olidity

El lenguaje de los smart contracts

ethereum

Blockchain

- Tecnología detrás de todas las criptomonedas
- Concebido en 2009 por Satoshi Nakamoto en el whitepaper que da origen al Bitcoin
- De Código Libre

Blockchain

- Resistencia a la censura

- Descentralización

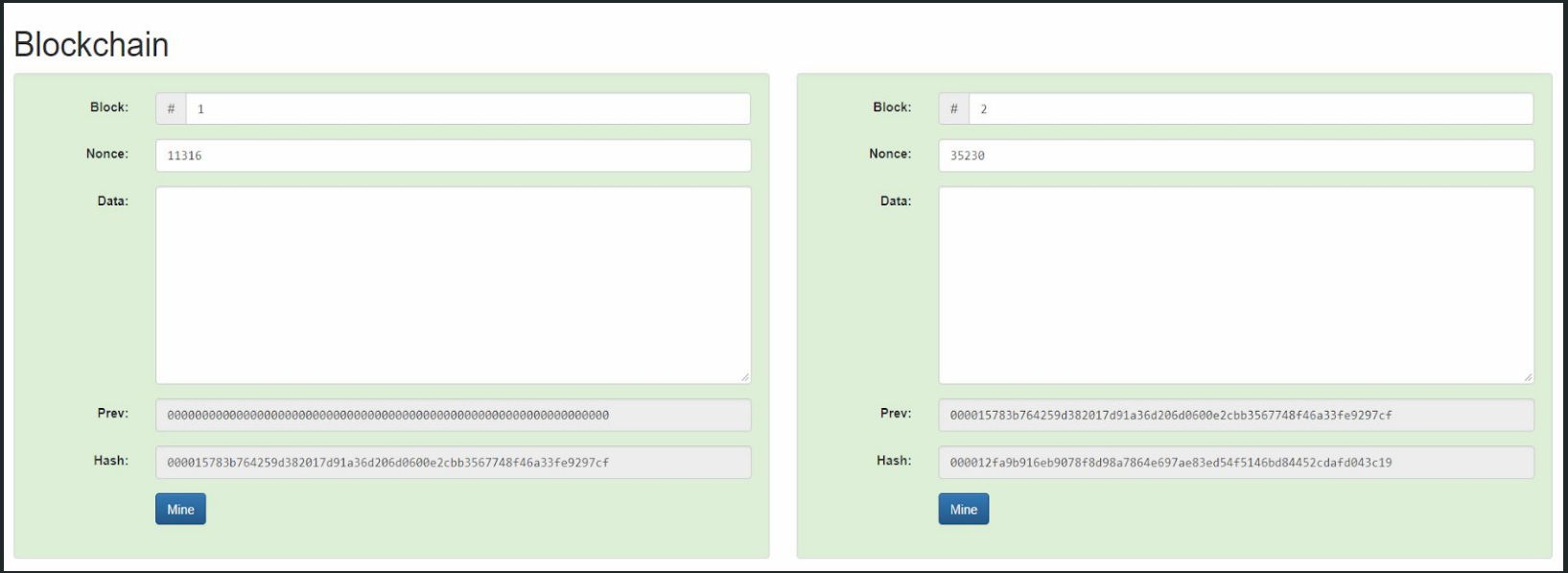
- Transparencia

- Globalidad y Neutralidad

Utilidad de la tecnología Blockchain

- Historias clínicas de hospitales
- Registros públicos del estado
- Transferencia de dinero
- Contratos legales

Blockchain - Ejemplo



https://anders.com/blockchain/blockchain.html

https://github.com/aleromer/topCoin

Ethereum

- Creado en 2015 por Vitalik Buterin

- Ethereum introduce el concepto de Smart Contracts. Código turing complete Que es ejecutado dentro de la EVM (Ethereum Virtual Machine)

- Solidity es el lenguaje utilizado para crear Smart Contracts

Solidity - **Características**

- Lenguaje similar en sintaxis a javascript
- Gasto computacional. GAS
- Turing Complete
- Código corre en cada uno de los nodos. Dentro de la EVM

Solidity - **Características**

- Fuertemente tipado
- Cada contrato se asemeja a una clase
- Permite herencia múltiple entre contratos - Polimorfismo
- Todas las funciones son virtuales

Solidity - **Características**

- Problema del diamante - Linearización C3
- Matemática en Solidity se hace usando punto fijo.
- Para el éter, no es necesario usar valores fraccionarios
- Muchas subunidades, entre ellas wei, finney, szabo y ether.

Características del lenguaje - **unidades**

```
var unitMap = {
'wei':          '1',
'shannon':      '1000000000',
'szabo':        '1000000000000',
'finney':       '1000000000000000',
'ether':        '1000000000000000000',
'gether':       '1000000000000000000000000000',
'tether':       '1000000000000000000000000000000'
};
```

Tipos de datos - **Generales**

Booleanos (`bool`)

Enteros hasta 256 bits (`int / uint`)

Números de punto fijo (`fixedMxN / ufixedMxN`)

Vectores (estáticos, o dinámicos) (`length / push`)

Constantes (Racionales y enteras, de direcciones, de strings, hexadecimales)

Enums / Structs

Diccionarios (`mapping(_KeyType => _ValueType)`)

Tipos de datos - **Generales**

Operadores de asignación

```
-=, *=, /=, %=, |=, &= y ^=
```

Conversiones explícitas

```
int8 y = -3;
uint x = uint(y);
```

Deducción de tipos

```
uint24 x = 0x123;
var y = x;
```

Tipos de datos - **Específicos**

Direcciones (`address`): Valor de 20 bytes -
Tienen miembros (`balance / transfer`) y pueden ser usadas de base para contratos Ethereum.

Funciones (`function (<parameter types>) {internal|external} [pure|constant|view|payable] [returns (<return types>)]`))

Modificadores de funciones (semántica declarativa en las funciones)

Tipos de datos - **Específicos**

Reference Types

Todos los tipos complejos poseen el atributo "`data location`" que puede tomar los valores: (`storage (donde se almacena state) / memory (no persistente) / calldata`)

Ejemplo - **Hola mundo**

```
pragma solidity ^0.4.0;

contract holaMundo
{
    function renderHolaMundo () public pure returns
(string)
    {
        return 'Hola Mundo';
    }
}
```

Ejemplo - **Almacenar un valor**

```
pragma solidity ^0.4.0;

contract SampleContract {
    uint storageData;
    function set(uint x) { storageData = x; }

    function get() constant returns (uint) {
        return storageData;
    }
}
```

Ejemplo - Compra segura - **Variables**

```
pragma solidity ^0.4.0;

contract Purchase {
    uint public value; // Valor de la compra
    address public seller; // Dirección del vendedor
    address public buyer; // Dirección del comprador
// Estados: Creado - Bloqueado - Finalizado)
    enum State { Created, Locked, Inactive }
    State public state;
```

Compra segura - **Firma del contrato**

```
function Purchase() public payable {
    seller = msg.sender;
    value = msg.value / 2;
    require((2 * value) == msg.value);
}
```

Compra segura - **Modificadores**

```
modifier condition(bool _condition) {
    require(_condition);
    _;
}
modifier onlyBuyer() {
    require(msg.sender == buyer);
    _;
}
```

Compra segura - **Modificadores**

```
modifier onlySeller() {
    require(msg.sender == seller);
    _;
}
modifier inState(State _state) {
    require(state == _state);
    _;
}
```

Compra segura - **Eventos**

```
event Aborted();
event PurchaseConfirmed();
event ItemReceived();
```

Compra segura - **Vendedor aborta Operación**

```
function abort()
    public
    onlySeller
    inState(State.Created)
{
    emit Aborted();
    state = State.Inactive;
    seller.transfer(this.balance);
}
```

Modificadores

Cambio de estado

Compra segura - **Comprador confirma Operación**

```
function confirmPurchase()
    public
    inState(State.Created)
    condition(msg.value == (2 * value))
    payable
{
    emit PurchaseConfirmed();
    buyer = msg.sender;
    state = State.Locked;
}
```

Modificadores

Cambio de estado

Compra segura - **Comprador confirma Recepción**

```
function confirmReceived()
    public
    onlyBuyer
    inState(State.Locked)
{
    emit ItemReceived();
    state = State.Inactive;
    buyer.transfer(value);
    seller.transfer(this.balance);
}
```

Modificadores

Cambio de estado

Ejemplo - **Herencia no compila**

```
// Esto no compila

pragma solidity ^0.4.0;

contract X {}
contract A is X {}
contract C is A, X {}
```

Ejemplo - **Herencia**

```
pragma solidity ^0.4.0;

contract owned {
    function owned() public { owner = msg.sender; }
    address owner;
}
contract mortal is owned {
    function kill() public {
        if (msg.sender == owner) selfdestruct(owner);
    }
}
```

Ejemplo - **Herencia**

```
contract Base1 is mortal {
    function kill() public { /* do cleanup 1 */
super.kill(); }
}
contract Base2 is mortal {
    function kill() public { /* do cleanup 2 */
super.kill(); }
}
contract Final is Base1, Base2 {}
```

Ejemplo - **Crear moneda**

```
pragma solidity ^0.4.21;
contract Coin {
    address public minter;
    mapping (address => uint) public balances;
    event Sent(address from, address to, uint amount);
```

Ejemplo - **Crear moneda**

```
function Coin() public {
    minter = msg.sender;
}
function mint(address receiver, uint amount) public {
        if (msg.sender != minter) return;
        balances[receiver] += amount;
}
```

Ejemplo - **Crear moneda**

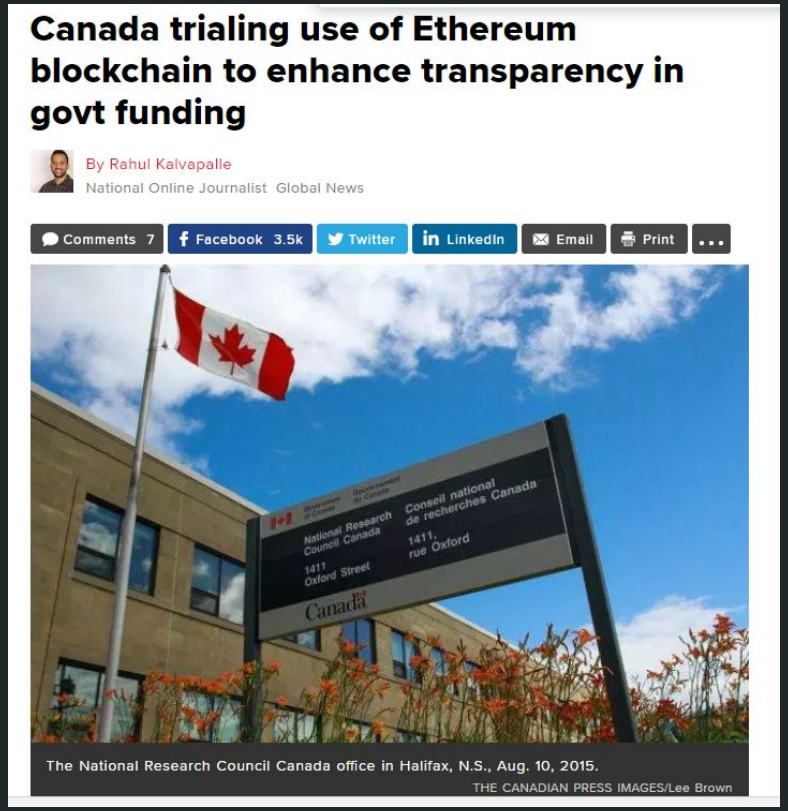
```
    function send(address receiver, uint amount)
public {
        if (balances[msg.sender] < amount) return;
        balances[msg.sender] -= amount;
        balances[receiver] += amount;
        emit Sent(msg.sender, receiver, amount);
    }
}
```

Cómo publicar en la red Ethereum - **Protocolo**

El protocolo de Ethereum posee 3 implementaciones originales, escritas en los lenguajes Go, C++ y Python.
Hay diversos ejemplos de cómo hacerlo en la web:

https://ethereum.gitbooks.io/frontier-guide/content/example_script.html

Casos de uso - **Registros Públicos**



Canada trialing use of Ethereum blockchain to enhance transparency in govt funding

By Rahul Kalvapalle
National Online Journalist, Global News

Comments 7 | Facebook 3.5k | Twitter | LinkedIn | Email | Print | ...

The National Research Council Canada office in Halifax, N.S., Aug. 10, 2015.
THE CANADIAN PRESS IMAGES/Lee Brown

Canadá está probando publicar en la blockchain de ethereum los fondos que destina a empresas externas.
Al estar la información disponible en ethereum. Resulta fácil crear smart contracts que puedan interactuar con esa información.

Casos de uso - **ICOs**

Initial Coin Offering

Creación de un token que se entrega a cambio de ethers.

https://www.ethereum.org/crowdsale

Casos de uso - **ICOs**

```
function Crowdsale(
    address ifSuccessfulSendTo,
    uint fundingGoalInEthers,
    uint durationInMinutes,
    uint etherCostOfEachToken,
    address addressOfTokenUsedAsReward
) public {
    beneficiary = ifSuccessfulSendTo;
    fundingGoal = fundingGoalInEthers * 1 ether;
    deadline = now + durationInMinutes * 1 minutes;
    price = etherCostOfEachToken * 1 ether;
    tokenReward = token(addressOfTokenUsedAsReward);
}
```

Casos de uso - **Juegos**

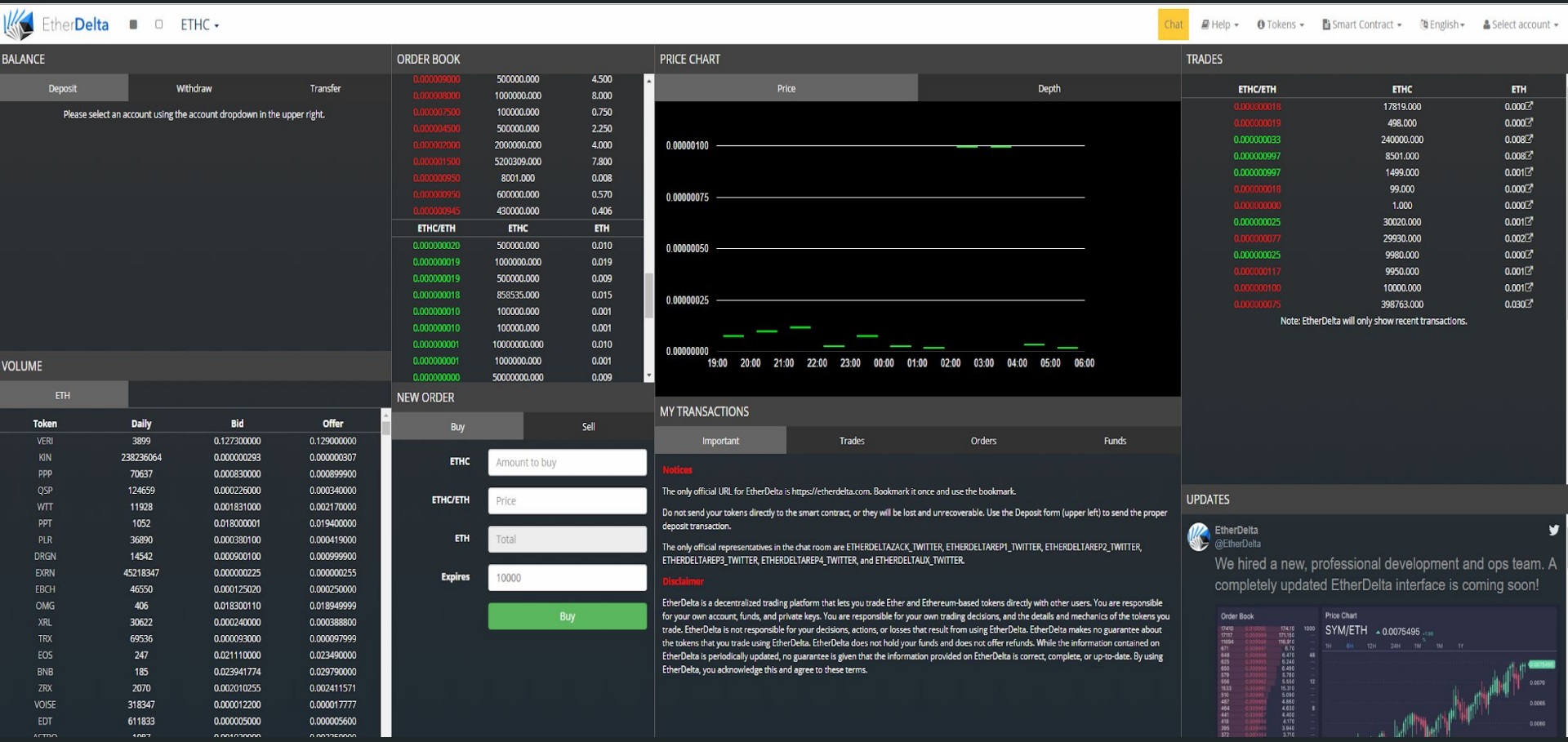
CryptoKitties

Juego que permite comprar, vender y criar gatitos virtuales adquiridos con ether.

Son únicos. Su ADN se basa en en la dirección de un token creado con este fin.

Los de primera generación llegaron a valer entre 50.000 y 113.000 dólares cada uno

Casos de uso - **EtherDelta**



Casos de uso - **EtherDelta**

```
 function transfer(address _to, uint256 _value) returns
(bool success) {
   if (balances[msg.sender] >= _value && balances[_to] +
_value > balances[_to]) {
   //if (balances[msg.sender] >= _value && _value > 0) {
     balances[msg.sender] -= _value;
     balances[_to] += _value;
     Transfer(msg.sender, _to, _value);
     return true;
```

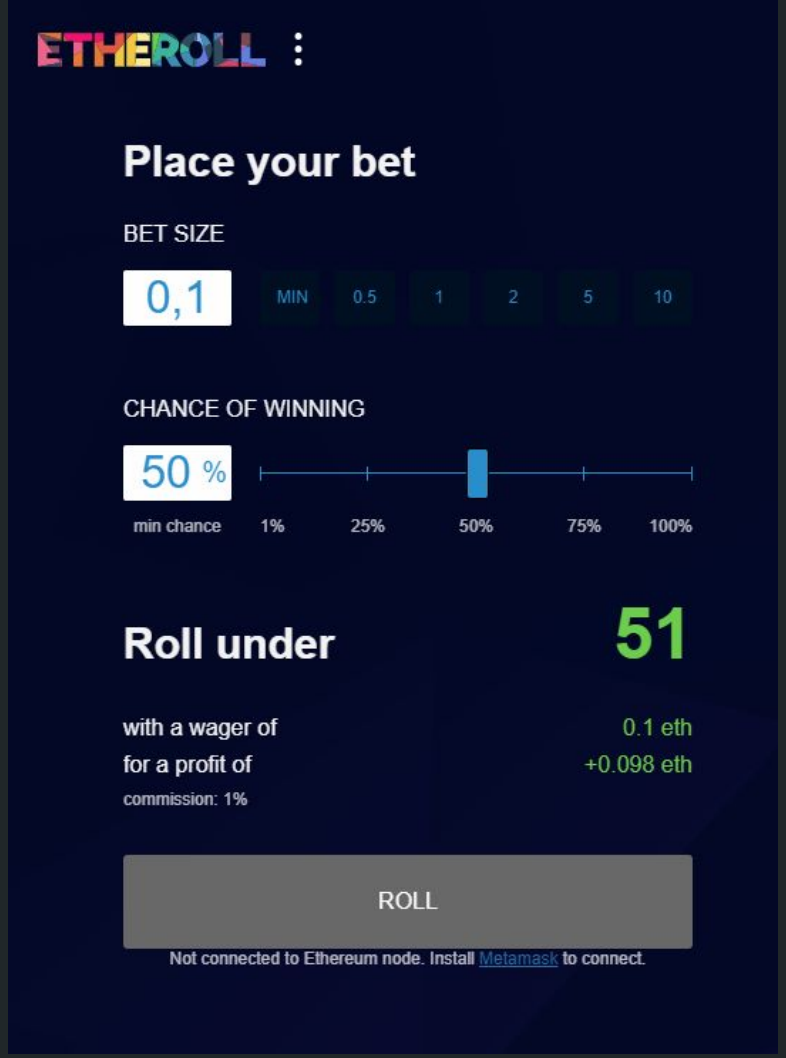
Casos de uso - **EtherDelta**

```
function deposit() payable {
  tokens[0][msg.sender] = safeAdd(tokens[0][msg.sender],
msg.value);
  Deposit(0, msg.sender, msg.value, tokens[0][msg.sender]);
}
function withdraw(uint amount) {
  if (tokens[0][msg.sender] < amount) throw;
  tokens[0][msg.sender] = safeSub(tokens[0][msg.sender], amount);
  if (!msg.sender.call.value(amount)()) throw;
  Withdraw(0, msg.sender, amount, tokens[0][msg.sender]);
}
```

Casos de uso - **EtherRoll**

https://github.com/snowsky/EthRoll/blob/master/contracts/eth.sol

Casino virtual sin intermediarios.

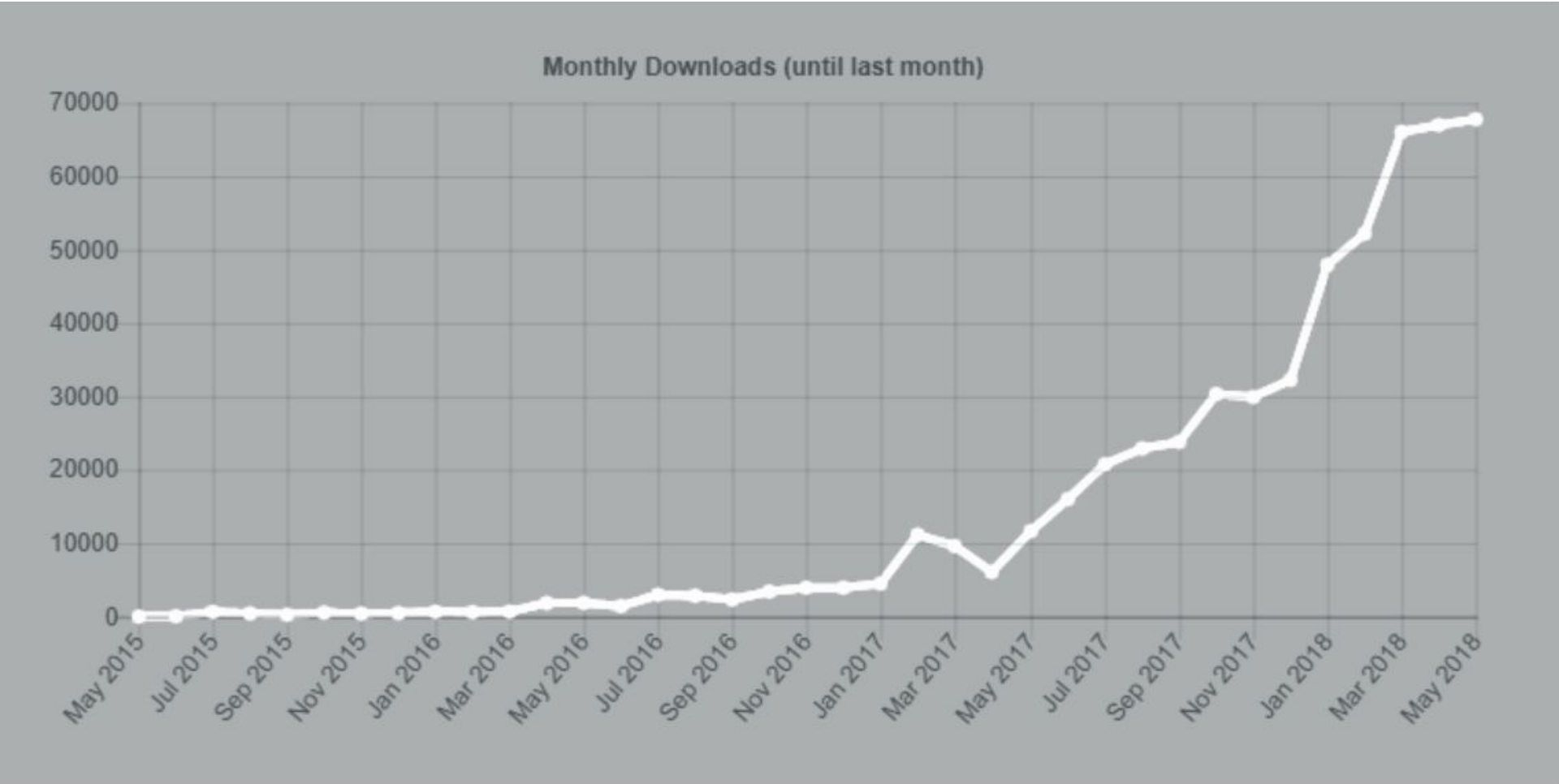


Estadísticas **- CNBC**

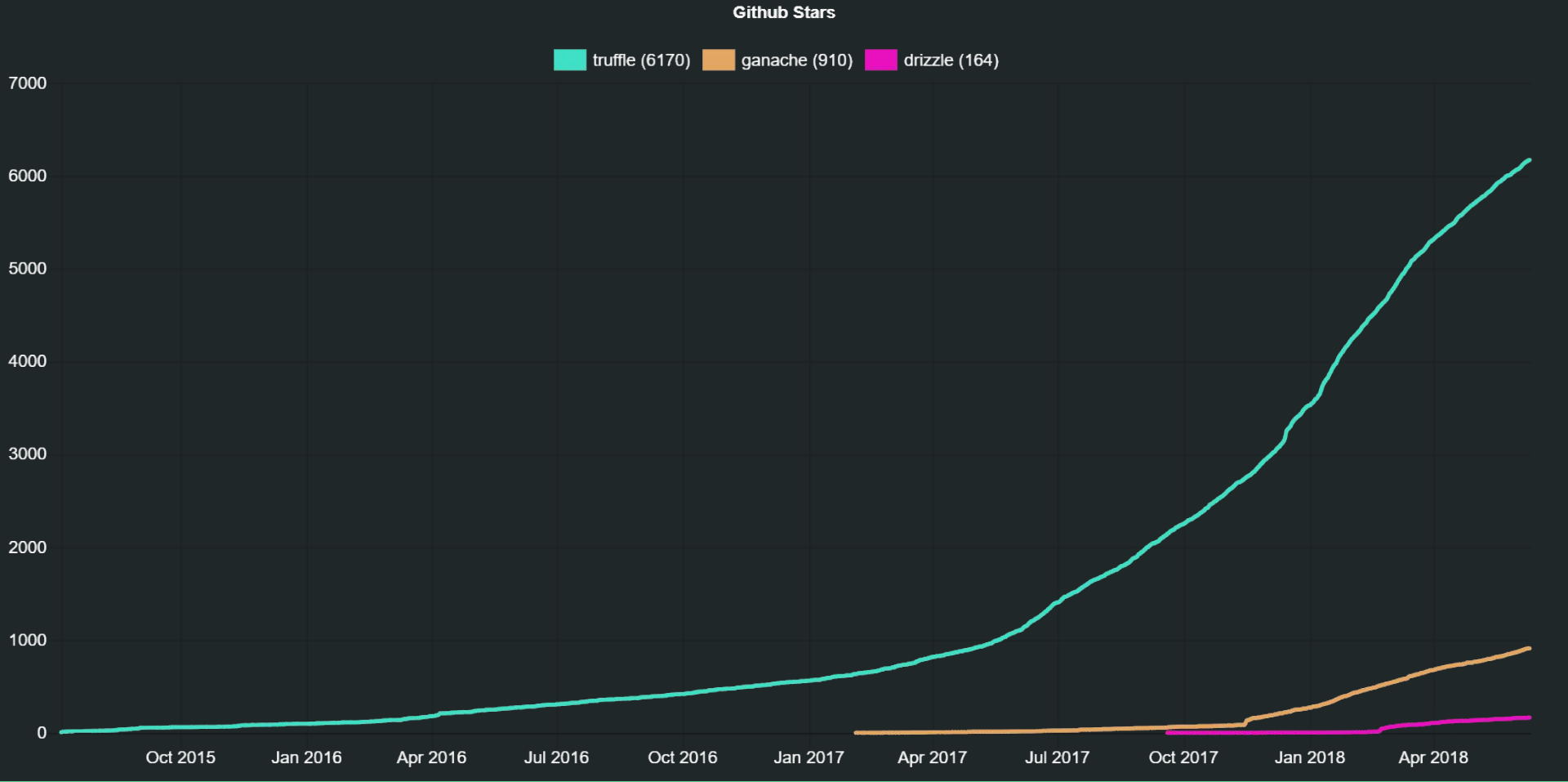
Según datos de Octubre de 2017 de la cadena de noticias estadounidense CNBC, Cerca de 35,000 desarrolladores y más de 500 start-ups desarrollan en esta plataforma.

Hay actualmente 70.000 Descargas mensuales del framework Truffle para desarrollo Ethereum.

Estadísticas - **Truffle (framework) - Descargas**



Estadísticas - **Github Stars - Truffle,Ganache,Drizzle**



Transacciones/segundo **vs Visa y PayPal**

Article & Sources:
https://howmuch.net/articles/crypto-transaction-speeds-compared
https://howmuch.net/sources/crypto-transaction-speeds-compared

howmuch.net

Precio de Ethereum vs BTC/USD - 3 años de vida

Videos

Creación de ICO's:

https://www.youtube.com/watch?time_continue=79&v=ac1P3GXkFxc

Desarrollo de Ethereum:

https://www.youtube.com/watch?v=V0XfIeKJSXM

Referencias

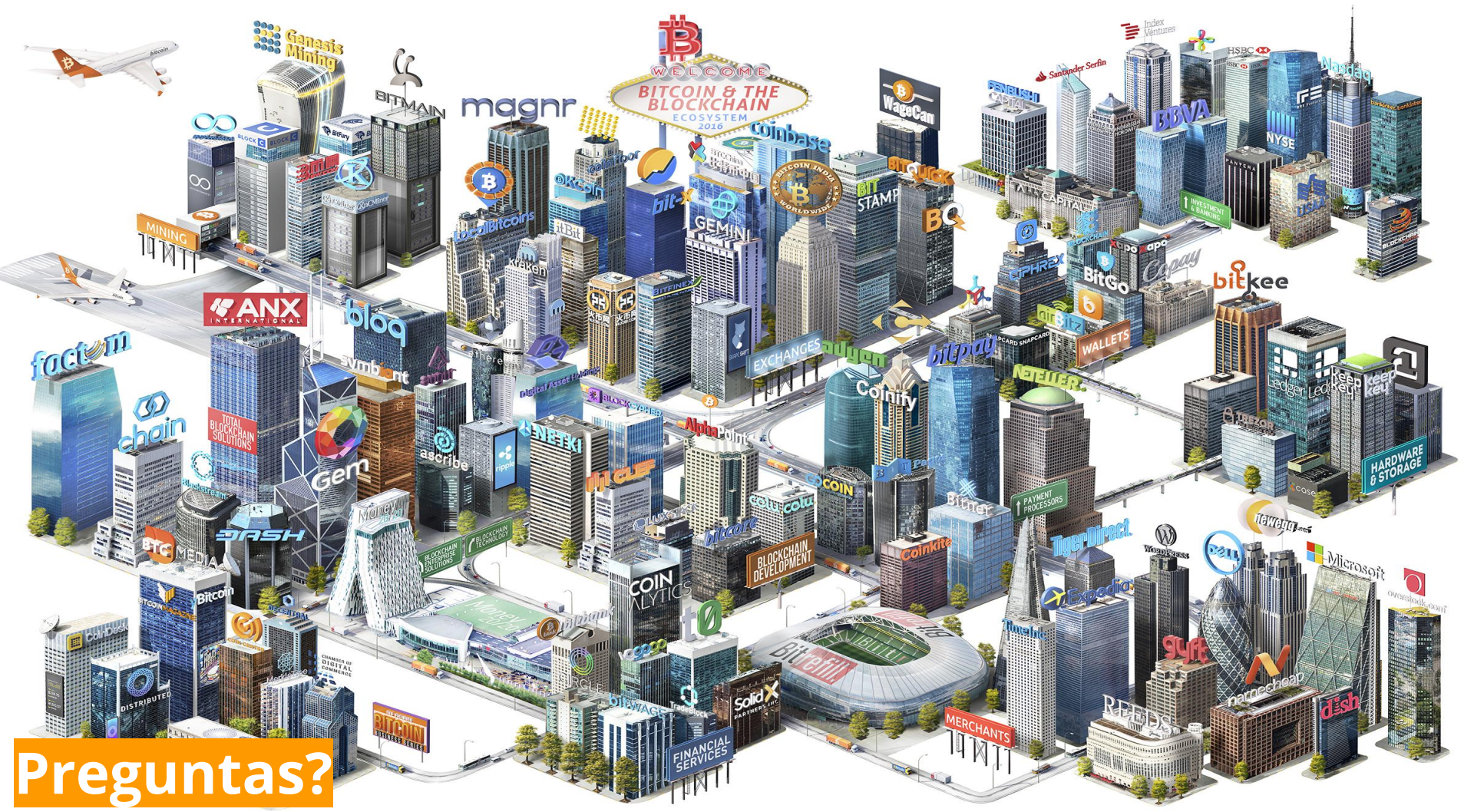
http://truffleframework.com/

https://coinmarketcap.com/

http://www.ethdocs.org/en/latest/

https://ethereum.gitbooks.io/frontier-guide/content/example_script.html

https://github.com/etherdelta/smart_contract/blob/master/etherdelta.sol


Genesis Mining
BITMAIN
magnr
WELCOME
BITCOIN & THE BLOCKCHAIN
ECOSYSTEM
2016
coinbase
WageCan
Index Ventures
Santander Serfin
HSBC
BBVA
NYSE
MINING
ANX INTERNATIONAL
bloq
factom
chain
TOTAL BLOCKCHAIN SOLUTIONS
Gem
DASH
BTC MEDIA
NETKI
ascribe
ripple
EXCHANGES
adyen
bitpay
Coinify
Alpha Point
COCOIN
Bitnet
coinkite
BLOCKCHAIN DEVELOPMENT
PAYMENT PROCESSORS
WALLETS
bitkee
Ledger
keepkey
HARDWARE & STORAGE
TigerDirect
Microsoft
Expedia
Bitrefill
MERCHANTS
FINANCIAL SERVICES
COIN ALYTICS
SolidX
DISTRIBUTED
BITCOIN INVESTORS SERIES
namecheap
dish
Preguntas?