

Auditoría

Administración de Bases de Datos

Departamento de Lenguajes y Sistemas Informáticos

eman ta zabal zazu



Universidad
del País Vasco

Euskal Herriko
Unibertsitatea

Contenido

1. Auditoría en SGBDs relacionales

1. Introducción
2. Herramientas

2. Auditoría y Blockchain

1. Introducción a Blockchain
2. Redes Blockchain públicas vs. privadas



Auditoría

Definición:

- Es el proceso que consiste en recoger, agrupar y evaluar evidencias para determinar si un sistema de información:
 - Salvaguarda el activo empresarial
 - Mantiene la integridad de los datos
 - Cumple con las leyes y regulaciones establecidas
 - Utiliza eficientemente los recursos



Auditoría

Definición alternativa:

- Proceso que permite medir, monitorizar y registrar los accesos a la información de una BBDD:
 - ¿Quién y cuando accede?
 - ¿Desde donde?
 - ¿Qué comandos / sentencias SQL se ejecutaron?
 - ¿Cuál fue el efecto?



Auditoría: Tipos

- Rutinaria

- Objetivos:

- Garantizar la integridad de la información en la BBDD.
 - Verificar que hay controles para prevenir riesgos (y el posible impacto).

- Actividades tipo:

- Revisión de los usuarios y permisos.
 - Revisión de la configuración del SGDB.



- Forense

- Objetivo: Encontrar el motivo de un problema / fallo.
 - Actividad principal: revisión de los logs/registros.



Auditoría: Requisitos

- No puede comprometer el funcionamiento rutinario.
 - Generalmente implican operaciones adicionales en el SGBD fuera de la lógica de negocio.
- Debe proveer información relevante, relativa a:
 - Seguridad.
 - Apoyo en la toma de decisiones.
 - Mejora en el funcionamiento de la organización.
- En lo posible, que no sea auto-aplicada.
 - Una persona externa encontrará los fallos mejor que nosotros.



Auditoría: Acciones a auditar

- Usuarios:

- Inicios/cierres de sesión (incluyendo fallidos)
- Propagación de permisos (incluyendo fallidos)



- Procesos:

- Cambios en la configuración / tablas del SGBD.
- Reinicio del servidor/proceso.



- Comandos:

- Acciones con permisos de administrador.
- Intentos fallidos de acceso por falta de autorización.
- Ejecución de procedimientos almacenados.



Herramientas de auditoría

- 1) Logs de la BBDD/Sistema
- 2) Aplicaciones de 3ºs
- 3) Triggers



Herramientas: Logs

- Ficheros que contienen el registro de actividad del SGBD/ sistema.
- Ventajas:
 - Flexibles: el nivel de detalle suele ser muy configurable.
- Inconvenientes
 - Requieren vigilancia para evitar problemas de almacenamiento.
 - Su análisis puede ser una tarea compleja.



Herramientas: Aplicaciones de 3ºs

- Software específico que se conecta al SGBD y permite monitorizar características
- Ventajas:
 - Facilitan la extracción de información.
 - Combinan información de diferentes sistemas.
- Inconvenientes:
 - Posible coste económico (de adquisición y formación).
 - Posible impacto en el rendimiento del SGBD/sistema.



Herramientas: Triggers

- Programas almacenados que se llaman automáticamente tras un evento concreto.
 - P.e. tras un Insert/Update/Delete en una tabla.
- El estándar SQL soporta 2 tipos de *Triggers*:
 - Nivel de fila: se activa por cada fila que se inserta/modifica/elimina en una tabla.
 - Nivel de sentencia: se activa por cada transacción, independientemente del número de filas.
- MySQL sólo soporta a nivel de fila.



Herramientas: Triggers

- Ventajas:
 - Una forma de vigilar la integridad de los datos.
 - Son una alternativa a las tareas programadas.
- Inconvenientes¹:
 - No permiten hacer todo tipo de validaciones.
 - Pueden ser difíciles de depurar.
 - Se ejecutan directamente en la BBDD.
 - Se debe evitar implementar lógica de la aplicación en ellos.
 - Dificulta la depuración de la aplicación.
 - Implican mayor sobrecarga de MySQL.



¹mysqltutorial.org: <https://www.mysqltutorial.org/mysql-triggers>

Herramientas: Triggers

- Cómo crear Triggers en MySQL¹:
 - Cada Trigger se asocia a una tabla concreta.

```
CREATE TRIGGER <nombre-trigger> <cuándo> <acción-tabla> on <tabla>  
FOR EACH ROW <acciones-trigger>
```

donde:

- nombre-trigger: Nombre arbitrario para el Trigger.
- cuándo: Momento a activar respecto a <acción-tabla>: Before/After.
- acción-tabla: Acción en la tabla que dispara el Trigger: Insert/Delete/Update.
- tabla: Tabla objeto del Trigger.
- Acciones-trigger: Acciones a ejecutar tras disparar el Trigger



¹MySQL 8 Docs, Trigger Syntax and examples: <https://dev.mysql.com/doc/refman/8.0/en/trigger-syntax.html>

Herramientas: Triggers

- Ejemplo¹:

- Crear tabla:

```
CREATE TABLE cuenta (id int, cantidad decimal(10,2));
```

- Crear Trigger:

```
CREATE TRIGGER mi_trigger BEFORE INSERT ON cuenta  
FOR EACH ROW SET @suma = @suma + NEW.cantidad;
```

NEW se refiere a la
nueva fila

- Inicializar variable asociada e insertar valores de prueba:

```
SET @suma = 0;  
INSERT INTO cuenta VALUES(1, 11.1), (2, 22.2), (3, 33.3);
```

- Verificar que el Trigger ha actualizado la variable asociada:

```
SELECT @suma AS 'Valor total en cuenta';
```



Herramientas: Triggers

- El cuerpo de un Trigger puede contener varias sentencias:
 - Ejemplo¹:

```
CREATE TRIGGER mi_trigger BEFORE UPDATE ON mi_tabla
FOR EACH ROW
BEGIN
    IF NEW.columna > THEN
        ...
    END
```

- Mostrar Triggers asociados a un BD:

```
SHOW TRIGGERS;
```

- Borrar Trigger:

```
DROP TRIGGER <nombre-trigger>
```



Ejercicio 1

- Crear una BBDD llamada "Auditada" y usarla para:
 - Crear 2 tablas:
 - Tabla "Datos" con 3 columnas: id (int), producto (text), precio (int)
 - Tabla "Audit" con 2 columnas: gasto (int), fecha (date)
 - Crear un Trigger:
 - Después de introducir una fila en "Datos", insertar una nueva fila en "Audit":
 - La columna "gasto" contendrá el valor de "precio" en la nueva fila.
 - La columna "fecha" contendrá la fecha actual.
- Introducir 2 filas nuevas en "Datos" y verificar que "Audit" se ha actualizado.
 - Utilizad datos aleatorios.



Blockchain y auditoría

- Las BBDD relacionales requieren de una configuración explícita para realizar labores de auditoría detallada.
- *Sin embargo:*
- Blockchain es un tipo de BBDD que mantiene un control exhaustivo sobre quién y cuándo realiza modificaciones sobre los datos.



Bitcoin



- Criptomoneda descentralizada
 - No depende de un banco o estado
- Presentada en 2008 en un artículo científico
 - <https://bitcoin.org/bitcoin.pdf>



- Autor anónimo bajo el nombre Satoshi Nakamoto
 - Aunque hay rumores sobre quién podría ser:
 - <https://www.businessinsider.es/elon-musk-revela-quien-podria-ser-verdadero-creador-bitcoin-987709>



Bitcoin

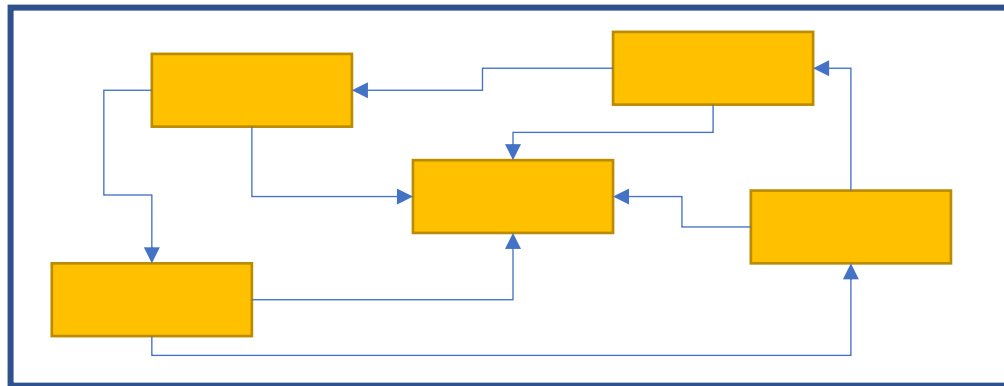


- Su uso comenzó en 2009
 - Implementación *open source*
 - Repositorio: <https://github.com/bitcoin/bitcoin>
- Actualmente 1 BTC \approx 21.443 €
 - A fecha 3 de febrero de 2023.
 - Muy volátil: <https://www.xe.com/currencycharts/?from=BTC&to=EUR>
- Blockchain se presentó como parte de Bitcoin
 - Es la BBDD que permite su funcionamiento.
 - Se utiliza como libro de cuentas (*ledger*) .



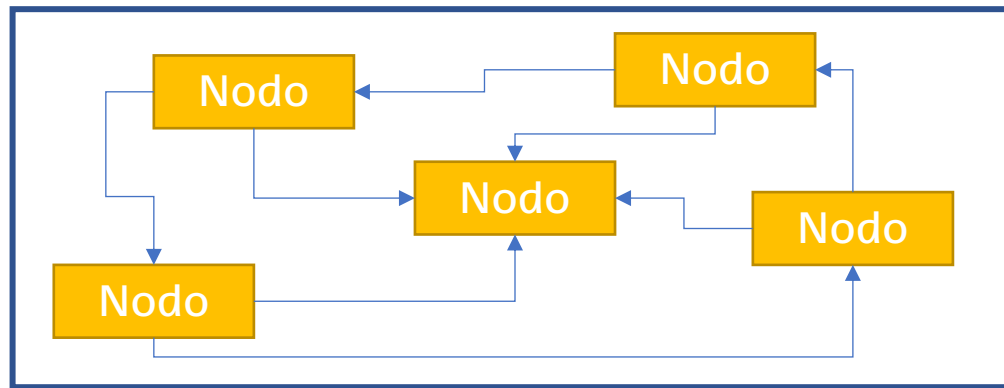
Blockchain

- Es una BBDD distribuida utilizada para almacenar datos.
 - Y, en el caso de Bitcoin, gestionar transferencias de dinero.
- Blockchain no es exclusivo de Bitcoin.
 - Hay múltiples implementaciones.



Blockchain

- Una Blockchain es una red compuesta por uno o más ordenadores (nodos).
- Cada nodo puede contener una copia completa de la Blockchain.
 - Una copia completa contiene todos los datos y transacciones.



Blockchain

- Simulador visual de Blockchain
 - Creado por Anders Brownworth
 - Demo general: <https://anders.com/blockchain>
 - Demo incluyendo pares de claves:
 - <https://andersbrownworth.com/blockchain/public-private-keys/>

The screenshot displays the 'Demo de Blockchain' interface. At the top, a dark navigation bar contains the title 'Demo de Blockchain' and several tabs: 'Hash', 'Bloque', 'Blockchain' (which is selected), 'Distribuido', 'Tokens', and 'Coinbase'. Below the navigation bar, the main heading 'Blockchain' is visible. The interface shows three blocks in a sequence, each with a light green border. Each block contains the following fields: 'Bloque: #' followed by a number (1, 2, and 3 respectively), 'Nonce:' followed by a number (11316, 35230, and 12937 respectively), and 'Datos:' followed by a large empty text area. At the bottom of each block, there is a label 'Anterior:' followed by a long alphanumeric string representing the previous block's hash.



Blockchain

- Simulador visual de Blockchain
 - El simulador valida que los Hashes comiencen por 0000.
 - *Sin embargo*
 - En realidad, lo que se busca es que el Hash sea inferior a un valor decimal concreto.
 - El hash es un número hexadecimal, convertible a decimal
 - P.e.:

Valor hexadecimal	Valor decimal
0x00000c52990ee86de55ec4b9b32beefd745d71675dc0eddfbc7b88336e2e296b	8.504e+70
0x21e5ccbe173e48d84ae64b6bafd0f0509aaf3ad45226bcef8da17dc3c9fddb0d	1.533e+76



Blockchain

- Minado de un bloque
 - Ejemplo con datos arbitrarios y asumiendo un umbral = 1000.

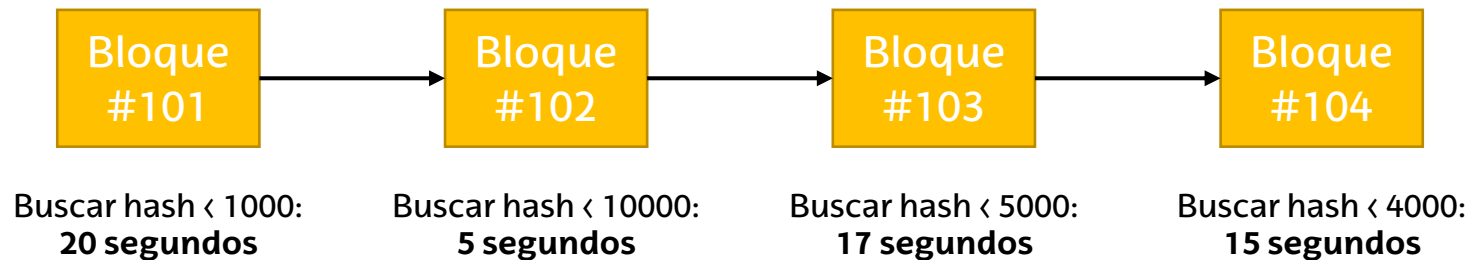
Datos	Nonce	Hash	Hash en base 10	¿Menos que 1000?
Hola Mundo	0	a231412f	2721136943	No
Hola Mundo	1	cbc1491	213652625	No
Hola Mundo	2	0ca24525	211961125	No
Hola Mundo	3	a9ee9d1	178186705	No
Hola Mundo	4	1488bec	21531628	No
	...			
Hola Mundo	27	0003bb	955	Si



Blockchain

- Minado de un bloque

- El tiempo de bloque se define como el tiempo aproximado que se tarda en crear un bloque nuevo en una Blockchain.
- El número de umbral se va ajustando para que siempre se tarde un tiempo parecido en calcular todos los bloques
 - Ejemplo: Tiempo de bloque objetivo = 15 segundos.



- *Motivo*: el # de nodos en una Blockchain puede variar.
- *Objetivo*: misma dificultad de crear un bloque constantemente.



Blockchain

- Minado de un bloque

- El cálculo del *nonce* se utiliza como algoritmo Proof-of-work
 - Resolución de un problema matemático complejo – pero no imposible.
 - Hace que la red sea resistente a ciertos ataques.
- Cada Blockchain tiene un tiempo de bloque esperado diferente¹:

Blockchain	Tiempo de bloque esperado
Bitcoin	10 minutos
Ethereum	15 segundos
Litecoin	2.5 minutos



¹Fuente: <https://blog.bitstamp.net/post/what-is-block-time/>

Blockchain

- La red funciona con las siguientes reglas:
 - 1) Cada nueva transacción se envía a todos los nodos.
 - 2) Cada nodo recopila las nuevas transacciones en un nuevo bloque.
 - 3) Cada nodo busca un *proof-of-work* difícil para su bloque.
 - 4) Cuando un nodo encuentra un *proof-of-work*, envía su bloque al resto de nodos.
 - 5) Los nodos aceptan el bloque sólo si todas las transacciones que contiene son válidas.
 - 6) Una vez se acepta el bloque, los nodos comienzan a trabajar en un nuevo bloque de la cadena.
 - Se usa el hash del bloque aceptado como hash anterior.



Blockchain

- *¿Qué sucede si 2 nodos envían diferentes versiones del siguiente bloque simultáneamente?*
 - Algunos nodos recibirán una versión, otros la otra.
- Solución:
 - Los nodos trabajan en la versión que reciban primero, pero guardan la otra.
 - El empate se rompe cuando se encuentre el siguiente *proof-of-work*, que indica cuál de los bloques hay que utilizar.
 - Los nodos siempre deben trabajar en la cadena más larga.



Blockchain

- El sistema *proof-of-work* mantiene la consistencia en una red Blockchain.
- Sin embargo, su uso tiene implicaciones negativas.
 - Requiere mucha capacidad de cómputo.
 - Implica un gasto elevado de electricidad.
- Algunas Blockchain están migrando a *proof-of-stake*.
 - Funciona como un sistema de pujas¹.
 - Menor impacto ambiental.



¹Proof-of-Stake (vs proof-of-work): https://youtu.be/M3EFi_POhps

Ejercicio 2

- Entrar en <https://gugger.guru/blockchain-demo/>
 - Es una variación del simulador de Anders Brownworth
- Activar el modo experto y, en la sección “Blockchain”, introducir datos aleatorios en, al menos, 3 de los bloques.
- Cambiar el nivel de dificultad y minar los bloques para que la cadena sea consistente.
 - Primero, usando dificultad = 6.
 - Después, usando dificultad = 8.
 - ¿Cuál es la diferencia de tiempo entre las diferentes dificultades?



Blockchain

- Existen 2 tipos de Blockchain:
- Sin permiso / Públicas:
 - Cualquier persona/entidad puede participar y contribuir.
 - Totalmente descentralizadas.
 - P.e. Bitcoin, Ethereum, Iota, ...
- Con permiso / Privadas:
 - La participación no está abierta al público.
 - Suelen estar gestionadas por una empresa/consorcio.
 - P.e. Hyperledger Fabric, Quorum, Corda, ...



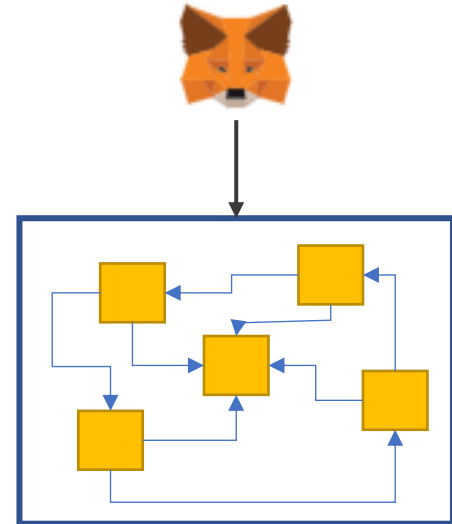
Ethereum

- Nació como un fork de Bitcoin en 2013
 - Propuesto por Vitalik Buterin
 - Artículo en su *blog*:
 - <https://web.archive.org/web/20131228111141/https://vbuterin.com/ethereum.html>
- Criptomoneda en uso: 1 ETH \approx 1.509 €
 - A fecha 3 de febrero de 2023 ([xe.com](https://www.xe.com/)).
- Capacidades más allá de una Blockchain básica
 - Además de transacciones económicas, permite ejecutar aplicaciones.



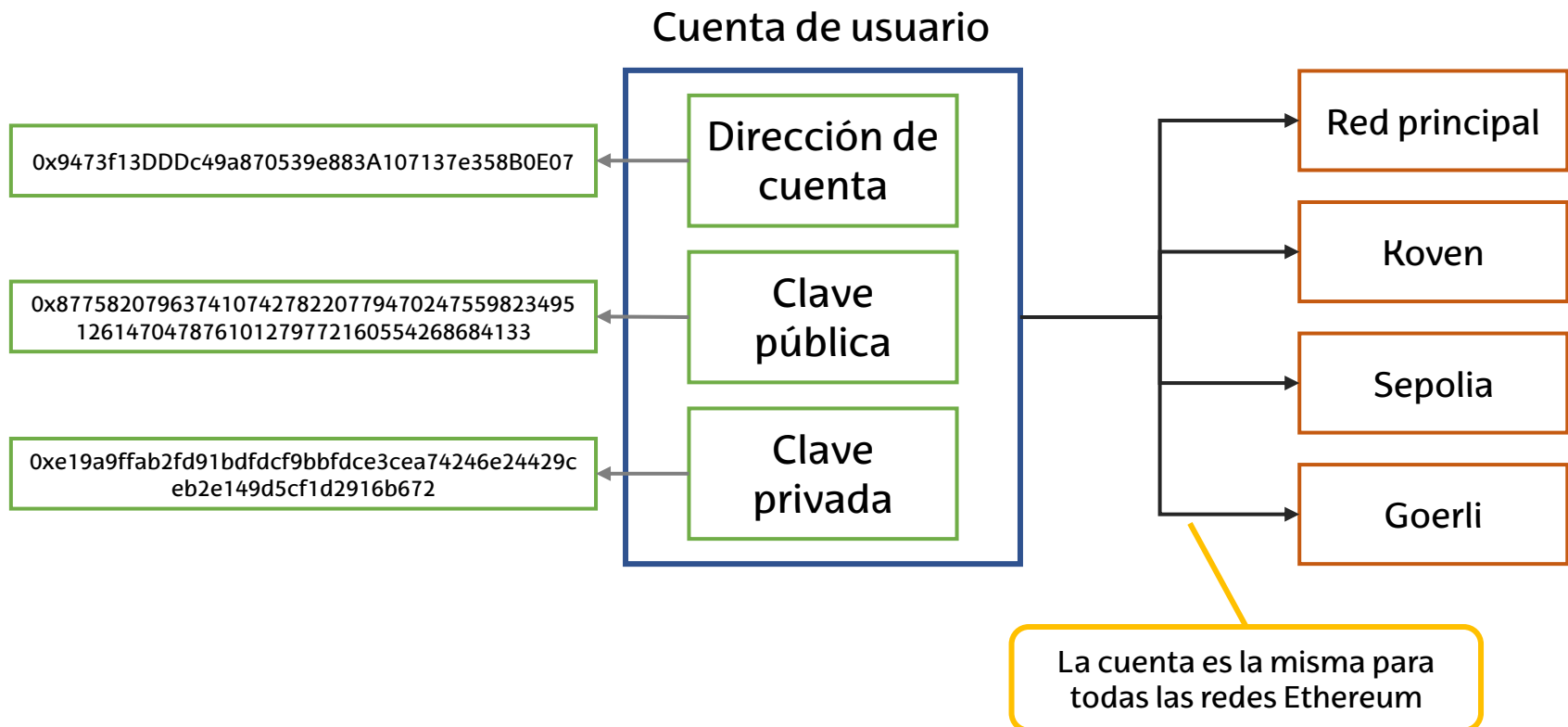
Ethereum

- Existen múltiples redes Ethereum
 - Red principal: asociada a la moneda ETH
 - Redes de prueba públicas: Goerli, Sepolia, ...
 - Redes privadas
- *Metamask*
 - Cliente para conectarse a Ethereum
 - No crea un nodo Ethereum localmente
 - URL: <https://metamask.io/>
 - Multi-plataforma:
 - Utilizaremos el Add-on de Chrome



Ethereum

- Al crear una cuenta en Metamask:



Ethereum

- *Conceptos concretos en Ethereum:*
- Ether: moneda de Ethereum
 - También representada como ETH.
- Wei: unidad, fracción de Ether.
 - $1 \text{ Ether} = 10^{18} \text{ Wei}$.
- Gas: unidad de cómputo.
 - Sirve para representar el cómputo necesario para ejecutar/incluir una transacción en un nuevo bloque.



Ethereum

- Realizar una transacción en Ethereum tiene un coste.
 - El coste es: *gasto de gas * (precio base del gas + propina)*.
 - Para evitar que el coste de la transacción sea muy elevado, el creador puede establecer un límite de gasto de gas.
- *Ejemplo:* Unai le quiere enviar a Mikel 1 ETH.
 - Unai establece un límite de gas de 21.000 unidades.
 - La red establece un precio de gas de 10 Wei/unidad.
 - Unai establece una propina de 2 Wei/unidad.

Coste: $21.000 * (10 + 2) = 252.000 \text{ Wei (o } 0.000252 \text{ ETH)}$



Ethereum

- Sobre el límite de gas:
 - Poner un valor muy alto asegura la ejecución pero puede incurrir en costes excesivos.
 - Poner un valor muy bajo puede implicar que no se adjudique tanto gas como el necesario para ejecutar la transacción. De ser así, la transacción falla.
- Sobre las "propinas":
 - Las transacciones que incluyen mayor propina obtienen mayor prioridad para ser ejecutadas e incluidas en un bloque.



Ejercicio 3

- *Realizar este ejercicio por parejas, seréis A y B.*
- A debe obtener 0.2 ETH en su cuenta de la red Goerli.
 - Utilizar <https://goerlifaucet.com/>
- A envía 0.1 ETH a B usando MetaMask.
- Cuando B haya recibido 0.1 ETH, buscad en Etherscan:
 - En qué # de bloque se ha incluido la transacción.
 - Cuántas transacciones contiene ese bloque.



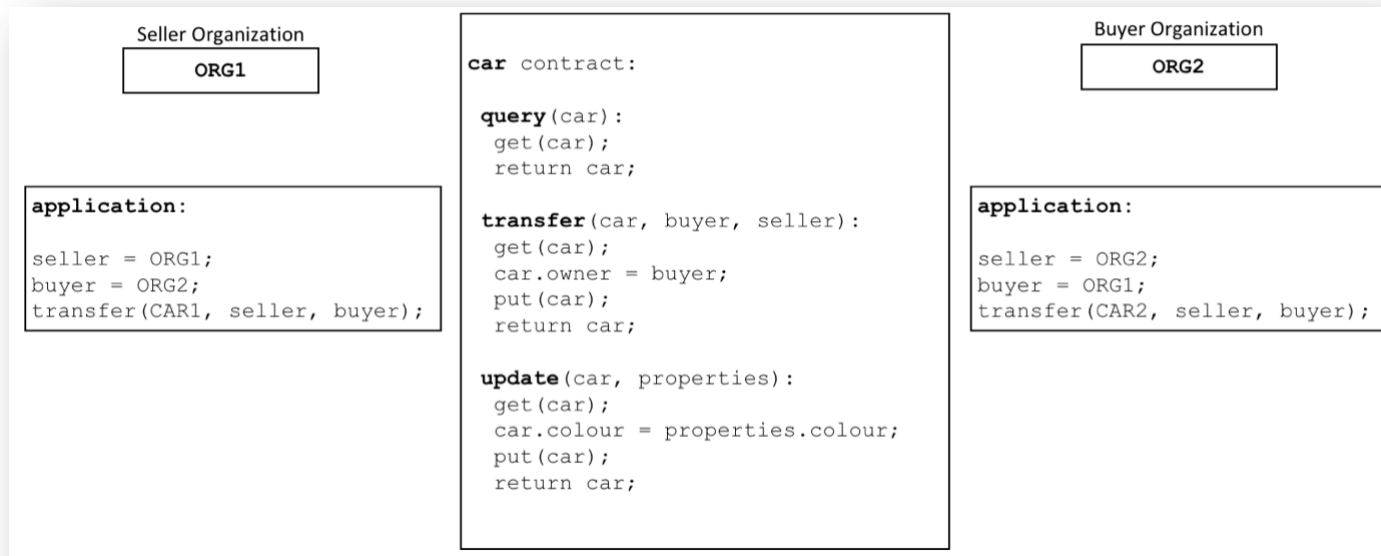
Smart Contracts

- Código compilado que se almacena y ejecuta en la Blockchain.
 - Expanden las capacidades de Blockchain más allá del intercambio monetario.
- Permiten definir una lógica de negocio.
 - Tipos de datos, validaciones, procesos, ...
- Introducidos inicialmente en Ethereum.
 - Se programan en el lenguaje Solidity.
 - Ahora disponibles en otras Blockchain.



Smart Contracts

- Escenario de ejemplo:
 - 2 organizaciones ORG1 y ORG2.
 - 1 Smart Contract "car" con diferentes operaciones.
 - Las aplicaciones de cada organización pueden usar "car" para validar la transferencia de propiedad de un coche.



Smart Contracts

- Ejemplo de Smart Contract

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.17;

contract SimpleStorage {
    uint public num;

    function set(uint _num) public {
        num = _num;
    }

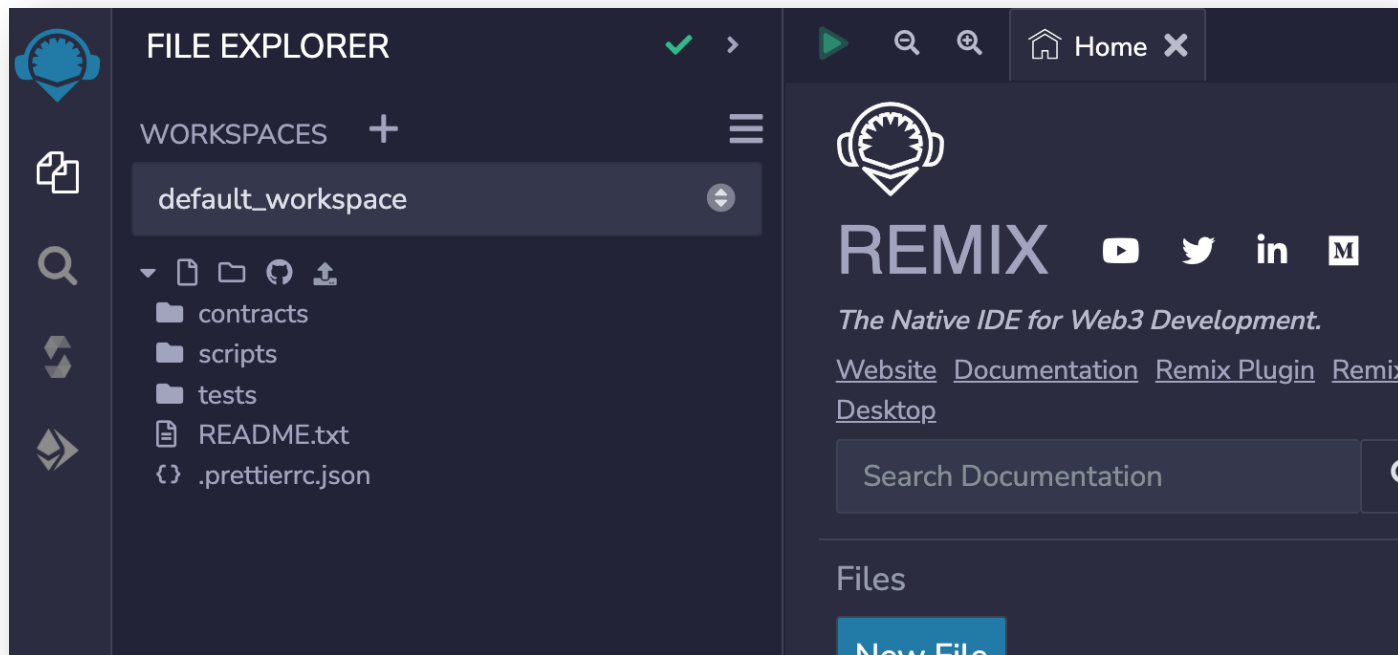
    function get() public view returns (uint) {
        return num;
    }
}
```

Es obligatorio indicar el tipo de licencia como comentario y la versión de Solidity



Smart Contracts

- Usaremos el editor online <https://remix.ethereum.org/>
 - Permite crear, desplegar y usar contratos en redes Ethereum.



Smart Contracts

- Para que otros usuarios puedan utilizar un contrato desplegado por nosotros:
 - 1) Nosotros tenemos que validarlo.
 - Incluir su información de compilación y código fuente en EtherScan.
 - 2) Los usuarios deben utilizar la dirección del contrato y la información ABI.
 - Se puede obtener de EtherScan:
<https://docs.alchemy.com/docs/integrating-your-smart-contract-with-the-frontend#loading-your-hello-world-smart-contract>



Smart Contracts

- Una vez que un contrato se ha desplegado en la Blockchain, no se puede eliminar.
- Es posible modificarlo para que se marque como inhábil y que no se pueda utilizar.
 - Añadir una función *selfdestruct*.
 - Más información: <https://ethereum-blockchain-developer.com/022-pausing-destroying-smart-contracts/04-destroy-smart-contracts/>



Ejercicio 4

- *Realizar este ejercicio en parejas, seréis A y B.*
- A define un contrato con:
 - 3 variables de tipo String: vendedor, comprador y ítem.
 - Una variable "precio" de tipo int.
 - Métodos set que definan el valor de cada variable.
- A despliega el contrato en Gorlei.
 - Rellena "vendedor" con su nombre y "ítem" con el modelo de su teléfono móvil.
- B accede al contrato.
 - Rellena los campos "comprador" con su nombre y "precio" con 100.
- A y B comprueban que las transacciones son correctas.



HyperLedger Fabric

- Infraestructura Blockchain “con permiso” (privada)
 - Originalmente desarrollada por IBM.
 - Ahora mantenida por la fundación Linux.
- Arquitectura modular
 - Compatible con Smart Contracts
- Código abierto
 - Repositorio oficial: <https://github.com/hyperledger/fabric>
- Orientada al desarrollo de aplicaciones y soluciones
 - Presente en entornos industriales



HyperLedger Fabric

- Caso de estudio 1¹:
 - *Empresa:* Walmart USA
 - *Problema:* Cuando se detecta una enfermedad en los alimentos que se proveen en las tiendas, puede llevar semanas encontrar su origen.

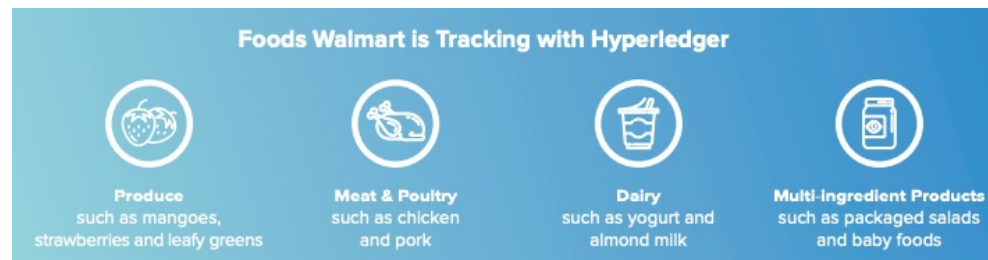


HyperLedger Fabric

- Caso de estudio 1¹:
 - *Empresa:* Walmart USA
 - Diseñaron una aplicación piloto centrada en Blockchain para trazar los alimentos desde su origen hasta la venta.
 - Consiguieron reducir el tiempo de trazado de 7 días a 2.2 segundos.



- Actualmente lo están aplicando a más alimentos.



¹HyperLedger Foundation, Walmart case study: <https://www.hyperledger.org/learn/publications/walmart-case-study>

HyperLedger Fabric

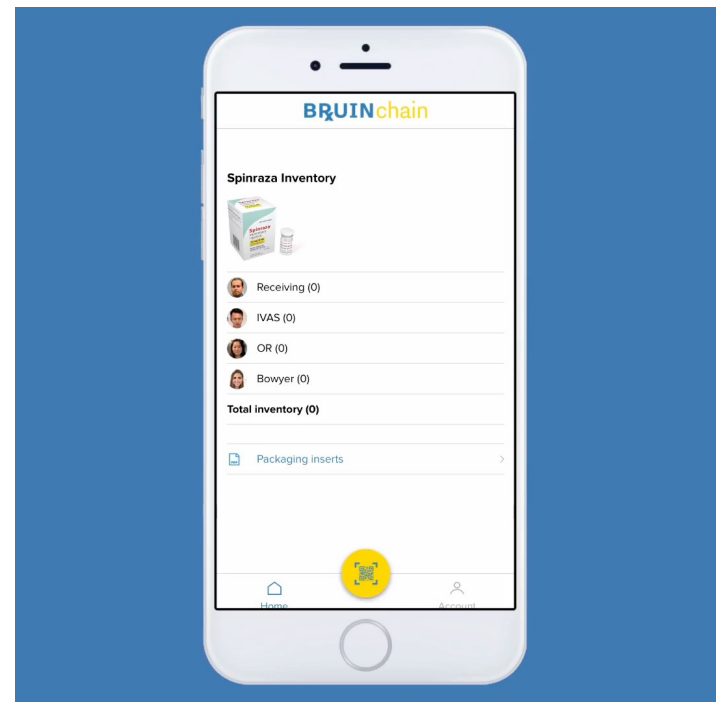
- Caso de estudio 2¹:
 - *Empresa:* BruinChain
 - *Problema:* Un estudio estimó que en 2019 se proporcionaron 100 millones de medicamentos de forma incorrecta en USA:
 - Problemas de fabricación
 - Errores de transporte, p.e. en la cadena de frío
 - Errores de distribución.
 - *Estos problemas se agravaron en 2020.*



¹Hyperledger Foundation, LD case study: <https://www.hyperledger.org/learn/publications/ledgerdomain-case-study>




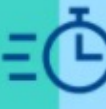
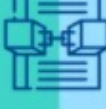
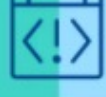

HyperLedger Fabric

- Caso de estudio 2¹:
 - *Empresa:* BruinChain
- Se desarrolló un prototipo que permite trazar un medicamento por diferentes fases:
 - Transporte
 - Cadena de frío
 - Farmacia
 - Paciente
- Ha contribuido a reducir la falsificación y contrabando de medicamentos.



¹Hyperledger Foundation, LD case study: <https://www.hyperledger.org/learn/publications/ledgerdomain-case-study>

Comparativa Ethereum y HLF¹

ETHEREUM VS. HYPERLEDGER FABRIC			
	ETHEREUM		HYPERLEDGER FABRIC
 Ledger Type	Permissionless		Permissioned
Governance	Ethereum developers		Linux Foundation
Cryptocurrency	Ether (ETH)		None
Throughput	~ 20 tps		> 2000 tps
Consensus Mechanism	Proof of Work (PoW)		Pluggable Mechanism
Smart Contract	✓		✓
Smart Contract Language	Solidity		NodeJS or Golang or Java
Application Type	Wide-ranging		Wide-ranging



¹Imagen de: <https://101blockchains.com/blockchain-infographics>

Web3

- Concepto abstracto: *la web que otorga propiedad de los activos digitales a sus usuarios.*
 - Evolución de la Web 2.0
- Tecnologías base:
 - Blockchain
 - Criptomonedas
 - Non Fungible Tokens (NFT)
- Más información: <https://ethereum.org/es/web3/>



Web3

- Hay bastante escepticismo al respecto:

‘I don’t think Web3 was that big or that metaverse stuff alone was revolutionary but AI is quite revolutionary’: Bill Gates

Published: Jan. 12, 2023 at 10:12 a.m. ET

By Anushree Dave [Follow](#)

Bill Gates answered a range of questions in an “Ask Me Anything” Reddit post on Wednesday evening



TECNOLOGÍA

La web3 no es lo mejor de la web1 y la web2, sino lo peor de ambas

La web3 es la nueva palabra de moda en internet, pero, ¿es realmente oro todo lo que reluce o es simplemente una buzzword más?

por Félix Palazuelos

25 de enero de 2022



Enlaces:

- <https://www.marketwatch.com/story/i-dont-think-web3-was-that-big-or-that-metaverse-stuff-alone-was-revolutionary-but-ai-is-quite-revolutionary-bill-gates-11673536355>
- <https://hipertextual.com/2022/01/web3-analisis-web1-web2>



Auditoría y Blockchain: Resumen

- Blockchain es más que un soporte para criptomonedas.
 - Es un sistema de gestión de datos en forma de libro de cuentas.
- Se basa en bloques de transacciones firmados por Hashes.
 - Los nodos de la red Blockchain compiten para encontrar la combinación Hash + *Nonce* de cada nuevo bloque ("minar").
- Los Smart Contracts permiten ejecutar código dentro de una Blockchain.
 - Los datos resultantes son inmutables.



Bibliografía

- R. Elmasri & S. B. Navathe, "Fundamentals of Database Systems", 7th ed., Pearson, 2016.
- A. Silberschatz, H. F. Korth, S. Sudarshan, "Database Systems Concepts", 7th ed., McGraw Hill, 2019.
- S. Grider, "Ethereum and Solidity: The Complete Developer's Guide". Udemy, 2021.
- Hyperledger Fabric Docs, 2022.
 - <https://hyperledger-fabric.readthedocs.io/>

