El futuro digital
es de todos

MinTIC

Blockchain & analítica de datos

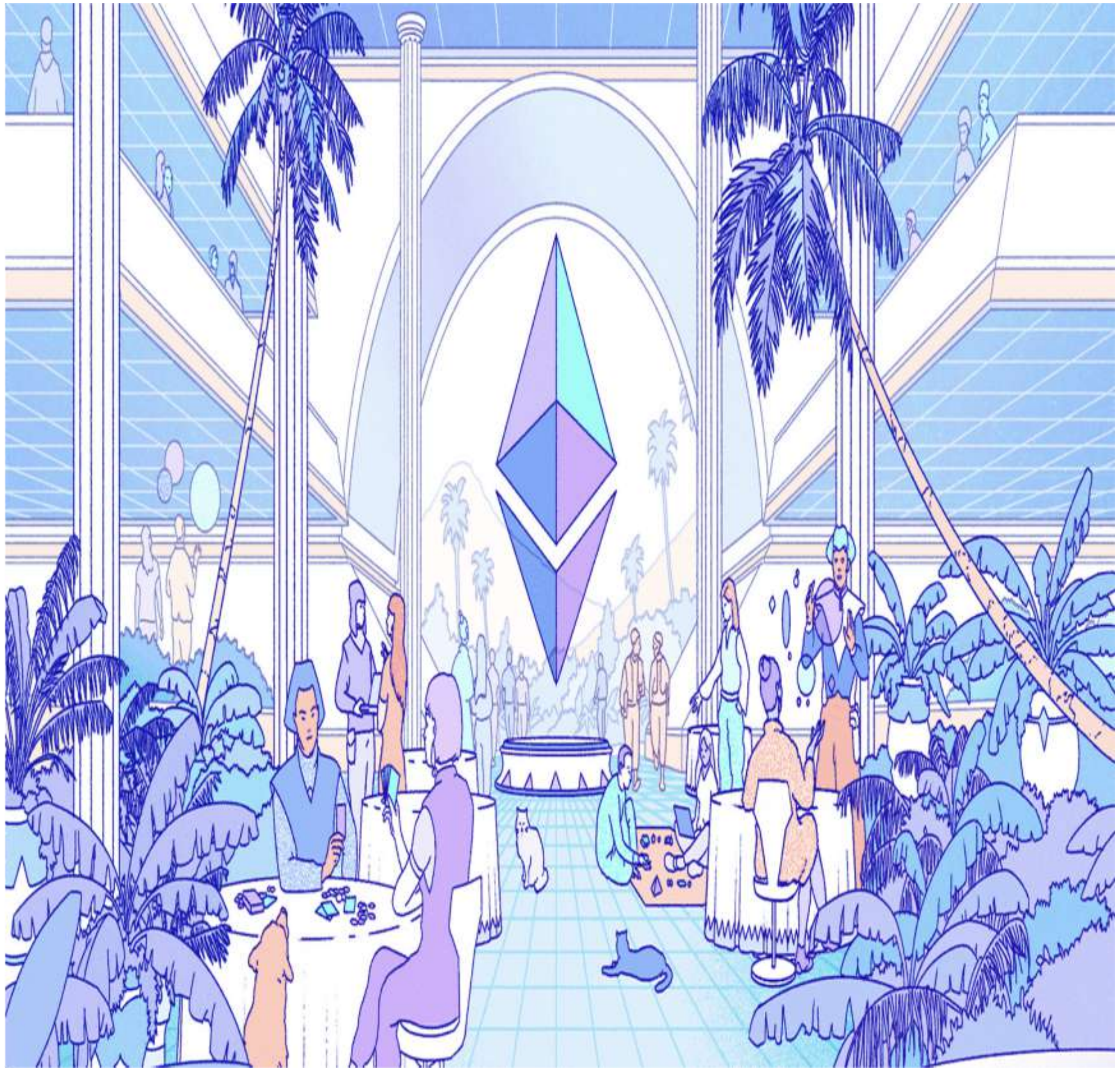
para industrias digitales

etraining
Diseñamos nuevas formas de aprender

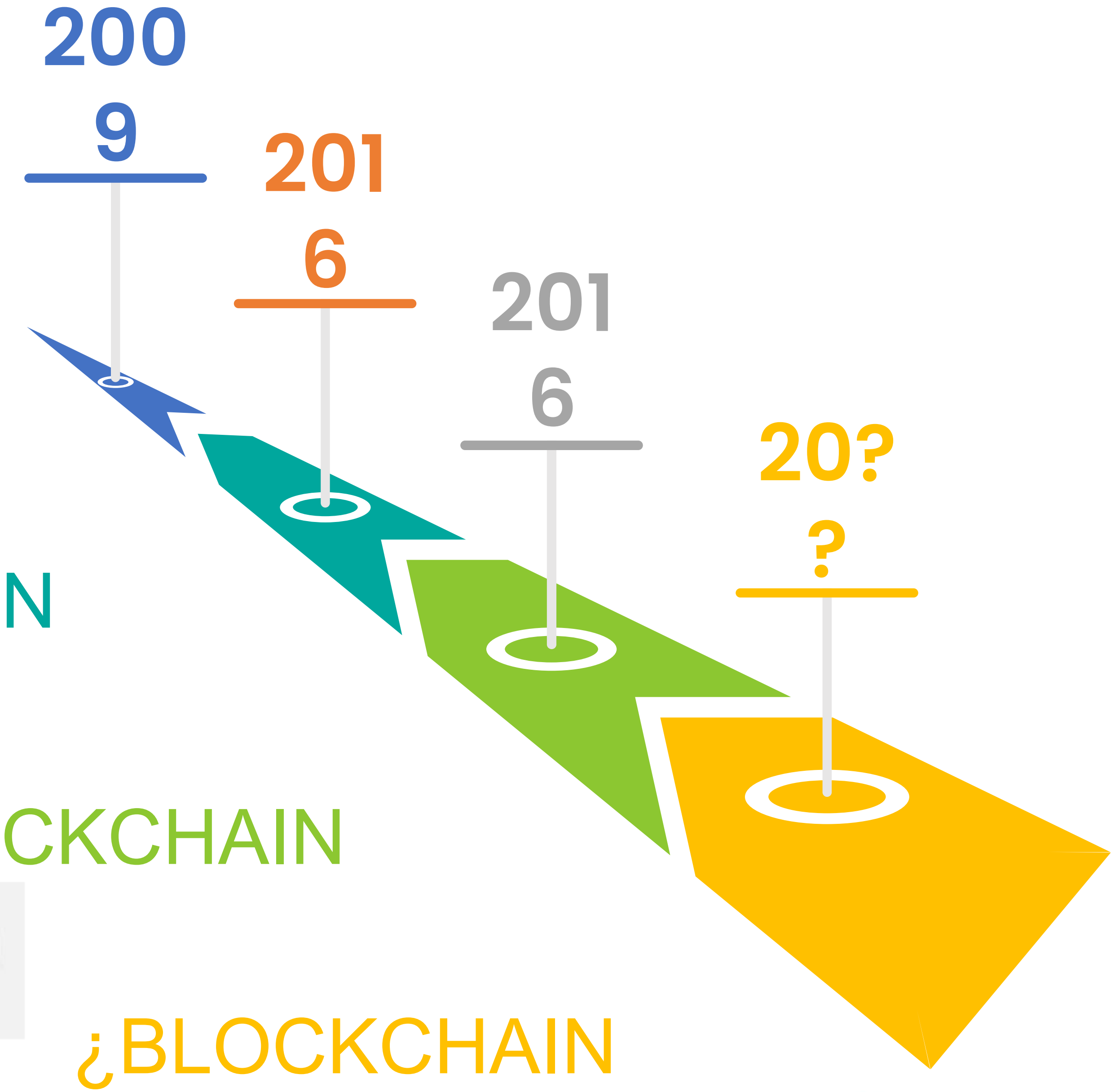
tecnalia Colombia
Inspiring Business

Lo que veremos hoy:

- ❑ ETHEREUM
- ❑ CONTRATOS INTELIGENTES (SC)
- ❑ DESPLEGANDO NUESTRO 2DO SC



1. EVOLUCIÓN
2. INICIOS
3. DEFINICIÓN
4. ETHEREUM 1.0/2.0



BLOCKCHAIN

BLOCKCHAIN

BLOCKCHAIN 3

IOTA

¿BLOCKCHAIN 4.0?

You read my t-shirt.
THAT'S ENOUGH SOCIAL INTERACTION FOR ONE DAY.

- **VITALIK BUTERIN: PROGRAMADOR RUSO-CANADIENSE.**

- **CO-FUNDADOR DE ETHEREUM CUANDO TENÍA 19 AÑOS DE EDAD.**

- **TRABAJÓ EN LA REVISTA BITCOIN MAGAZINE.**

MIHAI ALISIE

ANTHONY DI IORIO

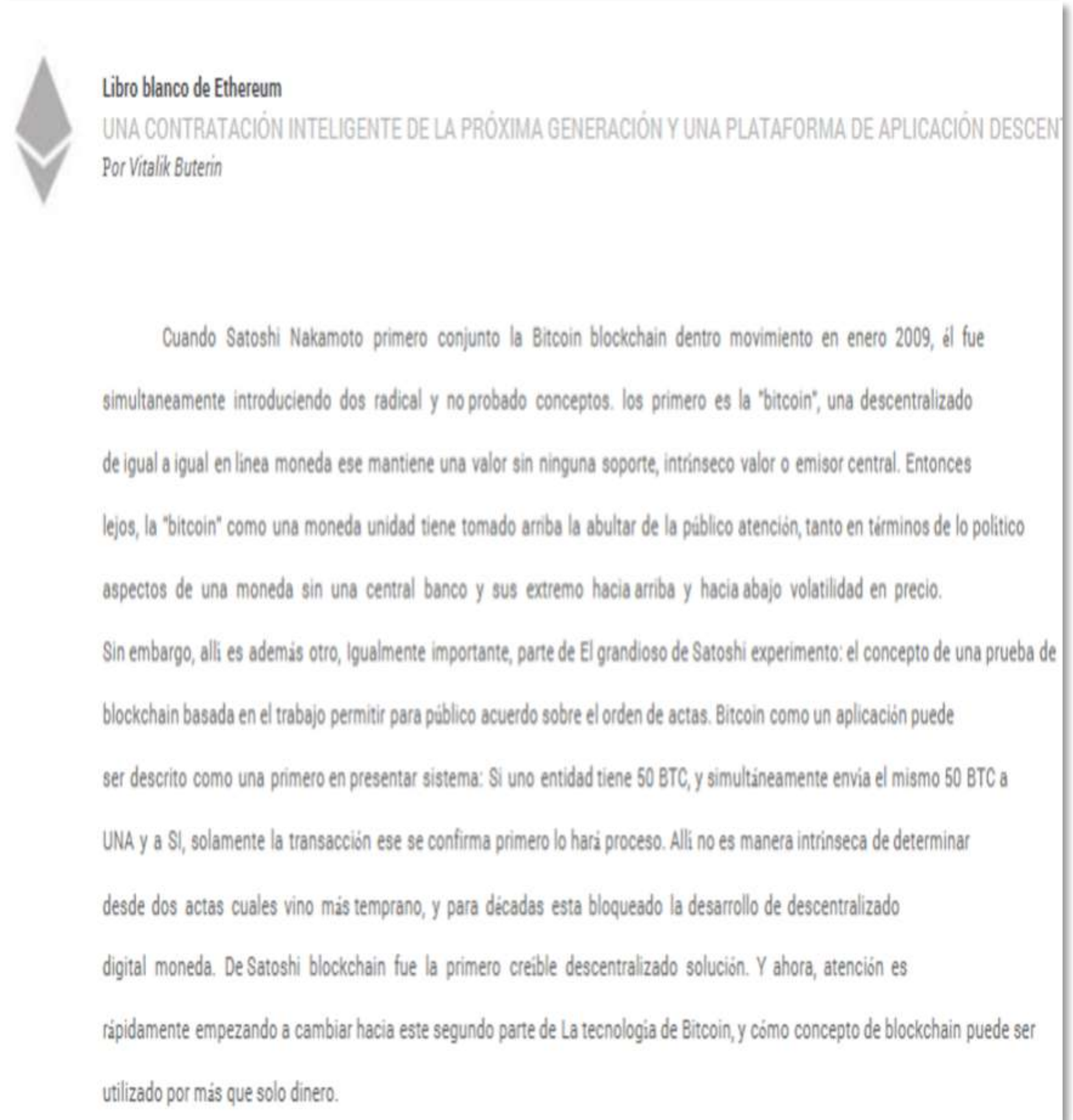
CHARLES HOSKINSON

JOE LUBIN

GAVIN WOOD

EL PROYECTO FUE ANUNCIADO PÚBLICAMENTE EN ENERO DE 2014, CON EL EQUIPO CENTRAL FORMADO POR VITALIK BUTERIN Y …

https://thehistoryofethereum.com/



Libro blanco de Ethereum

UNA CONTRATACIÓN INTELIGENTE DE LA PRÓXIMA GENERACIÓN Y UNA PLATAFORMA DE APLICACIÓN DESCEN

Por Vitalik Buterin

Cuando Satoshi Nakamoto primero conjunto la Bitcoin blockchain dentro movimiento en enero 2009, él fue simultaneamente introduciendo dos radical y no probado conceptos. los primero es la "bitcoin", una descentralizado de igual a igual en línea moneda ese mantiene una valor sin ninguna soporte, intrínseco valor o emisor central. Entonces lejos, la "bitcoin" como una moneda unidad tiene tomado arriba la abultar de la público atención, tanto en términos de lo político aspectos de una moneda sin una central banco y sus extremo hacia arriba y hacia abajo volatilidad en precio.

Sin embargo, allí es además otro, Igualmente importante, parte de El grandioso de Satoshi experimento: el concepto de una prueba de blockchain basada en el trabajo permitir para público acuerdo sobre el orden de actas. Bitcoin como un aplicación puede ser descrito como una primero en presentar sistema: Si uno entidad tiene 50 BTC, y simultáneamente envía el mismo 50 BTC a UNA y a SI, solamente la transacción ese se confirma primero lo hará proceso. Allí no es manera intrínseca de determinar desde dos actas cuales vino más temprano, y para décadas esta bloqueado la desarrollo de descentralizado digital moneda. De Satoshi blockchain fue la primero creíble descentralizado solución. Y ahora, atención es rápidamente empezando a cambiar hacia este segundo parte de La tecnología de Bitcoin, y cómo concepto de blockchain puede ser utilizado por más que solo dinero.

EN JULIO DE 2014 ETHEREUM LLEVÓ A CABO LA PRIMERA INITIAL COIN OFFERING (ICO) PARA ETH ZURICH.

LA FUNDACIÓN ETHEREUM PUDO GANAR UN TOTAL DE 31.591 BITCOINS POR VALOR DE MÁS DE 18 MILLONES DE DÓLARES.

2016 LANZAMIENTO OFICIAL (marzo)

2015 PRIMERA VERSIÓN (junio)

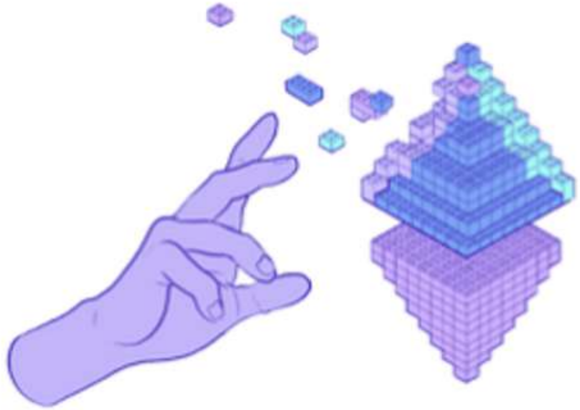
2014 **FUNDACIÓN ETHEREUM** (setiembre)

2014 **CROWDFUNDING** (julio)

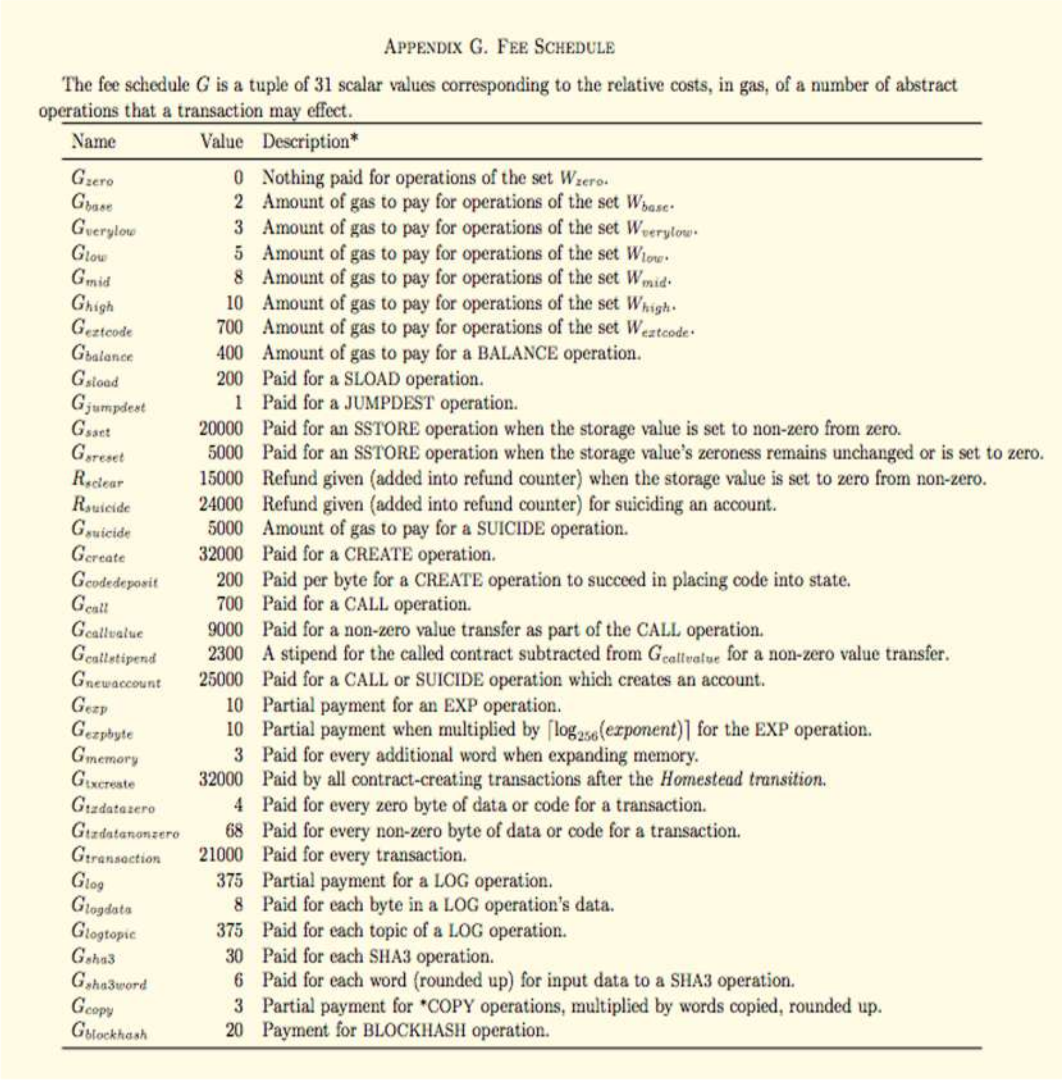
2013 **IDEA** (octubre)

❑DEFINICIÓN

“ES UNA PLATAFORMA DE SOFTWARE OPEN SOURCE CONSTRUIDA SOBRE TECNOLOGÍA BLOCKCHAIN QUE PERMITE A LOS DESARROLLADORES CONSTRUIR E IMPLEMENTAR APLICACIONES DESCENTRALIZADAS.”



❑ ¿CÓMO FUNCIONA?

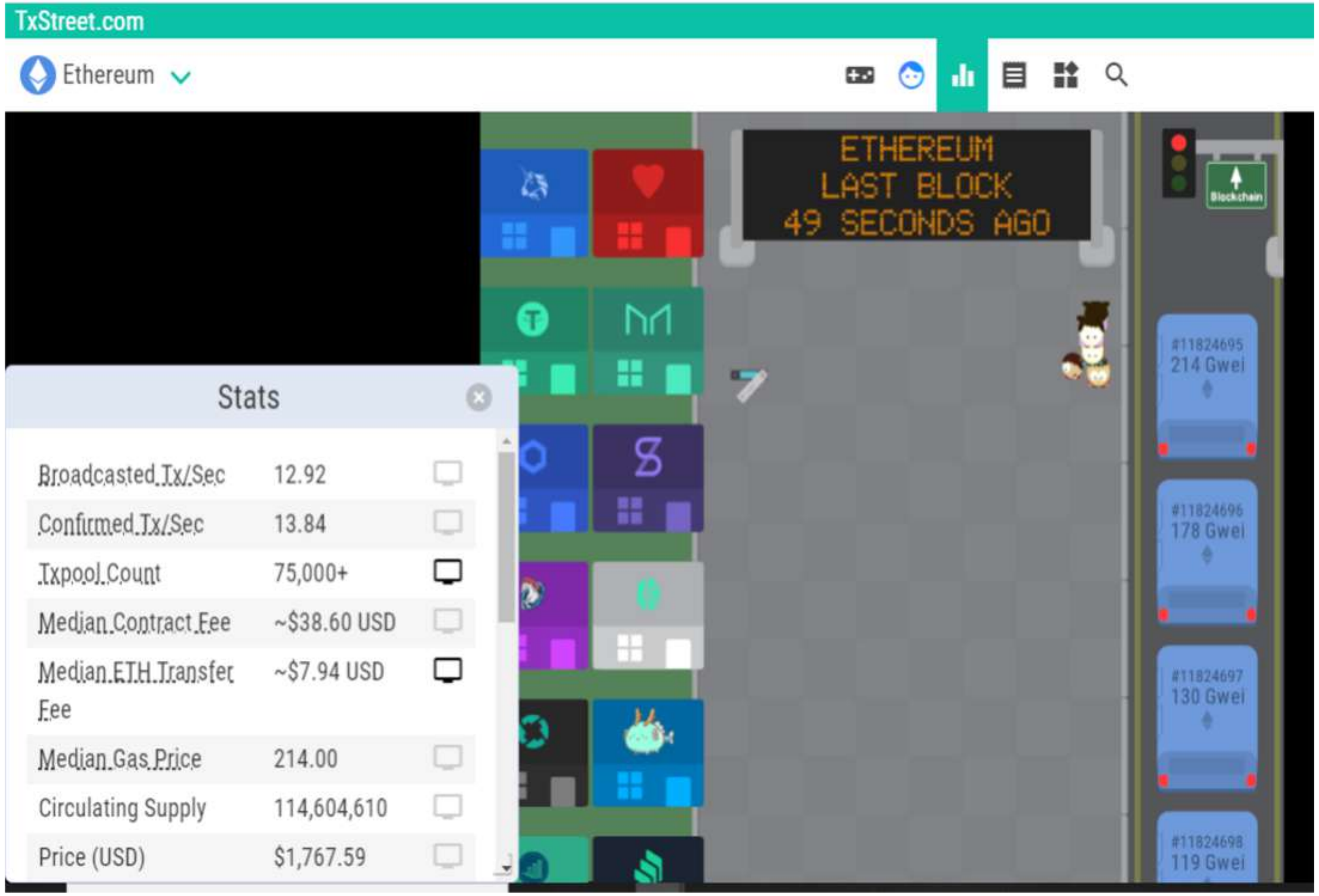


Appendix G. Fee Schedule

The fee schedule G is a tuple of 31 scalar values corresponding to the relative costs, in gas, of a number of abstract operations that a transaction may effect.

Name	Value	Description*
G_{zero}	0	Nothing paid for operations of the set W_{zero}.
G_{base}	2	Amount of gas to pay for operations of the set W_{base}.
$G_{verylow}$	3	Amount of gas to pay for operations of the set $W_{verylow}$.
G_{low}	5	Amount of gas to pay for operations of the set W_{low}.
G_{mid}	8	Amount of gas to pay for operations of the set W_{mid}.
G_{high}	10	Amount of gas to pay for operations of the set W_{high}.
$G_{extcode}$	700	Amount of gas to pay for operations of the set $W_{extcode}$.
$G_{balance}$	400	Amount of gas to pay for a BALANCE operation.
G_{sload}	200	Paid for a SLOAD operation.
$G_{jumpdest}$	1	Paid for a JUMPDEST operation.
G_{sset}	20000	Paid for an SSTORE operation when the storage value is set to non-zero from zero.
G_{sreset}	5000	Paid for an SSTORE operation when the storage value's zeroness remains unchanged or is set to zero.
R_{sclear}	15000	Refund given (added into refund counter) when the storage value is set to zero from non-zero.
$R_{suicide}$	24000	Refund given (added into refund counter) for suiciding an account.
$G_{suicide}$	5000	Amount of gas to pay for a SUICIDE operation.
G_{create}	32000	Paid for a CREATE operation.
$G_{codedeposit}$	200	Paid per byte for a CREATE operation to succeed in placing code into state.
G_{call}	700	Paid for a CALL operation.
$G_{callvalue}$	9000	Paid for a non-zero value transfer as part of the CALL operation.
$G_{callstipend}$	2300	A stipend for the called contract subtracted from $G_{callvalue}$ for a non-zero value transfer.
$G_{newaccount}$	25000	Paid for a CALL or SUICIDE operation which creates an account.
G_{exp}	10	Partial payment for an EXP operation.
$G_{expbyte}$	10	Partial payment when multiplied by $\lceil \log_{256}(exponent) \rceil$ for the EXP operation.
G_{memory}	3	Paid for every additional word when expanding memory.
$G_{txcreate}$	32000	Paid by all contract-creating transactions after the *Homestead transition*.
$G_{txdatazero}$	4	Paid for every zero byte of data or code for a transaction.
$G_{txdatanonzero}$	68	Paid for every non-zero byte of data or code for a transaction.
$G_{transaction}$	21000	Paid for every transaction.
G_{log}	375	Partial payment for a LOG operation.
$G_{logdata}$	8	Paid for each byte in a LOG operation's data.
$G_{logtopic}$	375	Paid for each topic of a LOG operation.
G_{sha3}	30	Paid for each SHA3 operation.
$G_{sha3word}$	6	Paid for each word (rounded up) for input data to a SHA3 operation.
G_{copy}	3	Partial payment for *COPY operations, multiplied by words copied, rounded up.
$G_{blockhash}$	20	Payment for BLOCKHASH operation.

https://txstreet.com/v/eth



Stats

Broadcasted Tx/Sec	12.92	☐
Confirmed Tx/Sec	13.84	☐
Txpool Count	75,000+	☐
Median Contract Fee	~$38.60 USD	☐
Median ETH Transfer Fee	~$7.94 USD	☐
Median Gas Price	214.00	☐
Circulating Supply	114,604,610	☐
Price (USD)	$1,767.59	☐

❑ ETHEREUM: 1.0

BLOCK # 0 - FRONTIER

Fase de desarrollo inicial 30 julio 20115 a marzo 2016.

BLOCK # 200.000 – ICE AGE

HARD FORK para introducir un incremento en la dificultad que incentive el cambio de PoW a PoS cuando este listo.

BLOCK # 1.500.000 - Homestead

Segunda fase lanzada en Marzo 2016.

BLOCK # 1.192.000 – DAO Hack

Hack en uno de los contratos del DAO. Se divide entre Ethereum y Ethereum Classic..

BLOCK # 2.463.000 – Tangenire Whstle

Otro Hard Fork para cambiar como se calcula el GAS debido a los ataques DoS.

BLOCK # 2.675.000 – Spurious Dragom

Otro Hard Fork para evitar más vectores de ataque de DoS y mecanismos de defensa contra ataques de replay.

BLOCK # 4.370.000 – Metrolopis Byzantium

Tercera fase del desarrollo de Ethereum – Octubre 2017. Primer Hard Fork de la fase Metropolis.

BLOCK # 7.280.000 - Constantinople

Segundo Hard Fork de la Fase Metropolis. Febrero 2019. Incluye cambios de seguridad.

BLOCK # 9.069.000 - Istanbul

Otro Hard Fork para enmendar problemas de seguridad e incentivos para moverse a PoS.

Serenity- ETHEREUM 2.0

La nueva Blockchain que introducirá PoS, una nueva EMV entre otros cambio.

PROBLEMAS

- **ESCALABILIDAD**
- **SOPORTA ENTRE 15-20 TPS MIENTRAS SISTEMA COMO VISA ALREDEDOR DE 45K**
- **PRECIO DEL GAS, EMPLEO DEL PoW, NO ES ATRACTIVO PARA “MINEROS”**

❑ ETHEREUM: 2.0

SERENITY

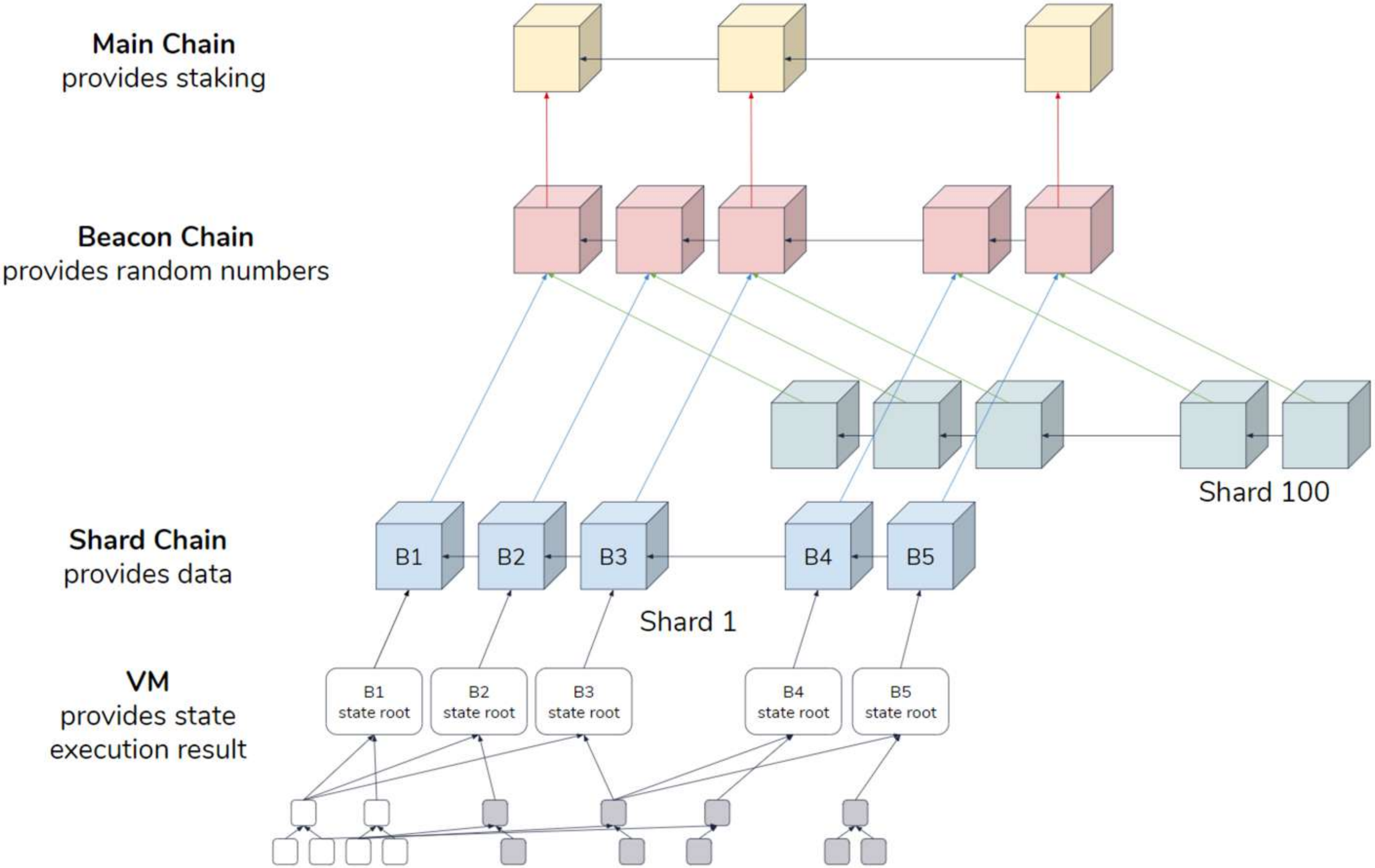
- FASE FINAL (?) DEL DESARROLLO DE ETHEREUM
- PROMETE ESCALAR ETHEREUM HACIÉNDOLA MAS RÁPIDA, MÁS SEGURA Y SIMPLEMENTE UNA MEJOR BLOCKCHAIN.

❑ ETHEREUM: 2.0

SERENITY

- UTILIZARÁ EL PROTOCOLO DE CONSENSO "PROOF OF STEAK (PoS)"
- INTRODUCIRÁ UNA NUEVA MOENADA "ETH 2"
- NUEVA ETHEREUM VIRTUAL MACHINE (EVM) Y OTRAS TECNOLOGÍAS

SHARDING



Main Chain
provides staking

Beacon Chain
provides random numbers

Shard Chain
provides data

VM
provides state
execution result

Shard 100
B1
B2
B3
B4
B5
Shard 1
B1 state root
B2 state root
B3 state root
B4 state root
B5 state root

¿POR QUÉ SHARDING

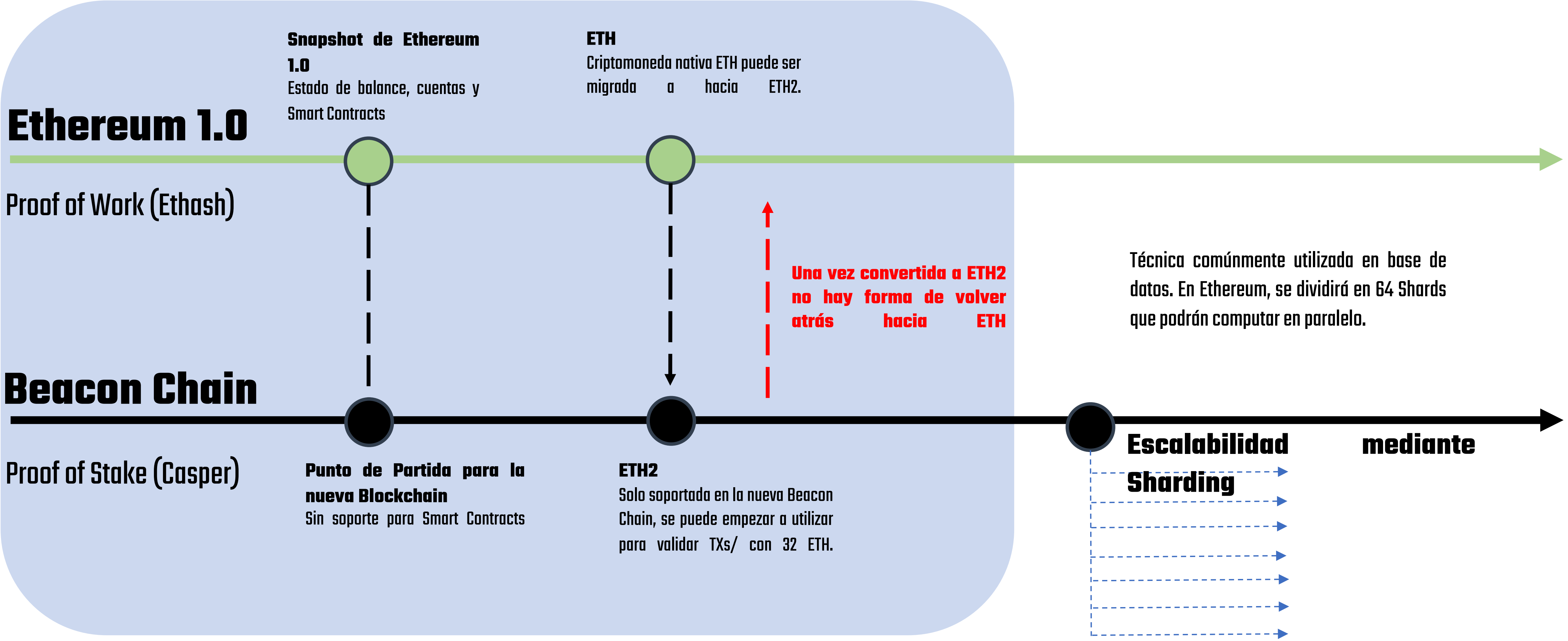
ESCALABILIDAD

VALIDADORES SOLO TENDRÁN QUE VERIFICAR Y VALIDAR TRANSACCIONES EN UN SHARD.

TODOS LOS SHARDS COMPARTIRÁN LA SEGURIDAD DE LA RED ETHEREUM

Fase 0 : Beacon Chain – Dic 1, 2020

Fase 1 : Shards Chains – 2021



Ethereum 1.0

Proof of Work (Ethash)

Snapshot de Ethereum 1.0
Estado de balance, cuentas y Smart Contracts

ETH
Criptomoneda nativa ETH puede ser migrada a hacia ETH2.

Una vez convertida a ETH2 no hay forma de volver atrás hacia ETH

Técnica comúnmente utilizada en base de datos. En Ethereum, se dividirá en 64 Shards que podrán computar en paralelo.

Beacon Chain

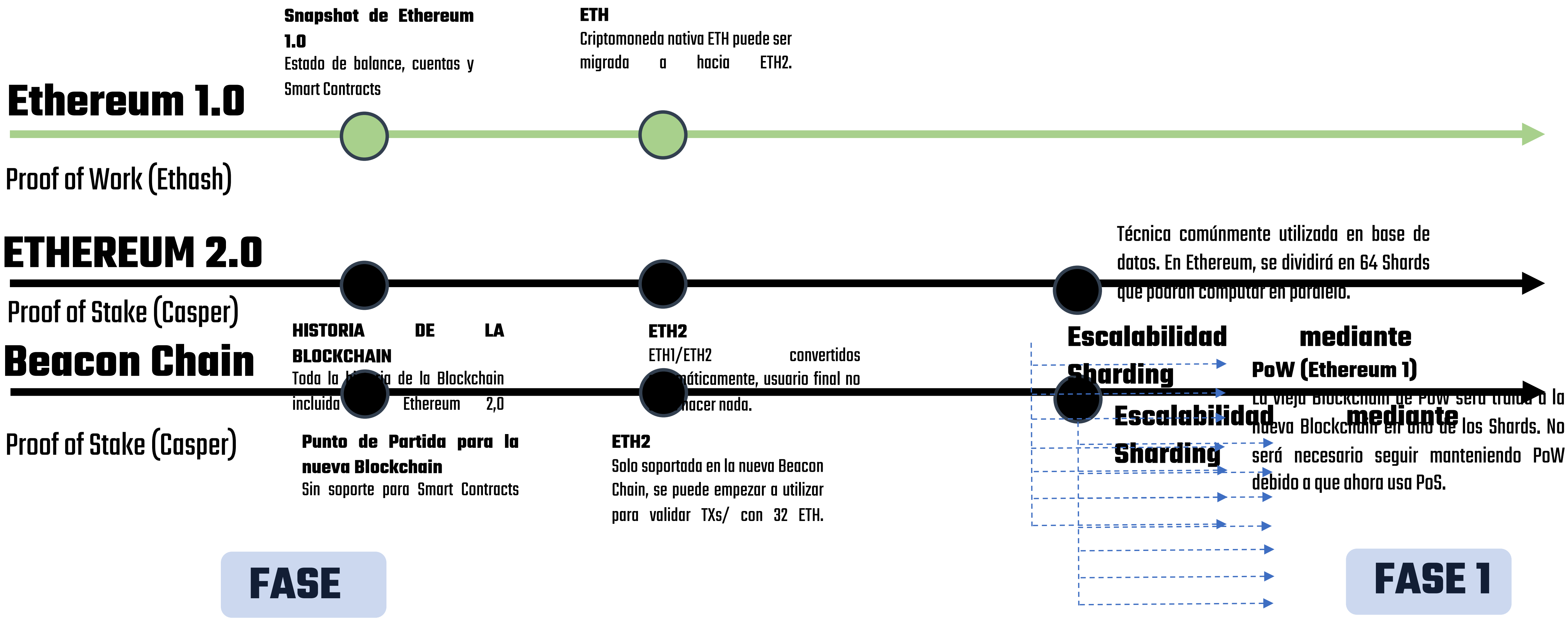
Proof of Stake (Casper)

Punto de Partida para la nueva Blockchain
Sin soporte para Smart Contracts

ETH2
Solo soportada en la nueva Beacon Chain, se puede empezar a utilizar para validar TXs/ con 32 ETH.

Escalabilidad mediante Sharding

Fase 1.5 : PoW & PoS Chain Merge – 2021 → 2022



Ethereum 1.0

Proof of Work (Ethash)

Snapshot de Ethereum 1.0
Estado de balance, cuentas y Smart Contracts

ETH
Criptomoneda nativa ETH puede ser migrada a hacia ETH2.

ETHEREUM 2.0

Proof of Stake (Casper)

HISTORIA DE LA BLOCKCHAIN
Toda la historia de la Blockchain incluida [illegible] Ethereum 2,0

ETH2
ETH1/ETH2 convertidos [illegible]máticamente, usuario final no [illegible] hacer nada.

Técnica comúnmente utilizada en base de datos. En Ethereum, se dividirá en 64 Shards que podrán computar en paralelo.

Escalabilidad mediante Sharding

Beacon Chain

Proof of Stake (Casper)

Punto de Partida para la nueva Blockchain
Sin soporte para Smart Contracts

ETH2
Solo soportada en la nueva Beacon Chain, se puede empezar a utilizar para validar TXs/ con 32 ETH.

Escalabilidad mediante Sharding

PoW (Ethereum 1)
La vieja Blockchain de PoW será traída a la nueva Blockchain en uno de los Shards. No será necesario seguir manteniendo PoW debido a que ahora usa PoS.

FASE

FASE 1

Fase 2 - Mejora continua - 2022

ETHEREUM 2.0

Proof of Stake (Casper)

HISTORIA DE LA BLOCKCHAIN
Toda la historia de la Blockchain incluida la Ethereum 2,0

ETH2
ETH1/ETH2 convertidos automáticamente, usuario final no debe hacer nada.

Escalabilidad mediante Sharding

PoW (Ethereum 1)
La vieja Blockchain de PoW será traída a la nueva Blockchain en uno de los Shards. No será necesario seguir manteniendo PoW debido a que ahora usa PoS.

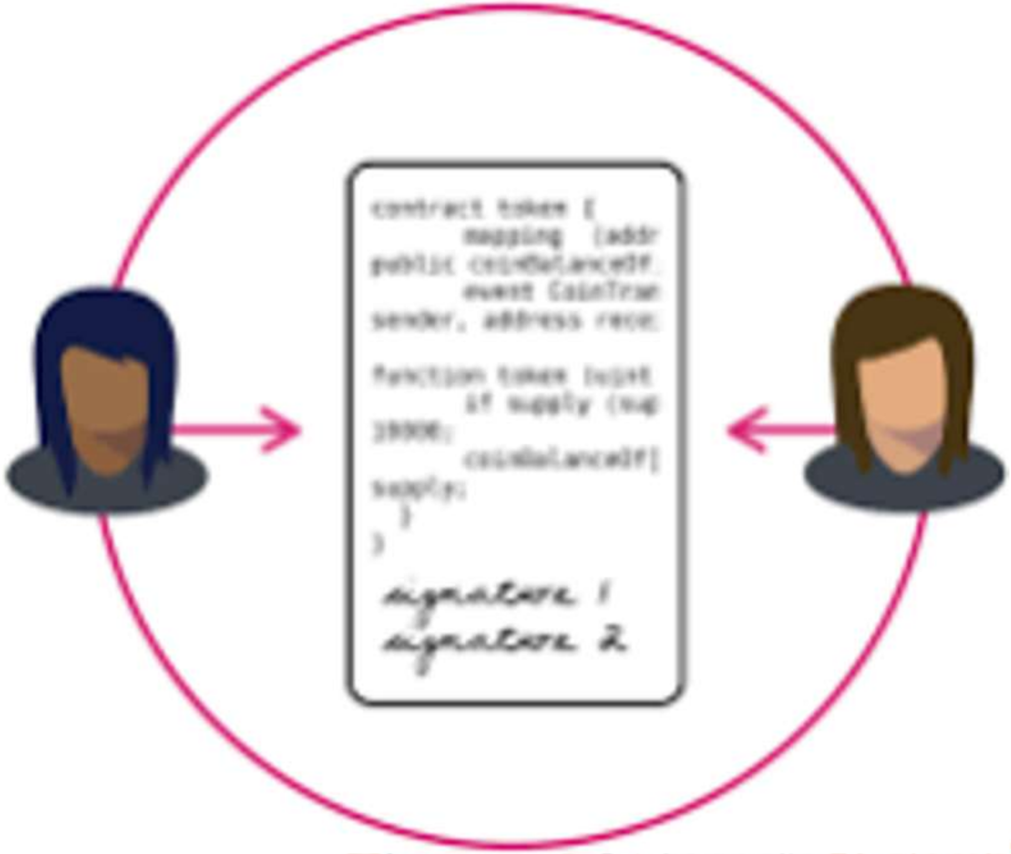
- Nueva Ethereum Virtual Machine (eWASM)
- Transacciones Cross Shard
- Lightweigth Clients
- Otras mejoras y soluciones que vayan apareciendo.

❑ CONTRATOS INTELIGENTES

1. ¿QUÉ ES?
2. ¿CÓMO FUNCIONA?
3. ¿CÓMO IMPLEMENTARLOS?
4. TALLER

❑ ¿QUÉ ES?

ES UN PROGRAMA INFORMÁTICO QUE EJECUTA ACUERDOS ESTABLECIDOS ENTRE DOS O MÁS PARTES HACIENDO QUE CIERTAS ACCIONES SUCEDAN COMO RESULTADO DE QUE SE CUMPLAN UNA SERIE DE CONDICIONES ESPECÍFICAS.

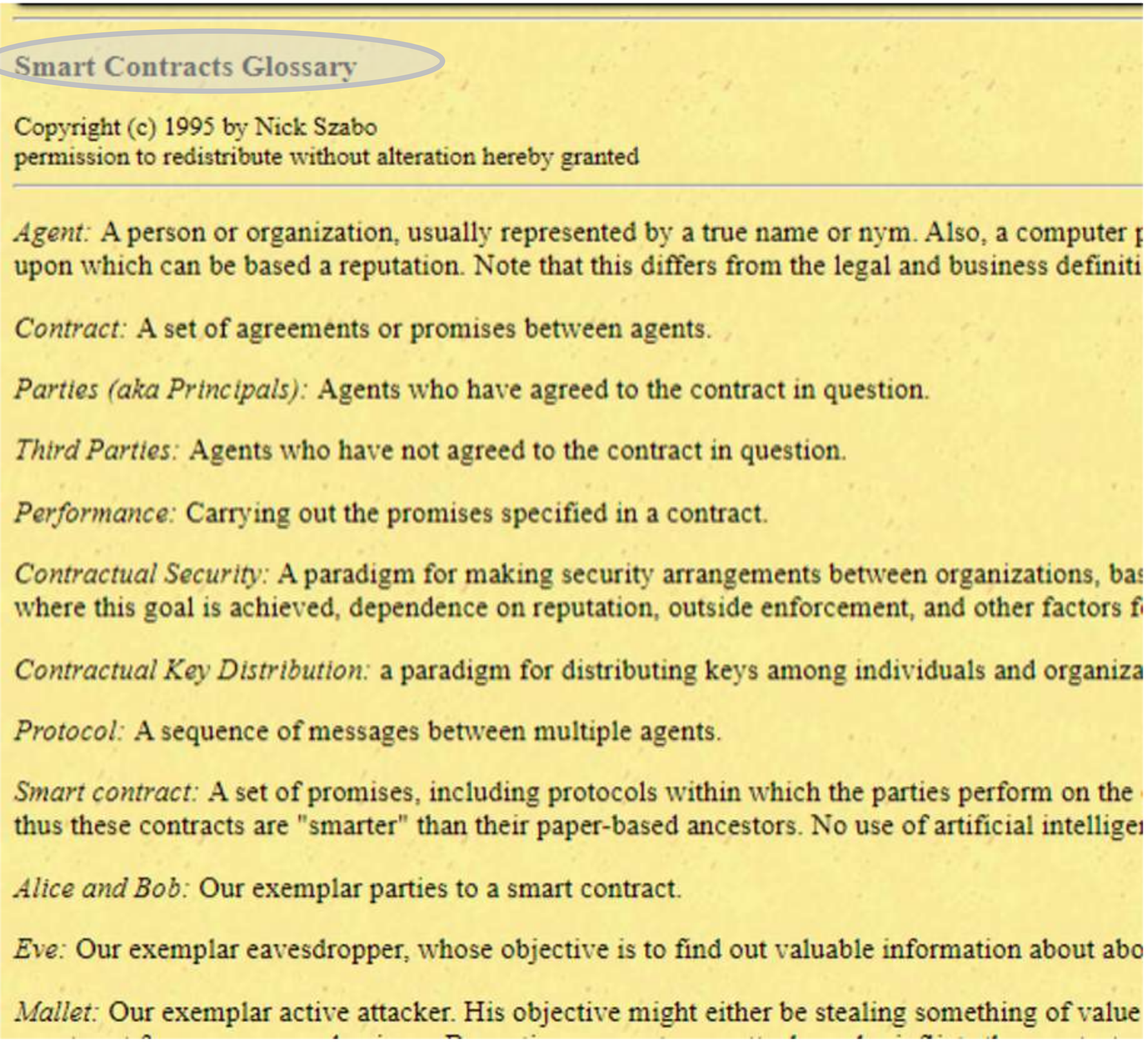


NICK SZABO
Blockchain, cryptocurrency, and smart contracts pioneer

LOS CONTRATOS INTELIGENTES LLEVAN DESARROLLÁNDOSE DESDE 1995, CUANDO EL FAMOSO CIENTÍFICO INFORMÁTICO, ABOGADO Y CRIPTÓGRAFO **NICK SZABO** ACUÑÓ EL TÉRMINO POR PRIMERA VEZ.

DOS AÑOS DESPUÉS, EN 1997, DESARROLLÓ UN DOCUMENTO MUCHO MÁS DETALLADO EXPLICANDO LOS SMART CONTRACTS.

http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/idea.html



Smart Contracts Glossary

Copyright (c) 1995 by Nick Szabo
permission to redistribute without alteration hereby granted

Agent: A person or organization, usually represented by a true name or nym. Also, a computer p
upon which can be based a reputation. Note that this differs from the legal and business definiti

Contract: A set of agreements or promises between agents.

Parties (aka Principals): Agents who have agreed to the contract in question.

Third Parties: Agents who have not agreed to the contract in question.

Performance: Carrying out the promises specified in a contract.

Contractual Security: A paradigm for making security arrangements between organizations, bas
where this goal is achieved, dependence on reputation, outside enforcement, and other factors f

Contractual Key Distribution: a paradigm for distributing keys among individuals and organiza

Protocol: A sequence of messages between multiple agents.

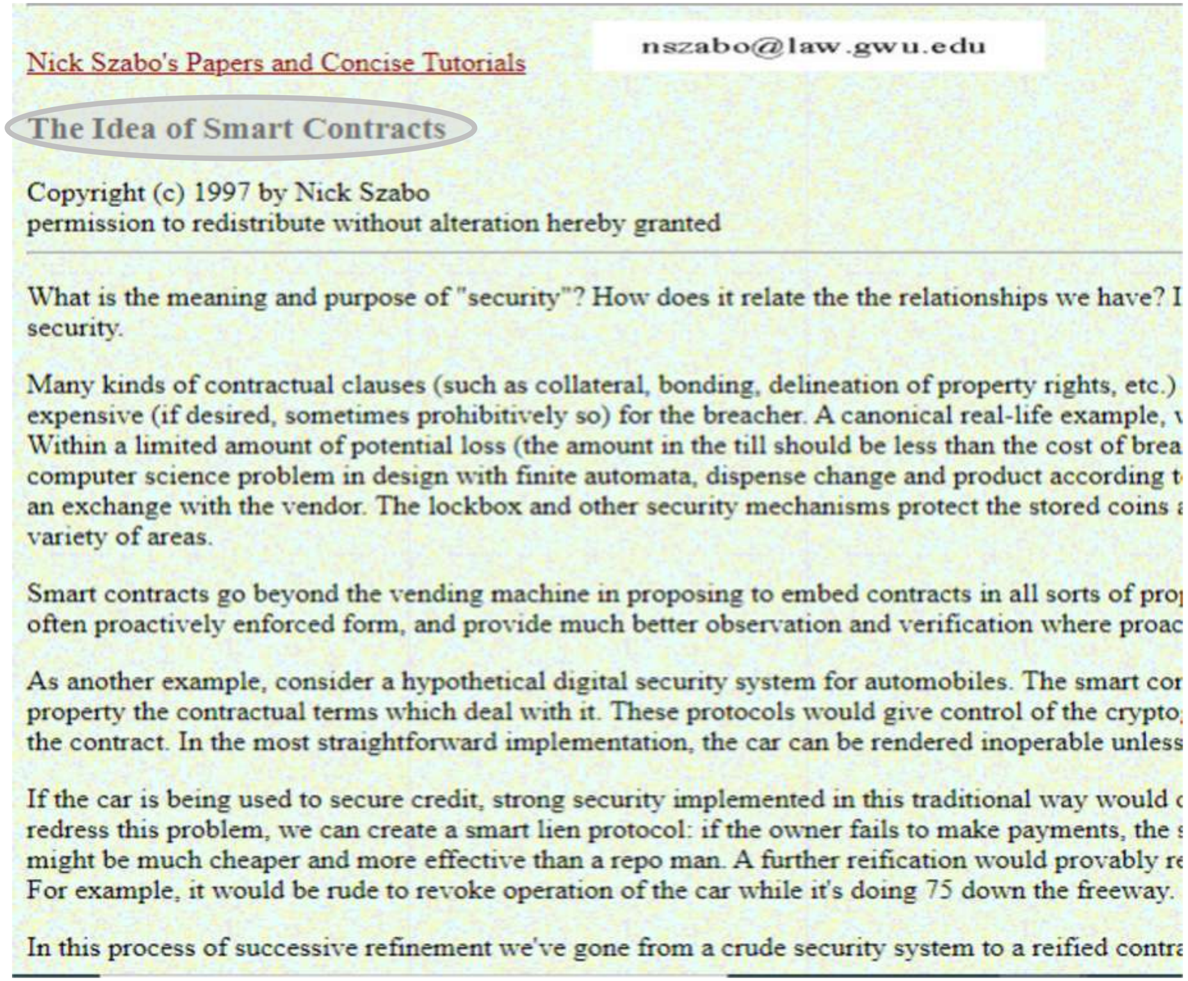
Smart contract: A set of promises, including protocols within which the parties perform on the
thus these contracts are "smarter" than their paper-based ancestors. No use of artificial intellige

Alice and Bob: Our exemplar parties to a smart contract.

Eve: Our exemplar eavesdropper, whose objective is to find out valuable information about abo

Mallet: Our exemplar active attacker. His objective might either be stealing something of value

http://web.archive.org/web/20160417212209/http://szabo.best.vwh.net/smart_contracts_glossary.html



Nick Szabo's Papers and Concise Tutorials

nszabo@law.gwu.edu

The Idea of Smart Contracts

Copyright (c) 1997 by Nick Szabo
permission to redistribute without alteration hereby granted

What is the meaning and purpose of "security"? How does it relate the the relationships we have? I
security.

Many kinds of contractual clauses (such as collateral, bonding, delineation of property rights, etc.)
expensive (if desired, sometimes prohibitively so) for the breacher. A canonical real-life example, v
Within a limited amount of potential loss (the amount in the till should be less than the cost of brea
computer science problem in design with finite automata, dispense change and product according t
an exchange with the vendor. The lockbox and other security mechanisms protect the stored coins a
variety of areas.

Smart contracts go beyond the vending machine in proposing to embed contracts in all sorts of pro
often proactively enforced form, and provide much better observation and verification where proac

As another example, consider a hypothetical digital security system for automobiles. The smart co
property the contractual terms which deal with it. These protocols would give control of the crypto
the contract. In the most straightforward implementation, the car can be rendered inoperable unless

If the car is being used to secure credit, strong security implemented in this traditional way would c
redress this problem, we can create a smart lien protocol: if the owner fails to make payments, the s
might be much cheaper and more effective than a repo man. A further reification would provably re
For example, it would be rude to revoke operation of the car while it's doing 75 down the freeway.

In this process of successive refinement we've gone from a crude security system to a reified contra

etraining Diseñamos nuevas formas de aprender | tecnalia Colombia Inspiring Business

SMART
CONTRACT

UN CONJUNTO DE PROMESAS, INCLUIDOS LOS PROTOCOLOS DENTRO DE LOS CUALES LAS PARTES CUMPLEN OTRAS PROMESAS. LOS PROTOCOLOS GENERALMENTE SE IMPLEMENTAN CON PROGRAMAS EN UNA RED INFORMÁTICA O EN OTRAS FORMAS DE ELECTRÓNICA DIGITAL, POR LO QUE ESTOS CONTRATOS SON «MÁS INTELIGENTES» QUE SUS ANTECESORES EN PAPEL. EL USO DE INTELIGENCIA ARTIFICIAL NO ESTÁ IMPLICADO”.

NICK SZABO

El futuro digital
es de todos

MinTIC

❑ ¿CÓMO FUNCIONA?

1

O

UNA PARTE CODIFICA EL CÓDIGO

2

ES

AMBAS PARTES FINALZAN LAS

REGLAS DEL CONTRATO

INTELIGENTE

3

ÓN

EL CONTARTO SE ACTIVA Y SE ALMACENA

EN EL REGISTRO

4

ÓN

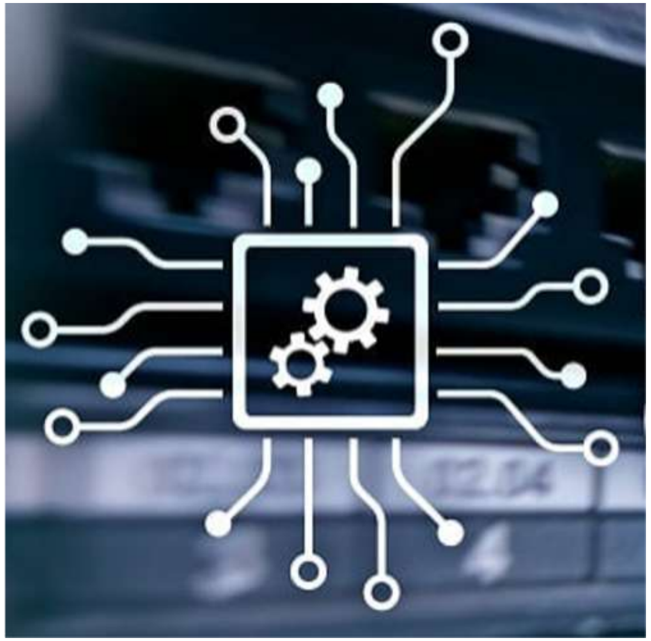
SE EJECUTA AUTOMATICAMENTE CUANDO

SE CUMPLEN LAS CONDICIONES

etraining
Diseñamos nuevas formas de aprender

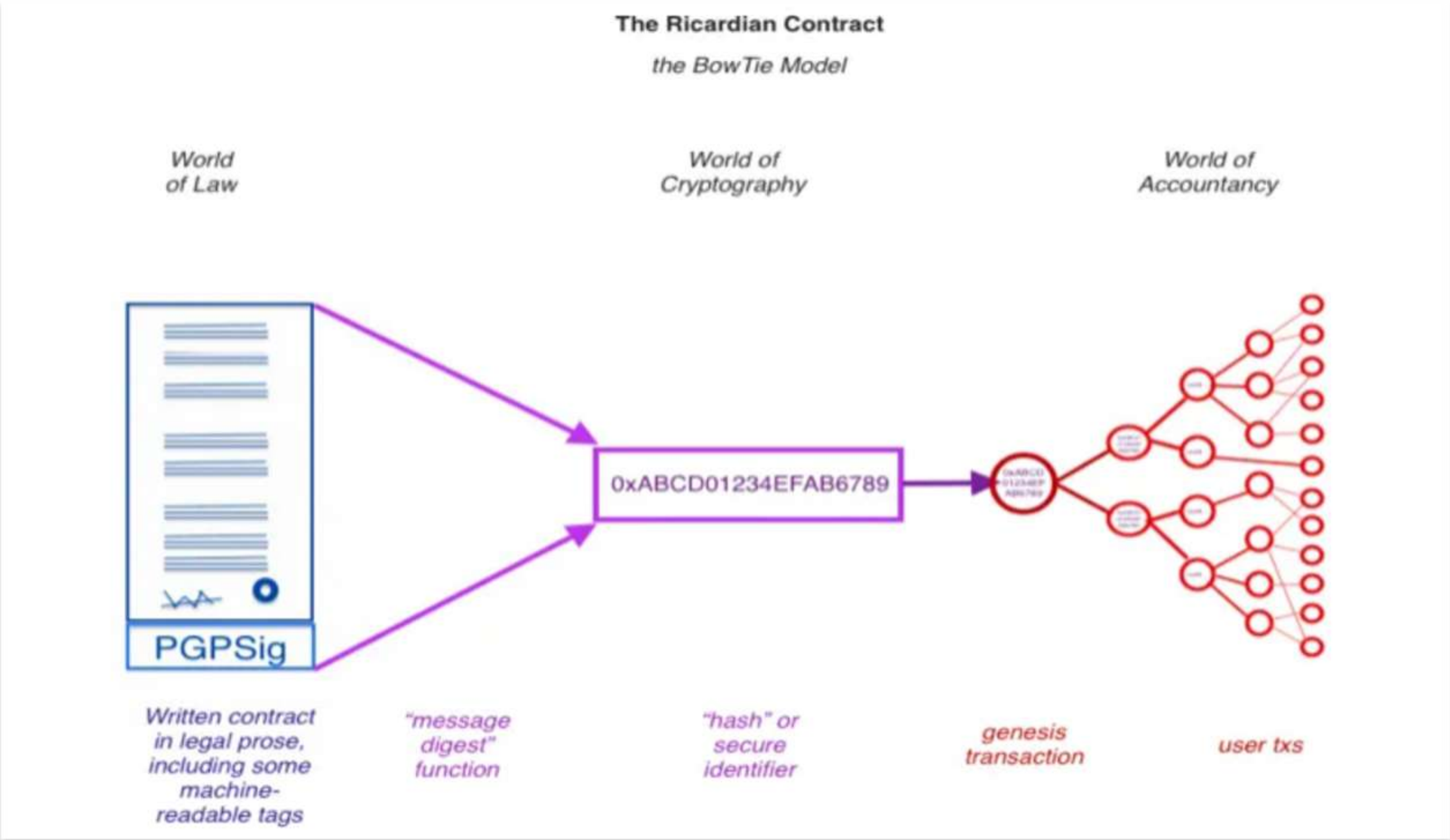
tecnalia Colombia
Inspiring Business

❑ BENEFICIOS



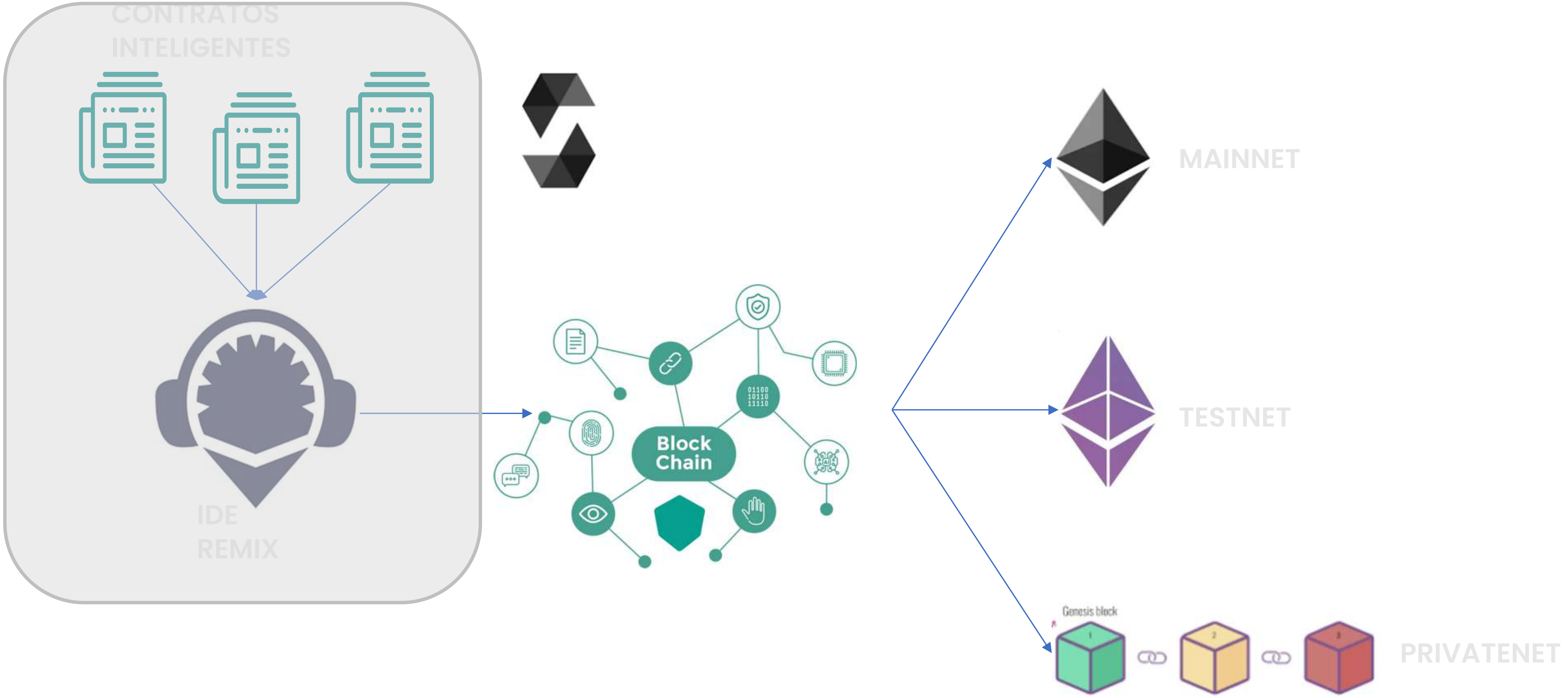
1. SEGURO
2. AUTÓNOMO
3. LIBRE DE INTERRUPCIONES
4. ECONOMICO
5. RENDIMIENTO RÁPIDO

❑ FUTURO


The Ricardian Contract
the BowTie Model
World of Law
World of Cryptography
World of Accountancy
PGPSig
0xABCD01234EFAB6789
Written contract in legal prose, including some machine-readable tags
"message digest" function
"hash" or secure identifier
genesis transaction
user txs

«Un contrato digital que define los términos y condiciones de una interacción, entre dos o más partes, que está firmado y verificado criptográficamente. Es importante destacar que es legible tanto para humanos como para máquinas y está firmado digitalmente».

❑ ¿CÓMO IMPLEMENTARLO?



CONTRATOS INTELIGENTES

IDE REMIX

MAINNET

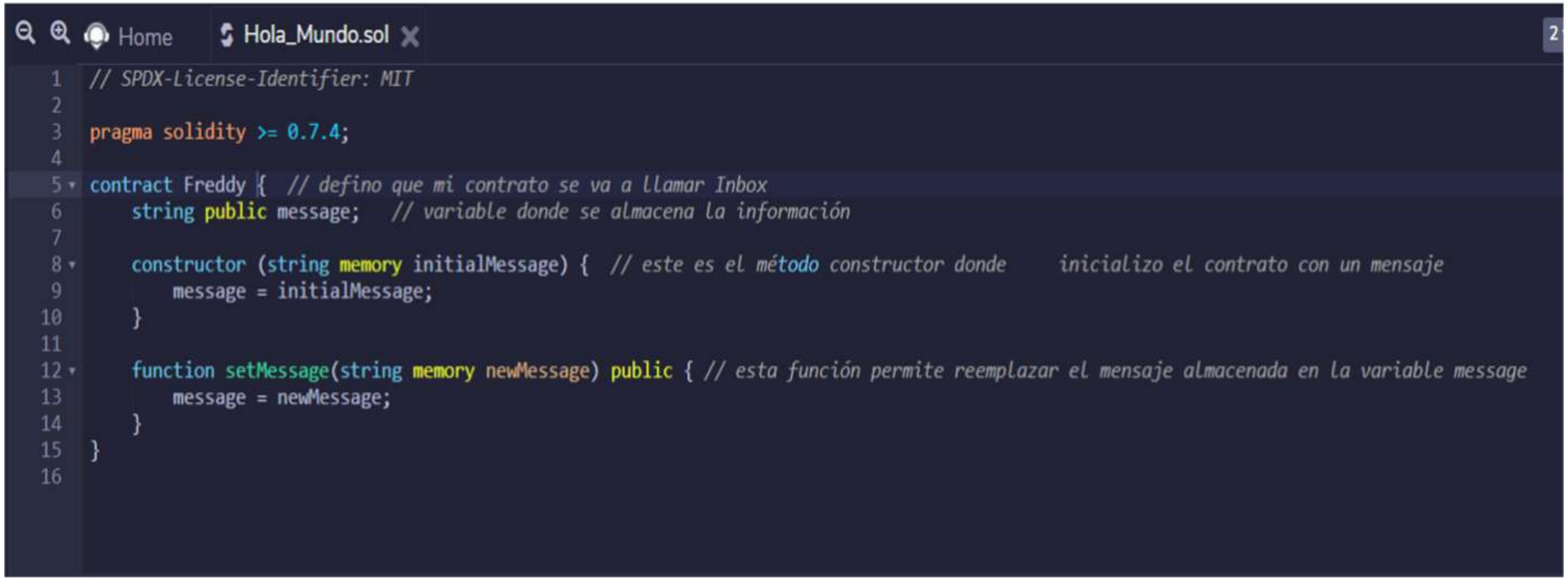
TESTNET

PRIVATENET

Block Chain
Genesis block

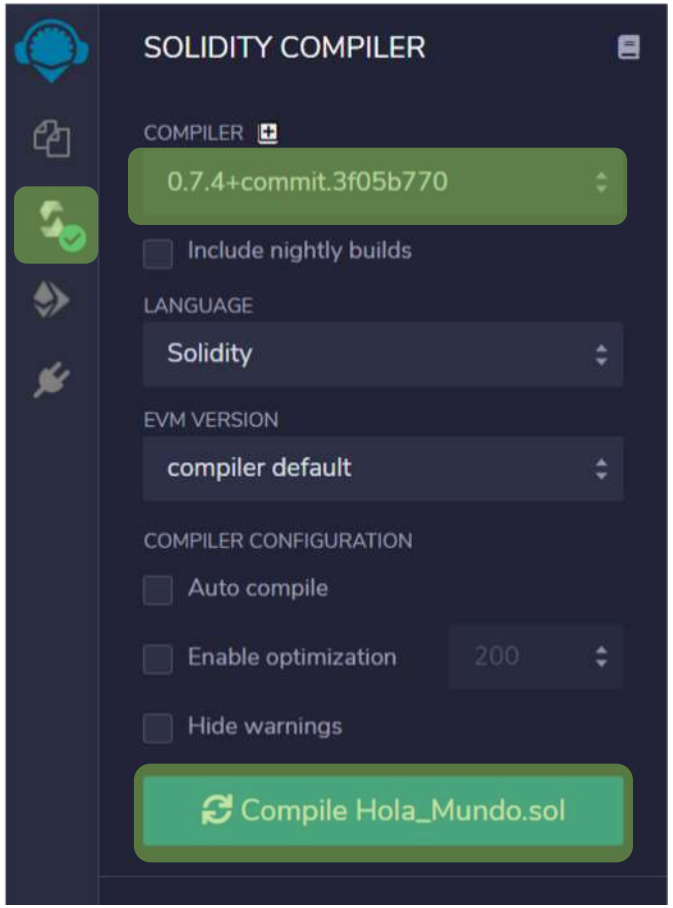
❑ TALLER

1: HOLA MUNDO



```
// SPDX-License-Identifier: MIT

pragma solidity >= 0.7.4;

contract Freddy {  // defino que mi contrato se va a llamar Inbox
    string public message;   // variable donde se almacena la información

    constructor (string memory initialMessage) {  // este es el método constructor donde     inicializo el contrato con un mensaje
        message = initialMessage;
    }

    function setMessage(string memory newMessage) public { // esta función permite reemplazar el mensaje almacenada en la variable message
        message = newMessage;
    }
}
```

MinTIC_Sesion_10_Codigo_HolaMundo_09.09.21.txt



SOLIDITY COMPILER

COMPILER: 0.7.4+commit.3f05b770

☐ Include nightly builds

LANGUAGE: Solidity

EVM VERSION: compiler default

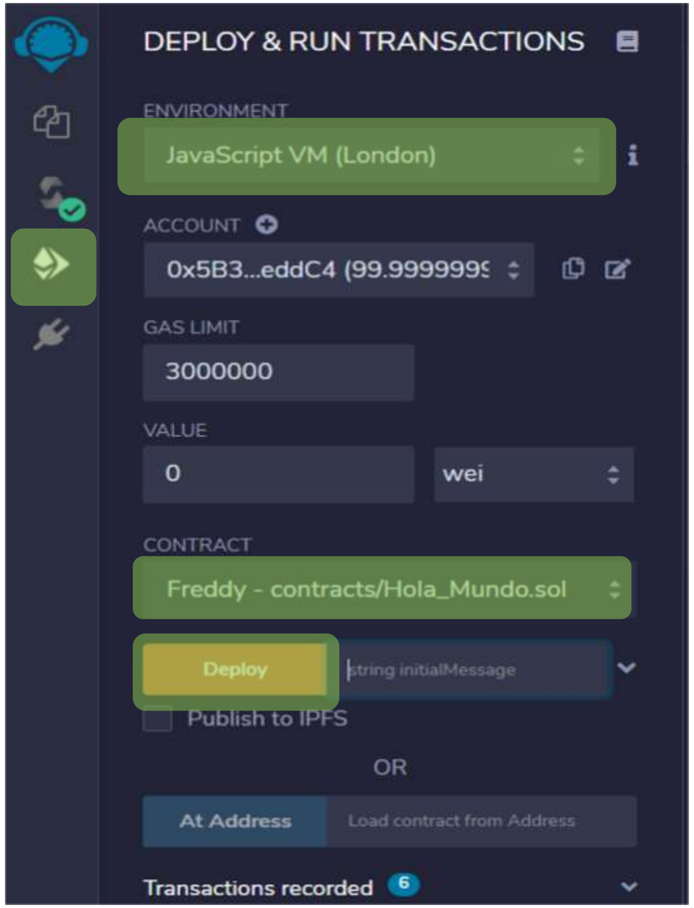
COMPILER CONFIGURATION

☐ Auto compile

☐ Enable optimization 200

☐ Hide warnings

Compile Hola_Mundo.sol



DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: JavaScript VM (London)

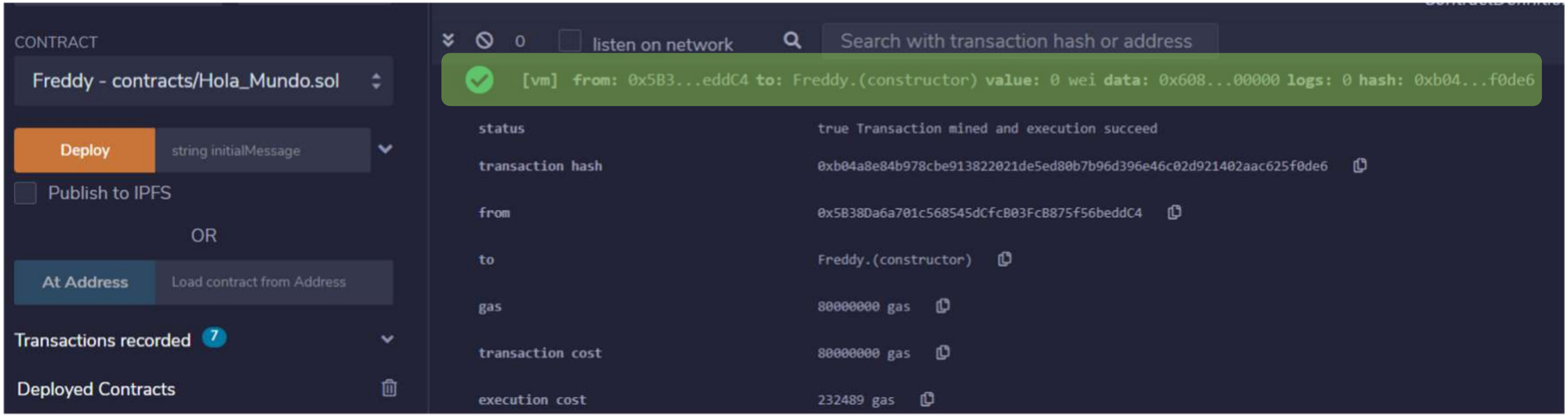
ACCOUNT: 0x5B3...eddC4 (99.999999

GAS LIMIT: 3000000

VALUE: 0 wei

CONTRACT: Freddy - contracts/Hola_Mundo.sol

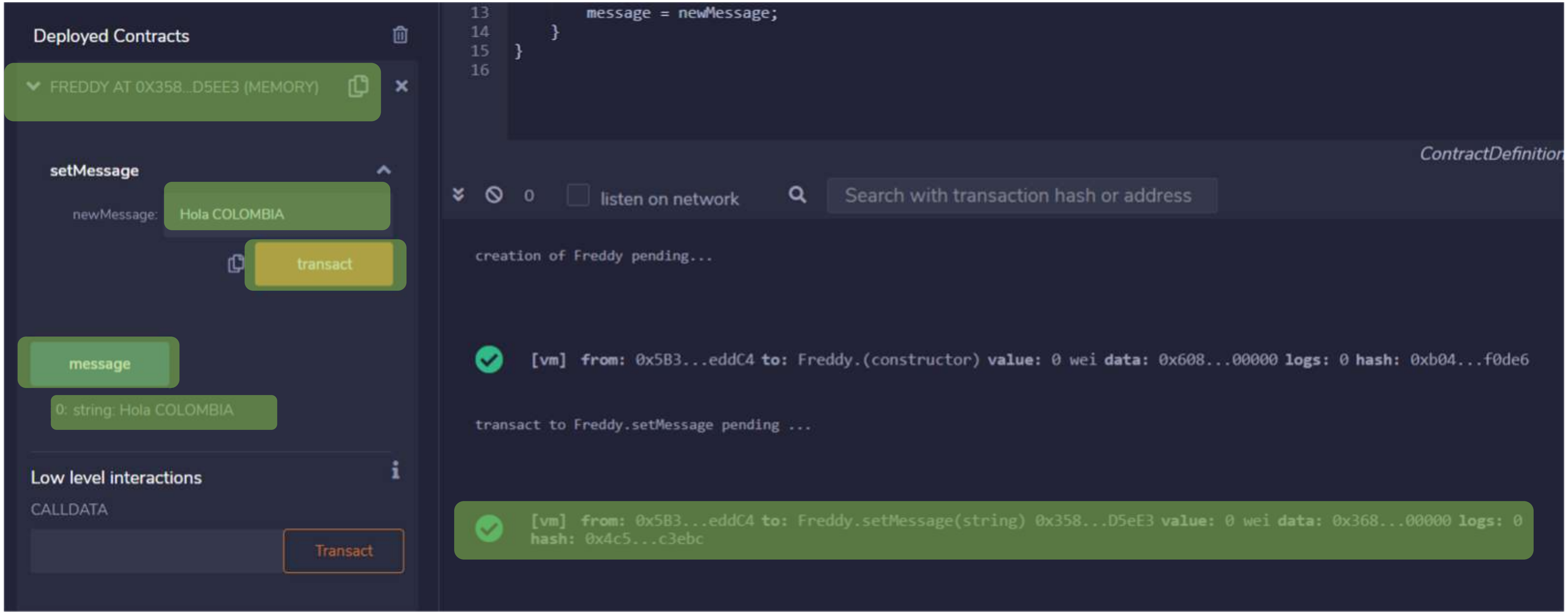
Deploy
string initialMessage
☐ Publish to IPFS
OR
At Address
Load contract from Address
Transactions recorded 6



CONTRACT
Freddy - contracts/Hola_Mundo.sol

Deploy
string initialMessage
☐ Publish to IPFS
OR
At Address
Load contract from Address
Transactions recorded 7
Deployed Contracts
0
☐ listen on network
Search with transaction hash or address

```
[vm] from: 0x5B3...eddC4 to: Freddy.(constructor) value: 0 wei data: 0x608...00000 logs: 0 hash: 0xb04...f0de6

status              true Transaction mined and execution succeed
transaction hash    0xb04a8e84b978cbe913822021de5ed80b7b96d396e46c02d921402aac625f0de6
from                0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to                  Freddy.(constructor)
gas                 80000000 gas
transaction cost    80000000 gas
execution cost      232489 gas
```


Deployed Contracts
FREDDY AT 0X358...D5EE3 (MEMORY)
setMessage
transact
message
Low level interactions
CALLDATA
Transact
listen on network
Search with transaction hash or address
ContractDefinition
0

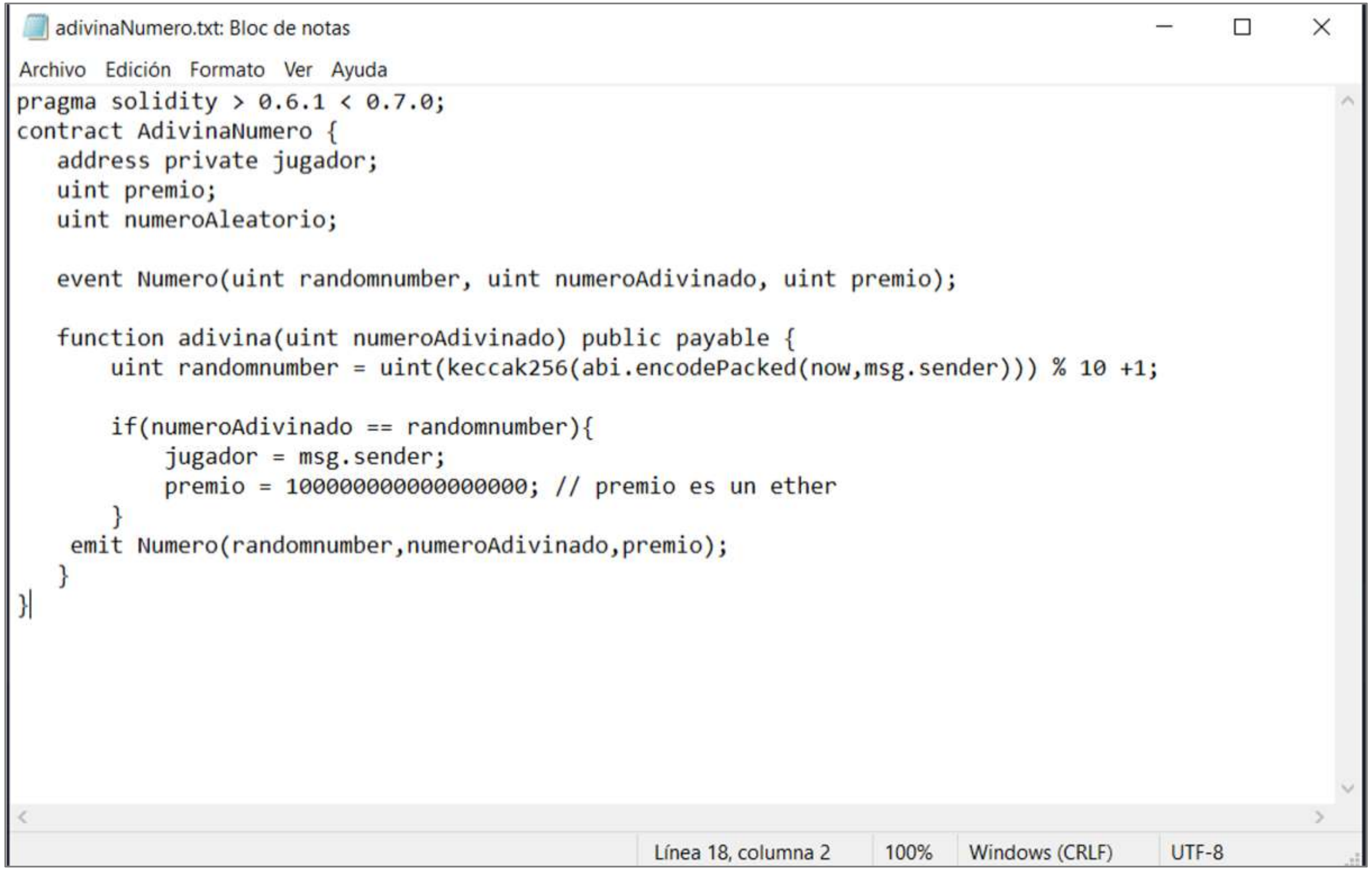
```
        message = newMessage;
    }
}
```

newMessage: Hola COLOMBIA

0: string: Hola COLOMBIA

```
creation of Freddy pending...

[vm] from: 0x5B3...eddC4 to: Freddy.(constructor) value: 0 wei data: 0x608...00000 logs: 0 hash: 0xb04...f0de6

transact to Freddy.setMessage pending ...

[vm] from: 0x5B3...eddC4 to: Freddy.setMessage(string) 0x358...D5eE3 value: 0 wei data: 0x368...00000 logs: 0
hash: 0x4c5...c3ebc
```

2: ADIVINA NÚMERO


adivinaNumero.txt: Bloc de notas
Archivo Edición Formato Ver Ayuda
Línea 18, columna 2
100%
Windows (CRLF)
UTF-8

```
pragma solidity > 0.6.1 < 0.7.0;
contract AdivinaNumero {
   address private jugador;
   uint premio;
   uint numeroAleatorio;

   event Numero(uint randomnumber, uint numeroAdivinado, uint premio);

   function adivina(uint numeroAdivinado) public payable {
       uint randomnumber = uint(keccak256(abi.encodePacked(now,msg.sender))) % 10 +1;

       if(numeroAdivinado == randomnumber){
           jugador = msg.sender;
           premio = 100000000000000000; // premio es un ether
       }
    emit Numero(randomnumber,numeroAdivinado,premio);
   }
}
```

El futuro digital
es de todos

MinTIC

Gracias

etraining
Diseñamos nuevas formas de aprender

tecnalia Colombia
Inspiring Business