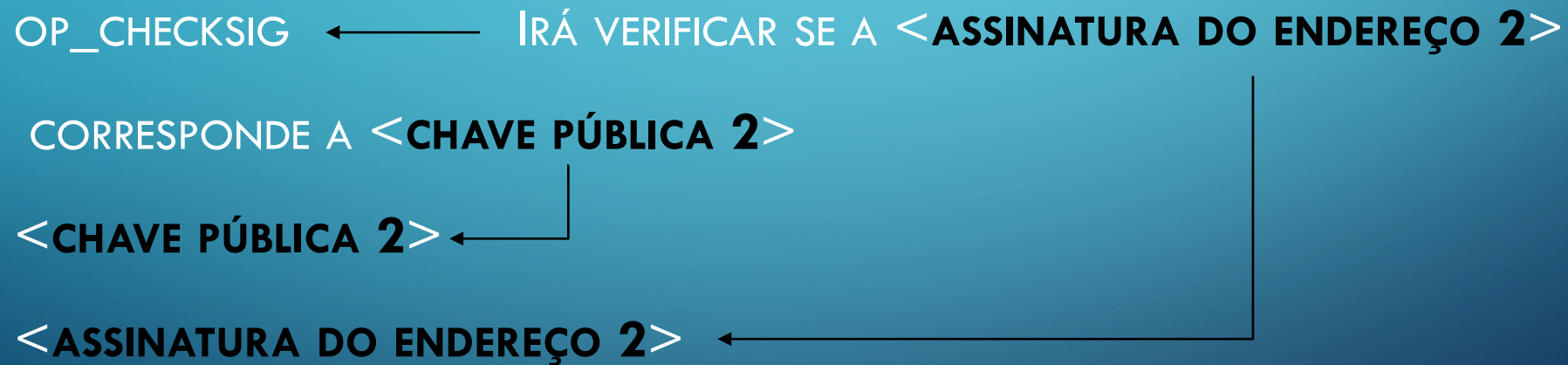
# EXEMPLO DE TRANSAÇÃO COM O BITCOIN SCRIPT

**<CHAVE PÚBLICA 2>**

**<ASSINATURA DO ENDEREÇO 2>**

# EXEMPLO DE TRANSAÇÃO COM O BITCOIN SCRIPT

OP\_CHECKSIG ← IRÁ VERIFICAR SE A **<ASSINATURA DO ENDEREÇO 2>**  
CORRESPONDE A **<CHAVE PÚBLICA 2>**  
**<CHAVE PÚBLICA 2>** ←  
**<ASSINATURA DO ENDEREÇO 2>** ←



# EXEMPLO DE TRANSAÇÃO COM O BITCOIN SCRIPT

OP\_CHECKSIG ← ○ SCRIPT É **VALIDADO** APENAS SE HOUVER A  
**CORRESPONDÊNCIA** ENTRE A CHAVE PÚBLICA E A ASSINATURA

<**CHAVE PÚBLICA 2**> ←

<**ASSINATURA DO ENDEREÇO 2**> ←



# SOLIDITY

APESAR DA REVOLUÇÃO TRAZIDA PELO BITCOIN, FOI O **ETHEREUM** QUE POPULARIZOU O CONCEITO DE **CONTRATOS INTELIGENTES**.

# SOLIDITY

○ 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.



# SOLIDITY

AS **TRANSAÇÕES** NA BLOCKCHAIN ETHEREUM SÃO REALIZADAS ATRAVÉS DOS **CONTRATOS INTELIGENTES**, QUE FUNCIONAM COMO **MEDIADORES** ENTRE AS ENTIDADES EXTERNAS.

# SOLIDITY

Entidade Externa 1



Contrato



Entidade Externa 2



Entidade Externa 1 envia 20  
para o Contrato



Contrato envia 20 para  
a Entidade Externa 2, se ela  
atender as condições...



# EXEMPLO DE CÓDIGO NO SOLIDITY

```
CONTRACT SENDER {
```

```
    FUNCTION SENDETH(ADDRESS PAYABLE RECEIVER) PUBLIC PAYABLE {
```

```
        RECEIVER.TRANSFER(MSG.VALUE);
```

```
    }
```

```
}
```



# EXEMPLO DE TRANSAÇÃO COM SOLIDITY

Entidade Externa 1



Contrato



Entidade Externa 2



Chama o método **sendETH()**,  
passando o endereço de  
entidade Externa 2 como  
parâmetro

Envia o valor recebido para  
Entidade Externa 2, através  
do método **transfer(msg.value)**

# O NOSSO PROJETO

**ESTENDER A LINGUAGEM ORIENTADA A OBJETOS 2** 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

A **BLOCKCHAIN** SERÁ REPRESENTADA POR UM **HASHMAP** CONTENDO UMA LISTA DE **ENDEREÇOS** E SEUS RESPECTIVOS **SALDOS**.

# O NOSSO PROJETO

A LINGUAGEM CONTARÁ COM **COMANDOS ADICIONAIS** DE ENTRADA/SAÍDA (**IO**), A FIM DE VIABILIZAR A **LEITURA E ESCRITA** NA ABSTRAÇÃO DA **BLOCKCHAIN**.

# O NOSSO PROJETO

UM **CONTRATO** DEVERÁ SER **DECLARADO** COMO UM TIPO DE CLASSE, COM O COMANDO **CLASSE CONTRATO**.

# O NOSSO PROJETO

UM **ENDEREÇO** DO TIPO **ENTIDADE EXTERNA** SERÁ 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 **ENDereco** (DO TIPO INTEIRO), QUE ARMAZENARÁ O SEU NÚMERO IDENTIFICADOR NA BLOCKCHAIN.

# O NOSSO PROJETO

TODO **CONTRATO** TERÁ UM **ATRIBUTO** CHAMADO **SALDO** (DO TIPO INTEIRO), QUE ARMAZENARÁ O SEU SALDO NA BLOCKCHAIN.

# O NOSSO PROJETO

OS **MÉTODOS** PODERÃO DEFINIDOS COMO PAGOS (**PROC PAGAVEL**) OU NÃO PAGOS (**PROC**).



# O NOSSO PROJETO

UM MÉTODO PAGO (**PROC PAGAVEL**) EXIGIRÁ 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 (ORIGEM E VALOR)**, NA SEGUINTE ORDEM: NOME DO MÉTODO (INT ORIGEM, INT VALOR);

# GRAMÁTICA

$\langle \text{PROGRAMA} \rangle ::= \{ \langle \text{DECLARACAO} \rangle ; \langle \text{COMANDO} \rangle \}$

$\langle \text{DECLARACAO} \rangle ::= \langle \text{DECLASSE} \rangle \mid \langle \text{DECLASSE} \rangle , \langle \text{DECLARACAO} \rangle$

$\langle \text{DECLASSE} \rangle ::= \langle \text{DECLASSETRADICIONAL} \rangle \mid \langle \textbf{DECLASSECONTRATO} \rangle$

# GRAMÁTICA

$\langle \text{COMANDO} \rangle ::= \langle \text{ATRIBUICAO} \rangle \mid \langle \text{COMDECLARACAO} \rangle \mid \langle \text{WHILE} \rangle \mid \langle \text{IFTHEELSE} \rangle$   
 $\mid \langle \text{IO} \rangle \mid \langle \text{SKIP} \rangle \mid \langle \text{NEW} \rangle \mid \langle \text{CHAMADAMETODO} \rangle \mid \langle \text{COMANDO} \rangle ";"$   
 $\langle \text{COMANDO} \rangle$



# GRAMÁTICA

$\langle \text{IO} \rangle ::= \text{"WRITE" "("} \langle \text{EXPRESSAO} \rangle \text{"} \mid \text{"READ" "("} \langle \text{ID} \rangle \text{"} \mid$   
 $\text{"ESCREVERNABLOCKCHAIN" "("} \langle \text{EXPRESSAO} \rangle \text{"} \mid \text{"LERDABLOCKCHAIN" "("}$   
 $\langle \text{EXPRESSAO} \rangle \text{"} \mid \text{"TRANSFERIR" "("} \langle \text{EXPRESSAO} \rangle \text{"}$

# GRAMÁTICA

$\langle \text{DECLASSETRADICIONAL} \rangle ::= \text{"CLASSE"} \text{ ID } [\text{"EXTENDS"} \text{ } \langle \text{ID} \rangle] \{ \text{" } \langle \text{DECVARIABEL} \rangle \text{ "};$   
 $\langle \text{DECCONSTRUTOR} \rangle \text{ "}, \text{" } \langle \textbf{DECPROCEDIMENTOTRADICIONAL} \rangle \text{ "}"$

# GRAMÁTICA

$\langle \text{DecClasseContrato} \rangle ::= \text{"CLASSE" "CONTRATO" } \langle \text{Id} \rangle \{ \langle \text{DecVariavel} \rangle \text{" ;" } \langle \text{DecConstrutor} \rangle \text{" ,"} \langle \text{DecProcedimento} \rangle \text{" } \}$

# GRAMÁTICA

$\langle \text{DecPROCEDIMENTO} \rangle ::= \langle \text{DecPROCEDIMENTOTradicional} \rangle \mid$   
 $\langle \text{DecPROCEDIMENTOPagavel} \rangle \mid \langle \text{DecPROCEDIMENTO} \rangle ", " \langle \text{DecPROCEDIMENTO} \rangle$



# GRAMÁTICA

$\langle \text{DECPROCEDIMENTO TRADICIONAL} \rangle ::= \text{"PROC"} \langle \text{ID} \rangle \text{"("}$   
 $\langle \text{LISTA DECLARACAO PARAMETRO} \rangle \text{")"} \text{"{"} \langle \text{COMANDO} \rangle \text{"}"}$

# GRAMÁTICA

**<DecPROCEDIMENTOPagavel>** ::= "PROC" "Pagavel" <Id> "("  
**<ParametrosPagaveis>** [<ListaDeclaracaoParametro> "]" ")" "{" <Comando> "}"

# ESQUELETO DE UM CÓDIGO DE EXEMPLO

```
{  
  
    CLASSE CONTRATO BANCO {  
        // DECLARAÇÃO DA CLASSE BANCO.  
  
    };  
  
    {  
        // COMANDOS...  
    }  
}
```

# DECLARAÇÃO DA CLASSE BANCO – PARTE 1

```
INT SINTAXE = 0;
```

```
BANCO(INT ENDERECO){
```

```
    THIS.ENDERECO := ENDERECO
```

```
},
```



# DECLARAÇÃO DA CLASSE BANCO – PARTE 2

```
PROC PAGAVEL RECEBER(INT ORIGEM, INT VALOR) {  
    TRANSFERIR(ORIGEM, THIS.ENDERECO, VALOR)  
},
```

## DECLARAÇÃO DA CLASSE BANCO – PARTE 3

```
PROC GETSALDO() {  
    LERDABLOCKCHAIN(THIS.ENDereco)  
  
},  
  
PROC GETENDereco() {  
    WRITE("ENDEREÇO: " ++ THIS.ENDereco)  
  
}
```

# COMANDOS – PARTE 1

```
BANCO B1 := NEW BANCO(9999), BANCO B2 := NEW BANCO(9998);
```

```
WRITE("ENDEREÇO E SALDO DE B1: ");
```

```
B1.GETENDereco();
```

```
B1.GETSALDO();
```

## COMANDOS – PARTE 2

```
WRITE("ENDEREÇO E SALDO DE B2: ");
```

```
B2.GETENDereco();
```

```
B2.GETSALDO();
```

```
B1.RECEBER(9998,25);
```



## COMANDOS – PARTE 3

```
WRITE("ENDEREÇO E SALDO DE B1: ");
```

```
B1.GETENDereco();
```

```
B1.GETSALDO();
```

```
WRITE("ENDEREÇO E SALDO DE B2: ");
```

```
B2.GETENDereco();
```

```
B2.GETSALDO();
```

# DÚVIDAS?

The background is a blue gradient. In the corners, there are decorative white line art elements resembling circuit boards or neural networks, with lines and small circles.

# OBRIGADO!