UNIVERSIDAD INTERNACIONAL DE LA RIOJA

Universidad Internacional de La Rioja
Máster Universitario en Seguridad Informática

Aseguramiento de Dispositivos *IoT* con *Blockchain* e Infraestructura de Clave Pública

Trabajo Fin de Máster, depósito final (extraordinaria)

Presentado por: Balmaseda Aranda, Francisco Javier

Director: Paniagua Díez, Fidel

Ciudad: Madrid
Fecha: 18/09/2018

Resumen

Este proyecto fin de Máster está basado en demostrar las capacidades que tienen las nuevas tecnologías, como la cadena de bloques al unirlas a otras implementaciones tradicionales de aseguramiento, como son las infraestructuras de clave pública para el manejo de claves criptográficas, distribuyendo y expandiendo sus posibilidades, lo que permite asegurar despliegues de dispositivos Internet de las Cosas garantizando su acceso remoto confiable mediante redes como Internet.

Una vez explicados los conceptos teóricos de las tecnologías utilizadas, así como los de los procesos de control y gestión digitales en base a las identidades de dispositivos y usuarios, se establece una metodología de auditoría basada en riesgos que nos permitirá evaluar los prototipos presentados para medir las mejoras obtenidas en términos de seguridad, un dispositivo sin asegurar en base a su configuración por defecto y otro mediante la implementación y despliegue de una infraestructura de clave pública distribuida bajo la tecnología Blockchain.

Palabras Clave: Internet de las Cosas, Cadena de Bloques, Clave Pública, Gestión Digital, Riesgos.

Abstract

This Master's degree project is based on demonstrating the capabilities of new technologies, such as the Blockchain by joining them to other traditional assurance implementations, such as Public Key Infrastructures for the management of cryptographic keys, distributing and expanding their possibilities, that allows to ensure deployments of Internet of Things devices guaranteeing its reliable remote access through networks such as the Internet.

Once explained the theoretical concepts of the technologies used, as well as the processes of digital control and management based on the identities of devices and users, a risk-based audit methodology is established that will allow us to evaluate the prototypes presented to measure the improvements obtained in terms of security, an unsecured device based on its default configuration, and another by implementing and deploying a distributed Public Key Infrastructure under the Blockchain technology.

Keywords: Internet of Things, Blockchain, Public Key, Digital Management, Risks.

Agradecimientos:

A mi familia.

A Lucía.

Amigos y compañeros que me han apoyado.

Contenido

1. Introducción

1.1. Presentación

Los nuevos ecosistemas digitales basados en redes de comunicaciones distribuidas no están originalmente orientados a un aseguramiento de las identidades de los dispositivos involucrados ni a una correcta trazabilidad en sus procesos, dándole mayor relevancia a otros factores, como son la disponibilidad o la resiliencia como base de su funcionamiento, donde normalmente existe una única entidad central que gestiona la base de datos de identidades.

En un entorno global donde la ciberseguridad es cada vez más relevante para salvaguardar y dotar de confianza digital a estos procesos, se hace necesario implementar mecanismos que identifiquen y autoricen de manera única, inequívoca y consensuada a estos dispositivos, con la capacidad de registrar de la misma forma todas las transacciones que estos realizan, de tal manera que se descentralicen las entidades de control y se evolucione a unos sistemas mucho más cooperantes y distribuidos capaces de crear identidades digitales únicas como base que sostenga una sólida estructura de Sociedad de la Información a prueba de fallos o manipulaciones.

La tecnología de cadena de bloques o *Blockchain,* con su carácter dinámico y disruptivo, se configura como una alternativa a estos modelos tradicionales de gestión y control de las identidades, mejorando sus procesos digitales y el intercambio de información de forma segura.

1.2. Planteamiento y motivación del trabajo

Los dispositivos del Internet de las Cosas (*Internet of Things, IoT*), se están convirtiendo en una plataforma muy popular tanto para consumidores, corporaciones o en entornos de Ciudades Inteligentes dada su versatilidad, facilidad de implementación y bajo coste, pero que aún se basan en modelos tradicionales centralizados de control y gestión de estos dispositivos.

Los modelos centralizados de implementación de dispositivos *IoT* donde es necesario intercambiar información entre ellos de forma autónoma se tornan poco prácticos e inseguros, lo que nos sugiere modelos descentralizados como el que facilita la tecnología *Blockchain*, que permitan el intercambio seguro de datos, la confianza en los procesos y el mantenimiento de registros de todos los mensajes. La identidad basada en infraestructuras de clave pública (*PKI*), ampliamente utilizadas para la distribución y gestión centralizada de certificados digitales, y donde cada dispositivo necesita tener un certificado raíz de una Autoridad de Certificación (*CA*), son necesarios para su verificación, pero crean una dependencia de la *CA* que limita su escalabilidad. *Blockchain* permite que la seguridad dependa habitualmente del uso de algoritmos criptográficos para esta verificación, que dependen a su vez de la disponibilidad de la cadena de bloques. (Crosby, Pattanayak, Verma, & Kalyanaraman, 2016)

Según la empresa americana de formación en Ciberseguridad *CyberTraining 365*, y dentro de las predicciones para 2018 en esta materia, definen un *TOP 5*, donde las dos primeras corresponden por un lado, a futuras y prometedoras implementaciones sobre *Blockchain* orientadas a la seguridad, lo que indica que la tecnología está madurando y avanza como candidata a consolidarse como un estándar de seguridad; y por otro lado, al progresivo aumento en los ataques sobre los dispositivos *IoT* en base a lo que con ellos está ocurriendo en los últimos años, que indican que van a ser cada vez más frecuentes, más masivos y agresivos, y con una mayor motivación económica para comprometer este tipo de dispositivos. (Kehal, 2018)

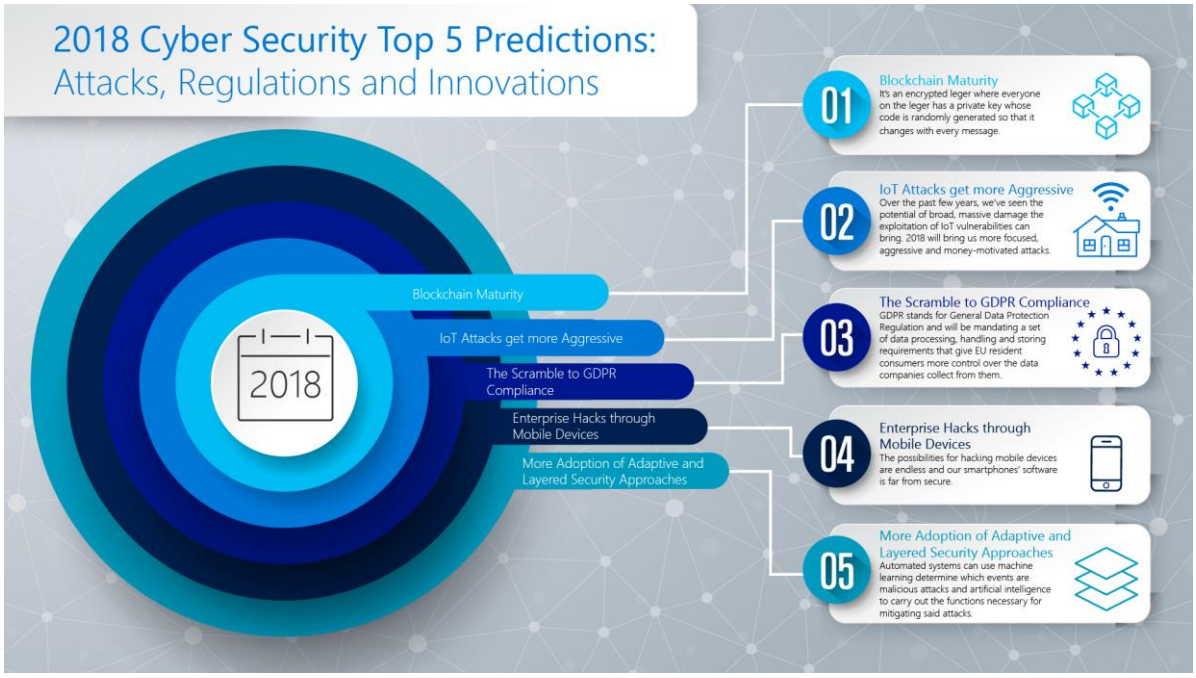

Ilustración 1. CyberTraining 365 Blog. (Kehal, 2018)

Bajo estos conceptos se plantea un estudio previo de la tecnología *Blockchain* que nos permita una implementación de un entorno funcional de *IoT* a escala donde se disponga de dispositivos autenticados, identificados y registrados mediante una *PKI* descentralizada que opere con la cadena de bloques al objeto de evaluar el riesgo mediante un proceso de auditoría de la seguridad y cuál serían las posibles mejoras en comparación con las soluciones tradicionales y con despliegues sin asegurar.

Como lo que se pretende es verificar principalmente las identidades digitales legítimas y las transacciones que estas realizan al acceder a recursos mediante el control de acceso, es la gestión distribuida de las claves de cifrado la que permite una autenticación confiable basada en certificados donde se mantiene la integridad de los procesos e identidades y se garantiza la disponibilidad del entorno. Asegurar la confidencialidad no es el objeto de la *PKI* distribuida (*D-PKI),* ya que se aloja en un entorno *Blockchain* de carácter público y se deja este aspecto a otros protocolos de transporte añadidos al sistema final de acceso remoto a los dispositivos protegidos.

1.3. Alcance del trabajo

Este trabajo pretende ser un acercamiento teórico / práctico a la gestión del acceso confiable de dispositivos y usuarios mediante el uso de la tecnología de cadena de bloques, de modo transversal e integrador, siendo capaces de crear un escenario donde los procesos de autenticación se doten de mayor seguridad bajo la implementación de una *PKI* descentralizada basada en *Blockchain* que permita gestionar de forma óptima los dispositivos *IoT* desplegados en la red, poniendo el foco en los procesos que aseguran la verificación de la identidad digital, dando solución a los problemas comunes de autenticación de las implementaciones tradicionales y añadiendo otros mecanismos de autorización adicionales como el control de acceso mediante permisos a los usuarios a nivel de aplicación y múltiples factores de autenticación; todo esto unido a la gestión descentralizada de certificados digitales y estándares de intercambio de datos de autenticación y autorización como *SAML (Amazon Web Services, 2018),* asegurados bajo protocolos criptográficos de cifrado como *SSL/TLS* a nivel de transporte, como es el caso de estudio de este proyecto de una *PKI* bajo *Blockchain* para el aseguramiento de dispositivos *IoT* conectados en comparación con dispositivos sin asegurar.

Evaluaremos un entorno de dispositivos *IoT* conectados a Internet, accesibles remotamente mediante consola de comandos *Linux*, en base a sus riesgos, amenazas y vulnerabilidades

de origen o por implementación y, por tanto, la metodología a utilizar será la habitual en los procesos de análisis de la seguridad, metodología de auditoria informática basada en riesgos, evaluación del riesgo (*EDR*). Dispondremos de un dispositivo accesible remotamente sin asegurar que nos dará un índice elevado de riesgo como base para poder actuar sobre los indicadores de compromiso detectados, y un segundo dispositivo al que se le va a aplicar la gestión del riesgo definida para dispositivos *IoT* y orientada a la implementación de una *D-PKI* que facilite una gestión del acceso confiable para el propio dispositivo, los servicios contenidos en él y los usuarios registrados mediante identidades criptográficas basadas en *Blockchain*, convirtiéndolo en un servidor web público confiable que permita el acceso remoto mediante servicio de terminal publicado en él y mediante un navegador compatible con protocolos de cifrado, lo que permitirá reducir drásticamente el nivel de riesgo a límites aceptables en términos de seguridad y en función de la metodología aplicada.

1.4. Justificación

Blockchain se perfila como una apuesta emergente y de futuro de la que se prevé un crecimiento exponencial de sus aplicaciones, que darán respuesta a las necesidades de ciberseguridad en muchos sectores de las *TIC.* Aunque aún está lejos de considerarse una solución integral, y siendo el estudio teórico y conceptual necesarios para la mejor comprensión del proyecto, es la parte práctica y de solución real la que justifica y aporta mayor valor a la investigación, siguiendo la línea de otros trabajos y estudios que proponen la aplicación de esta tecnología en el ámbito de la ciberseguridad, fruto de la fase de experimentación en la que aún está inmersa.

Se pretende dar una visión de *Blockchain* más allá del concepto obvio de cadena enlazada de bloques donde se elimina la necesidad de intermediarios, para enfatizar en su capacidad de validación y verificación de la información publicada en esta cadena, basada en el consenso de los nodos involucrados, lo que permite una seguridad robusta ante posibles alteraciones gracias al uso de los algoritmos criptográficos para el reconocimiento de las acciones implementadas, que faciliten el despliegue de una infraestructura de dispositivos correctamente validados. Esta inmutabilidad de los datos generados es la que proporciona una validación coherente que genera un fiable historial de transacciones y expande las posibilidades de otras tecnologías tradicionales en el mundo de la ciberseguridad,

ofreciendo características diferenciadoras que, si son tomados en cuenta, podrán aportar ventajas y beneficios que permitan casos de éxito en la aplicación de esta tecnología.

Según el estudio de la firma *Juniper Research, "The Internet of Things for Security Providers: Opportunities, Strategies, & Market Leaders 2016-2021"*, los dispositivos *IoT* conectados a Internet en 2021 alcanzarán los 15.000 millones de unidades, aumentando en un 120% con respecto al 2016. Las actuales redes *Botnets (Kaspersky, 2013)* o de dispositivos zombis infectados que dispararon los ataques de tipo Denegación de Servicio Distribuidos (*DDoS)* desde 2016, son sólo la punta del iceberg de lo que se prevé en un futuro cercano, lo que nos podría llevar a un escenario de riesgo inmanejable en cuanto a ciberseguridad, creado por redes de dispositivos comprometidos que llegarían a alcanzar más de un millón de unidades. (Sorrell, 2017)

Cabe destacar que la búsqueda de debilidades o vulnerabilidades al sistema se hace esencial para asegurar que el modelo es efectivo en términos de seguridad, como indican las predicciones de algunas firmas de investigación, como *Gartner* o *Trend Micro*. Se plantean que los despliegues de dispositivos *IoT* en diferentes entornos prometen ser masivos y de gran alcance en los próximos años, lo que va unido a la falta de gestión y regulación al respecto, en clara alusión a la tendencia de rápido progreso de esta tecnología, que supera en ocasiones al concepto de seguridad que se debe aplicar, y donde el diseño en origen de los propios dispositivos no contempla estas cuestiones y ha de ser motivo de preocupación real a corto plazo.

2. Estado del arte

2.1. Marco teórico de la tecnología *Blockchain*

Podríamos enmarcar esta tecnología dentro de lo que se conoce como las Tecnologías de Registro o Contabilidad Distribuida (*Distributed Ledger Technologies*), que hacen uso y combinan otras técnicas como la criptografía, el sellado de tiempo o las redes distribuidas, permitiendo a los dispositivos conectados a una red gestionar la información entre todos ellos de forma que se mantenga un registro distribuido en una cadena enlazada de bloques, de forma descentralizada y sincronizada, en sustitución de las tradicionales bases de datos, y que permiten trasmitir y guardar la información de forma segura haciendo prácticamente imposible su alteración, lo que se conoce como concepto de inmutabilidad y donde además se puede asegurar la identidad digital. (CTIC, 2017)

Blockchain hace uso de los principios de la criptografía y de la aplicación de configuraciones de bases de datos, donde ciertas funcionalidades pueden replicarse en un sistema coordinado de bases de datos tradicional, mientras que otras funcionalidades sólo son factibles en un verdadero entorno distribuido de cadena de bloques, donde se deben contemplar los tres ejes fundamentales de este concepto: procesamiento y coordinación de datos, registros confiables e inmutables y digitalización de activos. (Brent, 2018)

Procesamiento y coordinación de datos:

Incluye los mecanismos para una optimización en la coordinación de la información y para el consenso distribuido, en los que los datos se facilitan y se transfieren a través de una plataforma tecnológica.

Registro inmutable / confiable de los productos y transacciones:

Basado en el concepto de la digitalización de la confianza, donde la inmutabilidad se usa como sinónimo de confianza cuando se diseña un sistema para que sea beneficioso para sus usuarios, incentivando los comportamientos adecuados, lo que dará como resultado una plataforma robusta.

Digitalización de activos:

Son bienes digitales que irán creciendo a medida que las transacciones lo hagan, donde se debe tener especial atención a la accesibilidad de estos, ofreciendo una plataforma de digitalización que no limite su interacción con un nivel de incentivos adecuados.

Todo esto señala a *Blockchain* como una tecnología candidata a dotar de confianza digital necesaria en los nuevos modelos de Sociedad de la Información, con un carácter distribuido y que prescinde de los intermediarios o terceros a la hora de asegurar sus propios procesos, obteniéndose una fuente de certeza legítima y transparente que permite superar la falta de confianza en los procesos de validación, y que dificultan los posibles intentos de violación de la seguridad disminuyendo al mismo tiempo los valores de riesgo, garantizando la no alteración y confiablidad de los procesos.

2.2. Antecedentes

En la década de los 90 es cuando comienzan a aparecer algunos trabajos sobre soluciones descentralizadas para la realización de pagos electrónicos sin la dependencia o intervención de entidades centrales que supervisen o regulen las transacciones, pero no fue hasta 1998 con la propuesta de *Wei Dai* "*b-money*", cuando se esbozó la primera criptomoneda no rastreable como solución descentralizada para realizar pagos electrónicos, que estaba basada en criptografía de clave pública. (Dai, 1998)

Esto dio paso en 2008 a la publicación de un artículo con el pseudónimo de *Satoshi Nakamoto,* "*A peer-to-peer electronic cash system*", donde se definía el mecanismo para implementar una moneda digital, el *Bitcoin*, basado en el uso de las cadenas de bloques para registrar las transacciones en una red descentralizada. (Nakamoto, 2008)

El 3 de enero de 2009 entró en funcionamiento *Bitcoin* gracias al primer programa de código abierto y accesible por cualquier programador, permitiendo la verificación en su diseño por la comunidad, creándose los primeros *bitcoins*. Bajo el mismo concepto fueron apareciendo en paralelo otras criptomonedas con diferentes objetivos.

En 2013, *Vitalik Buterin,* un programador involucrado en el desarrollo de *Bitcoin,* propuso incorporar la posibilidad de aprovechar este modelo para ejecutar y gestionar aplicaciones en modo distribuido sin la dependencia de un servidor o de una entidad externa, y así poder realizar contratos de forma descentralizada, pero su propuesta no fue aceptada por la comunidad. No fue hasta 2014 cuando un grupo de desarrolladores decidió poner en marcha varios prototipos bajo el proyecto *Ethereum,* para el desarrollo de una plataforma pública distribuida de código abierto basada en la cadena de bloques, que permitía la ejecución de Contratos Inteligentes con la emisión de un *token* llamado "*Ether*", también

usado como criptomoneda. En Julio de 2015 se pone en marcha la primera versión en funcionamiento de esta plataforma. (Navarro, 2018)

En España en 1999, con participación de *FESTE, Fundación para el Estudio de la Seguridad de las Telecomunicaciones*, se esbozó el diseño de lo que algo después sería un proyecto pionero de gestión digital de endosos de títulos cambiarios. Es lo que luego se conocería como el proyecto "*PISTA – FIRMA – Títulos cambiarios electrónicos*", que se desarrolló con el impulso del Ministerio de Ciencia y Tecnología en el año 2002, desplegándose un sistema electrónico experimental de gestión de títulos cambiarios endosables desmaterializados, algo similar en su concepto a la infraestructura *BlockChain,* ya que se pretendía evitar la unicidad de documento, lo que se denomina en el ámbito de *Bitcoin* el doble gasto, y que, entre otros componentes, contaban con los *Prestadores de Servicios de Seguridad (PSS)*, también conocidos como *Prestadores de Servicios de Confianza Digital*, como servicios complementarios de seguridad informática y de gestión de certificados digitales usados en las firmas electrónicas. (Judiciary Blockchain, 2017)

2.3. Trabajos relacionados

En el ecosistema de la Ciberseguridad, el interés actual y las propuestas basadas en la cadena de bloques que podemos encontrar se pueden agrupar en cuatro ámbitos, principalmente apoyados en su mayoría en arquitecturas *Bitcoin* o *Ethereum*, de los que pondremos el foco en los tres primeros:

- Servicios de autenticación.
- Verificación de identidad digital.
- Mecanismos de seguridad informática y protocolos de comunicación.
- Privacidad.

(Muñoz, 2018)

Hay que destacar que estas propuestas se entrelazan abarcando más de un concepto y que la privacidad no es el objetivo principal de aseguramiento de esta tecnología ya que en los despliegues de *Blockchain* públicas los datos contenidos en ellas son accesibles por cualquier nodo que cumpla con los requisitos para formar parte de la cadena.

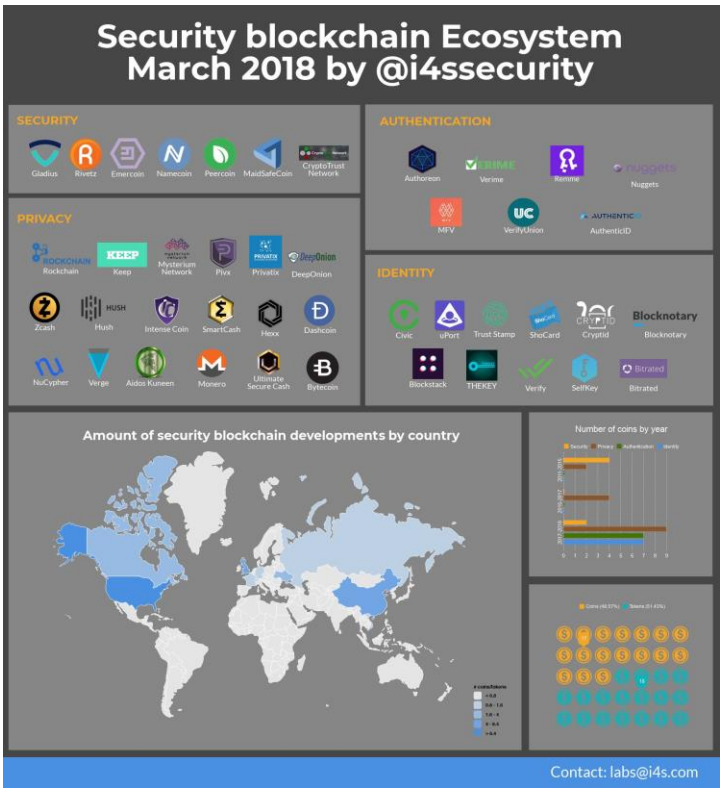

Ilustración 2. Ecosistema de proyectos de ciberseguridad usando tecnología blockchain - I4S Security Lab. (Muñoz, 2018)

2.3.1. Autenticación

Autenticación: procedimiento que asegura la integridad de la identidad, de que algo o alguien es quien dice ser. Se obtiene una vez que se completan los procesos de identificación y verificación de la identidad.

La empresa *Remme Capital Ltd*. presentó este año una solución denominada *REMME* de *PKI* descentralizada, que permite la autenticación sin contraseñas gracias a la generación de certificados propios para cada dispositivo que se almacenan en la cadena de bloques *Ethereum* de forma segura y con un bajo coste de implementación gracias al uso de la plataforma de terceros para la creación y despliegue de *Blockchain* públicas y privadas, llamada *Hyperledger Sawtooth Framework*. A los *tokens* generados para la realización de las operaciones se los denomina *REM*. (REMME CAPITAL LTD., 2018)

El modelo de plataforma descentralizada presentado por *VerifyUnion* propone el concepto de verificación de confianza autenticando a los usuarios incluyendo los perfiles, influencia, e intereses sociales unidos a la identificación digital (*social trust*), que generan una puntuación

del usuario y que utiliza las funciones avanzadas de la tecnología *Blockchain* de *Ethereum,* unidas a la criptografía de clave pública para salvaguardar sus credenciales. Son los propios usuarios los que gestionan su información en un espacio reservado y deciden con quién la comparten sin perder el control sobre ella. (VerifyUnion Team, 2017)

VeriME es otra solución basada en el *token* de *Ethereum* que facilita los mecanismos de autenticación mediante el empleo de biometría e inteligencia artificial para la autenticación de los usuarios por medio de la documentación aportada electrónicamente en los procesos de compras y permitiendo las transacciones seguras entre cliente y proveedor. Este modelo se conoce como *Verification as a service (VaaS),* y pretende ahorrar costes en las aplicaciones profesionales, al mismo tiempo que mejora la experiencia de los clientes. Los *tokens* se conocen como *VME.* (Verime Digital Pte. Ltd., 2018)

2.3.2. Identidad Digital

Identidad digital: prueba que contiene datos identificativos que permiten el intercambio de información con garantías entre titular e interlocutor, como conjunto de datos, atributos y actuaciones de una entidad en el ciberespacio que proporcionan la forma en la que se percibe e identifica en la red.

CIVIC es una propuesta que ofrece verificar la identidad de los usuarios bajo demanda mediante la *Blockchain* de *Ethereum,* donde se introducen datos personales y de registros públicos que se combinan desde diferentes fuentes acreditadas y permitiendo la reutilización de estas identidades, compartiendo las confirmaciones entre los proveedores de servicios. Esto permite reducir los costes y tener un mayor control de los datos de los usuarios, al mismo tiempo que provee de unas identidades digitales confiables. Hace uso de algoritmos de detección de fraude, auditorias y la capacidad de decisión a nivel interno. (Civic Technologies, 2017)

Con el objetivo de cambiar el modelo existente de Internet, que se ha tornado dependiente de las autoridades centralizadas y donde los datos de los usuarios están en muchos casos en los servidores de grandes empresas, haciendo a este ecosistema muy vulnerable a problemas de ciberseguridad, nace la propuesta *BLOCKSTACK* como *Sistema de Nombres de Dominio (DNS)* descentralizado (*Blockchain Name Sistem, BNS*), operando en la cadena de bloques, donde los datos de los usuarios deben permanecer bajo el control de estos y permitiendo gestionar sus propias aplicaciones, aportando mayor confiabilidad general en

los procesos. Como objetivos principales señalan: descubrimiento y designación de nombres descentralizado, almacenamiento descentralizado y rendimiento extremo a extremo comparable con el Internet tradicional. (Ali, Shea, Nelson, & Freedman, 2017)

Bajo el concepto de *Identidad Auto-Soberana (Self-Sovereign IDentity, SSID)* nace el modelo de *SELFKEY* como una arquitectura modular de microservicios desarrollada con tecnología *Blockchain* gracias a *Ethereum Virtual Machine (EVM)*, y orientada al usuario, donde son ellos los que se posicionan en el centro de los procesos de la gestión de identidades, de forma que la privacidad, la transparencia, la seguridad y los derechos individuales sean asegurados. Funciona bajo el mecanismo de confianza que facilita su *token* nativo conocido como *KEY (Key to Encrypt Yourself),* permitiendo intercambiar valores dentro del propio ecosistema de participantes, creando derechos de acceso, unidades de pago y pruebas de reputación. (The SelfKey Foundation, 2017)

2.3.3. Control de acceso

Control de acceso: verificación sobre las entidades que solicitan acceso a los recursos para comprobar que disponen de los derechos necesarios para ello.

En el ámbito de los sistemas de control de acceso cabe destacar el documento "*Blockchain Based Access Control*", donde se propone que los derechos de los usuarios para acceder a los recursos se publiquen como atributos en la cadena de bloques, en forma de políticas de acceso que se transfieren de forma distribuida entre los usuarios públicamente visibles, permitiendo la auditoría distribuida y evitando la posible negación fraudulenta de los derechos otorgados por una política ejecutable en base a su inmutabilidad. También proponen una posible implementación operativa basada en el protocolo *XACML,* basada en *Bitcoin*. (Maesa, Mori, & Ricci, 2017)

2.4. Estado actual

La tecnología *Blockchain* ha demostrado ser resistente a alteraciones y transparente por definición, proporcionando seguridad gracias a su descentralización de la confianza y a su naturaleza distribuida, eliminando el riesgo de posibles errores humanos, prescindiendo de intermediarios y protegiéndose contra ciberataques.

2.4.1. Definición y arquitectura

Blockchain es una base de datos transaccional protegida criptográficamente y organizada en bloques de transacciones relacionados matemáticamente en forma de cadena y en base a una confianza obtenida por consenso entre las partes, cuya información se guarda en un registro que se ubica y gestiona de forma distribuida entre los diferentes nodos de un sistema de red de dispositivos y computadoras para evitar posibles alteraciones, ya que cualquier mínimo cambio rompería esta cadena, asegurando por tanto su integridad.

2.4.1.1. Definición de Blockchain

- Cadena de bloques

  O en ingles *Blockchain*, es el nombre como se conoce al conjunto de elementos y operaciones que actúan bajo el mismo protocolo, que tienen como objetivo el asegurar la veracidad de la información que contienen los bloques, y que es la propia base de datos diseñada para almacenar una gran lista de registros creciente, realizados y ordenados por el sistema, que se replican y sincronizan por la red de nodos que lo forman.

2.4.1.2. Elementos de Blockchain

- Protocolo de comunicación

  Necesario para que exista un estándar común de comunicación en forma de protocolo que permita al sistema el intercambio de mensajes en base a unas reglas predefinidas, como es el protocolo *TCP/IP,* ampliamente usado en Internet.

- Red distribuida

  Basada en las redes descentralizadas donde no existe una única entidad o servicio que controla al resto, pero que, además permite conexión entre todos los nodos de la red, permitiendo así que la información esté siempre disponible y eliminando los posibles puntos únicos de fallo, asegurando la disponibilidad.

- Nodos

  Son cada uno de los dispositivos conectados con capacidad de cómputo que se encargan de verificar, almacenar y propagar de forma independiente una copia actualizada de la base de datos o libro mayor, haciendo cumplir las reglas establecidas.

- Transacción

  Acción mediante la que se registra el intercambio de valor o activo digital entre dos partes y que se almacena en la base de datos en forma de entradas y salidas firmadas digitalmente, similar a los libros mayores contables.

- Bloque

  Conjunto de transacciones empaquetadas que se firman digitalmente y que están formados por la propia información de las transacciones, que con la acción de los mineros incluye también el resultado de realizar la función resumen (*hash*), un identificador al bloque anterior, una marca de tiempo y un identificador que servirá de enlace con el bloque siguiente, permitiendo así su enlazamiento lineal (cadena de bloques), garantizando su integridad e inalterabilidad.

- Mineros

  Cualquier dispositivo o nodo participante de la red que esté encargado de comprobar, validar, registrar y propagar los bloques tras la resolución de desafíos matemáticos del protocolo de consenso para determinar su validez, pudiendo recibir incentivos del sistema por esta labor en forma de activos o *tokens*.

- Libro mayor

  Registro común donde se anotan todas las transacciones digitales realizadas por bloques bajo procedimientos de consenso de una mayoría de los participantes, que verifica su integridad al objeto de evitar que se dupliquen las operaciones y que se acciona en ausencia de cualquier autoridad central de control.

- *Token*

  Unidades de cuenta que son la representación de un activo digital en forma de serie de dígitos que representan de manera única un registro o transacción verificada y aceptada por el consenso común como resultado de las operaciones en las

aplicaciones descentralizadas dentro de la cadena de bloques. *ERC-20* de *Ethereum* es un ejemplo.

2.4.1.3. Técnicas clave

- Criptografía

  Conjunto de herramientas matemáticas, técnicas y algoritmos que con el uso de una o más claves permiten cifrar la información registrada protegiéndola y dotándola de confidencialidad e integridad. En *Blockchain* se usa principalmente criptografía simétrica o de clave secreta, criptografía asimétrica o de clave pública para el cifrado y algoritmos de *hashing* para obtener la huella digital.

- Consenso

  Protocolo común entre las partes de la red de nodos que verifica y confirma las transacciones realizadas, asegurando su inmutabilidad y proporcionando copias actualizadas y no alteradas de todas las operaciones realizadas en la cadena de bloques.

2.4.2. Redes y computación distribuida

Esta tipología de red está basada en la eliminación de posibles filtros en la comunicación entre los nodos que la componen, es decir, que todos están interconectados y cualquiera puede hacer funciones de emisor o receptor de la información contenida en los otros nodos, manteniendo este funcionamiento incluso si algunos nodos fallan o están limitados, manteniendo la fluidez de la comunicación global del sistema.

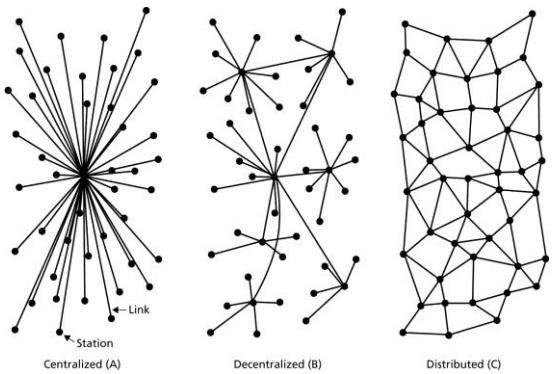

Ilustración 3. On Distributed Communications, Memorandum (Baran, 1964)

2.4.2.1. Machine to Machine (M2M)

Este término se refiere al intercambio de comunicación directa entre dispositivos que utilizan cualquier recurso de red de comunicaciones disponible para establecer conexiones con una infraestructura de aplicaciones remotas e interactuar entre máquinas o con el entorno que las rodea mediante sensores y actuadores, cuyo objetivo es simular un funcionamiento similar al que se produciría si todas estas máquinas estuvieran conectadas de forma directa; es como veremos más adelante, el precedente de lo que hoy se conoce como *Internet de las Cosas* o *Internet of Things (IoT).*

2.4.2.2. Peer to Peer (P2P)

Conocida como red de pares o red entre iguales, se refiere a las redes de dispositivos que basan su funcionalidad en la ausencia de clientes y servidores, en las que todos los elementos son nodos que se comportan como iguales entre sí permitiendo el intercambio directo de la información contenida en ellos, administrando el uso de la red de comunicación de manera óptima. En nuestro caso de estudio nos centraremos en el modelo puramente descentralizado sin gestión central usado en *Blockchain*.

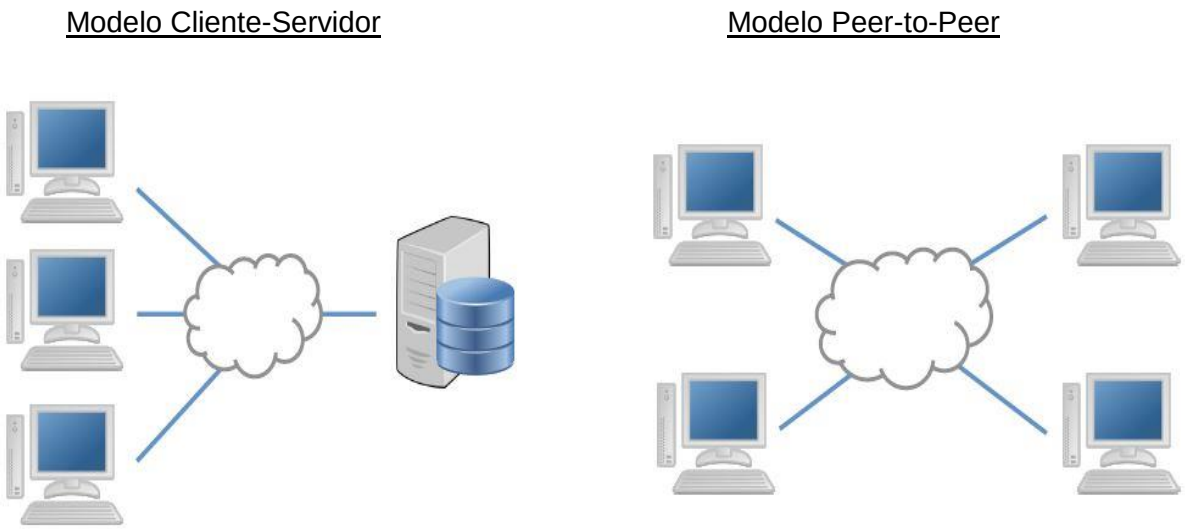

Ilustración 4. Peer-to-peer networks (Stoica et al., 2016, p. 3)

2.4.2.3. *Distributed Ledger Technology (DLT)*

Tecnologías de registro contable universal y distribuido que gestionan una base de datos compartida sin autoridades centrales de control, dotándolas de total transparencia y donde todas las transacciones están públicamente disponibles, dificultando su manipulación o su deterioro. La forma en que se estructure, distribuya y acuerde esta base de datos, establecerá el tipo de *DLT*; de hecho, *Blockchain* es un tipo específico de *DLT*.

2.4.2.4. *Cloud Computing*

Traducido como computación en la nube, es un modelo que ofrece las posibilidades de los sistemas informáticos mediante servicios alojados en *Centros de Procesos de Datos (CPD)*, que son publicados normalmente en Internet bajo suscripción y que ofrecen un catálogo de recursos y servicios flexibles y adaptativos, abstrayendo a los usuarios de la mayoría de especificaciones de arquitectura y distribución de los sistemas reales desplegados. Modalidades como *SaaS, PaaS* o *IaaS*, permiten adaptar el modelo a las necesidades de los usuarios de manera ágil, escalable, segura y de bajo costo en comparación a una implementación similar pero íntegramente física de manera local. Gran capacidad de computación y almacenamiento, pero de forma centralizada. *Blockchain* intentará competir con este modelo en el futuro con lo que se comienza a denominar *Distributed Cloud*.

2.4.3. Tipologías en *Blockchain*

Aunque en sus comienzos en 2009 *Bitcoin* no tenía restricciones a cualquier participante que desease formar parte del sistema, en la actualidad, dependiendo de si la implementación de *Blockchain* está abierta a la participación, o por el contrario está restringida a algunos usuarios, se pueden dividir en dos grandes grupos: públicas y privadas.

Existe también una implementación que une ambos conceptos como vamos a ver es el caso de las *Blockchain* híbridas.

2.4.3.1. Blockchain Pública (Permissionless)

Basado en el concepto de red descentralizada que pretende eliminar el problema del doble gasto; este modelo permite que cualquiera en el mundo (aunque no sea participante), pueda acceder y consultar todas las transacciones que estén en la cadena de bloques, del mismo modo que cualquiera puede convertirse en participante de forma abierta si lo desea, siendo todos estos participantes iguales para el sistema, como así son los nodos involucrados. El único filtro vendría de la cierta anonimidad de la que pueden hacer uso los participantes en lo que a sus datos personales se refiere. *Bitcoin* y *Ethereum* son los ejemplos más conocidos de *Blockchain* públicas.

2.4.3.2. Blockchain Privada (Permissioned)

Es un concepto posterior que significa que no está abierta al público en general, ya que sólo se puede acceder a su sistema mediante invitación previa a los usuarios, que quedan identificados, y donde se pueden establecer niveles de acceso, que normalmente vienen exigidos por requisitos regulatorios. Por tanto, sólo los usuarios registrados pueden consultar o registrar las transacciones de la cadena de bloques en base al perfil que posean. El número de nodos que pueden participar también está regulado y existen niveles de compromiso y anonimato que se deben cumplir para velar por la confidencialidad y estabilidad del sistema. *Hyperledger* de la *fundación Linux*, *Ripple* como protocolo de transferencia internacional de dinero, o *R3*, propiedad de un consorcio internacional de bancos, son algunos ejemplos de *Blockchain* privadas.

2.4.3.3. Blockchain Híbridas

Último modelo en aparecer, que es combinación de los dos anteriores, donde se invitan a los nodos participantes para asumir el compromiso establecido para su mantenimiento y demás restricciones de consenso, pero se dejan las transacciones de manera pública visibles para todo el mundo, no así su contenido. *BigchainDB* y *Evernym* son ejemplos de este modelo de *Blockchain* híbridas.

2.4.3.4. Cadenas laterales (Sidechains)

Evoluciones experimentales propuestas por parte de los desarrolladores de la tecnología *Blockchain* que pretenden mejorar y optimizar sus características de base o añadir nuevas funcionalidades con soluciones que pretenden aumentar su potencial.

Las *Sidechains* son *Blockchains* independientes en las que se puede experimentar las nuevas funcionalidades sin riesgo para los modelos consolidados, como nuevos diseños en las transacciones, nuevos modelos de confianza o funcionalidades criptográficas añadidas, que en muchos casos no pasan del marco teórico debido a que su implementación a nivel práctico supondría grandes cambios o bifurcación (*Hard Fork*), sobre los códigos fuente de las cadenas de bloques principales.

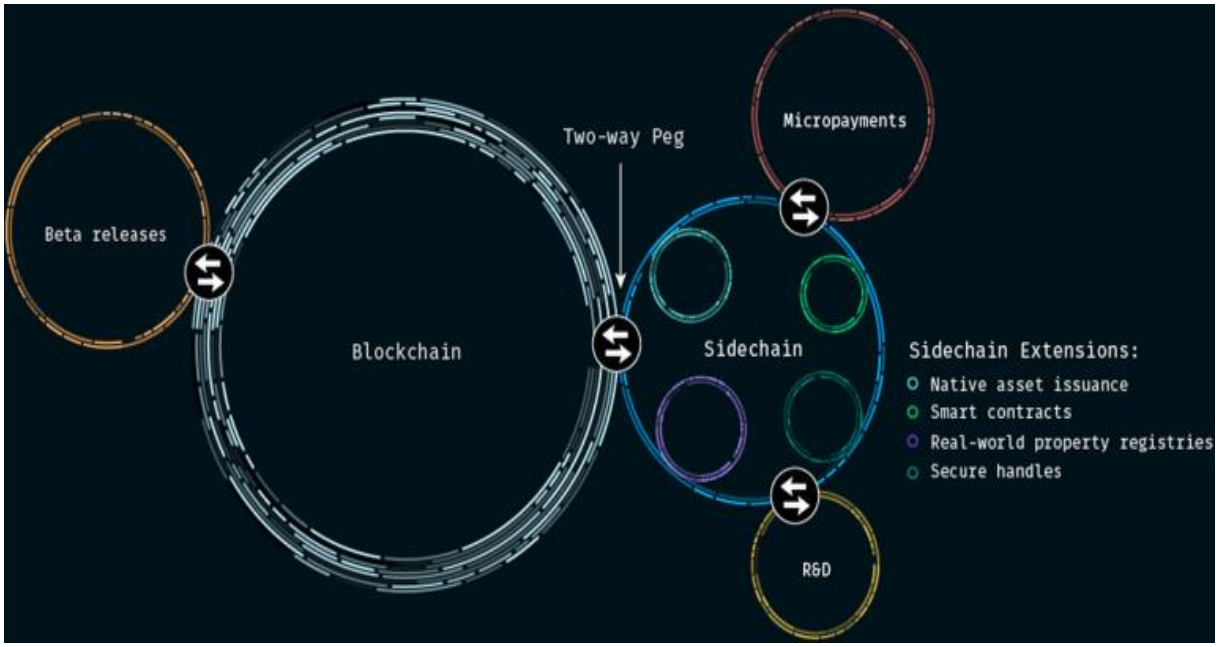

Ilustración 5. Blockstream y Sidechains: ¿Qué es una Sidechain o cadena lateral en Bitcoin? (Preukschat, 2014)

Un ejemplo de *Sidechain* de *Bitcoin* lo tenemos en el proyecto *Blockstream,* que añade una nueva pieza "*two-way-peg*" como conector entre las dos cadenas y la minería combinada "*merge mining*", donde los hashes son enviados entre las dos cadenas. En *Ethereum* tenemos *Lisk* como plataforma de código abierto para desarrollar y ejecutar contratos inteligentes mediante aplicaciones descentralizadas multiplataforma que utilizan el lenguaje de programación *Javascript.* (Back et al., 2014)

2.4.4. Tratamiento de datos

2.4.4.1. Bases de datos descentralizadas

A diferencia de las Bases de datos (*BBDD*) centralizadas, que residen en una sola ubicación que hace de servidor de datos, en las descentralizadas existen varias ubicaciones repartidas geográficamente, donde se guardan los datos en sus correspondientes servidores de procesamiento, que actúan como un grupo de *BBDD* independientes sin estar totalmente interconectadas entre todas ellas. Al no depender totalmente de un solo servidor no adolecen de un solo punto de fallo, por lo que son más estables y más rápidas que las centralizadas. Un ejemplo de este tipo de *BBDD* lo tenemos en *Informix* de *IBM*, que distribuye las operaciones sobre los datos entre diferentes servidores proporcionando un alto rendimiento y seguridad.

2.4.4.2. Bases de datos distribuidas

Este modelo sí actúa como una sola *BBDD* a nivel lógico, aunque su ubicación esté distribuida en diferentes lugares geográficos y en diferentes plataformas o servidores, pero con total interconexión entre ellas mediante las redes de comunicación, lo que elimina posibles cuellos de botella o los únicos puntos de fallo. Al almacenar la información en la totalidad de sus nodos que replican la información, la estabilidad e integridad del sistema están prácticamente garantizadas. En términos de seguridad también aporta beneficios, ya que el impacto de los posibles accidentes o ataques deliberados siempre estarían delimitados a los nodos comprometidos, cuya información seria invalidada por consenso.

Blockchain es el mejor ejemplo de este concepto, donde los nodos actuarían como servidores y clientes de esa gran *BBDD* que está almacenada a lo largo de toda la cadena de bloques, pero que va un paso más allá de las BBDD distribuidas, ya que la lógica del control de acceso para las operaciones de lectura y escritura de los datos queda totalmente descentralizada, y además estas operaciones, al ser tratadas como transacciones, quedan integradas, aseguradas y consensuadas, permitiendo que la gestión de la *BBDD* se realice de forma autónoma.

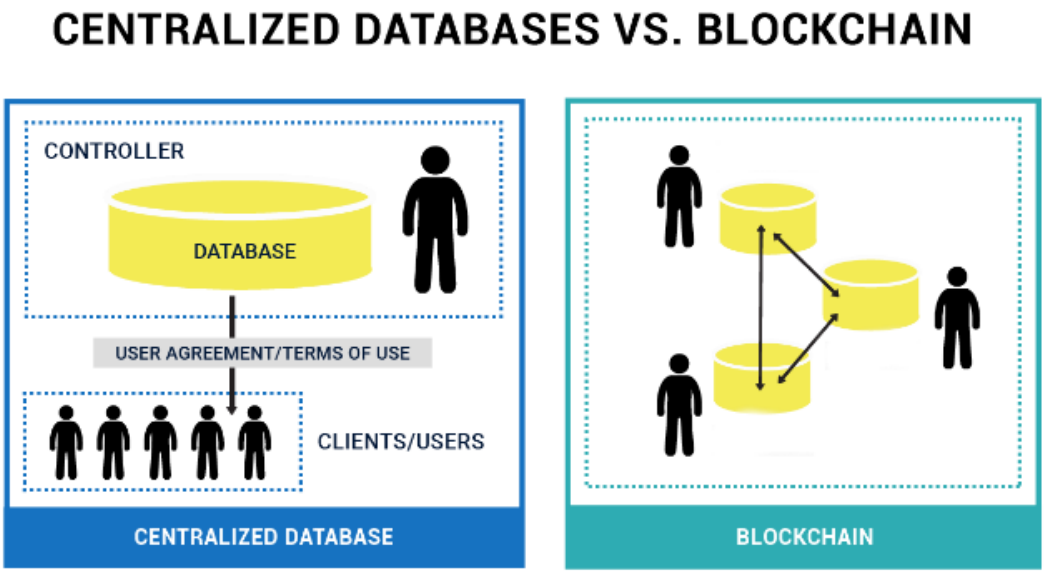

Ilustración 6. Differences between Blockchain and Distributed Databases. (Singh, 2017)

2.4.5. Uso de la Criptografía

Para comprender el principio de aseguramiento de la información en base a su integridad y del que provee *Blockchain,* es necesario realizar un acercamiento a los elementos criptográficos de los que dispone como fundamentos básicos de su funcionamiento, concretamente al proceso de firma digital, al objeto de que sólo los participantes legítimos y en disposición de las claves criptográficas apropiadas puedan editar las transacciones que les corresponden dentro de la cadena de bloques, y que éstas queden perfectamente verificadas y puedan pasar a la fase de sincronización del sistema asegurando el proceso.

Blockchain por sí sola no provee confidencialidad ni privacidad en los procesos, sino que su aportación se basa en la integridad y disponibilidad (esta última con matices), y que para conseguirlas es necesario añadir otras capas de seguridad en forma de implementaciones específicas, como las de las *Blockchain* privadas, que añaden mecanismos adicionales mediante controles, como el cifrado simétrico y de gestión descentralizada de certificados digitales y estándares de intercambio de datos de autenticación y autorización tipo *SAML* asegurados criptográficamente bajo protocolo *SSL/TLS* en la capa de transporte.

2.4.5.1. Algoritmos criptográficos

Blockchain hace uso principalmente del proceso de firma digital, en base a los algoritmos y procesos criptográficos típicos de esta disciplina y en función de la operación que se esté realizando.

- Algoritmos de cifrado simétrico:
  Comúnmente conocidos como de clave secreta, porque hacen uso de una sola clave privada conocida por los extremos que funciona en los dos sentidos, es decir, sirve tanto para cifrar como para descifrar un mensaje, siendo eficientes y rápidos en su desempeño. El reto de este concepto radica en encontrar el mecanismo para la entrega de esta clave a ambos extremos de forma segura.
  En general, este tipo de algoritmos criptográficos no está integrado en *Blockchain,* aunque se puede utilizar como medida de protección adicional en implementaciones de redes específicas que los combinan con otros algoritmos que sí están integrados, para dotar de confidencialidad a los procesos de las aplicaciones y asegurar a nivel de red los protocolos de comunicaciones. *DES, IDEA, AES,* son algunos ejemplos.

- Algoritmos de cifrado asimétrico:
  Conocidos como de clave pública, ya que disponen de dos claves vinculadas matemáticamente de forma inversa (entropía y reversibilidad), y con distinto cometido, donde una clave es secreta o privada y la otra se permite que sea conocida públicamente. El propietario de la clave privada generada aleatoriamente puede calcular la clave pública, pero cualquiera que conozca esta clave no podría calcular la privada, siendo computacionalmente improbable; por tanto, cada extremo posee dos claves inversas entre sí, de forma que lo que una cifra, la otra lo descifra. Son más complejos y lentos y se encargan de compartir de forma segura la clave secreta del anterior algoritmo de cifra simétrica, firmar digitalmente las transacciones (junto con la operación *hash*) y verificarlas.
  En las transacciones que se propagan entre los nodos de una *Blockchain* se propone la firma digital como una forma segura de demostrar la identidad del emisor de la operación, donde cada usuario crea un identificador propio en el sistema (dirección), y demuestra su propiedad firmándola y haciendo uso de su clave privada asociada (sólo conocida por él), y que sólo se puede verificar con la clave pública asociada (conocida de forma pública por cualquier nodo), siendo esta generación y verificación de firmas un proceso lento y computacionalmente complejo. *RSA, Diffie-Hellman, Curvas Elípticas (ECDSA)*, son algunos ejemplos.

- Cifrado híbrido:
  Hace referencia a la combinación del cifrado simétrico y el asimétrico para la realización de procesos criptográficos seguros, como es claro ejemplo el protocolo *TLS/SSL* que usamos cotidianamente para navegar por Internet mediante el uso de certificados, y que viene a paliar la mala gestión de claves de la que adolece el cifrado simétrico, añadiendo la posibilidad de firmar las transacciones y aprovechando la potencia de tasa de cifrado de contenidos extensos de datos que sí posee.

2.4.5.2. Operaciones criptográficas

Aquí es donde se demuestra la integridad de los bloques y de las transacciones, garantizando la no manipulación de sus datos gracias a las funciones *HASH* criptográficas de 256 *bits* o funciones resumen (*digest*), que carecen de claves y no se consideran algoritmos de cifra. Son funciones matemáticas computables, pero que permiten autenticar los mensajes al añadir una huella digital única y de tamaño fijo de un contenido de tamaño muy superior al mensaje, lo que permite al receptor comprobar su integridad, ya que cualquier mínimo cambio en el mensaje de origen impactaría en el resultado de este *hash,* haciéndolo completamente distinto por comparación y evitando la modificación no autorizada de contenidos. *MD5, SHA-1, SHA-2, SHA-3*, son algunos ejemplos.

La firma digital avanzada es la combinación de la huella digital (*Hash*) más el uso de la criptografía asimétrica, permitiendo una prueba de integridad, la verificación de la identidad y el no repudio.

2.4.5.3. Árboles de Merkle (Merkle trees)

Llamados así en honor a *Ralph Merkle,* que los planteó por primera vez en 1979, son también conocidos como árboles *hash* binarios y son parte fundamental de la tecnología *Blockchain,* usados tanto por *Bitcoin* como en *Ethereum* para mantener la integridad y validez de los datos, permitiendo resumir mediante funciones *hash* (normalmente *SHA-2*) todas las transacciones de un bloque, y creando una huella digital de todo el conjunto que se almacena en su encabezado y permite verificar rápidamente que una entrada determinada se haya incluido en un conjunto de datos en particular, y en qué orden.

Es una estructura en árbol de datos de validación que se crea en sentido ascendente y que se basa en una generalización de las listas y cadenas *hash*. Cada nodo hoja está etiquetado con el *hash* de un bloque de datos de transacciones individuales conocidas como "*Transaction ID*", donde cada nodo no terminal está etiquetado con el *hash* realizado sobre un par de hashes de las etiquetas de sus nodos hijos, recorriendo así el árbol de abajo hacia arriba hasta que se llega al nodo raíz del árbol de *Merkle,* también llamado *Root Hash* o *Merkle Root.* Es decir, la raíz de *Merkle* es el *hash* de todos los *hashes* de todas las transacciones en el bloque, permitiendo una verificación cronológica, eficiente y segura de los contenidos de las estructuras de datos de gran tamaño, que al representarlos sólo con los datos de validación reducen significativamente el uso de memoria, espacio de almacenamiento y capacidad computacional que se debe utilizar, reduciendo al mismo tiempo la cantidad de información transmitida a nivel de red. (Shaan, 2017)

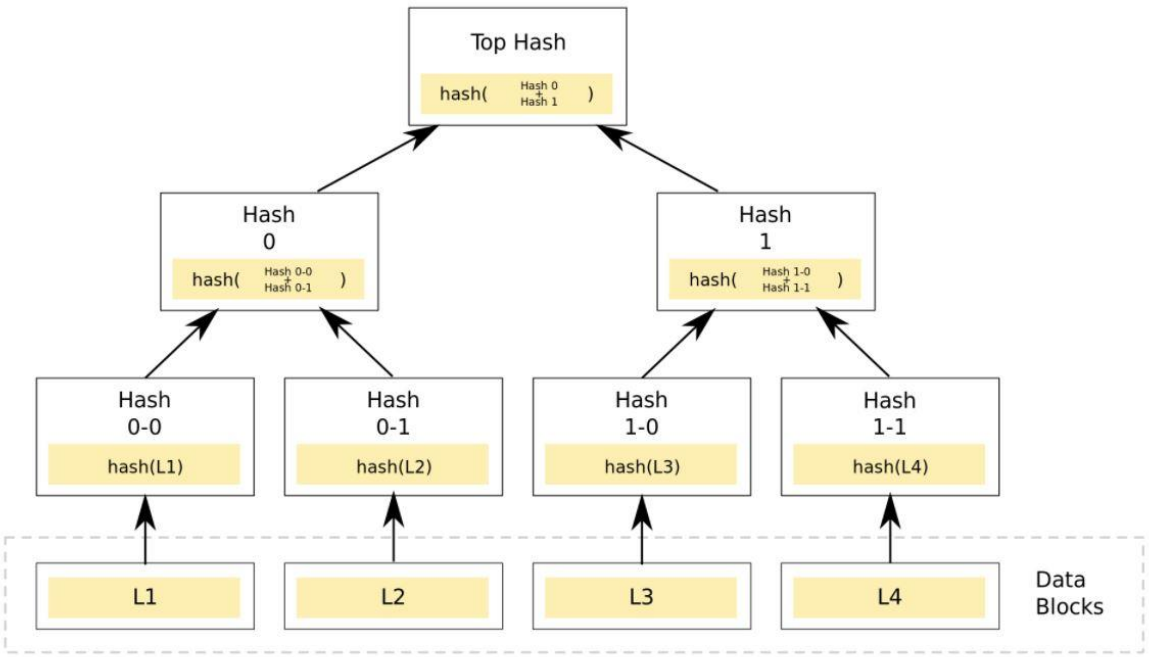

Ilustración 7. Merkle Trees—Introduction to Blockchain. (Suraj, 2017)

Los árboles de *Merkle* normalmente se representan como árboles binarios de hasta 2 hijos, pero en realidad se pueden implementar como árboles n-arios, mientras contengan un número par de nodos hojas: en el caso de *Ethereum* se usan tres árboles diferentes por cada bloque, uno para las transacciones, otro para el estado y el último como recibo de las transacciones.

Los árboles de *Merkle* garantizan que los bloques de datos que se intercambian los pares de la red de la cadena de bloques sean siempre los originales sin alteración ni corrupción de su información. Si se cambiase un solo bit de datos de la cadena de bloques, se modificaría el árbol en toda su altura desde el "*Transaction ID*" correspondiente hasta la raíz del árbol de *Merkle,* ya que es una de las cualidades de las funciones *hash*, lo que unido a la distribución de los datos de los bloques por toda la red y los mecanismos de consenso, hace que el proceso de verificación sea totalmente fiable y los datos permanezcan inmutables y resistentes a cualquier modificación.

2.4.5.4. Tokens y sus tipos

Representan a un activo o recurso digital transferible que de forma estandarizada pueden ser cantidades determinadas de bienes o servicios que se definen como entradas al libro mayor y que utilizan un conjunto de funciones para ser construidos al final de la cadena de bloques y poder operar en las reasignaciones de propiedades o derechos en la plataforma, pudiéndose comprar, vender, intercambiar, subdividir, etc., es decir, son interoperables.

Aparecieron con el ecosistema *Ethereum* representando los derechos sobre los activos o recursos digitales, que generalmente hacen uso del estándar conocido como *ERC20,* que define una lista de reglas comunes para los *tokens* creados sobre *Ethereum*, aunque existen otros como *ERC223 o ERC721*. Ejemplos: *TRON (TRX), ICON (ICX), OmiseGO (OMG), EOS*, etc.

- *Token* de uso

Son requeridos para el acceso y uso de algún servicio dentro de la cadena de bloques, como la creación de valores.

- *Token* nativo de trabajo

Forman parte del núcleo de *Blockchain* y permiten a los usuarios el derecho a contribuir con su trabajo de validación de bloques dentro de la red distribuida recibiendo incentivos (mineros).

Network	Coin	Issuance	Block-making incentive
Bitcoin	BTC	Created according to schedule. Total 21 million BTC in 2140.	Block reward + transaction fees
Ripple	XRP	100% pre-mined. 100 billion XRP created.	None
NXT	NXT	100% pre-mined. 1 billion NXT created.	Transaction fees
Ethereum	ETH	72 million pre-mined plus ongoing issuance of 18 million ETH per year.	Block reward + computation fees

Ilustración 8. A selection of distributed ledger systems and their intrinsic tokens. (Lewis & Bits on Blocks, 2015)

- *Token* de aplicación

Típicos en la capa de aplicación del ecosistema *Ethereum.* Son emitidos mediante las Aplicaciones Distribuidas (*DApps*), creadas por los desarrolladores mediante la capacidad de crear aplicaciones autoejecutables sobre la cadena de bloques de *Ethereum*.

2.4.6. Funcionamiento

Básicamente podemos describir el funcionamiento de la cadena de bloques como la creación e intercambio de transacciones distribuidas entre los nodos que forman la red para la creación de un libro de cuentas.

2.4.6.1. Ciclo de vida de las transacciones

Una vez que tenemos los nodos identificados criptográficamente con sus direcciones válidas y uno de ellos quiere registrar una transacción en la cadena de bloques, se produce el siguiente proceso:

- El nodo origen que solicita una transacción genera su contenido y le añade su identificador único (dirección), junto con su clave pública.

- Si esta transacción implica transferencia de activos (valores, información, contratos, etc.) a otro u otros nodos, añadirá las direcciones de los nodos destino en la propia transacción (cabecera).
- Se firma toda la transacción con la clave privada del nodo origen.
- Se envía la transacción mediante la red de pares (*P2P*), propagándose rápidamente entre todos los nodos.
- Los nodos destino comprueban que no habían recibido esta transacción anteriormente y verifican mediante los algoritmos acordados la autenticidad de la firma del nodo origen y la validez lógica de la transacción, repitiendo el proceso de propagación de la transacción por la red de pares.
- Una vez verificada la transacción por un número suficiente de nodos (51%), es empaquetada junto con otras transacciones ya verificadas (*pool de transacciones*) para crear un nuevo bloque de datos por los mineros.
- El bloque se añade a la cadena de bloques existente de forma permanente e inalterable, propagándose por la red de pares.
- La transacción se puede dar por completada.

(Lage & Berrocal, 2017)

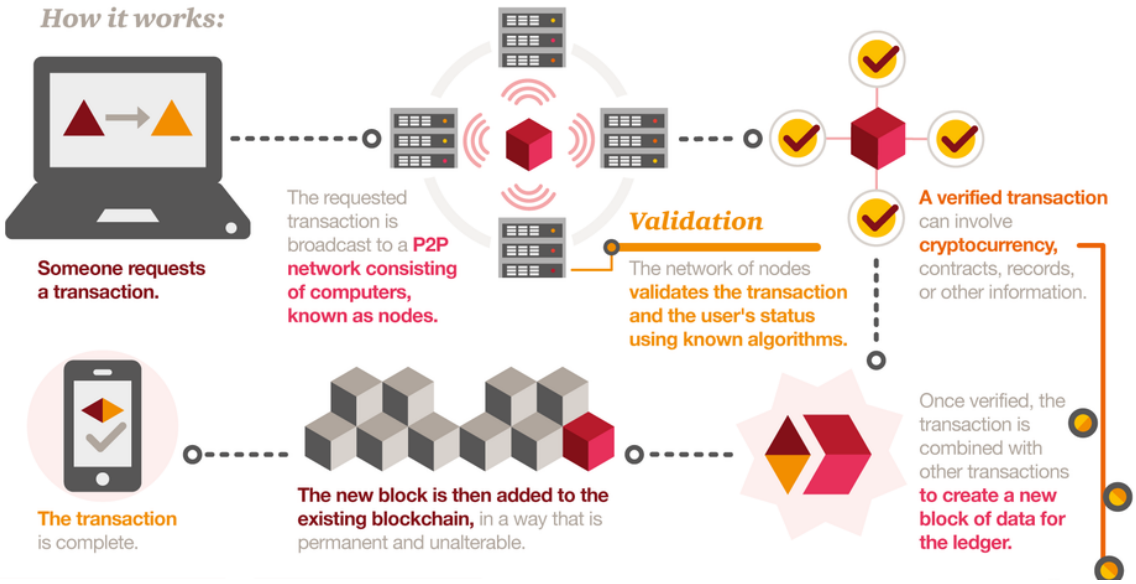

Ilustración 9 - A look at blockchain technology. (Morrison & Sinha, 2016)

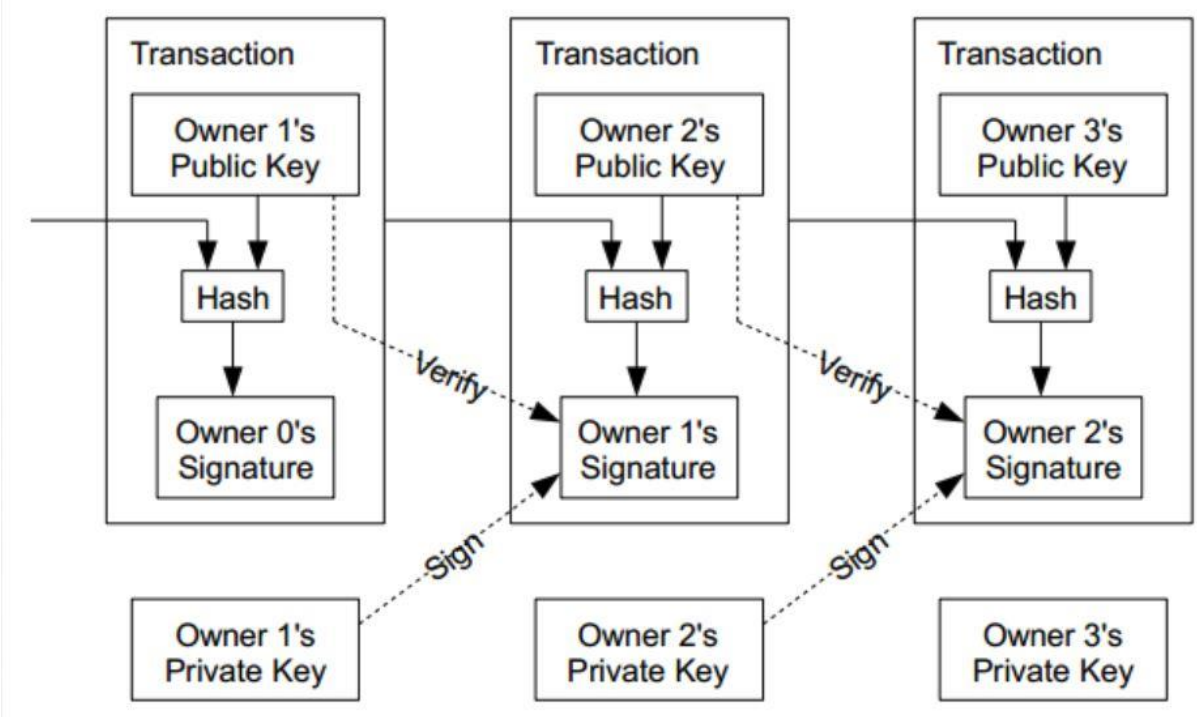

Ilustración 10. Transacciones en BlockChain. (Salvachúa, Quemada, & Alonso, 2016)

2.4.6.2. Minería

La descentralización es una de las características de *Blockchain*, donde no existen servidores centrales y cualquier nodo conectado es válido para procesar y verificar transacciones gracias a la colaboración de los usuarios. (Ochoa, 2018)

Se basa en el concepto de que los nodos que van a procesar los bloques para registrar las transacciones contenidas en él, deben realizar algún tipo de trabajo matemático variable con un coste computacional factible y que pueda ser fácilmente verificado por el sistema para llegar a un acuerdo de forma descentralizada y permitir la agregación de estos bloques validados a la cadena, haciéndolos oficiales.

Los mineros compiten por alcanzar la solución al problema propuesto pudiendo obtener algún tipo de recompensa o incentivo dentro de la cadena de bloques al obtener un bloque registrado, lo que dota de seguridad al sistema y evita la aparición de bloques defectuosos o inconsistencias, garantizando que todos los participantes dispongan de los mismos datos de registro de las transacciones distribuidos por la red.

Al no permitirse transacciones inválidas en un bloque se evita el problema del doble gasto, siendo la extracción de transacciones pendientes realizada mediante el cálculo de un *Hash*

especial y, por tanto, un nuevo bloque oficial es creado cada 10 minutos por el primer minero capaz de calcular dicho *hash*.

El conocido como ataque del 51% se podría llevar a cabo si un grupo de mineros controlara más de la mitad de los nodos de la red, pudiendo así invalidar transacciones oficiales y generar el temido doble gasto.

2.4.6.3. Tipos de nodos

Los nodos son la columna vertebral del sistema *Blockchain* desde el enlace en las comunicaciones hasta la distribución del consenso. Se dividen en dos tipos:

- Nodos Completos (*Full Nodes*)

Descargan todas las transacciones de los otros bloques.

Verifican la validez de todas las transacciones de los bloques.

Ejecutan la algoritmia del consenso establecido.

Generan los nuevos bloques.

- Nodos Ligeros (*Lightweight Nodes*)

No realizan todas las funciones de los Nodos Completos.

Descargan sólo pequeñas partes de la cadena de bloques relacionadas con usuarios en particular.

Conectan con los nodos completos de los que reciben la información básica sobre la cadena.

Más apropiados para entornos de dispositivos *IoT, D-PKI*, etc.

2.4.7. Operaciones de consenso

En *Blockchain* existen una serie de reglas codificadas como protocolos, que indican a los nodos cómo validar los bloques de transacciones de una manera consensuada para todos igual, y que permiten verificar que la estructura de los bloques junto con sus transacciones sean las adecuadas. Es lo que se conoce como el sistema distribuido de verificación de las transacciones o *Transaction Script*.

Son los mecanismos que ejecutan la algoritmia de consenso que permite confirmar las transacciones, garantizando su integridad e inalterabilidad para que se puedan propagar de forma sincronizada por toda la red de nodos *P2P* de la cadena de bloques, distribuyendo la base de datos sin necesidad de una lógica o administración central, al actuar todos como una entidad de garantía y permitir la gestión del estado del sistema de una forma determinista. El objetivo de cada nodo encargado de las tareas de procesado de bloques es alcanzar un consenso con el resto en base a las reglas establecidas, donde se tiene en cuenta cada transacción con sus correspondientes pruebas de validación y autorización, ejecutando el algoritmo propuesto que determine qué bloques son válidos para agregarse a la cadena de bloques, y compartiendo esta información, que debe ser validada y compartida a su vez por el resto de los nodos involucrados de forma independiente.

2.4.7.1. Prueba de trabajo (Proof of Work, PoW)

Protocolo de consenso utilizado en *Bitcoin* y *Ethereum* que basa su seguridad en la resolución de problemas matemáticos de gran potencia computacional para limitar la velocidad en la que los bloques sean generados por la red (10 minutos de media) y que nace como necesidad de evitar que una mayoría de nodos pudieran mantener comportamientos indeseados en base a intereses comunes, o que los nodos registren bloques al azar, así como para establecer un mecanismo de información para que la cadena sepa qué nodos han conseguido el registro satisfactorio de un bloque y qué bloques son válidos, existiendo un acuerdo para el registro de las transacciones y evitando que estos no se puedan alterar.

Consiste en operaciones matemáticas que se deben realizar a los bloques. Son las funciones *hash*, normalmente *SHA-256*, donde se irán probando pequeñas variaciones a la

entrada (el encabezado del bloque), hasta encontrar una válida que nos devuelva el valor *hash* que se está buscando, convirtiéndose en su identificador en la cadena de bloques.

Básicamente cada bloque está compuesto de un encabezado que lo describe seguido por el cuerpo con la información de las transacciones contenidas en él. Este encabezado con datos relativos a la versión del protocolo, con *hashes* de otros bloques (punteros), con los *hashes* de las transacciones, una marca de tiempo y el "*nonce*" o valor de la dificultad del minado del bloque, que se modifica a medida que se producen intentos de conseguir valores *hash* válidos del bloque completo. Normalmente los mineros deben hacer muchos intentos de cálculo de hash variando este "*nonce*" hasta que obtienen un valor *hash* en particular que comienza con múltiples ceros y que encaja con un espacio objetivo, que se considera como solución al desafío matemático y que genera un bloque minado que se integra oficialmente en la cadena de bloques. (Soto, 2017)

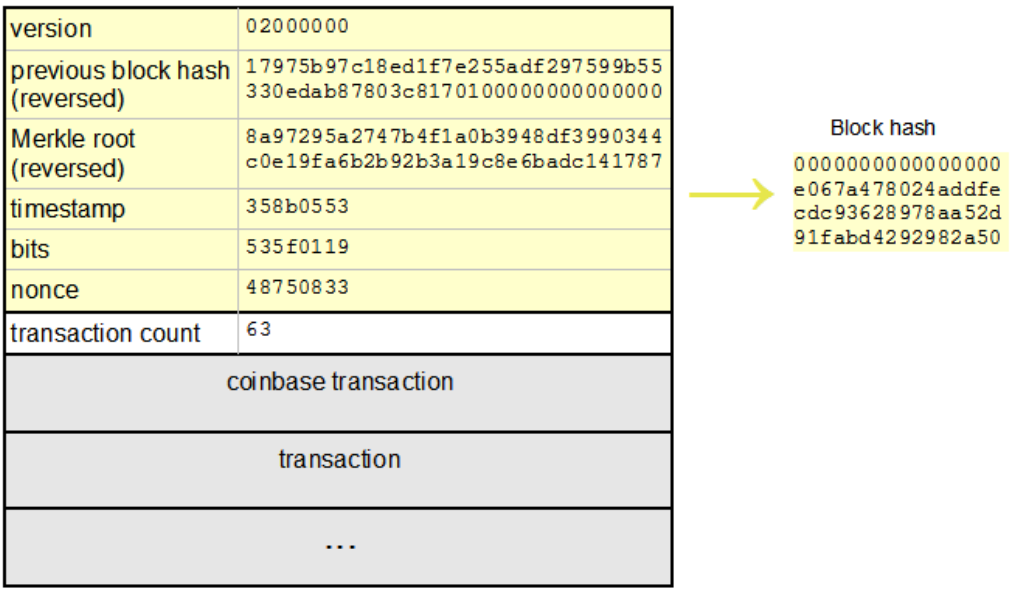

Ilustración 11. Estructura de un bloque Bitcoin.(Soto, 2017)

La gran mayoría de las veces el resultado del *hash* no es el que se busca, así que se modifica ligeramente la cabecera del bloque (*nonce*) y se vuelve a intentar, así tantas veces como sean necesarias hasta encontrar el valor de *hash* adecuado, lo que implica un esfuerzo computacional considerable que proporciona un bloque nuevo minado aproximadamente cada 10 minutos.

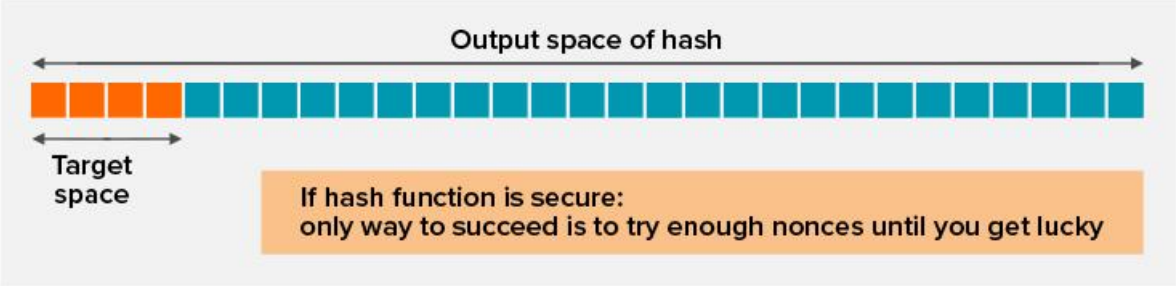

Ilustración 12. Hashing en PoW. (Chiner, 2017)

La desventaja de este protocolo está en la necesidad de enormes recursos computacionales y, por tanto, de un gasto económico, e incluso ambiental, en los procesos de minería; la ventaja es la seguridad que dota al sistema, siendo también muy costoso romperla.

2.4.7.2. Prueba de participación (Proof of Stake, PoS)

Al igual que el protocolo anterior, éste busca conseguir el consenso entre los nodos para validar los bloques, pero con la realización de cálculos más sencillos que no dependen del cálculo intensivo en busca de una recompensa, y teniendo en cuenta la optimización en el gasto de recursos para reducir el coste asociado, en base a que los nodos puedan demostrar que disponen de una participación predominante en la red al haber conseguido con anterioridad un porcentaje de *tokens* establecidos, sumado a otro factores, no existiendo recompensas por los bloques minados, sino que deben ajustarse a las tarifas por transacción conocidas como "*Transaction Fees*".

Este protocolo fue usado por primera vez con *Peercoin* en 2012 y quiere ser adoptado por *Ethereum* en la actualidad. Es, por tanto, la suposición de que los nodos que más han participado para conseguir *tokens* o tienen un mayor nivel de riqueza conocido como "*Stake*", también son lo más interesados en el buen funcionamiento de la red que los proporciona, haciéndolos de los más idóneos para realizar las tareas que supongan cierta responsabilidad en la seguridad del sistema. (BitMEX Research & Gubatron, 2018)

Su principal beneficio es que, al no requerir una ingente cantidad de energía para los procesos computacionales necesarios, se convierte en una alternativa más barata y con menor impacto ambiental.

2.4.7.3. Prueba de posesión (Proof of Possession, PoP)

Cuando se accede a los recursos protegidos de un sistema remoto mediante cifrado, en ocasiones es necesaria una protección de seguridad adicional para poder demostrar que la posesión de las claves criptográficas es legítima, proporcionando un mecanismo para vincular estas claves con *tokens* de seguridad, agregando firmas a las solicitudes *HTTP* hacia el destino de los recursos, quien utiliza estas claves para asegurarse de que el solicitante de la petición es la misma entidad que solicitó el *token* en primera instancia, y así poder descartar posibles robos de estos *tokens*. (IdentityServer3, 2018)

La Prueba de Posesión posibilita que una entidad pueda demostrar que estaba en posesión de algún dato en concreto en un momento en el tiempo, como son los datos almacenados en un servidor confiable, junto con una prueba de que estos datos no han sido manipulados, proporcionando garantías de seguridad probabilísticas. Este concepto dentro del marco de la cadena de bloques se puede extrapolar al supuesto en el que un servidor no autorizado propone alguna operación en *Ethereum*, no superando el control de acceso de grano fino que este mecanismo permite, y evitando de esta forma los costes de evaluación y rendimiento que supondría de no existir este control. (Knirsch, Unterweger, Karlsson, Engel, & Wicker)

2.4.7.4. Ethereum y Bifurcaciones

- *Ethereum*

Proyecto de código abierto iniciado en 2014 por *Vitalik Buterin, Gavin Wood y Jeffrey Wilcke,* y desarrollado por la *Ethereum Foundation.* Es un conjunto de protocolos que definen una plataforma descentralizada que opera bajo tecnología *Blockchain* personalizada con una infraestructura global compartida que permite la programación y ejecución de aplicaciones autoejecutables con tiempo de actividad garantizado del 100%, sin interferencias de terceros y sin posibilidad de algún tipo de fraude o censura. De cara a los desarrolladores dispone de una cartera (*Ethereum Wallet*) como puerta de entrada de las aplicaciones descentralizadas que se van a desarrollar, que permite mantener y gestionar los *tokens (Ether)* creados, aparte de herramientas de línea de comando para operar en terminal integradas en *Go, C++, Python, Java,* etc.

El núcleo de esta tecnología es la *Ethereum Virtual Machine (EVM),* ejecutada por cada nodo de la red y que implementa el código programado por los desarrolladores en lenguajes como *Python* o *JavaScript,* que se convertirán en las aplicaciones descentralizadas manteniendo el consenso de la propia *Blockchain* para así poder aplicarlo en cualquier entorno (no solo financiero) donde la confianza, la seguridad y la permanencia de los datos sean de especial relevancia. (Ethereum Foundation, 2018)

Esta plataforma tiene como objetivo cambiar los mecanismos o protocolos de consenso heredados de la *Blockchain* al objeto de que tengan un menor impacto en el consumo de energía, debido al intenso proceso computacional en sus cálculos gracias a tipos de consenso económicos.

- Bifurcaciones (*forks*)

En términos de desarrollo del software, una bifurcación es la creación de un subproyecto desde la rama principal que toma otra dirección distinta a la oficial, pero que tiene como base el código desarrollado hasta ese punto. En el desarrollo de *Blockchain* también existen bifurcaciones que son utilizadas del mismo modo y permiten modificar las reglas de consenso para actualizarlas o añadirles nuevas características, que dependiendo de su intensidad se subdividen en suaves y fuertes (*Soft Forks & Hard Forks*). Las suaves son más difíciles de programar porque reducen las reglas de consenso, pero no obligan a todos los nodos a actualizar su software, mientras que las fuertes son más sencillas de implementar, agregando o cambiando las reglas existentes, pero donde es necesario actualizar el software por parte de los usuarios o serán excluidos de esa bifurcación pudiendo derivar en un tipo de cadena independiente de la anterior. (Criptonoticias.com, 2017)

El ejemplo más conocido de *Hard-Fork* fue la división entre *Ethereum* y *Ethereum Classic*, esta última bifurcada de la *Blockchain* de la plataforma de contratos inteligentes de *Ethereum* debido a una disparidad de opiniones en los fundamentos básicos sobre la descentralización de las criptomonedas.

QuorumChain es una bifurcación de *Ethereum* como nuevo algoritmo de consenso que reemplaza *PoW* basado en un mecanismo de votación y donde las transacciones se pueden marcar como privadas dentro de la cadena pública de bloques, pero sólo como resúmenes hash de las cargas reales cifradas que se comparten con los nodos relevantes. Son estos nodos los que sólo podrán contener y descifrar los datos reales de la transacción, siendo ignorados por el resto, que lo sellan en el histórico de *Blockchain*.

CASPER es el nuevo protocolo de *PoS* que pretende que los validadores entreguen cantidades de tokens en depósito como medida de seguridad para poder participar en el sistema de consenso, de tal forma que, si no cumpliesen con las condiciones o reglas establecidas, dejaría de formar parte del consenso y se les retiraría este depósito. *Casper* determina la cantidad de *tokens* a entregar por los validadores en base al control sobre los depósitos de seguridad ya establecidos. Se conoce como consenso económico.

Beame.io, la plataforma de servicios que hemos utilizado para el desarrollo de la contribución del prototipo asegurado con una *D-PKI,* como veremos más adelante, proporciona una Prueba de Posesión (*PoP*) de una clave criptográfica única utilizada junto con la autenticación de múltiples factores, lo que permite a los desarrolladores de aplicaciones y dispositivos implementar fácilmente credenciales asimétricas para el cifrado de extremo a extremo en cualquier lugar. (GlobalSign, 2016)

2.5. Control y gestión de identidades

El estado actual del control y gestión de las identidades en los servicios que podemos acceder mediante las diferentes redes a nuestra disposición, especialmente *Internet* o los servicios *on-line,* están generalmente basadas en autenticación por contraseñas o de un solo factor (algo que se conoce), inseguro por definición, o de doble factor, o de múltiple factor (la combinación de dos aspectos o de los tres: de algo que se conoce, algo que se tiene o algo que se es), lo que presenta normalmente el inconveniente de depender de un tercero que provea de estos factores adicionales, por lo que se hace necesario una mejor gestión de las identidades digitales que mejoren los mecanismos de seguridad en cuanto a la autenticación y verificación.

2.5.1. Autenticación y verificación

Partiendo del concepto de que la autenticación es el procedimiento por el cual se permite asegurar que un usuario que quiere acceder a un servicio es realmente el que dice ser, es decir, se comprueba que es auténtico, se habrá completado el proceso de verificación de esa identidad que solicita el acceso a ese servicio. Esto normalmente implica que el

proveedor de los servicios a los que se quiere acceder almacene las credenciales del usuario para poder verificarlas, o lo haga un tercero de confianza delegado.

La autenticación de doble factor (*2FA*), o de múltiple factor (*MFA*), aunque aumenta la seguridad en el proceso, no está exenta de problemas, ya que necesita de canales de comunicación alternativos que pueden ser interceptados y comprometer los datos de verificación, al margen de la naturaleza propietaria de estos servicios, que dejan en manos de terceros los datos identificativos, así como los códigos necesarios para completar el proceso de autenticación, lejos del enfoque descentralizado y que no almacena credenciales.

Mediante el ecosistema *Blockchain,* este enfoque puede descentralizarse, y la gestión y control de estas credenciales quedaría implementada en la propia cadena de bloques con las posibilidades de aseguramiento de los datos asociados, aportando mayor seguridad a estos procesos gracias a los propios mecanismos criptográficos y de distribución que los hacen más confiables.

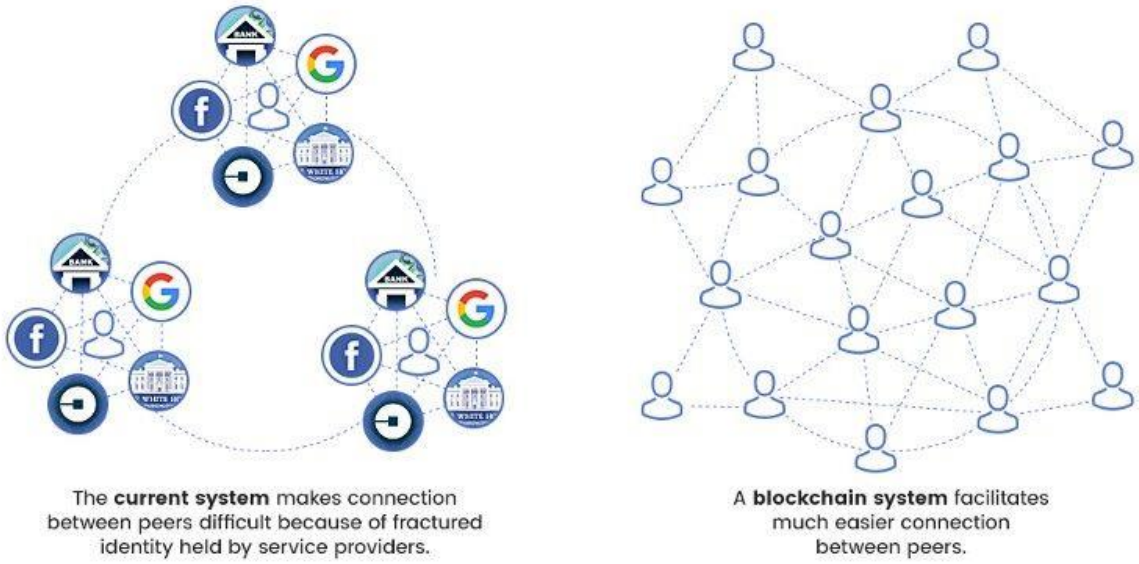

Ilustración 13. Autenticación y verificación Blockchain. (Ramiro, 2017)

Si consideramos estas identidades como parte de los bloques de datos de la cadena, estos podrían ser verificados por los nodos gracias a los algoritmos de firma digital, como *ECDSA*, que asocian una clave pública a la emisión de la identidad en la red, transfiriendo la propiedad de la clave privada al propietario de la identidad y permitiéndole firmarla, así como

la comprobación con la clave publica a disposición en la cadena de bloques, como parte de una autenticación descentralizada.

En términos de una aplicación podríamos vincularlo a un portal de inicio de sesión único, donde los servicios pueden recurrir para los procesos de autenticación y verificación, pero que no están enmarcados en una sola entidad individual, sino que se realiza de forma descentralizada, solicitando la identidad y la firma digital asociada de las peticiones de acceso para completar el proceso, validando la firma y verificando la identidad sin ser necesario almacenar ni intercambiar información sensible e innecesaria como credenciales o contraseñas únicas. (Ramiro, 2017)

2.5.1.1. Autorización basada en Blockchain

El proceso de autorización básicamente protege los recursos de un sistema permitiendo sólo el acceso a los usuarios que tienen permiso para ello.

En terminología de redes, un *Handshake* es un método utilizado para configurar una conexión *TCP/IP* a través de una red, donde hay tres mensajes transmitidos por *TCP* para negociar e iniciar una sesión entre dos computadoras negociando los parámetros de la conexión del *socket* de la red antes de transmitir datos, como hacen las solicitudes del navegador web *SSH* y *HTTP*. (InetDaemon, 2018)

En un entorno de *Blockchain* podemos encontrar flujos de autenticación genéricos basados en *Handshake,* pero orientado a bloques, que como si de un proceso de comunicación se tratase, verifican a la aplicación protegida que solicita la autenticación por un lado y al usuario que solicita el acceso a la aplicación por otro, sin intermediarios ni entidades centrales que contengan información privada de identidad.

A diferencia de un inicio de sesión común, aquí no es necesario introducir una contraseña, sino que la aplicación protegida mostrará un formulario para la identificación del usuario, que una vez completado mostrará, por ejemplo, un código *QR* para verificar la solicitud y enviar una respuesta completando el proceso de autenticación de forma codificada.

Es en este punto donde se debe garantizar que la autenticación sea correcta. Del lado de la aplicación se debe comprobar que la solicitud de acceso sea legítima, y del lado del usuario, que la aplicación protegida que está a la espera también sea la legítima.

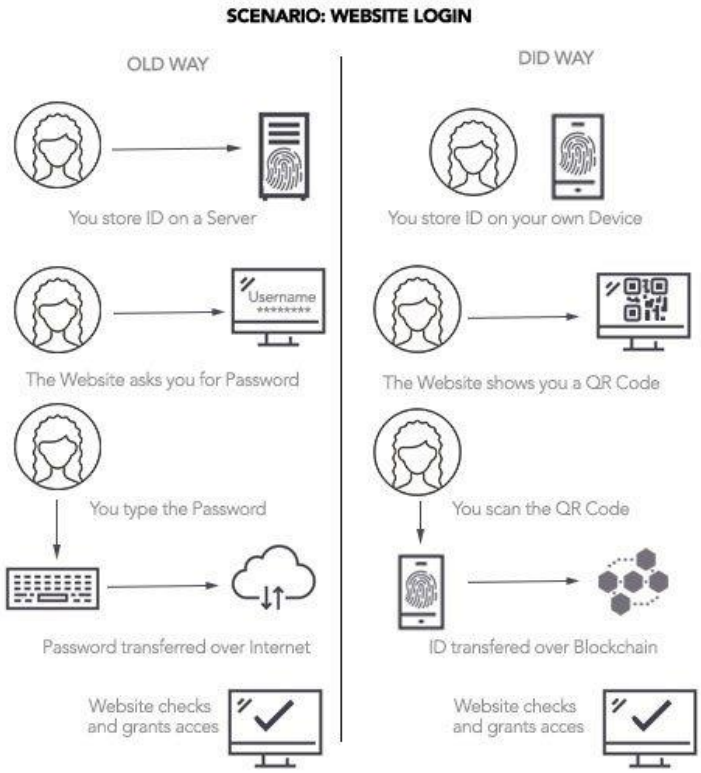

Ilustración 14. Website Login. (Decentralized ID Ltd., 2018)

Aquí es donde entra en juego la criptografía de clave pública basada en certificados digitales, donde la aplicación firma la solicitud, pudiéndose verificar públicamente en la *Blockchain,* que haría las veces de autoridad de certificación intermedia. Por otra parte, el usuario con su interacción en el formulario crearía una respuesta que también firmaría y enviaría de vuelta a la aplicación protegida. La solicitud es verificada de nuevo por la aplicación gracias a la criptografía de clave publica, y se registra al usuario en un proceso del todo descentralizado donde las dos partes han mantenido bajo su control las claves privadas y la *Blockchain* ha puesto a su disposición las claves publicas necesarias para completar el proceso.

2.5.1.2. Certificados X.509

El estándar internacional *X.509* en su versión 3, definido en 1994, es un marco de trabajo propuesto por *ISO (International Organization for Standardization)* e *ITU (International Telecommunication Union)* para proporcionar servicios de autenticación en directorios de

grandes redes de ordenadores mediante la definición del formato de certificados digitales de clave pública usados en protocolos de Internet, como *TLS/SSL*, base de las comunicaciones seguras bajo *HTTPS* de amplio uso en los navegadores web, y en otras aplicaciones, como la firma digital.

Contienen una clave pública, una entidad en forma de nombre único y una serie de campos con información añadida, estando firmados por una Autoridad de Certificación *CA* de confianza pública o auto-firmados por su propietario, de tal forma que quien posea estos certificados puede confiar en la clave pública que contiene para establecer comunicaciones seguras con otra parte, o validar documentos mediante la firma digital que proporciona la clave privada correspondiente facilitada por la *CA,* o bien que el mismo propietario ha generado.

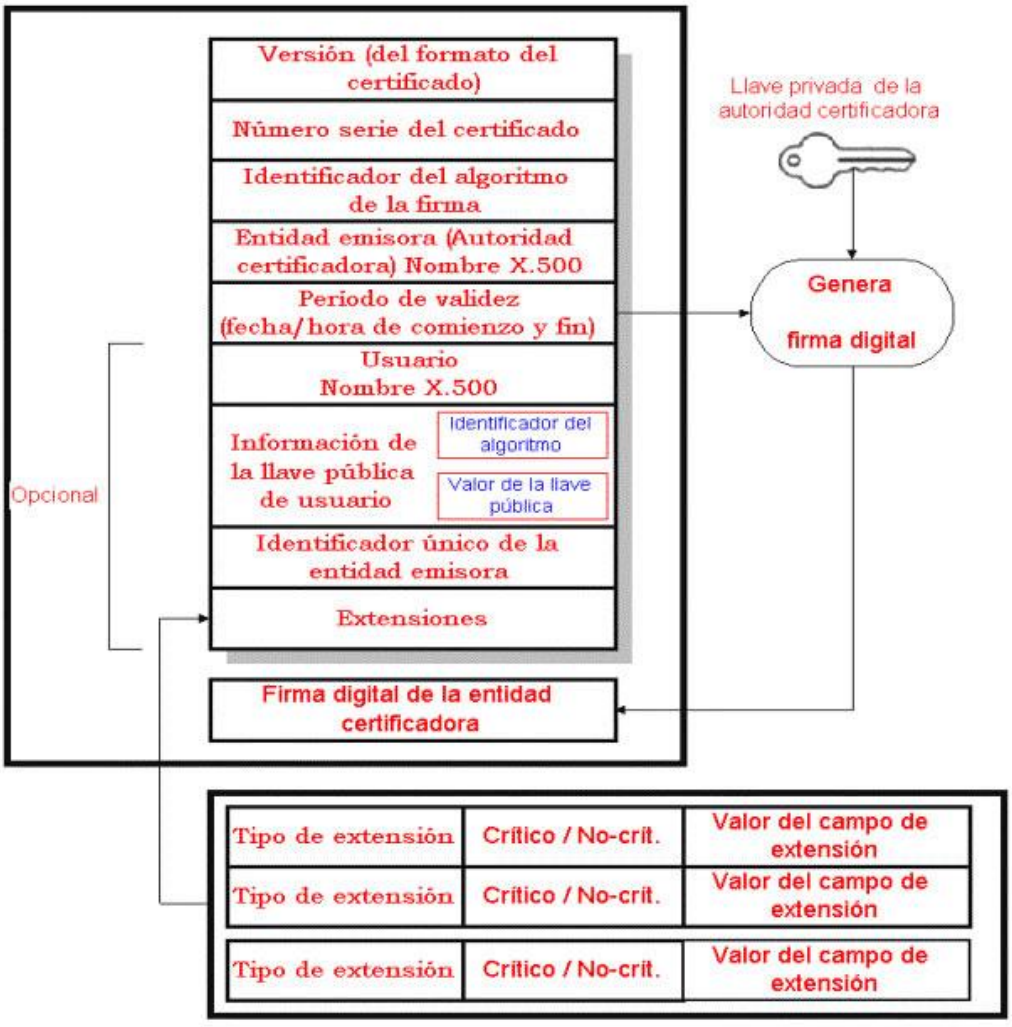

Ilustración 15. Estructura del Certificado X.509 v3. (Droguett, 2006)

2.5.1.3. *Ciclo de vida de los certificados*

Son las operaciones de gestión de la información contenida en el propio certificado, que básicamente son el proceso de emisión, distribución, verificación, expiración, renovación, reemisión, actualización y revocación de los certificados de una Autoridad Certificadora de confianza *CA*, así como la gestión de las claves.

X.509 también especifica listas de revocación de los certificados como medio para distribuir información sobre los certificados que ya no son válidos, y un algoritmo de validación de ruta de certificación, que permite que los certificados estén firmados por certificados *CA* intermedios, que están en a su vez firmados por otros certificados, llegando finalmente a una "Raíz de Confianza".

2.5.2. Infraestructura de Clave Pública (*PKI*)

También conocidas como *ICP*, o *Public Key Infraestructure*.

2.5.2.1. *Definición y funciones*

Conjunto de recursos *Hardware y Software*, procedimientos y políticas de seguridad que dan soporte a un servicio que hace uso de cifrado y firma digital con criptografía de clave pública para autenticar a las partes de una transacción e intercambiar información de forma segura, normalmente con el uso de certificados digitales bajo el estándar *X.509* que gestiona un tercero de confianza, es decir, con capacidad para aprovechar terceros confiables para vincular identidades a una clave criptográfica.

- Registro de las solicitudes de certificados mediante la verificación de la identidad del usuario.
- Generación de pares de claves (criptografía asimétrica).
- Garantizar la confidencialidad de las claves privadas.
- Certificar la vinculación entre cada clave pública y su correspondiente usuario.
- Revocación de los certificados según políticas de seguridad.

2.5.2.2. Usos y componentes

- Firma digital.
  - Documentos, software, etc., con la clave privada del certificado.
- Autenticación.
  - Verificando la identidad del usuario legítimo mediante *tokens* de seguridad.
- Integridad.
  - Garantizando que la información no se ha modificado sin autorización, mediante funciones resumen y firma digital.
- No repudio.
  - Al existir trazabilidad en los procesos se pueden consultar las transacciones.
- Gestión de claves.
  - Almacenamiento seguro y protección de claves privadas
- *CA* - Autoridad de Certificación.
  - Emiten y gestionan los certificados una vez que la *RA* ha verificado la información proporcionada.
- *RA* - Autoridad de Registro.
  - Verifican la información necesaria para que la *CA* emita el certificado y lo asocie a un usuario.
- *VA* - Autoridad de Validación.
  - Suministra información del estado de un certificado.
- Titular del certificado.
  - El propio usuario o la entidad correspondiente.
- Repositorio o directorio.
  - Almacenan y distribuyen los certificados y sus estados.
  - Confirman la validación de la cadena de certificados (*Certification Path*).

2.5.2.3. Servicios de las CA

- Emisión de certificados.
- Validación de certificados.
- Consulta online de estado de certificados (mediante la *VA*).
  - *OCSP (Online Certificate Status Protocol).*
- Revocación de certificados (mediante la *VA*).

- *CRL (Certificate Revocation List).*

- Gestión de claves.
- Marcado de tiempo *(TimeStamping).*
  - *TSA (Time Stamp Authority*).

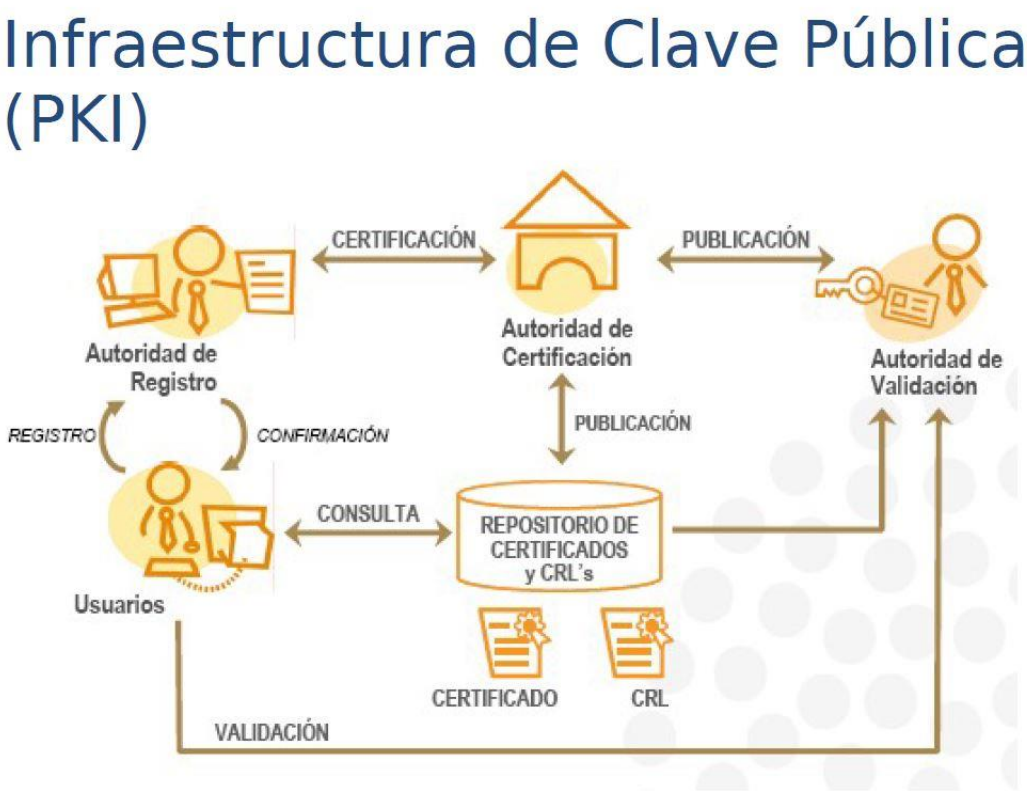

Ilustración 16. Infraestructura de Clave Pública (PKI). (Fernandez, 2015)

2.5.3. *PKI* distribuida (*D-PKI*)

Cuando hablábamos de *PKI* y certificados digitales, lo definíamos como la capacidad para aprovechar terceros de confianza para vincular identidades a una clave criptográfica; el concepto distribuido erradica ese tercero de confianza como posible compromiso a la integridad y seguridad del sistema, y lo sustituye por la propia confianza en la cadena de bloques y sus procesos criptográficos y de consenso de forma totalmente descentralizada.

2.5.3.1. Principios de seguridad D-PKI

- *D-PKI* es más segura, ya que cada identidad única tiene control sobre su identificador actual y el enlace de la clave pública, en contraposición a las *PKI* tradicionales, donde esta gestión recae sobre la *CA*, que en términos de seguridad implicaría comprometer a muchas identidades una por una, o comprometer una sola *CA*.

- El sistema de identidades en una *D-PKI* debe actualizarse por completo en el momento que cualquier nodo haga modificaciones en alguna identidad, y esto es así precisamente para proteger al sistema de posibles ataques a nivel de red, sólo dejando la posibilidad de atacar a todos los nodos a la vez.

Estos principios se consiguen mediante el uso de almacenes de datos clave-valor descentralizados, donde se alojan los enlaces entre los identificadores y los *hashes* de las claves públicas. (Allen et al., 2015)

2.5.4. Identidades Digitales Descentralizadas (*DID*)

Es una nueva definición de identidad digital autónoma donde los usuarios son los que tienen el control sobre la gestión de sus propias identidades creadas en entornos descentralizados como *Blockchain*, y donde la verificación de la identidad ya no recae en una entidad central, sino en la propia cadena de bloques.

2.5.4.1. Entidad soberana / auto-soberana

Es el sobrenombre con el que se conoce a estas nuevas *DID* generadas a partir de un sistema digital cifrado, donde además se pueden almacenar los datos relativos a esa identidad de forma segura, fácilmente controlables por el propio usuario, de tal forma que la propiedad de los datos de identificación recaen sobre ellos mismos, siendo al mismo tiempo y de forma autosuficiente la autoridad emisora, el titular y el gestor de la clave privada, con componentes añadidos que prueban la propiedad de la identidad, exponiendo sólo los datos relevantes que sean necesarios, quedando el resto protegidos de manera oculta o privada y

dependiendo sólo de la parte verificadora, algo que tiene que ver con los contratos inteligentes de *Blockchain* que permiten al usuario probar su identidad y demostrar su propiedad.

Grandes empresas y organizaciones como *IBM, Microsoft, DIF (Decentralized Identity Foundation)* o *Ernst & Young*, están uniendo esfuerzos para crear sistemas de identidades auto-soberanas basadas en tecnología *Blockchain* que hagan especial énfasis en que los datos personales sean controlados por los individuos a los que correspondan dichos datos, remarcando el carácter de identidad con soberanía propia. Cabe destacar la aplicación "*Stratis*", que con la ayuda de *Microsoft Azure Active Directory* como capa adicional de seguridad permite el manejo y registro de los datos de los usuarios en una plataforma *Blockchain* distribuida haciendo uso de los datos de identificación que ya posean en Microsoft, LinkedIn o Google. (Faria, 2017)

3. Objetivos y metodología de trabajo

3.1. Objetivo general

El principal objetivo de este trabajo es demostrar cómo se puede asegurar un despliegue de dispositivos *IoT* expuestos a Internet y que de manera general no implementan una protección suficiente del acceso remoto en sus diseños o configuraciones por defecto mejorando su seguridad, reduciendo la superficie de ataque y las vulnerabilidades detectadas.

Tras el estudio teórico de las tecnologías a utilizar y en base a la metodología de auditoria de la seguridad realizada, se plantea un experimento para la gestión segura del acceso a dispositivos *IoT* conectados y la gestión de sus servicios y usuarios convirtiéndolos en servidores web públicos confiables que permitan su acceso remoto restringido, su monitoreo y control de forma confiable, usando el estándar de cifrado *SSL/TLS* incluido en los navegadores web y la autenticación criptográfica de la identidad digital proporcionada por la implementación de una *PKI* descentralizada (*D-PKI*) bajo tecnología *Blockchain* desplegada por la empresa *beame.io* en la nube de *Amazon Web Services (I. Amazon Web Services, 2018)*, unidos a otros mecanismos de autenticación y autorización de usuarios como *MFA,* mediante aplicación móvil y estándares de seguridad como *SAML*.

3.2. Objetivos específicos

- Conocer el marco tecnológico de la tecnología *Blockchain*.
- Entender su aplicabilidad en la autenticación y el control de accesos.
- Describir como una *PKI* controla y gestiona las identidades digitales.
- Especificar los objetivos del proyecto y una metodología basada en riesgos *EDR*.
- Medir el riesgo de los dispositivos *IoT* en base a su configuración por defecto.
- Analizar y proponer mejoras en base al riesgo detectado.
- Establecer un entono de *PKI* bajo *Blockchain (D-PKI)* para dispositivos *IoT*.
- Verificar y validar los resultados obtenidos.
- Cuantificar de nuevo el riesgo mediante métricas y mejoras obtenidas.
- Reconocer posibles nuevas amenazas al sistema.

3.3. Metodología de trabajo

En la realización del presente Trabajo Fin de Máster se ha decidido hacer uso de una metodología de auditoría informática basada en riesgo como proceso del análisis de la seguridad de los dispositivos *IoT* conectados, y en concreto una evaluación del riesgo *EDR* genérico *(Risk Oriented Assessment ROA),* recomendada por *ISACA* y creada por la consultora *Arthur Andersen*, (l. e. l. Wikipedia, 2018) donde serán los propios dispositivos los que identificaremos como los activos a evaluar en función de las tres dimensiones de la seguridad de la información: Confidencialidad, Integridad y Disponibilidad. (Tejero, 2017)

El riesgo se entiende como la probabilidad de amenazas sobre estos activos que puedan impactar a nivel técnico de forma negativa en su seguridad y la de los datos que manejan y que tras su análisis, identificación y cuantificación se procederá a la implantación de los controles pertinentes para poder reducirlos o mitigarlos, asegurando de esta manera mediante las tecnologías propuestas (*Blockchain* + *PKI*) el sistema desplegado.

Ilustración 17. Proceso básico de análisis y gestión de Riesgos. (Lean Manufacturing et al., 2018)

3.3.1. Identificación de riesgos

En este apartado se van a identificar los posibles riesgos potenciales en ausencia de controles relacionados al objeto de poder cuantificarlos, y como punto de partida vamos a utilizar la clasificación de vulnerabilidades del proyecto *OWASP top 10* en su variante para dispositivos IoT de 2014. (owasp.org, 2015)

Tabla 1.Identificación de riesgos potenciales

OWASP IoT Top 10 - 2014	Agentes de amenaza Aplicación específica	Vectores de ataque
R01: Interfaz Web insegura	Considere a cualquiera que tenga acceso a la interfaz web, incluidos los usuarios internos y externos.	El atacante usa credenciales débiles, captura de credenciales en texto sin formato o enumera cuentas para acceder a la interfaz web.
R02: Autenticación / Autorización insuficientes	Considere a cualquiera que tenga acceso a la interfaz web, la interfaz móvil o la interfaz en la nube, incluidos los usuarios internos y externos.	El atacante usa contraseñas débiles, mecanismos de recuperación de contraseña inseguros, credenciales mal protegidas o falta de control de acceso granular para acceder a una interfaz en particular.
R03: Servicios de red inseguros	Considere a cualquiera que tenga acceso al dispositivo a través de una conexión de red, incluidos los usuarios externos e internos.	El atacante usa servicios de red vulnerables para atacar el dispositivo en sí o redirigir ataques hacia el dispositivo.
R04: Falta de Cifrado de transporte	Considere a cualquiera que tenga acceso a la red a la que está conectado el dispositivo, incluidos los usuarios externos e internos.	El atacante usa la falta de cifrado de transporte para ver los datos que se pasan a través de la red.
R05: Privacidad	Tenga en cuenta a cualquiera que tenga acceso al dispositivo en sí, la red a la que está conectado el dispositivo, la aplicación móvil y la conexión a la nube, incluidos los usuarios externos e internos.	El atacante utiliza múltiples vectores, como autenticación insuficiente, falta de cifrado de transporte o servicios de red inseguros para ver datos personales sin proteger adecuadamente o que se recopilan innecesariamente.
R06: Interfaz Cloud insegura	Considere a cualquiera que tenga acceso a Internet.	El atacante usa varios vectores, como la autenticación insuficiente, la falta de cifrado de transporte y la enumeración de cuentas para acceder a los datos o controles a través del sitio web de la nube.
R07: Interfaz Móvil insegura	Considere a cualquiera que tenga acceso a la aplicación móvil.	El atacante usa varios vectores, como la autenticación insuficiente, la falta de cifrado de transporte y la enumeración de cuentas para acceder a datos o controles a través de la interfaz móvil.
R08: Configuración de Seguridad insuficiente	Considere a cualquiera que tenga acceso al dispositivo.	El atacante usa la falta de permisos granulares para acceder a datos o controles en el dispositivo. El atacante también podría aprovechar la falta de opciones de contraseña y cifrado para realizar otros ataques que pueden comprometer el dispositivo y / o los datos.
R09: Software / Firmware inseguro	Considere a cualquiera que tenga acceso al dispositivo y / o a la red en la que reside el dispositivo. Considere también a cualquiera que pueda obtener acceso al servidor de actualización.	El atacante usa varios vectores, como capturar archivos de actualización a través de conexiones no cifradas, el archivo de actualización en sí no está cifrado o se puede realizar la actualización maliciosa mediante secuestro de DNS. Según el método de actualización y de la configuración.
R10: Pobre seguridad física	Considere a cualquiera que tenga acceso físico al dispositivo.	El atacante usa vectores como puertos USB, tarjetas SD u otros medios de almacenamiento para acceder al sistema operativo y, potencialmente, a cualquier información almacenada en el dispositivo.

Tabla 2. Nivel de impacto técnico

OWASP IoT Top 10 - 2014	Impacto técnico	Nivel
R01: Interfaz Web insegura	Puede provocar la pérdida o corrupción de datos, la falta de responsabilidad proactiva o la denegación de acceso y pueden llevar a la adquisición completa del dispositivo.	**ALTO**
R02: Autenticación / Autorización insuficientes	Puede ocasionar la pérdida o corrupción de datos, la falta de responsabilidad proactiva o la denegación de acceso y puede llevar a un compromiso completo del dispositivo y / o las cuentas de usuario.	**ALTO**
R03: Servicios de red inseguros	Pueden provocar la pérdida o corrupción de datos, denegación de servicio o la facilitación de ataques a otros dispositivos.	**MEDIO**
R04: Falta de Cifrado de transporte	Puede ocasionar la pérdida de datos y, dependiendo de los datos expuestos, puede llevar a un completo compromiso del dispositivo o de las cuentas de usuario.	**ALTO**
R05: Privacidad	La recopilación de datos personales de usuarios junto con la falta de protección de estos datos puede llevar a su compromiso.	**ALTO**
R06: Interfaz Cloud insegura	Podría comprometer los datos del usuario y controlar el dispositivo.	**ALTO**
R07: Interfaz Móvil insegura	Podría comprometer los datos del usuario y controlar el dispositivo.	**ALTO**
R08: Configuración de Seguridad insuficiente	Podría llevar a un compromiso del dispositivo ya sea intencional o accidental y / o pérdida de datos.	**MEDIO**
R09: Software / Firmware inseguro	Podría llevar a un compromiso de los datos del usuario, control sobre el dispositivo y ataques contra otros dispositivos.	**ALTO**
R10: Pobre seguridad física	Podría comprometer el dispositivo en sí y cualquier dato almacenado en ese dispositivo.	**ALTO**

Los niveles de riesgo están calculados en base a la clasificación *OWASP Top 10 IoT Vulnerabilities* en función de su impacto técnico, donde ALTO corresponde a *SEVERE*, MEDIO corresponde a *MODERATE* y BAJO corresponde a *MINOR*.

Por niveles de detectabilidad entendemos el riesgo de que un posible atacante detecte esta vulnerabilidad en los dispositivos; del mismo modo el nivel de explotabilidad es el riesgo de posible explotación de esa vulnerabilidad una vez detectada.

Tabla 3. Nivel de detectabilidad / explotabilidad

OWASP IoT Top 10 - 2014	Nivel de detectabilidad	Nivel de explotabilidad
R01: Interfaz Web insegura	**ALTO**	**ALTO**
R02: Autenticación / Autorización insuficientes	**ALTO**	**MEDIO**
R03: Servicios de red inseguros	**MEDIO**	**MEDIO**
R04: Falta de Cifrado de transporte	**ALTO**	**MEDIO**
R05: Privacidad	**ALTO**	**MEDIO**
R06: Interfaz Cloud insegura	**ALTO**	**MEDIO**
R07: Interfaz Móvil insegura	**ALTO**	**MEDIO**
R08: Configuración de Seguridad insuficiente	**ALTO**	**MEDIO**
R09: Software / Firmware inseguro	**ALTO**	**BAJO**
R10: Pobre seguridad física	**MEDIO**	**MEDIO**

Los niveles de riesgo están calculados en base a la clasificación *OWASP Top 10 IoT Vulnerabilities* en función de su nivel de detectabilidad, donde ALTO corresponde a *EASY*, MEDIO corresponde a *AVERAGE* y BAJO corresponde a *DIFICULT,* y en función de su explotabilidad, donde ALTO corresponde a *EASY*, MEDIO corresponde a *AVERAGE* y BAJO corresponde a *DIFICULT*.

Para el cálculo del riesgo potencial en función de cada riesgo clasificado se le ha asociado un valor numérico a cada valor de riesgo: ALTO = 3, MEDIO = 2, BAJO = 1, que sumándolos y dividiéndolos entre 3 nos dan el valor final, que una vez redondeado (en escala de 0,5) podemos asociarlo a uno de los tres niveles definidos.

Tabla 4. Nivel de riesgo potencial

OWASP IoT Top 10 - 2014	Nivel de impacto	Nivel de detectabilidad	Nivel de explotabilidad	Nivel de riesgo
R01: Interfaz Web insegura	ALTO	ALTO	ALTO	ALTO
R02: Autenticación / Autorización insuficientes	ALTO	ALTO	MEDIO	ALTO
R03: Servicios de red inseguros	MEDIO	MEDIO	MEDIO	MEDIO
R04: Falta de Cifrado de transporte	ALTO	ALTO	MEDIO	ALTO
R05: Privacidad	ALTO	ALTO	MEDIO	ALTO
R06: Interfaz Cloud insegura	ALTO	ALTO	MEDIO	ALTO
R07: Interfaz Móvil insegura	ALTO	ALTO	MEDIO	ALTO
R08: Configuración de Seguridad insuficiente	MEDIO	ALTO	MEDIO	MEDIO
R09: Software / Firmware inseguro	ALTO	ALTO	BAJO	MEDIO
R10: Pobre seguridad física	ALTO	MEDIO	MEDIO	MEDIO

ALTO = 3; MEDIO = 2; BAJO = 1 TOTAL nivel de riesgo inicial = 26

Los objetivos de control y los controles están basados en las recomendaciones de *OWASP top 10 IoT 2014*; las pruebas de cumplimiento y pruebas sustantivas son propias y están en su mayoría dirigidas a la implementación de la D-PKI y el aseguramiento de las conexiones de los dispositivos.

3.3.2. Objetivos de control

En esta sección vamos a tratar de reducir los riesgos potenciales cuantificados anteriormente, estableciendo unos objetivos de control que nos permitan identificar y definir la forma de reducir el riesgo.

Nos vamos a centrar en el nivel de explotabilidad al objeto de intentar reducirlo al máximo con la implantación de controles sobre las debilidades de seguridad encontradas y en base a las tecnologías que vamos a implementar para remediarlo.

Tabla 5. Objetivos de control

OWASP IoT Top 10 - 2014	Debilidad de seguridad	Objetivo de control
R01: Interfaz Web insegura	Problemas como enumeración de cuentas, falta de bloqueo de cuentas o credenciales débiles.	Asegurar adecuadamente la gestión de cuentas y credenciales, así como su exposición.
R02: Autenticación / Autorización insuficientes	Uso de contraseñas débiles o poco protegidas. Las deficiencias se encuentran presentes tanto en interfaces internas como externas.	Prevenir accesos no autorizados al sistema e información, evitando la modificación o deterioro de estos.
R03: Servicios de red inseguros	Susceptibles a ataques de desbordamiento de búfer o ataques de denegación de servicio propios o a terceros desde el dispositivo.	Garantizar la seguridad de los servicios de red evitando la falta de disponibilidad de este o accesos y usos indebidos.
R04: Falta de Cifrado de transporte	Los datos son visibles a medida que viajan a través de las redes locales o Internet.	Implementar las capas necesarias de seguridad añadidas para que el tráfico de red no sea inseguro.
R05: Privacidad	Recopilación y falta de protección adecuada de datos personales revisando los datos que se recopilan a medida que el usuario configura y activa el dispositivo.	Mantener fuera del alcance de un posible atacante los datos personales o evitar su uso con otro tipo de información de identificación.
R06: Interfaz Cloud insegura	Uso de credenciales fáciles de adivinar o es posible la enumeración de cuentas.	Asegurar que las credenciales no estén expuestas en Internet.
R07: Interfaz Móvil insegura	Uso de credenciales fáciles de adivinar o la enumeración de la cuenta es posible.	Asegurar que las credenciales no estén expuestas mientras están conectadas a redes inalámbricas
R08: Configuración de Seguridad insuficiente	Cuando los usuarios del dispositivo tienen capacidad limitada o no para alterar sus controles de seguridad o la interfaz web del dispositivo no tiene opciones para crear permisos granulares de usuario o, por ejemplo, forzar el uso de contraseñas seguras.	Establecer políticas de seguridad apropiadas para que el dispositivo garantice una óptima gestión de la seguridad y del control de acceso al mismo.
R09: Software / Firmware inseguro	La falta de capacidad para actualizar un dispositivo es una debilidad de seguridad por sí misma y cuando los propios archivos actualizados y la conexión de red en la que se entregan no estén protegidos.	Los dispositivos deben tener la capacidad de actualizarse cuando se descubran vulnerabilidades y las actualizaciones de software / firmware deben ser seguras.
R10: Pobre seguridad física	Cuando un atacante puede desmontar un dispositivo para acceder fácilmente al medio de almacenamiento y a cualquier información almacenada en ese medio o cuando los puertos USB u otros puertos externos se pueden utilizar para acceder al dispositivo utilizando funciones destinadas a la configuración o mantenimiento.	Mantener los dispositivos físicamente inaccesibles al personal no autorizado garantizando su seguridad, la de sus componentes y de la información que contienen.

3.3.3. Identificación de controles

Una vez cuantificados los riesgos y definidos los objetivos de control, vamos a identificar las acciones de control necesarias para minimizar el riesgo inicial, que pueden ser uno o varios controles por objetivo, siendo estos de carácter preventivo o correctivo.

Tabla 6. Identificación de controles

OWASP IoT Top 10 - 2014	Objetivo de control	Controles
R01: Interfaz Web insegura	Asegurar adecuadamente la gestión de cuentas y credenciales, así como su exposición.	Asegurar que las contraseñas predeterminadas y nombres de usuario predeterminados se cambiarán durante la configuración inicial.
R02: Autenticación / Autorización insuficientes	Prevenir accesos no autorizados al sistema e información, evitando la modificación o deterioro de estos.	Asegurar un control de acceso granular y autenticación de múltiple factor.
R03: Servicios de red inseguros	Garantizar la seguridad de los servicios de red evitando la falta de disponibilidad de estos o el acceso y uso indebido.	Asegurar que solo los puertos necesarios estén expuestos y disponibles.
R04: Falta de Cifrado de transporte	Implementar las capas necesarias de seguridad añadidas para que el tráfico de red no sea inseguro.	Asegurar que los datos se cifren usando protocolos como SSL y TLS mientras transitan las redes.
R05: Privacidad	Mantener fuera del alcance de un posible atacante los datos personales o evitar su uso con otro tipo de información de identificación.	Asegurar que cualquier información personal o de identificación recolectada sea anonimizada o no fácil de identificar.
R06: Interfaz Cloud insegura	Asegurar que las credenciales no estén expuestas en Internet.	Asegurar la gestión de cuentas y credenciales con estándares de intercambio de datos de autenticación y proveedores de identidad delegados.
R07: Interfaz Móvil insegura	Asegurar que las credenciales no estén expuestas mientras están conectadas a redes inalámbricas.	Asegurar la conexión inalámbrica del dispositivo con protocolos seguros de conexión (WPA2).
R08: Configuración de Seguridad insuficiente	Establecer políticas de seguridad apropiadas para que el dispositivo garantice una óptima gestión de la seguridad y del control de acceso al mismo.	Garantizar la capacidad de separar usuarios normales de usuarios administradores.
R09: Software / Firmware inseguro	Los dispositivos deben tener la capacidad de actualizarse cuando se descubran vulnerabilidades y las actualizaciones de software / firmware deben ser seguras.	Garantizar que el firmware del dispositivo es actualizable.
R10: Pobre seguridad física	Mantener los dispositivos físicamente inaccesibles al personal no autorizado garantizando su seguridad, la de sus componentes y de la información que contienen.	Ubicar los dispositivos en lugares de no libre acceso y donde quede registro del personal que accede a los mismos.

3.3.4. Pruebas de cumplimiento

Se definen las pruebas que se deben realizar para probar y verificar el cumplimiento de las anteriores técnicas de control propuestas en base a los riesgos identificados y al objeto de mitigarlos.

Tabla 7. Pruebas de cumplimiento

OWASP IoT Top 10 - 2014	Controles	Pruebas de cumplimiento
R01: Interfaz Web insegura	Asegurar que las contraseñas predeterminadas y nombres de usuario predeterminados se cambiarán durante la configuración inicial.	Comprobar que no se accede a los dispositivos con los usuarios y contraseñas por defecto.
R02: Autenticación / Autorización insuficientes	Asegurar un control de acceso granular y autenticación de múltiple factor.	Comprobar que el acceso al dispositivo está protegido con usuarios registrados mediante IDP y doble factor de autenticación que inician sesión con SP (Blockchain).
R03: Servicios de red inseguros	Asegurar que solo los puertos necesarios estén expuestos y disponibles.	Comprobar con un escáner de red los puertos expuestos de los dispositivos.
R04: Falta de Cifrado de transporte	Asegurar que los datos se cifren usando protocolos como SSL y TLS mientras transitan las redes.	Comprobar la emisión de certificados válidos para las conexiones cifradas en los dispositivos y el traspaso seguro de las claves privadas simétricas.
R05: Privacidad	Asegurar que cualquier información personal o de identificación recolectada sea anonimizada o difícil de identificar.	Comprobar que los datos de identificación para acceder a los sistemas no son datos de carácter personal.
R06: Interfaz Cloud insegura	Asegurar la gestión de cuentas y credenciales con estándares de intercambio de datos de autenticación y proveedores de identidad delegados.	Comprobar que la comunicación de los dispositivos con los proveedores de servicios Cloud se hacen de forma segura.
R07: Interfaz Móvil insegura	Asegurar la conexión inalámbrica del dispositivo con protocolos seguros de conexión (WPA2).	Comprobar que las comunicaciones de los dispositivos móviles con los enrutadores WiFi se realizan bajo conexiones WPA2.
R08: Configuración de Seguridad insuficiente	Garantizar la capacidad de separar usuarios estándar de usuarios administradores.	Comprobar que los usuarios no privilegiados no pueden realizar acciones administradoras.
R09: Software / Firmware inseguro	Garantizar que el firmware del dispositivo es actualizable.	Comprobar que los dispositivos no están descontinuados y tienen soporte de actualización.
R10: Pobre seguridad física	Ubicar los dispositivos en lugares de no libre acceso y donde quede registro del personal que accede a los mismos.	Comprobar que la ubicación física del dispositivo es dentro de un entorno seguro.

3.3.5. Pruebas sustantivas

Pruebas adicionales opcionales para ampliar muestras y verificar que las pruebas anteriores han sido cumplidas y comprobar que los controles realizados sobre los riesgos detectados son efectivos o, por el contrario, detectar su falta de efectividad.

Tabla 8. Pruebas sustantivas

OWASP IoT Top 10 - 2014	Pruebas de cumplimiento	Pruebas sustantivas
R01: Interfaz Web insegura	Comprobar que no se accede a los dispositivos con los usuarios y contraseñas por defecto.	n/a
R02: Autenticación / Autorización insuficientes	Comprobar que el acceso al dispositivo está protegido con usuarios registrados mediante IDP y múltiple factor de autenticación que inician sesión con el SP (Blockchain).	Comprobar uso de Blockchain para la emisión y verificación de identidades criptográficas en base a la lógica del árbol de confianza de la cadena que establece los roles y emite los tokens de autorización y códigos QR para el emparejamiento usuario / dispositivo.
R03: Servicios de red inseguros	Comprobar con un escáner de red los puertos expuestos de los dispositivos.	n/a
R04: Falta de Cifrado de transporte	Comprobar la emisión de certificados válidos para las conexiones cifradas en los dispositivos y el traspaso seguro de las claves privadas simétricas.	Comprobar los certificados clientes X.509 con claves RSA de 2048 bits creados en el dispositivo y los tokens de autorización firmados que se envían a la PKI junto con la clave publica firmada.
R05: Privacidad	Comprobar que los datos de identificación para acceder a los sistemas no son datos de carácter personal.	Comprobar identidad con FQDN bajo subdominio beameio.net
R06: Interfaz Cloud insegura	Comprobar que la comunicación de los dispositivos con los proveedores de servicios Cloud se hacen de forma segura.	Capturar el tráfico de red y comprobar que las credenciales no son transmitidas como texto en claro.
R07: Interfaz Móvil insegura	Comprobar que las comunicaciones de los dispositivos móviles con los enrutadores WiFi se realizan bajo conexiones WPA2.	n/a
R08: Configuración de Seguridad insuficiente	Comprobar que los usuarios no privilegiados no pueden realizar acciones administradoras.	Ejecutar script con usuario no privilegiado
R09: Software / Firmware inseguro	Comprobar que los dispositivos no están descontinuados y tienen soporte de actualización.	n/a
R10: Pobre seguridad física	Comprobar que la ubicación física del dispositivo está dentro de un entorno seguro.	n/a

4. Desarrollo específico de la contribución

4.1. Descripción del piloto

Compuesto principalmente por un proveedor de identidades inicial (*IDP*) de la *D-PKI* y dos dispositivos *IoT* sensorizados y conectados, accesibles y gestionables mediante Internet; uno configurado de forma deliberadamente vulnerable (*IoT01*) y otro asegurado con la implementación de una *PKI* funcional bajo el despliegue *Blockchain* del proveedor *beame.io* (beame.io, 2017) (*IoT02*), para poder evaluar mediante Auditoría de Sistemas de Información en base a la Metodología de Gestión de Riesgos (*EDR*) los resultados obtenidos y poder compararlos, permitiéndonos medir el grado de mejora en función de la seguridad implementada.

4.1.1. Prototipo *D-PKI* y Proveedor de Identidades

Integración de un sistema *PKI* de credenciales descentralizado bajo *Blockchain* para el posterior registro y aprovisionamiento de los dispositivos *IoT* conectados, permitiendo mediante las aplicaciones publicadas en ellos, su gestión y control, haciendo uso de los servicios que presta *beame.io* de forma segura.

Permite desplegar un entorno donde los dispositivos sean accesibles desde Internet sin dirección IP pública, mediante sus nombres de dominio únicos generados de forma aleatoria y bajo el subdominio beameio.net en el momento de su registro y bajo túnel *HTTPS* cifrado con certificados *SSL/TLS* para el acceso remoto autenticado mediante identidad criptográfica implementada en una *PKI* basada en *Blockchain*, gracias a una aplicación móvil que permita al dispositivo reconocer los certificados del cliente o su posesión y un navegador web que implemente el protocolo de cifrado.

El intercambio de la información de las identidades federadas se realiza mediante un *Identity Provider IDP* (máquina virtual en portátil con la aplicación *beame-gatekeeper*) y un *Security Provider SP* (*D-PKI* en la *Blockchain Ethereun* de *beame.io*), bajo el estándar de seguridad *SAML* y mediante el uso de un navegador web que soporte el protocolo *HTTPS* y un dispositivo móvil como parte necesaria para la autenticación de múltiple factor utilizada en el despliegue.

4.1.1.1. Descripción funcional y tecnologías utilizadas

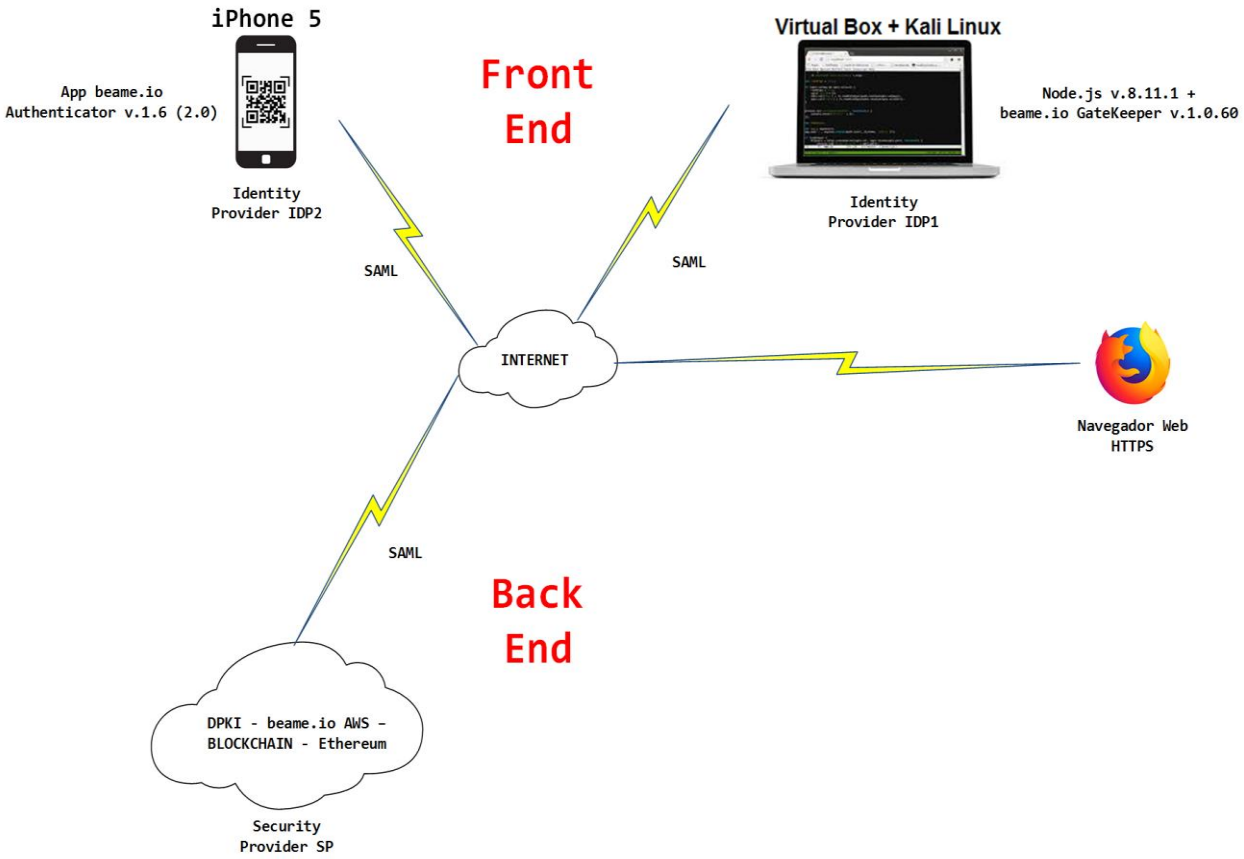

Ilustración 18. Esquema de funcionamiento Prototipo D-PKI

4.1.1.2. Herramientas de implementación

- En la máquina *Kali Linux*:
  - *Kali 4.15.0 x64 GNU/Linux Rolling*
  - *Node.js v8.11.1.*
  - SERVIDOR *beame-gatekeeper v. 1.0.60* + *token* de registro.
  - Emisión y validación de credenciales de identidad para la autenticación.
  - Almacenamiento y administración de certificados.
  - Entidad que mantiene el libro mayor y hace copias de seguridad.
  - Proporciona respuestas a la autenticación mediante *SAML.*

- En el dispositivo móvil:

    - *iOS 10.3.3.*
    - APP *Beame Authenticator v1.6 (2.0).*
    - Pareado con el servicio de *beame.io.*
    - Capaz de alojar las credenciales que gestionan el árbol del *Blockchain* como identidad principal, siguiendo el modelo jerárquico construido con certificados *X.509.*

- En la nube de *beame.net:*

    - Granja de servidores *Proxies* en *Amazon Web Services.*
    - *Blockchain* Pública *Ethereun.*
    - *D-PKI.*
    - *Security Provider SP.*

4.1.1.3. Registro en el servicio e instalación en Linux

Una vez registrados en el servicio *beame-gatekeeper* de *beame.io* (beameio.net, 2018) recibimos por correo electrónico, con un *token* codificado en Base 64 para configurar en la máquina *Linux* virtual, el servidor *beame-gatekeeper,* como un servicio que se ejecuta en el sistema al iniciar la máquina y poder disponer de nuestro *IDP* principal en funcionamiento, que nos servirá para establecer la lógica del árbol de confianza, creando el certificado raíz de confianza de la cadena *Level 0* “L0” de nivel superior.

```
sudo beame-gatekeeper-install 'B64 TOKEN email’
```

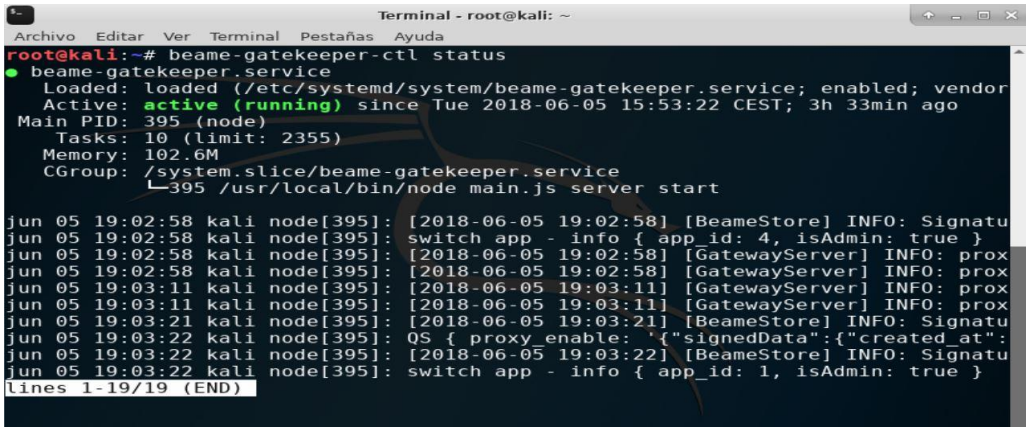

Ilustración 19. Servidor beame-gatekeeper iniciado

```
beame-gatekeeper-ctl status
```

Para acceder a la interfaz de administración del servidor introducimos el siguiente comando:

```
beame-gatekeeper-ctl admin
```

Permite, “entre otras operaciones”, el registro y actualización de los datos del usuario y los servicios y base de datos de identidades para su sincronización con la *Blockchain Ethereum* de *beame.io* desplegada en *Amazon Web Services AWS.* Nos presenta una *URL* temporal para poder acceder a la administración del gatekeeper desde el propio navegador de la máquina *Kali Linux,* donde se puede observar el árbol de confianza de credenciales estructurado en capas con el nivel superior (padre) ya creado (*Highest FQDN*) de la cadena de certificados, y donde el nombre del servicio (mi nombre) representa el hash del *FQDN* del certificado raíz, así como los de los servicios locales (L) asociados del *GatewayServer*, *LoginManager, MatchingServer* y el usuario remoto (R) fj.balmaseda.

Cabe destacar que esta administración nos permite tener control sobre los certificados emitidos, como es su revocación o exportación y otras operaciones sobre la identidad generada, como la consulta de los *FQDN, DNS*, roles, etc.

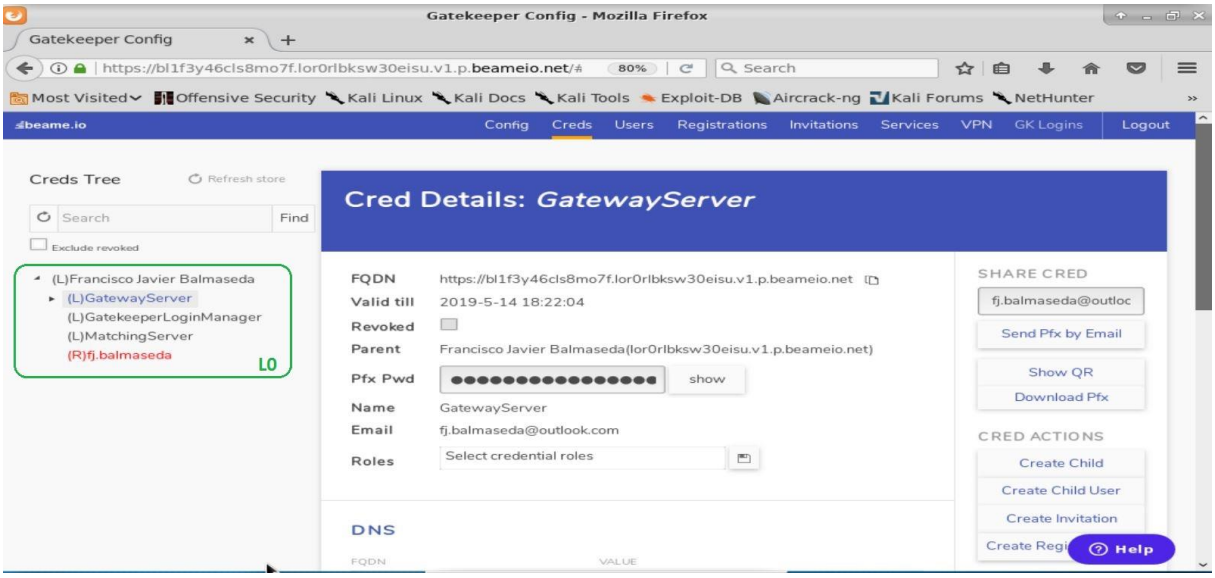

Ilustración 20. Administración del gatekeeper y árbol de confianza con L0

Del mismo modo, esta administración tiene la capacidad de emitir *tokens* adicionales para la creación de más certificados, creando nuevos nodos hijos para el establecimiento de múltiples conjuntos de credenciales dependientes del nodo raíz, donde podremos registrar

los dispositivos que formarán parte de la cadena de confianza y que seguirán la ruta de certificados en base a los criterios de confianza para su validación, permitiéndonos tener el control de todo el árbol.

Este prototipo inicial se ha configurado para que no sea accesible de forma remota, ya que no hemos publicado ninguna aplicación adicional en él y puede desconectarse una vez que el despliegue de dispositivos esté operativo, pudiendo hacer uso de nuevo de él, en caso de necesitar hacer modificaciones en la totalidad del árbol de confianza, siendo los nodos inferiores hijos capaces por sí solos de gestionar sus propios subárboles o crear nuevas ramas a partir de su propio nivel.

El dispositivo móvil con la aplicación *Beame-Authenticator* es necesario para la administración de usuarios y la prueba de identidad del acceso autenticado de múltiples factores a las aplicaciones publicadas en los dispositivos *IoT*.

4.1.1.4. Configuración de la aplicación móvil

Aprovisionamiento de la aplicación móvil que instalamos en el *iPhone 5, Beame Authenticator,* pareándola con nuestro servicio registrado en *beame.io* para la creación de otro *IDP* secundario, pero con múltiples factores de autenticación para administrar los usuarios y el acceso autenticado a las aplicaciones publicadas en los dispositivos *IoT*.

Desde el *IDP* inicial *Gatekeeper* generamos la *URL* de acceso con el siguiente comando:

```
beame-gatekeeper-ctl info
```

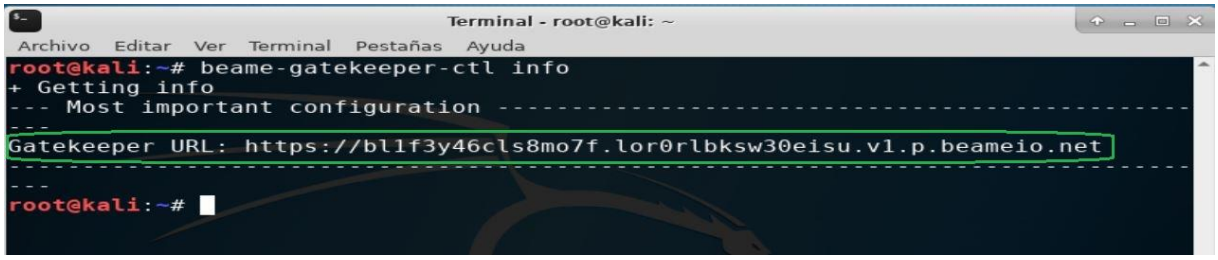

Ilustración 21. Generación de URL para registro de aplicación móvil

Al introducir esta *URL* en un navegador nos aparecerá un formulario de registro que rellenaremos con nuestros datos y el nombre público que dimos al servicio.

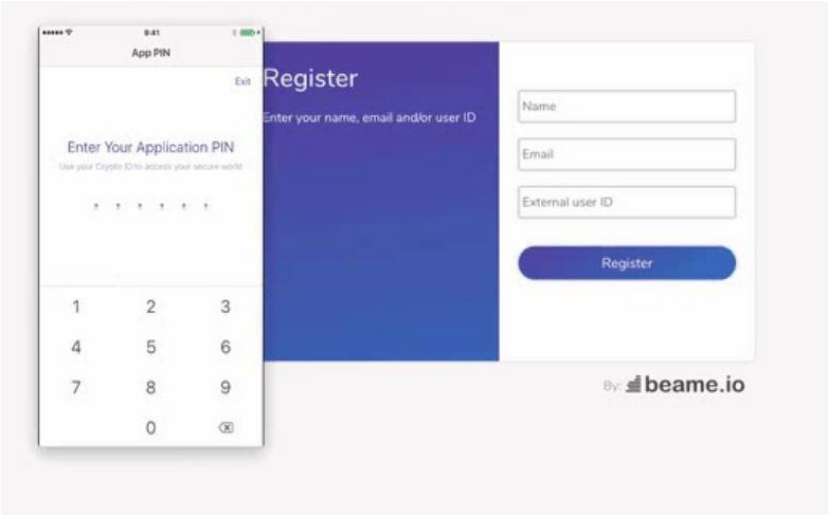

Ilustración 22. Registro de la aplicación móvil

Iniciamos la aplicación móvil *Beame-Authenticator*, seleccionando conectarse al navegador y escaneando del código *QR* que se genera, iniciando el proceso de pareado.

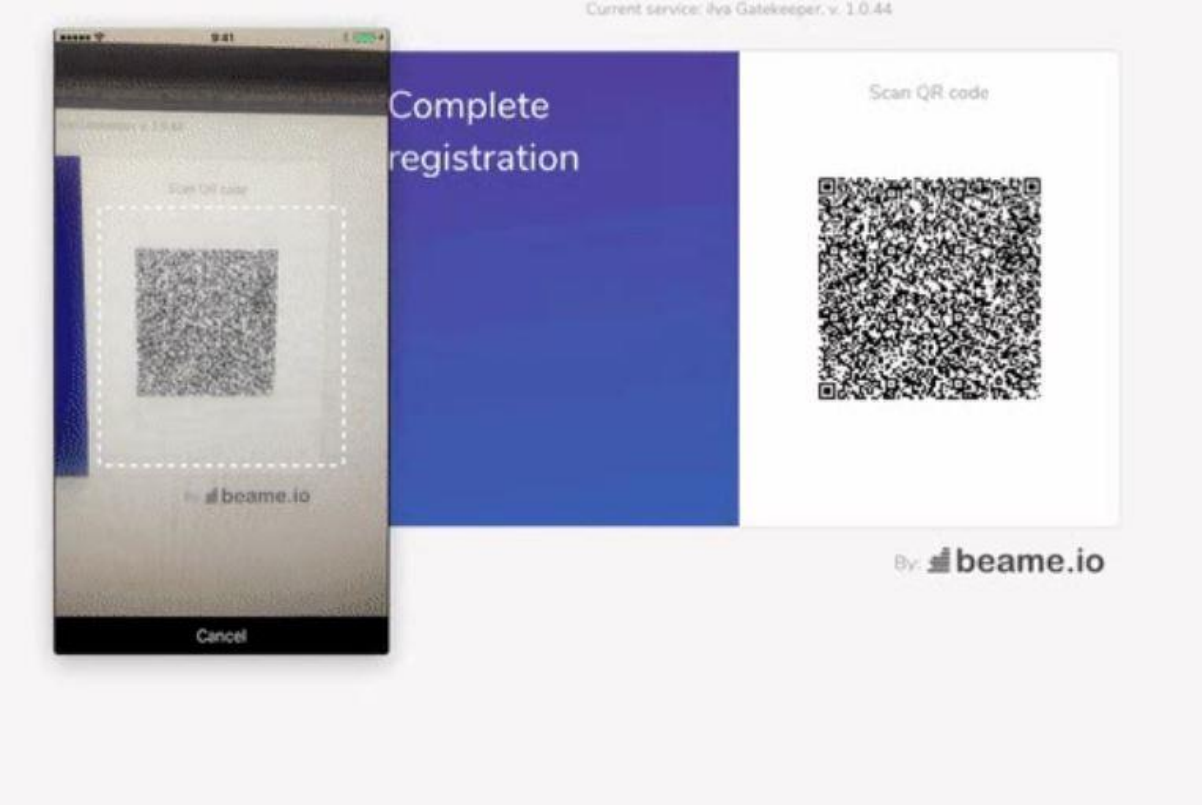

Ilustración 23. Escaneado de código QR para el registro de la aplicación móvil

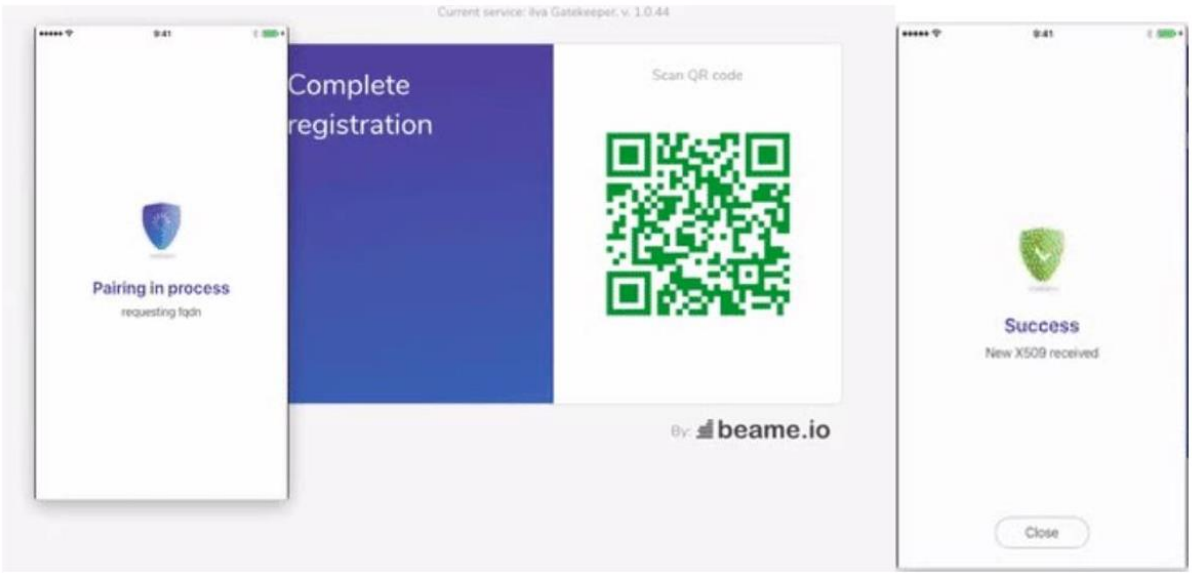

Ilustración 24. Proceso de pareado de aplicación móvil

De esta forma ya disponemos de la estructura inicial de nuestro sistema escalable para el registro y aprovisionamiento de dispositivos *IoT* con *D-PKI* y múltiples factores de autenticación mediante la aplicación móvil con el uso de la identidad criptográfica basada en *Blockchain.*

4.1.2. Prototipo *IoT01* (Vulnerable sin *D-PKI*)

Sistema básico de despliegue de un dispositivo *IoT* sensorizado, conectado a Internet y controlado remotamente mediante terminal por navegador web, que de forma deliberada se ha configurado sin una gestión del riesgo en base a la seguridad para poder evaluarlo en comparación con el despliegue asegurado.

4.1.2.1. Descripción funcional y tecnologías utilizadas

El prototipo se compone básicamente de:

- Conexión a Internet mediante router *WiFi* con direccionamiento *IP* público estático.
- Microcontrolador programable Intel Edison conectado por *WiFi* al *router* (*IoT01*).
- Código *Python* (led.py), para la captura y muestra de datos del dispositivo (Anexo III).

- Servicio emulador de terminal basado en web *WETTY*. (krishnasrinivas et al., 2014)
- Acceso remoto al dispositivo mediante los protocolos *NAT* y *HTTP* por el puerto *TCP* 8080.

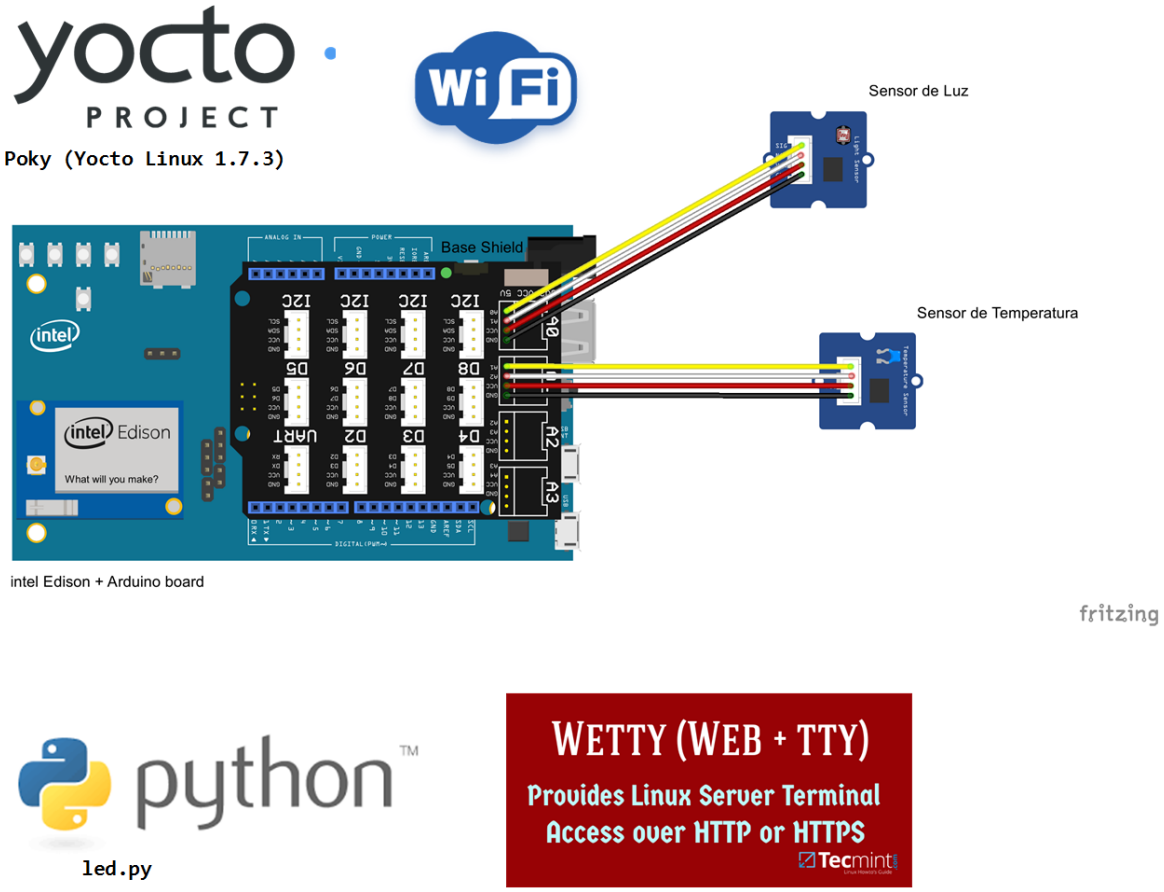

Ilustración 25. Dispositivo IoT01 Intel Edison

4.1.2.2. Mapa de conexiones

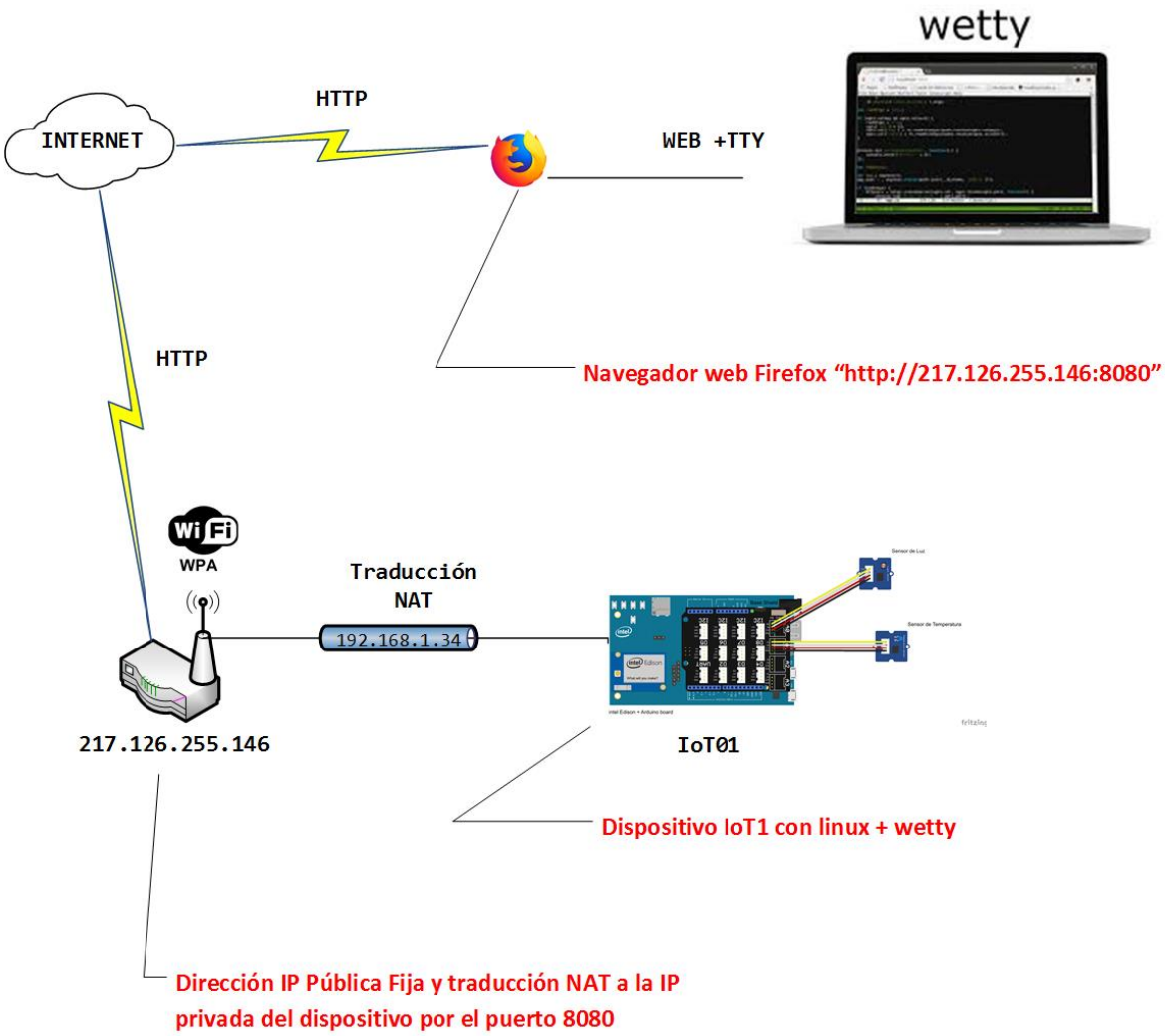

Ilustración 26. Mapa de red IoT01

4.1.2.3. Ejecución del prototipo

Una vez definidos todos los elementos del prototipo, procedemos a su ejecución:

- Desde un equipo conectado a Internet accedemos mediante el navegador *Firefox* (mozilla.org, 2018), y su barra de direcciones a la dirección del *router* del que conocemos su *IP* y el puerto *NAT,* donde está conectado el dispositivo *IoT01*, es decir: http://217.126.255.146:8080/

- Se establece conexión con el servicio *WETTY* de emulación de terminal (WEB + TTY) que está ejecutándose en el dispositivo y que nos devuelve un proceso de autenticación (*login*) en el navegador web.

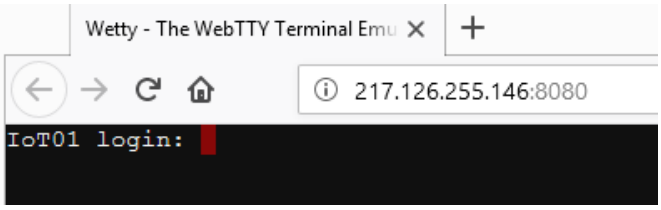

Ilustración 27. Proceso de login en IoT01

- Introducimos las credenciales de acceso y obtenemos una terminal Linux emulada y funcional con privilegios de *root* que está siendo transportada de forma insegura mediante el protocolo *HTTP* sin cifrado y de la que se podría capturar datos en claro.

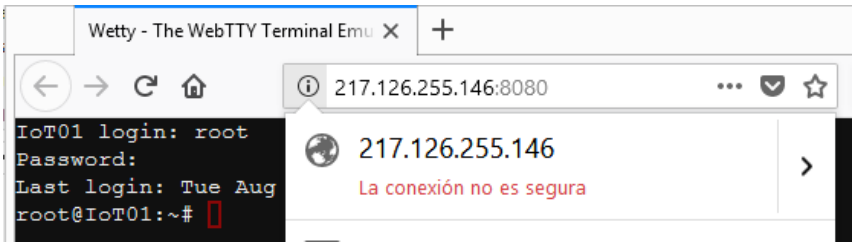

Ilustración 28. Acceso inseguro por HTTP en IoT01

- Ejecutamos el código *Python* (Anexo III).

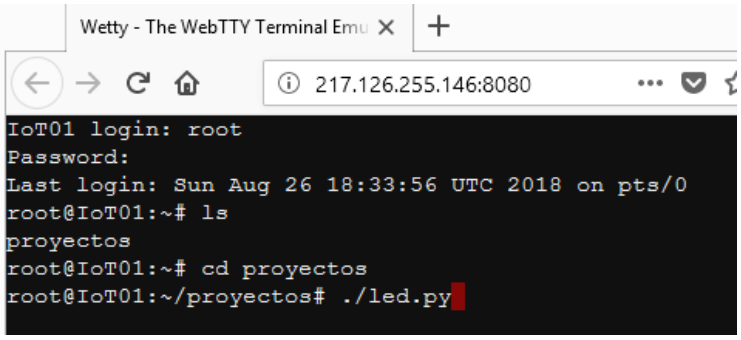

Ilustración 29. Ejecución del código python en IoT01

- Obtenemos la salida de la información capturada del dispositivo con el código *Python*.

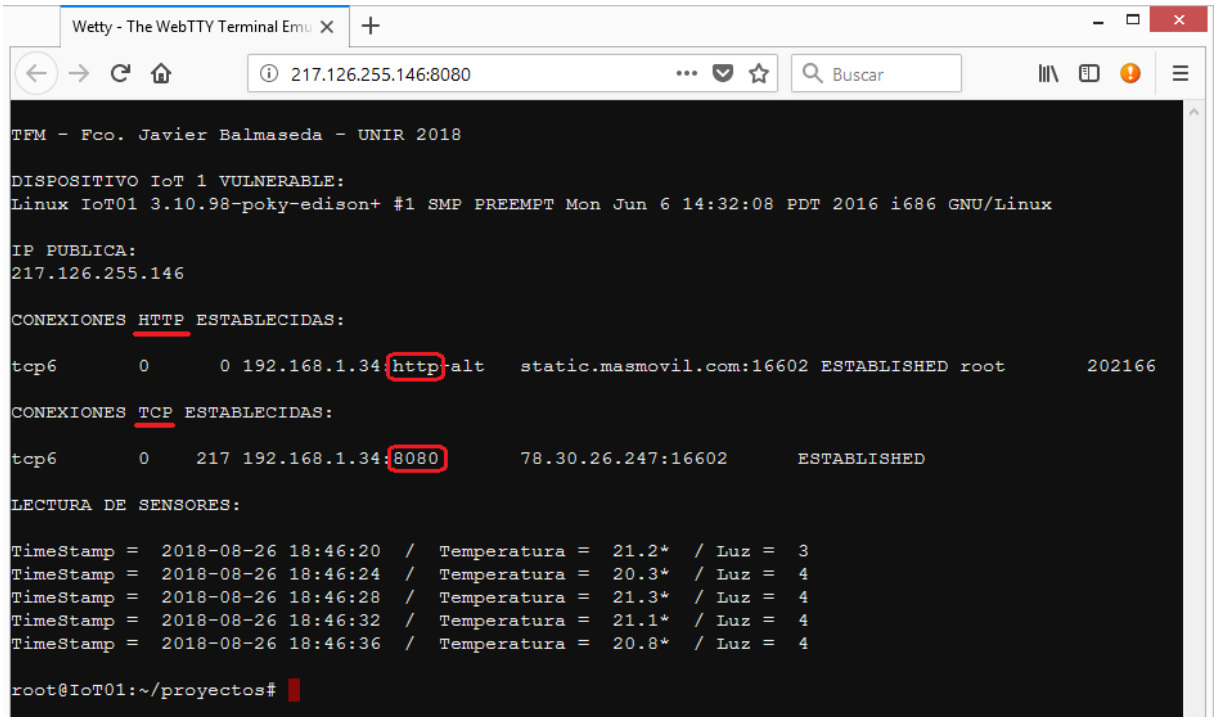

Ilustración 30. Salida de datos capturados en IoT01

4.1.2.4. Valoración preliminar

Es una configuración potencialmente insegura y sin una gestión del riesgo apropiada, principalmente porque todo el tráfico que viaja entre el dispositivo y el equipo remoto que lo administra no está cifrado, y en caso de ser capturado podría revelar datos en claro o débilmente protegidos (enmascaramiento), otros datos sensibles, como credenciales de acceso, u otros datos, como los valores devueltos por los sensores.

4.1.3. Prototipo *IoT02* (Asegurado con *PKI + Blockchain*)

Los dispositivos conectados mediante este sistema e inscritos en la *Blockchain* pública de *beame.io* mediante gestión del acceso confiable se comportarán como servidores web públicos que permitirán el intercambio de datos cifrados *App-to-App* mediante tráfico firmado con túnel *HTTPS* como medio de transporte, sobre redes no confiables o con protocolos inseguros, Internet o redes de área local, gracias a los certificados *X.509* con clave *RSA* de 2048 *bits,* permitiendo servicios como el acceso remoto, monitoreo y administración, gracias al uso del estándar nativo *SSL/TLS* de los navegares web.

Este sistema está basado en una *PKI* pública basada en *Ethereum,* donde cada identidad en el sistema tiene un nombre único bajo el subdominio *beameio.net*.

IDENTIDAD DEL DISPOSITIVO

- *FQDN (Fully Qualified Domain Name)* – Nombre de host único bajo el subdominio beameio.net generado aleatoriamente en el registro del dispositivo, que es la *URL* de acceso.
- Certificado digital *X.509* y prueba de identidad basada en aplicación móvil.
- Capacidad de emitir *tokens* adicionales para la creación de más certificados.
- Nombre privado para configurar el servicio.

ACCESO CIFRADO AL DISPOSITIVO

- Canal autenticado y cifrado sin dirección *IP* pública, sólo por *HTTPS*.
- Clave pública conocida.
- Libro mayor del *Blockchain* que facilite la autorización verificando la validez del certificado.
- Aplicación móvil *beame-authenthicator* para permitir al dispositivo reconocer los certificados del cliente o su posesión.

4.1.3.1. Descripción funcional y tecnologías utilizadas

El prototipo se compone básicamente de:

- Conexión a Internet mediante *router WiFi* con direccionamiento IP público dinámico.
- Microcontrolador programable *Raspberry Pi 3 Model B+* conectado por *WiFi* al *router* (*IoT02*).
- Código *Python* (sensores.py) para la captura y muestra de datos del dispositivo (Anexo IV).
- Servicio emulador de terminal basado en web *WETTY*.
- *IDP – Identity Provider* – Proveedor de identidades – *Front End* – Aplicaciones de *beame.io* (*gatekeeper* + aplicación móvil en *iPhone 5*).
- *SP – Securit Provider* – Proveedor de seguridad – *Back End – Blockchain Ethereum*.
- Intercambio de Información de identidad federada entre *IDP* y *SP* con *SAML* mediante el navegador web y la aplicación móvil.

- Acceso seguro al dispositivo por terminal mediante túnel *HTTPS* cifrado y *MFA*.

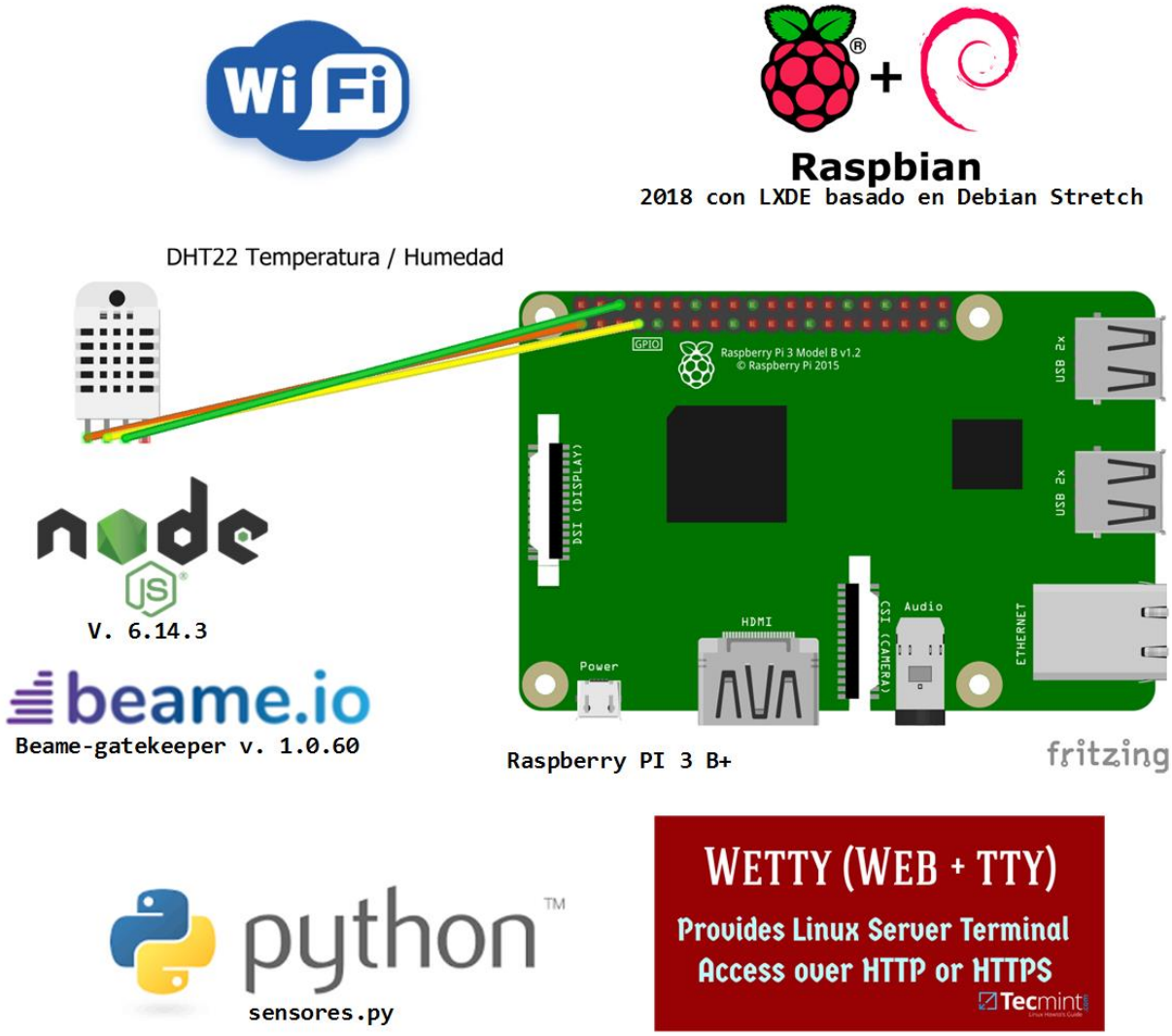

Ilustración 31. Dispositivo IoT02 Raspberry Pi3 B+

4.1.3.2. Herramientas de implementación

- En el microcontrolador:
  - *Node.js v.6.14.3.*
  - *Python* y *wetty.*
  - SERVIDOR *beame-gatekeeper v. 1.0.60* + *token* de registro.
  - Emisión y validación de credenciales de identidad para la autenticación.
  - Almacenamiento y administración de certificados.

- Entidad que mantiene el libro mayor Blockchain y hace copia de seguridad.
- Proporciona respuestas a la autenticación mediante *SAML*.

4.1.3.3. Mapa de conexiones

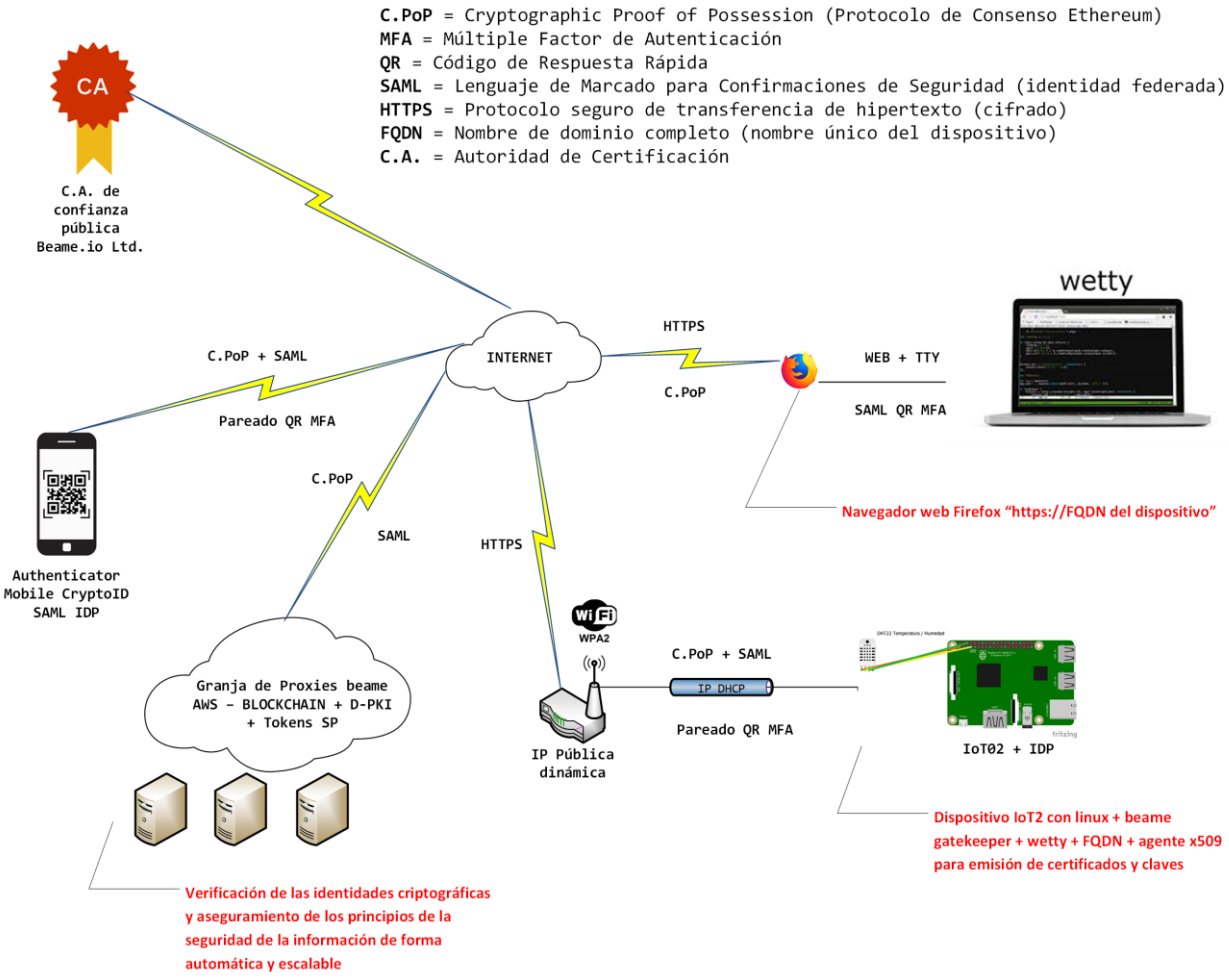

Ilustración 32. Mapa de red I0T02

4.1.3.4. Registro del dispositivo e instalación

En este paso es necesario recurrir a nuestro *IDP* inicial y su interfaz de administración, accediendo de manera local para generar un nuevo subnivel en el árbol de confianza ya creado, para que la identidad nueva del dispositivo *IoT02* que registremos dependa criptográficamente del nivel superior siguiendo la estructura en capas y que exista una cadena de certificados como múltiples conjuntos de credenciales.

Utilizaremos el nivel superior L0 como certificado raíz del cliente *SSL/TLS* creado en el *IDP* inicial, y en concreto el del *GatewayServer,* para crear el nuevo nodo hijo *RPI B+* con la opción de creación "*Create Child*", lo que nos abrirá un formulario de creación de nuevo certificado hijo local para administrar la identidad del dispositivo; nos situaremos en este nuevo nodo y usaremos la opción de creación de *token* de registro "*Create Registration Token",* lo que nos proporcionará el token codificado en Base 64 necesario para configurar en el dispositivo *IoT02* otro servidor *beame-gatekeeper* como un servicio que se ejecuta en el sistema al iniciar el dispositivo y poder disponer de nuestro *IDP* adicional en funcionamiento, que nos servirá para seguir estableciendo la lógica del árbol de confianza, creando el certificado de la cadena del siguiente nivel, "L1".

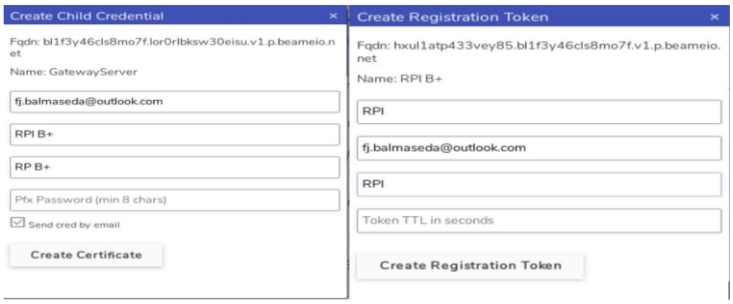

Ilustración 33. Formularios de creación de certificado y Token de registro

La instalación del servidor *beame-gatekeeper* en el dispositivo *IoT02* se realiza exactamente igual que en el punto 4.1.1.4, a excepción de la parte de configuración de la aplicación móvil, que ya no es necesaria, quedando el árbol de confianza visto desde la interfaz de administración del *IDP* inicial de la siguiente manera:

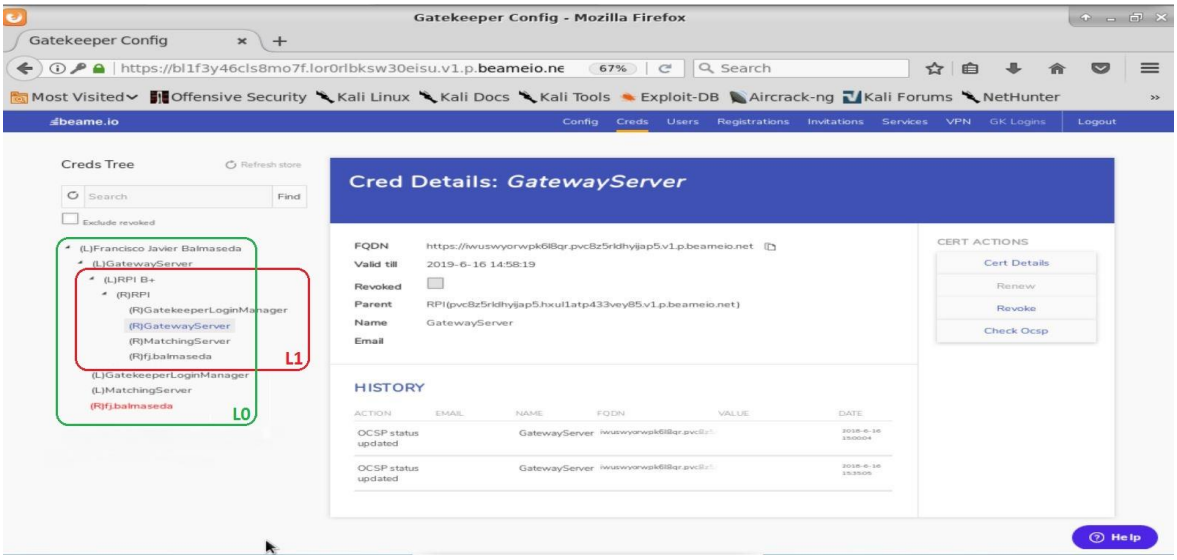

Ilustración 34. Estado del árbol de confianza desde IDP inicial

Podemos observar cómo el árbol de confianza de credenciales estructurado en capas ha crecido desde el nivel superior L0 (*Highest FQDN*) de la cadena de certificados, con el certificado raíz, y desde el certificado local (L) asociado del *GatewayServer*, y cómo ahora aparecen el certificado local (L) del nuevo dispositivo *RPI B+* con el siguiente nivel L1 y el certificado remoto (R) *RPI*, así como el de los servicios remotos (R) asociados del *GatewayServer, LoginManager, MatchingServer* y el usuario remoto (R) fj.balmaseda.

De especial importancia es el *FQDN* del nuevo *GatewayServer*, ya que será la *URL* de acceso web al dispositivo *IoT02* desde Internet, aunque también se resuelve por *DNS* desde el *FQDN* de la identidad principal *RPI B+*:

```
https://iwuswyorwpk6l8qr.pvc8z5rldhyijap5.v1.p.beameio.net
```

Ilustración 35. Detalle nuevo árbol de confianza

Desde un navegador en el mismo dispositivo *IoT02* accedemos a la interfaz de administración con la *URL* temporal generada por el comando:

```
beame-gatekeeper-ctl admin
```

Podemos observar el árbol de confianza visto desde la interfaz de administración del dispositivo *IoT02,* donde sólo es posible ver el nivel L1.

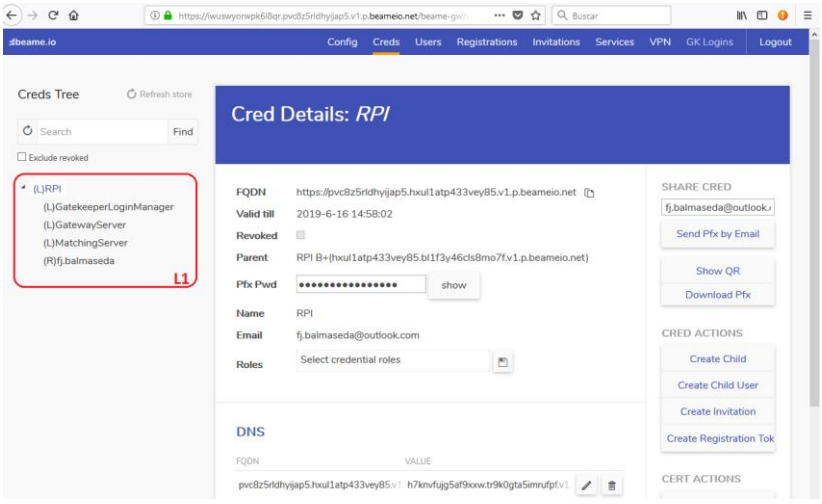

Ilustración 36. Estado del árbol de confianza desde IoT02

Configuramos desde la sección de servicios para que la aplicación *WETTY* sea publicada por el puerto 8556 desde el servidor *Gatekeeper* del *localhost* una vez que el acceso ha sido permitido por el servicio de autenticación y que se aprovisione en la aplicación móvil *Authenticator* para su validación.

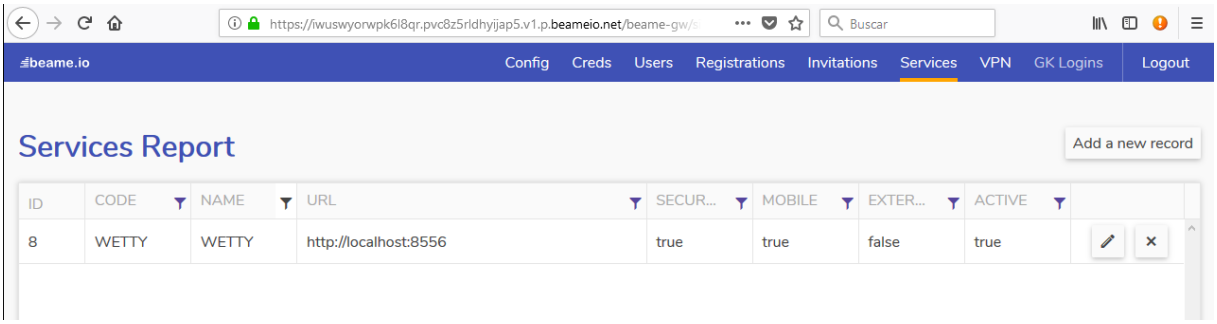

Ilustración 37. Servicio WETTY en Gatekeeper de IoT02

4.1.3.5. Ejecución del prototipo

Con toda la configuración del piloto terminada procedemos a la ejecución del prototipo *IoT02*:

- introduciendo la *URL* del *FQDN* del *GatewayServer* del dispositivo *RPI* del nivel 1 en cualquier navegador compatible con *SSL/TLS* de un dispositivo conectado a Internet.

- El navegador extrae el código *HTML* de inicio de sesión del servicio de autenticación y lo presenta al usuario en forma de código *QR*:

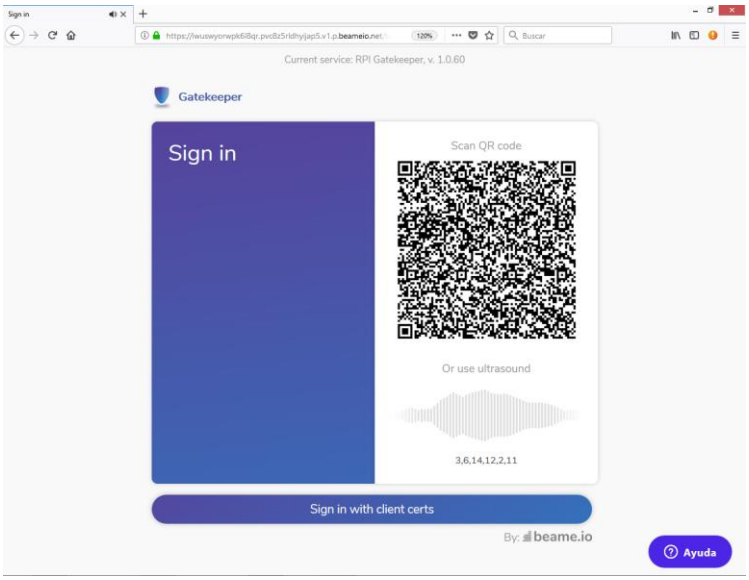

Ilustración 38. Inicio de sesión mediante QR

- Con el dispositivo móvil y la aplicación *Authenticator* escaneamos el código *QR* que demuestra la posesión de la clave privada, ya que se crea un *token* firmado que contiene el *FQDN* que el usuario obtuvo en el registro.

- El servicio de autenticación valida el *token* y busca el *FQDN* en su base de datos de clientes de la *PKI* bajo la *Blockchain Ethereum* de *beame.io*.

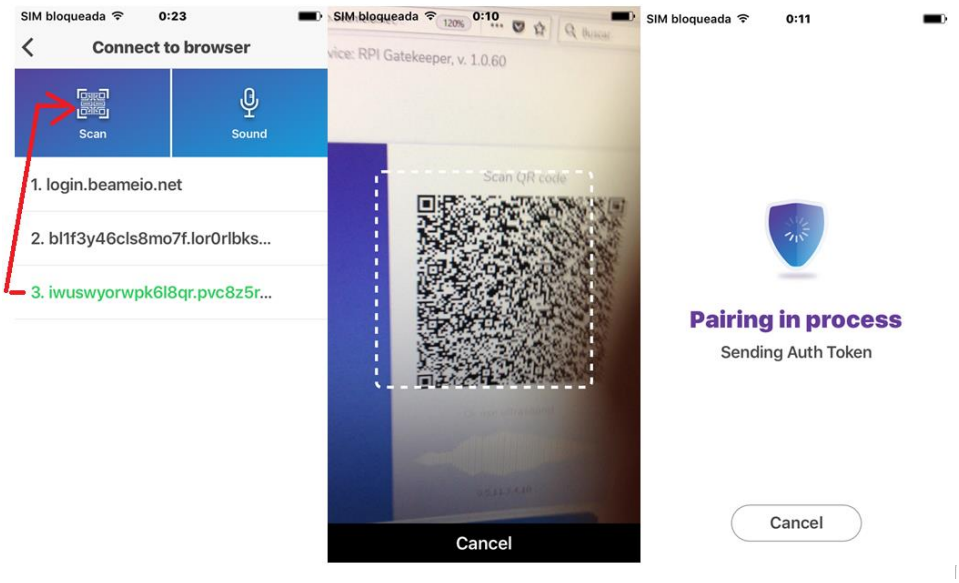

Ilustración 39. Proceso de autenticación con código QR

- Si se valida el *token* y el *FQDN* es encontrado, se permite el acceso al cliente a los servicios publicados y seleccionamos *WETTY* desde la aplicación móvil.

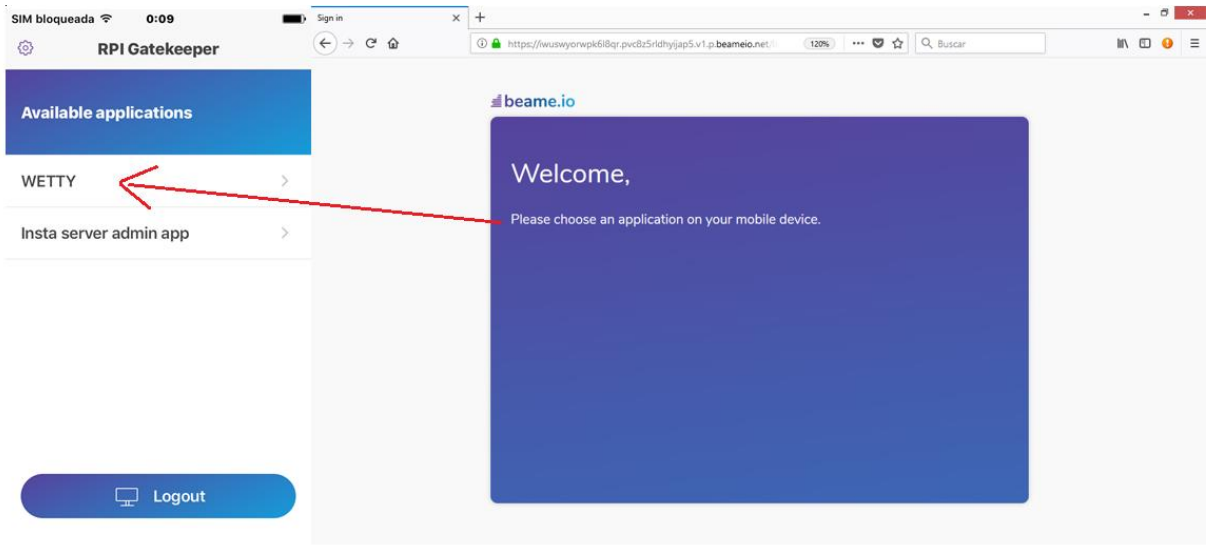

Ilustración 40. Acceso a los servicios publicados desde la aplicación móvil.

- Se establece conexión *HTTPS* cifrada con el servicio *wetty* de emulación de terminal (*web* + *tty*), que está ejecutándose en el dispositivo y que nos devuelve un proceso de autenticación (*login*) en el navegador web.

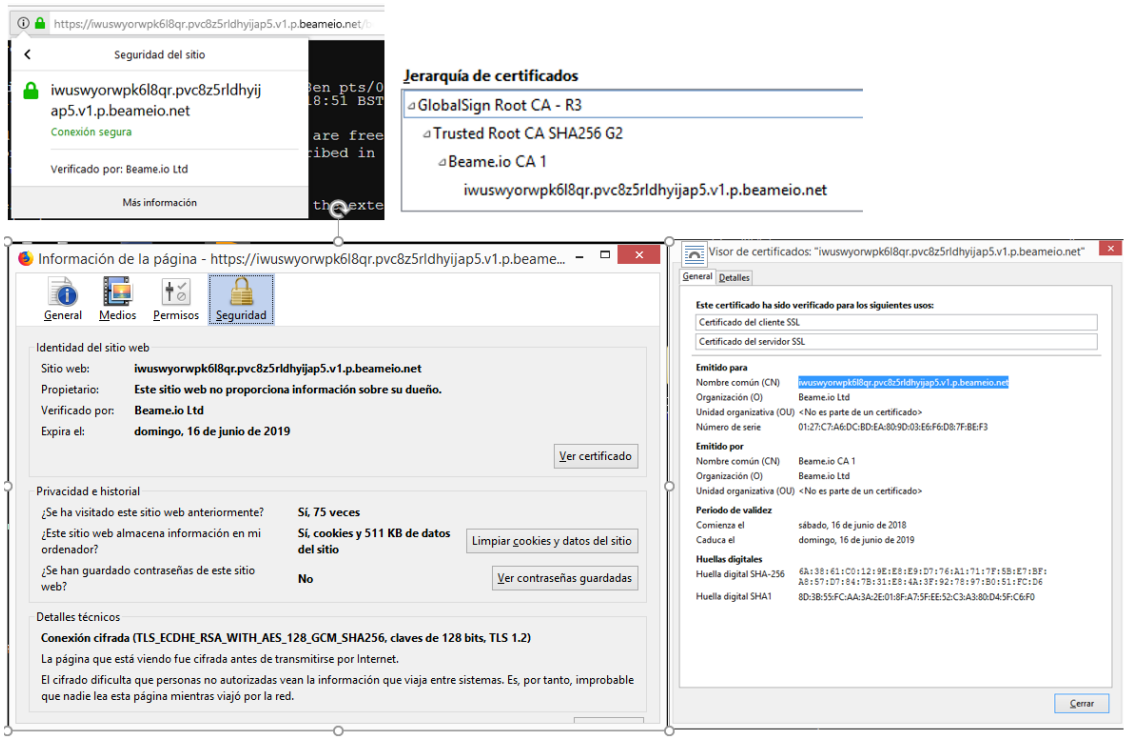

Ilustración 41. Información de certificado y su jerarquía en conexión segura con IoT02

- Introducimos las credenciales de acceso y obtenemos una terminal *Linux* emulada y funcional con privilegios de *root* que está siendo transportada de forma segura mediante el protocolo *HTTPS* con cifrado.

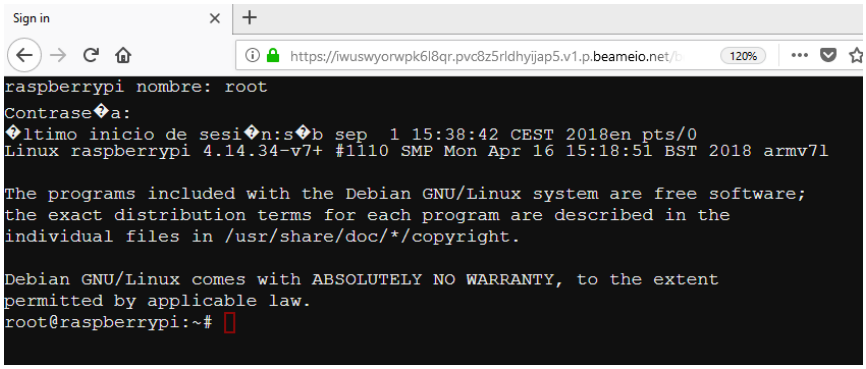

Ilustración 42. Acceso root mediante terminal emulado en IoT02

- Ejecutamos el código *Python* (Anexo IV).

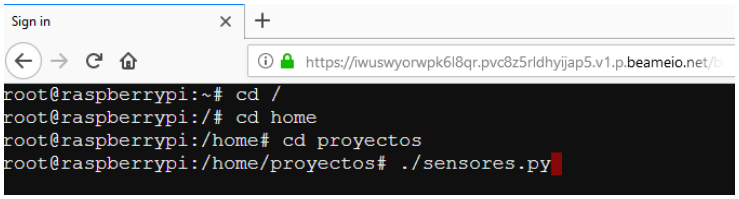

Ilustración 43. Ejecución del código python en IoT02

- Obtenemos la salida de la información capturada del dispositivo con el código *Python*.

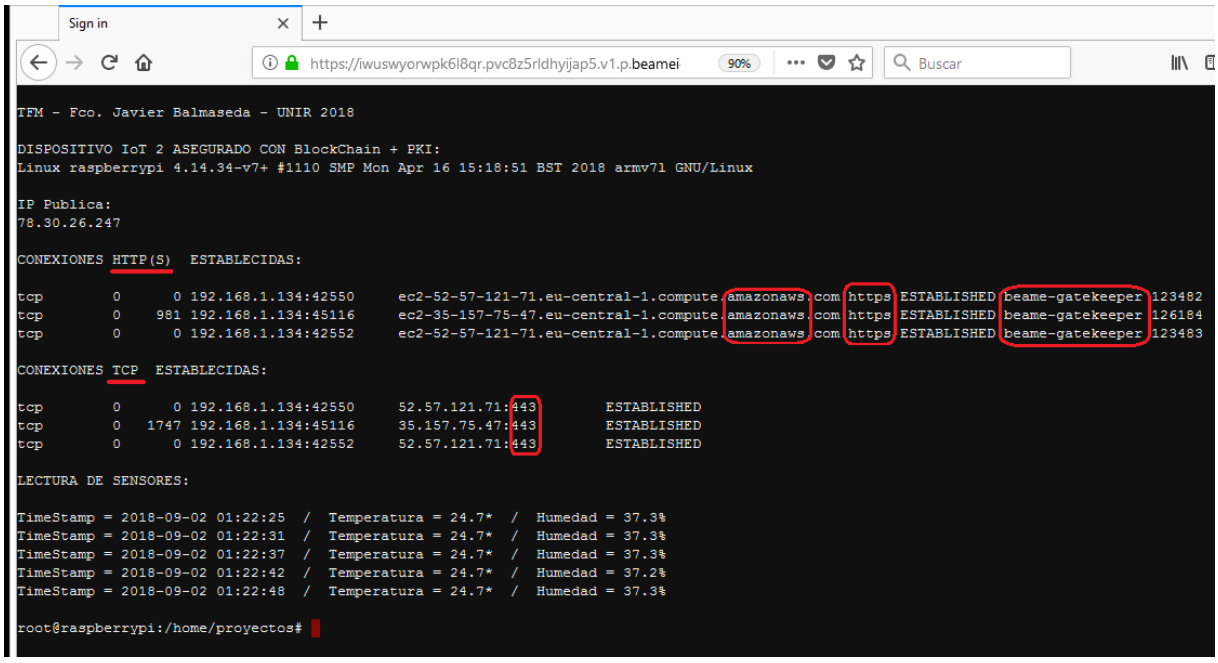

Ilustración 44. Salida de datos capturados en IoT02

4.1.3.6. Valoración preliminar

En este caso de uso ya disponemos de un despliegue donde la gestión de acceso al dispositivo se realiza de forma segura y confiable, gracias al uso de trafico de transporte cifrado y a la implementación de identidades criptográficas basadas en *Blockchain* que residen en cada parte del ecosistema de la *D-PKI,* interactuando de manera que se puedan verificar en un espacio de nombres universal que comprueba que las claves de acceso son las correctas y permiten mantener el control sobre el entorno desplegado.

4.2. Aplicación de la metodología

4.2.1. Revisión de la seguridad

En este apartado se ha diseñado un cuestionario de seguridad para revisar el cumplimiento de las pruebas definidas en los apartados *3.3.4* y *3.3.5* de la metodología de trabajo *EDR*.

Tabla 9. Cuestionario EDR

OWASP IoT Top 10 - 2014	Pregunta	SÍ / NO
R01: Interfaz Web insegura	¿Se ha modificado el acceso a los dispositivos con los usuarios y contraseñas por defecto?	
R02: Autenticación / Autorización insuficientes	¿Los usuarios acceden al dispositivo mediante el uso de identidades verificadas con IDP, SP y MFA?	
R03: Servicios de red inseguros	¿Están sólo expuestos a Internet los puertos TCP/UDP necesarios para su funcionamiento?	
R04: Falta de Cifrado de transporte	¿Está la comunicación cifrada mediante certificados X.509 con claves RSA de 2048 bits creadas en el dispositivo y los tokens de autorización firmados que se envían a la PKI junto con la clave pública firmada?	
R05: Privacidad	¿Están los datos de acceso al dispositivo desanonimizados o son FQDN generados aleatoriamente?	
R06: Interfaz Cloud insegura	¿Está el intercambio de datos de autenticación y autorización de los dispositivos con los proveedores de servicios Cloud protegidos con estándares seguros como SAML?	
R07: Interfaz Móvil insegura	¿Está la conexión inalámbrica del dispositivo protegida con protocolos seguros de conexión como WPA2?	
R08: Configuración de Seguridad insuficiente	¿Se impide ejecutar scripts con los usuarios no administradores / root?	
R09: Software / Firmware inseguro	¿Existe un servicio de soporte activo de actualizaciones del firmware del dispositivo?	
R10: Pobre seguridad física	¿Está el dispositivo físicamente ubicado en un lugar que no es de libre acceso?	

4.2.2. Obtención del riesgo final

Vamos a calcular el riesgo final de los prototipos *IoT* en base al anterior cuestionario y el riesgo inicial, donde las respuestas que se correspondan con SÍ, restarán un 1 en el cómputo de cada riesgo, y las que se correspondan con NO, no restarán, (anexos V y VI).

4.2.2.1. Riesgo final IoT01

Tabla 10. Riesgo final IoT01

OWASP	Riesgo Inicial	SÍ / NO	Riesgo Final	Observaciones
R01	3	SÍ -1	2	Necesario cambio de contraseña al instalar
R02	3	NO	3	Sólo credenciales nativas del dispositivo
R03	2	NO	2	TCP: 90, 554, 8080, 9000
R04	3	NO	3	Comunicación no cifrada con protocolo HTTP
R05	3	NO	3	Los datos de acceso al dispositivo son la IP pública del mismo y el puerto TCP
R06	3	NO	3	No hace uso de SAML
R07	3	NO	3	WPA vulnerable
R08	2	NO	2	Sólo dispone de un usuario (root)
R09	2	NO	2	Poky (Yocto Linux 1.7.3) DESCONTINUADO
R10	2	SÍ -1	1	Situado en un CPD
TOTAL:	26		24	

4.2.2.2. Riesgo final IoT02

Tabla 11. Riesgo final IoT02

OWASP	Riesgo Inicial	SÍ / NO	Riesgo Final	Observaciones
R01	3	SÍ -1	2	Cambiada contraseña usuario root
R02	3	SÍ -1	2	IDP inicial y secundario SP Blockchain y QR
R03	2	SÍ -1	1	No accesible directamente desde Internet sin conocer FQDN
R04:	3	SÍ -1	2	D-PKI bajo Blockchain con certificados X.509 clave privada RSA 2048 en el dispositivo
R05	3	SÍ -1	2	Nombre único de host FQDN creado aleatoriamente en el registro
R06	3	SÍ -1	2	Aserciones SAML entre IDPs y SP
R07	3	SÍ -1	2	WPA2
R08	2	SÍ -1	1	Scripts sólo ejecutables con root
R09	2	SÍ -1	1	Raspbian 2018 con soporte activo
R10	2	SÍ -1	1	Situado en CPD
TOTAL:	26		16	

4.2.3. Evaluación de la metodología

Evaluaremos el funcionamiento de la metodología mediante la realización de los análisis de seguridad de los dispositivos *IoT01* e *IoT02* en base al supuesto de que el riesgo ideal es el de que las 10 categorías *OWASP IoT* arrojasen un valor de 1, es decir, riesgo bajo, con un total de 10 puntos de cómputo total.

4.2.3.1. Análisis de seguridad IoT01

Tabla 12. Riesgos IoT01

OWASP	Riesgo Inicial	Riesgo Final	Riesgo Ideal
R01	3	2	1
R02	3	3	1
R03	2	2	1
R04:	3	3	1
R05	3	3	1
R06	3	3	1
R07	3	3	1
R08	2	2	1
R09	2	2	1
R10	2	1	1
TOTAL:	26	**24**	10

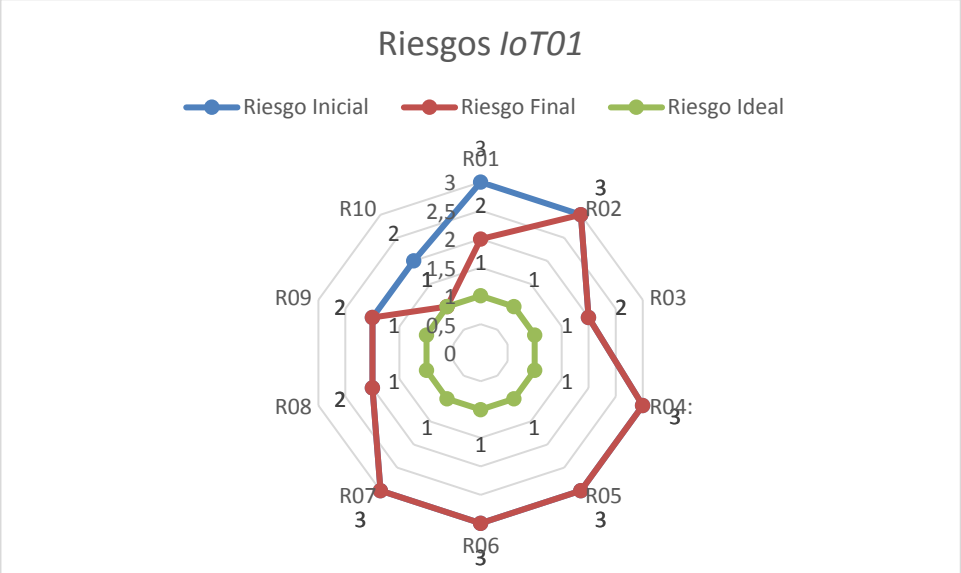

Ilustración 45. Gráfico de riesgos IoT01

Para reducir el riesgo final de este dispositivo se le podrían añadir, en los casos que se puedan implementar, los controles definidos en la metodología *EDR* y volver a evaluar el sistema; en nuestro caso nos sirve, de esta forma, como referencia de dispositivo vulnerable, al objeto de la comparación con el dispositivo asegurado.

4.2.3.2. Análisis de seguridad IoT02

Tabla 13. Riesgos IoT02

OWASP	Riesgo Inicial	Riesgo Final	Riesgo Ideal
R01	3	2	1
R02	3	2	1
R03	2	1	1
R04:	3	2	1
R05	3	2	1
R06	3	2	1
R07	3	2	1
R08	2	1	1
R09	2	1	1
R10	2	1	1
TOTAL:	26	**16**	10

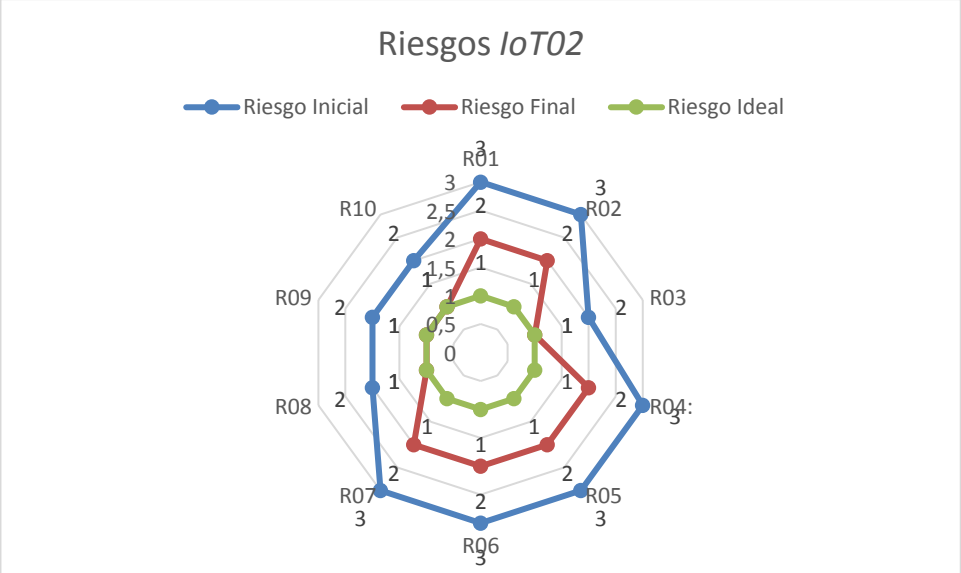

Ilustración 46. Gráfico de riesgos IoT02

Para reducir el riesgo final de este dispositivo se podrían añadir nuevos objetivos de control, controles y pruebas en aquellos riesgos donde no se ha conseguido rebajar el riesgo a niveles bajos (R01, R02, R04, R05, R06 y R07), y volver a realizar una evaluación de la seguridad.

4.2.4. Comparativa de resultados

En base a los anteriores resultados y haciendo una comparación del riesgo final de los dos dispositivos *IoT* y el riesgo inicial e ideal, podemos observar cómo el despliegue de la *D-PKI* basada en *Blockchain* del dispositivo *IoT02* es el que consigue rebajar más los índices de riesgo hasta el valor 16 y se acerca más a un hipotético nivel de riesgo ideal con valor 10; de hecho, no contiene ningún riesgo de nivel 3 o ALTO, en contraposición a los 26 puntos del dispositivo IoT01, que además queda aún con cinco indicadores de riesgo de nivel ALTO.

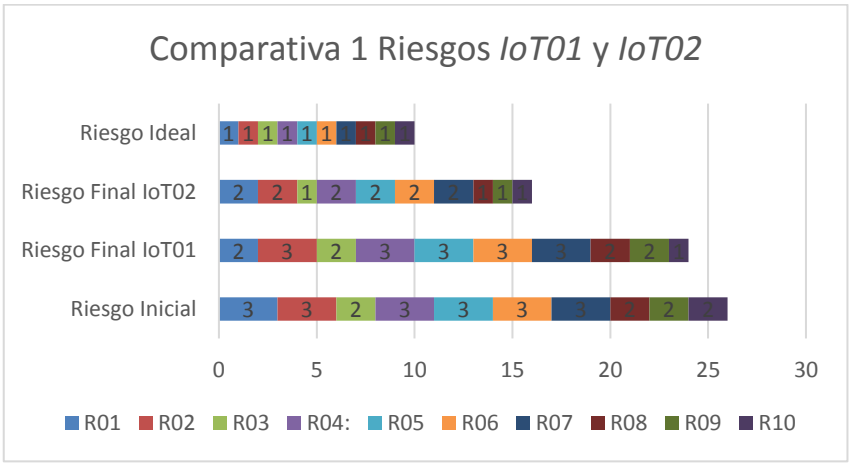

Ilustración 47. Gráfico comparativo de riesgos 1

	R01	R02	R03	R04:	R05	R06	R07	R08	R09	R10
Riesgo Ideal	1	1	1	1	1	1	1	1	1	1
Riesgo Final IoT02	2	2	1	2	2	2	2	1	1	1
Riesgo Final IoT01	2	3	2	3	3	3	3	2	2	1
Riesgo Inicial	3	3	2	3	3	3	3	2	2	2

Ilustración 48. Gráfico comparativo de riesgos 2

4.2.5. Posibles mejoras de la metodología

Es evidente que en el marco de una auditoría de seguridad de evaluación del riesgo *EDR* se pueden aumentar los parámetros en función de los riesgos detectados, es decir, aumentar los controles y las pruebas en base al grado de seguridad que queramos implementar, así como aumentar las muestras que nos ayuden a determinar qué pruebas son más eficaces y añadir nuevos cuestionarios para su evaluación.

En este trabajo, y para la identificación de los riesgos como indicadores de compromiso del sistema, nos hemos ceñido a la clasificación de vulnerabilidades *OWASP top 10 2014*, pero existen otras clasificaciones, como "*Payatu IoT Top Ten Vulnerabilities*". (Payatu, 2015)

4.3. Amenazas al sistema y remediación

En este apartado vamos a poner el foco en otros posibles indicadores de compromiso del sistema *IoT02* que pudieran ser objeto de nuevas auditorías de seguridad y proponer posibles actuaciones para mitigar estos escenarios.

4.3.1. Compromiso de certificados

Si los certificados de alguna entidad son comprometidos y se lograse superar los controles de seguridad añadidos como *PoP*, se podría suplantar su identidad en el sistema y operar de forma ilegítima.

La propuesta de este trabajo permite, una vez detectado este compromiso, el uso del parámetro de tiempo especificándolo mediante la verificación y correlación de eventos, para que a partir de ese momento se puedan tomar decisiones y poder cancelar la confianza en esa entidad gracias a la *D-PKI*, donde la historia de las entidades se mantiene actualizada en el libro mayor de *Blockchain*, pudiendo dar respuesta al compromiso revocando el certificado, al tener delegada y descentralizada su confianza mediante las herramientas de administración del sistema de los *IDP*.

4.3.2. Inseguridad de las *C.A.* privadas

El uso de Autoridades de Certificación privadas supone que los certificados necesarios para el cifrado de la información estén autofirmados, al no pertenecer a una *C.A.* de empresas públicas, escenario típico de las *VPN* que utilizan estas *C.A.* privadas para firmar ambos certificados, que se utilizan para autenticar tanto al servidor como al cliente.

El hecho de no existir una *C.A.* de confianza pública en la cadena de certificados (ruta de confianza), implica necesariamente la instalación de un certificado de *C.A.* privado en el dispositivo, lo que permite en escenarios de compromiso de este certificado que se pueda descifrar la comunicación del dispositivo, e incluso la emisión de nuevos certificados falsos para acceder al sistema, donde los mecanismos de revocación de certificados requieren de una configuración adicional que en numerosas ocasiones se descuida.

En este proyecto el planteamiento sobre los certificados que son emitidos con *gatekeeper* y están firmados por una *C.A.* de confianza pública es distinto, no siendo necesario que existan certificados de una *C.A.* privada adicionales instalados en el dispositivo. Se crean archivos de configuración de las conexiones cifradas que se actualizan mediante el escaneo de los códigos *QR* que muestra la interfaz, actualizando estos archivos cada minuto, agregando identidades criptográficas nuevas y eliminando las que estuvieran revocadas, llegando a desconectar a los usuarios si fuese necesario.

4.3.3. *TLS Termination* / *SSL Inspection*

El protocolo de transferencia *HTTPS* hace uso de otro protocolo criptográfico de bajo nivel, *SSL/TLS,* para proteger mediante criptografía la comunicación con los recursos de forma autenticada, pero en entornos de red donde existen *proxies*, equilibradores o balanceadores de carga con terminación *TLS* (*TLS Termination*), o Inspección *SSL* (*SSL Inspection*), las conexiones seguras entrantes son descifradas en ese punto y pasan la solicitud no asegurada a otros servidores del despliegue, permitiendo reducir la carga en los servidores principales al no tener que realizar los procesos criptográficos necesarios para el descifrado del tráfico, es decir, se rompe el cifrado extremo a extremo antes de que se complete, dejando en parte del recorrido interno el tráfico sin cifrar, y por tanto expuesto a ser capturado en claro. (Wikipedia, 2018)

4.4. Alternativas a la *D-PKI*

4.4.1. *PKI* Convencional

Ecosistema que permite la creación y gestión de certificados digitales *X.509* mantenido globalmente por las Autoridades de Certificación *CA*, que asegura los principios de la seguridad de la información, que ha sido probado en Internet por más de tres décadas, proporcionando gran interoperabilidad y compatibilidad con el software y los protocolos existentes, ya que hace uso de estándares públicos y abiertos, siendo masivamente escalable. Su mayor implementación está en el uso del protocolo de cifrado *SSL* en servicios web.

En nuestro caso, el piloto del dispositivo asegurado *IoT02* bajo una *D-PKI*, en principio se podría implementar sobre una *PKI* tradicional, ya que ésta daría soporte al uso de los certificados digitales *X.509,* haciendo las veces de tercero de confianza que facilita el vínculo entre entidades y claves criptográficas, sustituyendo el despliegue sobre *Blockchain*, permitiendo los servicios de cifrado y firma digital entre las partes autenticadas; de hecho esta implementación sería más sencilla de desplegar.

Pero los nuevos escenarios de despliegues de dispositivos *IoT* exigen nuevas consideraciones para asegurar las identidades de todos estos dispositivos de manera automática, y de tal forma que aborde todo el ecosistema (aprovisionamiento) que *PKI* por sí sola no provee. Tampoco se mantendría una historia de eventos común de las identidades en un libro mayor y no dispondríamos de trazabilidad e integridad sobre ellas, ya que las identidades no residen en cada parte del ecosistema que interactúa con los datos, los usuarios o los dispositivos de forma verificable y con un espacio de nombres universal que compruebe que las claves sean las correctas.

En términos de seguridad en una *PKI* tradicional las claves privadas van en tránsito desde un tercero de confianza que las genera y no dispondríamos de control sobre los certificados perdidos o robados o si el dispositivo es vulnerado o suplantado, al no disponer de eventos comunes, ni una lógica flexible, lo que nos impediría la toma de decisiones que nos permitiera limitar los daños en escenarios de compromiso.

Otra de las desventajas del uso de este sistema sería la incapacidad de establecer sistemas de control de acceso de grano fino u otros problemas comunes a las *PKI,* como *Key Rotation, Revocation Reissue* y *Renewal*.

4.4.2. Red Privada Virtual *VPN*

Para establecer un canal seguro entre dispositivo y administrador se podría implementar el uso de una *VPN* de acceso remoto que crease un túnel cifrado. Este escenario en sí es seguro y funcional, pero es la gestión de los certificados y claves necesarias para ello donde radica el problema principal, ya que sería necesario trasladar estos certificados hasta el dispositivo y el otro extremo de alguna manera, lo que podría suponer un escenario de riesgo, y donde la escalabilidad del sistema sería un problema añadido, ya que estos despliegues no están pensados para implementarse de forma masiva.

También es necesaria la instalación de software específico para que la *VPN* se complete en ambos extremos, lo que dificulta los grandes despliegues o el simple aprovisionamiento de un nuevo dispositivo, que unidos a la complejidad de los ecosistemas *IoT,* donde es necesario un control y administración de las autorizaciones y un control de accesos, hacen que esta alternativa no sea la más recomendable.

Otro inconveniente de esta alternativa es la necesidad de disponer de direcciones *IP* públicas accesibles para los dispositivos en lugar de nombres únicos de dispositivos asociados a un certificado, como es el caso del despliegue con *D-PKI* que evita conexiones no autorizadas en la capa de red.

5. Aplicabilidad del sistema

5.1. Escalabilidad

En el sistema propuesto en este trabajo sólo se ha desplegado un *IDP* inicial y un dispositivo *IoT* asegurado mediante aplicación móvil, pero por su diseño en capas permite una escalabilidad donde los dispositivos del nivel superior son capaces de autorizar a otros dispositivos de nivel inferior, de tal forma que se crea un árbol de confianza basado en la lógica de la D-PKI bajo la *Blockchain Ethereum* de beame.io, que permite que el ecosistema crezca en función de las necesidades del despliegue de dispositivos *IoT* que sea necesario, pudiendo definir nuevas funcionalidades al sistema, agregar tipos de eventos e incluso aumentar la cantidad de mineros de forma adecuada para su mejor funcionamiento.

Cada dispositivo contiene su propio *proxy* que permite que las aplicaciones sean accesibles en redes privadas mediante transacciones basadas en la confianza, pudiendo conectarse una vez e intercambiar sus libros contables, de tal forma que se administra y crece por sí sola a diferencia de los sistemas centralizados, lo que dota a este despliegue de mayor flexibilidad y escalabilidad.

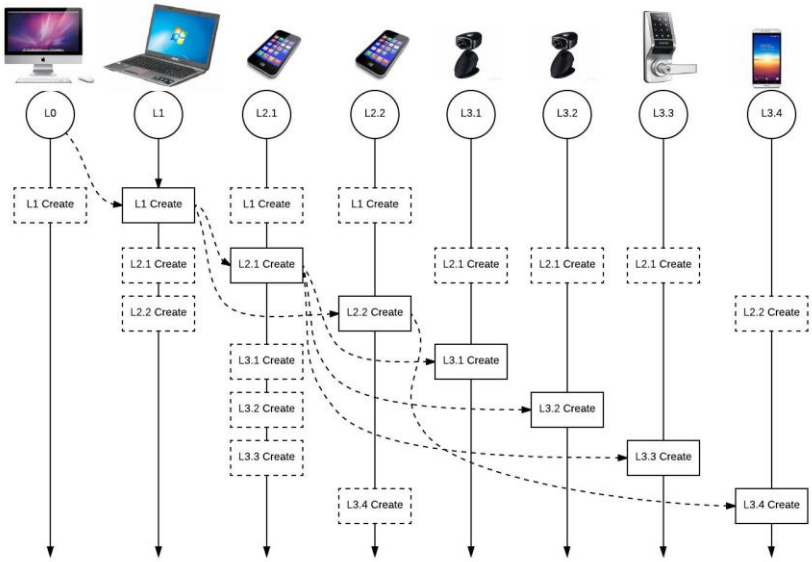

Ilustración 49. Escalabilidad de dispositivos IoT.(beame.io, 2017)

6. Conclusiones

6.1. Cumplimiento de objetivos

En esta sección vamos a analizar el cumplimiento de los objetivos específicos que se marcaron en el punto *3.2*, enumerándolos y añadiendo un breve resumen de lo aportado en cada uno de ellos.

- Conocer el marco tecnológico de la tecnología *Blockchain*.

  Se detalla el marco que rodea a la tecnología *Blockchain* a lo largo de todo el punto *2* "Estado del arte", donde se realiza un estudio teórico que abarca desde sus antecedentes hasta el estado actual, explicando su definición, arquitectura, tipologías, funcionamiento, operaciones y su relación con la computación distribuida o la criptografía, entre otros aspectos.

- Entender su aplicabilidad en la autenticación y el control de accesos.

  También en el punto *2.3* se revisaron trabajos relacionados con la aplicación de la tecnología *Blockchain* en los ámbitos de la autenticación y el control de accesos, como son la solución de *D-PKI* llamada *REMME* o el protocolo *XACML* para la implementación de control de accesos.

- Describir cómo *PKI* controla y gestiona las identidades digitales.

  En el ámbito de la gestión de identidades nos hemos centrado en el punto *2.5* en entender los conceptos de autenticación y verificación, una introducción a los certificados digitales de tipo *X.509*, cómo funciona una Infraestructura de Clave Pública *PKI,* alcanzando incluso el concepto distribuido que suponen las *D-PKI,* y una explicación conceptual de lo que son las identidades digitales descentralizadas.

- Especificar los objetivos del proyecto y una metodología basada en riesgos.

  El objetivo del proyecto una vez concluido el estudio teórico y de marco de la tecnología a aplicar, es el aseguramiento de dispositivos *IoT* mediante la implementación de una *D-PKI* bajo la *Blockchain Ethereum* pública de un proveedor

de servicios como es la empresa *beame.io*, que facilite su acceso remoto confiable mediante Internet y donde se plantea el desarrollo de dos prototipos *IoT* con diferentes niveles de seguridad al objeto de poder evaluarlos mediante el uso de la metodología de auditoria de *S.I.* en base a sus riegos detectados, es decir, una Evaluación del Riesgo *EDR*.

- Medir el riesgo de los dispositivos *IoT* en base a su configuración por defecto.

  Dentro del proceso de *EDR* sobre los dispositivos *IoT* se establecen unos indicadores de compromiso, que en base a los riesgos potenciales definidos en la lista del *OWASP Top 10 IoT 2014* y a los índices aplicados, generan un nivel de riesgo medible en ambos supuestos, que nos servirán posteriormente para su análisis y propuestas de mejora en lo que a la seguridad del dispositivo se refiere.

- Analizar y proponer mejoras en base al riesgo detectado.

  Como el mayor riesgo es el del dispositivo *IoT01,* que prácticamente está implementado con valores por defecto en la configuración y sin que la seguridad del sistema y sus comunicaciones sean un aspecto que se haya tenido como prioridad, se posiciona como el punto de partida para poder proponer mejoras para el segundo dispositivo *IoT02*, ya que obtiene un índice de riesgo muy alto. Estas mejoras se ven reflejadas en las diferentes pruebas de cumplimiento de nuestra metodología *EDR*.

- Establecer un entono de *PKI* bajo *Blockchain* para dispositivos *IoT* conectados.

  Entre las diferentes pruebas de cumplimiento y pruebas sustantivas de la metodología *EDR*, encontramos el despliegue y uso de una *D-PKI* para el aseguramiento de dispositivos *IoT,* de forma que sean accesibles de manera confiable y autenticada mediante Internet.

- Verificar y validar los resultados obtenidos.

  En la evaluación de la metodología disponemos de los análisis de seguridad, donde se puede verificar de manera cuantificada el nivel de riesgo de cada uno de los dos dispositivos, obteniendo los resultados que nos servirán para completar nuestra auditoría de seguridad en función del riesgo detectado.

- Cuantificar de nuevo el riesgo mediante métricas y mejoras obtenidas.

  Una vez obtenido el riesgo final tras la aplicación de la metodología *EDR*, podemos identificar las mejoras en términos de seguridad y gracias a las métricas obtenidas que han posibilitado que el dispositivo *IoT02* rebaje el índice de compromiso en base a los indicadores detectados, ya que es el objeto de este estudio.

- Reconocer posibles nuevas amenazas al sistema.

  Se identifican amenazas como el compromiso de certificados, la inseguridad de los certificados autofirmados o el uso de tecnologías que rompen el cifrado, como *TLS Termination* o *SSL Inspection*.

6.2. Continuidad al estudio

Como posibles aportaciones que le dieran continuidad a futuros trabajos relacionados, podrían realizarse desarrollos de pequeñas aplicaciones que una vez instaladas en los dispositivos *IoT* permitiesen evaluar de forma independiente la seguridad de los mismos, y que proporcionen nuevos indicadores de compromiso o la fiabilidad del sistema de forma automática.

También en este sentido se podría cambiar el tipo de metodología de evaluación de la seguridad y gestión del riesgo, unido al uso de guías de buenas prácticas como *MAGERIT (administracionelectronica.gob.es, 2018)*, *COBIT* (isaca.org, 2018), o *ISO 27005* (iso.org, 2018), que aportarían nuevas visiones de los escenarios de riesgo, así como de sus posibles remediaciones, permitiéndonos aumentar el conocimiento sobre esta tecnología y sus posibles implementaciones seguras.

En este trabajo se ha optado por el uso de un despliegue de *Blockchain Ethereum* del proveedor de servicios *beame.io*, pero existen otros muchos proveedores que permiten despliegues similares, cada uno con sus particularidades, pero que podrían añadir funciones o enfoques distintos, o la alternativa de crear nuestra propia cadena de bloques pública en *Ethereum*.

7. Referencias

administracionelectronica.gob.es. (2018). *MAGERIT versión 3.* Retrieved 09/17, 2018, from https://administracionelectronica.gob.es/ctt/magerit#.W6AMgRF9jcs

Ali, M., Shea, R., Nelson, J., & Freedman, M. J. (2017). *Blockstack technical whitepaper*.

Allen, C., Brock, A., Buterin, V., Callas, J., Dorje, D., Lundkvist, C., et al. (2015). *Decentralized public key infrastructure.*

Amazon Web Services. (2018). *Acerca de la federación basada en SAML 2.0.* Retrieved 09/03, 2018, from https://docs.aws.amazon.com/es_es/IAM/latest/UserGuide/id_roles_providers_saml.html

Amazon Web Services, I. (2018). *Amazon web services.* Retrieved 09/16, 2018, from https://aws.amazon.com

Back, A., Corallo, M., Dashjr, L., Friedenbach, M., Maxwell, G., Miller, A., et al. (2014). *Enabling blockchain innovations with pegged sidechains. URL: Http://www.Opensciencereview.com/papers/123/enablingblockchain-Innovations-with-Pegged-Sidechains,*

Baran, P. (1964). *On Distributed Communications: I.Introduction to Distributed Communications Networks,*

beame.io. (2017). *Beame.io BlockChain based PKI identity solutions.* Retrieved 09/01, 2018, from https://www.beame.io/

beameio.net. (2018). *Beame-gatekeeper.* Retrieved 09/01, 2018, from https://ypxf72akb6onjvrq.ohkv8odznwh5jpwm.v1.p.beameio.net/gatekeeper

BitMEX Research, & Gubatron. (2018). *Guía completa sobre consenso bajo prueba de interés (proof of stake) y entrevista a vitalik buterin.* Retrieved 09/03, 2018, from

https://www.diariobitcoin.com/index.php/2018/04/21/guia-completa-sobre-consenso-bajo-prueba-de-interes-proof-of-stake-y-entrevista-a-vitalik-buterin/

Brent, X. (2018). *Blockchain vs. distributed ledger technologies.* Retrieved 09/16, 2018, from https://media.consensys.net/blockchain-vs-distributed-ledger-technologies-1e0289a87b16

Cervigni, L. S. (2016). *El blockchain en la práctica: Una introducción simple para profesionales* British Institute For Decentralized Internet Technologies.

Chiner, E. (2017). *Análisis a fondo de blockchain: Utilidad y características.* Retrieved 06/03/2018, 2018, from https://blog.gft.com/es/2017/09/15/analisis-a-fondo-de-blockchain-utilidad-y-caracteristicas/

Civic Technologies, I. (2017). *CIVIC whitepaper*

Criptonoticias.com. (2017). *¿Qué es una bifurcación (fork) de blockchain?* Retrieved 05/27, 2018, from https://www.criptonoticias.com/informacion/que-es-bifurcacion-fork-soft-hard-blockchain/

Crosby, M., Pattanayak, P., Verma, S., & Kalyanaraman, V. (2016). *Blockchain technology: Beyond bitcoin*. *Applied Innovation, 2*, 6-10.

CTIC, C. T. (2017). *¿Qué es el "blockchain" del que todo el mundo habla?.* Retrieved 05/13, 2018, from http://www.fundacionctic.org/ctic/articulos-y-otras-publicaciones/que-es-el-blockchain-del-que-todo-el-mundo-habla?gclid=Cj0KCQjwxN_XBRCFARIsAIufy1YceIv6sTD6dRHS_Lio8d5VUOYCPCmJxH0pvWQtsVc9SPsYo3CbSrEaAqI_EALw_wcB

Dai, W. (1998). *Bmoney.* Retrieved 05/13, 2018, from http://www.weidai.com/bmoney.txt

Decentralized ID Ltd. (2018). *An organic, self-sovereign approach to blockchain-based ID management.* Retrieved 06/04, 2018, from https://decentralized.id/images/website-login.jpg

Droguett, J. (2006). *Estructura del certificado X.509.* Retrieved 07/05, 2018, from http://zeromdg.blogspot.com/2006/05/estructura-del-certificado-x509.html

Ethereum Foundation. (2018). *Ethereum project: Blockchain app platform.* Retrieved 06/04, 2018, from https://www.ethereum.org/

Faria, E. (2017). *Panorama del desarrollo de la gestión de identidades en blockchain.* Retrieved 06/04, 2018, from https://www.criptonoticias.com/aplicaciones/panorama-desarrollo-gestion-identidades-blockchain/

Fernandez, M. (2015). *Infraestructura de clave pública.* Retrieved 06/04, 2018, from https://es.slideshare.net/manuelbarcell/t04-01-pki

GlobalSign. (2016). *Beame.io chooses GlobalSign's high-scale digital certificate services to turn any machine with a web browser into a secure server.* Retrieved 09/03, 2018, from https://www.globalsign.com/en/company/news-events/news-archive/beameio-chooses-globalsign-high-scale-digital-certificate-service/

IdentityServer3. (2018). *Proof of possession - overview.* Retrieved 08/30, 2018, from https://identityserver.github.io/Documentation/docsv2/pop/overview.html

InetDaemon. (2018). *TCP 3-way handshake.* Retrieved 09/03, 2018, from https://www.inetdaemon.com/tutorials/internet/tcp/3-way_handshake.shtml

isaca.org. (2018). *What is COBIT 5?* Retrieved 09/17, 2018, from http://www.isaca.org/COBIT/Pages/default.aspx

iso.org. (2018). *iso/iec 27005:2018.* Retrieved 09/17, 2018, from https://www.iso.org/standard/75281.html

Judiciary Blockchain. (2017). *Antecedentes de blockchain en España.* Retrieved 05/13, 2018, from https://judiciaryblockchain.org/2017/08/31/antecedentes-de-blockchain-en-espana/

Kaspersky. (2013). *¿Qué es un botnet?* Retrieved 09/03, 2018, from https://www.kaspersky.es/blog/que-es-un-botnet/755/

Kehal, B. (2018). *2018 top 5 cyber security predictions: Attacks, regulations and innovation.* Retrieved 05/07, 2018, from http://www.refworks.com/refworks2/default.aspx?r=references|MainLayout::init

Knirsch, F., Unterweger, A., Karlsson, K., Engel, D., & Wicker, S. B. *Evaluation of a blockchain-based proof-of-possession implementation.*

krishnasrinivas, rabchev, lucamilanesio, jarrettgilliam, nathanleclaire, khanzf, et al. (2014). *Wetty - terminal in browser over http/https*. https://github.com/krishnasrinivas/wetty:

Lage, O., & Berrocal, J. (2017). *Blockchain, entendiendo la criptografía que cambiará tu negocio.*

Lean Manufacturing, Kaizen, Kanban, Lean, VSM & 5S. (2018). *El riesgo es un concepto importante e imprescindible considerar en la gestión lean.* Retrieved 07/05, 2018, from http://www.progressalean.com/el-riesgo-es-un-concepto-importante-e-imprescindible-considerar-en-la-gestion-lean/

Lewis, A., & Bits on Blocks. (2015). *A gentle introduction to digital tokens.* Retrieved 09/10, 2018, from https://bitsonblocks.net/2015/09/28/gentle-introduction-digital-tokens/

Maesa, D. D. F., Mori, P., & Ricci, L. (2017). *Blockchain based access control. IFIP International Conference on Distributed Applications and Interoperable Systems,* pp. 206-220.

Morrison, A., & Sinha, S. (2016). *A primer on blockchain.* Retrieved 05/27/2018, 2018, from http://usblogs.pwc.com/emerging-technology/a-primer-on-blockchain-infographic/

mozilla.org. (2018). *Firefox.* Retrieved 09/16, 2018, from https://www.mozilla.org/es-ES/firefox/

Muñoz, A. (2018). *Tecnología blockchain y ciberseguridad. una visión crítica.* Retrieved 05/13, 2018, from https://www.i4s.com/tecnologia-blockchain-y-ciberseguridad-una-vision-critica/

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.

Navarro, W. (2018). *Historia del blockchain, la solución a un problema.* Retrieved 05/13, 2018, from https://www.addalia.com/historia-del-blockchain-la-solucion-problema/

Ochoa, R. (2018). *Entendiendo la minería en blockchain.* Retrieved 09/03, 2018, from https://elcriptografo.com/2018/06/26/entendiendo-la-mineria-en-blockchain/

owasp.org. (2015). *Top 10 IoT vulnerabilities (2014).* Retrieved 09/03, 2018, from https://www.owasp.org/index.php/Top_10_IoT_Vulnerabilities_(2014)

Payatu. (2015). *IoT security – part 3 (101 – IoT top ten vulnerabilities).* Retrieved 09/03, 2018, from https://payatu.com/iot-security-part-3-101-iot-top-ten-vulnerabilities/

Preukschat, A. (2014). *Blockstream y sidechains: ¿Qué es una sidechain o cadena lateral en bitcoin?* Retrieved 07/05, 2018, from https://www.oroyfinanzas.com/2014/11/sidechain-bitcoin-blockstream-que-es-un-sidechain/

Ramiro, R. (2017). *Blockchain para la autenticación y verificación de nuestra identidad.* Retrieved 06/04, 2018, from https://ciberseguridad.blog/blockchain-para-la-autenticacion-y-verificacion-de-nuestra-identidad/

REMME CAPITAL LTD. (2018). *Distributed public key infrastructure (PKI) protocol and access management DApps report on business model overview*

Salvachúa, J., Quemada, J., & Alonso, A. (2016). *BlockChain: Una base de datos distribuida de solo escritura para mantener libros de cuentas distribuidos.*

Shaan, R. (2017). *Merkle trees.* Retrieved 07/04, 2018, from https://hackernoon.com/merkle-trees-181cb4bc30b4

Singh, A. (2017,). *Differences between blockchain and distributed databases.* Message posted to https://www.quora.com/How-is-the-blockchain-different-from-distributed-databases-in-terms-of-record-keeping

Sorrell, S. (2017). *The internet of things for security providers: Opportunities, strategies, & market leaders 2016-2021*. *Juniper,*

Soto, M. G. (2017). *La mineria de bitcoin explicada con sencillez….* Retrieved 05/26, 2018, from https://medium.com/@marvin.soto/la-mineria-de-bitcoin-explicada-con-sencillez-7aa9fe561ecc

Stoica, I., Morris, R., Liben-Nowell, D., Karger, D. R., Kaashoek, M. F., Dabek, F., et al. (2003). Chord: *A scalable peer-to-peer lookup protocol for internet applications. IEEE/ACM Transactions on Networking (TON), 11*(1), 17-32.

Suraj, K. (2017). *Merkle Trees—Introduction to blockchain.* Retrieved 07/04, 2018, from https://medium.com/@skj48817/merkle-trees-introduction-to-blockchain-c80c0247046

Tejero, A. (2017). *Metodología de análisis de riesgos para la mejora de la seguridad del internet de las cosas.*

The SelfKey Foundation. (2017). *Selfkey*

VerifyUnion Team. (2017). *VerifyUnion whitepaper*

Verime Digital Pte. Ltd. (2018). *Verime digital identification & authentication made easy*

Wikipedia. (2018). *TLS termination proxy.* Retrieved 09/03, 2018, from https://en.wikipedia.org/wiki/TLS_termination_proxy

Wikipedia, l. e. l. (2018). *Arthur andersen.* Retrieved 09/10, 2018, from https://es.wikipedia.org/wiki/Arthur_Andersen

8. Índice de ilustraciones

9. Índice de tablas

10. Anexos

10.1. Anexo I - lista de materiales *IoT01*

Assembly List

Label	Part Type	Properties
Base Shield	Base Shield V2	Variante variant 1; Tipo shield
intel Edison + Arduino board	Intel Edison Arduinobreakout	Tipo Edison Arduinobreakout
Sensor de Luz	Grove Light Sensor	Variante variant 6; tamaño 1x1; interface analog
Sensor de Temperatura	Grove Temperature Sensor	Variante variant 7; tamaño 1x1; interface analog

Shopping List

Amount	Part Type	Properties
1	Base Shield V2	Variante variant 1; Tipo shield
1	Intel Edison Arduinobreakout	Tipo Edison Arduinobreakout
1	Grove Light Sensor	Variante variant 6; tamaño 1x1; interface analog
1	Grove Temperature Sensor	Variante variant 7; tamaño 1x1; interface analog

Exported with Fritzing 0.9.3- http://fritzing.org

10.2. Anexo II - Lista de materiales *IoT02*

Assembly List

Label	Part Type	Properties
DHT22 Temperatura / Humedad	Humidity and Temperature Sensor RHT03	power supply 3.3-5.5V DC; sensing element Polymer humidity capacitor; output signal Digital Signal
Raspberry Pi 1	Raspberry Pi 3	Procesador Broadcom BCM2837 64-bit ARMv8; Variante Raspberry Pi 3; revision RPI-3-V1.2; Número

Shopping List

Amount	Part Type	Properties
1	Humidity and Temperature Sensor RHT03	power supply 3.3-5.5V DC; sensing element Polymer humidity capacitor; output signal Digital Signal
1	Raspberry Pi 3	Procesador Broadcom BCM2837 64-bit ARMv8; Variante Raspberry Pi 3; revision RPI-3-V1.2; Número de componente RPI-3-V1.2

Exported with Fritzing 0.9.3- http://fritzing.org

10.3. Anexo III - Código *Python IoT01*

led.py

```
#!/usr/bin/env python

import mraa # importacion de modulos necesarios
import time
import math
import datetime
import os

B=3975 # valor del termistor del sensor de temp. para covertirlo desde el dato analogico
# variables asociadas a las posiciones de los sensores en la placa
ain = mraa.Aio(3)
light = mraa.Aio(0)

os.system('clear')
print
print ('TFM - Fco. Javier Balmaseda - UNIR 2018')
print
print ('DISPOSITIVO IoT 1 VULNERABLE: ')
os.system('uname -a') # identificacion del S.O. de la placa
print
print ('IP PUBLICA: ')
os.system ('curl icanhazip.com') # captura de la IP publica del dispositivo
print
print ('CONEXIONES HTTP ESTABLECIDAS: ')
print
os.system ('netstat -eW | grep ESTABLISHED') # captura de las conexiones HTTP establecidas
print
print ('CONEXIONES TCP ESTABLECIDAS: ')
print
os.system ('netstat -n | grep ESTABLISHED') # captura de las conexiones TCP establecidas
print
print ('LECTURA DE SENSORES: ')
print

X=0 # inicializacion de variable para bucle principal
# calibracion de los sensores
a=ain.read()
l=light.read()
time.sleep(1)
a=ain.read()
l=light.read()
a=ain.read()
l=light.read()
time.sleep(4)

while X<5: # bucle principal de 5 iteraciones
        # captura de la marca de tiempo del S.O.
        ts = time.time()
        st = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d %H:%M:%S')
        a = ain.read() # captura en bruto del sensor de temperatura
        time.sleep(1)
        l = light.read() # captura del sensor de luz
        time.sleep(1)
        # calcula la resistencia de corriente del termistor en función del valor del sensor de temp.
        resistencia = float((1023-a)*10000/a)
        # calcula la temperatura basada en el valor de la resistencia
        temp = float(1/(math.log(resistencia/10000)/B+1/298.15)-273.15)
        X = X+1 # avance de la variable del bucle principal
        # si existen datos de captura muestra la informacion
        if a is not None and l is not None:
                print "TimeStamp = ",st," /  Temperatura = ", "{0:.1f}*".format(temp)," / Luz = ",l
                time.sleep(2)
        # si faltan datos de captura muestra el error
        else:
                print st, ('Fallo al leer sensores, espere...')
                time.sleep(10)

time.sleep(5)
print
```

10.4. Anexo IV – Código *Python IoT02*

sensores.py

```python
#!/usr/bin/python

import datetime # importacion de modulos necesarios
import time
import Adafruit_DHT
import os

os.system('clear')
print
print ('TFM - Fco. Javier Balmaseda - UNIR 2018')
print
print ('DISPOSITIVO IoT 2 ASEGURADO CON BlockChain + PKI: ')
os.system('uname -a') # identificacion del S.O. de la placa
print
print ('IP Publica: ')
os.system('curl icanhazip.com') # captura de la IP publica del dispositivo
print
print('CONEXIONES HTTP(S)  ESTABLECIDAS:')
print
os.system('netstat -eW | grep ESTABLISHED | grep http') # captura de las conexiones HTTP establecidas
print
print('CONEXIONES TCP  ESTABLECIDAS:')
print
os.system('netstat -n | grep ESTABLISHED | grep 443') # captura de las conexiones TCP establecidas por el puerto 443
print
print('LECTURA DE SENSORES:')
print
X=0 # inicializacion de variable para bucle principal

try:
        # bucle principal de 5 iteraciones
        while X<5:
                # captura de la marca de tiempo del S.O.
                ts = time.time()
                st = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d %H:%M:%S')
                # captura de los ddatos de los sensores de humedad y temperatura
                humidity, temperature = Adafruit_DHT.read_retry(Adafruit_DHT.DHT22, 4)
                X = X+1 # avance de la variable del bucle principal
                # si existen datos de captura muestra la informacion
                if humidity is not None and temperature is not None:
                        print ('TimeStamp ='),st,(' /  Temperatura = {0:0.1f}*  /  Humedad = {1:0.1f}%'.format(temperature, humidity))
                        time.sleep(5)
                # si faltan datos de captura muestra el error
                else:
                        print st, ('Fallo al leer sensores, espere...')
                        time.sleep(10)
# captura de excepciones
except Exception,e:
        print str(e)
print
```

10.5. Anexo V – Cuestionario de seguridad *IoT01*

OWASP IoT Top 10 - 2014	Pregunta	SÍ / NO
R01: Interfaz Web insegura	¿Se ha modificado el acceso a los dispositivos con los usuarios y contraseñas por defecto?	SÍ
R02: Autenticación / Autorización insuficientes	¿Los usuarios acceden al dispositivo mediante el uso de identidades verificadas con IDP, SP y MFA?	NO
R03: Servicios de red inseguros	¿Están sólo expuestos a Internet los puertos TCP/UDP necesarios para su funcionamiento?	NO
R04: Falta de Cifrado de transporte	¿Está la comunicación cifrada mediante certificados X.509 con claves RSA de 2048 bits creadas en el dispositivo y los tokens de autorización firmados que se envían a la PKI junto con la clave pública firmada?	NO
R05: Privacidad	¿Están los datos de acceso al dispositivo desanonimizados o son FQDN generados aleatoriamente?	NO
R06: Interfaz Cloud insegura	¿Está el intercambio de datos de autenticación y autorización de los dispositivos con los proveedores de servicios Cloud protegidos con estándares seguros como SAML?	NO
R07: Interfaz Móvil insegura	¿Está la conexión inalámbrica del dispositivo protegida con protocolos seguros de conexión como WPA2?	NO
R08: Configuración de Seguridad insuficiente	¿Se impide ejecutar scripts con los usuarios no administradores / root?	NO
R09: Software / Firmware inseguro	¿Existe un servicio de soporte activo de actualizaciones del firmware del dispositivo?	NO
R10: Pobre seguridad física	¿Está el dispositivo físicamente ubicado en un lugar que no es de libre acceso?	SÍ

10.6. Anexo VI - Cuestionario de seguridad *IoT02*

OWASP IoT Top 10 - 2014	Pregunta	SÍ / NO
R01: Interfaz Web insegura	¿Se ha modificado el acceso a los dispositivos con los usuarios y contraseñas por defecto?	SÍ
R02: Autenticación / Autorización insuficientes	¿Los usuarios acceden al dispositivo mediante el uso de identidades verificadas con IDP, SP y MFA?	SÍ
R03: Servicios de red inseguros	¿Están sólo expuestos a Internet los puertos TCP/UDP necesarios para su funcionamiento?	SÍ
R04: Falta de Cifrado de transporte	¿Está la comunicación cifrada mediante certificados X.509 con claves RSA de 2048 bits creadas en el dispositivo y los tokens de autorización firmados que se envían a la PKI junto con la clave pública firmada?	SÍ
R05: Privacidad	¿Están los datos de acceso al dispositivo desanonimizados o son FQDN generados aleatoriamente?	SÍ
R06: Interfaz Cloud insegura	¿Está el intercambio de datos de autenticación y autorización de los dispositivos con los proveedores de servicios Cloud protegidos con estándares seguros como SAML?	SÍ
R07: Interfaz Móvil insegura	¿Está la conexión inalámbrica del dispositivo protegida con protocolos seguros de conexión como WPA2?	SÍ
R08: Configuración de Seguridad insuficiente	¿Se impide ejecutar scripts con los usuarios no administradores / root?	SÍ
R09: Software / Firmware inseguro	¿Existe un servicio de soporte activo de actualizaciones del firmware del dispositivo?	SÍ
R10: Pobre seguridad física	¿Está el dispositivo físicamente ubicado en un lugar que no es de libre acceso?	SÍ