UNIVERSIDAD DE MÁLAGA

E.T.S. INGENIERÍA
INFORMÁTICA
UNIVERSIDAD DE MÁLAGA

GRADO EN INGENIERÍA DEL SOFTWARE

Aplicabilidad de tecnologías Blockchain diseñadas para entornos IoT

Applicability of IoT-designed Blockchain technologies

Realizado por
Pablo Gamarro Lozano

Tutorizado por
Rodrigo Román Castro

Departamento
Lenguajes y Ciencias de la Computación
UNIVERSIDAD DE MÁLAGA

MÁLAGA, SEPTIEMBRE 2021

Agradecimientos

En primer lugar, me gustaría agradecer a mi tutor, Rodrigo Román Castro, por haberme tutorizado y por haber enfocado el proyecto hacia las tecnologías Blockchain. Me ha guiado y atendido en todo momento, y me ha enseñado nuevas referencias y métodos que me han sido de gran ayuda para el proyecto que seguramente me servirán para toda mi carrera profesional. Gracias a la Universidad de Málaga, por haberme facilitado los medios necesarios para la realización de este Trabajo Final de Grado, por mostrarme las distintas ramas de la informática que desconocía, por formarme específicamente en el desarrollo software y por conformar una bonita etapa en mí vida.

También me gustaría agradecer a todos aquellos investigadores y desarrolladores que con su esfuerzo, dedicación y trabajo, me han facilitado la investigación sobre el tema de mi TFG. Sin ellos y sus hallazgos, no hubiese sido posible siquiera plantearme seguir esta línea temática.

Por último, pero no por ello menos importante, le dedico este TFG y les doy las gracias a mi familia y amigos, en especial a mis padres Inma y Pablo, a Bety, a mi primo Rafa, a Sergio y a Christian, que han supuesto en todo momento un apoyo moral muy importante, ayudándome a superar los obstáculos que el día a día durante estos años de carrera y un proyecto de tal envergadura llevan consigo.

Gracias.

Resumen

En este proyecto se ha realizado un estudio de aplicabilidad de la tecnología Blockchain en dispositivos IoT. Para ello, primero se ha realizado el desarrollo de ambos conceptos y se han analizado de forma teórica las soluciones propuestas por diferentes organizaciones que mencionan de forma explícita la IoT como su principal campo de aplicación. Y segundo, se ha desarrollado un caso práctico como prueba de concepto para validar el análisis anterior, demostrando como la blockchain más óptima (según el estudio anterior) puede ejecutar una aplicación IoT en un hardware de bajo coste (p.ej. Raspberry Pi).

La prueba de concepto consiste en un sistema de 3 aplicaciones que serán capaces de gestionar un sistema de iluminación de pistas de una instalación deportiva. La primera, un smart contract desplegado en la Testnet de IoTeX; la segunda, una aplicación web a través de la cual se interactúa con el estado del smart contract; y la tercera, una aplicación que únicamente leerá el estado del smart contract para decidir cuándo encender las luces mediante los pines de salida de una Raspberry Pi.

Palabras clave: **Blockchain - Internet de las Cosas - Contratos Inteligentes - IoTeX - Plataformas web**

Abstract

In this project, an applicability study of blockchain technology in IoT devices has been carried out. To do this, the development of both concepts has first been carried out and the solutions proposed by different organizations that explicitly mention the IoT as their main field of application have been theoretically analyzed. And second, a practical case has been developed as a proof of concept to validate the previous analysis, demonstrating how the most optimal blockchain (according to the previous study) can run an IoT application on low cost hardware (eg Raspberry Pi).

The proof of concept consists of a system composed by 3 applications that will be able to manage a track lighting system of a sports facility. The first, a smart contract deployed on the IoTeX Testnet; the second, a web application through which you interact with the status of the smart contract; and the third, an application that will only read the status of the smart contract to decide when to turn on the lights through the output pins of a Raspberry Pi.

Keywords: **Blockchain - Internet of Things - Smart Contracts - IoTeX - Web platforms**

Índice

1

Introducción

En este apartado se explicará el contenido de la memoria, así como antecedentes, motivación, objetivos y conceptos clave.

1.1 Antecedentes y motivación

Una blockchain (o cadena de bloques) es una estructura de datos que permite asegurar y verificar transacciones realizadas entre entidades sin necesidad de una tercera parte que proporcione la confianza. Las tecnologías que utilizan dicha estructura de datos como base permiten crear un ecosistema distribuido, en el que varias entidades (que pueden conocerse y autenticarse entre sí en un entorno privado, o ser completas desconocidas dentro de un entorno público) pueden colaborar entre sí. Al principio, las tecnologías Blockchain se centraron en la creación de monedas digitales (conocidas comúnmente como criptomonedas). Sin embargo, en los últimos años, las tecnologías Blockchain han encontrado múltiples áreas de aplicación sobre las que se están desarrollando pruebas de concepto **[1]**, tales como las cadenas de suministro, los servicios financieros, y las redes eléctricas inteligentes. Además, existen varias soluciones empresariales basadas en plataformas de código abierto, como la solución TrustOS de Telefónica (basada en la plataforma Hyperledger) que se utilizará para desplegar una red blockchain que conecte a todos los parques tecnológicos españoles **[2]**. Finalmente, dentro del programa Horizonte 2020 **[3]**,existen múltiples proyectos de investigación que analizan las posibilidades y desafíos asociados a estas tecnologías.

Una de las áreas de aplicación en donde se puede aplicar las tecnologías Blockchain es la Internet de las Cosas, o IoT. La IoT se basa principalmente en la instalación de sensores y la capacidad de recopilación de información a través de estos, por lo que es necesario asegurar que la información es fiable y no ha sido modificada, o al menos trazar el origen de dicha información. En este caso, es interesante aprovechar las características de las

tecnologías Blockchain, para así mantener la información distribuida dentro de un círculo de confianza, de forma que los datos se mantengan inmutables y cada usuario tenga los medios necesarios para verificar el origen de la información y que esta no ha sido modificada.

Esta unión entre la IoT y Blockchain ha dado lugar a que se trate de vender ciertas tecnologías blockchain (o basadas en los principios Blockchain) como especialmente diseñadas y adecuadas para la IoT, pero que en cambio han resultado ser nefastas al contradecir muchos de los principios esenciales en la aplicación de dicha tecnología. Un claro ejemplo de ello es IOTA (https://www.iota.org/), cuyo funcionamiento depende de un nodo coordinador centralizado. En febrero de 2020, este nodo coordinador se apagó tras recibir un ataque **[4]**, lo cual habría inutilizado cualquier aplicación IoT que utilizase IOTA como base.

Dadas estas circunstancias, en caso de que se quiera desarrollar una aplicación IoT que se base en tecnologías blockchain supuestamente especialmente diseñadas para la IoT (como por ejemplo IoTeX, Iroha y Qtum), se hace necesario analizar dichas blockchains antes de aplicarlas en proyectos reales, ya que podemos acabar aplicando una solución ineficiente a problemas reales y ser víctima del marketing y el hype que a día de hoy aún plagan estas tecnologías.

1.2 Objetivos

Este proyecto tiene dos objetivos principales. Primero, realizar un análisis que muestre de forma teórica la aplicabilidad de aquellas tecnologías blockchain que mencionan de forma explícita la IoT como su principal campo de aplicación. Segundo, desarrollar un caso práctico como prueba de concepto que valide el análisis anterior, demostrando como la blockchain más óptima (según ese estudio anterior) puede ejecutar una aplicación IoT en un hardware de bajo coste (p.ej. Raspberry Pi).

1.3 Metodología

En este apartado se explicará cuál ha sido la metodología utilizada durante el desarrollo del proyecto.

Para el cumplimiento del primer objetivo, se procederá a un estudio analítico que estudie de forma exhaustiva los requisitos funcionales y no funcionales de las aplicaciones IoT, las propiedades técnicas de las tecnologías Blockchain aplicables a la IoT, y la adecuación de estas tecnologías a los requisitos anteriormente mencionados. Para el cumplimiento del segundo objetivo, se procederá a un desarrollo iterativo, en donde se desarrollará de forma incremental (partiendo desde un esqueleto del sistema y añadiendo funcionalidades en cada iteración) la aplicación IoT sobre la plataforma blockchain más idónea.

1.4 Conceptos clave

En esta sección se explican los conceptos claves que se utilizarán en adelante en el proyecto.

Blockchain: Es una tecnología cuyas ideas fundamentales se basan en los registros de contabilidad. Esta es soportada por el uso de una red de nodos y ordenadores para almacenar y validar la información de forma consensuada entre los nodos de la red.

Internet of Things: Es un término que hace referencia al conjunto de objetos físicos que tienen una inteligencia e identidad propia para integrarse e interactuar de manera independiente en Internet con otros dispositivos o usuarios.

Smart Contract: Es un programa capaz de ejecutarse y hacerse cumplir por sí mismo o a partir de eventos, de manera autónoma y automática, sin intermediarios ni mediadores.

Distributed Ledger Technology: La Tecnología de Libro Mayor Distribuido o DLT es un sistema electrónico o base de datos ejecutado por múltiples entidades para registrar información. De forma que permiten almacenar y utilizar datos que pueden ser descentralizados y distribuidos de forma pública o privada.

Función Hash, o Hash: Es un algoritmo matemático capaz de convertir un conjunto o bloque de datos en una serie de caracteres con una longitud fija independientemente de la longitud de los datos de entrada. Este tipo de función se diseña de forma que un pequeño cambio en la entrada cambie completamente el resultado obtenido, y que además sea irreversible obtener el conjunto de entrada a partir del resultado tras ser aplicada.

1.5 Acrónimos

En esta sección se enumeran los acrónimos que se utilizarán en adelante en el proyecto.

DLT - Distributed Ledger Technology
P2P - Peer To Peer
IoT - Internet of Things
DAG - Grafos Acíclicos Dirigidos
PBFT - Practical Byzantine Fault Tolerance
DApp - Decentralized application
M2M - Machine to Machine
LPWAN - Low Power Wide Area Networks
NB IoT - NarrowBand IoT
BLE - Bluetooth de baja energía
PAN - Personal Area Network
SOC - System-On-Chip
SBC - Single-Board-Computer
IIoT - Industrial IoT
PoW - Proof of Work
PoS - Proof of Stake
PoA - Proof of Authority
PBFT - Practical Byzantine Fault Tolerance
TIC - Tecnologías de la Información y Comunicación

DPoS - Delegated Proof of Stake
Roll-DPoS - Roll Delegated Proof of Stake
DID - Decentralized Identifiers
TEE - Trusted Execution Enviroment
UTXO - Unspent Transaction
EVM - Ethereal Virtual Machine
YAC - Yet Another Consensus
DGP - Decentralized Governance Protocol
SegWit - Segregated Witness
SPV - Simple Payment Verification
AAL - Account Abstract Layer
zk-SNARK - zero-knowledge Succinct Non-interactive ARgument of Knowledge
ARM - Advanced RISC Machine
LED - Light-Emitting Diode
HTML - HyperText Markup Language
CSS - Cascading Style Sheets
PHP - Hypertext Preprocessor
API - Application Programming Interface
ABI - Application Binary Interface

2

Estudio preliminar

En este capítulo se realizará una investigación sobre las DLTs, Blockchain e IoT para conocer sus aspectos técnicos, como funcionan, sus capacidades, hardware, protocolos red que utilizan, retos a superar o desventajas de las mismas y casos de uso ya implementados y funcionales.

2.1 Distributed Ledger Technologies

La Tecnología de Libro Mayor Distribuido o DLT es un sistema electrónico o base de datos ejecutado por múltiples entidades para registrar información. De forma que permiten almacenar y utilizar datos que pueden ser descentralizados y distribuidos de forma pública o privada **[5]**.

Las DLT suelen fundamentarse en tres tecnologías ya existentes. Las redes P2P, donde cada participante actúa tanto como servidor como cliente. La criptografía asimétrica, que mediante el uso de una clave privada y otra pública permite un intercambio seguro de información entre dos partes, dada sus capacidades de encriptar, desencriptar y firmar información. Y los algoritmos de consenso, que permiten a los participantes llegar a un acuerdo para lograr hacer nuevas escrituras al registro sin necesidad de conocerse ni confiar entre ellos.

Dadas sus características las DLT tienen el potencial de mejorar la eficiencia, diseño, seguridad y costes en aquellos servicios donde la eliminación del coste de mensajería y el uso de terceras partes de confianza para realizar transacciones supongan un problema, y donde se necesite reducir la complejidad de las transacciones y aumentar la transparencia y trazabilidad de estas.

Una de sus principales capacidades es que consiguen solventar el problema del doble gasto. El problema del doble gasto es un problema del dinero digital en el que una misma moneda digital se puede gastar más de una vez. Esto es posible ya que cada moneda consta de un archivo digital que se puede duplicar o incluso falsificar. Como con el dinero falsificado, el doble gasto también lleva a la inflación dado que se crean monedas que no existían, devaluando así el valor de dicha moneda respecto a otras y disminuyendo la confianza entre los usuarios.

Sin embargo, las DLT no están exentas de riesgos y tienen sus limitaciones, tales como el trilema entre la descentralización, la robustez y escalabilidad. Además, en algunos casos, su funcionamiento plantea retos legales (tales como la regulación acerca de cómo debe interpretarse el «derecho al olvido» o un marco jurídico para el reconocimiento de las cadenas de bloque como fuentes de veracidad inmutables y a prueba de manipulación **[6]**). También algunas de sus implementaciones pueden tener un gran impacto en el medioambiente, debido al gasto energético.

Una de las instanciaciones más famosas de las DLT es la tecnología Blockchain. Sin embargo, también existen otras implementaciones con diferentes características:

- **Grafos Acíclicos Dirigidos (DAG)** que se explicará en profundidad en el análisis de IOTA en este proyecto.

- **Hashgraph** es una DLT que para mejorar su eficiencia se fundamenta en el algoritmo chismes sobre chismes (Algoritmo Gossip), en el que cada nodo de forma aleatoria va informando al resto de los detalles de las transacciones que se van añadiendo, y estos a su vez informan a otro nodo aleatorio, creando una red de nodos donde todos los nodos tienen el hash del nodo anterior. Para más tarde realizar una votación virtual, que es la forma en la que un nodo sabe si una transacción es válida o no, y que también sirve para decidir el orden de las transacciones (ya que estas se pueden almacenar en una misma marca de tiempo). Una vez que se han procesado una cantidad definida de transacciones, se inicia una votación donde cada participante busca el evento que mejor se ajusta a la red. Este evento se conoce como testigo "famoso". Aquellos que son elegidos contienen la información sobre eventos antiguos que se registran en los nodos y si el nuevo encaja con el anterior, entonces se vota como sí. Convirtiéndose el evento con mayor cantidad de votos en el testigo "famoso" de esa ronda. Dicho evento luego proporciona las órdenes de transacción.

- **Holochain** es otra red distribuida que está centrada en el agente en lugar de en la estructura centrada en los datos. La técnica para ello es la validación DHTs (Distributed Hash Tables), donde cada usuario tiene su propia cadena que ha de cumplir un conjunto de reglas llamadas "ADN" para ser confiable. Al contrario que en las estructuras centradas en los datos, donde todos los nodos de la red

se ven obligados a verificar las transacciones individuales de una cola y agregarlas a la cadena, volviéndose más lenta a medida que son agregadas. Con el uso del ADN evitan el uso de algoritmos de consenso globales, resolviendo problemas relacionados con la escalabilidad.

- **Tempo**. Para superar las limitaciones de escalabilidad y eficiencia Radix ha desarrollado una base de datos distribuida junto a un algoritmo de consenso llamado Tempo. En él las transacciones se agregan al registro en orden de evento (cada uno relativo al anterior) en lugar de marcas de tiempo, ya que estas últimas pueden dar lugar a inconsistencias a la hora de aceptar los datos. Y para mejorar su eficiencia y no tener que indexar la base de datos cada vez que se añade una gran cantidad de información hacen uso de la fragmentación, la base de datos global se divide en partes llamadas "shards" con identificadores únicos que se les atribuye a cada nodo. A su vez, Tempo es asíncrono, por tanto, no existe tiempo de bloqueo, y al mismo tiempo es detector de fallas bizantinas, es decir, que también es capaz de reconocer e interrumpir violaciones del protocolo de un sistema no permisionado.

2.2 Blockchain

Blockchain son libros de contabilidad digitales a prueba de manipulaciones implementados en un sistema distribuido y sin una autoridad central. A su nivel más básico, permiten que una comunidad de usuarios registre transacciones en un registro compartido por dicha comunidad, de tal forma que bajo un funcionamiento correcto de la red blockchain las transacciones una vez publicadas no pueden modificarse **[7]**. Fue aplicada por primera vez en 2009 como parte de Bitcoin.

La información dentro de Blockchain se organiza en conjuntos de transacciones, los bloques, de un tamaño que normalmente es limitado para facilitar su minado (ver sección 2.2.3). Una vez que dichas transacciones son validadas, al bloque se le añade el hash del bloque anterior y se le aplica una función hash que se utiliza como cabecera en el siguiente bloque.

2.2.1 Características

2.*2.1.1 Inmutabilidad de datos* y seguridad

Gracias al diseño de Blockchain, donde varias entidades colaboran de forma descentralizada para mantener la consistencia de la información, es posible garantizar la inmutabilidad de los datos una vez estos se guarden dentro de la Blockchain. Así, la descentralización e inmutabilidad solucionan varios problemas, incluyendo i) evitar que personas o autoridades que forman parte del sistema modifiquen datos o añadan nuevos datos inconsistentes sin la verificación del resto de participantes, y ii) evitar que un nodo corrupto pueda estafar en una transacción al resto de nodos proporcionando datos manipulados de transacciones anteriores **[8]**.

2.2.1.2 Descentralización

La descentralización es una parte fundamental del diseño de la tecnología Blockchain, ya que una parte de su seguridad se fundamenta en ello. La descentralización además hace posible que una sola persona, grupo o autoridades concretas no tengan el control o privilegios del sistema, de forma que al usuario se le deja en una posición donde ellos pueden directamente almacenar sus documentos, criptomonedas, contratos inteligentes y otros tipos de datos digitales sin necesidad de un tercero que actúe como autoridad de confianza. Esto provoca que todas las partes intervinientes puedan confiar plenamente en el sistema. También se consigue un sistema más fiable al ejecutarse de forma distribuida, es decir, menos tendente a averías que inutilicen el sistema completamente, como el que podría ser provocado por un ataque de denegación de servicios (DDoS). También reduce considerablemente las tareas de registro y control de datos en las transacciones, evitando duplicidades de registro, es decir, que exista solo un registro distribuido independientemente del número de intervinientes (sin Blockchain, si hay tres intervinientes supone que cada una de las tres partes haga el registro de la transacción en su propio registro).

2.2.1.3 Transacciones en tiempo real y anonimato

Las transacciones son realizadas en tiempo real con la entidad con la que se trata, sin intermediarios, por tanto, su inmediatez minimiza el riesgo producido en comparación con otras transacciones, donde el pago puede llegar a tardar días en ejecutarse (evitando concursos de acreedores, riesgos de impago, fraudes… durante el proceso de pago). Y a diferencia de los libros contables de empresas a las que sólo el personal autorizado puede acceder, cualquier usuario de blockchain (pública) puede ver y comprobar las transacciones realizadas. Que estas transacciones sean transparentes a todos los usuarios no quiere decir que se conozcan las personas detrás de las mismas. Están los casos en los que mediante el uso de pseudónimos su usuario puede mantener su privacidad como Bitcoin, como se ve en la figura 1, donde aparecen direcciones, cantidad enviada, cuota, etc.

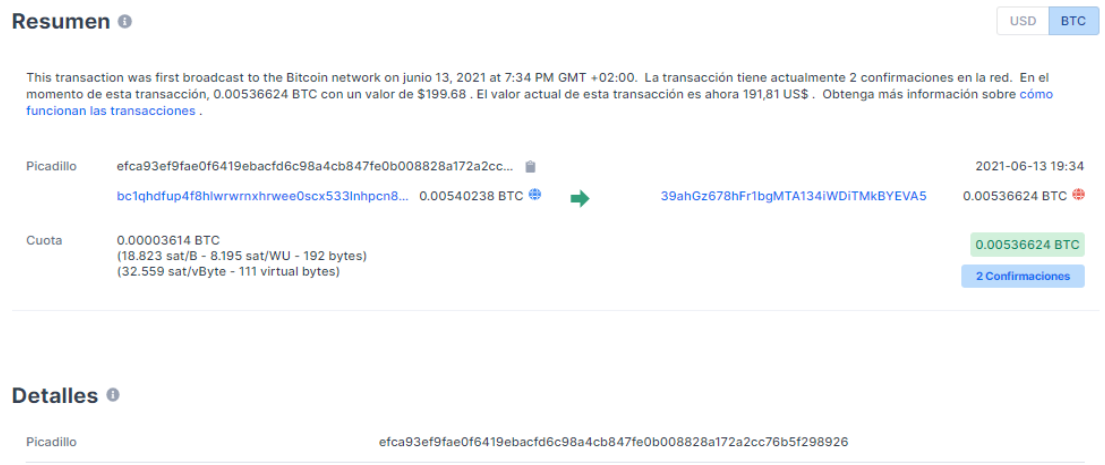

Figura 1. Ejemplo de transacción Bitcoin.

Y también están los casos en los que se utilizan técnicas reales de anonimato como en zCash, que mediante técnicas criptográficas basadas en pruebas de cero conocimiento permite ocultar las direcciones a las que se envían las criptomonedas, y la cantidad de las mismas, quedando grabado en la blockchain como se muestran en las figuras 2 y 3.

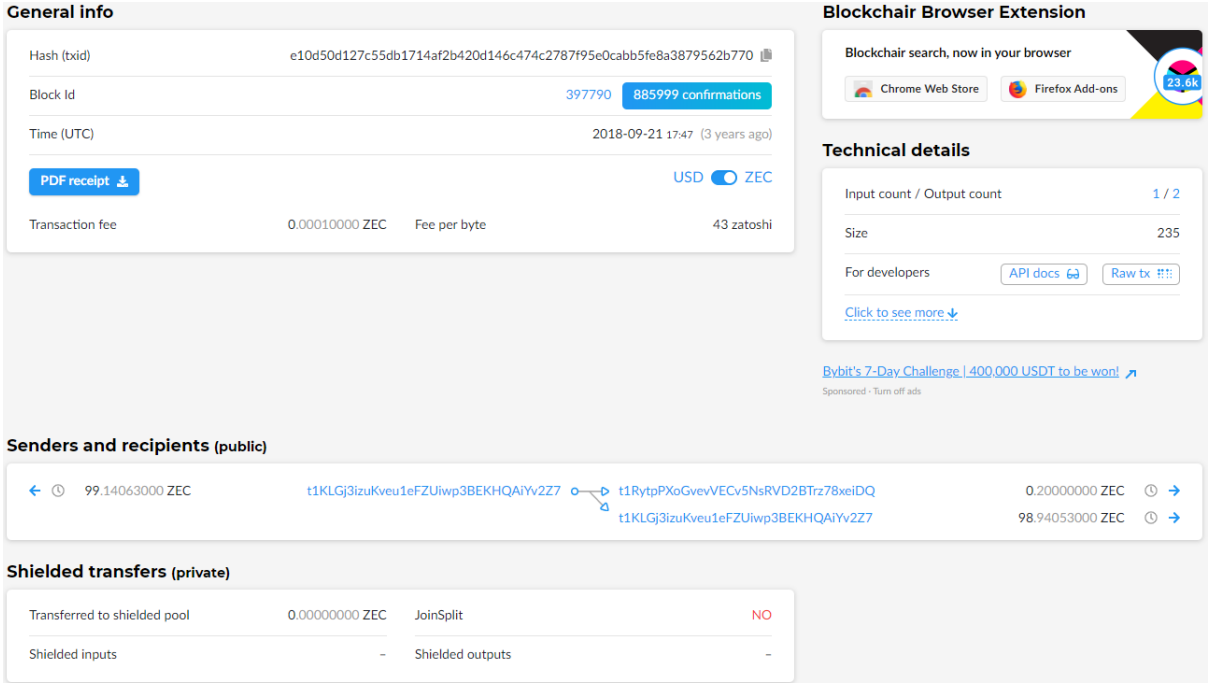

Figura 2. Transacción en zCash con direcciones públicas.

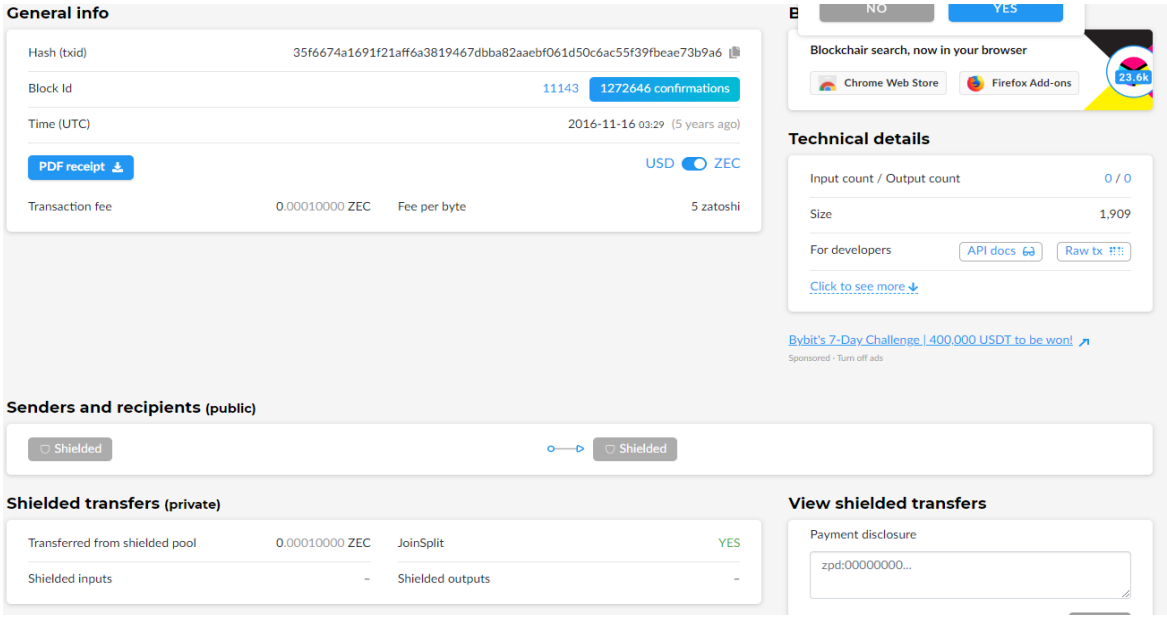

Figura 3. Transacción en zCash con direcciones ocultas.

2.2.1.4 Desventajas

La tecnología Blockchain también presenta varias desventajas. Un ejemplo son los casos donde la inmutabilidad de la información se puede convertir en un problema. En caso de guardar información personal, dicha información no puede eliminarse – violando así uno de los principios de la ley de protección de datos. Otro problema es la imposibilidad de recuperar el acceso a una cuenta de la que hemos olvidado su clave, siendo esta la razón por la que se debe tener mucho cuidado con olvidar o perder las claves privadas. También la velocidad del procesamiento de la información puede variar, debido a razones como un fallo en la red, o que las comisiones por procesamiento se reduzcan – haciendo que los mineros estén menos dispuestos a realizar las tareas. Aunque esto también puede suceder al revés, que la red se congestione y el uso de la misma por parte de los usuarios se mantenga, de forma que los mismos usuarios ajusten el precio que están dispuestos a pagar como cuota por unidad de procesamiento computacional (comúnmente llamada gas) para que los mineros les atiendan con mayor antelación, aumentando así el precio general según la oferta y demanda. Finalmente, otras desventajas importantes incluyen los problemas de escalabilidad, la lentitud en la velocidad de las transacciones (salvo en las blockchain permisionadas), y el aumento en algunos casos exponencial del tamaño en memoria que ocupa la propia blockchain.

2.2.2 Tipos de Blockchain

Todas las Blockchain tienen unos fundamentos base sobre las que se construyen pero no todas han de tener las mismas características, ya que existen diferentes casos de uso para estas. Principalmente se diferencian hasta 4 tipos **[9]**:

El primer tipo son las *públicas no permisionadas*, cualquiera puede unirse a estas blockchains y ser partícipe de su red. En estas no existe la figura de administrador, y el minado de bloques suele ser recompensado económicamente. Todos sus usuarios son anónimos y la información es almacenada y mantenida por todos los usuarios que lo deseen de forma masivamente distribuida. Estas blockchains atienden normalmente a los casos de uso de criptomonedas.

El segundo tipo son las *privadas no permisionadas*, en este tipo de blockchain la información es privada pero mantiene las características de las blockchain no permisionadas. Sus casos de uso tienden a tener que ver con votaciones.

El tercer tipo son las *privadas permisionadas*. Son aquellas que implementan restricciones tanto en quien participa en la red como en el tratamiento de la información. De esta forma se controla quién almacena los datos y quién puede acceder a ellos. Por tanto, en esta clase de blockchain los usuarios son conocidos e identificables de manera que se les puedan aplicar roles y permisos. Y por ello, este tipo de blockchain se caracteriza de la administración por parte de una entidad que se encarga de mantener la cadena y asignar roles. De esta forma se asegura cierto nivel de seguridad, privacidad, rendimiento y otras propiedades que las blockchains privadas pueden ofrecer. Para una blockchain privada existen diferentes opciones, y las más comunes son Hyperledger, R3 Corda y Quorum.

El cuarto tipo son las *públicas permisionadas*, la información se mantiene abierta, existe el sistema de roles y permisos donde no todo el mundo puede participar de la misma forma. Sus casos de uso están relacionados con cadenas de suministros, registros financieros del gobierno y declaraciones de renta de corporaciones.

2.2.3 Minado y tipos de consenso

Los algoritmos de consenso son procesos de toma de decisiones para un grupo dentro de un sistema distribuido, donde cada individuo dentro del grupo construye y apoya la decisión que funcione mejor para ellos. Es una forma de resolución donde los miembros deben aceptar la decisión mayoritaria **[10]**, y en el contexto de una blockchain se utilizan para la creación distribuida de los bloques que pertenecen a la cadena de bloques. En estos sistemas, existen usuarios denominados mineros de bloques, que se encargan de actualizar y organizar la blockchain, verificar y agrupar las transacciones en bloques, y además, son partícipes del algoritmo de consenso.

Un ejemplo de los algoritmos de consenso son los sistemas denominados *Proof of Work (PoW)*. En estos sistemas, el minado de bloques consiste en concatenar un número aleatorio (llamado nonce) que se añade al bloque, y buscar que dicho elemento concatenado tenga un hash que empiece por un determinado número de ceros. Cuando se cumpla esa condición, el nuevo bloque (incluyendo dicho número aleatorio) se envía al resto de nodos para que lo comprueben. En caso de que este sea correcto, pasará a ser el siguiente bloque de la blockchain.

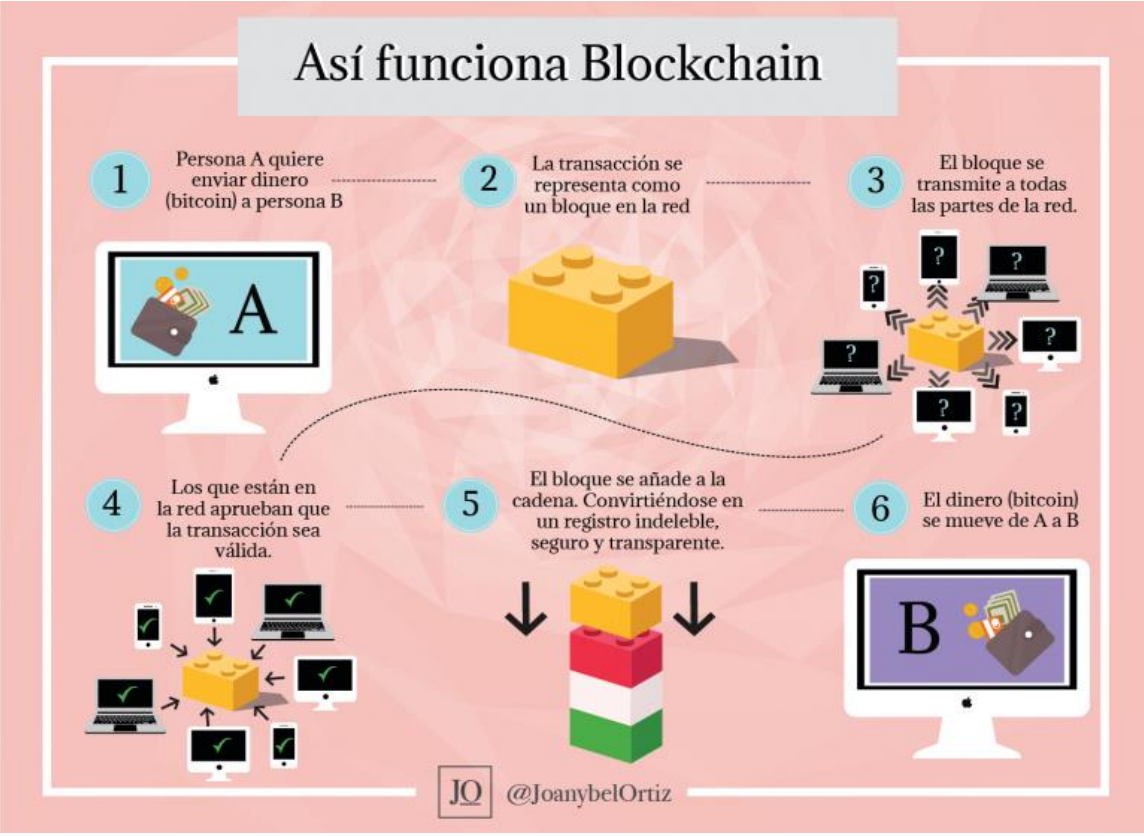

Figura 4 **[11]**. Esquema de funcionamiento de una Blockchain.

El proceso de Proof of Work explicado en el párrafo anterior es extraordinariamente costoso a nivel computacional, ya que se requiere que toda la red realice el cálculo de trillones de hashes para lograr encontrar el número aleatorio buscado. Ahora bien, existen otros algoritmos de consenso diferentes para regular la minería y repartir el trabajo de otras formas, los cuales buscan reducir este alto gasto energético asociado a los algoritmos PoW.

Uno de esos algoritmos de consenso es el *Proof of Stake* (PoS). En este algoritmo cada bloque es validado antes de añadirse a la red con la peculiaridad de que los mineros pueden utilizar sus criptomonedas para participar en el proceso de minería, sin necesidad de resolver un costoso acertijo computacional como en PoW. Por tanto, en este escenario, a mayor riqueza mayor probabilidad de minar un bloque. En este sistema la salud de una blockchain está asegurada por los poseedores de criptomonedas y reciben recompensas por ello. Todos los mineros de la red se eligen aleatoriamente cuando se tiene una cantidad específica de criptomonedas almacenadas. Después, para ser partícipes del proceso de minería se crea una cartera especial donde los mineros apuestan una cantidad mínima requerida o una cantidad superior. Finalmente, cada minero podrá minar un número de bloques proporcional a la cantidad apostada inicialmente para ser recompensado por ello. Sin embargo, este tipo de algoritmos traen un problema, el problema de nada en juego o Nothing-At-Stake que aprovecha que validar bloques no tiene ningún coste. Por tanto, cuando se intenta bifurcar una cadena ya sea de forma accidental porque dos validadores honestos proponen un bloque diferente o por un ataque malicioso, a los mineros les interesa minar en ambas bifurcaciones por dos razones:

- No hay ningún coste o penalización para el minero por minar en ambas.
- Si un minero solo mina en una de las bifurcaciones y esta acaba con una longitud menor que la otra y se abandona, el minero no consigue ningún beneficio del tiempo que invirtió, cosa que no ocurre al minar en ambas.

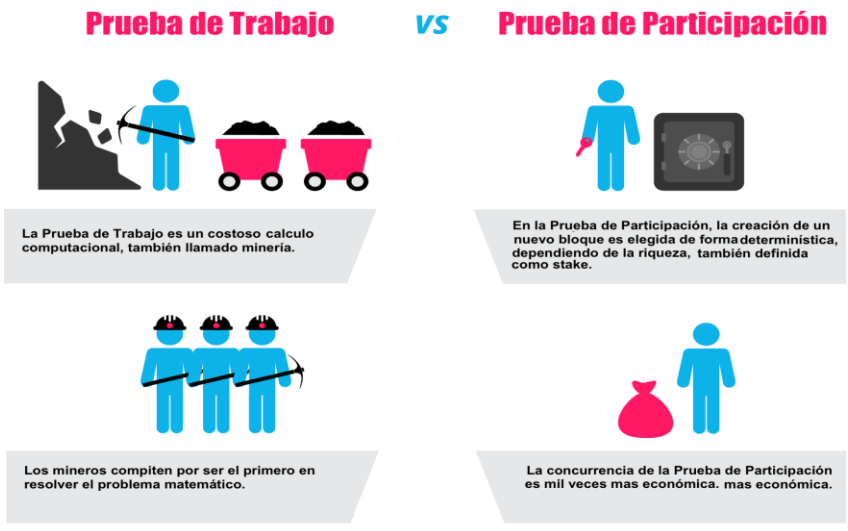

Figura 5 **[12]**. PoW en comparación a PoS.

Por último, un algoritmo de consenso más común entre las blockchains privadas es Proof of Authority (PoA). Este algoritmo aprovecha las identidades reales para permitir las validaciones dentro de una blockchain. Es decir, un número limitado de validadores ponen su identidad real y reputación como garantía de transparencia, y por tanto, cualquier acto que atente contra la fiabilidad y transparencia de la red, recae directamente sobre esa persona o institución **[13]**.

Sin embargo, la existencia del PoA por sí solo no evita los ataques bizantinos. Por ello, dentro de los sistemas PoA, uno de los mecanismos más utilizados para proteger las blockchain privadas es el algoritmo "Practical Byzantine Fault Tolerance" (PBFT) que se centra principalmente en el estado de la máquina. De forma que replica el sistema y elimina el problema de los generales bizantinos. El algoritmo es capaz de asumir el mal funcionamiento de f (faulty) nodos dentro de una red de $3f + 1$ nodos. Para empezar los nodos del sistema se encuentran organizados. De todos ellos, uno es seleccionado como el principal y el resto como respaldo, aunque al final todos funcionan de forma sincronizada y comunicándose entre sí. El nivel de comunicación entre nodos resulta ser alto, específicamente $1 + 3f + 3f(3f - f) + (3f - f + 1)(3f + 1) + 3f - 1$ mensajes **[14]**, para así poder verificar toda la información que se almacena en la red y detectar si algún nodo está comprometido. Por ejemplo, para f = 2 (es decir, 2 potenciales nodos maliciosos en una red de 7 nodos) se acaban enviando 71 mensajes. En cada fase se envían la siguiente cantidad de mensajes (seguir junto a la figura 6):

- En la petición se genera un mensaje un mensaje
- En la fase pre-preparación se generan 3f = 6 mensajes.
- En la preparación se generan 3f(3f-f) = 24 mensajes.
- En la fase de confirmación se generan (3f-f+1)(3f+1)= 35 mensajes.
- En la fase de respuesta se generan 3f-1 = 5 mensajes.

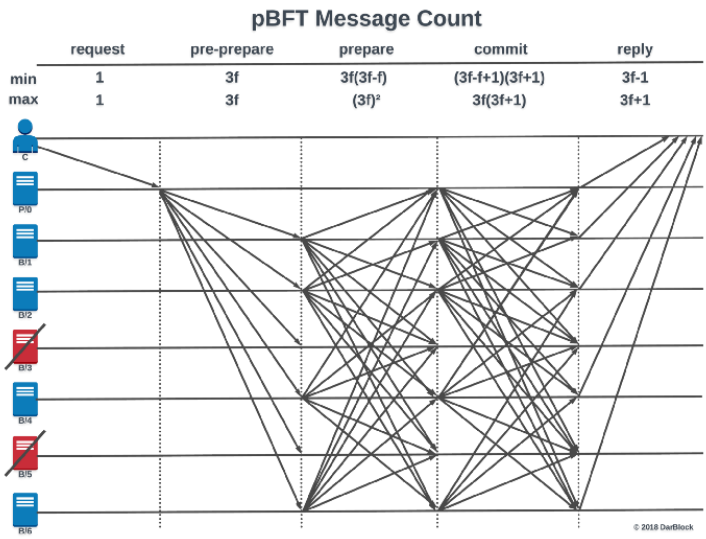

Figura 6 **[14]**. Aplicación de la función para 2 nodos que fallan.

2.2.4 Redes Capa-1 y Capa-2

En el ecosistema de la descentralización, una red de Capa-1 se refiere , por ejemplo, a una blockchain, mientras que los protocolos de Capa-2 son integraciones que se pueden utilizar integradas a las soluciones de Capa-1.

2.2.4.1 Soluciones de escalabilidad Capa-1

En este caso las soluciones de escalabilidad en la capa uno afectan a la capa base de un protocolo blockchain en sí mismo, modificándolo para obtener un ratio de transacciones por segundo mayor o ser capaz acomodar a más usuarios y datos. Un ejemplo de estos cambios son el incremento de información contenido en un bloque, acelerar el ratio de confirmación de los bloques, o el cambio de algoritmo de consenso a uno más eficiente.

Otra actualización fundamental para una blockchain que logre el escalado de red de Capa-1 es la fragmentación o **Sharding**. La fragmentación es un mecanismo adaptado desde las bases de datos distribuidas que se ha convertido en la solución de escalabilidad más popular, a pesar de su naturaleza algo experimental en el sector Blockchain. La fragmentación implica dividir el estado de la blockchain en diferentes conjuntos de datos llamados fragmentos o shards. Mantener el estado de estos fragmentos supone una tarea más manejable que requerir a los nodos mantener el estado completo de la red. Estos fragmentos se ejecutan en paralelo a través de la red permitiendo el trabajo secuencial en numerosas transacciones de forma simultánea, donde cada fragmento proporciona pruebas a la blockchain principal e interactúa con otros fragmentos para compartir información (direcciones, balance, estado…) mediante protocolos de comunicación entre fragmentos. Ethereum 2.0 es un perfil de blockchain que explora el uso de esta técnica junto a Zilliqa, Tezos y Qtum.

2.2.4.2 Soluciones de escalabilidad Capa-2

La Capa-2 es la red o tecnología que opera sobre una blockchain subyacente para mejorar su escalabilidad y eficiencia. Esta categoría de soluciones implica trasladar una porción de la carga transaccional desde la blockchain a una arquitectura de sistema adyacente, que maneje la mayor parte del procesamiento de la red e informe a la blockchain principal para finalizar sus resultados. Al abstraer la mayor parte del procesamiento de datos a un sistema auxiliar, la blockchain se congestiona menos, y en última instancia, aumenta su escalabilidad.

Las diferentes soluciones Capa-2 son:

Blockchain anidadas. Una blockchain anidada es esencialmente una blockchain dentro de otra. La arquitectura de blockchains anidadas típicamente suele involucrar una cadena principal que fija los parámetros para una red más amplia, mientras las ejecuciones se llevan a cabo en una red interconectada de cadenas secundarias. Se pueden construir múltiples niveles de blockchain en una cadena principal, con cada nivel implementando una conexión padre-hijo. La cadena principal delega el trabajo a las cadenas hijas para que procesen la información y la retornen a la cadena padre una vez terminen la tarea. La blockchain principal no participa en las funciones de red de las

cadenas secundarias a menos que sea necesaria la resolución de disputas. La distribución del trabajo bajo este modelo reduce la carga de procesamiento de la cadena principal para mejorar de forma exponencial su escalabilidad. El proyecto OMG Plasma es un ejemplo de solución Capa-2 de blockchains anidadas que se ejecuta sobre Ethereum para procesar transacciones de forma más rápida y barata **[15]**.

Canales de estado. Un canal de estado posibilita la comunicación bidireccional entre una blockchain y un canal de transacciones fuera de la cadena (transacciones off-chain) y mejora la capacidad y velocidad de las transacciones en general. Un canal de estado no necesita ser validado por los nodos de la Capa-1. En lugar de eso, es un recurso adyacente a la red que se cierra mediante el uso de un mecanismo como los contratos inteligentes o la firma múltiple. Cuando una transacción o un lote de transacciones se completa, el estado final del canal y todas las transiciones inherentes al mismo son grabadas en la blockchain. La Lightning Network de Bitcoin y la Raiden Network de Ethereum son ejemplos de canales de estado. Los canales de estado sacrifican cierto grado de descentralización para lograr una mayor escalabilidad.

Sidechains. Una sidechain o cadena lateral es una blockchain adyacente utilizada para procesar grandes lotes de transacciones. Las sidechains suelen utilizar un algoritmo de consenso independiente al de la cadena principal, para optimizar la velocidad de las transacciones y la escalabilidad del sistema. En esta arquitectura la cadena principal cumple el rol de mantener la seguridad del sistema, confirmando los lotes de transacciones grabados en las sidechains y resolviendo conflictos. Las sidechains se diferencian de los canales de estado en que las transacciones que se realizan no son privadas entre sus participantes, estas se graban de forma pública en la cadena. Además, las brechas de seguridad de una sidechain no afectan a la cadena principal ni a otras sidechains. Establecer una sidechain requiere un esfuerzo sustancial ya que la infraestructura se construye desde cero.

Rollups. Las Rollups (o Acumulaciones) son una solución de “capa 2” que permiten que las DApps agrupen transacciones en una única transacción fuera de la cadena principal, de forma que luego se genere una prueba criptográfica y se envíe a la cadena principal. Dentro de este grupo destacan ZK-Rollups y Optimistic Rollups. Por un lado, las **ZK-Rollups** (Zero Knowledge Rollups) agrupan cientos de transacciones fuera de la cadena y crea una prueba criptográfica, conocida como SNARK (succinct non-interactive argument of knowledge) que se publica en la capa 1. El contrato inteligente que lo implementa mantiene la traza de estas pruebas de validación y su estado es únicamente modificable a través de estas, por tanto, validar un bloque es más rápido y barato debido a que en lugar de cargar todos los datos de varias transacciones solo se utiliza una validación que puede probar un gran número de estas. Por otro lado están las **Optimistic Rollups** (Rollups Optimistas) que mejoran la escalabilidad debido a que una vez enviadas las transacciones en lotes, se propone el cambio de estado a la red principal, y en caso de que alguien note o sospeche que una transacción es fraudulenta, el rollup ejecuta lo que se conoce como prueba de fraude y si demuestra que hay una transacción incorrecta, el contrato inteligente vuelve al estado anterior.

2.2.5 Casos de uso

2.2.5.1 Smart Contract

Un Smart Contract o Contrato Inteligente es un programa capaz de ejecutarse y hacerse cumplir por sí mismo o a partir de eventos, de manera autónoma y automática, sin intermediarios ni mediadores. Al ser escrito como código informático, no existe la interpretación por las diferentes partes que participan en el contrato, que puede acompañar a los lenguajes naturales que hablamos, evitando que alguno de los firmantes no cumpla su parte del acuerdo debido a una interpretación subjetiva. Un smart contract puede ser creado y ejecutado por personas jurídicas o físicas, pero además, por máquinas o programas que funcionan de forma autónoma. Los Smart Contracts constan de validez sin depender de autoridades debido a que es un código visible por todos y este no se puede cambiar sobre la tecnología Blockchain. Por ello se le atribuye un carácter inmutable, descentralizado y transparente.

2.2.5.2 Cadenas de suministro

La tecnología Blockchain es aplicable también a la gestión de cadenas de suministro donde aún se siguen utilizando técnicas antiguas que llevan a la falta de transparencia, dejándolas vulnerables a sujetos con intenciones maliciosas que modifiquen dicha información. Dicho problema se puede resolver aplicando la tecnología Blockchain, que permite verificar en cuanto queramos de forma rápida y confiable la veracidad de la información, tras digitalizar los activos de una cadena de suministros, de forma que cada activo lleve un número de serie o de identificación único asociado a él. De esta forma, a través de la cadena de suministro se puede monitorear y rastrear cualquier activo. Un ejemplo de este tipo sería Food Trust de IBM y Walmart **[16]**.

2.2.5.3 Identidad digital

Un gran problema de la actualidad que se ha agravado debido a Internet es el robo de identidad. Hacerse pasar por otra persona se ha vuelto más sencillo. A su vez, la gestión de identidad se ha ido complicando a lo largo de los años, se necesitan una gran cantidad de documentos, tales como el carnet de conducir, DNI, tarjetas de seguro médico, universidad o cualquier otro documento de identidad asociado a una entidad con la que se relaciona cada individuo. Blockchain permite tener una identidad digital asociada a cada individuo, donde cada persona sólo tiene una identidad en toda la red. Ofreciendo al mismo tiempo varios beneficios como la carencia de necesidad de llevar documentos de identificación y la protección del individuo frente al robo de identidad. Proyectos destacados en este ámbito son Civic **[17]** y el framework Hyperledger Indy **[18]**.

2.2.5.4 Tokenización de activos

La tokenización de activos es de vital importancia respecto a la protección de los activos del mundo real. La tokenización de activos consiste en la representación de un derecho sobre un activo en un registro distribuido (blockchain en este caso) privado a efectos legales y público o semipúblico a efectos tecnológicos. Dicha representación en anotaciones contables unitarias atienden al nombre de tokens. La tokenización de activos hace que el manejo de activos sea más práctico y eficiente acelerando procesos como los de compra y venta, e ignorando por completo las formas tradicionales de

resolver activos, permitiendo ahorrar tiempo al no tener que pasar por largos procesos para ello, convirtiéndolo en uno de los mejores casos uso de Blockchain, dada su seguridad. Polymath **[19]** y Harbor **[20 y 21]** son proyectos que trabajan en este caso de uso de Blockchain.

2.2.5.5 Mercado de la energía

El mercado energético es un mercado cerrado en general, controlado por grandes compañías en su mayoría. Cuando se desea obtener energía, se ha de solicitar y esperar una instalación que por diversos motivos puede ser negada tanto a individuos como a empresas. Pero con Blockchain dicho mercado puede experimentar una revolución en la gestión de la energía. Con Blockchain la energía se convierte en otro activo, y por tanto registrar, liquidar y comerciar con energía se vuelve un proceso más sencillo mediante el uso de la tecnología Smart Contract. Cualquiera que participe en dicha red puede aprovechar dicho mercado abierto, impactando así sobre los precios de la energía que pueden encontrarse de antemano regulados y fijados (para bien o para mal), a pasar a verse regulados por la oferta y la demanda. Grid+ **[22]** es un ejemplo funcional de esta aplicación de Blockchain.

2.2.5.6 Cuidado de la salud

Uno de los casos de uso más críticos de la tecnología Blockchain es en el sector de la salud. Actualmente el sector sanitario está sufriendo del enfoque centralizado con el que se diseñaron sus sistemas, que les lleva a tratar la información de los pacientes de una forma ineficiente. Por ello, los pacientes se ven obligados a llevar sus datos médicos de un médico a otro, y en el caso de acudir a otros centros el problema no hace más que engrandecerse por motivos como la dificultad que puede suponer la recuperación de la información del paciente, debido a la diferencia de formatos en los que se guarde la información. Blockchain puede simplificar estos problemas al proporcionar un enfoque descentralizado. Con dicho enfoque, los interesados podrán acceder a los datos de los pacientes según los permisos que posean, además de permanecer seguros. Por último, relativo a la salud y las cadenas de suministro, el uso de Blockchain es capaz de salvar vidas al aplicarse a las cadenas de suministro de empresas farmacéuticas eliminando completamente la falsificación de medicamentos. SimplyVital Health **[23]** y MediBloc **[24]** son ejemplos de proyectos Blockchain relacionados con el cuidado de la salud.

2.2.5.7 Criptomoneda

La criptomoneda es el caso más común de la tecnología Blockchain, resuelve una gran cantidad de problemas según su implementación. Por ejemplo, Bitcoin es una moneda para transacciones. Ethereum además de ser similar en este aspecto ofrece los ya mencionados Smart Contracts y DApps, aplicaciones descentralizadas que se ejecutan sobre la tecnología Blockchain. La criptomoneda permite enviar dinero a nivel mundial sin necesidad de intermediarios ni comisiones (a excepción del gas), además de ser más rápido que los métodos actuales que pueden llegar a tardar entre 1 y 5 días. Además, puede ser una opción para aquellos que no tengan acceso a servicios bancarios.

2.3 IoT

2.3.1 Características

IoT describe un sistema donde objetos físicos con sensores incorporados a ellos, están conectados a la red a través de cables o de forma inalámbrica. Este conjunto de objetos posee una inteligencia e identidad propia, que les permite formar parte e interactuar de forma inasistida en Internet con otros dispositivos o usuarios **[25 y 26]**. Una plataforma IoT se caracteriza entonces por las siguientes 6 características. Estos son:

'Cosas'. Por 'cosas' nos referimos a todo aquel dispositivo que puede estar conectado y está diseñado para ello. Esta definición incluye tanto a sensores y electrodomésticos como a gafas o relojes inteligentes.

Inteligencia. Las aplicaciones IoT necesitan ser capaces de responder de forma inteligente en situaciones particulares según la información que obtienen de su entorno para llevar a cabo tareas específicas. Dicha inteligencia sólo comprende las interacciones entre dispositivos.

Interconectividad. En la IoT todo dispositivo debe ser capaz de conectarse con la información global y la infraestructura de comunicación. La conexión entre estos dispositivos es crucial ya que estas simples interacciones contribuyen a la inteligencia colectiva de la red IoT facilitando la accesibilidad y la compatibilidad de estos.

Sistemas de almacenamiento de datos. La información generada por las 'cosas' se suele almacenar ya sea en bases de datos locales, clouds o edge, debido a las ventajas que ofrecen y sirven para facilitar el análisis de información, como el acceso inmediato a los datos a través de consultas, su fácil mantenimiento, la capacidad que ofrecen de centralizar la información y sobre todo por la capacidad de poder ingresar información 'ilimitada' en el caso de las bases de datos virtuales.

Análisis y visualización. Tras la obtención de información del entorno esta es tratada mediante una serie de complejos análisis de la agrupación de datos básicos y de aprendizaje automático. De estos análisis se obtienen estadísticas y tablas que pueden ser representadas a través de los diferentes tipos de gráficos estadísticos.

Cambios dinámicos. El estado de los dispositivos cambia de forma dinámica. Por ejemplo, pueden encontrarse en estado suspenso, conectado, arrancando o desconectado además de las posibles localizaciones y velocidades a las que se encuentren. Y adicionalmente, el número de dispositivos conectados entre sí también suele variar.

2.3.2 Tecnologías de comunicación

Las redes de comunicaciones IoT se caracterizan por tener bajas velocidades de datos, baja frecuencia de transmisión, movilidad y servicios de localización, conexiones

bidireccionales seguras, bajo consumo de energía y largo alcance de comunicación **[27]**. Los estándares más utilizados se definen a continuación.

2.3.2.1 M2M

Las redes de comunicación M2M (Machine to Machine) han sido las principales utilizadas por las compañías de telecomunicaciones. Estas son aquellas vinculadas por tarjeta sim que se caracterizan por el pago por Byte transmitido, como actualmente en el 3G y 4G. Sin embargo, para proyectos IoT ambiciosos en los que se conectan miles de dispositivos que envíen muchos datos (en cantidades pequeñas), esta tecnología resulta inadecuada por su difícil escalabilidad, cobertura dependiente de un operador y el coste por dato transmitido. Además, debido al coste energético en el envío de datos, los dispositivos alimentados por baterías que necesitan de una larga durabilidad (varios años) no lo pueden asumir.

2.3.2.2 Sigfox

Sigfox es una red LPWAN (Low Power Wide Area Networks) específica para IoT. Está pensada para comunicaciones de baja velocidad reduciendo los precios y el consumo de energía de los dispositivos conectados. Se basa en una infraestructura de antenas y estaciones independientes a cualquier otra red cuyo alcance se encuentra entre el alcance del Wi-Fi y la comunicación móvil, muy útil para cuando el Wifi queda corto y la comunicación móvil sea cara (además de resolver el problema inherente a la consumición de energía). Al enviar la información por canales de espectro muy estrecho (Ultra Narrow Band), estas redes son capaces de lograr el enlace a larga distancia (desde 5km en áreas urbanas hasta 25km en campo abierto) entre transmisor y receptor, a velocidades de 10 a 10.000 bits por segundo.

2.3.2.3 NarrowBand IoT

La NarrowBand IoT (NB IoT) es otra red LPWAN por la que están apostando las operadoras de telecomunicaciones españolas **[28]**. Al haber sido diseñada para IoT, esta permite un gran número de dispositivos conectados (soportando hasta 100.000 conexiones por móvil) además de otras características, como un mayor alcance en interiores y subterráneos, y una mayor seguridad garantizada por doble autenticación y una interfaz fuertemente encriptada. Su factor diferencial es que su espectro de funcionamiento entra dentro del 4G.

2.3.2.4 Bluetooth de baja energía

El Bluetooth de baja energía (BLE) es otra tecnología inalámbrica aplicable a IoT. Es utilizado para conectar pequeños dispositivos diseñados para usar Bluetooth para mandar paquetes de datos reducidos. Se utiliza en aquellos dispositivos para dar servicios de localización y señalización que cuentan con poca batería, permitiendo que estas lleguen a durar meses. Sin duda es clave para el desarrollo de proyectos IoT para electrónica de consumo, como wearables y electrodomésticos que no necesiten de mucho alcance. A pesar de su corto alcance, que lo convierte de poca utilidad para redes de sensores de largo alcance, su gran escalabilidad lo convierte en propio para entornos industriales y en general dispositivos de conexión PAN (Personal Area Network).

2.3.2.5 ZigBee

ZigBee es una tecnología inalámbrica centrada en aplicaciones domóticas e industriales que cumple con las especificaciones de tasas de envío de datos bajas con un alcance de cobertura cercano a los 100 metros. En caso de que los dispositivos a comunicar se encuentren más alejados se descarta el uso de esta tecnología. Sin embargo, esta ofrece un bajo consumo, una seguridad robusta fundamentada en la criptografía de clave simétrica y una alta escalabilidad.

2.3.3 Dispositivos

Entre los diferentes dispositivos de la IoT **[29]** se pueden diferenciar los *System-On-Chip (SOC)* un sistema empotrado o embebido de 8 bits, es un tipo de hardware en el que no suele instalarse ningún sistema operativo. Uno de los ejemplos más conocidos de este tipo de dispositivos es Arduino Uno. En el siguiente nivel de dispositivo se encuentran los que tienen una *arquitectura de 32 bits* como Atheros o ARM. Estos dispositivos suelen basarse en plataformas de Linux Embedded y otros sistemas operativos empotrados. Arduino Zero o Arduino Yun son ejemplos de este tipo de dispositivos. Y por último, los dispositivos IoT más capaces, que son los sistemas completos de 32 y 64 bits, los *Single-Board-Computer (SBC)*. Son capaces de ejecutar sistemas operativos variados como Linux y Android. Pueden actuar como puentes para dispositivos pequeños. En muchos casos, estos son smartphones o cualquier tipo de dispositivo basado en tecnologías móviles. Por ejemplo: un wearable que se conecta vía Bluetooth a un Smartphone o a una Raspberry Pi, es típicamente un puente para conectarse a Internet.

Otro tipo de dispositivo son los *Gateways IoT* puede ser hardware físico o software que sirve como punto de conexión entre la nube y los controladores, sensores y dispositivos inteligentes. Todos los datos que se dirigen o vienen de la nube pasan por el gateway que también recibe el nombre de pasarela inteligente.

Además en esta lista se pueden añadir los *microcontroladores industriales o PLCs*, dispositivos capaces de utilizar la instalación eléctrica de una casa para transmitir la conexión a internet; softPLCs, un dispositivo en el que se instala software capaz de emular la funcionalidad de un PLC; o cualquier dispositivo que se pueda conectar a internet y obtener datos, por poner otro ejemplo, los wearables **[30 y 31]**.

Y por último, los *sensores*, es hardware que tiene alguna capacidad de procesamiento, de captación de datos sensoriales y de comunicar la información a otros nodos conectados en la red. Estos dispositivos pueden tener conectados a ellos otro tipo de hardware como actuadores, periféricos y transceptores.

2.3.4 Casos de uso

2.3.4.1 IOT Industrial (IIoT)

La aplicación de IoT a la industria manufacturera se llama IIoT (Industrial IoT). Los dispositivos IIoT van desde los ya mencionados sensores ambientales hasta complejos

robots industriales. El IIoT trata de revolucionar la fabricación de productos al permitir adquirir y estudiar los datos sobre la maquinaria. Varias empresas innovadoras han comenzado a implementar el IIoT aprovechando los dispositivos inteligentes conectados en sus fábricas. IIoT incorpora tecnologías de Aprendizaje Automático y Big Data para analizar los datos recolectados en sensores, y las tecnologías de comunicación y automatización máquina a máquina que han existido en entornos industriales durante años. Así, toda la información es recolectada para que los líderes empresariales puedan usar los datos de IIoT para tomar mejores decisiones al obtener una visión completa y precisa de cómo funciona su empresa. Ejemplos de casos de uso en IIoT industrial son:

Fabricación industrial. Un ejemplo actual es Airbus, una empresa fabricante de aviones comerciales que ha integrado sensores tanto en máquinas y herramientas del taller como en empleados mediante gafas inteligentes, para reducir errores y mejorar la seguridad y productividad en el lugar de trabajo **[32]**.

Gestión de mantenimiento. ABB es una empresa robótica que utiliza sensores conectados para monitorear las necesidades de mantenimiento de los robots, y así solicitar reparaciones antes de que se rompan las piezas **[33]**.

Hostelería. En la hostelería ya se están viendo como algunos restaurantes tienen dispositivos que avisan cuando la comida está preparada para ir a recogerla, por ejemplo, la cadena de restaurantes “100 Montaditos”. Desde el lado de la gestión de alimentos, los almacenes utilizan dicha tecnología para controlar la caducidad de alimentos, necesidades, pedidos automáticos y gestión de aparatos como cámaras frigoríficas, cocinas, hornos, etc… Una de las empresas que ofrecen este tipo de aplicación IoT es Powerhouse Dynamics **[34]**.

Agricultura y ganadería. Aplicada a la agricultura, la tecnología IoT se utiliza sobre todo para hacer seguimiento de magnitudes como la temperatura, humedad, luminosidad y demás factores que pueden influir en la producción, permitiendo a agricultores predecir y cuantificar cada cosecha antes de recogerla. Con respecto a la ganadería, el seguimiento biométrico de los animales y su geolocalización es un factor a tener en cuenta. En este sector, la empresa Infiswift ofrece servicios especializados **[35]**.

Flotas de vehículos para logística. Uno de los sectores donde más influencia tienen las aplicaciones IoT es la logística. Dentro de este ámbito el control de paquetes (envíos de productos, no confundir en este caso con paquetes de red), la gestión de vehículos y evitar robos son propios ejemplos de dicha aplicación. La empresa española Libelium, es una de las más avanzadas en este y en otros sectores **[36]**.

2.3.4.2 Aplicaciones del IoT para el uso doméstico

Ejemplos de casos de uso en IoT para uso doméstico son:

Domótica. La domótica es uno de los grandes retos de esta tecnología. Debido a la dificultad en las infraestructuras y la incapacidad de comunicar aparatos de diferentes marcas con facilidad, aún no ha conseguido despegar plenamente. Aún existen muchos

caminos de desarrollo de este ámbito y uno de ellos es piscinas inteligentes, que envían toda la información relevante para controlar la calidad del agua (Ph y temperaturas entre otros). Un proyecto que implementa esta idea es Blue de Riiot **[37]**. Otros productos de las grandes marcas que hacen un uso intensivo del IoT son Amazon Echo, Google Home y Apple Homekit, que para minimizar la interfaz con los humanos utilizan la voz.

Salud. La idea principal es permitir a los médicos poder hacer el seguimiento de las condiciones de sus pacientes tanto fuera como dentro del hospital en tiempo real, mediante el uso de wearables o sensores conectados a los pacientes. A través de la medición y las alertas automatizadas de sus signos vitales, la tecnología IoT ayuda a la prevención de eventos mortales y el control de asistencia en pacientes de alto riesgo. Otro posible avance en dicho campo consiste en la creación de camas de hospital inteligentes, equipadas con sensores que observen los signos vitales.

2.3.4.3 Aplicaciones del IoT en ciudades inteligentes

Una Smart City es aquella ciudad que hace uso de la tecnologías de la información y la comunicación (TIC) con el objetivo de crear mejores infraestructuras para los ciudadanos **[38]**. Actualmente es donde más se nota la introducción de aplicaciones IoT. Se pretende resolver problemas que en la actualidad están afectando a todas las ciudades del mundo. Principalmente se aplica en la gestión de suministros, la calidad ambiental y el tráfico. Ejemplos de casos de uso de IoT en ciudades inteligentes son:

Gestión de suministros. En algunas ciudades, por ejemplo, se da el caso en el que los aspersores de riego comienzan a funcionar en los parques los días de lluvia. IoT puede solucionar dicho problema al permitir controlar la gestión de suministros como el agua. Otra aplicación posible para el suministro de agua es la acumulación de datos sobre el uso de esta misma para que los distribuidores no solo puedan comprender mejor el consumo por parte de los consumidores, que también tendrían acceso a dicha información para su propio beneficio, sino además para reportar averías. Por ejemplo, la empresa Gestagua ha desarrollado un nuevo sistema de gestión para contadores inteligentes de agua que se está probando en varios municipios **[39]**.

Dentro de la gestión de suministros destacan también los **Smart Grids** o redes eléctricas inteligentes, que aplican el uso de medidores de energía equipados con sensores en diferentes puntos estratégicos, que comprenden desde las plantas de producción hasta los puntos de distribución, permitiendo un mejor control de la red eléctrica, reporte de fallos y reparación de los mismos.

Gestión ambiental. La contaminación es uno de los mayores problemas de las ciudades a niveles globales. IoT ha permitido que recopilar la información sobre la calidad del aire y agua sea más sencillo. La empresa española Libelium también ofrece un sistema capaz de hacer dichos análisis **[40]**.

Dentro de la gestión ambiental también destaca la recogida de residuos de forma inteligente. En España, Grupo Defesa consiguió avanzar hacia procesos más eficientes en la recuperación y tratamiento del papel. Para conseguirlo, introdujeron sistemas

inteligentes e IoT en contenedores, de manera que son capaces de controlar su contenido, informar al sistema central en el momento de recogida más adecuado, comprimir el papel para obtener mayor capacidad de almacenamiento y alertar en el caso de identificar personas en su interior o en el caso de robo de papel **[41]**.

Gestión del tráfico. Está muy vinculado a la contaminación. Una gestión eficiente del tráfico puede evitar niveles de contaminación altos. Google ya ofrece Maps, donde en todo momento informa de la situación de tráfico, llegando a evitar entrar en atascos, por ejemplo, tras un accidente en la autovía. A través de los datos anónimos como la posición y velocidad recogidos a través de los móviles son capaces de mostrar dicha información.

2.3.5 Ventajas

Entre las numerosas ventajas dentro de la IoT las principales son:

Dispositivos interconectados. Esta característica permite recolectar grandes cantidades de información, que combinadas con las estadísticas y otros métodos de análisis, permiten tomar mejores decisiones. Decisiones capaces de afectar a los productos de una empresa para optimizar su vida, de reducir costes o de crear nuevos modelos de trabajo en entornos laborales.

Mejorar el servicio al consumidor y aumentar la productividad. Ser capaz de unificar la experiencia de un producto o servicio con los dispositivos IoT es un punto de venta clave para una empresa moderna. Muchas empresas invierten en aplicaciones móviles para dar un mejor servicio a sus clientes. Por ejemplo, en un gimnasio estas aplicaciones sirven para evaluar monitores, recibir rutinas diarias en base a los objetivos de cada individuo y recibir información del progreso (peso, masa muscular, etc.), al mismo tiempo que esta información es recolectada. Con la información recolectada, las empresas pueden anticiparse a las necesidades de sus consumidores y de esta forma conocer mejor a sus clientes, para conocer sus expectativas, entender su comportamiento y automatizar procesos. Con la automatización de procesos las empresas pueden dedicar más tiempo y recursos a aquellas tareas que son importantes y se alinean con los objetivos estratégicos y delegar las restantes a la inteligencia artificial.

La monitorización. A través de dispositivos IoT se puede mejorar la seguridad en entornos de trabajo de alto riesgo. Trabajadores de sectores como la minería o construcción pueden recibir avisos en caso de derrumbes, caídas u agotamiento. En el entorno sanitario, las constantes vitales de pacientes, tanto fuera como dentro de hospitales, pueden ser monitorizadas, e incluso de manera automática notificar cuando estas fallan. En el entorno de las pequeñas empresas, el uso de cámaras de vigilancia, cerraduras inteligentes y otro tipo de sensores pueden prevenir los robos y el espionaje industrial.

2.3.6 Desventajas

IoT no solo genera una serie de beneficios, también viene acompañado de una serie de retos que deben ser superados:

Seguridad. Dado que los sistemas IoT se encuentran interconectados a través de Internet, también son vulnerables a aquellos ciberataques que pueden afectar a consumidores, comercios, industrias y sistemas gubernamentales. Un ejemplo de ello es el ataque masivo a dispositivos IoT vulnerables que tuvo lugar en 2016, la botnet "Mirai" tomó el control de cientos de miles de dispositivos para posteriormente realizar un ataque de denegación masivo **[42]**. Muchos dispositivos IoT son fabricados por software y hardware similar, y frecuentemente no son diseñados teniendo en cuenta la ciberseguridad. Un ejemplo de ello son aquellos dispositivos que no poseen un sistema de actualizaciones, y por tanto, en caso de error no podrán ser parcheados remotamente de forma efectiva. Otro problema similar es que el fabricante deje el negocio, y por tanto, deje de dar soporte. Esto puede tener consecuencias muy graves que pueden resultar en lesiones o incluso la muerte, ya que en sistemas como vehículos, donde la centralita puede tener el acceso a las distintas funcionalidades del coche como los frenos, giro de volante y acelerador; o máquinas de soporte vital, en hospitales; podrían ser hackeados.

Privacidad. Debido al incremento de conectividad entre dispositivos y de recolección de datos, proteger la privacidad del consumidor se vuelve más complicado a medida que las tecnologías IoT se hacen más populares. Como se ha mencionado en el párrafo anterior, se puede perder el control de los dispositivos. En el caso de los móviles, estos suelen portar una gran cantidad de información personal como cuentas de banco, emails e incluso el acceso a accesorios de casa (una webcam que vigile la entrada, el control de aire acondicionados, luces, puertas, etc). Además toda la información recolectada de los clientes por parte de las compañías, no solo es utilizada para mejorar la experiencia del cliente o venderle productos que le puedan interesar, sino que en numerosas ocasiones esta información ha sido vendida a otras empresas.

Financiación. Se requiere de una gran inversión para poner en marcha proyectos IoT de gran envergadura, como puede ser una Smart City. No todas las ciudades pueden asumir ese coste, y por tanto, IoT es un fenómeno que puede aumentar la brecha digital, es decir, qué usuarios y cuáles no pueden acceder a dicha tecnología. Este problema se hace notable cuando se comparan distintos países, pero se acentúa más cuando se compara el medio urbano con el rural, entre varios motivos debido a las limitaciones de la conexión a Internet que pueden surgir.

Incompatibilidad entre dispositivos. No todos los dispositivos IoT siguen estándares y por este motivo se pueden dar casos en los que dispositivos diseñados para el mismo objetivo no puedan trabajar conjuntamente.

3

Tecnologías DLT diseñadas para IoT

La tecnología Blockchain ha ido demostrando que es una tecnología de gran valor, pero el hecho de que realizar transacciones tenga un coste económico para el que la realiza, para así poder sustentar el sistema de bloques y minado, supone una gran desventaja en el mercado de las microtransacciones, que a su vez va tomando mayor importancia con el desarrollo de la IoT, ya que la cuota para minar el bloque podría superar el del mismo micropago que se desee realizar. Además, la capacidad de las blockchains más tradicionales, como Bitcoin (7 transacciones por segundo de media), no son lo suficientemente eficientes como para poder almacenar miles de datos en un tiempo aceptable. Ante estos dilemas totalmente opuestos a las características de los sistemas IoT, han ido surgiendo distintas DLTs diseñadas para enmendarlos. Dentro de esta sección se introducirán y explicarán las características de varias de ellas para su posterior análisis. Respecto a Ethereum, aunque esta plataforma no se publicite como específica para IoT, se ha añadido a este estudio como plataforma generalista de referencia, de forma que sea posible analizar cuál es la diferenciación real con el resto de plataformas específicas de la IoT.

3.1 Ethereum

Ethereum es una de las blockchains públicas capaz de soportar contratos inteligentes más populares a día de hoy, y más que un libro de registros distribuidos es una máquina de estados distribuida.

La red Ethereum actúa como si existiese un único ordenador, llamado Ethereum Virtual Machine o EVM, cuyo estado es guardado y acordado por cada nodo de la red. El balance

de las cuentas se almacena en grandes tablas que conforman parte del estado del EVM. Adicionalmente, todos sus participantes pueden emitir una petición a esta computadora virtual para realizar un cómputo arbitrario, en una petición que atiende al nombre de solicitud transaccional. Por otro lado, el término transacción en Ethereum es el término formal que se refiere a una solicitud transaccional cumplida y al cambio de estado asociado a ella del EVM, que pueden ser enviar una cantidad de criptomonedas, publicar un contrato inteligente o hacer una ejecución de ellos. Cuando las peticiones de este tipo se emiten, el resto de participantes de la red verifican, validan y llevan a cabo dicha computación. Causando un cambio de estado del EVM, que a su vez es confirmado y propagado a través de toda la red.

El algoritmo de consenso utilizado en Ethereum actualmente es Proof of Work, y a cambio del minado de bloques los mineros, mediante el uso de la capacidad de computación de sus equipos, reciben Ether. El Ether (ETH) es la criptomoneda nativa de Ethereum, y la cantidad que se paga a los mineros aumenta en función de la cantidad de cómputo realizada por el minero.

3.1.1 Ethereum 2.0

Actualmente Ethereum se encuentra en un proceso de transición hacia el algoritmo de consenso Proof of Stake, junto a la implementación de otros mecanismos para mejorar su eficiencia. Por tanto, es posible que esta información quede desactualizada con la salida completa de Ethereum 2.0. Actualmente se encuentra en una primera versión lanzada el 1 de diciembre de 2020 conocida como “Phase 0”, y se estima que la versión completa no se lanzará hasta 2022-2023.

Ethereum 2.0 se refiere a un conjunto de actualizaciones interconectadas que harán a Ethereum más escalable, más seguro y más sostenible. Estas actualizaciones están siendo diseñadas por múltiples equipos de todo el ecosistema de Ethereum y están segmentadas en 3 grupos:

La **Cadena de Baliza o Beacon Chain**, conocida como "Fase 0" en los mapas técnicos de la ruta, coordinará la expansión de la red a fragmentos (las Shard Chains) y apostadores (stakers). Esta cadena no maneja cuentas ni contratos inteligentes. La Beacon Chain introduce PoS en Ethereum, y se espera que esto ayude a hacer a Ethereum más seguro a largo plazo. Cuanta más gente participe en la red, más descentralizada y resistente a ataques será. Con el tiempo, la Cadena de Baliza también será responsable de asignar aleatoriamente participantes para validar las cadenas fragmentadas. Esta será la clave para dificultar que los apostadores conspiren y se hagan con el control de una Shard Chain. La Cadena Baliza es un primer paso importante en la introducción de cadenas fragmentadas, dado que estas requieren de las apuestas (stake) para trabajar de forma segura.

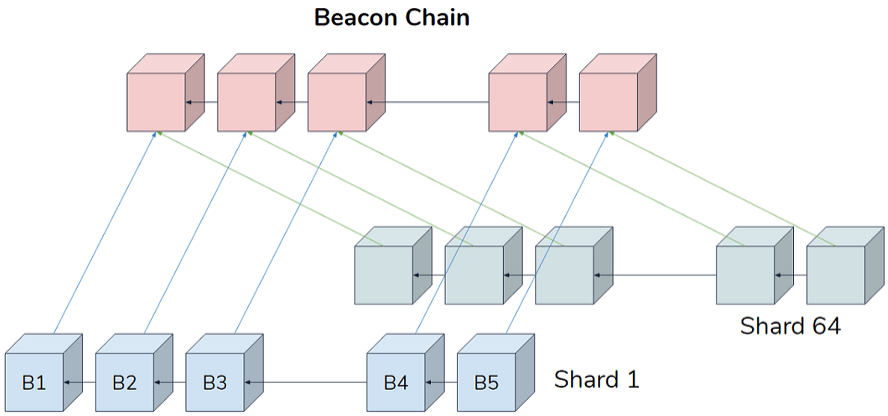

Figura 7 **[43]**. Modelo Ethereum 2.0.

Las **cadenas fragmentadas o Shard Chains**, serán implementadas a lo largo de la fase 1 y 2, y extenderán la red a 64 blockchains independientes que se sincronizarán con la Beacon Chain, aumentando la capacidad de la red, reduciendo su congestión e incrementando las transacciones por segundo. Este cambio además supondrá que dispositivos con menor capacidad de almacenamiento también sean capaces de ejecutar nodos completos, debido a que sólo necesitarán almacenar o ejecutar datos para la Shard Chain que validen, volviendo la red más segura debido al incremento de participación en la red con nuevos dispositivos menos costosos. *En primera instancia* pretenden que cuando las primeras Shard Chains se lancen estas no manejen ni transacciones ni smart contracts, únicamente que provean información extra a la red. Aunque cuando se combinen con las **Rollups** se conseguirá una mejora sustancial en el ratio de transacciones por segundo (explicado en la sección 2.2.4.2). *En segunda instancia* las Shards Chains serán modificadas para que funcionen de una forma parecida a la red actual de Ethereum, pudiendo almacenar y ejecutar contratos inteligentes y operar con las cuentas.

La mezcla, The Merge también conocida como la “Fase 1.5” es la actualización en la que se acoplará la red principal actual de Ethereum con la Beacon Chain de Ethereum 2.0, marcando la transición completa a Proof of Stake. Donde la red principal se convertirá en un fragmento (shard) más que utilice PoS.

3.2 IOTA Tangle

IOTA Tangle es una tecnología de registro distribuido que de manera inmutable almacena datos de forma que su información sea confiable y segura.

IOTA no es una red descentralizada, debido a la existencia del **coordinador**. La figura de este coordinador tampoco es una centralización total, no puede impedir la participación de cualquiera en la red pero sí decidir si una transacción es válida o no, o incluso parar el funcionamiento de la red. Esto va en contra de las bases de la tecnología Blockchain (no permisionada), donde su funcionamiento tiene que estar disponible siempre y ser totalmente fiable. Aunque a su vez puede suponer una ventaja, IOTA ya tuvo problemas

con su propio algoritmo hash, se consiguió forzar colisiones adrede y tuvo que desconectar el sistema durante varios días **[4]**.

IOTA ofrece una solución diferente a la tecnología Blockchain basada en grafos acíclicos dirigidos, denominada Tangle. Las transacciones conforman el conjunto de nodos en el que forman el registro de transacciones. Para añadir un nuevo nodo o transacción, se eligen otros dos nodos no validados (o tips) para ser validados y se deja constancia con dos aristas dirigidas. Al generar el Tangle existe una primera transacción que es validada por el resto de la red, ya sea de forma directa o indirecta. En esta primera transacción todas las criptomonedas que van a existir en toda la red son creadas, de forma que no existe el minado, en el sentido de que los mineros no reciben una recompensa económica, sino que para poder añadir las transacciones, cada nodo de la red debe resolver por su cuenta un **PoW de dificultad reducida**.

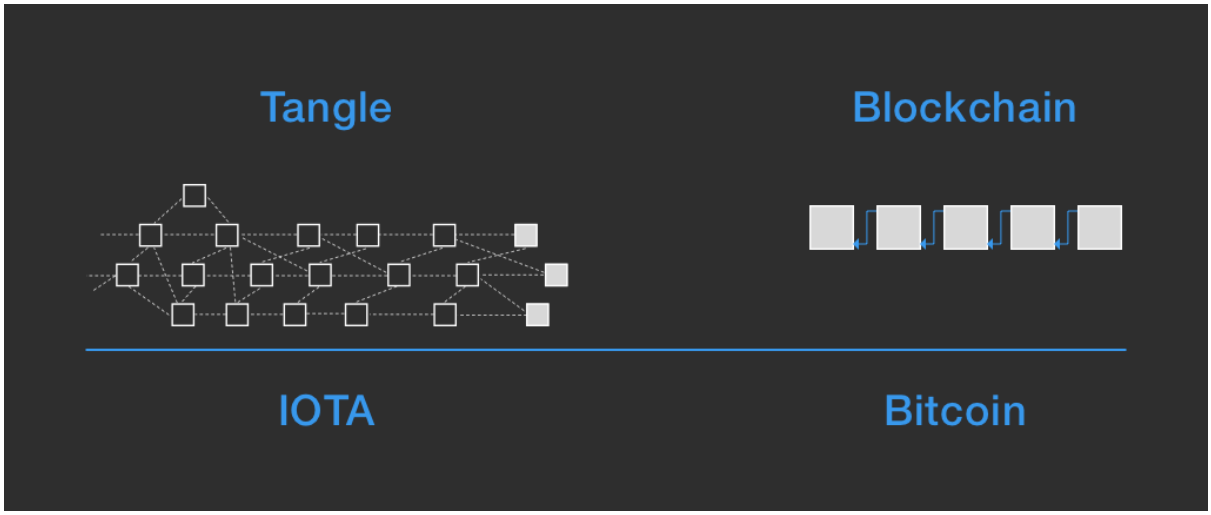

Figura 8 **[44]**. Comparación gráfica entre el diseño del Tangle y Blockchain.

La idea que persigue esta disposición es que para realizar una transacción, el nodo de la red que la envía tendrá que hacer el trabajo de validar otras, y por tanto contribuir de esta forma a la seguridad de la red, asumiendo que el nodo comprueba que no haya conflictos con la transacción que está verificando. A medida que una transacción va obteniendo mayor aprobación (ya sea directa o indirecta) consigue un nivel mayor de confianza. Además de esta manera se dificulta al sistema aceptar transacciones de doble gasto. Por ello esta tecnología a priori resulta más escalable que otras blockchains como Bitcoin, donde el consenso en el que se decide cual va a ser el siguiente bloque hace cuello de botella, y donde Bitcoin es capaz de conseguir almacenar de 3-4 transacciones por segundo, el Tangle de IOTA afirma ser capaz de almacenar cientos por segundo. Como aclaración respecto al dato anterior, la velocidad a la que el Tangle es capaz de validar las transacciones es muy dependiente del ratio de transacciones que van generando los nodos para ser validadas.

Para elegir las transacciones que se validan esta tecnología no especifica ninguna regla en particular (es decir, se podría hacer de forma aleatoria o siguiendo diferentes criterios), pero si es recomendable que todos los nodos de la red los validen según los mismos criterios.

Actualmente, IOTA Tangle **no soporta Smart Contracts** (aunque que se encuentra en proceso de desarrollo para habilitarlos), debido a que no hay consenso y a que es asíncrona, cada nodo tiene una visión distinta de cuando se generaron las transacciones.

Otra diferencia entre esta y otras tecnologías Blockchain es su protección que tiene frente a los ataques realizados mediante ordenadores cuánticos. Estos ordenadores tienen una capacidad de cómputo muy superior a los ordenadores clásicos binarios. Por ello, con estos ordenadores se pueden realizar con más facilidad ataques del 51% para corromper una red de blockchain típica. Esta protección la logra a través de la inclusión de firmas Winternitz en su criptografía. Sin embargo, como contrapartida, no se debe reutilizar una misma dirección varias veces, ya que una parte de la clave privada es expuesta tras cada uso.

3.3 IoTeX

IoTeX es una blockchain **pública** desarrollada principalmente para generar confianza en el ambiente IoT, concepto al que se refieren como Internet of Trusted Things (**IoTT**), mediante un sistema que combina hardware seguro, blockchain y una identificación única llamada identidad descentralizada (DID). Un **DID** es un nuevo tipo de identificador único global. Están diseñados para permitir que individuos y organizaciones puedan generar sus propios identificadores utilizando sistemas en los que confían. Estos identificadores permiten a las entidades demostrar su posesión de estos mediante autenticación, a través de pruebas criptográficas como firmas digitales. Con el reemplazo por código de los intermediarios semi-confiables y este identificador, IoTeX trata de avanzar hacia los nuevos modelos de negocios basados en la privacidad, la confianza y la descentralización **[45]**.

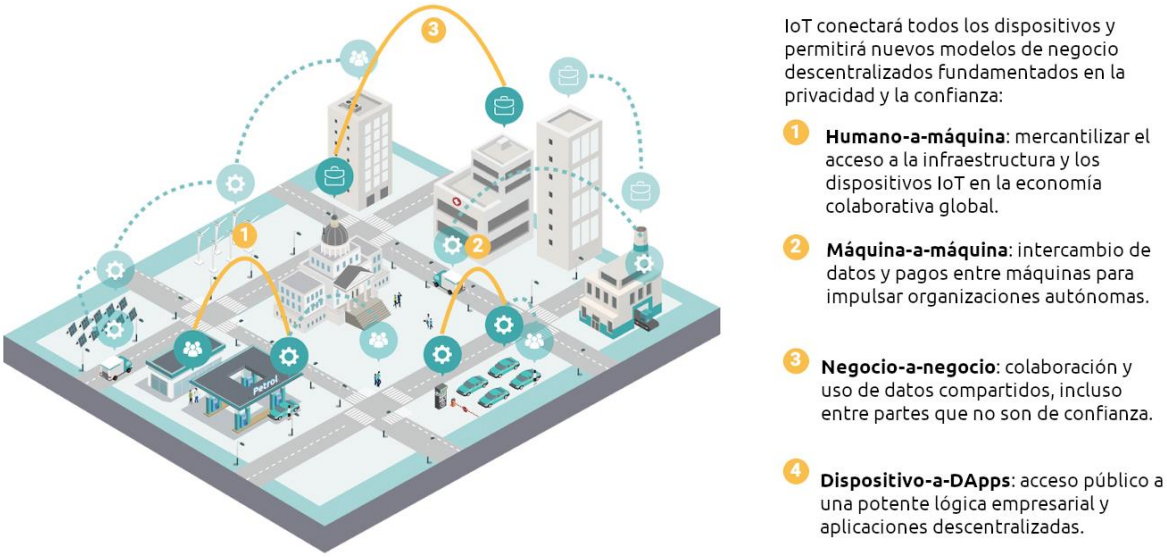

Figura 9 **[45]**. Gráfico de interacciones en la red IoTeX.

En el ecosistema de IoTeX por un lado se encuentra *el backend*. IoTeX utiliza un *híbrido entre cloud y blockchain* diseñado para almacenar de forma segura la información, a la vez que permite a sus propietarios autorizar su uso en aplicaciones de confianza o comerciar con ella en el mercado de datos IoT. Cuando se habla de aplicaciones de

confianza, se refiere a las aplicaciones o servicios en los que no hay riesgo en el que la información confidencial, recogida por los dispositivos de un propietario, sea utilizada y expuesta a menos que este les haya dado el permiso para usarla. Mediante filtros, como el sistema operativo, organización, servicios, etc. el propietario de la información puede decidir quién accede a sus datos.

Garantizar la confianza desde el principio del ciclo de vida de los datos y su privacidad, ya sea cuando se está transfiriendo o en reposo y en las aplicaciones en las que se utiliza, es como IoTeX se diferencia del resto de blockchains y otras soluciones.

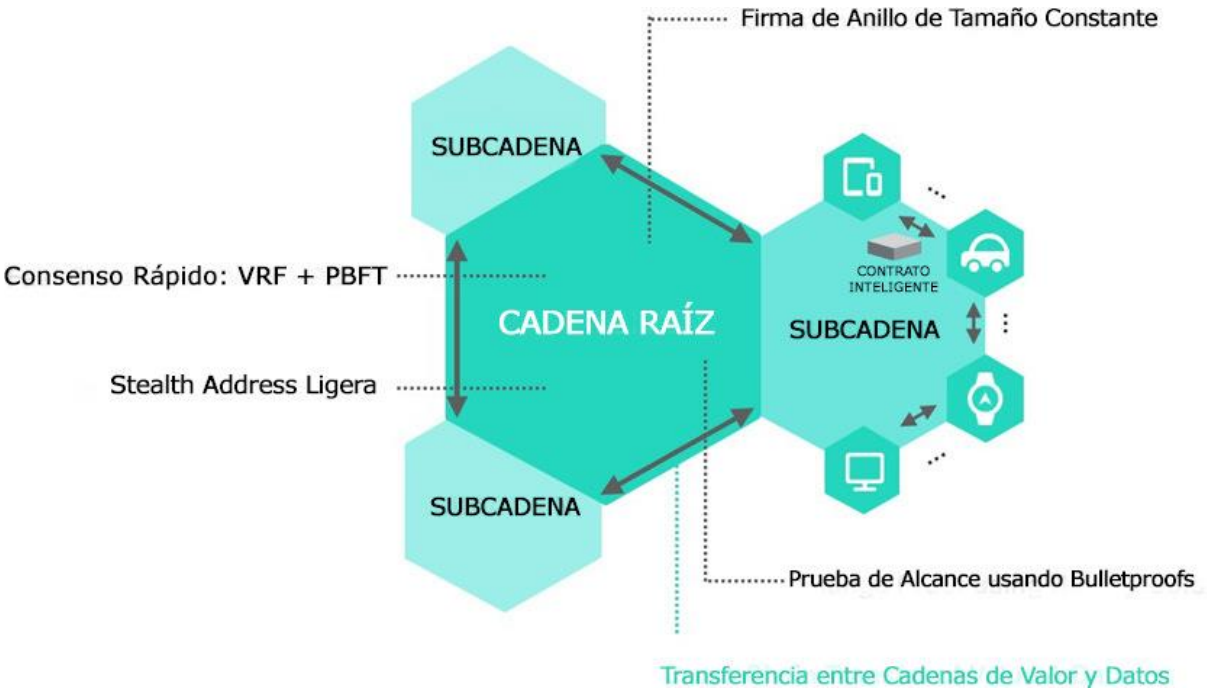

Figura 10 **[46]**. Esquema de funcionamiento y características de IoTeX.

Internamente IoTeX es una red de *muchas blockchains ordenadas según una jerarquía*. En la estructura de IoTeX la cadena principal (rootchain) gestiona muchas otras independientes o subcadenas (subchains o sidechains). Las **subchains** se conectan e interactúan con los dispositivos que tengan propósitos similares, trabajen en entornos parecidos o tengan el mismo nivel de confianza. Cuando estas subchains presentan errores o son atacadas, no afecta a la rootchain. Además entre ambos tipos de cadenas se posibilitan las transacciones para transmitir la información.

Para conseguir la máxima eficiencia y escalabilidad, IoTeX desarrolló su propio algoritmo de consenso **Roll-DPoS** basado en una variación del PoS (explicado en la sección 2.2.3). DPoS (Delegated Proof of Stake) se diferencia de PoS en que al principio existe un conjunto de delegados semi-confiables que serán elegidos por aquellos que tengan la cantidad de criptomonedas necesarias para poder votar. Esta variante permite generar bloques más rápido con un mayor rendimiento y menor latencia. Roll-DPoS mejora DPoS para poder soportar la arquitectura de subchains y aplicaciones descentralizadas (DApps) IoT a gran escala. Además, consigue una mayor democracia que DPoS, al seleccionar cada cierto tiempo un nuevo conjunto de delegados de forma aleatoria; y eficiencia, al implementar la finalidad de bloque. La **finalidad de bloque** es la garantía

de que cada bloque generado es definitivo y no se puede cambiar. Esto repercute especialmente en cómo se implemente la comunicación entre cadenas, debido a que las subcadenas deben esperar hasta que se logre la finalidad de bloque en la blockchain emisora antes de aceptar otras transacciones entrantes entre subcadenas. En la mayoría de blockchains públicas que no poseen la finalidad de bloque, la blockchain receptora tiene una garantía probabilística que depende del número de mineros que confirmen la transacción. Este consenso finalizador permite que la blockchain receptora tenga la *garantía con una sola confirmación de bloque en la blockchain emisora*. Ya que en IoT se espera que la transferencia de datos sea rápida, este mecanismo se vuelve indispensable tanto para la cadena principal como las subcadenas.

Y por el otro lado se encuentran los *elementos en el mundo físico*. El camino que recorren los datos que se mueven por la IoT suelen ser desde el dispositivo (por ejemplo sensores) a una pasarela (edge), para acabar directamente almacenado en una infraestructura backend (clouds, blockchains, bases de datos…). Desde ahí estos datos son analizados para sacarle algún partido en el mundo físico.

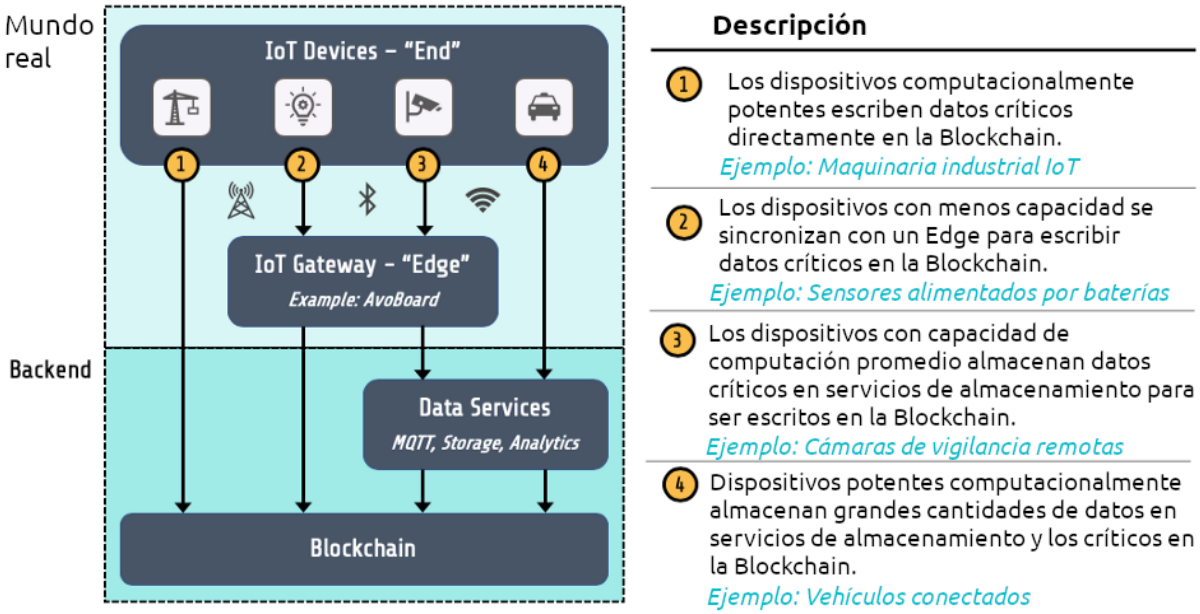

Figura 11 **[45]**. Ejemplo de los ciclos de vida de la información en IoTeX.

Para asegurar la total confianza, IoTeX hace uso de i) DIDs, para asegurar quienes son aquellos que envían y reciben los datos, y ii) dispositivos seguros diseñados y fabricados por ellos mismos basados en TEE, aunque no sean estrictamente obligatorios para utilizar la Blockchain. **TEE** o Trusted Execution Enviroment es un área segura en el procesador principal. Dicha área se encuentra en ejecución en paralelo junto al sistema operativo del dispositivo, es un entorno aislado, y garantiza que los datos y código cargados en dicho entorno mantengan tanto la integridad como la confidencialidad.

En Bitcoin y Ethereum la privacidad que proporcionan se basa en pseudónimos, y sus detalles de transacción no son confidenciales. Existen 3 aspectos clave de la privacidad en el contexto de las transacciones: la privacidad del emisor, receptor y la cantidad transferida. IoTeX implementa Stealth Adress para la privacidad del receptor, firma de

anillo para la del emisor y los Compromisos de Pedersen para la cantidad transferida. Una breve definición de estos últimos conceptos mencionados son:

Stealth Address es una tecnología que mejora la privacidad para proteger la privacidad de los receptores de criptomonedas. Las direcciones ocultas requieren que el remitente cree una dirección aleatoria única para cada transacción en nombre del destinatario, de modo que los diferentes pagos realizados al mismo beneficiario no se puedan vincular.

Las firmas de anillo son un protocolo integrado que sirve para comprobar que un mensaje ha sido firmado por uno de los signatarios de una lista autorizada.

Los compromisos de Pedersen son algoritmos criptográficos que permiten a un probador (alguien que tiene que demostrar) comprometerse con un determinado valor sin revelarlo ni poder cambiarlo, para convencer a un validador.

3.4 Qtum

Este proyecto de blockchain **pública** es de código abierto y su plataforma resulta de la unión del Modelo Bitcoin **UTXO** (Unspent Transaction), que ayuda en la trazabilidad de las transacciones, y el Modelo de Cuentas Ethereum y su **EVM** (Ethereal Virtual Machine) que logra añadir para los usuarios las funcionalidades de ejecución de programas sobre la blockchain o Smart Contracts. Además de mejorar la seguridad de la red mediante el manejo de acceso a los recursos de los computadores, ejecutando aplicaciones en el entorno controlado de la máquina virtual. La mezcla de ambas tecnologías permite que las diferentes técnicas de escalabilidad que se aplican para mejorar tanto a Bitcoin (la **Lightning Network**) como a Ethereum (la **Raiden Network**) sean aplicables a Qtum. Ambas consisten en lo mismo, en que la escritura sobre la blockchain se realice para gestionar la apertura y cierre de pagos bidireccionales, es decir **canales de pago**. Para que los usuarios puedan mediante este método enviar criptomonedas a cualquier otro destino en la red, la transacción se encamina utilizando Onion Routing (el mensaje encriptado en diferentes capas para que la comunicación sea anónima y difícil de seguir ya que cada nodo sólo conoce al contiguo), siguiendo diferentes canales de pago hasta alcanzar al destinatario. Esta *transferencia es casi instantánea y con costes reducidos*, y a su vez el receptor tiene la certeza de que no es vulnerable a un ataque de doble gasto debido a que los nodos intermedios no pueden robar los fondos ya que la transacción está encriptada. Aquellos nodos que mantengan un canal de pago tienen la responsabilidad de guardar el estado asociado a la última transacción y eliminar los previos.

El algoritmo de consenso utilizado por Qtum es **PoS v3**. Para poder comprender las diferencias entre el algoritmo de consenso PoS y PoS v3 se explicarán PoS v1 y v2 previamente. En PoS los usuarios reciben compensaciones según la cantidad de criptomonedas apostadas y el tiempo que las mantienen en la apuesta. **PoS v1** además añade el concepto de **edad de la moneda**, es un término que indica el tiempo que ha pasado desde la última vez que se utilizó la moneda. A mayor edad de la moneda la dificultad (el número que regula cuánto tiempo tardan los mineros en añadir nuevos bloques a la cadena) se reduce, si una moneda tiene la edad suficiente, de manera casi

instantánea produciría nuevos bloques. El problema con este enfoque fue que los usuarios *dejaron de gastar las monedas* y sólo utilizarlas en la prueba de participación y cuando lo hacían *apagaban los nodos* para ahorrarse el coste de mantenimiento de los nodos, lo cual era perjudicial para la salud de la red. **PoS v2** termina con el término de edad de la moneda y cambia el mecanismo de participación donde incluye el momento de creación del bloque anterior. Este momento de creación hacía vulnerable a la cadena al ataque de rango corto y fue eliminado. En **PoS v3** de QTum se hace *necesario mantener los nodos en línea* en todo momento *para recibir la recompensa*. Además, aquellos usuarios que no posean una gran cantidad de criptomonedas pero que mantienen su nodo en línea serán capaces de obtener recompensas por su apuesta frente aquellos que tienen más pero apagan los nodos. En cuanto a su seguridad, los datos de los bloques son organizados en varias partes ya que utiliza un núcleo hash, siendo capaz así de solventar de esta forma el problema Nothing-At-Stake **[47 y 48]**.

3.5 Hyperledger Iroha

Iroha es una blockchain **permisionada** perteneciente a la comunidad Hyperledger cuyo diseño trata de ser lo más confiable, simple y eficiente. Para facilitar y mejorar la experiencia del desarrollador no solo existen una variedad de librerías, sino que existen además una serie de mecanismos que hacen su mantenimiento y despliegue más sencillo. Iroha pretende ser una plataforma de contabilidad distribuida, autorizada y modular. Y ofrece una solución en el mundo empresarial donde los clientes que forman parte de una red deben ser conocidos y por lo tanto autorizados. En Iroha *no existe ningún tipo de criptomoneda de forma nativa*, aunque puede ser creada por un participante elegible según sea necesario para su propio uso empresarial. Tanto las interacciones con cambio de estado (escribir) como las consultas con el sistema son permisionadas, por tanto, sólo aquellos participantes con permiso pueden interactuar con el sistema. Una de las principales diferencias con otras Blockchains es que permite a los usuarios realizar funciones comunes como crear y transferir bienes digitales, mediante el uso de comandos (acciones atómicas que se permiten efectuar sobre el sistema) que se encuentran en el sistema, permitiendo a los desarrolladores realizar ciertas tareas más rápido con un riesgo menor.

Además, Iroha posee su propio algoritmo de consenso tolerante a fallas bizantinas llamado **YAC** (Yet Another Consensus) de alto rendimiento y escalabilidad, que implementa la finalidad de las transacciones con una baja latencia.**[49, 50 y 51]**. Dicho algoritmo consta de 5 fases:

1. El sistema comparte una propuesta para todos los participantes. Una propuesta es un bloque sin firmar creado y compartido por sus participantes que contiene un lote de transacciones ordenadas.

2. Los participantes calculan el hash de una propuesta verificada y lo firman. La tupla <Hash, Signature> resultante se denomina voto.

3. Según los hashes calculados anteriormente, cada participante realiza una lista de participantes en orden. Para ello, la función de ordenar ha de ser conocedora de todos los participantes que votan en la red y se basa en el hash del bloque propuesto. El primer participante de la lista se llama el líder. El líder se responsabilizará de recoger los votos de otros participantes y enviar la señal de confirmación.

4. Cada participante vota, y el líder recolecta todos los votos y determina la mayoría de votos para un cierto hash. Con los votos del bloque de confirmación el líder envía un mensaje de confirmación. Esta respuesta se llama commit.

5. Una vez enviado el commit, los participantes verifican la confirmación y añaden el bloque a la cadena. En este punto, el consenso está completo.

Si con el líder se da algún fallo o sufre una caída de la conexión, otros participantes establecen un tiempo límite para recibir el mensaje de confirmación del líder. Cuando este tiempo se excede el siguiente participante de la lista se convierte en líder.

4

Análisis de las distintas DLTs

En esta sección se analizarán las distintas DLTs introducidas en la sección 3 y supuestamente diseñadas para escenarios IoT según una serie de requisitos relacionados con la aplicación de dichas DLTs en entornos IoT. Posteriormente, se seleccionará(n) aquella(s) DLT(s) considerada(s) más óptimas para la prueba de concepto.

4.1 Requisitos de DLTs en escenarios IoT

Los requisitos han sido seleccionados de forma coherente con las necesidades de proyectos IoT generales de gran envergadura, y teniendo en cuenta las características de las aplicaciones y escenarios IoT introducidos en la sección 2.

R1. Descentralización

La descentralización es el proceso de distribuir o dispersar funciones, poderes, personas o cosas fuera de una ubicación o autoridad central. El hecho de que no exista ninguna figura con una autoridad mayor que cualquier otra sobre el sistema es suficiente para afirmar que una blockchain es descentralizada.

R2. Capacidad para procesar transacciones

Las aplicaciones IoT dependiendo de la envergadura del proyecto pueden tener la necesidad de procesar grandes o pequeñas cantidades de información. Debido a las limitaciones de proceso que tienen las blockchains por el cuello de botella que surge con

PoW y el consenso, las transacciones por segundo validadas no serían suficientes para mantener una aplicación IoT, ya que estas están caracterizadas no por el tamaño de los datos que envían sino por su cantidad. Por tanto, para que podamos considerar una blockchain mejor diseñada para IoT, en este aspecto, que otras blockchain tradicionales, sus transacciones por segundo deberán ser capaces de alcanzar un ratio mayor a 1000.

R3. Escalabilidad

La escalabilidad es la capacidad del sistema para crecer y así acomodar una demanda creciente. En el ámbito Blockchain la escalabilidad se refiere principalmente a incrementar su capacidad para gestionar un número de transacciones mayor. Por tanto, según las diferentes soluciones que ofrezca cada DLT estudiada, ya sea por el diseño de su DLT, implementación de canales de pago, subchains y otros protocolos que consigan aumentar su escalabilidad, serán evaluadas.

R4. Concienciada con los dispositivos de baja potencia computacional y la optimización del consumo de batería del dispositivo al máximo para alargar su vida.

Para que una aplicación IoT pueda desplegarse sobre una blockchain es necesario que esta tenga en cuenta la clase de dispositivos que se van a conectar a ella. En algún ejemplo de marketing, ya se ha visto cómo se intenta vender que una blockchain sea mejor para este ámbito porque utilizan un algoritmo hash más apropiado para ellos, o uno que requiera de una menor potencia computacional. Sin embargo, el hecho de que dispositivos con baterías tengan que realizar PoW no es adecuado ya que dicho proceso, como ya se ha mencionado (sección 2.2.3) es muy costoso energéticamente. Por tanto, según las diferentes soluciones que ofrezca cada compañía a los retos implícitos en el uso de la tecnología Blockchain en el entorno IoT, se considerará este requisito como cumplido o no si realmente lo que ofrecen cumple con el requisito.

R5. Seguridad en los datos (confidencialidad e integridad de los datos)

Los dispositivos IoT además de conectarse a internet, ofrecen servicios más inteligentes y de mayor complejidad que un dispositivo convencional. Por ello se ha de ser consciente que al conectarlos a internet se aumenta el riesgo de sufrir un ciberataque. Y el hecho de que estos además sean bastante heterogéneos y combinables entre sí no mejora la situación. Además, en el transcurso de la información desde los dispositivos al servicio que la almacenará, se ha de suponer que el medio de transmisión es siempre inseguro, y por tanto, se ha de asegurar que la información se mantenga siempre en confidencialidad con el usuario. Por otro lado, se tendrá en cuenta los sistemas que utilizan las DLT estudiadas para mantener la integridad de sus datos. Por tanto, en este apartado se tendrá en cuenta de forma individual la solución que ofrece cada DLT analizada.

4.2 Análisis de DLTs orientadas a la IoT

DLT\Requisito	R1	R2	R3	R4	R5
Ethereum	++	-	+*	++	++
IOTA	-	-	-	-	-
IoTeX	++	+	++	++	++
QTUM	++	++	++	++	++
Iroha	-	+	++	++	++

Tabla 1: Cumplimiento de los requisitos por parte de cada DLT.

Un resumen del análisis de las DLTs descritas en los apartados anteriores se encuentra en la Tabla 1. En la tabla se resume el cumplimiento de los requisitos por parte de cada DLT. Cuando no se cumplen se anota con un '-', cuando se cumplen y no se han encontrado estudios o pruebas que lo amparen se le anota con '+', y cuando existan estudios o pruebas del cumplimiento del objetivo con '++'. En el caso de Ethereum, '+*' significa que con la finalización del desarrollo de las soluciones capa 2, sobre las que el equipo de desarrollo trabaja, mejorará la escalabilidad de Ethereum. A continuación, se describirán en más detalle los pros y contras de cada una de estas DLTs.

4.2.1 Ethereum

Ethereum es una red descentralizada. Respecto a la *capacidad de Ethereum para procesar transacciones*, es capaz de procesar entre 10 y 15 transacciones por segundo.

Respecto a la escalabilidad de Ethereum, actualmente existen soluciones implementadas como la Raiden Network. La Raiden Network es una capa construida sobre la blockchain de Ethereum para incrementar su escalabilidad, actualmente se encuentra en la versión 2.0.0. Raiden permite a sus usuarios enviar transacciones de una manera más eficiente y menos costosa. Esta red interactúa con tokens que utilizan el estándar de Ethereum ERC20 (tokens divisibles que se utilizan como moneda). Esta red no requiere de un consenso global, en su lugar, para conservar la integridad de las transacciones utiliza firmas digitales y hashlocks (hashes encriptados con criptografía asimétrica). Conocido como Proof of Balance, este tipo de intercambio de tokens utiliza canales de pago. Los canales de pago o estado facilitan la transferencia de tokens de forma bidireccional entre dos participantes sin necesidad de la intervención de la blockchain.

Además, como se mencionó anteriormente en la introducción a Ethereum, con Ethereum 2.0 se están desarrollando nuevas medidas de escalabilidad tanto en la capa 1 como en la 2. *En la capa 1* se está trabajando en implementar el algoritmo de consenso Proof of Stake y además el Sharding, que consiste en dividir la información de una base

de datos y repartir su carga. En el contexto de las blockchain el Sharding reduce la congestión de la red incrementando las transacciones por segundo y aligerando la carga de los nodos validadores, que no tendrán que procesar la totalidad de todas las transacciones de la red. *En la capa 2* se está trabajando en la implementación de Rollups, Canales de Estado, Plasma y Sidechains. Cabe destacar que, aunque estas soluciones buscan integrarse dentro de Ethereum 2.0, también podrían integrarse sobre Ethereum 1.0.

Dentro de las Rollups (explicadas en la sección 2.2.4.2), Ethereum 2.0 añadirá las ZK-Rollups y las Optimistic Rollups.

Los Canales de Estado utilizan contratos multifirma que permiten a los participantes realizar transacciones de forma rápida fuera de la cadena, y luego envían su estado final a la cadena principal. Esta solución minimiza la congestión de la red, las tarifas por el coste en gas y retrasos.

Las Sidechains son cadenas independientes compatibles con la EVM que se ejecutan en paralelo con la red principal. Estas cadenas pueden comunicarse entre ellas y con la principal, además de poder utilizar reglas de consenso y parámetros de bloque distintos a la cadena principal.

Las cadenas Plasma son blockchains separadas pero que están ancladas a la red principal de Ethereum (cadenas anidadas). Y utiliza prueba de fraude para resolver disputas al igual que en las Rollups Optimistas.

Respecto a la *concienciación con los dispositivos IoT*, Ethereum posee 3 tipos de clientes o nodos: el completo, el de archivo y el ligero. Los nodos ligeros almacenan las cabeceras de los bloques de la cadena y solicitan el resto de información cuando la necesitan, además de poder verificar la validez de los datos recogidos en la blockchain. De esta forma, los dispositivos que no dispongan de los recursos necesarios para almacenar la cadena completa podrán utilizarla. Con Ethereum 2.0 esto cambiará, aunque los dispositivos no tengan una gran capacidad de computación, con ser capaces de almacenar la blockchain completa y cumplir los requisitos para ejecutar un nodo completo, podrán participar en la validación de bloques, ya sea de forma individual (apostando 32 ETH como mínimo) o en grupo (llegando a la apuesta mínima entre todos), debido al cambio de algoritmo de consenso de PoW a PoS.

La seguridad de Ethereum está intrínsecamente ligada a la criptografía (asimétrica y hashes), la tecnología Blockchain (descentralizada y distribuida) y a la minería. Para poder modificar una transacción y usar dos veces el mismo token o falsear la cantidad de criptomonedas que se poseen, el atacante debe competir con el resto de la red y controlar más de la mitad de la capacidad de cómputo de la red (el ya explicado ataque del 51%). Lo que hace el ataque económicamente inviable. Se pueden dar casos en los que se exploten fallos de vulnerabilidad como la reentrada, pero estas vulnerabilidades son inherentes al desarrollo de Contratos Inteligentes y no a las diferentes plataformas en sí, por tanto no se tendrá en cuenta. Ethereum no refleja en su documentación protocolos y algoritmos adicionales que cuiden de la privacidad de los usuarios o datos

representados en la blockchain, más allá del pseudoanonimato proporcionado por las direcciones de los usuarios.

4.2.2 IOTA

La arquitectura de IOTA no es totalmente descentralizada, dada la existencia de la figura del coordinador. Como ya se ha mencionado en la sección 3.2, no puede impedir la participación de cualquiera en la red, pero sí decidir si una transacción es válida o no, o incluso parar el funcionamiento de la red.

Respecto a *su capacidad para procesar datos*, IOTA es ineficiente. Su computación se basa en la lógica ternaria. Dado que el hardware común y sistemas de redes utilizan el sistema binario, supondría una ineficiencia dada la necesidad de traducir de un sistema a otro las operaciones mediante software. No obstante, ni utilizando hardware preparado para ello sería eficiente, como demuestra el estudio "Ternary circuits: why R=3 is not the Optimal Radix for Computation" **[52]**, en el que se demuestra como los circuitos binarios en general y especialmente los operadores aritméticos superan a los circuitos ternarios.

Además la ineficiencia de IOTA también ha quedado demostrada a nivel práctico, donde el rendimiento en las transacciones por segundo era mucho menor a las esperadas. En un test donde se envían 10 transacciones a la vez, cada transacción tomaba alrededor de 1 minuto para ser enviada. En otros tests con distintos parámetros incluso más, probando que es incapaz de soportar distribución de datos en tiempo real **[53]**.

Respecto a *la escalabilidad de IOTA*, en unas circunstancias ideales podría una de las más escalable de todas debido a su diseño basado en DAGs. En lugar de almacenar transacciones en bloques, cada transacción actúa como un nodo que valida otras dos transacciones mediante un PoW de dificultad reducida. Por tanto, mientras el ratio de transacciones generadas por los participantes sea alto, la velocidad a la que se confirman y añaden nuevas transacciones será más alta. Sin embargo, este enfoque añade problemas en su seguridad (mencionados en el análisis de su seguridad), que es lo que lleva a IOTA a sacrificar su descentralización y necesitar de un nodo coordinador. Además, su límite de escalabilidad vendrá dado por el ancho de banda de los nodos, ya que para crear consistencia en la red y poder operar en ella es necesario sincronizar la información de todos los nodos, y por tanto la velocidad a la que estos se comuniquen es crucial. En escenarios reales este factor no es siempre controlable, en una red que proporciona confirmaciones instantáneas sin consenso (de forma asíncrona) constantemente, causa problemas en aquellos nodos con limitaciones de red, que rápidamente quedan desincronizados, y en su lugar, comenzarán a ver acumularse transacciones no confirmadas, lo que evita que las nuevas transacciones se resuelvan con la misma rapidez **[54]**.

Con respecto a la *adaptabilidad de IOTA a los dispositivos IoT*, para que un nodo pueda enviar información, este ha de ser un nodo completo y realizar PoW, lo cual es contraproducente para casos de uso IoT, y subir datos a un edge/cloud para que este envíe los datos a una DLT es una mejor solución también aplicable al resto de DLTs.

Además, en las pruebas de rendimiento a las que se ha hecho referencia en la evaluación del segundo requisito, IOTA tomaba el 100% de los recursos del dispositivo impidiendo la ejecución de otros procesos en el mismo **[53]**.

Con respecto a la *seguridad de los datos en IOTA*, esta es muy limitada: debido a que no existe un coste económico en primera instancia (ya que todos los dispositivos en la red son mineros), no existe nada que impida realizar grandes cantidades de transacciones a modo de spam, especialmente cuando se cuenta dentro de la propia red con dispositivos con diversos ratios de hasheo. En el paper de IOTA, "The Tangle", se hace la asunción de que no existe una entidad capaz de generar un gran número de transacciones con un peso (dificultad en el PoW) aceptable en un periodo corto de tiempo **[55]**. Las barreras que tratan de solventar el problema de que el Tangle crezca de forma ilimitada y sea validada completamente por el mismo usuario es i) el coordinador, que tiene poder para decidir si una transacción es válida o si se debe considerar como spam o directamente corrupta; y ii) un PoW de dificultad inferior al de otras DLTs, debido a que busca encontrar el balance para que los dispositivos IoT puedan realizarlo y prevenir el spam. Por tanto, ya que cada transacción requiere realizar PoW con una dificultad reducida, la cantidad de recursos requeridos para superar la potencia de procesado del resto de la red no sería tan elevada como en otras plataformas blockchain. Lo que hace a IOTA más vulnerable a ataques del 51%. Para finalizar este apartado, el equipo de desarrollo IOTA decidió utilizar su propio algoritmo hash por motivos de diseño, vulnerando la buena práctica en criptografía de no utilizar algoritmos criptográficos propios, lo que resultó en un problema más grave cuando fue atacada y tuvo que suspender el funcionamiento del sistema durante un tiempo. En el estudio liderado por Neha Narula, *"Cryptographic vulnerabilities in IOTA"* **[56]** se describen algunas vulnerabilidades de IOTA en mayor profundidad.

4.2.3 IoTeX

IoTeX es una DLT totalmente descentralizada.

Respecto a su *capacidad para procesar un gran número de transacciones,* IoTeX afirma que las transacciones por segundo de su DLT puede alcanzar más de 1000 transacciones según la cantidad de delegados.

Figura 12 **[57]**. Transacciones por segundo y latencia de IoTeX según el número de delegados.

IoTeX es escalable debido a la combinación de dos factores de su diseño. Por un lado, se encuentra la implementación de sidechains, que funcionan de manera que validan sus propias transacciones en paralelo y se actualizan frente a la rootchain de forma periódica. De forma que, donde la rootchain percibe una actualización de las sidechains, en cada sidechain se han generado numerosas transacciones. Lo que se traduce en un incremento en el ratio de producción de transacciones en la rootchain. Y por el otro lado, el algoritmo de consenso Roll-DPoS y la finalidad de bloque (explicado en profundidad en la sección 3.3) que agilizan la validación de las transacciones lo que supone una gran ventaja.

IoTeX tiene en cuenta que las capacidades de los dispositivos en su red son heterogéneas. Para poder enviar transacciones, usar contratos inteligentes y almacenar y consultar la información en su red, no es necesario mantener almacenada toda la información de la blockchain en el dispositivo. Utiliza Roll-DPoS como algoritmo de consenso y por tanto al contrario que IOTA no fuerza a ningún participante a hacer PoW. Además también implementa carteras ligeras que los dispositivos pueden utilizar.

Respecto a la seguridad en los datos, además de las propiedades inherentes a la tecnología blockchain, IoTeX se preocupa de la protección de los mismos desde que se generan sus dispositivos de confianza como la Ucam, hasta que se almacenan mediante una combinación de características (DID, firma de anillo, Stealth Address y permisos a nivel usuario sobre los datos propios que almacenan) que asegura la confidencialidad, privacidad e integridad de los datos.

4.2.4 QTUM

QTUM es una blockchain descentralizada. Como característica adicional, implementa un protocolo de configuración descentralizado, **Decentralized Governance Protocol** (DGP),

que mediante votaciones permite realizar cambios en los parámetros de configuración de la blockchain (como el tamaño de bloque), sin necesidad de realizar hard forks (cambios a la blockchain que pueden volver inválidos bloques anteriores debido a cambios en su configuración), permitiendo a la blockchain adaptarse fácilmente a las necesidades de la red.

Respecto a la capacidad para procesar transacciones QTUM ha demostrado en pruebas de rendimiento superar las 10.000 transacciones por segundo **[58]**.

Respecto a la escalabilidad de QTUM, en sus fases iniciales al estar fundamentada en Bitcoin y Ethereum padecía de sus mismos problemas de escalabilidad. Con el desarrollo de ambos proyectos dieron lugar a soluciones igualmente aplicables a QTUM, la Lightning Network de Bitcoin y Raiden de Ethereum. De esta manera los dispositivos IoT son capaces de realizar grandes cantidades de transacciones contando con requisitos de potencia computacional y almacenamiento mínimos. También implementa el protocolo Segregated Witness (SegWit), una mejora que cambia la forma en la que se almacenan los datos en Blockchain y aumenta su rendimiento. En una blockchain que no implementa SegWit, los datos de las firmas en las transacciones se almacenan en el los bloques donde se añaden, con SegWit, esta información se separa y se escribe en sidechains, liberando el espacio del bloque y permitiendo almacenar más transacciones en él.

Con *respecto a su concienciación con el uso de dispositivos IoT,* QTUM utiliza PoS y por tanto, los dispositivos IoT no necesitan realizar PoW. Además, implementa Unspent Transactions Outputs (UTXO), el mismo sistema que Bitcoin. En UTXO las criptomonedas se almacenan en forma de transacciones no gastadas. Funciona de la siguiente manera: un usuario A que tiene 7 monedas almacenadas en dos transacciones Trx1 y Trx2 de 5 y 2 monedas respectivamente. Si quiere transferir 6 al usuario B se creará una Trx3 de seis monedas para B y quedará una Trx4 en de 1 moneda para A en lugar de las dos anteriores. Mediante este modelo se vuelve más sencillo el rastrear el historial de cada transacción a través de la blockchain volviéndola una opción más segura.

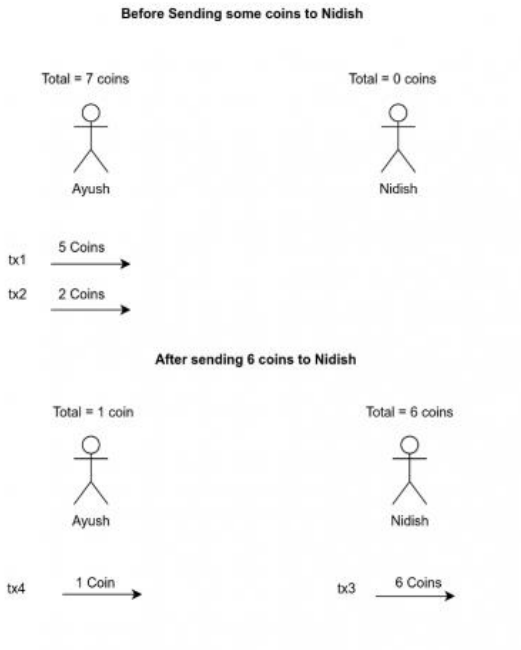

Figura 13 **[59]**. Funcionamiento del sistema UTXO.

UTXO además soporta **SPV** (Simple Payment Verification). A aquellos clientes no interesados ni en la creación de bloques ni en el consenso (lightweight client), nodos que únicamente quieren verificar que ciertas transacciones se encuentran dentro de la blockchain sin necesidad de almacenarla al completo, les permite descargar exclusivamente las cabeceras de los bloques (que son menos pesadas que el bloque al completo), para verificar la inclusión de las transacciones mediante **Proof of Inclusion** (un tipo de prueba que permite proporcionar información sobre el camino a través de un árbol Merkle que prueba que una hoja está en el árbol) **[48]**. En Ethereum directamente se asocia un número de criptomonedas a las cuentas, lo que hace mucho más complicado rastrear transacciones. Para conseguir que la EVM funcionara sobre UTXO el equipo de QTUM implementó una capa de abstracción (Account Abstract Layer, AAL) que sirve como interfaz entre la blockchain y la EVM. Con la combinación de ambos sistemas (EVM y UTXO) se hace posible que dispositivos móviles sean capaces de ejecutar contratos inteligentes con carteras lite y hacer uso del protocolo SVP. Al contrario que otras plataformas de contratos inteligentes, que requieren que sus usuarios ejecuten un nodo completo para tener una copia completa de la cadena, y poder hacer uso de las DApps, haciéndolas insostenibles para el entorno IoT.

Respecto a la seguridad en los datos, además de las propiedades inherentes a la tecnología blockchain, QTUM ha desarrollado el protocolo **Phantom**, basado en zk-SNARK (zero-knowledge Succinct Non-interactive ARgument of Knowledge), que consiste en pruebas capaces de probar la posesión sobre un dato, mientras que al mismo tiempo demuestra que dicho dato es correcto, sin que este sea revelado, manteniendo la confidencialidad y permitiendo a sus usuarios crear transacciones de forma anónima.

4.2.5 Hyperledger Iroha

Hyperledger Iroha es una blockchain permisionada, sus usuarios poseen distintos permisos en el sistema, y por tanto, no es descentralizada.

Respecto a su *capacidad para procesar transacciones* no se han encontrado estudios de su rendimiento, pero Hyperledger tiene como objetivo alcanzar las 2000 transacciones por segundo.

Respecto a su escalabilidad, el uso del algoritmo de consenso YAC tiene la capacidad de mejorar bastante el rendimiento debido a que se basa en votar por el hash de los bloques y es capaz de garantizar la finalidad de los bloques.

Con *respecto a su concienciación con el uso de dispositivos IoT,* además de no forzar a realizar a hacer PoW a los dispositivos que conforman la red, también han logrado que en dispositivos de hardware ARM con linux, como o son las Raspberry Pi o dispositivos Android rooteados, sea posible ejecutar un nodo completo de forma eficiente. El resto de dispositivos IoT podrán acceder a la Blockchain a través de carteras ligeras, para comprobar transacciones y enviarlas para su posterior validación.

Respecto a su seguridad, Iroha cuenta con un sistema de permisos robusto, de forma que aquellos nodos autorizados puedan leer o escribir información, lo que asegura la privacidad en los datos, entre las diferentes entidades que conforman su red.

4.3 Elección de la tecnología

Para elegir la tecnología que se utiliza en la prueba de concepto se ha tenido en cuenta todos los puntos anteriores. De primera mano, por ello se descarta a IOTA, que no cumple las expectativas según diversas fuentes y estudios. Debido al tipo de sistema que se implementa en la prueba de concepto, las blockchains que encajan con sus requisitos son las blockchains públicas no permisionadas con una moneda criptográfica propia para realizar transacciones de pago, por ello también se ha descartado Iroha, aunque sea una buena opción para otros casos de uso.

Las tecnologías DLT resultantes más adecuadas a los requisitos fueron IoTeX y QTUM. En un principio se escogió la tecnología QTUM, ya que esta presenta unas buenas capacidades de escalabilidad y además ha demostrado ser capaz de soportar un ratio mayor de transacciones por segundo que IoTeX. Sin embargo, cuando se empezó el desarrollo de la prueba de concepto, aunque el envío de transacciones fuese rápido y el funcionamiento del contrato inteligente correcto (a través de la aplicación cartera de QTUM, QTUM Wallet), durante el desarrollo de la aplicación web, QTUM exigía el uso de un plugin de navegador que sería el que almacenaría la clave privada de la cartera para así poder firmar transacciones. Dicho plugin se encontraba en fase beta, y además de la poca documentación sobre como programar sobre dicho plugin, este daba problemas como quedarse congelado sin dar ninguna información al usuario. Esta clase de problemas da lugar a confusiones al usuario sobre si las transacciones de

criptomonedas o llamadas a los contratos inteligentes se llegan a efectuar y puede dar lugar a pérdidas de dinero en caso de volver a realizarlas.

Además de los fallos anteriores y la poca documentación disponible respecto al plugin, QTUM sólo disponía de dos opciones de configuración (la red de testing y la principal), impidiendo que se pudieran realizar las pruebas de la aplicación en un entorno local - es decir, en un entorno en donde las criptomonedas no sean limitadas para desarrollar la aplicación. Este impedimento es importante ya que hay un coste en realizar transacciones, ejecutar llamadas al contrato que modifiquen la blockchain, y desplegar dicho contrato. Desafortunadamente, la página web para obtener criptomonedas en la red de testeo de QTUM provee de una cantidad variable entre quince y cuarenta criptomonedas al día si se reclaman, lo cual era insuficiente para la realización de las pruebas. Todos estos problemas no se encuentran en la red IoTeX, por lo cual al final la prueba de concepto se desarrolló sobre dicha plataforma.

5

Prueba de concepto

En este apartado se explicará en qué ha consistido el sistema que hace uso de la blockchain de IoTeX para funcionar. Dicha prueba de concepto deberá cumplir los siguientes objetivos:

- Desarrollar un contrato inteligente que funcione como backend donde se despliegue toda la lógica de negocio, permitiendo realizar pagos y almacenar y modificar la información que almacene.

- Desarrollar un programa para dispositivos IoT capaz de interactuar con un contrato inteligente con el objetivo de causar algún efecto en el mundo físico.

- Desarrollar una aplicación web que sirva como interfaz de usuario que sea capaz de comunicarse con un contrato inteligente. Dicha aplicación web podrá ser utilizada a través de ordenadores y dispositivos móviles.

5.1 Introducción

La idea para la prueba de concepto consiste en un sistema de iluminación para pistas de un polideportivo. Para llevar a cabo la prueba de concepto se hace uso de la blockchain Testnet de IoTeX. En la blockchain se almacenan precios, permisos de acceso para ejecutar funciones de gestión y los datos necesarios sobre las pistas, que actualizarán su estado cuando el contrato recibe una transferencia. También se ha implementado una aplicación web para facilitar el uso del sistema a los clientes y la administración del contrato inteligente a los gestores y administradores. La aplicación web no envía información a la blockchain, para recibir los datos los consulta a través de la aplicación cartera IoPay, una vez haya sido desbloqueada una cartera, y para el envío de

transacciones las construye dentro de IoPay para ser confirmadas por el usuario. Y finalmente, el dispositivo IoT utilizado para la prueba de concepto, una Raspberry Pi, mediante llamadas a la Testnet comprobará el estado de las pistas que se les indique en el arranque del programa, para encender y apagar las luces de un circuito de LEDs controlado por sus pines de entrada y salida.

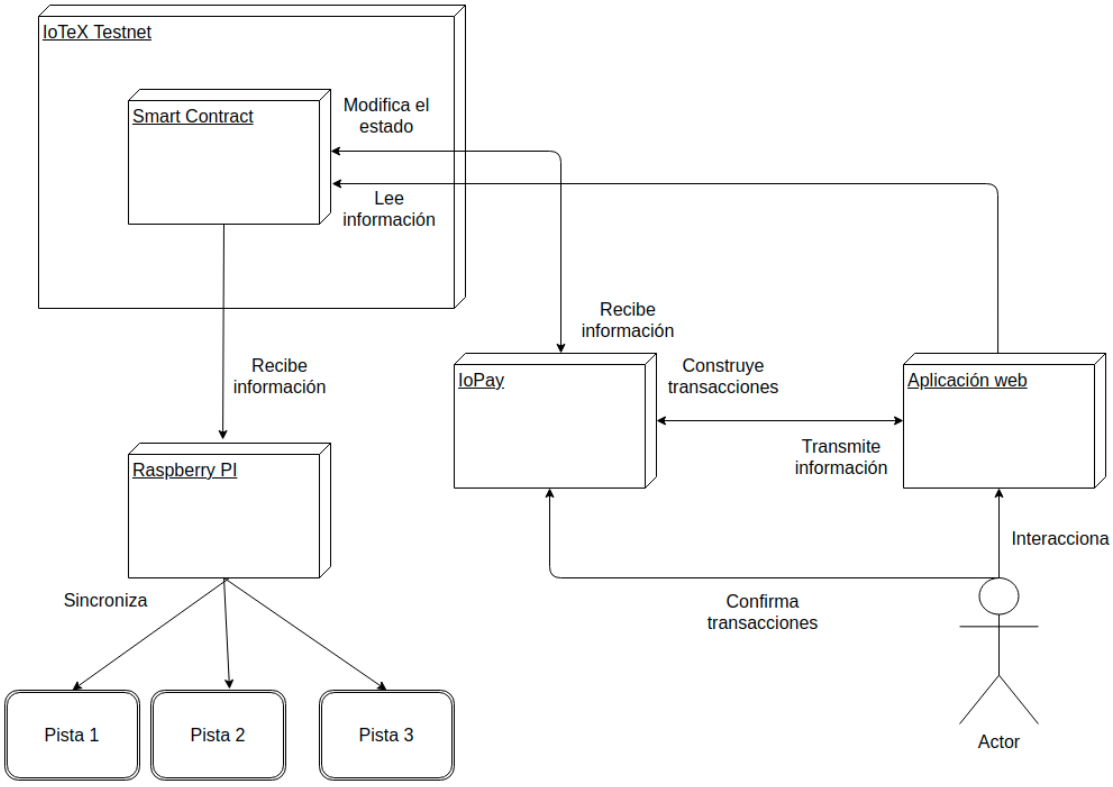

Figura 14. Esquema de comunicación entre las distintas partes del sistema.

Antes de continuar con la definición detallada de la prueba de concepto, es necesario justificar con detalle la adecuación en el uso de las tecnologías Blockchain para la realización de esta prueba de concepto (el sistema de iluminación para pistas de un polideportivo). Esto es necesario debido al "hype" y el peligro que existe con el uso indiscriminado de dichas tecnologías.

La decisión de haber escogido Blockchain frente a otras tecnologías backend y almacenamiento para este caso de uso de la IoT se justifica de diversas maneras frente a la típica estructura cliente, servidor y base de datos.

- La primera es por la capacidad de funcionar de forma automática mediante pagos instantáneos. Blockchain permite de manera instantánea enviar dinero (criptomonedas) de forma global con una cuota ínfima en comparación a las transferencias express de los bancos. De esta forma, un usuario puede realizar un pago a través de la interfaz web y el sistema de iluminación se activará de forma automática a la hora requerida, gracias a la interacción entre el contrato inteligente y el objeto IoT encargado de la iluminación.

- La segunda es que no es necesario mantener servidores ni contratar servicios cloud para el backend o servicios de almacenamiento de datos. Una vez se despliega el contrato inteligente, la propia red lo mantiene de forma descentralizada, lo que le aporta mayor fiabilidad (menos tendencias a averías) y no tener que pagar una cuota anual o mensual para mantener los servicios. La única cuota a pagar son el despliegue del contrato inteligente y las comisiones cobradas por el uso de la red blockchain en aquellas funciones que modifican el estado del contrato inteligente.

- La tercera es la seguridad e integridad de los datos. En comparación con otras tecnologías de almacenamiento donde la información es modificable, Blockchain es más segura. La única forma en la que un usuario malicioso puede modificar el contenido de los datos almacenados en el contrato inteligente es corrompiendo la red blockchain completa (suponiendo que el contrato inteligente no contiene vulnerabilidades), lo cual requiere de un esfuerzo tanto computacional como económico inasumible. En comparación, cualquier agente malicioso puede corromper el estado de un sistema (como una base de datos) si consigue infiltrarse.

5.2 Tecnologías y herramientas utilizadas

5.2.1 Node.js

Node.js es un entorno de tiempo de ejecución de JavaScript. Este entorno de tiempo de ejecución en tiempo real incluye todo lo que se necesita para ejecutar un programa escrito en JavaScript. Ideado como un entorno de ejecución de JavaScript orientado a eventos asíncronos, Node.js está diseñado para crear aplicaciones network escalables. La versión de Node.js utilizada ha sido la 10.19.0.

Figura 15. Logo de Node.

5.2.2 React

React es una librería de Javascript para construir interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página. React se basa en

componentes y permite diseñar vistas interactivas capaces de gestionar su propio estado y de actualizarse y renderizarse de forma sencilla y eficiente entre otras de sus numerosas características.

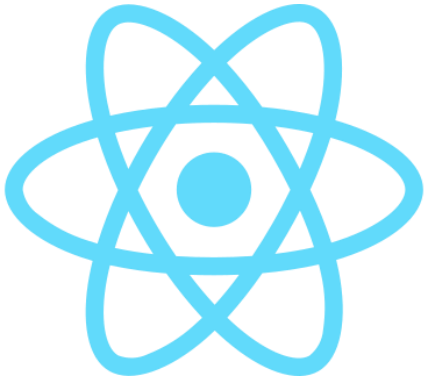

Figura 16. Logo de React.

5.2.3 Bootstrap

Bootstrap es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios web y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como extensiones de JavaScript adicionales. A diferencia de muchos frameworks web, solo se ocupa del desarrollo front-end.

Figura 17. Logo de Bootstrap.

5.2.4 IoTeX

IoTeX es una blockchain desarrollada principalmente para generar confianza en el ambiente IoT, concepto al que se refieren como Internet of Trusted Things (IoTT), mediante un sistema que combina hardware seguro, blockchain y una identificación única llamada identidad descentralizada (DID).

Figura 18. Logo de IoTeX.

5.2.5 Visual Studio Code

Visual Studio Code es un editor de código fuente ligero pero potente que se ejecuta en escritorio y está disponible para Windows, macOS y Linux. Viene con soporte incorporado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes (como C++, C#, Java, Python, PHP, Go, Solidity) y entornos en tiempo de ejecución (como .NET y Unity).

Figura 19. Logo de Visual Studio Code.

5.2.6 Firebase

Firebase es una plataforma para el desarrollo de aplicaciones web y aplicaciones móviles lanzada en 2011 y adquirida por Google en 2014. Se encuentra ubicada en la nube, integrada con Google Cloud Platform. Su principal función es ayudar al desarrollo de aplicaciones de alta calidad de forma rápida, pudiendo almacenar todo en la nube, testear la app o poder configurarla de manera remota. Además, cuenta con un sistema de almacenamiento, hosting, bases de datos NoSql y de tiempo real, sistemas de almacenamiento y la posibilidad de autenticación a través de Google, todo ello de forma sencilla a través de la API que cuenta con muchos ejemplos en su documentación. También cuenta con funciones analíticas y de escalabilidad para proporcionar el mayor control posible sobre el rendimiento de la aplicación.

Figura 20. Logo Firebase.

5.3 Smart Contract

En esta sección se explicará el porqué del uso de la tecnología Smart Contract en el proyecto, además de explicar cómo se estructura y qué funciones implementa el que se ha desarrollado.

Para la implementación de lo que conformaría el backend, en la blockchain, se ha utilizado la tecnología Smart Contract, ya que esta es la forma en la que se puede almacenar información, excluyendo las transacciones con la moneda nativa, y programar funcionalidades sobre dicha información en IoTeX.

En el desarrollo de estos smart contracts, la modularización se consigue mediante la herencia de funcionalidades entre contratos. Esto quiere decir que las funciones y variables de estado del smart contract se encuentran separadas en diferentes contratos, de forma que queden agrupadas según la funcionalidad que implementan.

Así pues, en la blockchain se despliega un contrato inteligente final (denominado PistaCliente) que contiene la funcionalidad que ha heredado funcionalidad de los anteriores. Una descripción de estos contratos se ofrece a continuación. La descripción más detallada de dichos contratos se incluye en el apéndice A.

- El *primer contrato inteligente*, AccessControl, implementa la funcionalidad necesaria para la creación, otorgación, comprobación y eliminación de roles. Access Control forma parte de la librería de contratos OpenZeppelin, una librería con diferentes tipos de contratos inteligentes considerados estándar, que facilitan el desarrollo seguro en plataformas blockchain ofreciendo soluciones que implementan, por ejemplo, operaciones matemáticas seguras, criptografía, tokens fungibles (moneda digital) y no fungibles (NFTs) y otras utilidades.

- El *segundo contrato inteligente*, PistasRolAdmin es un contrato que hereda de AccessControl y en el que se declara, además del rol administrador que viene por defecto en AccessControl, el rol de gestor y un modifier, onlyGestor, que sirve para que las funciones a las que se le aplica solo se puedan ejecutar cuando el transactor posee el rol Gestor. Un modifier es una función que se añade a las cabeceras de otras funciones para modificar su comportamiento.

- El *tercer contrato inteligente*, PistasGestion, es un contrato que hereda de PistasRolAdmin e implementa la estructura de datos según la que se representarán las pistas en el sistema, y que añade la funcionalidad para modificar su modo de funcionamiento y su creación, que se verá limitada exclusivamente al uso de gestores. Además, implementa otro modifier para comprobar que una pista se encuentra en el estado de funcionamiento común de cara al cliente. Dicho contrato se compone de pistas que a su vez tienen un atributo Estado de tipo enumeración.

- *El contrato final*, PistaCliente, es un contrato que hereda de PistasGestion e incluye funciones para administrar el precio por minuto de las pistas, el tiempo extra de cortesía que se ofrece hasta que se confirma la transacción, la función para reservar la luz con un pago y otra que retira las criptomonedas del smart contract a quien la invoca, que tiene que ser un administrador del contrato.

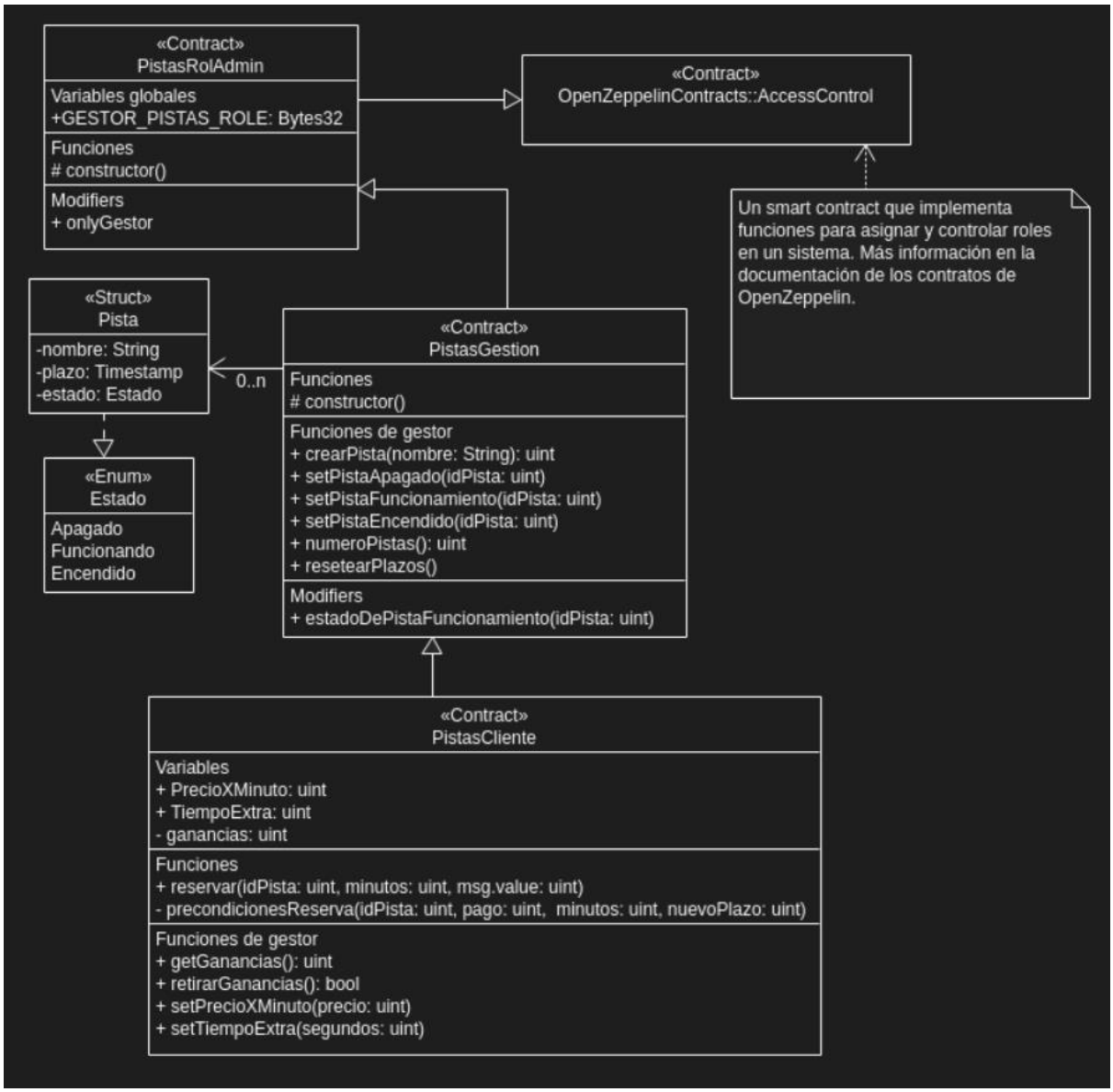

Figura 21. Modelo del contrato inteligente.

5.4 RPI Script

Para que la prueba de concepto sea satisfactoria dentro del sistema diseñado, deben existir dispositivos IoT que se comuniquen con la blockchain y que interactúen con la iluminación de la pista dependiendo del estado almacenado dentro de la blockchain por los contratos inteligentes mostrados en la sección anterior.

Para esta prueba de concepto, se ha utilizado una Raspberry Pi. Dentro de dicha Raspberry Pi, existe una aplicación Javascript muy simple compuesta de un módulo “main” y dos librerías “GPIO Utils.js” y “PistaUtils.js”, cuyo cometido es leer del smart sontract los datos de las pistas que se le indica en el archivo de configuración cada cinco segundos. Según el estado de las pistas leído en dichos smart contracts, se utiliza la librería GPIO.js para enviar una señal digital a través de sus pines, para así encender y apagar las luces de un circuito.

De esta forma, se logra que el dispositivo IoT (una Raspberry Pi en este caso) pueda interactuar con el mundo físico y controlar la iluminación de las pistas según los pagos realizados por los usuarios a través de una aplicación muy sencilla.

5.5 Aplicación Web

Para simular el funcionamiento de un sistema real y comprobar que realmente la tecnología de IoTeX ofrece las herramientas necesarias para realizar la prueba de concepto, es necesario crear un interfaz de usuario que permita tanto a los usuarios como al administrador del contrato interactuar con toda la funcionalidad del contrato. Dicha interfaz de usuario se ha realizado mediante una aplicación web.

Esta aplicación web se ha implementado utilizando Javascript, específicamente el framework frontend React (explicado en la sección 5.2.2), debido a la sencillez a la hora de añadir dinamismo dentro de los componentes de una aplicación web, a mi familiarización con dicho framework y por la necesidad de utilizar un framework basado en aplicaciones web de una sola página, para poder mantener una conexión constante con IoPay, en lugar de tener que estar rehaciendo la conexión en cada recarga de página.

La aplicación web se divide en 5 pantallas o componentes principales, como puede verse en la figura 22: *Índice*, *Tutorial*, *Pistas*, *Reserva*, y *Administración*. Adicionalmente, existe una librería Javascript, *PistaUtils.ts* (una ampliación de la anterior mencionada en la sección 5.4), que permite la comunicación con el smart contract, a través del signer web proporcionado por IoTeX. Este signer web es una interfaz que permite a la aplicación web conectar con IoPay Desktop, para sincronizarse y enviar la información de las transacciones a IoPay, para ser firmadas y enviadas posteriormente.

La forma de comunicarse con el contrato inteligente depende del tipo de función ejecutada: de lectura, donde la petición se hace directamente desde la web; o escritura, donde se envía a IoPay la información para construir la transacción, y posteriormente, ser enviada a través del mismo. La información según la que los componentes visuales muestran información, se recupera mediante las funciones de lectura que se ejecutan en la creación de las vistas que hacen uso del smart contract. Ejemplos de estas son la carga de los roles que pueden tener los usuarios, y el precio de las pistas. Aquellas acciones que modifican el estado del smart contract, las funciones de escritura, son llamadas a través de los eventos desencadenados por los usuarios, como clicar un botón, que como respuesta hacen las llamadas pertinentes a las funciones contenidas en el módulo PistaUtils.ts, que comienzan la comunicación con IoPay y quedan a la espera de una respuesta del mismo que será tratada por el frontend, mostrando un mensaje u otro en función de si el envío ha tenido éxito o no. Ejemplos de estas son las funciones de reservar pista, o cambiar los precios generales.

Una explicación más detallada de dicha aplicación web se incluye en el apéndice C.

Los componentes que conforman la aplicación web son:

1. El *índice* (*índex*), que no implementa ninguna funcionalidad, lleva directamente al tutorial de como empezar a utilizar la aplicación.

2. El *tutorial*, que explica brevemente de donde descargar IoPay, como crear la cartera, configurarla para que utilice la Testnet, y donde conseguir IOTX para la Testnet de forma gratuita (criptomoneda nativa de IoTeX).

En el momento de conexión con la página web se intenta realizar la sincronización con la blockchain, y por tanto, una vez se ha completado los pasos del tutorial y se mantiene una cartera abierta, en la barra de navegación aparecen una o dos pestañas (dependiendo de los privilegios del usuario).

3. La pestaña *Pistas*, que al igual que las siguientes aparece al iniciar la conexión con IoPay, recoge la información de todas las pistas disponibles de la blockchain haciendo uso de la librería PistaUtils.js, y muestra los datos de todas las pistas. En esta información que se muestra se puede ver si se encuentran disponibles o no. En caso de estar disponibles se puede acceder a la siguiente ventana.

4. Cuando se clica en una de las pistas de la pantalla anterior se accede a la pantalla *Reserva*, donde en un formulario se le indica la cantidad de tiempo en minutos que se desea usar. Cuando se clica el botón de reserva la aplicación web realiza la comunicación con IoPay para crear la transacción, y desde IoPay se confirma y se envía.

5. En caso de ser administrador o gestor, aparece la pestaña de *Administración*, desde la cual se pueden modificar los parámetros y roles del contrato inteligente y los estados de las pistas (forma en la que funcionan las pistas: criptomonedas por minuto, encendido o apagado).

Respecto al manejo de usuarios en esta aplicación web no se utilizan credenciales tradicionales. Al contrario, se utiliza la interacción con el smart contract para definir estos roles. Así, la aplicación web considera a un usuario como administrador a aquel que posee el rol administrador dentro del contrato, que podrá realizar las acciones de relacionadas con la gestión de asignación de roles, y retirar las criptomonedas acumuladas en el contrato. Mientras que la figura del gestor corresponde a aquel, que tiene el control sobre el estado y creación de las pistas (el encargado del polideportivo).

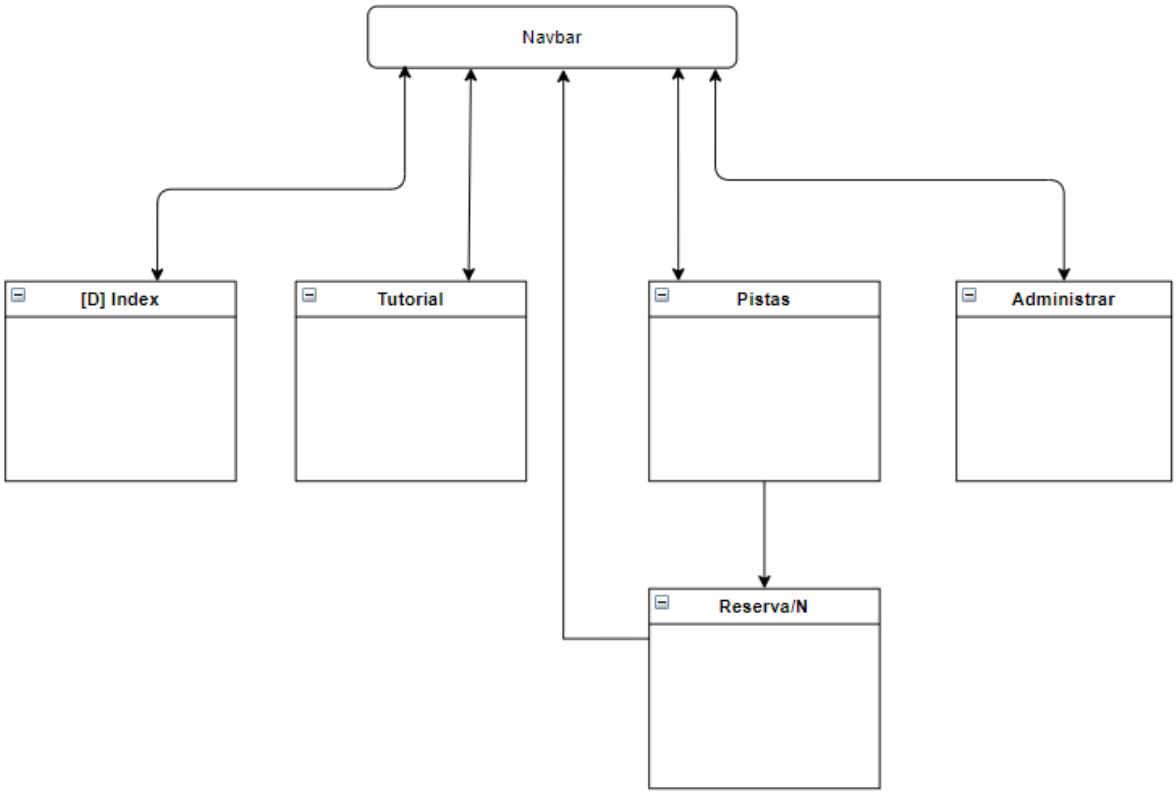

Figura 22. Esquema de navegación de la aplicación web.

5.6 Problemas durante el desarrollo del sistema

En este apartado se comentan los principales contratiempos que se tuvieron durante el desarrollo del sistema, así como la forma en la que dichos contratiempos fueron resueltos.

En primer lugar, cuando se comenzó a desarrollar la aplicación web se encontraron problemas al decidir cómo comenzar el desarrollo, ya que al principio, al probar la aplicación de ejemplo que se muestra en la documentación de IoTeX **[60]**, traía en el proyecto una serie de frameworks de los cuales algunos no eran necesarios para el desarrollo de la prueba de concepto. Al final se decidió empezar un proyecto React base e ir añadiendo los frameworks necesarios, no solo por lo anterior, sino porque además ese ejemplo no funcionaba correctamente en mi móvil.

Otro de los problemas en el uso de IoTeX es la utilización de los signers. IoTeX ofrece un signer web plenamente funcional dentro de la librería Antenna Utils, el cual se utilizó para este proyecto. No obstante, ninguno de los ejemplos del signer móvil llegó a funcionar, aun cuando se contactó con los desarrolladores en los foros correspondientes. Esto es contraproducente, ya que debido a su orientación específica a entornos IoT, IoTeX debería implementar un módulo compatible en los sistemas operativos móviles populares, Android y Apple, para ser más competitiva de cara a los desarrolladores.

Dentro de la realización de la prueba de concepto también surgieron desafíos en el desarrollo de los contratos inteligentes, y en particular con el lenguaje Solidity. Una de las peculiaridades de Solidity es que es un lenguaje de bajo nivel, en el que para realizar operaciones matemáticas es necesario el uso de comprobaciones que se aseguren la robustez del programa (debido a que no comprueba el error por acarreo de bits), OpenZeppelin solventa el problema con la librería SafeMath. Además, en el caso de IoTeX, no permitía extraer el array de structs que contiene la información de las pistas. Dicho problema tiene que ver con la versión del codificador ABI. El ABI (Application Binary Interface) es la manera estándar de interactuar con los contratos inteligentes, desde fuera de la blockchain y desde dentro en la interacción entre contratos. La versión del codificador del ABI estándar no soporta arrays dinámicos ni estructuras anidadas. En el momento de realización del proyecto, la segunda versión se encontraba en fase beta de desarrollo. Actualmente es la que viene por defecto en la versión 0.8 de Solidity. Aun así, tras hacer una prueba y desplegar un contrato inteligente con versión de Solidity 0.6.12 utilizando la segunda versión del codificador ABI (experimental), las funciones con estructuras anidadas en arrays no retornaban ninguna información. La solución dada a este problema fue extraer la información de cada pista una a una.

También cabe mencionar que IoTeX no soporta eventos, como en otras plataformas como QTUM o Ethereum. Sin embargo, Solidity, el lenguaje de programación de contratos inteligentes en IoTeX, trae de forma nativa la posibilidad de emitir eventos al ejecutar las funciones de los contratos inteligentes, aunque IoTeX no la implemente. Si estuviera disponible, sería posible optimizar la forma de sincronizar la Raspberry Pi con las pistas, en lugar de comprobar su estado cada 5 segundos como se encuentra implementado actualmente.

Además, durante el desarrollo del contrato inteligente, surgieron varios problemas para encontrar una versión del compilador compatible con la versión del código. Aunque en Internet se pueden encontrar diversos entornos de desarrollo online, el problema es que el de IoTeX sólo compila hasta la versión 0.5.13, mientras que el código del smart contract se encuentra en la versión 0.6. Al final se utilizó un plugin de Visual Studio Code capaz de compilar cualquier versión de solidity.

Finalmente, respecto al desarrollo de la aplicación, al actualizar las dependencias, Node mostró una alerta sobre una librería criptográfica de curva elíptica que usaba la librería iotex-antenna y se encontraba desactualizada. Este problema es un ejemplo de los desafíos con los que nos encontramos a día de hoy en el uso de código fuente externo, debido a la multitud de dependencias externas que escapan a nuestro control.

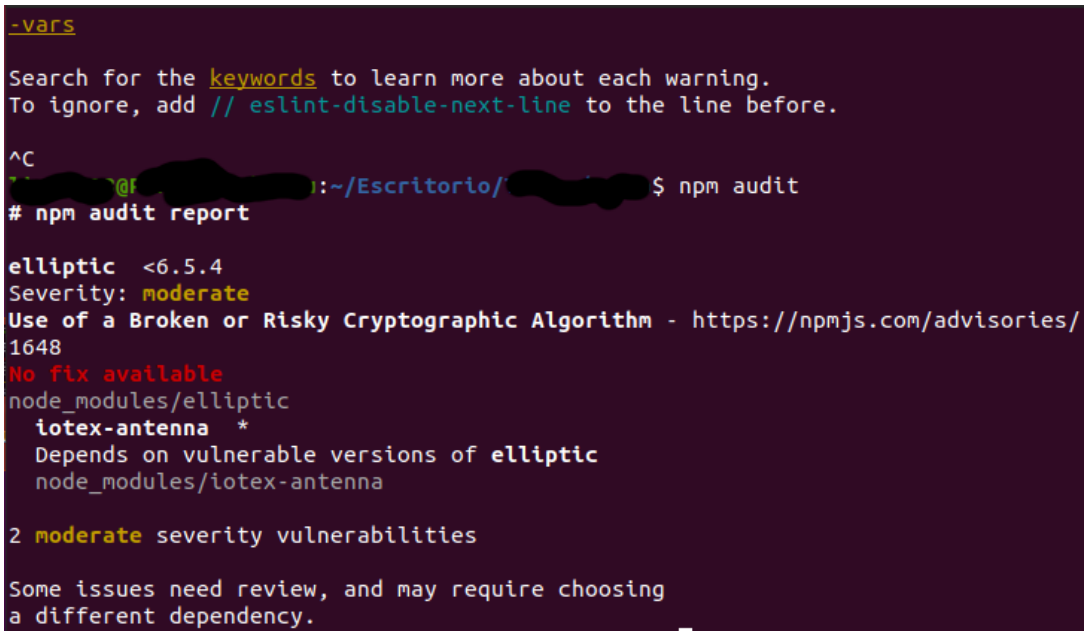

Figura 23. Aviso de la vulnerabilidad criptográfica.

5.7 Análisis final

Con respecto a los objetivos de la prueba de concepto, se ha conseguido implementar un sistema IoT capaz de hacer uso de la funcionalidad inherente a Blockchain y se han cumplido casi todos los objetivos:

- El objetivo de desarrollo de un contrato inteligente capaz funcionar como backend del sistema se ha completado con éxito. En la prueba de concepto dicho contrato admite pagar en criptomonedas IOTX en caso de que las pistas se encuentren en funcionamiento, extraer todas las criptomonedas que recibe y, de una manera sencilla, cambiar los precios y el funcionamiento de las pistas.

- El objetivo de desarrollo de un programa para dispositivos IoT capaz de tener un impacto en el mundo físico también ha sido completado con éxito. En la prueba de concepto se ha desarrollado un programa para Raspberry Pi, que permite leer datos de la blockchain y actuar en base a ellos, para encender las luces de un circuito conectado a sus pines.

- El objetivo que no se ha podido cumplir completamente es el relacionado con el uso del sistema desde un teléfono móvil. Esto ocurre porque dentro del sistema, como se ha descrito en la sección 5.6, en la aplicación web es la que no alcanza los objetivos que se pretendían debido a los problemas de sincronización con IoPay Mobile. Por lo tanto, se dejó de lado la parte enfocada a dispositivos móviles, y por tanto, su diseño responsive. Sin embargo, debido a la existencia de la de la cartera física de IoTeX y la posibilidad de generar y cargar una cartera a partir de 12 palabras, el sistema no queda totalmente inutilizado si se desbloquea la cartera en un ordenador dentro de la institución que utilice un sistema similar.

6

Conclusiones

En este apartado se detallarán las conclusiones obtenidas de la realización del proyecto.

A lo largo del desarrollo del análisis se ha ido desvelando que una de las blockchains que expresan ser específicas para el entorno IoT, IOTA, no tiene la capacidad técnica que afirma tener, con numerosos estudios y pruebas en su contra. Respecto a otras blockchains que teóricamente han sido diseñadas explícitamente para la IoT, encontrar pruebas a favor o en contra de sus afirmaciones ha sido una tarea laboriosa si no imposible debido a su inexistencia, a excepción de QTUM. En este aspecto, opino que deberían ser las propias empresas o equipos de desarrollo quienes deberían ser capaces de demostrar con pruebas lo que afirman sobre su DLT, por competitividad frente al resto y para generar confianza con el consumidor de su producto.

Tras el estudio realizado a nivel teórico, se puede afirmar que el principal problema del uso de la tecnología Blockchain en el entorno IoT es su escalabilidad. Sin embargo, la escalabilidad de las soluciones estudiadas no es especialmente sobresaliente. Y por ello no tiene cabida el que se vendan como específicas para IoT, sino mejores que otras en cuanto a escalabilidad y características específicas. Precisamente, otras soluciones Blockchain que no se venden como tales, como Ethereum, también están trabajando para que en un futuro su DLT alcance altos ratios de transacciones por segundos tras implementar soluciones que mejoren su escalabilidad.

La conclusión de la prueba de concepto con IoTeX es que se pueden realizar aplicaciones IoT sin mucha dificultad, capaces de ser controladas a través de una aplicación cartera (o “wallet”) de ordenador, y ejecutando la funcionalidad desde la cartera de forma manual o utilizando una interfaz de usuario que construya las transacciones. No obstante, las tecnologías blockchain que expresan ser diseñadas específicamente para

entornos IoT aún no son lo suficientemente maduras, y existen mejoras que deberían ser consideradas en el ecosistema de herramientas y aplicaciones.

La tecnología Blockchain se popularizó en 2008 con Bitcoin, y ha ido evolucionando constantemente. Sin embargo, tras los avances realizados hasta la actualidad sigue siendo una tecnología inmadura a la que aún le quedan por mejorar varios aspectos, como su escalabilidad limitada, para conseguir procesar y almacenar un alto ratio de datos; y el ecosistema de herramientas y aplicaciones que la envuelven. Entre estos se incluyen las aplicaciones que van dirigidas a la interacción entre la blockchain y el usuario final, la optimización en la transferencia de datos desde un contrato inteligente a un dispositivo e incluso la actualización de los entornos de desarrollo online de Smart Contract, destacados por las propias empresas blockchain estudiadas.

Referencias

Referencias principales

[1] Rodriguez, Nelson. (23 de julio de 2019). *Uso De Blockchain: Lista De 20+ Casos De Uso De La Tecnología Blockchain*. 101 Blockchains.

https://101blockchains.com/es/uso-de-blockchain/

[2] Telefónica. (s.f.). *¿Qué es TrustOS?*.

https://blockchain.telefonica.com/soluciones-para-tu-negocio/trustos/

[3] ESHORIZONTE2020. (s.f.). *¿Qué es Horizonte 2020?.*

https://eshorizonte2020.es/que-es-horizonte-2020

[4] Cuen, Leigh. (25 de febrero de 2020). *IOTA Being Shut Off Is the Latest Chapter in an Absurdist History*. CoinDesk.

https://www.coindesk.com/iota-being-shut-off-is-the-latest-chapter-in-an-absurdist-history

[5] Colaboradores de Wikipedia. (s.f.). *Distributed Ledger Technology (DLT)*. Wikipedia. Recuperado el 16 de septiembre de 2021 (última fecha de edición).

https://es.wikipedia.org/wiki/Distributed_Ledger_Technology_(DLT)

[6] Tena, María. (21 de junio de 2017). *Siete retos regulatorios a los que se enfrenta blockchain*. BBVA.

https://www.bbva.com/es/siete-retos-regulatorios-los-se-enfrenta-blockchain/

[7] Yaga, Dylan., Mell, Peter., Roby, Nik., y Scarfone, Karen. (Octubre de 2018). Blockchain Technology Overview. National Institute of Standards and Technology (NIST).

https://nvlpubs.nist.gov/nistpubs/ir/2018/NIST.IR.8202.pdf

[8] Sullivan, James. (Diciembre de 2019). *Learning Path: Introduction to Blockchain Technology*. O'Reilly.

https://learning.oreilly.com/learning-paths/learning-path-introduction/0636920327844/

[9] Gaurav. (11 de agosto de 2021). *Permissionless Blockchain vs Permissioned Blockchain*. CoinCodeCap.

https://blog.coincodecap.com/permissionless-and-permissioned-blockchain

[10] Rodriguez, Nelson. (20 de septiembre de 2018). *Algoritmos De Consenso: La Raíz De La Tecnología Blockchain*. 101 Blockchains.

https://101blockchains.com/es/algoritmos-de-consenso-blockchain/#1

[11] Ortiz, Joanybel. (s.f.). *¿Es el Blockchain el libro de contabilidad del mundo 2.0?*.

http://www.joanybelortiz.com/blockchain-el-libro-de-contabilidad-del-mundo-2-0/

[12] Criptotario. (s.f.). *PRUEBA DE TRABAJO VS PRUEBA DE PARTICIPACIÓN*.

https://criptotario.com/prueba-de-trabajo-vs-prueba-de-participacion

[13] Bit2Me. (s.f.). *¿Qué es PoA (Proof of Authority – Prueba de Autoridad)?*.

https://academy.bit2me.com/que-es-proof-of-authority-poa/

[14] Seffield, Nolan. (19 de noviembre de 2018). *pBFT— Understanding the Consensus Algorithm*. Medium.

https://medium.com/coinmonks/pbft-understanding-the-algorithm-b7a7869650ae

[15] OMG. (s.f.). *OMG Network Documentation.*

https://docs.omg.network/

[16] Cant, Joeri. (5 de octubre de 2019). *Walmart utiliza tecnología blockchain para rastrear las cadenas de suministro de camarones*. Cointelegraph.

https://es.cointelegraph.com/news/walmart-uses-blockchain-tech-to-track-shrimp-supply-chains

[17] Civic Ledger. (s.f.).

https://civicledger.com/

[18] Hyperledger. (s.f.). *Hyperledger Indy*.

https://www.hyperledger.org/use/hyperledger-indy#:~:text=Hyperledger%20Indy%20provides%20tools%2C%20libraries,applications%2C%20and%20any%20other%20silo

[19] Polymath. (s.f.).

https://polymath.network/

[20] Harbor. (s.f.).

https://harbor.com/

[21] Mathis, Mark. (21 de diciembre de 2018). *A Security Token — Harbor's R-Token*. Medium.

https://medium.com/coinmonks/a-security-token-harbors-r-token-c147ba9557b4

[22] Gridplus. (s.f.).

https://gridplus.io/

[23] Damiani, Jesse. (6 de noviembre de 2017). *SimplyVital Health Is Using Blockchain To Revolutionize Healthcare*. Forbes.

https://www.forbes.com/sites/jessedamiani/2017/11/06/simplyvital-health-blockchain-revolutionize-healthcare/?sh=7df898b0880a

[24] Team Tokens25. (18 de abril de 2018). *¿Qué es MediBloc?*. Tokens 24.

https://www.tokens24.com/es/cryptopedia/coin-guides/que-es-medibloc

[25] Lopez Research. (Noviembre de 2013). *An Introduction to the Internet of Things (IoT)*. CISCO.

https://www.cisco.com/c/dam/en_us/solutions/trends/iot/introduction_to_IoT_november.pdf

[26] Fernandez Cejas, Miguel. (15 de noviembre de 2017). *IoT: ¿Cuáles son sus componentes principales?*. ITop.

https://www.itop.es/blog/item/iot-cuales-son-sus-componentes-principales-y-aplicaciones.html

[27] Colaboradores de Wikipedia. (s.f.). *LoRaWAN* . Wikipedia. Recuperado el 28 de marzo de 2021 (última fecha de edición).

https://es.wikipedia.org/wiki/LoRaWAN

[28] Telefónica. (3 de julio de 2020). *Telefónica consolida el mercado IoT en España con más de 2,6 millones de líneas y despliegues en todo el país*.

https://www.telefonica.com/es/web/sala-de-prensa/-/telefonica-consolida-el-mercado-iot-en-espana-con-mas-de-2-6-millones-de-lineas-y-despliegues-en-todo-el-pais

[29] Aprendiendo Arduino. (s.f.). *Dispositivos Hardware IoT*. Wordpress.

https://aprendiendoarduino.wordpress.com/2018/11/14/dispositivos-hardware-iot/

[30] Satoshi. (27 de noviembre de 2017). *¿Qué es un SoftPLC?*. Opiron.

https://www.opiron.com/2017/11/27/que-es-un-softplc-ventajas/

[31] García, Jose. (12 de julio de 2019). *El mejor PLC para llevar Internet a toda la casa: guía de compra y comparativa*. Xataka.

https://www.xataka.com/perifericos/mejor-amplificador-wifi-plc-guia-compra-comparativa

[32] Airbus. (12 de mayo de 2021). *Delivering Internet of Things (IoT) services worldwide*.

https://www.airbus.com/newsroom/news/en/2021/05/Delivering-Internet-of-Things-IoT-services-worldwide.html

[33] ABB. (s.f.). *Industrial IoT applications*.

https://new.abb.com/control-systems/features/industrial-IoT-services-people-use-cases

[34] Middlebay. (s.f.). *Powerhouse Dynamics*.

https://www.middleby.com/brands/powerhouse-dynamics/

[35] Infiswift. (s.f.). *Agriculture*.

https://infiswift.tech/agriculture/

[36] Libelium. (16 de octubre de 2012). *Smart Cars: a practical implementation of M2M communications is becoming a reality ever closer*.

https://www.libelium.com/libeliumworld/smart_cars_m2m_accident_prevention/

[37] Blueriiot. (s.f.).

https://www.blueriiot.com/eu-es

[38] Del Valle Hernández, Luis. (s.f.). *#107 Aplicaciones del IoT usos prácticos en el mundo real*. Programar Facil.

https://programarfacil.com/podcast/aplicaciones-del-iot-reales/#Dispositivos_inteligentes

[39] iagua. (4 de octubre de 2016). *Gestagua desarrolla un nuevo sistema de gestión para contadores inteligentes de agua*.

https://www.iagua.es/noticias/espana/gestagua/16/10/03/gestagua-desarrolla-nuevo-sistema-gestion-contadores-inteligentes

[40] Libelium. (17 de febrero de 2021). *Libelium participates in an ambitious air quality IoT project for the port of Genoa, Italy*.

https://www.libelium.com/success-stories/libelium-participates-in-an-ambitious-air-quality-iot-project-for-the-port-of-genoa-italy/

[41] Redacción Computing. (6 de junio de 2019). *IoT e IA mejoran la gestión ambiental de las ciudades*. Computing.

https://www.computing.es/mercado-ti/casos-exito/1112377046401/iot-ia-mejoran-gestion-ambiental-de-ciudades.1.html

[42] Dobbins, Roland. (26 de octubre de 2016). *Mirai IoT Botnet Description and DDoS Attack Mitigation*. NETSCOUT.

https://www.netscout.com/blog/asert/mirai-iot-botnet-description-and-ddos-attack-mitigation

[43] ethos.dev. (23 de mayo de 2020). *The Beacon Chain Ethereum 2.0 explainer you need to read first*.

https://ethos.dev/beacon-chain/

[44] Shah, Hardik. (6 de enero de 2019). *Building Industrial IoT with IOTA: Introduction and How IOTA Works*. Simform.

https://www.simform.com/industrial-iot-iota-part-1/

[45] IoTeX Foundation. (19 de septiembre de 2019). *The Internet of Trusted Things*. IoTeX.

https://iotex.io/blog/internet-of-trusted-things-possibilities/

[46] Willemse, Linda. (3 de febrero de 2019). *Project overview: IoTeX the Decentralized Network for the Internet of Things (IoT)*. Hacker Noon.

https://hackernoon.com/project-overview-iotex-the-decentralized-network-for-the-internet-of-things-iot-b35f6498b765

[47] Mitra, Rajarshi. (25 de marzo de 2020). *What is QTUM? [The Most Comprehensive Guide] - Blockgeeks*. Blockgeeks.

https://blockgeeks.com/guides/what-is-qtum/#Qtum_and_Proof_of_Stake_(POS)

[48] Ricottone, Alessandro. (2 de febrero de 2020). *Introduce Merkle Trees and Inclusion Proofs*. Lisk.

https://research.lisk.io/t/introduce-merkle-trees-and-inclusion-proofs/213

[49] Muratov, Fedor., Lebedev, Andrei., Iushkevich, Nikolai., Nasrulin, Bulat., y Takemiya, Makoto. (3 de septiembre de 2018). *YAC: BFT Consensus Algorithm for Blockchain*. Arxiv.

https://arxiv.org/pdf/1809.00554.pdf

[50] Hyperledger Iroha. (s.f.). *Overview of Iroha*. Iroha Readthedocs. Recuperado el 17 de julio de 2020.

https://iroha.readthedocs.io/en/main/overview.html

[51] Regueiro, Mercedes. (7 de agosto de 2018). *La movilidad de Hyperledger Iroha (いろは).* Medium.

https://medium.com/@M.R.M./la-movilidad-de-hyperledger-iroha-%E3%81%84%E3%82%8D%E3%81%AF-e8a2204a098f

[52] Etiemble, Daniel. (19 de agosto de 2019). *Ternary circuits: why R = 3 is not the Optimal Radix for Computation*. Arxiv.

https://arxiv.org/pdf/1908.06841.pdf

[53] Fuentes Contreras, Alberto. (Junio de 2019). *Benchmarking of Blockchain Technologies used in a Decentralized Data Marketplace*. Universidad Politécnica de Madrid (UPM).

http://oa.upm.es/55775/1/TFG_ALBERTO_FUENTES_CONTRERAS.pdf

[54] Radix Blog. (9 de marzo de 2018). Why DAGs Don't Scale Without Centralization. RadixDLT.

https://www.radixdlt.com/post/dags-dont-scale-without-centralization

[55] Popov, Sergei. (30 de abril 2018). *The Tangle*. CTF Assets.

https://assets.ctfassets.net/r1dr6vzfxhev/2t4uxvsIqk0EUau6g2sw0g/45eae33637ca92f85dd9f4a3a218e1ec/iota1_4_3.pdf

[56] Narula, Neha. (7 de septiembre de 2017). *Cryptographic vulnerabilities in IOTA*. Medium.

https://medium.com/@neha/cryptographic-vulnerabilities-in-iota-9a6a9ddc4367

[57] IoTeX Team. (16 de noviembre de 2018). *IoTeX Mainnet Preview (Photon) Release*. Medium.

https://medium.com/@iotex/iotex-mainnet-preview-photon-release-6c316da17fbb

[58] Unita. (22 de enero de 2019). *QtumX Reaches 10,000 TPS in Benchmark Tests*. Qtum Blog.

https://blog.qtum.org/qtumx-reaches-10-000-tps-in-benchmark-tests-cee6452166fd

[59] Techracers. (22 de junio de 2018). *Things you should know about QTUM Blockchain*. Medium.

https://medium.com/techracers/things-you-should-know-about-qtum-blockchain-23b0f63a3ea1

[60] IoTeX. (s.f.). *IoTeX dApp Sample*. Documentación de IoTeX. Recuperado el 17 de septiembre de 2021.

https://docs.iotex.io/get-started/iotex-dapp-starter#get-started

Otras referencias

1 DLTs

- Banco de España. (16 de octubre de 2018). *BOLETÍN ECONÓMICO 4/2018, ARTÍCULOS ANALÍTICOS*. Banco de España.

  https://www.bde.es/f/webbde/SES/Secciones/Publicaciones/InformesBoletinesRevistas/ArticulosAnaliticos/2018/T4/descargar/Fich/beaa1804-art26.pdf

- Colaboradores de Wikipedia. (s.f.). *Doble gasto*. Wikipedia. Recuperado el 27 de enero de 2021 (última fecha de edición).

  https://es.wikipedia.org/wiki/Doble_gasto#:~:text=El%20doble%20gasto%20es%20un,gastarse%20m%C3%A1s%20de%20una%20vez.&text=En%20el%20caso%20concreto%20de,la%20blockchain%20y%20verific%C3%A1ndola%20despu%C3%A9s

- Rodriguez, Nelson. (31 de enero de 2019). *Tecnología De Registro Distribuido: Donde La Revolución Tecnológica Comienza*. 101 Blockchains.

  https://101blockchains.com/es/tecnologia-de-registro-distribuido-dlt/

- Madrid, Abelardo. (16 de agosto de 2020). *¿Qué es y cómo comprar Radix (EXRD)?* Toda la información. Bitcoin.es.

  https://bitcoin.es/actualidad/radix-el-sucesor-de-blockchain/#Tempo_y_la_muerte_de_blockchain

2 Blockchain

- EFPA España. (3 de diciembre de 2019). *Blockchain, mucho más que la tecnología implícita de Bitcoin. Aplicaciones en las finanzas.* Asesores financieros EFPA.

  https://www.asesoresfinancierosefpa.es/opinion-financiera/blockchain-2/

- Calvo, Mar. (28 de julio de 2018). *Conoce los diferentes tipos de blockchain*. Blockchain Services.

  http://www.blockchainservices.es/novedades/conoce-los-diferentes-tipos-de-blockchain/

- Vazquez, Antonio. (17 de octubre de 2018). *El minado en Blockchain ¿Quiénes son y qué hacen los mineros?*. CYSAE.

  https://www.cysae.com/el-minado-en-blockchain/

- Bit2Me. (s.f.). *¿Qué es un bloque en blockchain?*. Bit2Me Academy.

  https://academy.bit2me.com/que-es-un-bloque-dentro-de-la-blockchain/

- Santander Global Tech. (12 de julio de 2019). *Blockchain: sobre confianza y certeza. La verdad sobre los consensos*.

  https://santanderglobaltech.com/concepto-consenso-blockchain-relevancia-consecuencias/

- Camargo, Federico. (s.f.). *¿Cuáles son las ventajas y desventajas de Blockchain?*. Camargo Life.

  https://camargo.life/blockchain-ventajas-y-desventajas/

- Ramsey, Bradley. (20 de junio de 2018). *The Impressive Hardware Used in Cryptocurrency Mining*. Medium.

  https://medium.com/supplyframe-hardware/the-impressive-hardware-used-in-cryptocurrency-mining-31771edb857c

- Antminer. (s.f.). *Antminer S9*. Amazón.

  https://www.amazon.es/Antminer-S9-13-5TH-APW3-PSU/dp/B01MCZVPFE

- Massessi, Demiro. (12 de diciembre de 2018). *Public Vs Private Blockchain In A Nutshell*. Medium.

  https://medium.com/coinmonks/public-vs-private-blockchain-in-a-nutshell-c9fe284fa39f

- Colaboradores de Wikipedia. (s.f.). *Problema de los generales bizantinos. Wikipedia*. Recuperado el 27 de enero de 2020 (última fecha de edición).

  https://es.wikipedia.org/wiki/Problema_de_los_generales_bizantinos

- IBM. (s.f.). *What is blockchain technology?*.

  https://www.ibm.com/topics/what-is-blockchain

3 Redes Capa-2

- Cryptopedia Staff. (11 de agosto de 2021). *Layer-1 and Layer-2 Blockchain Scaling Solutions*. Gemini.

  https://www.gemini.com/cryptopedia/blockchain-layer-2-network-layer-1-network#section-boosting-blockchain-networks-scalability

- Blanco Crespo, Luis Jesús. (25 de abril de 2021). *Una aproximación a las soluciones de Capa 2 en la red Ethereum (ETH) y su importancia*. BeInCrypto.

  https://es.beincrypto.com/aproximacion-soluciones-capa-2-red-ethereum-eth-importancia/

4 IoT

- Chandrashekhar, Kavya. (20 de septiembre de 2016). *Internet of Things (IoT) Characteristics*. Linkedin.

  https://www.linkedin.com/pulse/internet-things-iot-characteristics-kavyashree-g-c/

- K Patel, Keyur., y M Patel, Sunil. (Mayo de 2016). *Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges*. Maharaja Sayajirao University of Baroda. Research Gate.

  https://www.researchgate.net/publication/330425585_Internet_of_Things-IOT_Definition_Characteristics_Architecture_Enabling_Technologies_Application_Future_Challenges

- CepymeNews. (29 de junio de 2018). *Cómo cambiará el mundo la quinta generación digital 5G*.

  https://cepymenews.es/como-cambiara-mundo-quinta-generacion-digital-5g

- Cárdenas, Alvaro. (28 de noviembre de 2016). *¿Qué es una plataforma IoT?*. Secmotic.

  https://secmotic.com/plataforma-iot/

- I-Scoop. (s.f.). *Making sense of IoT (Internet of Things) – the IoT business guide*.

  https://www.i-scoop.eu/internet-of-things/

- Colaboradores de Wikipedia. (s.f.). *Sensor node*. Wikipedia.

  https://en.wikipedia.org/wiki/Sensor_node

- Ovacen. (s.f.). *Internet de las cosas; Qué es y cuáles son sus ventajas y desventajas*.

  https://ovacen.com/internet-de-las-cosas/

- Gracia, María. (s.f.). *IoT - Internet Of Things*. Deloitte.

  https://www2.deloitte.com/es/es/pages/technology/articles/IoT-internet-of-things.html

- ABDC. (2 de julio de 2019). *EL INTERNET DE LAS COSAS*.

  https://abdc.es/blog/internet-de-las-cosas-ventajas-desventajas/

- Codigonexo. (s.f.). *Ventajas y desventajas del Internet de las Cosas*.

  https://www.codigonexo.com/blog/nfc/internet-de-las-cosas/ventajas-e-inconvenientes-del-internet-las-cosas/

- Biplaza. (6 de junio de 2018). *VENTAJAS DEL IOT, INTERNET OF THINGS, EN LOS NEGOCIOS*.

  https://www.biplaza.es/ventajas-del-iot-internet-of-things-los-negocios/

- Skaldion. (16 de enero de 2018). *Ventajas y desventajas del IoT*.

  http://skaldion.com/2018/01/ventajas-y-desventajas-del-iot/

- Franco, Rolando. (s.f.). *Internet de las Cosas*. Universidad Católica Nuestra Señora de la Asunción. JeuAzarro.

  http://jeuazarru.com/wp-content/uploads/2016/11/IoT.pdf

- HPE. (s.f.). *¿QUÉ ES EL INTERNET DE LAS COSAS INDUSTRIAL (IIOT)?*.

  https://www.hpe.com/es/es/what-is/industrial-iot.html

- Paneles ACH. (s.f.). *¿QUÉ SON LAS SMART CITIES O CIUDADES INTELIGENTES?*.

  https://www.panelesach.com/blog/smart-cities-o-ciudades-inteligentes-que-son/

- Fracttal. (10 de octubre de 2018). *Las 9 aplicaciones más importantes del Internet de las Cosas (IoT)*.

  https://www.fracttal.com/blog/2018/10/10/9-aplicaciones-importantes-iot

- Redacción Computing. (6 de junio de 2019). *IoT e IA mejoran la gestión ambiental de las ciudades*. Computing.

  https://www.computing.es/mercado-ti/casos-exito/1112377046401/iot-ia-mejoran-gestion-ambiental-de-ciudades.1.html

- eFor. (s.f.). *Tecnologías de comunicación para el IoT*.

  https://www.efor.es/sites/default/files/tecnologias-de-comunicacioon-para-iot.pdf

- Aprendiendo Arduino. (7 de marzo de 2018). *Ultra Narrow Band (UNB)*.

  https://www.aprendiendoarduino.com/tag/unb/#:~:text=Ultra%20Narrow%20Band%20(UNB)%20generalmente,el%20transmisor%20y%20el%20receptor.

- Bluetooth. (s.f.). *Mesh Networking*.

  https://www.bluetooth.com/learn-about-bluetooth/bluetooth-technology/mesh/

- Wilson, Richard. (10 de octubre de 2014). *The pros and cons of Bluetooth Low Energy*. ElectronicsWeekly.

  https://www.electronicsweekly.com/news/design/communications/pros-cons-bluetooth-low-energy-2014-10/

- Epic. (s.f.). *Internet of Things (IoT)*.

  https://epic.org/privacy/internet/iot/

5 Ethereum

- il3ven. (12 de septiembre de 2021). *ETHEREUM DEVELOPMENT DOCUMENTATION*. Ethereum Docs.

  https://ethereum.org/en/developers/docs/

- Ethereum. (s.f.). *Upgrading Ethereum to radical new heights*.

  https://ethereum.org/en/eth2/

- CoinMarketCap. (s.f.). *What Is the Raiden Network?*.

  https://coinmarketcap.com/alexandria/glossary/raiden-network

6 IOTA

- V. Hauge, Bjorn. (21 de febrero de 2018). *Major pros and cons of the crypto coin IOTA*. Medium.

  https://medium.com/@bjornvhauge/major-pros-and-cons-of-the-crypto-coin-iota-11b716c200ee

- Johnson, Nick. (26 de septiembre de 2017). *Why I find Iota deeply alarming*. Hacker Noon.

  https://hackernoon.com/why-i-find-iota-deeply-alarming-934f1908194b

- Kurokawa, Kay. (9 de febrero de 2018). *IOTA Doesn't Scale*. Medium.

  https://medium.com/@kaykurokawa/iota-doesnt-scale-fff54f56e975

7 IoTeX

- Credentials Community Group (W3C). (29 de diciembre de 2020). *A Primer for Decentralized Identifiers*. W3C Credentials Community Group.

  https://w3c-ccg.github.io/did-primer/#how-dids-differ-from-other-globally-unique-identifiers

- Guilbon, Joffrey. (19 de junio de 2018). *Introduction to Trusted Execution Environment: ARM's TrustZone*. Quarkslab's Blog.

  https://blog.quarkslab.com/introduction-to-trusted-execution-environment-arms-trustzone.html

- Willemse, Linda. (3 de febrero de 2019). *Project overview: IoTeX the Decentralized Network for the Internet of Things (IoT).* Hacker Noon.

  https://hackernoon.com/project-overview-iotex-the-decentralized-network-for-the-internet-of-things-iot-b35f6498b765

- Dant. (17 de septiembre de 2018). *IoTeX - Infraestructura Blockchain centrada en la Privacidad y Escalabilidad del IoT*. ForoBits.

  https://forobits.com/t/iotex-infraestructura-blockchain-centrada-en-la-privacidad-y-escalabilidad-del-iot/21396

- Sun Star. (13 de abril de 2018). *IoTex:Conencting the physical world block by block*. Medium.

  https://medium.com/@zk19657543/iotex-conencting-the-physical-world-block-by-block-653d8a268606

- IoTeX Team. (16 de noviembre de 2018). *IoTeX Mainnet Preview (Photon) Release*. Medium.

  https://medium.com/@iotex/iotex-mainnet-preview-photon-release-6c316da17fbb

- Binance. (29 de mayo de 2019). *A Decentralized Network For Internet Of Things Powered By A Privacy-Centric Blockchain*. Research Binance.

  https://research.binance.com/en/projects/iotx

- IoTeX Team. (16 de abril de 2019). *Everything You Need to Know About IoTeX Mainnet Alpha*. Medium.

  https://medium.com/@iotex/everything-you-need-to-know-about-iotex-mainnet-alpha-b8d790e0bd55

- Takahashi, Dean. (6 de enero de 2020). *IoTex's Ucam is a blockchain-based, encrypted private home camera*. Venture Beat.

  https://venturebeat.com/2020/01/06/iotexs-ucam-is-a-blockchain-based-encrypted-private-home-camera/#:~:text=Silicon%20Valley%20startup%20IoTex%20has,encryption%20to%20ensure%20user%20privacy.&text=The%20device%20might%20be%20just,everyday%20objects%20smart%20and%20connected.

- Sporny, Manu., Longley, Dave., Sabadello, Markus., Reed, Drummond., Steele, Orie., y Allen, Christopher. (3 de agosto de 2021). *Decentralized Identifiers*. World Wide Web Consortium. W3.org.

  https://www.w3.org/TR/did-core/#introduction

8 QTUM

- Frumkin, Daniel., Dean, Alex., y Shaddox, Thomas. (s.f.). *Nothing-at-stake problem*. Golden.

  https://golden.com/wiki/Nothing-at-stake_problem#:~:text=Nothing%2Dat%2Dstake%20is%20a,system%20more%20vulnerable%20to%20attacks.

- Colaboradores de Wikipedia. (s.f.). *Prueba de participación*. Wikipedia. Recuperado el 6 de junio de 2021 (última fecha de edición).

  https://es.wikipedia.org/wiki/Prueba_de_participaci%C3%B3n

- Techracers. (22 de junio de 2018). *Things you should know about QTUM Blockchain*. Medium.

  https://medium.com/techracers/things-you-should-know-about-qtum-blockchain-23b0f63a3ea1

- Colaboradores de Wikipedia. (s.f.). *Lightning Network*. Wikipedia. Recuperado el 29 de agosto de 2021 (última fecha de edición).

  https://es.wikipedia.org/wiki/Lightning_Network

- Buck, Jon. (28 de diciembre de 2017). *Escalabilidad, privacidad y gobernabilidad: principales problemas de las DApps, dice el cofundador de Qtum*. Cointelegraph.

  https://es.cointelegraph.com/news/scalability-privacy-and-governance-main-problems-for-dapps-says-qtum-co-founder

- O'Neal, Stephen. (23 de enero de 2019). *La competencia en curso de las blockchains por las transacciones por segundo ¿Quién lo escala mejor?*. Cointelegraph.

  https://es.cointelegraph.com/news/who-scales-it-best-inside-blockchains-ongoing-transactions-per-second-race

- Mundo Criptomonedas. (s.f.). *Qué es SegWit (Segregated Witness)*.

  https://www.mundocriptomonedas.org/que-es-segwit/#:~:text=SegWit%20es%20una%20actualizaci%C3%B3n%20de,Bitcoin%20en%20Diciembre%20de%202015.

- Qtum. (16 de junio de 2020). *Qtum Security Audit Confirmation by Trail of Bits Security Company*. Blog Qtum.

  https://blog.qtum.org/qtum-security-audit-confirmation-by-trail-of-bits-security-company-375b324277cc

- Posnak, Ed. (7 de abril de 2018). *On the Origin of Qtum*. Medium.

  https://medium.com/on-the-origin-of-smart-contract-platforms/on-the-origin-of-qtum-5f2e6daf798a

- BountyX. (Septiembre de 2021). *Qtum revela la estrategia "Go-Mobile" para Smart Contracts y IoT*.

  https://es.bountyx.com/qtum-reveals-go-mobile-strategy-for-smart-contracts-and-iot-2407

- Qtum. (15 de agosto de 2019). *Thinking About Blockchain Privacy: Confidential Assets on the Blockchain*. Blog Qtum.

  https://blog.qtum.org/thinking-about-blockchain-privacy-confidential-assets-on-the-blockchain-8d5218888d07

- Qtum. (27 de marzo de 2020). *Qtum Phantom Protocol: Unveiling Layer-1 Privacy*. Blog Qtum.

  https://blog.qtum.org/qtum-phantom-protocol-unveiling-layer-2-privacy-2d31c8a527a2#:~:text=On%20February%2013%2C%202020%2C%20Qtum,few%20privacy%20solutions%20in%20China

- Bit2Me. (s.f.). *¿Qué son las pruebas zk-SNARK?*. Bit2Me Academy.

  https://academy.bit2me.com/que-son-las-pruebas-zk-snark/

9 HyperLedger Iroha

- Takeshi Miyamae, Takeo Honda., Masahisa Tamura., y Motoyuki Kawaba. (s.f.). Performance Improvement of the Consortium Blockchain for Financial Business Applications. P2PFisy.

  https://www.p2pfisy.com/wp-content/uploads/2019/04/Takeshi_Miyamae.pdf

- Bc. Dávid Urbančok. (Otoño de 2019). *Blockchain open-source software comparison*. Masaryk University Faculty of Informatics (MUNI).

  https://is.muni.cz/th/qr98z/thesis.pdf

- Palanivel, Chandrasekaran. (30 de abril de 2018). *Hyperledger Iroha - Architecture, Functional/Logical Flow & Consensus(YAC) Mechanism*. Linkedin.

  https://www.linkedin.com/pulse/hyperledger-iroha-architecture-functionallogical-chandrasekaran/

10 Tecnologías y herramientas utilizadas

- Lucas, Jesús. (4 de septiembre de 2019). *Qué es NodeJS y para qué sirve*. Open Web Binars.

  https://openwebinars.net/blog/que-es-nodejs/

- Node.Js. (s.f.). *Acerca de Node.js*.

  https://nodejs.org/es/about/

- Colaboradores de Wikipedia. (s.f.). React. Wikipedia. Recuperado el 9 de septiembre de 2021 (última fecha de actualización).

  https://es.wikipedia.org/wiki/React

- React. (s.f). *Página principal*. https://es.reactjs.org/

- Colaboradores de wikipedia. (s.f.). *Bootstrap (framework).* Wikipedia. Recuperado el 17 de agosto de 2021 (última fecha de actualización).

  https://es.wikipedia.org/wiki/Bootstrap_(framework)

- Visual Studio Code. (s.f.). *Getting Started*.

  https://code.visualstudio.com/docs

Apéndice A

Smart Contract

En este apéndice se explicarán en detalle los distintos contratos inteligentes que conforman el que se despliega en la Testnet de IoTeX (mencionados en la sección 5.3)

A.1 PistasRolAdmin

Es el contrato inteligente en el que se declara el rol Gestor. Y en el que se añade el modifier onlyGestor, que impedirá el funcionamiento de algunas funciones si el emisor de la transacción no posee dicho rol. Un modifier es una función que se añade a las cabeceras de otras funciones para modificar su comportamiento. El constructor del contrato es una función que se ejecuta cuando este es desplegado. En este caso se le asignan todos los roles a la dirección que despliega el contrato.

El rol DEFAULT_ADMIN_ROLE es el rol superior por defecto en AccessControl, este permite asignar, crear, eliminar y establecer jerarquías entre roles. Por tanto, las direcciones que poseen este rol poseen acceso a todos los permisos del contrato, una vez se los otorgan a sí mismos.

```
contract PistasRolAdmin is AccessControl {

    bytes32 public constant GESTOR_PISTAS_ROLE = keccak256("Gestor");

    modifier onlyGestor {
        require(hasRole(GESTOR_PISTAS_ROLE, msg.sender));
        _;
    }

    constructor() public {
        _setupRole(DEFAULT_ADMIN_ROLE,msg.sender);
        _setupRole(GESTOR_PISTAS_ROLE,msg.sender);
    }
}
```

Figura 24. Contrato PistasRolAdmin.

A.2 PistasGestion

Este smart contract se utiliza para definir el funcionamiento de las pistas, su creación y otras funciones de lectura que facilitan la programación en el lado del cliente. En él se declara la información almacenada en un array referente a cada pista, un nombre, un estado y plazoEncendido, un timestamp que será actualizado cuando se efectúe un pago. El estado es la forma de funcionamiento, una pista puede estar encendida o apagada ignorando el timestamp, o en funcionamiento, atendiendo al timestamp.

```
contract PistasGestion is PistasRolAdmin {
    enum Estado {Apagado, Funcionamiento, Encendido}

    struct Pista {
        string nombre;
        uint256 plazoEncendido;
        Estado estado;
    }

    Pista[] public pistas;
```

Figura 25. Contrato inteligente PistasGestion.

Las funciones implementadas en PistasGestion son:

- La función crearPista toma una cadena de texto para el nombre y añade una pista al array que almacena las pistas.

```
function crearPista(string memory _nombre)
    public
    onlyGestor
    returns (uint256)
{
    Pista memory p = Pista(_nombre, 0, Estado.Funcionamiento);
    uint256 n = pistas.length;
    pistas.push(p);
    return n;
}
```

Figura 26. Contrato inteligente PistasGestion, función crearPista.

- La función numeroPistas retorna el número de pistas registradas en el sistema.

- La función resetearPlazos recibe el id de una pista (posición en el array) y en ella almacena un cero en plazoEncendido.

- Las funciones que cambian el modo de funcionar de las pistas reciben el id de una pista y si el valor es correcto y la pista no se encontraba en ese mismo estado lo cambia.

```
function setPistaApagado(uint256 idPista) public onlyGestor {
    require(
        idPista < pistas.length,
        "No existe identificador de la pista reclamada."
    );
    require(
        pistas[idPista].estado != Estado.Apagado,
        "La pista ya está apagada."
    );
    pistas[idPista].estado = Estado.Apagado;
}
```

Figura 27. Contrato inteligente PistasGestion, función setPistaApagado.

- Y por último se ha añadido un modifier que comprueba que el estado de una pista sea Funcionamiento que formará parte de las precondiciones a la hora de pagar para encender una pista.

```
modifier estadoDePistaFuncionamiento(uint256 idPista) {
    require(idPista < pistas.length);
    require(
        pistas[idPista].estado == Estado.Funcionamiento,
        "La pista no se encuentra disponible para el uso al cliente en este momento."
    );
    _;
}
```

Figura 28. Contrato inteligente PistasGestion, modifier estadoDePistaFuncionamiento.

A.3 PistasCliente

Este contrato inteligente implementa la funcionalidad que permite al cliente encender una pista y al administrador del contrato recibir las criptomonedas que se han enviado al contrato. Para ello se mantiene almacenada dicha cantidad en una variable, ganancias. En el contrato también se declaran las variables PrecioXMinuto que almacena cuantas criptomonedas cuesta cada minuto que se mantiene una pista encendida, y TiempoExtra en la que se almacena la cantidad de segundos que se le dan al cliente en compensación al tiempo que tarda en validarse la transacción. Todas las variables son accesibles en lectura excepto ganancias que requiere de permisos de administrador. Y sólo PrecioXMinuto y TiempoExtra son modificables si se posee el rol Gestor.

```
contract PistasCliente is PistasGestion {
    uint256 public PrecioXMinuto = 1 ether;

    uint256 public TiempoExtra = 1 minutes;

    uint256 private ganancias = 0;
```

Figura 29. Contrato inteligente PistasCliente.

El resto de funciones implementadas son:

- La función reservar permite recibir criptomonedas (en la cabecera se denota con payable, si no se incluyera el contrato no aceptaría o realizaría una transferencia

de criptomonedas) y tras superar las precondiciones las anota y actualiza el estado de la pista.

```
function reservar(uint256 idPista, uint256 minutos)
    public
    payable
    estadoDePistaFuncionamiento(idPista)
{
    uint256 nuevoPlazo = now + minutos * 1 minutes + TiempoExtra;
    precondicionesReserva(idPista, msg.value, minutos, nuevoPlazo);

    pistas[idPista].plazoEncendido = nuevoPlazo;
    ganancias += msg.value;
}
```

Figura 30. Contrato inteligente PistasCliente, función reservar.

- La precondición de reserva comprueba que el msg.value, la cantidad de criptomonedas enviada sea la adecuada, que la suma con ganancias no conlleve a un acarreo de bits, que la pista que se quiere encender no esté ocupada y que el nuevo timestamp sea mayor al actual, para asegurar que no ha habido ningún error con el acarreo de bits.

```
function precondicionesReserva(
    uint256 idPista,
    uint256 value,
    uint256 minutos,
    uint256 nuevoPlazo
) private view {
    require(value > 0, "No se ha introducido
    require(
        ganancias + value > ganancias,
        "Error de acarreo de bits: Es necesa
    );
    require(
        value == minutos * PrecioXMinuto,
        "No se ha enviado la cantidad de IOT
    );
    require(
        pistas[idPista].plazoEncendido < now
        "La pista ya está solicitada."
    );
    require(
        nuevoPlazo > now,
        "Error de acarreo de bits, el nuevo
    );
}
```

Figura 31. Contrato inteligente PistasCliente, función precondicionesReserva.

- La función retirarGanancias envía todas las criptomonedas al administrador del contrato que la invoque.

```
function retirarGanancias() public payable onlyAdmin returns (bool) {
    if (!msg.sender.send(ganancias)) {
        return false;
    }
    ganancias = 0;
    return true;
}
```

Figura 32. Contrato inteligente PistasCliente, función retirarGanancias.

Apéndice B

RPI Script

Dentro de la Raspberry Pi, existe una aplicación Javascript muy simple compuesta de un módulo “main” y dos librerías “GPIO Utils.js” y “PistaUtils.js”, cuyo cometido es leer del smart contract los datos de las pistas que se le indica en el archivo de configuración cada cinco segundos. Según el estado de las pistas visto en dichos smart contracts, se utiliza la librería GPIO.js para enviar una señal digital a través de sus pines, para así encender y apagar las luces de un circuito.

B.1 Index.js

En el fichero index se encuentra la tarea principal que realiza el sistema. Cada cinco segundos verifica el estado de las pistas que se le indican en el fichero .env y enciende los pines de salida asignados a cada pista.

```
import { getPista, logPista, isEncendida } from "./src/PistaUtils.js";
import { initGpio, setState, clearGpio } from "./src/GPIO Utils.js";
require("dotenv").config();

const pistas = initGpio(process.env.PISTAS);

const tarea = () => {
  console.log("***** Nueva Comprobacion *******");
  pistas.forEach((pista) => {
    getPista(pista.id).then((res) => {
      console.log("----------------------------");
      logPista(res);
      const enciende = isEncendida(res);
      setState(enciende, pista.led);
    });
  });
};

setInterval(tarea, 5000);
```

Figura 33. Index.js.

B.2 GPIO Utils.js

GPIO Utils es una librería en la que se han implementado funciones para facilitar el uso de los pines de la Raspberry en la prueba de concepto.

En ella se declara la constante GPIO_Numbers donde se encuentran ordenados los GPIO en este orden: 4, 17, 27, 22, 5, 6, 13, 19, 26, 18, 23, 24, 25, 12, 16, 20, 21 para que primero se asocien las pistas con los pines de la columna izquierda y luego con los de la derecha, como se muestra en la figura 34.

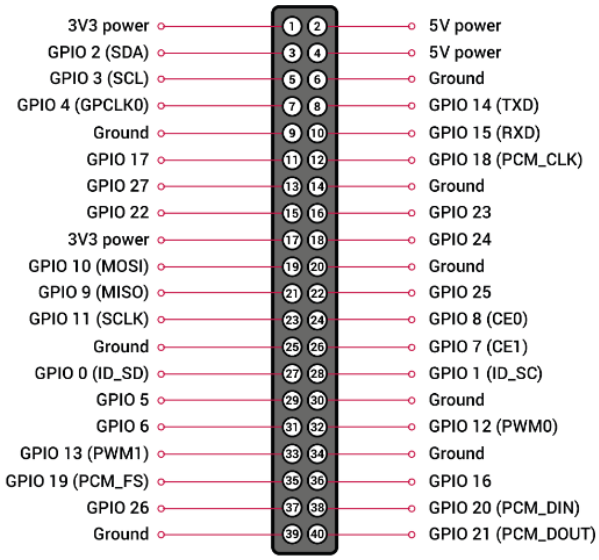

Figura 34. Diagrama de pines de Raspberry

La función initGpio recibe la cadena de texto configurada en .env hace las transformaciones pertinentes para trabajar con datos numéricos y asigna a cada pista un pin GPIO de forma ordenada, devolviendo las asignaciones al main en forma de array.

```
export const initGpio = (cadenaTexto) => {
  const IdPistas = transformarArray(cadenaTexto);
  if (IdPistas.length > GPIO_Numbers.length) {
    throw "La cantidad de pistas superan los GPio disponibles.";
  }
  const arrayDePistaLed = [];
  for (let index = 0; index < IdPistas.length; index++) {
    const pistaLed = {
      id: IdPistas[index],
      led: new Gpio(GPIO_Numbers[index], "out"),
    };
    arrayDePistaLed.push(pistaLed);
  }

  return arrayDePistaLed;
};
```
Figura 35. Función initGpio.

También se ha implementado una función que limpia el estado de todos los pines de la Raspberry (figura 36).

```
export const clearGpio = (arrayDePistaLed) => {
  arrayDePistaLed.forEach((pista) => {
    setState(false, pista.led);
  });
};
```
Figura 36. Función clearGpio.

B.3 PistaUtils.js

PistaUtils es la librería que inicializa la cartera y prepara la conexión, permitiendo a la Raspberry extraer información de la blockchain y comprobar el estado de las pistas.

```
const antenna = new Antenna("http://api.testnet.iotex.one:80");
const wallet = antenna.iotx.accounts.privateKeyToAccount(process.env.PRIV_KEY);
```

Figura 37. Inicialización de conexión y cartera.

La función getPista obtiene una lista con el estado actual de una pista. En la llamada al método de lectura se le indica la dirección de la cartera, la dirección del contrato, el ABI (una descripción de la interfaz del smart contract) y el método.

```
export const getPista = async (idPista) => {
  const pista = await antenna.iotx.readContractByMethod(
    {
      from: wallet.address,
      contractAddress: process.env.CONTRATO_DIRECCION,
      abi: process.env.CONTRATO_ABI,
      method: "pistas",
    },
    idPista
  );
  return pista;
};
```

Figura 38. Función getPista.

La función isEncendida interpreta según el estado de la pista si esta se ha de encender o no. En caso de estar en el modo Funcionando se compara el valor de un timestamp (dividido entre mil porque está en milisegundos) y el valor del timestamp plazoEncendido de la pista (en segundos). Y muestra ambos valores por pantalla en forma de fecha.

```
export const isEncendida = (pistaArray) => {
  switch (pistaArray[2].toNumber()) {
    case Estados.Apagado:
      return false;

    case Estados.Encendido:
      return true;

    case Estados.Funcionando:
      const now = new Date();
      const timestamp = now.getTime();
      console.log("Timestamp actual:\n" + new Date(timestamp));
      console.log("Timestamp plazo:\n" + new Date(pistaArray[1] * 1000));
      return Math.trunc(timestamp / 1000) < pistaArray[1];
  }
};
```

Figura 39. Función isEncendida.

B.4 .env

En este fichero de configuración se indican las pistas que la Raspberry va comprobar, una clave privada para generar la cartera, la dirección del contrato, y su ABI.

```
PISTAS="0,1,2"
PRIV_KEY="este string genera una clave privada"
CONTRATO_DIRECCION="io1gr8j7qdzjgphmckuk28mlgdddwt6xnymhtwa6g"
CONTRATO_ABI=[{"anonymous":false,"inputs":[{"indexed":true,"internal
```

Figura 40. Fichero de configuración .env.

Apéndice C

Aplicación web

En este apéndice se documenta la parte de la aplicación web que realiza la comunicación con IoPay, su fichero de configuración y sus pantallas, sin centrarse en el código de las últimas.

C.1 Public.ts

Es un fichero de configuración donde se indica el punto final de comunicación, la dirección del contrato y su ABI, dentro de un String y como objeto Javascript.

```
export const publicConfig = {
  IOTEX_CORE_ENDPOPINT: "https://api.testnet.iotex.one:443",
  CONTRATO_DIRECCION: "io1gr8j7qdzjgphmckuk28mlgdddwt6xnymhtwa6g",
  CONTRATO_ABI:
    '[{"anonymous":false,"inputs":[{"indexed":true,"internalType":"b
  CONTRATO_ABI_NO_STRING: [
    {
      anonymous: false,
      inputs: [
        {
```

Figura 41. Fichero de configuración Public.ts

C.2 Wallet.js

Wallet.js es el módulo donde se prepara la estructura de datos que se va a almacenar en el estado global de la aplicación web en el lado del cliente. Para facilitar la actualización dinámica de las vistas frente a dicho estado global se hace uso del framework MobX, que permite seguir dichos cambios definiendo las variables observadas, los componentes que las observan y las acciones que las modifican.

La cartera almacena la dirección, el balance y roles de la cartera abierta en IoPay, además de una variable, isConnectWsError, donde se almacena información en caso de error. En el constructor se indican el rol del resto de componentes que forman parte de la cartera. Variables observables y acciones.

```
@remotedev({ name: "wallet" })
export class WalletStore {
  account = {
    address: "",
    balance: "",
    gestor: false,
    admin: false,
  };

  isConnectWsError = false;
  constructor() {
    makeObservable(this, {
      account: observable,
      isConnectWsError: observable,
      initWS: action,
      init: action,
    });
  }
```

Figura 42. Estructura del estado de la cartera en la aplicación web.

La aplicación web al cargarse intenta sincronizarse con IoPay a través de la función init.

```
function App() {
  const { wallet } = useStore();

  useEffect(() => {
    wallet.init();
  });
```

Figura 43. Llamada a la función de inicialización de la cartera.

Esta función llama a initEvent que inicializa la escucha de eventos definidos en EventBus (parte de la librería Antenna Utils), y que en caso de recibir los eventos de cierre o error de conexión por parte de IoPay, limpia el estado de la cartera.

```
async init() {
  this.initEvent();
  await this.initWS();
}

initEvent() {
  utils.eventBus
    .on("client.iopay.connected", () => {
      console.log("iopay-desktop connected.");
      this.isConnectWsError = false;
    })
    .on("client.iopay.close", () => {
      this.account = {
        address: "",
        balance: "",
        gestor: false,
        admin: false,
      };
    })
    .on("client.iopay.connectError", () => {
      this.account = {
        address: "",
        balance: "",
        gestor: false,
        admin: false,
      };
      this.isConnectWsError = true;
    });
}
```

Figura 44. Función init e init Event.

La función initWS mediante la clase AntennaUtils recibe la información de la cartera abierta en IoPay. Empezando por su dirección, si esta falla, se vuelve a ejecutar en dos segundos. Si no hay ningún error, se cambia el estado de la dirección dentro de la acción, para que MobX pueda actualizar la información que dependa de la dirección en los componentes activos que la observen. Y finalmente se carga el resto del estado con loadAccount.

```
async initWS() {
  const [err, address] = await utils.helper.promise.runAsync(
    AntennaUtils.getIoPayAddress()
  );

  if (err || !address || address === "") {
    return setTimeout(() => {
      this.initWS();
    }, 2000);
    // @ts-ignore
  }
  runInAction(() => {
    this.account.address = address;
  });
  this.loadAccount();
}
```

Figura 45. Función initWS.

La función loadAccount empieza cargando el balance de la cuenta desbloqueada en IoPay. En caso de error la función vuelve a ejecutarse, en caso contrario, la cantidad de criptomonedas se guarda en el estado global en IOTX (haciendo un cambio de moneda de Rau a IOTX, 1 IOTX = 10^{18} Rau).

```
async loadAccount() {
  // @ts-ignore
  const [err, data] = await utils.helper.promise.runAsync(
    // @ts-ignore
    AntennaUtils.getAntenna().iotx.getAccount({
      address: this.account.address,
    })
  );

  if (err || !data) {
    return setTimeout(() => {
      this.loadAccount();
    }, 2000);
  }

  if (!err && data) {
    if (data?.accountMeta) {
      const { balance } = data?.accountMeta;
      runInAction(() => {
        this.account.balance = fromRau(balance, "iotx");
      });
    }
  }
```

Figura 46. Función loadAccount.

Después del balance, de la misma forma, se cargan ambos roles. En la figura 47 se muestra cómo se hace en uno de ellos, el código del otro es igual, cambiando la variable de rol. Si durante el proceso de recuperación de dicha información hay un error, se vuelve a ejecutar la función tras dos segundos.

```
const [err1, data1] = await utils.helper.promise.runAsync(
  hasRol(Roles.ADMIN_ROLE, this.account.address)
);

if (err1 || null == data1) {
  return setTimeout(() => {
    this.loadAccount();
  }, 2000);
}

runInAction(() => {
  this.account.admin = data1;
});
```

Figura 47. Función loadAccount, carga de rol de administrador.

C.3 PistaUtils.ts

Este módulo de la aplicación es una ampliación del módulo PistaUtils.js con algunos cambios, entre ellos la transformación del código a Typescript debido al tamaño del mismo. Al principio de este módulo se declara una interfaz para facilitar la manipulación de la información en el resto del programa. Y además se declaran las constantes con el valor de los roles y los estados de las pistas.

```
interface PistaObjeto {
  id: number;
  estado: string;
  plazoEncendido: string;
  nombre: string;
}

export const Roles = {
  ADMIN_ROLE: "0x0",
  GESTOR_ROLE:
    "0x687797022fb88d524c9d386b0ec1c01a0799fff1f1f8403113dffa17f4bf6ab2",
};

export const Estados = {
  Apagado: 0,
  Funcionando: 1,
  Encendido: 2,
};
```

Figura 48. Interfaz y constantes de PistaUtils.ts.

La función getContrato crea la interfaz con la que se interactúa con el contrato inteligente. Se le envía como parámetros el ABI, la dirección del contrato, el provider y el signer. El provider almacena la información referente a la dirección del usuario que se utiliza en el contrato, y el signer la interfaz a través la cual se va a realizar la comunicación con IoPay (que en teoría debería variar de forma automática entre móvil y web).

```
function getContrato() {
  const antenna = AntennaUtils.getAntenna();
  return new Contract(
    publicConfig.CONTRATO_ABI_NO_STRING,
    publicConfig.CONTRATO_DIRECCION,
    // @ts-ignore
    { provider: antenna.iotx, signer: antenna.iotx.signer }
  );
}
```

Figura 49. Función getContrato.

El siguiente tipo de función a documentar son las de lectura del smart contract. Las funciones de lectura no realizan la comunicación a través de IoPay, estas envían la petición al endpoint de IoTeX directamente, aunque si obtienen la dirección de la cuenta desde IoPay. Estas funciones están programadas de dos formas, mediante la llamada readContractByMethod o a través de la interfaz creada por la función getContrato.

Las que utilizan readContractByMethod hacen uso del objeto antenna para obtener la información de la cuenta usuario y consultar el contrato. Estas funciones son:

- getPrecio (Figura 50).
- getTiempoExtra.
- getGanancias.
- numeroPistas.
- getPista.
- getAllPistas. Llama en bucle a getPista.

```
export async function getPrecio() {
  const antenna = AntennaUtils.getAntenna();
  // @ts-ignore
  const precio = await antenna.iotx.readContractByMethod({
    from: await AntennaUtils.getIoPayAddress(),
    contractAddress: publicConfig.CONTRATO_DIRECCION,
    abi: publicConfig.CONTRATO_ABI,
    method: "PrecioXMinuto",
  });
  return precio;
}
```

Figura 50. Función getPrecio.

Las que utilizan la interfaz del contrato se han utilizado por la necesidad de enviar más de un parámetro como argumento de la función. También utilizan el objeto antenna para obtener la dirección del usuario. Estas funciones son:

- hasRol.
- isLoggedAccountGestor.
- getNumGestores.
- getAllGestores. Esta realiza llamadas al contrato en bucle (figura 51).

```
export async function getAllGestores() {
  const contrato = getContrato();
  const cuentaActual = await AntennaUtils.getIoPayAddress();
  const numGestores = await getNumGestores();
  let gestor;
  const arrayAccounts = [];

  for (let index = 0; index < numGestores; index++) {
    gestor = await contrato.methods.getRoleMember(Roles.GESTOR_ROLE, index, {
      account: cuentaActual,
    });
    arrayAccounts.push(gestor);
  }
  return arrayAccounts;
}
```

Figura 51. Función getAllGestores.

El siguiente tipo de función a documentar son las de escritura. Como se ha mencionado anteriormente en la sección 5.5, estas funciones contactan con IoPay para que el usuario confirme la transacción y sea enviada a través del mismo. En su programación no se aprecia una gran diferencia con las anteriores. Estas funciones son:

- setTiempoExtra (figura 52).
- reservar.
- crearPista.
- darRolGestor.
- setPrecioXMinutos.
- retirarGanancias.
- setPistaEstado. Esta función tiene un switch interno que según el estado que se le envía por argumento, llama a un método del contrato u otro para cambiar el estado.
- quitarRolGestor.

```
export async function setTiempoExtra(segundos: number) {
  const contrato = getContrato();
  const cuenta = await AntennaUtils.getIoPayAddress();

  const hash = await contrato.methods.setTiempoExtra(segundos, {
    account: cuenta,
    amount: 0,
    ...AntennaUtils.defaultContractOptions,
  });

  return hash;
}
```

Figura 52. Función setTiempoExtra.

El resto de funciones solo contiene lógica Javascript. Una de ellas es isEncendida explicada en el apéndice B.3, isReservable, que comprueba que una pista se encuentre en estado de funcionamiento y no esté encendida; getPistaObject, que llama a getPista

(lectura de contrato) y transforma el array que devuelve en el objeto tipado por la interfaz PistaObjeto (figura 48); y dos funciones que muestran información en la consola sobre las transacciones y el estado de las pistas.

C.4 Interfaz de Usuario

Las interfaces de usuario son componentes React, cada uno de estos componentes pueden ejecutar código cuando son renderizados. Para ello, dentro del componente se hace la llamada a useEffect, una función que se utiliza comúnmente para cargar información de forma asíncrona en los componentes, y que ejecuta la función que se le pase como argumento (este comportamiento es modificable, si se especifica también se puede hacer que useEffect se ejecute cuando una variable de estado local del componente es modificada, aunque esta característica no se use en este proyecto), el primer ejemplo de uso de esta función se encuentra en la figura 43.

El resto de funciones de escritura se ejecuta a través de los eventos generados por botones. En el ejemplo de la figura 53 se muestra el ejemplo de la función asignada al botón de reserva de la vista Reserva. En esta se muestra cómo se llama a la función reservar de la librería PistaUtils.ts y el tratamiento de la respuesta, que dependiendo de si se lanza una excepción, mostrará un mensaje de error o de envío satisfactorio en la vista. En el resto de ejemplos se omitirá la parte del tratamiento de la respuesta.

```
function reservaOnClick(event) {
  event.preventDefault();
  reservar(pista, Number(minutos))
    .then((response) => {
      const link = "https://testnet.iotexscan.io/action/" + response;
      setLogMessage(
        <p className="text-success">
          La transacción se ha enviado con éxito, puedes verificar si se ha
          ejecutado correctamente clicando aquí: <a href={link}>{link}</a>.
        </p>
      );
    })
    .catch((response) => {
      setLogMessage(
        <p className="text-danger">
          La transacción no ha podido realizarse debido a un error con el
          envío de la transaccion
        </p>
      );
    });
}
```

Figura 53. Función para el botón reserva.

C.4.1 Index.jsx

Index es una vista cuyo código es puramente visual y de navegación (Cuando se clica en "Empieza ahora" lleva al tutorial). Como aclaración, la barra de navegación es un componente aparte, en esta vista y en el resto.

Figura 54. Vista de la página principal.

C.4.2 Tutorial.jsx

Es una vista cuyo código también es puramente visual. Esta explica cómo hacer la instalación de IoPay, cómo configurarlo en modo Testnet, cómo recibir criptomonedas de la Testnet y cómo empezar a utilizar la aplicación.

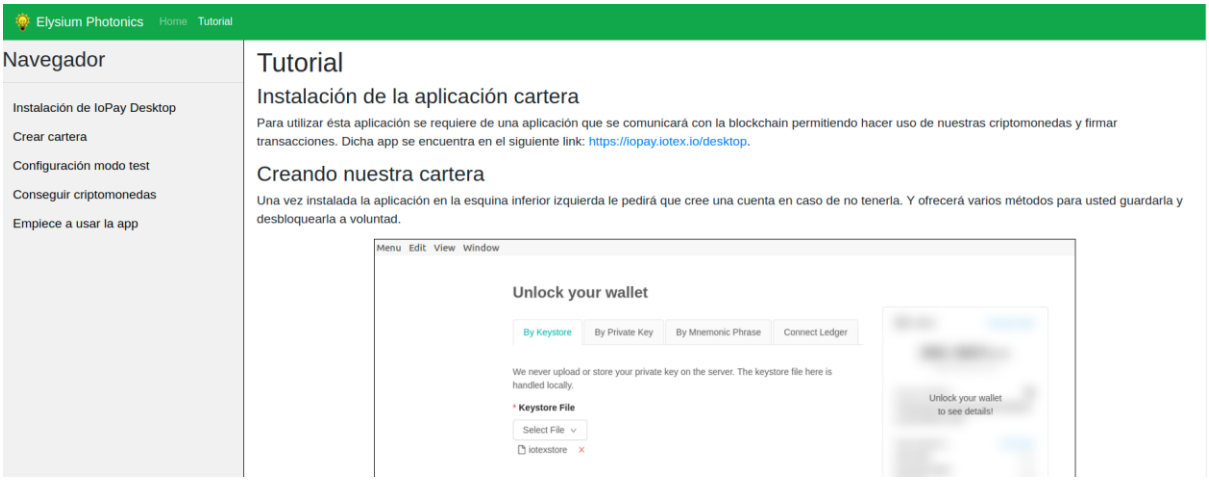

Figura 55. Vista del tutorial.

C.4.3 VistaPistas.jsx

En la vista de las pistas se recoge la información sobre estas, y según su estado muestra botones "Pista ocupada" o "Reserve ahora" (este botón lleva a la pantalla reserva).

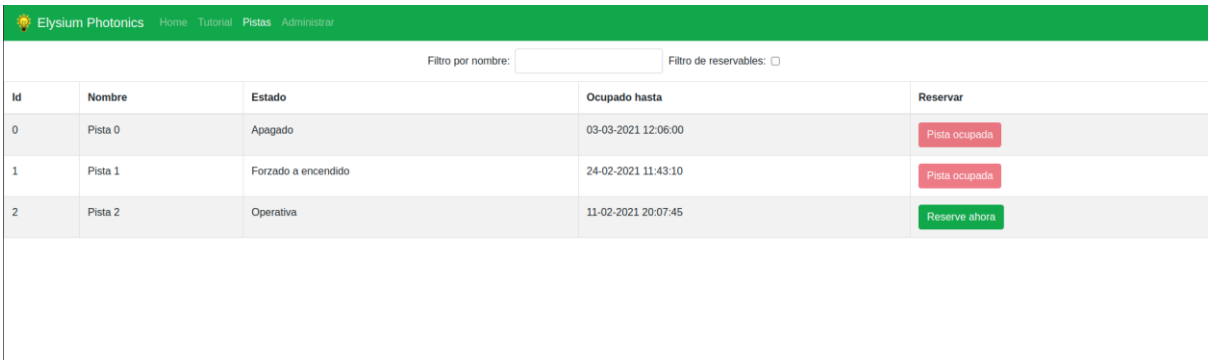

Figura 56. Vista de las pistas.

Cuando se renderiza, trae la información de todas las pistas (figura 57) del sistema, esta información es tratada y mostrada de la forma mencionada en el párrafo anterior.

```
useEffect(() => {
  getAllPistas().then((data) => {
    setPistas(data);
  });
}, []);
```

Figura 57. Llamada a useEffect de VistaPistas invocando a getAllPistas.

C.4.4 Reserva.jsx

Reserva es un formulario que extrae del smart contract el precio de las pistas con la función getPrecio, y la pista que le llega por parámetro en el link con la función getPistaObject de PistaUtils. Y cuando se registra la cantidad de minutos que se desea reservar y se pulsa el botón de enviar, se invoca la función reservar de PistaUtils.

```
useEffect(() => {
  getPistaObject(parseInt(id, "10")).then((response) => {
    setPista(response);
  });
  getPrecio().then((response) => {
    setPrecioXMinuto(response);
  });
}, [id]);

function reservaOnClick(event) {
  event.preventDefault();
  reservar(pista, Number(minutos))
}
```

Figura 58. UseEffect y acción del botón de reserva de Reserva.jsx.

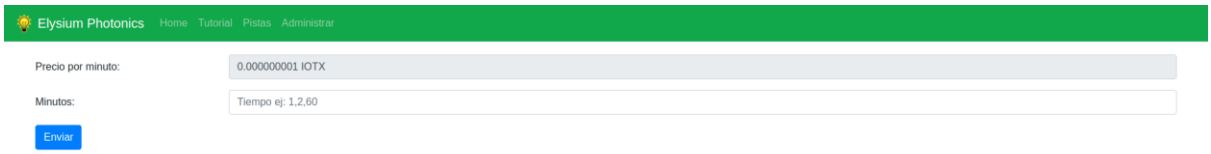

Figura 59. Vista de Reserva.

C.5 Administración del sistema

Dentro de esta vista se encuentran el resto de formularios que modifican el estado de la Blockchain.

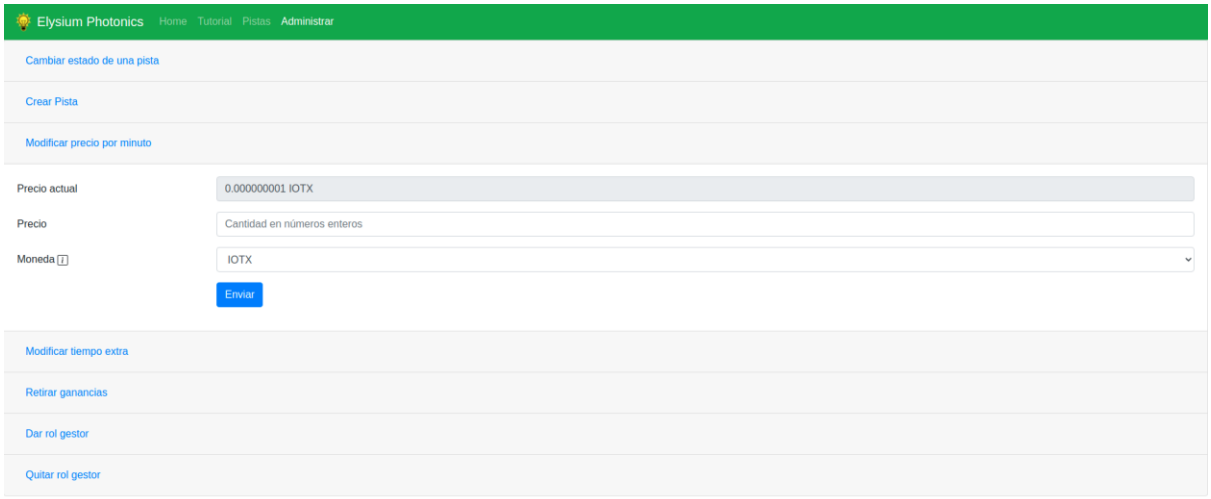

Figura 60. Vista de Administrar.jsx.

C.5.1 FormCrearPista.jsx

Este formulario permite la creación de pistas, llamando a la función crearPista de PistaUtils. Hay que poseer el rol gestor para ello.

```
const enviar = (event) => {
  event.preventDefault();
  if (nombrePista != null && nombrePista.length > 0) {
    crearPista(nombrePista)
  }
};
```

Figura 61. Llamada a crearPista desde FormCrearPista.

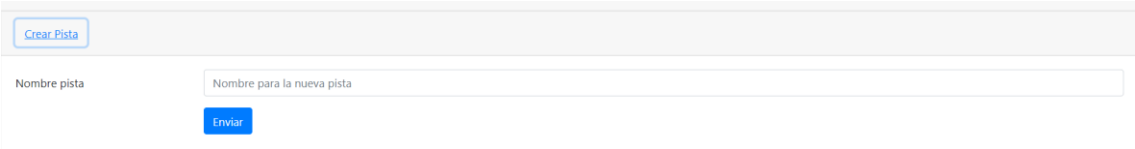

Figura 62. Vista de FormCrearPista.

C.5.2 FormEstadoPista.jsx

Este formulario permite cambiar el estado de una pista (Funcionando, Apagado y Encendido), llamando a la función setPistaEstado de PistaUtils. Hay que poseer el rol gestor para ello.

```
function enviar() {
  if (id !== null && estado !== null) {
    setPistaEstado(id, estado)
  }
}
```

Figura 63. Llamada a setPistaEstado desde FormEstadoPista.

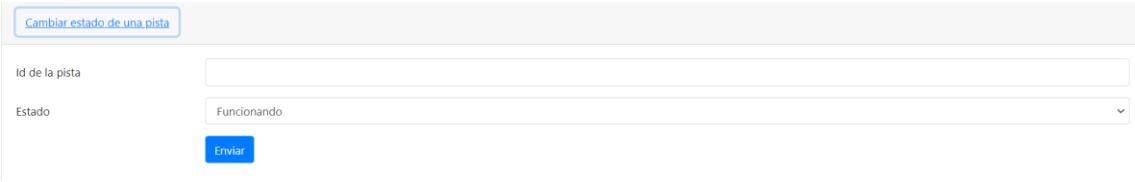

Figura 64. Vista del formulario FormEstadoPista.

C.5.3 FormPrecio.jsx

Este formulario recupera la información del precio actual en el useEffect con la función getPrecio de PistaUtils, y lo almacena en el estado local del componente para ser renderizado (setPrecioActual) tras ser transformado de moneda Rau a IOTX (1 IOTX = 10^{18} Rau). El botón de enviar permite cambiar el precio de todas las pistas haciendo uso de la función setPrecioXMinutos de PistaUtils. Hay que poseer el rol gestor para ello.

```
useEffect(() => {
  getPrecio().then((response) => {
    const precioEnIotex = fromRau(response, "IOTX") + " IOTX";
    setPrecioActual(precioEnIotex);
  });
}, []);

const enviar = (event) => {
  event.preventDefault();
  setPrecioXMinutos(nuevoPrecio, selectValue)
};
```

Figura 65. UseEffect y acción del botón de FormPrecio.

Figura 66. Vista de FormPrecio.

C.5.4 FormTiempoExtra.jsx

Este formulario recupera la información del tiempo extra, que se ofrece para compensar el tiempo que tarda en confirmarse la transacción en el useEffect con la función getTiempoExtra de PistaUtils, almacena el valor en el estado local del componente para ser renderizado (al igual que en el formulario anterior). El botón de enviar permite cambiar dicho tiempo extra haciendo uso de la función setTiempoExtra de PistaUtils. Hay que poseer el rol gestor para ello.

```
useEffect(() => {
  getTiempoExtra().then((response) => {
    setTiempoExtraActual(response + " minutos.");
  });
}, []);

function enviar(event) {
  event.preventDefault();
  setTiempoExtra(tiempo)
}
```

Figura 67. UseEffect y acción del botón de FormTiempoExtra.

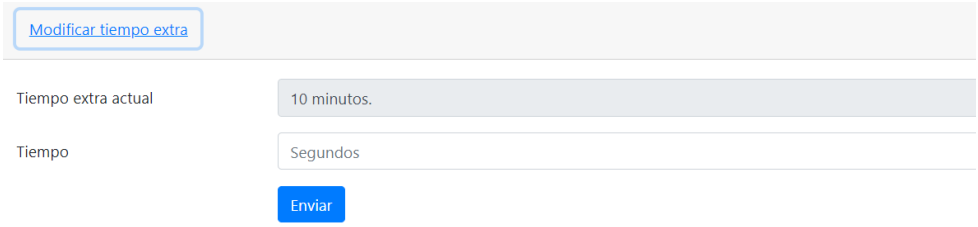

Figura 68. Vista de FormTiempoExtra.

C.5.5 FormRetirarGanancia.jsx

Este formulario recupera la información de la cantidad de criptomonedas recibidas por el contrato mediante el useEffect con la función getGanancias de PistaUtils, y lo almacena en el estado local del componente para ser renderizado (al igual que en el formulario anterior). El botón de enviar permite retirar dicha cantidad haciendo uso de la función retirarGanacias. Hay que poseer el rol administrador para ello.

```
useEffect(() => {
  getGanancias().then((response) => {
    const gananciasString = fromRau(response, "IOTX") + " IOTX";
    setGanancias(gananciasString);
  });
}, []);

const enviar = (event) => {
  event.preventDefault();
  retirarGanancias()
};
```

Figura 69. UseEffect y acción del botón de FormRetirarGanancia.

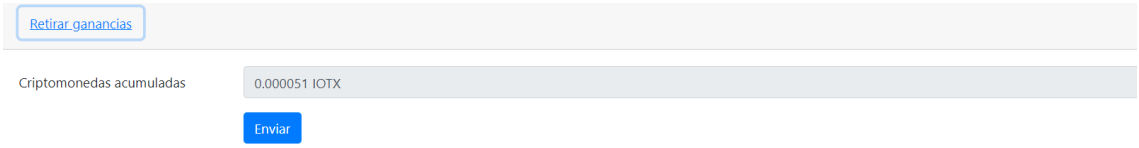

Figura 70. Vista de FormRetirarGanancia.

C.5.6 FormDarRolGestor.jsx

Este formulario permite otorgar el rol gestor haciendo uso de la función darRolGestor. Hay que poseer el rol administrador para ello.

```
function enviar(event) {
  event.preventDefault();
  darRolGestor(cuenta)
}
```

Figura 71. Acción del botón de FormDarRolGestor.

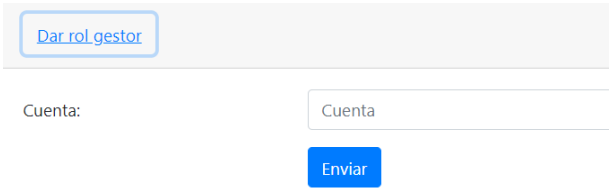

Figura 72. Vista de FormDarRolGestor.

C.5.7 FormQuitarRolGestor.jsx

Este formulario extrae las direcciones de todos los gestores del smart contract en el useEffect, invocando a la función getAllGestores de PistaUtils. Y además permite revocar el rol gestor haciendo uso de la función quitarRolGestor de PistaUtils. Hay que poseer el rol administrador para ello.

```
useEffect(() => {
  getAllGestores().then((response) => {
    setFilas(response);
  });
}, []);

const buttonOnClick = () => {
  quitarRolGestor(cuenta)
};
```

Figura 73. UseEffect y acción del botón de FormQuitarRolGestor.

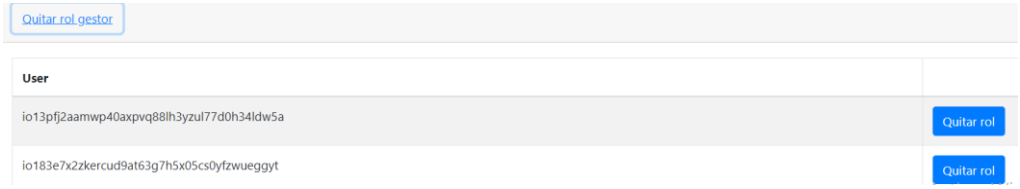

Figura 74. Vista de FormQuitarRolGestor.

Apéndice D

Instalación

Para la instalación de la aplicación web y del script para RPi requieren de la instalación de Node y Python 3, una vez instalados, para recuperar los paquetes de librerías con los que trabajan ambas ejecutar la instrucción *npm install* en sus respectivas carpetas. En caso de que node lo recomiende, ejecutar *npm audit fix*. Para que comience la ejecución de ambos programas, es necesario ejecutar el comando *npm start*.

Para el despliegue del smart contract se selecciónа la pestaña Smart Contracts en IoPay y Deploy contract.

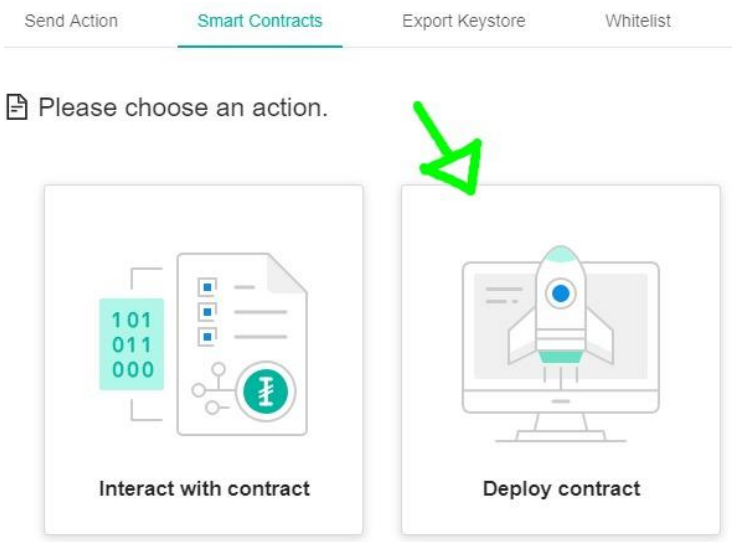

Figura 75. Pestaña Smart Contracts.

Una vez clicado se despliega el menú donde hay que rellenar los datos del smart contract, con el ABI y el archivo binario que viene en la carpeta Contracts, en el código entregado.

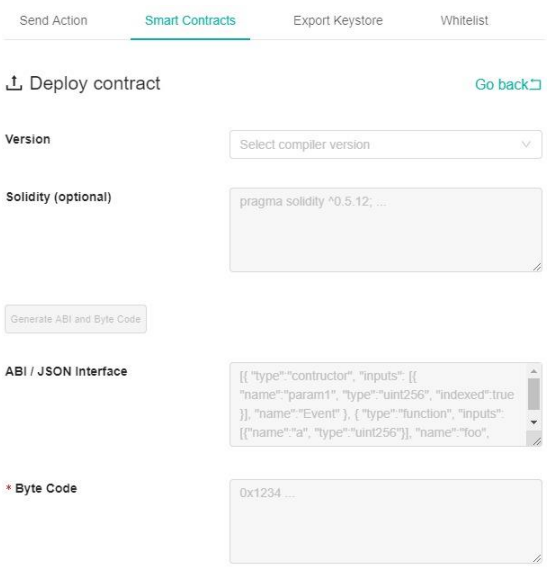

Figura 76. Pestaña Deploy Smart Contract.

Para hacer a menos la parte a quien quiera comprobar el sistema, se ha dejado en la carpeta con el código de la aplicación una cartera con todos los privilegios (Cuenta full privilegios.json). Esta se desbloquea con la contraseña "contraseña1234". La aplicación web ha sido desplegada utilizando Firebase en el siguiente link: "https://mi-tfg-e78be.web.app/".

UNIVERSIDAD DE MÁLAGA | uma.es

E.T.S. DE INGENIERÍA INFORMÁTICA

E.T.S de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga