UNIVERSIDAD TÉCNICA DEL NORTE

INSTITUTO DE POSTGRADO

MAESTRÍA EN TELECOMUNICACIONES

Proyecto del Trabajo de Titulación previo a la obtención del Título de Magíster en Telecomunicaciones.

TEMA:

"***Cifrado de datos usando Cadena de Bloques (BlockChain) como tecnología de convergencia para dispositivos móviles asociados con IoT (Internet of Things), en la capa de aplicación del modelo de capas IoT***"

Línea de Investigación: Innovación tecnológica y productos de telecomunicación.

Investigador: VERÓNICA VIVIANA BURGOS YAR
Tutor: FREDDY MAURICIO TAPIA LEON

IBARRA - ECUADOR

Julio 2021

APROBACIÓN DEL TUTOR

Yo, Freddy Mauricio Tapia León, en mi calidad de tutor del Trabajo de Grado: "***Cifrado de datos usando Cadena de Bloques (BlockChain) como tecnología de convergencia para dispositivos móviles asociados con IoT (Internet of Things), en la capa de aplicación del modelo de capas IoT***", presentado por la maestrante: Verónica Viviana Burgos Yar, para otorgar el grado de Magíster en Telecomunicaciones, doy fe: de que dicho trabajo reúne los requisitos y méritos suficientes para ser sometido a la presentación (pública o privada) y evaluación por parte de un jurado examinador que se designe.

Firmado electrónicamente por:
FREDDY
MAURICIO
TAPIA LEON

MSc. Freddy Mauricio Tapia León
Cédula: 1714745690

APROBACIÓN DEL ASESOR

Yo, José Fernando Garrido Sánchez, en mi calidad de asesor del Trabajo de Grado: "***Cifrado de datos usando Cadena de Bloques (BlockChain) como tecnología de convergencia para dispositivos móviles asociados con IoT (Internet of Things), en la capa de aplicación del modelo de capas IoT***", presentado por la maestrante: Verónica Viviana Burgos Yar, para otorgar el grado de Magíster en Telecomunicaciones, doy fe: de que dicho trabajo reúne los requisitos y méritos suficientes para ser sometido a la presentación (pública o privada) y evaluación por parte de un jurado examinador que se designe.

Fernando Garrido S.

Firmado digitalmente por Fernando Garrido S.
Fecha: 2021.08.11 17:09:14 -05'00'

MSc. José Fernando Garrido Sánchez
Cédula: 1707852081

UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1002862207		
APELLIDOS Y NOMBRES:	VERONICA VIVIANA BURGOS YAR		
DIRECCIÓN:	QUITO, LA ARMENIA SEBASTIÁN DE BENALCAZAR Y PEDRO FERMÍN.		
EMAIL:	vivianaburgos@live.com vburgos@utn.edu.ec		
TELÉFONO FIJO:		**TELÉFONO MÓVIL:**	0994177643

DATOS DE LA OBRA	
TÍTULO:	"Cifrado de datos usando Cadena de Bloques (BlockChain) como tecnología de convergencia para dispositivos móviles asociados con IoT (Internet of Things), en la capa de aplicación del modelo de capas IoT"
AUTOR (ES):	Verónica Viviana Burgos Yar
FECHA: DD/MM/AAAA	15/07/2021
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	☐ **PREGRADO** ■ **POSGRADO**
TITULO POR EL QUE OPTA:	**MAESTRIA EN TELECOMUNICACIONES**
ASESOR /DIRECTOR:	**Tutor:** MSc. Freddy Mauricio Tapia León **Asesor**: MSc. José Fernando Garrido Sánchez

2. CONSTANCIAS

El autor (es) manifiesta (n) que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es (son) el (los) titular (es) de los derechos patrimoniales, por lo que asume (n) la responsabilidad sobre el contenido de la misma y saldrá (n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Quito, a los 12 días del mes de agosto de 2021

EL AUTOR:

(Firma): Firmado electrónicamente por: VERONICA VIVIANA BURGOS YAR

Nombre: VERONICA VIVIANA BURGOS YAR

DEDICATORIA

El presente trabajo investigativo lo dedico principalmente a Dios, por ser mi inspirador y darme la fuerza para continuar en este proceso y alcanzar esta meta.

A mis hijos por haber sido ese motor que me impulsa a seguir luchando por nuevos objetivos

Verónica Viviana

AGRADECIMIENTO

Quiero expresar mi gratitud a Dios, quien con su bendición llena mi vida y la de mi familia, por ser mi inspirador, amigo y guía.

Mi agradecimiento a la Universidad Técnica del Norte, al Instituto de Posgrado, a la Maestría en Telecomunicaciones y en especial a todos mis docentes quienes con su conocimiento hicieron que crezca profesionalmente.

Finalmente quiero expresar mi mayor agradecimiento al MSc. Freddy Mauricio Tapia León, mi tutor de investigación quien durante todo este proceso me guio y apoyó constantemente con su dirección, conocimiento y enseñanza para conseguir el cabal desarrollo de este trabajo de investigación.

CONSAGRACIÓN

Este trabajado de investigación quiero consagrarlo al "Sagrado Corazón de Jesús y al Dulce Corazón de María" quienes han sido mi fortaleza durante todo este tiempo de estudio.

Quiero ofrecer mi cansancio, mis incertidumbres, mi desanimo, mis tristezas, mis metas alcanzadas, todo lo que soy y todo lo que tengo, mi pequeñez y mi nada a mi Amado Jesús y esconderlas en su Sacratísimo Corazón con la finalidad de aliviar el peso de Cruz.

A mi santísima Madre la Virgen María, Reina de la paz, quiero ofrecerle mis dones para que Ella, la llena de gracias los perfeccione y sea quien interceda por mi para que pueda culminar con éxito esta meta que me ha planteado.

Jesús y María dadme su santa protección y bendición para alejar todo espíritu de maldad que ponga en mi: pensamientos de desánimo, tentación, tristeza y dolor. Por lo contrario, iluminen mi pensamiento y permitan que vaya encontrando el camino correcto para finalizar mi trabajo de investigación, ruego a su amor y a su protección para que mi mente y mis dones se vean iluminados notablemente para que pueda ofrecerles un trabajo digno de ustedes.

Amado Jesús tú que conocer el presente, pasado y futuro, Tú que eres dueño del tiempo te ruego que permitas alargar mi tiempo, y permitas que mis manos escriban palabras llenas de sabiduría, conocimiento y discernimiento.

"Yo Verónica Viviana Burgos Yar, me doy y consagro toda entera al Sagrado Corazón de Nuestro Señor Jesucristo y al Dulce Corazón de María, con ello: mi persona, mi vida, oraciones, penas y sufrimientos, mis metas, especialmente mi proyecto de investigación: *Desarrollo de un marco metodológico para cifrado de datos usando Cadena de Bloques (BlockChain) como tecnología de convergencia para dispositivos móviles asociados con IoT (Internet of Things), en la capa de aplicación del modelo de capas IoT*", para no querer servirme de ninguna parte de mi ser sino para honrar, amar y glorificar a mi Señor y a su Santísima Madre. Es mi voluntad irrevocable ser toda de Ellos y hacer todo por su amor, renunciando de todo corazón a todo lo que pueda disgustarles.

Yo os tomo, pues, Oh Sagrados Corazones, por el único objeto de mi amor, los protectores de mi vida, la seguridad de mi salvación, el remedio de mi fragilidad y de mi inconstancia, los reparador de todos los defectos de mi vida, y mi asilo en la hora de mi muerte.

Jesús, ¡Corazón de bondad! sed mi justificación para con Dios vuestro Padre, y alejad de mí, los rayos de su justa cólera. ¡Oh Corazón de amor! yo pongo toda mi confianza en vos, pues todo lo temo de mi malicia y de mi debilidad, pero todo espero de vuestra bondad. ¡Extinguid pues en mí todo lo que os pueda desagradar o resistir! Que vuestro puro amor os imprima con tanta fuerza en mi corazón que no pueda jamás olvidaros, ni estar separada de vos, a quien conjuro, por todas vuestras bondades, que mi nombre sea escrito en vos, pues yo quiero hacer construir mi gloria en vivir y morir en calidad de esclava vuestra. Amen".

ÍNDICE DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

En los últimos años, los dispositivos inteligentes han tomado un espació importante en el desarrollo tecnológico, están presentes en todos los hogares y al menos 8 de cada 10 personas tienen un dispositivo móvil. Estos dispositivos recopilan toda clase de datos como: hábitos, rutinas, búsquedas, música, videos favoritos, claves, entre otros elementos personales, ocasionando que todos esos datos se encuentren expuestos a cualquier ataque o robo debido a su ligereza y falta de medidas de seguridad, exponiendo los datos personales de sus usuarios.

A pesar de que el Internet de las Cosas (en inglés, Internet of Things - IoT)se ha convertido en una de las tecnologías más populares para varios fabricantes como (CISCO, IBM, Ge, Google, Microsoft, Salesforce, Oracle, entre otros), debido a su facilidad en la implementación, también se ha convertido en un reto en cuando a la seguridad y privacidad de los datos, pues su sencillez y falta de madurez deja varios elementos expuesto en cada una de sus capas, por lo que se deben investigar nuevos mecanismos de cifrado de datos para proteger los datos transmitidos y almacenados.

BlockChain se perfila como una de las tecnologías compatible para IoT con la finalidad de establecer un cifrado altamente seguro, fuerte, elástico, escalable y descentralizada, que permite asegura datos de cualquier tipo, gracias a contratos inteligentes (Chaincode) que diseñan la lógica de negocio y permiten proteger la información enviada a la BlockChain.

Por lo antes mencionado esta investigación propone combinar las dos tecnologías (IoT y BlockChain) como tecnologías convergentes para atenuar y eliminar la inseguridad y así mejora la privacidad de los datos transmitidos mediante un dispositivo IoT en la capa de aplicación en un dispositivo móvil.

Con la finalidad de evaluar si estas dos tecnologías tan diferentes y a la vez tan nuevas pueden unirse en una solución que permita proteger los datos y robustecer la escasa seguridad de un dispositivo IoT en su capa de aplicación, se implementó una BlockChain privada con Hyperledger Fabric con tres organizaciones miembros los mismos que son virtualizados con Docker Compose, adicionalmente se diseñó una API Rest IoT que permite interactuar con seis contratos inteligentes que permitieron crear la lógica de negocio para la inserción y recuperación de datos leídos por varios sensores (temperatura, humo, ventana y movimiento) que se ensamblaron en una placa LoLin new NodeMCU V3, que envía sus lecturas a un servidor Mosquito MQTT, los cuales son recuperados y transaccionados desde un módulo Python hacia la API Rest IoT. Posterior a ello se diseñó un Front - End en WordPress que muestra los datos leídos desde la BlockChain por medio de la API Rest IoT.

Una vez concluido el experimento se ejecutaron dos casos de prueba los que permitieron evidenciar que la red BlockChain no permite la inserción de datos si no se lo realiza desde la API Rest IoT que interactúa con los contratos inteligentes, hacia la red BlockChain. Por lo tanto, se concluye que esta tecnología se convierte en una alternativa para la protección de los datos transmitidos por los dispositivos IoT en la capa de aplicación.

ABSTRACT

In recent years, smart devices have taken an important place in technological development, they are present in every home and at least 8 out of 10 people have a mobile device. These devices collect all kinds of data such as: habits, routines, searches, music, favorite videos, passwords, among other personal elements, causing that all these data are exposed to any attack or theft due to its lightness and lack of security measures, exposing the personal data of its users.

Although the Internet of Things (IoT) has become one of the most popular technologies for several manufacturers (CISCO, IBM, Ge, Google, Microsoft, Salesforce, Oracle, among others), due to its ease of implementation, it has also become a challenge in terms of security and data privacy, because its simplicity and lack of maturity leaves several elements exposed in each of its layers, so new data encryption mechanisms must be investigated to protect the transmitted and stored data.

BlockChain is emerging as one of the compatible technologies for IoT in order to establish a highly secure, strong, elastic, scalable and decentralized encryption, which allows securing data of any type, thanks to smart contracts (Chaincode) that design the business logic and protect the information sent to the BlockChain.

For the above mentioned this research proposes to combine the two technologies (IoT and BlockChain) as converging technologies to mitigate and eliminate insecurity and thus improves the privacy of data transmitted through an IoT device at the application layer in a mobile device.

In order to evaluate whether these two very different and at the same time very new technologies can be united in a solution to protect data and strengthen the poor security of an IoT device in its application layer, a private BlockChain with Hyperledger Fabric was implemented with three member organizations that are virtualized with Docker Compose, Additionally, a Rest IoT API was designed to interact with six smart contracts that allowed creating the business logic for the insertion and retrieval of data read by several sensors (temperature, smoke, window and movement) that were assembled on a LoLin new NodeMCU V3 board, which sends its readings to a Mosquito MQTT server, which are retrieved and transacted from a Python module to the Rest IoT API. After that, a Front-End was designed in WordPress that shows the data read from the BlockChain through the Rest IoT API.

Once the experiment was completed, two test cases were executed, which showed that the BlockChain network does not allow the insertion of data if it is not done from the Rest IoT API that interacts with smart contracts, to the BlockChain network. Therefore, it is concluded that this technology becomes an alternative for the protection of data transmitted by IoT devices at the application layer.

CAPÍTULO I: EL PROBLEMA

TEMA:

"Cifrado de datos usando Cadena de Bloques como tecnología de convergencia para dispositivos móviles asociados con Internet de las Cosas, en la capa de aplicación del modelo de capas IoT".

Problema de investigación

El Internet de las Cosas (en inglés, Internet of Things - IoT) gracias a su ligereza, fácil implementación, bajo consumo de recurso y baja inversión se ha convertido en una tecnología líder, donde varios fabricantes como: CISCO, IBM, Ge, Google, Microsoft, Salesforce, Oracle, entre otros, han apostado por incursionar en esta tecnología con la finalidad de potenciar la conectividad entre dispositivos, sensores y redes basadas en protocolo IP. Sin embargo, este entorno hiperconectado se ve opacado por varios problemas de seguridad y privacidad de datos (Karen Rose, Scott Eldridge, 2015).

IoT tiene un cifrado de información bastante débil, sus recursos (memoria, procesador, conectividad) están principalmente enfocados a la disponibilidad de aplicaciones, no en la protección de información y canales de conexión; los dispositivos IoT almacenan información sensible, por ejemplo: a través de un dispositivo inteligente se puede instalar aplicaciones que permiten realizar transacciones financieras, comerciales, y prestación de servicios. Los proveedores y fabricantes de estas tecnologías no han hecho esfuerzos suficientes por desarrollar protocolos y arquitecturas que mejoren la seguridad, la penetración de la tecnología IoT y los ingresos económicos, superan las necesidades de desarrollar soluciones que mejoren la seguridad de los dispositivos y la protección de los datos (Gélvez Rodríguez & Santos Jaimes, 2020).

Una tecnología que está revolucionando la seguridad y el cifrado de datos es la Cadena de Bloques también conocida como BlockChain (BC), esta tecnología garantiza una encriptación de datos mucho más eficiente que otros algoritmos de seguridad. BlockChain es una base de datos descentralizada, repartida en diferentes computadores conectados entre sí a través de un algoritmo matemático, estos computadores se los conoce como nodos; esta red está protegida criptográficamente y organizada en bloques. Este sistema permite que los datos pasen a cada uno de los nodos y sean validados para obtener un dato seguro (Carlos Kuchkovsky, Gonzalo Gomez, Lardies, Daniel Diéz Garcia, 2017).

El uso de BlockChain beneficia considerablemente las aplicaciones IoT, debido a que éstas dependen del modelo cliente – servidor para funcionar; en esencia, esto significa que todos los dispositivos están conectados a una autoridad central[1]. El modelo BlockChain evitará que ocurran ataques en un porcentaje elevado a su autoridad central también conocida como servidor; dado que una red de datos descentralizada eliminaría cualquier punto de debilidad, los atacantes tendrían que apuntar a nodos individuales en la red (Alladi Tejasvi et al., 2019; Fiorentino et al., 2020).

Los dispositivos IoT se están convirtiendo en una tecnología omnipresentes (Germán et al., 2019), esto genera varias preocupaciones de seguridad, debido a su capacidad de recopilación y procesamiento de datos relacionados con los usuarios. Estos datos en muchas ocasiones son sometidos a Inteligencia Artificial (IA) y a otro tipo de procesos que permiten conocer tendencias y comportamientos, causando que la infraestructura informática se haga más débil y existan vacíos en la seguridad de los datos. Es aquí donde la tecnología de BlockChain gracias a sus características se convierta en una solución a varios problemas de seguridad de los sistemas IoT, al ser una tecnología distribuida evita tener un punto común de falla. Los registros de transacciones que contienen datos de red no son variables y se pueden encontrar en el historial anterior de la red BlockChain, lo que hace ganar la confianza del público en la red de sistemas IoT. Esta confianza juega un papel importante en la ejecución de transacciones que son de naturaleza financiera y pública (Mahapatra, 2020).

Formulación del problema

IoT y BlockChain son dos tecnologías compatibles para generar una metodología de cifrado de datos

IoT opera un modelo centralizado de cliente - servidor que requiere un administrador para gestionar la red, esto es el punto débil cuando se trata de seguridad de IoT, los dispositivos IoT confían en este administrador para determinar cómo se comportan. Si hay una violación de seguridad, la información que envían los dispositivos inteligentes queda en gran medida a disposición de los piratas informáticos (Fabiano, 2017).

BlockChain es una red distribuida que trabaja en consenso con otros computadores para verificar si un dato es seguro o no. Gracias a esta naturaleza BlockChain negara cualquier ataque a un

[1] **Autoridad Central:** Servidor de aplicaciones o base de datos que autentica el acceso de un usuario.

servidor principal, debido a que por su naturaleza distribuida los atacantes (piratas informáticos, hackers) tendrían que comprometer a todos los miembros de la red BlockChain de forma individuales para tratar de obtener los datos que desean, de tal manera que, si un miembro detecta algún parámetro anormal, la red BlockChain puede aislar (no se inserta en el Ledger) una transacción o petición para evitar el acceso a datos confidenciales (Fiorentino et al., 2020).

El análisis de esta problemática lleva a plantearnos la siguiente interrogante de investigación:

¿Es posible implementar el cifrado de datos con BlockChain como tecnología de convergencia en las aplicaciones IoT por medio de dispositivos móviles, protegiendo la capa de aplicación del modelo de capas IoT?.

Justificación de la investigación.

La evolución de la tecnología de IoT y BlockChain permitirá cambios inimaginables en las industrias de TIC[2], los dispositivos IoT permiten que la mayoría de las cosas se conecten entre sí, muchos de ellos ponen en riesgo datos críticos a posibles violaciones de seguridad, BlockChain se perfila como una tecnología para dar solución a los diferentes problemas de seguridad en este tipo de dispositivos (Abbas & Sung-Bong, 2019; Fabiano, 2017; Homayoun et al., 2019; Samaniego & Deters, 2017).

En un estudio realizado por la firma Juniper Research en el 2016, "The Internet of Things for Security Providers: Opportunities, Strategies, & Market Leaders 2016-2021", menciona que para este año 2021 quince millones de dispositivos IoT enviarán y recibirán datos a través del Internet, incrementando el desarrollo de la tecnología en un ciento veinte (120) por ciento respecto al 2016 donde la tecnología apenas se encontraba en auge (Francisco Javier Balmaseda Aranda, 2018; Steffen Sorrell, 2017).

Kaspersky en el 2018 muestra una estadística de los vectores más populares de los ataques en tecnología IoT, la contaminación de los dispositivos está basada en conseguir la contraseña de Telnet y a partir de esta acceder a datos sensibles, alojados en la memoria principal del dispositivo inteligente, en el segundo trimestre de 2018, el número de este tipo de ataques se coloca de la siguiente manera: Telnet: 75,40%, SSH: 11,59% servicio de ataques, otros: 13,01%. (Kaspersky, 2018)

[2] **TIC:** Tecnologías de la Información y la Comunicación

BlockChain pretende romper la arquitectura tradicional de las redes centralizadas, donde los dispositivos con tecnología IoT dependen de un servidor principal que procesa sus transacciones y autentica a los dispositivos, si el servidor central de alguna manera es atacado y vulnerado todo lo procesado y administrado aquí corre peligro. BlockChain al emplear una arquitectura distribuida, formada por varios nodos se transforma en un entorno más confiable, en la que los dispositivos inteligentes se conectar de forma seguro, por medio de una API que valida su identidad evitando amenazas por medio de la falsificación de dispositivos y la substracción de información. (Rocha Robson, 2020)

Basados en el estudio[3] de la firma Juniper en los próximos años se recopilará gran cantidad de información por medio de los dispositivos IoT causando una híper conectividad y mucha información navegando libremente por la red sin ninguna protección, esto hace pensar: ¿cómo se debe proteger la información disponibles en la red y en los dispositivos IoT?, este proyecto de investigación realizará un diseño metodológico que ayude a proteger a las diferentes aplicaciones de IoT ejecutadas desde un dispositivo móvil basadas en la arquitectura de BlockChain con la finalidad para garantizar la seguridad, integridad y confidencialidad, evaluar si la BlockChain se ajusta mejor al entorno IoT, validar sus niveles de seguridad respecto a los sistemas de seguridad tradicionales.

Este proyecto de investigación intenta evaluar cualitativamente las técnicas de seguridad y cifrado de datos utilizadas en tecnología IoT versus BlockChain para desarrollar aplicaciones y soluciones que permitan disminuir la fuga de información, asegurar los datos y proteger la privacidad de las personas y cosas, con la finalidad de aumentar la confianza de los consumidores en los dispositivos IoT móviles y ofrecer una alternativa que solucione las dificultades a las que se enfrenta.

Objetivos de la investigación

Objetivo general

Desarrollar un cifrado de datos usando BlockChain como tecnología de convergencia para dispositivos móviles asociados con IoT, en la capa de aplicación del modelo de capas de IoT.

[3] **Estudio Juniper:** "The Internet of Things for Security Providers: Opportunities, Strategies, & Market Leaders 2016-2021"

Objetivos específicos

- Estudiar y analizar las características, estándares y protocolos de la estructura de BlockChain para aplicaciones IoT.
- Determinar los niveles de seguridad de BlockChain para aplicaciones IoT, por medio de una matriz comparativa entre la implementación de BlockChain con soluciones gratuitas vs soluciones comerciales.
- Realizar una evaluación cualitativa de las técnicas de seguridad y cifrado de datos utilizados en tecnologías IoT versus BlockChain.
- Desarrollar un prototipo base, para validar los niveles de seguridad en un BlockChain a través de una aplicación móvil.
- Medir y validar los niveles de seguridad en IoT aplicados a los dispositivos móviles: comparando los sistemas de seguridad predefinidos versus BlockChain, a través del uso del prototipo base y encuestas aplicadas a diferentes consumidores.

Hipótesis y preguntas directrices

¿Es posible implementar el cifrado de datos con BlockChain como tecnología de convergencia en las aplicaciones IoT por medio de dispositivos móviles, protegiendo la capa de aplicación del modelo de capas IoT?

Preguntas directrices

- ¿Qué factores influyen en la seguridad de los dispositivos IoT?
- ¿Cuáles son las capas vulnerables de un dispositivo IoT?
- ¿Existen investigaciones o trabajos relacionados con la implementación de un cifrado de datos entre BlockChain e IoT?
- ¿Es posible implementar una red BlockChain para cifrar datos en una aplicación IoT?

Variables e indicadores

Independiente: Cifrado de datos
Dependiente: Estándares Protocolos

Independiente: Cifrado de datos				
Conceptualización	**Dimensiones**	**Indicadores**	**Ítems básicos**	**Técnica o Instrumento**
El cifrado de la información utiliza algoritmos matemáticos para la codificación y decodificación de los mensajes. Entre las características del cifrado de la información se puede señalar el control de acceso.(QUEMBA MARTINEZ, 2020)	Cifrado por hardware	Dispositivos físicos	¿Es posible incorporar o mejorar los niveles de cifrado de datos a nivel de hardware en un dispositivo IoT?	Análisis de documentos
	Cifrado por Software	Algoritmos y Software	¿Es posible incorporar cifrado de datos como tecnología de convergencia por medio de BlockChain hacia dispositivos IoT?	Análisis de documentos

Dependiente: Estándares Protocolos				
Conceptualización	**Dimensiones**	**Indicadores**	**Ítems básicos**	**Técnica o Instrumento**
Estándar: Un algoritmo criptográfico diseñado para cifrar y descifrar datos utilizando un determinado número de bloques de Bytes. **Protocolos: Un protocolo describe la forma en que un estándar debe comportarse en el momento de encriptar la información.** **Los protocolos se los usa ampliamente para transporte de datos a nivel de aplicación, mientras que un estándar nos ayuda a definir como cifrar los datos.** (Carle, 2003)	Estándares de cifrados	Tipos de estándares de cifrado.	Es posible mejorar los niveles de seguridad de redes IoT por medio de estándares asociados al cifrado de los datos	Análisis de documentos
	Protocolos para cifrado de Datos	Tipos de Protocolo	¿Es posible mejorar los niveles de seguridad de redes IoT por medio de protocolos asociados al cifrado de los datos?	Análisis de documentos

Independiente: Tecnologías de Convergencia
Dependiente: Tendencias y Optimización de resultados.

Independiente: Tecnologías de Convergencia				
Conceptualización	**Dimensiones**	**Indicadores**	**Ítems básicos**	**Técnica o Instrumento**
La convergencia tecnológica es la tendencia de diferentes sistemas tecnológicos en la evolución hacia la realización de tareas similares. **La supone la homogeneización de los soportes, productos,** **lógicas de emisión y consumo de las industrias info - comunicacionales.**	BlockChain	Tecnología BlockChain para IoT.	¿Es posible que BlockChain se convierta en una tecnología de convergencia para IoT?	Análisis de documentos
	IoT	Aseguramiento de los dispositivos IoT, con BlockChain.	¿Es posible asegurar los datos que transmite un dispositivo IoT, Usando BlockChain como tecnología de convergencia para el cifrado de datos?	Análisis de documentos

Dependiente: Tendencias y Optimización de resultados				
Conceptualización	**Dimensiones**	**Indicadores**	**Ítems básicos**	**Técnica o Instrumento**
Las nuevas tecnologías que están surgiendo en el mundo como IoT, BlockChain, inteligencia artificial, entre otras se perfilan para ser las más implementadas a nivel mundial por los negocios y las industrias, eso con la finalidad de mejorar su desempeño, competitividad y mejorar sus resultados en los sistemas de TI. Estas tecnologías pueden aumentar la eficiencia y mejorar la agilidad operativa de los negocios, ayudan a reducir errores y costos. (enRed, 2019)	BlockChain	BlockChain	¿Es posible que BlockChain se convierta en una tecnología que permita mejorar, optimizar resultados en el cifrado de datos?	Análisis de documentos
	IoT	Aseguramiento de los dispositivos IoT, con BlockChain.	¿IoT es una tecnología de tendencia que podría mejor y optimizar la transmisión de sus datos mediante el uso de BlockChain?	Análisis de documentos

Independiente: Seguridad de la Información
Dependiente: Niveles y técnicas de seguridad

Independiente: Seguridad de la Información				
Conceptualización	**Dimensiones**	**Indicadores**	**Ítems básicos**	**Técnica o Instrumento**
La convergencia tecnológica es la tendencia de diferentes sistemas tecnológicos en la evolución hacia la realización de tareas similares. **La supone la homogeneización de los soportes, productos,** **lógicas de emisión y consumo de las industrias info - comunicacionales.**	BlockChain	Tecnología BlockChain para IoT.	¿Es posible que BlockChain se convierta en una tecnología de convergencia para IoT?	Análisis de documentos
	IoT	Aseguramiento de los dispositivos IoT, con BlockChain.	¿Es posible asegurar los datos que transmite un dispositivo IoT, Utilizando BlockChain como tecnología de convergencia para el cifrado de datos?	Análisis de documentos

Dependiente: Niveles y técnicas de seguridad				
Conceptualización	**Dimensiones**	**Indicadores**	**Ítems básicos**	**Técnica o Instrumento**
La aparición de la IoT ha traído consigo la probabilidad de conectar todo tipo de objetos, en este contexto es necesario que las aplicaciones para dispositivos IoT, aseguren la confidencialidad, integridad y autenticidad de los que almacenan y se envían entre distintos componentes. (Ganz et al., 2018) **Por lo que es importante proveer de seguridad no solo a los dispositivos IoT sino a todos los componentes que intervienen (capas de red, capas IoT, Aplicaciones Móviles).**	Aplicaciones móviles	Técnicas y niveles de seguridad en aplicaciones móviles.	¿Es posible que BlockChain mejore los niveles de seguridad de una aplicación móvil?	Análisis de documentos
	Dispositivos IoT	Técnicas y niveles de seguridad de un dispositivo IoT	¿Es posible que las técnicas que usa BlockChain mejore los niveles de seguridad en un dispositivo IoT?	Evaluar seguridad del prototipo desarrollado

Esto mediante uso de técnicas de seguridad: cifrado de datos, criptografía, algoritmos.				

CAPITULO II: MARCO REFERENCIAL

Marco Teórico

Antecedentes Investigativos

En los años noventa inicia una explosión investigativa sobre nuevas soluciones con arquitectura descentralizadas para ejecutar transacciones de pagos digitales eliminando la dependencia e intervención de servidores principales que monitoreen o regulen las transacciones realizadas, uno de los ejemplos más famosos es: "b-money" desarrollado por Wei Dai, la primera criptomoneda no rastreable y descentralizada para realizar pagos electrónicos, usando clave pública (Wey Dai, 2018).

El Ministerio de Ciencia y Tecnología de España en el año 2002 impulsa crear plataformas descentralizadas no dependientes de una autoridad central similares a BlockChain y se analiza sus posibles aplicaciones: una de ellas el IoT (BlockChain, 2017).

En 2008 Satoshi Nakamoto, inicia un estudio para crear una moneda digital llamada Bitcoin, la cual se basó en crear un sistema de seguridad prácticamente inmutable, combinando redes existentes P2P con técnicas de seguridad y algoritmos criptográficos de alto nivel, hoy conocido como BlockChain, el cual permite registrar todo tipo de transacciones en una red descentralizada, garantizando la seguridad de la moneda (Bahga & Madisetti, 2016; Satshi Nakamoto, 2008).

El 3 de enero de 2009 se publica con un programa de código abierto una de las criptomonedas más importantes en el mundo, todas sus operaciones se basan en una red BlockChain de tal manera que se registran a modo de un libro contable para evitar robos de transacciones y así asegurar que no existiese estafas cuando la moneda viajara entre nodos o circulaba por Internet (Francisco Javier Balmaseda Aranda, 2018).

En 2013, Vitalik Buterin[4], propuso incorporar contratos inteligentes a la red de BlockChain, los cuales permiten crear una lógica de negocio con la emisión de un "Token[5]" llamado "Ether", usado como criptomoneda (Navarro, 2018).

[4] Programador que Participó en el desarrollo de BlockChain
[5] Es un aparato electrónico que se le da a un usuario autorizado para facilitar el proceso de autenticación.

La teoría del Internet de las Cosas (IoT) nace en el siglo XIX, con la instalación de dispositivos de información meteorológica y de profundidad de nieve en la cima del Monte Blanc[6]. A través de un enlace de radio de onda corta, los datos eran transmitidos a París. En el siglo XX, varios estudios de telemetría impulsan la evolución de tecnologías de telecomunicación (Bruno Cendón, 2017).

El objeto más usado dentro de la tecnología IoT son los dispositivos móviles, los cuáles han evolucionado desde los años 80, hasta la presente fecha, desarrollando nuevas tecnologías de conexión, algunos protocolos de transmisión de datos y un sin número de aplicaciones que permiten interactuar con diferentes servicios y el Internet.

En 2014, "we live security by Eset" se destacó el descubrimiento de 73.000 equipos alrededor del mundo en este caso cámaras de seguridad que usaban las contraseñas por defecto o no tenían clave (Welivesecurity, 2014) esta recopilación la hace Isecam[7], una página que muestra cámaras en línea sin seguridad, el día 2 de agosto 2020 esta página muestra 13.436 cámaras sin seguridad.

En los primeros 5 meses de 2017, los investigadores de Kaspersky Lab detectaron 7242 muestras de programas malignos (malware) en dispositivos conectados a Internet, 74% más que el total de muestras detectadas en el lapso de 2013 a 2016, según los datos revelados por Kaspersky Lab durante la Séptima Cumbre Latinoamericana de Analistas de Seguridad, los ataques a dispositivos IoT se remontan al 2016, con la aparición del malware Mirai[8], donde el mundo se dio cuenta de los enormes riesgos que existen en los dispositivos "inteligentes" (Saldana Gustavo, 2017).

El ataque estaba dirigido al sistema de resolución de nombres (DNS), encargado de asegurar que las peticiones sean entregadas a la IP que corresponde, esto fue posible por la gran cantidad de dispositivos digitales conectados a Internet entre ellos routers y cámaras de vigilancia, aprovecho que son dispositivos que pueden generar mucho tráfico provocando un desbordamiento en los servidores. Los dispositivos atacados no contaban con seguridad, varios de ellos tenían contraseñas por defecto y estas en su gran mayoría son de conocimiento público en Internet (Jorge Alberto Virguez, 2019).

Actualmente, China es el mayor mercado del mundo en servicios de conectividad de IoT móvil. La base instalada creció un 124% respecto al año 2017, hasta alcanzar los 767 millones a fines de

[6] Grupo montañoso situado entre el Valle de Aosta en Italia y la Alta Saboya, en Francia.

[7] http://www.insecam.org/en/bycountry/FR/

[8] **Mirai**: malware de la familia de las botnets (robots automáticos de la red) destinada a infectar los equipos conformantes del IoT.

2018. Esto correspondió al 63% de los dispositivos a nivel global. Las cifras informadas por los operadores móviles no dejan ninguna duda de que China está liderando la adopción global del IoT masivo. Este país ha superado a Europa y América del Norte en términos de tasa de penetración (54,7 conexiones de IoT por cada 100 habitantes a finales de 2018) y está en camino de alcanzar los 1.000 millones de conexiones de IoT durante 2019 (IoT, 2019).

IoT vive una expansión acelerada debido a la conexión diaria de miles de dispositivos, sensores y chips al Internet los cuales obtienen y distribuyen información por el Internet para brindar servicios en distintos ámbitos de la vida diaria. IoT es capaz de recolectar y distribuir gran cantidad de información pero esta poderosa tecnología trae asociados nuevos riegos los cuales pueden amenazar la seguridad de los datos(Germán et al., 2019).

La tecnología IoT debe atravesar varios desafíos como: el desarrollo de infraestructura robusta, métodos de cifrado y seguridad que garanticen la confidencialidad, integridad y disponibilidad de la información. El gran desafío es la complejidad del ecosistema IoT, que impide que la mayoría de las empresas elaboren un marco de seguridad y privacidad. A diferencia de los equipos de TI, los dispositivos conectados no son diseñados pensando en la seguridad, y muchos de ellos no poseen capacidades esenciales de encriptación o autenticación (Germán et al., 2019).

Existen varias normas creadas para protección de dispositivos IoT, por ejemplo, la ITU-T Y.4806 presenta una clasificación de amenazas hacia los dispositivos IoT. Esta misma recomendación divide las capacidades de seguridad en 6 grupos: seguridad en la comunicación, seguridad en la gestión de datos, seguridad en la provisión de servicios, integridad de la seguridad, autenticación y autorización mutua, auditoria de la seguridad.

Por otra parte la IoT Security Foundation (Fundación de Seguridad IoT), propone un marco de referencias para cubrir aspectos de seguridad de los dispositivos IoT, basados en seguridad de objeto y procesos de negocio, puntos de conexión (cableado, conexiones wifi), resaltando 6 puntos: gestión responsable de la seguridad, diseño para la seguridad, criptografía, asegurar los framework[9] de redes y aplicaciones, asegurar los procesos de producción y la cadena de suministros, y seguridad para los consumidores (IoT Security Foundation, 2018).

CISCO hace foco en los controles que se realizarán sobre los datos recolectados, para poder detectar anomalías, así como la protección en las conexiones de redes, el crecimiento de los

[9] **Framework:** es un esquema de trabajo utilizado por programadores, permite agilizar los procesos de desarrollo evitando escribir código de forma repetitiva.

sensores basados en IP corresponde al crecimiento de la superficie de ataque en dispositivos IoT. Esto destaca el hecho de que se requieren nuevos protocolos de seguridad y técnicas de identificación, y la seguridad de los puntos finales de IoT debe correlacionarse con sus capacidades mejoradas. Claramente, IoT presenta nuevos desafíos para los arquitectos de redes y seguridad. Los sistemas de seguridad más inteligentes que incluyen la detección de amenazas administrada, la detección de anomalías y el análisis predictivo (CISCO, 2019).

Por su parte la Online Trust Alliance (OTA), propone un marco de confianza IoT que incluye un juego de principios estratégicos necesarios para asegurar los dispositivos IoT y sus datos cuando son enviados y a través de su ciclo de vida. El marco de confianza de IoT se divide en 4 áreas claves: 1. principios de seguridad, 2. acceso y credenciales del usuario, 3. privacidad, divulgaciones y transparencia, 4. notificaciones y mejores prácticas relacionadas. De estos principios, encontramos un nuevo ítem para esta clasificación: las notificaciones y avisos que proporciona el dispositivo al usuario final (Allliance Online TRust, 2018).

Por su parte, el CIS (Center for Internet Security), define una serie de controles de seguridad aplicables a Internet de las Cosas, y cada uno de estos controles vamos a incluirlos en nuestra clasificación. Como así también agregaremos una sección de respuesta a incidentes (Security, 2015).

El artículo denominado: "Modelo de Seguridad IoT" propone un modelo de seguridad basado en 62 recomendaciones de seguridad divididas en 10 áreas claves: procesos, políticas y responsabilidades, hardware de dispositivos y seguridad física, software de dispositivo y sistema operativo, interfaces cableadas e inalámbricas del dispositivo, autenticación y autorización, seguridad en la capa de aplicación, configuración de los dispositivos, analíticas de seguridad, respuestas a incidentes, notificaciones y alertas(Germán et al., 2019).

Debido a la naturaleza de los dispositivos IoT los profesionales deben seguir analizando tecnologías que puedan asegurar a los dispositivos y a los datos que se transmiten por esta tecnología con la finalidad de no solo tener políticas o lineamientos de buenas prácticas de seguridad, si no arquitecturas robustas que se ajusten al crecimiento y desarrollo de la IoT, BlockChain aspira ser una tecnología convergente que permita alcanzar resultados en seguridad muy eficientes.

Fundamentación Legal

Ley Orgánica de Protección de Datos Personales del Ecuador, con la finalidad de proteger la privacidad de los datos, se establece en los artículos 18, 26 y 27, algunas normativas para garantizar la protección de datos de las y los ecuatorianos.

"El artículo 18: Seguridad de datos personales, se refiere a los roles de los responsables y encargados de procesar los datos, así como el implementar medidas de seguridad adecuadas y de calidad para la protección de datos".

"Los artículos 26: Derecho de eliminación y el artículo 27: Derecho al olvido digital, destacan el derecho del consumidor a acceder a sus datos en cualquier entorno digital, así como su eliminación definitiva".

La mayoría de los usuarios desconocen el alcance, la capacidad de los dispositivos inteligentes que tiene en su custodia (celulares, tabletas) y la cantidad de datos que estos pueden recopilar a través de la tecnología IoT, al igual que la disponibilidad de sus datos los cuales pueden ser accesibles a terceros fuera del propósito para el que fueron adquiridos inicialmente.

Con el objetivo de analizar ciertos aspectos legales, es necesario destacar el Reglamento General de Protección de Datos de la Unión Europea concretamente, en los artículos 21 y 22 RGPD (María Burzaco Samper, 2020).

El artículo 21 introduce el derecho de oposición al procesamiento de datos, incluidos los perfiles de cualquier aplicación o herramienta. Si el propósito del procesamiento de datos es el marketing directo, el interesado tiene derecho absoluto a oponerse. Por otro lado, el artículo 22 introduce Restricciones adicionales contra la toma de decisiones automatizadas, donde una aplicación puede acceder a sus datos y replicarlos en otro lugar para fines específicos como el marketing y las ventas.

Una vez analizado todos estos reglamentos podemos concluir que tienen la finalidad de proveer a al profesional de lo legal, cómo proteger a sus clientes frente al uso y acceso de los datos, por lo que es importante y adecuado complementar toda esta parte legal con técnicas (hardware) que mejore los niveles de seguridad.

Esquema del Marco Teórico de la Investigación

IoT: Internet de las cosas.

IoT se puede definir como una red altamente interconectada de entidades heterogéneas, tales como, etiquetas, sensores, dispositivos embebidos, portátiles, móviles, etcétera, que interactúan y se comunican entre sí en tiempo real (Diego Cárdenas Quintero, Exel Ropero Silva, Karla Puerto López, Karla, Sanchez Mojica Sergio, Castro Casadiego Jhon, 2020).

IoT es una tecnología que se encuentra en auge que permite a objetos comunes interconectarse entre sí y transmitir datos. La Tecnología IoT se introdujo en los años 2008 y 2009 cuando el número de dispositivos fue mayor que el número de personas conectadas a Internet (Generalitat Valenciana, Union Europea, 2014).

Dada la evolución de esta tecnología, se evidencian varios problemas de seguridad e integridad de información, por lo que es fundamental ofrecer metodologías de investigación que generen soluciones a estos problemas, la tecnología IoT es uno de los elementos que se incluyó en el prototipo a desarrollarse, para mejorar su seguridad.

Modelos de Conectividad IoT.

Las implementaciones de IoT utilizan diferentes modelos de conectividad, cada uno de los cuales tiene sus propias características. Los cuatro modelos de conectividad descritos por la Junta de Arquitectura de Internet[10] (Internet Architecture Board**,** IAB**)** incluyen: dispositivo a dispositivo (Device-to-Device), dispositivo a la nube (Device-to-Cloud), dispositivo a puerta de enlace (Device-to-Gateway) intercambio de datos a través de la Aplicación (Back-End Data-Sharing). Estos modelos destacan la flexibilidad en las formas en que los dispositivos de IoT pueden conectarse y proporcionar un valor para el usuario.

Los dispositivos conectados hacia la nube son los que se encuentran más expuestos a los riesgos de seguridad, pues su información viaja a través del Internet, con la escasa seguridad que estos ofrecen, la intención de esta investigación es estudiarlos minuciosamente para plantear una solución segura.

[10] **Internet Architecture Board IAB**, Comité independiente de investigadores y profesionales con un interés técnico en la salud y la evolución del sistema de Internet. Los miembros de la IAB están profundamente comprometidos en hacer funcionar Internet de forma efectiva y evolucionar para satisfacer el futuro de alta velocidad a gran escala.

La página web 101BlockChain[11] explica como varios dispositivos inteligentes se pueden interconectar entre sí y a la nube sin la necesidad de la intervención humana, pero a pesar de ser una tecnología prometedora presenta varios problemas de seguridad, los cuales se enumeran en la Figura 1. Problemas de seguridad IoT.

Figura 1. Problemas de Seguridad IoT
Elaborado: (RODRÍGUEZ, 2019)

[11] https://101blockchains.com/

Arquitectura IoT:

En el artículo: Análisis Sistemático de la Seguridad en Internet of Things, sus autores realizan un análisis que se centra en detectar, encontrar y proponer soluciones que sean adecuadas para reducir las amenazas que los dispositivos IoT pueden sufrir debido a su fácil acceso y manipulación, Además se estable que es prioritario garantizar la seguridad en los dispositivos debido su impacto directo en la vida de los usuarios pudiendo así invadir su privacidad (Norma Beatriz Perez, Miguel Alfredo Bustos, Dr. Mario M. Berón, 2018).

Este mismo artículo hace un análisis de la arquitectura general de IoT y sus amenazas, la Figura 2 muestra dicha arquitectura y las amenazas que sufre cada capa (Norma Beatriz Perez, Miguel Alfredo Bustos, Dr. Mario M. Berón, 2018).

Los sistemas tradicionales de IoT a menudo están basados en una arquitectura centralizada (cliente – servidor). Los dispositivos recolectan la información y la envían a la nube donde será procesada y enviada de vuelta al dispositivo IoT, esto es una debilidad pues si el servidor principal es atacado todos los dispositivos corren riesgo, BlockChain al basarse en una red distribuida, compuesta por varios nodos participantes que verifican una o varias transacciones, garantiza que las transacciones ejecutadas sean seguras.

Arquitectura General IoT y sus Amenazas

Capa de Percepción: Realiza la identificación de objetos, recopila datos a través de sus sensores. En esta capa la seguridad se ve afectada debido a que varios sensores de diferentes marcas utilizan criterios de seguridad bastante limitados y en otras ocasiones bastante escasos, lo que genera que se puedan crear réplicas de las etiquetas y componentes o a su vez se produzcan lecturas, modificaciones y eliminaciones no autorizadas. Además, los dispositivos no autorizados pueden difundir información falsa (Mitrokotsa, Aikaterini & Rieback, Melanie & Tanenbaum, 2008).

Capa de Red: Transmite datos obtenidos de la capa de percepción a través de Internet, red móvil o cualquier otro tipo de red de comunicación confiable, en la capa de transporte, o capa de red, los delincuentes tienen la oportunidad de explotar conmutadores y puertos para robar o alterar los datos. Los delincuentes informáticos están buscando las vulnerabilidades de la red para ejecutar ataques como: inyección de código malicioso, ataque y contaminación al servidor central, y ataques conocidos como privación del sueño, que consisten en agotar la batería de los dispositivos (Norma Beatriz Perez, Miguel Alfredo Bustos, Dr. Mario M. Berón, 2018).

Capa de Nivel Medio: También conocido como middleware es un software que sirve de canal entre los componentes IoT para interactuar con otros dispositivos. Esta capa es parte de la arquitectura que permite la conectividad para una gran cantidad de cosas y garantizar conectividad para los sensores y también para las capas de aplicaciones que brindan servicios (Khan et al., 2012).

Capa de aplicación IoT: Es la encargada ejecutar las aplicaciones de IoT, para acceder a recursos o servicios (López, 2017).

La capa de aplicación de IoT es la más vulnerable; con este proyecto de investigación se pretende conocerla a profundidad, encontrar riesgos potenciales, crear una solución enfocada a esta capa, para eliminar vulnerabilidades, mediante la implementación de una red BlockChain que permita almacenar los datos transmitidos en una forma segura.

Figura 2. Arquitectura en IoT

Fuente:(Norma Beatriz Perez, Miguel Alfredo Bustos, Dr. Mario M. Berón, 2018)

BlockChain:

Una BlockChain no es otra cosa que una base de datos que se halla distribuida entre diferentes participantes, protegida criptográficamente y organizada en bloques de transacciones relacionados entre sí por medio de un modelo matemático secuencial para toda la red BlockChain. Expresado de forma más breve, es una base de datos descentralizada que no puede ser alterada la misma que garantiza la integridad de los datos. (Carlos Kuchkovsky, Gonzalo Gomez, Lardies, Daniel Diéz Garcia, 2017).

Dispositivos Móviles:

Cuando hablamos de dispositivos móviles no se refiere únicamente a celulares, se llama dispositivo móvil, a un artefacto electrónico, grande o pequeño, con algunas capacidades de procesamiento de datos que puede tener una conexión permanente o intermitente a una red, con una capacidad de memoria, que ha sido diseñado para ejecutar diferentes funciones. Existen multitud de dispositivos móviles, desde los reproductores de audio portátiles, hasta los navegadores GPS[12], pasando por teléfonos móviles, PDA[13], videojuegos portátiles, Tablet, PC's; cada uno de estos dispositivos móviles están diseñados para uso individual (Baz Arturo, Ferreira Irene, Álvarez María, 2011)

Los dispositivos móviles han demostrado ser parte indispensable de la vida diaria debido a la gran cantidad de información que se procesa en ellos, poniendo en riesgo la privacidad y disponibilidad de los datos. La necesidad de proteger la información se hace cada día más evidente, a medida que avanza la tecnología, las amenazas también son variadas, por ello en este plan de investigación se estudiará el funcionamiento de los diferentes dispositivos IoT que se relacionan con dispositivos móviles para desarrollar un método de protección eficiente.

Adopción de IoT con BlockChain

En el mes de diciembre 2019 Gartner lanzó una encuesta en línea en EE. UU. enfocada en tendencias de implementación de IoT con BlockChain, llamada: "Adoptadores[14] de IoT adoptan BlockChain", en la Figura 3. Se muestran los resultados de la encuesta efectuada a 500 empresas, donde se señala que el 75% de los adoptantes de tecnología IoT en los EEUU ya han adoptado BlockChain o planean adoptarlo para fines del 2020. Entre los adoptantes de BlockChain, el 90% está implementando las dos tecnologías juntas en varios proyectos como: farmacéuticas, cuidado de la salud, energía, trasporte, ciudades inteligentes en los próximos 25 meses. El estudio reveló que los productos farmacéuticos, la energía, la atención médica y el transporte son las industrias líderes en la adopción de BlockChain (Litan Avivah, 2019).

[12] **GPS**: Global Positioning System, en español Sistema de Posicionamiento Global

[13] **PDA**: Personal Digital Assistant, Asistente Digital Personal

[14] **Adoptadores**: Que adoptan, adquirir una configuración o forma determinada de algo.

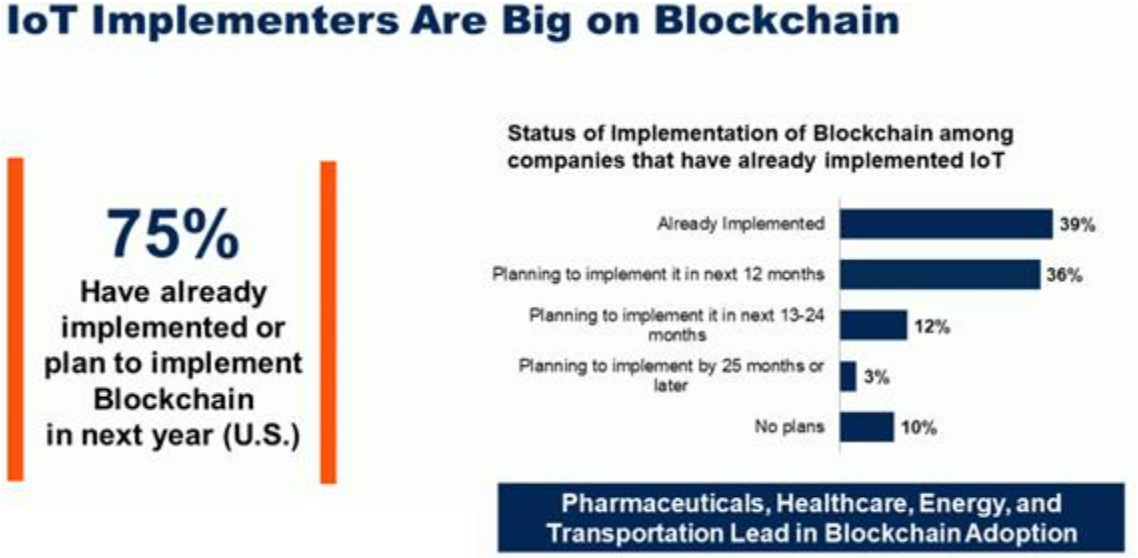


Figura 3. Adoptadores de IoT adoptan BlockChain

Fuente: (Litan Avivah, 2019)

Avivah Litan, vicepresidente y analista distinguida de Gartner, escribió: "Las redes BlockChain han surgido como una innovación prometedora debido a su capacidad para afirmar la integridad de los datos compartidos entre los constituyentes en la colaboración de procesos de múltiples partes. IoT ha surgido como un método para cerrar la brecha entre los recursos (o 'cosas') y sus procesos comerciales asociados. La integración de las redes IoT y BlockChain es un punto ideal para la transformación digital y la innovación"

IoT y BlockChain son tecnologías que prometen mucho por separado, ahora imaginemos cuanto potencial pueden entregar cuando se las integra: se creará un entorno confiable para IoT, el cual nunca ha sido evidenciado en la transmisión de datos entre redes o dispositivos virtuales, mejorando el intercambio de dicha información.

En el estudio de Gartner: "Adoptadores de IoT adoptan BlockChain" se descubre que el 63% de quienes han implementado ambas tecnologías considera un incremento en la seguridad y confianza como los motivos principales para la adopción de esta estrategia, con la finalidad de garantizar la confidencialidad, integridad y disponibilidad de la información. El informe de este estudio explica que, "la conexión de sensores y dispositivos de IoT a una BlockChain permite una pista de auditoría[15] inmutable de datos clave de IoT y eventos comerciales relacionados que se comparten entre múltiples participantes y que cada parte puede verificar independientemente", lo que permite

[15] **Auditoria:** Inspección o verificación de las operaciones tecnológicas de una empresa o una entidad, realizada por un auditor con el fin de comprobar si existe algún comportamiento inadecuado.

construir más seguridad, confianza y transparencia en torno a la gestión de cosas físicas conectadas entre sí y hacia el Internet.

La Figura 4. se muestra un resumen de los principales beneficios de BlockChain con IoT, entre los beneficios citados tenemos: mayor confianza, reducción de cotos. Una de las preguntas principales fue: ¿su empresa ha implementado o planea implementar la integración de una BlockChain con IoT? el 86% contestó si y el restante 14% no. El 63% dijo que el principal beneficio es mayor seguridad y confianza en transacciones y datos compartidos de múltiples partes. El 56% dijo que aumentó de la eficiencia empresarial y reducción de costes. El 43% eligió aumentar los ingresos y las oportunidades comerciales, mientras que solo el 37% eligió una experiencia mejorada de los constituyentes o participantes.

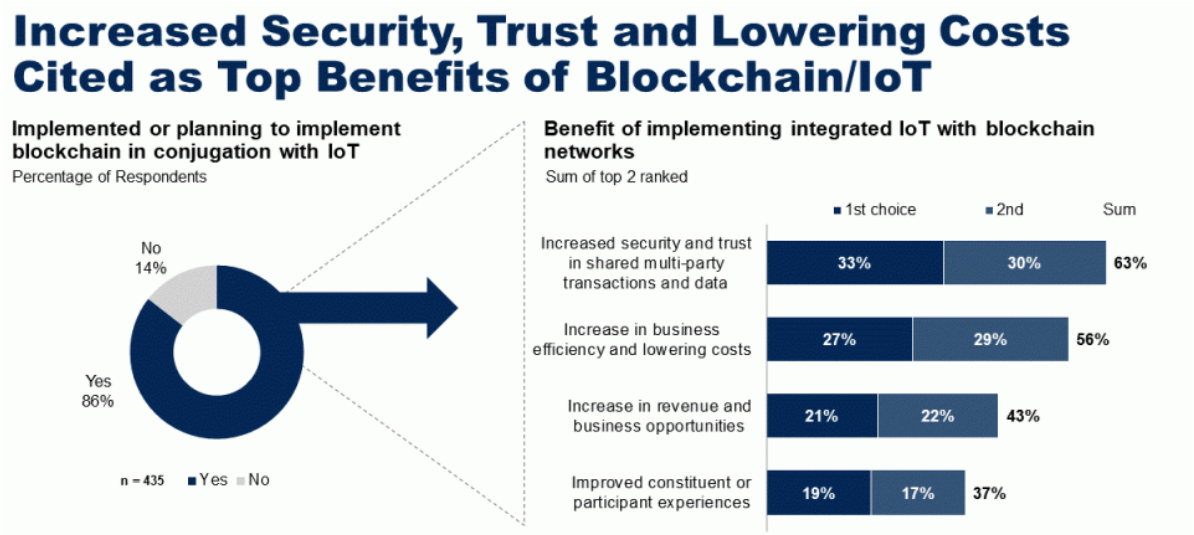


Figura 4. Razones porqué adoptar BlockChain con IoT
Fuente: (Litan Avivah, 2019)

A pesar de que el estudio de Gartner muestra algo prometedor entre la combinación de IoT y las tecnologías de BlockChain, se debe madurar e investigar mucho para obtener un verdadero afinamiento de estas dos tecnologías tanto en el ámbito técnico, en su arquitectura y su distribución comercial, las implementaciones de BlockChain implican cambios de protocolos siendo esto un desafío para los dispositivos IoT de larga duración. Algunas implementaciones de BlockChain luchan para escalar a las tasas de transacción que pueden generarse por un gran número de cosas conectadas o incluso gran acopio de datos transmitidos. A largo plazo, se espera que la combinación de IoT y BlockChain permita tener dispositivos innovadores y modelos comerciales mucho más eficientes, dijo Avivah Litan.

El crecimiento acelerado en el sector IoT pronostica que durante la próxima década habrá miles de millones de dispositivos IoT recientemente implementados(Dahlqvist et al., 2019). Este desarrollo introduce cambios en la arquitectura de TI con desafíos para mantener un control centralizado y pistas de auditoría adecuadas para resolver disputas forenses independientes.

Las seguridades actuales de IoT se basan principalmente en modelos cliente - servidor asegurando al dispositivo en un modelo centralizado, es decir nodos supervisados por una autoridad central, pero la tecnología avanza hacia los sistemas distribuidos y es necesario crear un nivel de seguridad que se ajuste a esta necesidad. La tecnología BlockChain, basada en un DLT (Distributed Ledger Technology) por su concepción y comportamiento es la solución ideal para dicho propósito (Wickström et al., 2020).

Los DLT asumen una comunidad de n* nodos que garantizan un consenso para validar una transacción, evitando tener una autoridad central la cual puede ser benevolente a cualquier ataque o suplantación de identidad. Una DLT puede soportar una gran cantidad de nodos que interactúan entre sí, mediante una secuencia matemática mejor conocida como código Hash, a la vez se puede comunicar mediante contratos inteligentes almacenados en la BlockChain en los cuales se almacenan reglas, que definan el acceso de los dispositivos IoT (Wickström et al., 2020).

La utilización de contratos inteligentes en BlockChain permite que los dispositivos IoT registren sus transacciones de una manera segura y auditable, manteniendo un historial fiable de sus acciones: conexión, inicio de sesión, instalación, reinstalación, actualización, envío de datos, entre otros. Esto permite llevar una bitácora de comportamientos de cada uno de los dispositivos en manera confiable e inmutable, al igual que las transacciones realizadas.

Cómo funciona la tecnología BlockChain

BlockChain nace como una respuesta a la necesidad de registrar la información en forma distribuida y operar de tal manera que los computadores o servidores conectados al registro tengan una copia de toda la Cadena de Bloques, BlockChain no solo se encarga de conocer donde está la información si no de saber a cargo de quien está la misma, cada integrante de BlockChain conocido como nodos[16] tiene un mismo nivel jerárquico y puede tomar decisiones, proposiciones y

[16] **Nodos:** El término nodo se utiliza en el lenguaje de las bases de datos para hacer referencia al servidor físico en el que se aloja la información. En el lenguaje de BlockChain, este término es usado por extensión para referirse a la persona o participante que está conectada al BlockChain utilizando el servidor nodo, puesto que, en las redes públicas, de las que hablaremos más adelante, todos los participantes van a interactuar como nodos con la red. En las redes privadas y federadas es necesario ser más preciso o explícito al definir los conceptos de nodo y participante.

validaciones. Lo que no pasa en un modelo centralizado, un único servidor está a cargo de la toma de decisiones, pudiendo ser vulnerable a un ataque. En BlockChain todos los miembros o nodos deben estar de acuerdo en una transacción, caso contrario si uno de ellos no está de acuerdo no existe un consenso lo que haría que cualquier acción que se vaya a realizar se la rechace, es decir no es aceptada (Carlos Kuchkovsky, Gonzalo Gomez, Lardies, Daniel Diéz Garcia, 2017).

BlockChain y P2P (Peer to Peer)

"P2P hace referencia a la interacción que tiene cada nodo de BlockChain, Los nodos no están conectados todos entre sí, cada nodo está conectado con un número determinado del Resto. Cuando un nodo quiere realizar una transacción, le envía la información sobre la misma a aquellos con los que está conectado y estos la replican con aquellos con los que ellos, a su vez, están conectados" (Marcos Allende, 2018).

La arquitectura peer-to-peer de BlockChain ofrece muchos beneficios. Entre los más importantes está el hecho de que las redes P2P ofrecen mayor seguridad que los arreglos tradicionales de cliente-servidor. La distribución de BlockChain en un gran número de nodos los hace prácticamente inmunes a los ataques de Denegación de servicio (DoS) que afectan a numerosos sistemas. Del mismo modo, dado que la mayoría de los nodos deben establecer un consenso antes de agregar datos a una BlockChain, es casi imposible que un atacante altere los datos (Academy, 2020).

Criptografía y algoritmos Matemáticos de BlockChain

Para entender el principio de seguridad con el que trabaja BlockChain para la protección e integridad de los datos, a continuación, se analizarán los elementos criptográficos con los que trabaja esta tecnología. Los elementos criptográficos mencionados son los fundamentos básicos hacia la firma digital que se emite cuando un objeto es un participante legítimo de una red BlockChain, con la capacidad de guardar, editar, eliminar una determinada transacción. La misma que será distribuida a toda la red (Francisco Javier Balmaseda Aranda, 2018).

BlockChain es el conjunto de varios elementos que proveen una seguridad robusta, la cadena de bloques sin sus mecanismos anexos de seguridad como: cifrado asimétrico, algoritmos criptográficos, certificados digitales, autenticación y autorización tipo SAM y protocolo SSL/TLS utilizado para la capa de transporte, no podría proveer confidencialidad y privacidad de los datos; la sinergia de estos elementos es lo que permite trabajar en un entorno seguro (Francisco Javier Balmaseda Aranda, 2018).

Algoritmos Criptográficos (Francisco Javier Balmaseda Aranda, 2018)

La red BlockChain emplea firmas digitales, estas firmas son generadas por procesos criptográficos innatos de la red. La versatilidad de la red BlockChain permite que los desarrolladores de este tipo de soluciones escojan el mejor algoritmo dependiendo de su lógica de negocio y las transacciones a almacenarse.

Algoritmo de cifrado simétrico: Utiliza una única clave para cifrar y descifrar los datos transmitidos, la clave esta compartida entre el transmisor y el receptor (Diego Cárdenas Quintero, Exel Ropero Silva, Karla Puerto López, Karla, Sanchez Mojica Sergio, Castro Casadiego Jhon, 2020).

Algoritmo de cifrado asimétrico: Conocido como clave pública, se dispone de dos claves vinculadas matemáticamente de forma inversa conocida como entropía[17] y reversibilidad[18]. Que se traduce a la capacidad de que el propietario de la clave privada puede calcular la clave pública, sin embargo, quien posee la clave pública no puede calcular la clave privada, siendo esto computacionalmente algo improbable. Este algoritmo se caracteriza porque cada extremo posee dos claves inversas entre sí, de tal manera que en un extremo se cifra la información y en el otro se la descifra. Otra de sus características es que son complejos y lentos, cuando se habla de BlockChain la red firmar digitalmente las transacciones enviadas, junto con la operación hash con la finalidad de verificar que son seguras y demostrar la identidad del emisor de la transacción (Francisco Javier Balmaseda Aranda, 2018).

En BlockChain se utiliza este tipo de cifrado con dos finalidades:

Identificar Cuentas*:* BlockChain mantiene una bitácora donde registra a los usuarios o cuentas de usuarios con la finalidad de tener un mapeo entre los propietarios y las transacciones que son de propiedad de dichos usuarios, ésta bitácora se la conoce como Wallet o billetera digital, este elemento almacena todos los datos necesarios de un usuario para permitir la ejecución de transacciones o a su vez la negación de la misma (René Bástian Silva, 2019).

***Validar Transacciones*:** Las transacciones incluyen un dato que sirve como prueba de que un determinado usuario de cuenta transfiere la propiedad en la transacción enviada con la finalidad

[17] La entropía puede ser considerada como una medida de la incertidumbre y de la información necesaria para, en cualquier proceso, poder acotar, reducir o eliminar la incertidumbre

[18] Es la capacidad de un dato

de que todos los nodos de la red BlockChain puedan inspeccionar los datos de la transacción. Aquí se aplica el principio de asimetría: el propietario de la cuenta que transfiere la propiedad crea un texto cifrado con su clave privada. Todos los demás nodos de la red pueden verificar esta prueba de acuerdo, mediante el uso de la clave criptográfica pública (René Bástian Silva, 2019).

Operaciones Criptográficas Hash.

Esta operación es la que garantiza la integridad de las transacciones enviadas entre los bloques de la red BlockChain, garantizando que los datos no hayan recibido ninguna manipulación o cambio alguno (Juan Gómez Ortega, 2019). La función Hash está compuesta por 256 bits también se la conoce como función resumen, debido a que puede conocer donde se aloja un determinado dato mediante una huella digital única, estas funciones no son consideradas algoritmos de cifrado, son funciones matemáticas computables que permite autenticar mensajes y comprobar integridad. Un Hash tiene tamaño fijo, el mismo que se adjunta al mensaje enviado, este tamaño es superior al mensaje transmitido; si se llegara a cambiar de alguna manera el mensaje el Hash es modificado notablemente. La firma digital avanzada es la combinación de la función Hash con la criptografía asimétrica, lo que permite una prueba de integridad al verificar la identidad de los dispositivos que transmiten (Francisco Javier Balmaseda Aranda, 2018).

Una función Hash toma como entrada un mensaje que llamaremos m y calcula un string de tamaño fijo *H(m)*. Los valores contenidos en un hash pueden tener ceros a la izquierda, con la finalidad de completar la longitud la requerida estas funciones deben cumplir las siguientes propiedades:

- Proporciona valores Hash para cualquier tipo de datos transmitido en forma rápidamente.
- Produce idénticos valores Hash para datos de entrada idénticos.
- El valor hash devuelto cambia impredeciblemente cuando se cambian los datos de entrada, haciendo evidente un ataque en los datos.
- No proporciona ninguna forma de rastrear sus valores de entrada a partir de sus salidas. Lo que lo hace seguro y fiable,
- Es muy difícil encontrar dos o más datos distintos para los que arroja el mismo valor Hash.

Debido a estas propiedades, las funciones Hash se las conoce como "huellas digitales" porque cada dato transmitido queda marcado como único, de tal manera que los valores transmitidos pueden ser utilizado para comprar, referencia y verificar cambios. A continuación, algunos

ejemplos de Hash obtenidos con la función Keccak-256 que se emplea en la BlockChain de Ethereum[19].

Tabla 1. Ejemplos de función Hash Keccak-256

H (Ciervo) ↔ 77df4019245db73f96931eed9405972cb521b17c821a456cc8f9f0e28ed5f211
H (siervo) ↔ a9771c52fa9066cb7018b52d1dfdcaca9df9a5bad637c13681bc0ba4a3e352db
H (siervos) ↔ 792826d67143f9969e723d068a3f84370fb48a579734043fc35808151edbe717

Fuente: (Tool Online, 2020.)

Como se puede apreciar en la Tabla 1. Pese a la similitud en las palabras de entrada, se obtiene Hash totalmente distintos y de una longitud fija (36 caracteres).

En la Figura 5, se describe el proceso de cifrado con clave pública y hash, un emisor *X* obtiene el hash *h* de su mensaje *m* y adjuntarlo al envío de la transacción, luego se cifrar este paquete (m;h) con su llave privada, y nuevamente con la clave pública de su receptor y cuando es recibido el paquete es descifrado por el receptor con la llave privada, el receptor *Y* puede ejecutar la misma función Hash usada por X sobre el mensaje, con la finalidad de verificar que la "huella digital" del mensaje sea la misma que se recibió, si resulta diferente, entonces significa que el texto del mensaje real fue dañado o alterado en el camino.

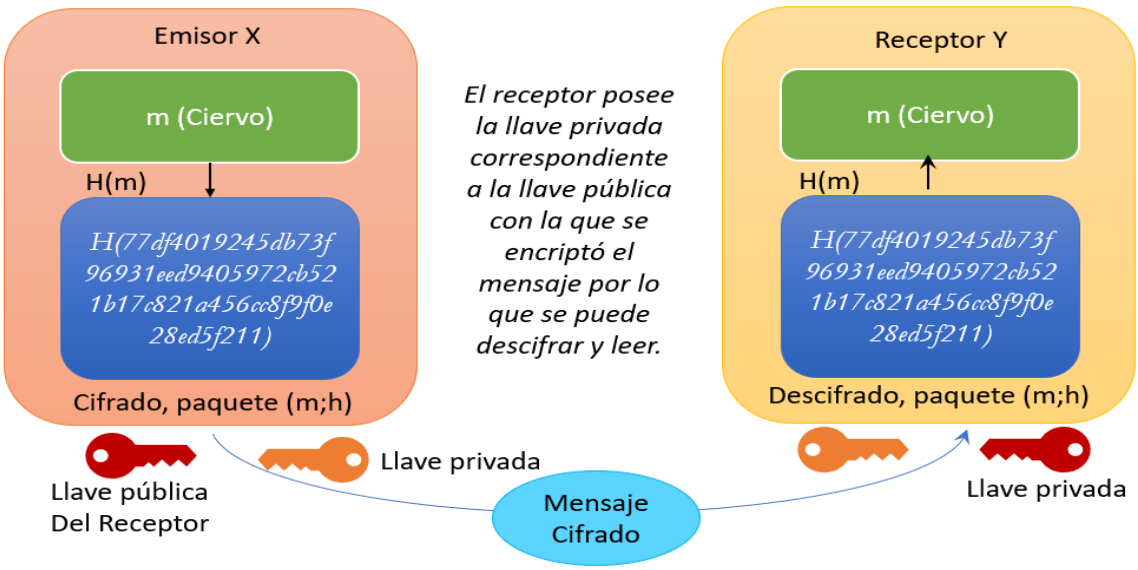


Figura 5. Cifrado Asimétrico con Hash

Fuente: Investigador

[19] **Ethereum:** Es una plataforma open source, que sirve para programar contratos inteligentes. La plataforma es descentralizada a diferencia de otras cadenas de bloques.

Árbol Hash de Merkle

Creado por Ralph Merkle, en el año de 1970 conocido como un árbol binario, es una estructura de datos empleada para resumir y verificar de una manera eficiente la integridad de un gran volumen de datos, contiene Hash criptográficos, se lo conoce como un "árbol" debido a su estructura de ramificaciones. Cada nodo participante en la red BlockChain tiene una etiqueta Hash la misma que hace referencia al nodo anterior y al subsiguiente. Esta estructura de datos permite que un gran número de datos, ubicados en diferentes ramificaciones puedan ser entificados por el valor que almacena el Hash y conocer su historial de una manera eficiente. De esta manera se proporciona un método de verificación segura y eficiente de los contenidos de grandes estructuras de datos. En sus aplicaciones prácticas normalmente el hash del nodo raíz va firmado para asegurar su integridad y que la verificación sea totalmente fiable. La Figura 6. muestra cómo se estructura el árbol de Merkle.(VALDÉS, 2019)

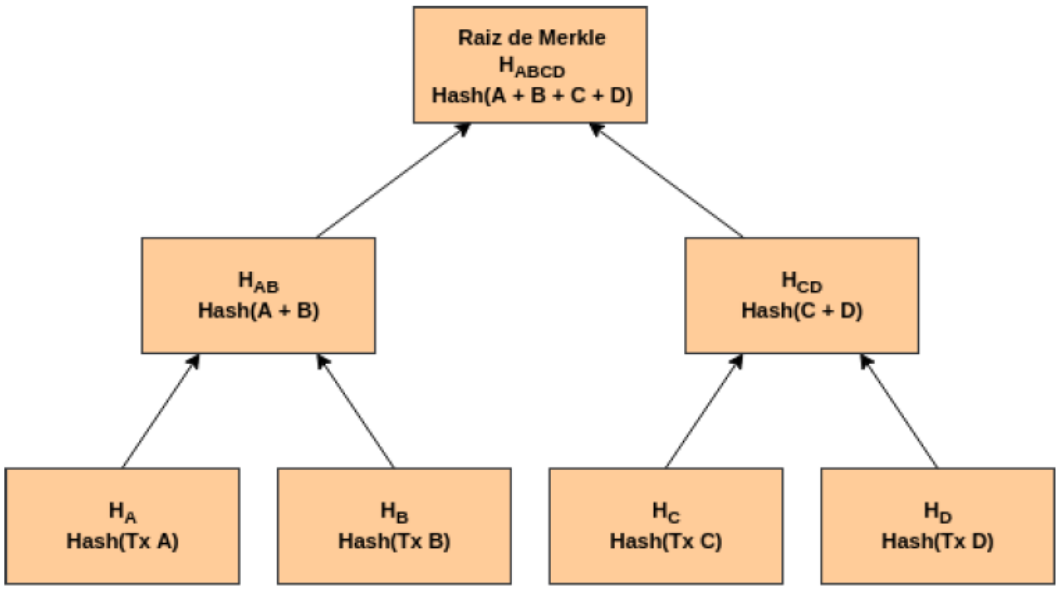


Figura 6. Árbol de Merkle

Fuente: (VALDÉS, 2019)

Consenso la clave de BlockChain

"BlockChain es una base de datos que se encuentra distribuida entre diferentes miembros/ nodos protegida criptográficamente y organizada en bloques de transacciones relacionados entre sí matemáticamente mediante una función hash, la misma que no puede ser alterada. Se trata de un sistema que permite que partes que no confían plenamente entre ellas pueden mantener un consenso sobre la existencia, estado y evolución de una serie de factores compartidos" (Carlos Kuchkovsky, Gonzalo Gomez, Lardies, Daniel Diéz Garcia, 2017).

El consenso es la clave de un sistema BlockChain permite que todos los participantes puedan confiar en la información que se encuentra grabada en él. Este consenso se sustenta en un protocolo común que verifica y confirma las transacciones realizadas y asegura la irreversibilidad de ellas mismas. Este consenso deber proporcionar a todos los usuarios una copia inalterable y actualizada de todas las operaciones realizadas en BlockChain.

Smart Contracts

Para John Stark, los smart contracts o contratos inteligentes también conocido como Chaincode software agente son un elemento fundamental de la tecnología BlockChain, establecen y definen cómo y quién puede llevar a cabo qué transacciones.(Christidis & Devetsikiotis, 2016). Son contratos en los que se definen y especifican una serie de cláusulas, como los controles a cumplir por la mercancía mencionados en la sección previa y el pago final acordado en caso de que estos sean superados. La diferencia con los contratos usuales es que estos son incorporados a un BlockChain o Cadena de Bloques en la red, que garantiza su seguridad y proporciona el entorno adecuado para su procesamiento automático (Marcos Allende, 2018).

Un contrato inteligente es un código informático que actúa como un acuerdo vinculante entre dos o más partes cualesquiera sin necesidad de un intermediario y cuyas cláusulas se programan previamente otorgándole la capacidad de ejecutarse validando así, el cumplimiento de las cláusulas. Este contrato debe ser totalmente digital, tiene capacidad sobre los activos digitales que se vayan transmitir , con la finalidad de validar el cumplimiento de las condiciones acordadas y debe ejecutarse de forma autónoma ya automática(Carlos Kuchkovsky, Gonzalo Gomez, Lardies, Daniel Diéz Garcia, 2017).

Tipos de Contratos Inteligentes (Academy, 2020)

Distribuidos. Son replicados y distribuidos por todos los nodos de la red BlockChain.

Determinísticos. Se caracteriza por realizan únicamente acciones para las que fueron diseñados, siempre y cuando las condiciones se cumplan. El resultado será siempre el mismo, sin importar que nodo de la red lo ejecute.

Autónomos. Pueden automatizar todo tipo de tareas, funcionando como programas autónomos o autoejecutables.

Inmutables. Permanecen estáticos, no pueden ser modificados, pero si eliminados una vez desplegados en la red BlockChain.

Customizables. Antes de ser desplegados, pueden ser codificados de muchas maneras distintas. Así que, pueden ser empleados para crear múltiples tipos de aplicaciones descentralizadas (DAPPS[20]). Está característica está vinculada al hecho de que la red BlockChain como la de Ethereum puede ser afinada completamente, para un determinado uso.

Trustless. Dos o más partes pueden interactuar sin conocerse o confiar la una en la otra. Además, la tecnología BlockChain garantiza que los datos sean precisos.

Transparentes. Dado que están basados en una BlockChain pública, su código fuente no sólo es inmutable sino también visible para todo el mundo.

Seis pasos para transmitir datos en un BlockChain – BC (Marcos Allende, 2018)

Paso 0: Cualquier persona puede convertirse en parte de la red BC, mediante la descarga de la aplicación correspondiente o accediendo vía una interfaz web, donde la primera opción será descargar la versión más actual de la BlockChain ya sea pública o privada.

Paso 1: Una vez que los participantes están conectados a la red BlockChain es momento de enviar datos en forma de transacciones, las cuales son validadas por los nodos hacia los que están conectados.

Paso 2: Cada nodo registra las transacciones que va escuchando es su lista o pool. Las listas de otros nodos no siempre coinciden debido a que los nodos escuchan las transacciones en ordenes distintos.

Paso 3: En cada ronda de transmisión de datos, un nodo es escogido aleatoriamente para proponer el almacenamiento de un bloque, el cual debe ser consensuado por toda la red BlockChain.

El nodo elegido que propone el bloque para consenso sigue el siguiente proceso:

- Roma la versión más actual de toda red, al cual añade al final del bloque las transacciones que haya registrado en su pool
- Envía esta nueva copia de la cadena a todos los nodos con los cuales esté conectado.
- Los nodos replican esta cadena a toda la red

[20] **DAPPS:** Decentralized Applications o aplicaciones descentralizadas.

Paso 4: El nodo elegido propone un bloque con las transacciones que ha ido escuchando y registrando en su pool. Antes de grabar el bloque este debe ser validado con una función Hash el cual es obtiene a partir de toda la información del bloque.

Paso 5: El sistema BlockChain tiene protocolos internos que permiten validar si se tiene un Hash válido, en el caso de determinar que es válido, el Resto de los nodos de la red BC verifican que todas las transacciones también sean correctas y actualizan su copia con la nueva versión que tiene el nuevo bloque.

Propuestas basadas en contratos inteligentes de BlockChain para aseguran la implementación de IoT

Revisada la literatura asociada, se determinó la conveniencia de analizar algunos artículos científicos, los cuales muestran la importancia de aplicar BlockChain para mejorar la seguridad en los dispositivos IoT.

Wickström y Col: en su artículo Rethinking IoT Security: A Protocol Based on BlockChain Smart Contracts for Secure and Automated IoT Deployments, donde utilizan la tecnología de bloques basados en criptografía, para asegurar que las redes distribuidas de IoT se mantengan seguras y fiables a lo largo de su ciclo de vida. La solución presentada en este artículo emplea contratos inteligentes deterministas[21] e interrelacionados en la Cadena de Bloques del Ethereum para gestionar y mantener seguros de los dispositivos de IoT, se desarrolla protocolos para asegurar el despliegue de dispositivos de IoT y los medios para comunicarse de forma segura con el Resto de dispositivos (Wickström et al., 2020).

Restuccia y Col: proporciona una encuesta de aplicación de IoT basada en BlockChain para energía inteligente, entornos inteligentes, robótica, transporte, cadena de suministro y otros. Destacan que los resultados muestran que varias aplicaciones de IoT se beneficiarían de las tecnologías BlockChain, como los contratos inteligentes. Varias soluciones basadas en BlockChain propuestas para controlar el acceso a dispositivos IoT dependen de contratos inteligentes(Restuccia et al., 2019; Wickström et al., 2020).

Anderson y col.: presentan un sistema de autorización basado en BlockChain llamado WAVE (Wide Area Veri fi ed Exchange) para dispositivos IoT. Los agentes integrados representan dispositivos IoT, servidores y puertas de enlace en la capa de superposición. Los agentes y

[21] Determinista: Implica entender la realidad como la consecuencia directa de una causa

enrutadores interactúan con la Cadena de Bloques Ethereum subyacente a través de cuatro contratos inteligentes: contrato de objeto WAVE, contrato de registro, contrato de alias y contrato de afinidad de enrutador (Andersen John Kolb Kaifei Chen Gabriel Fierro David E Culler Raluca Ada Popa et al., 2017; Wickström et al., 2020).

Novo: propone una arquitectura para la gestión de acceso a IoT mediante contratos inteligentes. La solución utiliza unidades llamadas centros de administración entre la red IoT y los contratos inteligentes. Esto se debe en parte a las limitaciones de recursos, para evitar tener que sincronizar un nodo BlockChain completo para cada dispositivo IoT. Su solución introduce un componente adicional en el sistema que constituye un punto de centralización (Oscar Novo, 2019).

Rashid y Siddique: comparan BlockChain con soporte de contrato inteligente y proponen las siguientes instrucciones de investigación relacionadas con el uso de contratos inteligentes en soluciones de seguridad de IoT basadas en BlockChain: resistencia contra ataques híbridos, plataformas óptimas del sistema IoT, protocolos de auditoría, gestión de confianza y recuperación alianza en redes sociales, minería con eficiencia energética y protocolos de consenso híbridos (Rashid & Siddique, 2019).

Biswas y col.: proponen un marco escalable basado en BlockChain para sistemas IoT, donde una Autoridad de Certificación (CA) local registra y autentica dispositivos IoT locales. Las aplicaciones de IoT usan un Kit de desarrollo estándar (SDK) de BlockChain para crear contratos y transacciones inteligentes. Una configuración de banco de pruebas para el marco propuesto utiliza HyperLedger Fabric (Androulaki et al., 2018; Biswas et al., 2019; Hyperledger, 2020).

Hang y Kim: proponen una plataforma de cadena de bloques IoT en capas para garantizar la integridad de los datos detectados por los dispositivos IoT. Los dispositivos IoT vinculados constituyen una capa física de IoT. Un cliente de capa de aplicación ejecuta un contrato inteligente que consiste en una función para el registro del propietario del dispositivo, una función de registro del dispositivo, una función de lectura del sensor, una función de escritura del actuador y una notificación de evento para actualizar el libro mayor de BlockChain (Hang & Kim, 2019).

Protocolos de Consenso de BlockChain (Marcos Allende, 2018)

El protocolo de consenso es el procedimiento mediante el cual se elige a un nodo para proponer un nuevo bloque, la elección es aleatoria, esto garantiza que la responsabilidad no caiga sobre un solo bloque.

Proof-of-Work (PoW). Este protocolo exige un esfuerzo a los participantes en el sorteo para determinar quién propone el siguiente bloque, y se da una recompensa al ganador. El esfuerzo aquí consiste emplear capacidad computacional muy costosa para encontrar el código hash que valide el bloque anterior. Comúnmente conocidos como mineros, en Bitcoin.

Proof-of-Stake (PoS). Este protocolo funciona bajo la petición de pruebas de posesión consisten en asignar mayor probabilidad de ganar el sorteo a aquellos que tienen más activos (criptomonedas) en la red. El algoritmo premia a aquellos que tienen más monedas mediante una baja dificultad computacional para conseguir nuevos bloques.

Leased-Proof-of-Stake (LPoS). LPoS una variación de PoS en la que usuarios con poco capital pueden ceder sus probabilidades de ganar el sorteo. En caso de que el nodo en el que hayan delegado resulte el ganador y haya algún tipo de recompensa por el minado, se reparte proporcionalmente entre él y las personas que lo apoyaron.

Delegated-Proof-of-Stake (DPoS). Este protocolo mejor conocido como la prueba de participación delegada, es uno de los algoritmos más fiables, robustos y más eficientes dentro de una red BlockChain, creado por: Daniel Larimer (Jose Maldonado, 2019), desarrollador principal de Bitshare[22]. La prueba de participación delegada funciona de tal manera que los miembros son los principales actores en cualquier red BlockChain, ellos son los responsables de votar por los testigos a quienes se les asigna la tarea de verificar las transacciones y producir los bloques.

Proof of Autority (PoA): es un mecanismo de consenso alternativo donde existen ya varios nodos de autoridad pre aprobados, que reciben el nombre de selladores, es una prueba modificada de prueba de PoS (Jose Maldonado, 2019).

Proof of Importance (PoI). PoI funciona como PoW pero asignando la probabilidad de ser elegido en función de la actividad -transacciones, balance o reputación- en la red del nodo en lugar de su dinero. La idea es la misma, premiar con el derecho de proposición de bloques a las personas que más interesadas están en el buen funcionamiento de la cadena, de forma que no les convenga proponer bloques maliciosos que puedan perjudicarla.

[22] **Bitshare**: Se trata de una casa de cambio o Exchange descentralizado y basado en la tecnología BlockChain, donde los usuarios pueden hacer transacciones y negociar con criptomonedas y otros valores.

Tipos de BlockChain

Luego del surgimiento de la red BlockChain de Bitcoin, nació la idea que la Cadena de Bloques se puede ajustar a cualquier tipo de transacción, o a cualquier acuerdo realizado por el Internet., con la finalidad de mantener transacciones seguras, fiables, rastreables e inmutables. A partir de esta idea nació el proyecto Ethereum una red BlockChain con propiedades muy diferentes a la de Bitcoin, se caracteriza por las transacciones programables. Las instituciones privadas como los bancos tomaron esta idea para emplearla como un registro de un libro contable, el mismo que se encuentra distribuido por toda la red hacia todos sus nodos, dando lugar a las redes BlockChain privadas, las cuales únicamente pueden ser accedidas mediante una autenticación(Reyes Delgado, 2018).

A partir de estos ejemplos las redes BlockChain son categorizadas en:

BlockChain Pública: Cualquier persona sin ser usuario puede acceder y consultar las transacciones almacenadas de la red. Ejemplos: Bitcoin, Ethereum, Monero, Dash, Litecoin, Dodgecoin, entre otras (Alexander Preukschat, Carlos Kuchkovsky, Gonzalo Gómez Lardies, 2017).

BlockChain Privada: Aquí las transacciones se mantienen privadas, solo pudiendo ser visualizada para aquellos nodos que forman parte de la red y sus miembros se encuentran registrados en una Wallet. Ejemplos: Hyperledger Fabric, Corda de R3 (industrias financieras), XRP Ledger de Ripple.

Hybrid BlockChain: Sus transacciones son públicas, pero sus miembros y los accesos a la red son restringido, los accesos se los controla mediante un archivo o mediante un administrador de red. Ejemplos: Hyperledger Fabric, BigchainDB, Evernym.

Herramientas de BlockChain

Hyperledger

"La Fundación Linux desarrolló el proyecto Hyperledger en 2015 para promover nuevas tecnologías entre industrias. En lugar de declarar un único estándar BlockChain, fomenta un enfoque colaborativo para desarrollar tecnologías BlockChain a través de un proceso comunitario, con derechos de propiedad intelectual que fomentan el desarrollo abierto y la adopción de estándares clave a lo largo del tiempo". (hyperledger.org, 2021)

Hyperledger es una comunidad de código abierto enfocada al desarrollo de un conjunto de framework[23], herramientas y librerías, con el fin de ser usadas en soluciones BlockChain en el sector empresarial para la mejora de servicios.

Hyperledger Fabric

Es una plataforma que diseña soluciones para utilizar un libro mayor distribuido para datos, este libro funciona de forma descentralizada, registra todas las transacciones que se ejecutan en una red BlockChain. Hyperledger está respaldada por una arquitectura modular que ofrece altos grados de confidencialidad, resistencia, flexibilidad y escalabilidad. Está diseñado para admitir implementaciones convergentes entre varios elementos y componentes los cuales permiten adaptarse a la complejidad que existen en todo el ecosistema económico, industrial y tecnológico.

La arquitectura de Hyperledger Fabric emplea contratos inteligentes también conocidos como Smart Contracts o Chaincode: Un contrato inteligente es un código, invocado por una aplicación cliente externa a la red BlockChain, que administra el acceso y las modificaciones a un conjunto de pares clave-valor en el estado mundial a través de la transacción. En Hyperledger Fabric, los contratos inteligentes se empaquetan como código de cadena. Chaincode se instala en pares y luego se define y se usa en uno o más canales.

[23] **Framework:** Un entorno de trabajo, o marco de trabajo es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

CAPITULO III: MARCO METODOLÓGICO

Tipo de Investigación

La investigación planteada es de tipo cuantitativo y cualitativo; se hizo un análisis de los procesos y elementos que intervienen en el experimento de BlockChain para luego emplear parámetros que permitan medir sus niveles de seguridad. En la Figura 7. Se muestra el proceso investigado para alcanzar los resultados del experimento.

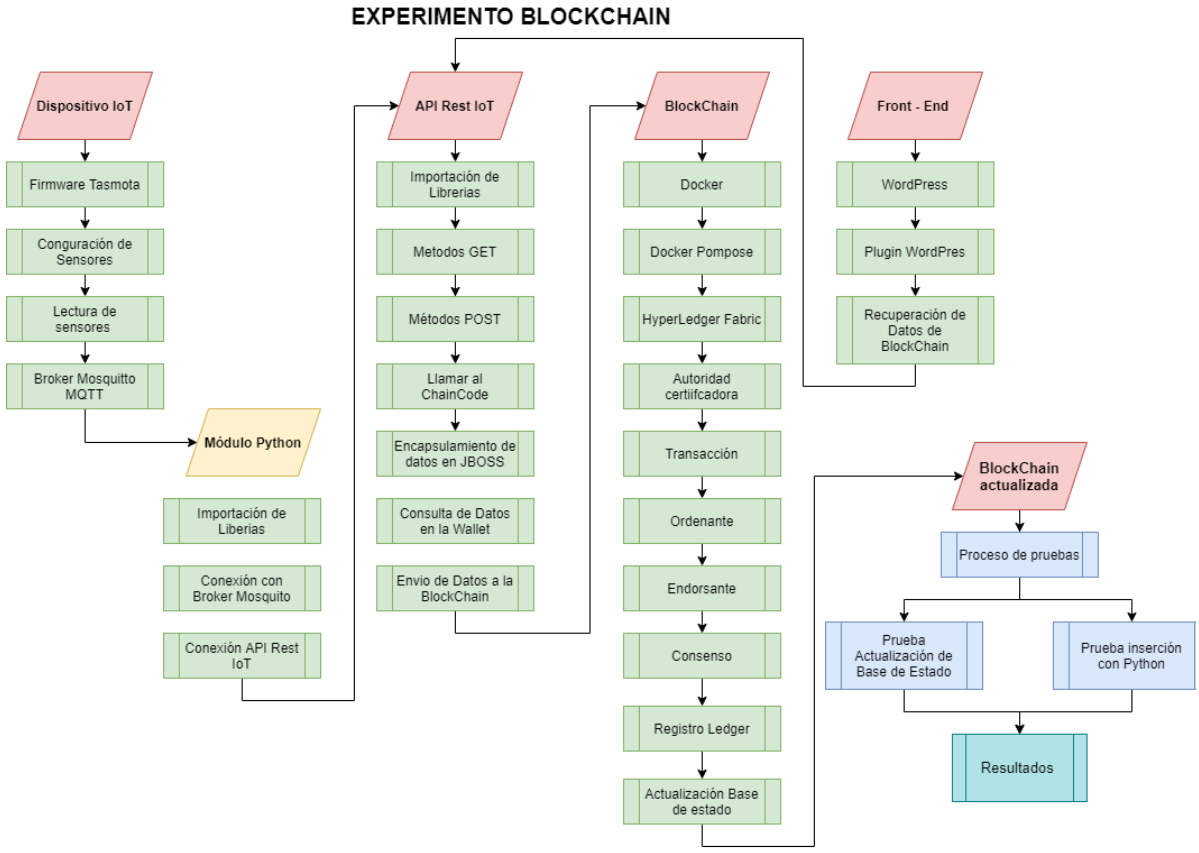


Figura 7. Arquitectura IoT BlockChain

Fuente: Investigador

Diseño de Investigación

Modalidad de Investigación

Investigación Documental: Se usó diversas fuentes de investigación como: libros, documentos relacionados, artículos y revistas científicas los cuales sirvieron para la construcción del marco metodológico, así como para el análisis de las variables dependientes e independientes. Con ello permiten desarrollar fundamentos que se plantear a lo largo del documento.

Investigación Bibliográfica: Este estudió se apoyó en investigaciones similares que permitieron ser una guía para diseñar el prototipo planificado. Además de aportar conceptos actuales, criterios científicos, filosóficos y legales(José Daniel & Leticia, Bertha, n.d.).

Investigación Experimental: Esta investigación se realizó en un ambiente controlado con Servidores Virtuales, con el fin de recrear un ambiente seguro con el fin de describir la causa y efecto que produce la investigación planteada (Bernal, 2012).

Para lo cual se crearon varios ambientes virtualizados para: BlockChain, APPI Rest IoT, servidor Mosquito, WordPress, con la finalidad de validar la teoría de asegura los datos.

Investigación exploratoria: esta investigación permite obtener nuevos datos y elementos que pueden ayudar posteriormente con una investigación descriptiva, esta investigación proveerá una metodología aplicable de BlockChain para dispositivos IoT móviles.

Investigación descriptiva: esta investigación describe las características de BlockChain e IoT, aspectos que ayudaran a entender mejor su funcionamiento y forma de aplicación.

Investigación aplicada: esta investigación se centra en encontrar mecanismos y estrategias que permitan proponer y desarrollar un prototipo que mejore la protección de información conjugando dos tecnologías: BlockChain e IoT.

Métodos de Investigación

Inductivo: Para esta investigación se empleó el método argumentos inductivos van de los casos particulares a una hipótesis general. Por lo tanto, las inferencias inductivas son ampliadoras o no-explicativas: sus conclusiones van más allá de lo que está contenido en sus premisas. El método inductivo se conoce como experimental y sus pasos son: 1) observación, 2) formulación de hipótesis, 3) verificación, 4) tesis, 5) ley y teoría (Dávila Newman, 2006). "Este Método utiliza el razonamiento para obtener conclusiones que parten de hechos particulares aceptados como válidos para llegar a conclusiones cuya aplicación sea de carácter general." (Bernal, 2012).

Estrategias Técnicas

Se utilizará la técnica de análisis de artículos, libros, y varia bibliografía debido a que es una investigación experimental y los resultados del experimento se almacenarán en una base de datos distribuida.

Instrumentos de Investigación

- Red BlockChain
- Sensores IoT
- Aplicación WordPress
- Servidores Virtuales en la nube

CAPITULO IV: MARCO ADMINISTRATIVO

Viabilidad

El desarrollo de la presente investigación ha tomado a consideración los siguientes aspectos:

Factibilidad Operativa: Durante el análisis de la documentación que conforma el marco metodológico de esta investigación se determina que existe factibilidad operativa por las razones descritas en la Tabla 2.

Tabla 2. Factibilidad Operativa

ASPECTO DE LA FACTIBILIDAD OPERATIVA (Varela, 2008)	INVESTIGACIÓN
El sistema puede ser demasiado complejo para los usuarios.	Se proveerá de toda la documentación necesaria tanto para la implementación como para el uso de la aplicación.
Un sistema puede hacer que los usuarios se resistan a él.	Al ser una aplicación móvil la gente está bastante familiarizada con este tipo de tecnologías.
Un sistema nuevo puede introducir cambios demasiado rápidos que no permita al personal adaptarse a él y aceptarlo.	La tecnología BlockChain e IoT es relativamente nueva por lo que permitirá desarrollar investigaciones similares.
La probabilidad de obsolescencia subsecuente en el sistema propuesto.	Desarrollar un aplicativo basado en BlockChain con Dispositivos IoT y que sea manipulado desde una aplicación móvil su probabilidad de obsolescencia es mínima, debido a que son tecnologías de tendencia y convergencia según la encuesta aplicada por Gartner. (SignalIoT, 2020)

Fuente: Investigador

Factibilidad Técnica.

Permite evaluar si el equipo y software están disponibles y tienen las capacidades técnicas requeridas por cada alternativa del diseño que se esté planificando, también se consideran las interfaces gráficas entre los sistemas actuales y los nuevos.

Así mismo, estos estudios consideran si las organizaciones tienen el personal que posee la experiencia técnica requerida para diseñar, implementar, operar y mantener el sistema propuesto.

Factibilidad Económica.

Dentro de estos estudios se pueden incluir el análisis de costo/beneficio asociado con cada alternativa del proyecto.

Con análisis de costo/beneficios, todos los costos y beneficios de adquirir y operar cada sistema alternativo se identifican y se establece una comparación entre ellos. Esto permite seleccionar el más conveniente para la empresa.

Dentro de esta comparación la Tabla 3, se debe tomar en cuenta lo siguiente:

Tabla 2. Factibilidad Operativa

ASPECTO DE LA FACTIBILIDAD OPERATIVA	INVESTIGACIÓN
Se comparan los costos esperados de cada alternativa con los beneficios esperados para asegurarse que los beneficios excedan los costos.	En el mercado no existen muchos desarrolladores de estas tecnologías, por ello es importante generar esta investigación como material didáctico para el uso de desarrolladores que permitan bajar los costos.
La proporción costo/beneficio de cada alternativa se comparan con las que proporcionan los costos/beneficios de las otras alternativas para escoger la mejor.	Actualmente los almacenamientos son en sistemas tradicionales, el almacenamiento en una BlockChain permite garantizar la integridad de los datos, al encontrarse almacenado en un servidor virtual los costos son bastante reducidos.

Se determinan las formas en que la organización podría gastar su dinero	Las organizaciones dedicadas al almacenamiento de los datos podrían vender la data almacena, generando un bien de valor.
La probabilidad de obsolescencia subsecuente en el sistema propuesto.	Las tecnologías empleadas de la investigación se encuentran en auge por lo tanto no hay obsolescencia tecnológica.

Fuente: (Varela, 2008)

CAPITULO V: EXPERIMENTO

ELEMENTOS DEL EXPERIMENTO

Elementos del Experimento en el Entorno IoT

LoLin new NodeMCU V3: Es una tarjeta de desarrollo orientado a IoT, basado en un chip ESP826624, el mismo que da la posibilidad de conectarse a Wifi promedio del protocolo TCP/IP, está diseñado especialmente para trabajar montado en un proto board[25] o soldado sobre una placa. Es programable con Arduito por lo permite una rápida integración, la placa está compuesta por 30 pines, 15 a cada lado. Tienen una salida USB para alimentación de energía y un mini USB de carga y envío de datos.(Prometec . Net, 2019)

Tasmota (Firmware): Creado y Mantenido por Theo Arends. Es un es un programa informático de código abierto que establece la lógica de más bajo nivel permitiendo controlar circuitos electrónicos conectados a una placa de cualquier tipo, basados en ESP8266 como Sonoff[26], Wemos[27] y NodeMCU; permite controlarlos a través de una interfaz web, denominada Web UI, donde se pueden configurar diferentes parámetros, como las entradas, el protocolo de comunicación MQTT[28] ((Message Queue Telemetry Transport)) o HTTP; habilitar entradas y salidas GPIO[29] (General Purpose Input/Output) que la ESP8266 puede soportar.(Arends., n.d.)

Sensores: Un sensor es un dispositivo que está capacitado para detectar acciones o estímulos externos y responder o ejecutar una acción. Estos aparatos pueden transformar las magnitudes

---

[24]**ESP8266:** Es chip integrado con conexión WiFi y compatible con el protocolo TCP/IP.

[25] **Proto board:** es un tablero con orificios que se encuentran conectados eléctricamente entre sí de manera interna, habitualmente siguiendo patrones de líneas, en el cual se pueden insertar componentes electrónicos y cables para el armado y prototipado.

[26] **Sonoff:** Interruptor inteligente que permite controlar el encendido/apagado de cualquier dispositivo o aparato eléctrico/electrónico.

[27] **Wemos:** Marca china que fabrica placas electrónicas basadas en ESP8266.

[28] **MQTT:** transporte de telemetría de Message Queue Server es un protocolo de red ligero de publicación-suscripción que transporta mensajes entre dispositivos.

[29] **GPIO:** Entrada/Salida de Propósito General, Es un pin genérico en un chip, cuyo comportamiento (incluyendo si es un pin de entrada o salida) se puede controlar (programar) por el usuario en tiempo de ejecución.)

físicas o químicas en magnitudes eléctricas y como resultado se tienen lecturas numéricas, lecturas booleanas.(Cázarez Ayala et al., 2014)

Servidor Eclipse Mosquitto: Es un broker [30] Open Source desarrollado por la fundación Eclipse y distribuido bajo licencia EPL[31]/EDL[32]. Está programado en C y es multiplataforma, Mosquitto MQTT es uno de los broker más utilizados debido a su agilidad y los bajos recursos que emplea en el momento de publicar y suscribir mensajes, por ello su aplicación es bastante amplia cuando se refiera a tecnología IoT.(Mosquito.org, n.d.)

Python: Es un lenguaje de programación bastante sencillo de aprender, su lógica se interpreta no se compila, sus palabras claves son claras y legibles Se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional. (Leonardo J. Caballero G., 2019)

Modulo Python: Permite organizar lógicamente código Python. Agrupando código relacionado dentro de un módulo hace el código más fácil de entender y usar. Un módulo es un objeto de Python con atributos con nombres arbitrarios que puede enlazar y hacer referencia.(Leonardo J. Caballero G., 2019)

Elementos del Experimento Entorno Api de Servicios (API Rest)

Node.js: Es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome, es de código abierto, su lógica pertenece a una comunicación asíncrona para la capa del servidor, además tienen una arquitectura orientada a eventos cuando se trata de acciones entrada / salida. Node.js está diseñado para crear aplicaciones de tipo red las cuales pueden ser escalables. (Gallagher, 2021)

API Rest: API significa interfaz de programación de aplicaciones, son un conjunto de funciones que se definen según la lógica de negocio que se requiera implementar, utiliza protocolos para desarrollar e integrar el software de las aplicaciones. Las API permiten que los productos y servicios

[30] **Broker**: es el servidor que acepta mensajes publicados por clientes y los difunde entre los clientes suscritos. Publicar: cuando un cliente envía un mensaje al broker. Tópico: los mensajes deben estar etiquetados con algún tópico o tema para detectar de donde viene.

[31] **EPL:** Licencia Pública Eclipse.

[32] **EDL:** Licencia de Distribución de Eclipse.

de una solución se comuniquen con otros entornos, sin necesidad de saber cómo están implementados (Red Hat, 2021).

SDK Hyperledger Fabric: permite que las aplicaciones interactúen con una red BlockChain. Proporciona una API simple para enviar transacciones a un libro mayor o consultar el contenido de un libro mayor con un código mínimo (Nao Nishima, 2021).

Wallet: Una billetera contiene un conjunto de identidades de usuario. Una aplicación ejecutada por un usuario selecciona una de estas identidades cuando se conecta a un canal[33]. Los derechos de acceso a los recursos del canal, como el libro mayor, se determinan utilizando esta identidad en combinación con un MSP[34] (Proveedores de membresías para servicios).

Profile.yaml: Es un perfil de conexión, describe un conjunto de componentes, incluidos pares, ordenadores y autoridades de certificación en una red BlockChain. También contiene información sobre el canal y la organización relacionada con estos componentes. Una aplicación utiliza principalmente un perfil de conexión para configurar una puerta de enlace que maneja todas las interacciones de red, lo que le permite centrarse en la lógica empresarial. Un perfil de conexión lo crea normalmente un administrador que comprende la topología de la red BlockChain (Nao Nishima, 2021).

Elementos del Experimento Entorno BlockChain:

Hyperledger Fabric: Es una plataforma que diseña soluciones para utilizar un libro mayor distribuido para datos, este libro funciona de forma descentralizada, registra todas las transacciones que se ejecutan en una red BlockChain. Hyperledger está respaldada por una arquitectura modular que ofrece altos grados de confidencialidad, resistencia, flexibilidad y escalabilidad. Está diseñado para admitir implementaciones convergentes entre varios elementos y componentes los cuales permiten adaptarse a la complejidad que existen en todo el ecosistema económico, industrial y tecnológico. La Arquitectura de Hyperledger Fabric emplea contratos inteligentes también conocidos como Smart Contracts o Chaincode: Un contrato inteligente es un código, invocado por una aplicación cliente externa a la red BlockChain, que administra el acceso y las modificaciones a un conjunto de pares clave-valor en el estado mundial a través de la transacción.

[33] **Canal:** Un canal es una capa de BlockChain privada que permite el aislamiento y la confidencialidad de los datos.

[34] **MSP:** Membership Service Provider

En Hyperledger Fabric, los contratos inteligentes se empaquetan como código de cadena. Chaincode se instala en pares y luego se define y se usa en uno o más canales.

Chaincode (Contrato Inteligente): Es un programa escrito en algún lenguaje compatible con la red BlockChain (Go[35], node.js o Java), dicho programa ejecuta la lógica empresarial y permite mantener una comunicación con la BlockChain de tal manera que permite actualizar el libro mayor, además, se ejecuta en un contenedor de Docker seguro. El Chaincode inicializa y administra el estado del libro mayor o Ledger a través de las transacciones enviadas por aplicaciones (Nao Nishima, 2021).

Peer (Pares): Una red BlockChain se compone principalmente de un conjunto de nodos pares (o, simplemente, pares). Los pares son un elemento fundamental de la red porque albergan el libro mayor y los Chaincode. Este libro mayor registra de manera inmutable todas las transacciones generadas por los Chaincode o Contratos Inteligentes. Los contratos inteligentes y los libros de contabilidad se utilizan para encapsular los procesos y la información compartidos en una red, respectivamente (Akhil, 2018; Nao Nishima, 2021).

Canal: Es una capa de la BlockChain privada que permite el aislamiento y la confidencialidad de los datos. Existen dos tipos de canales, el canal de aplicaciones en el que los miembros del canal comparten el mismo libro mayor y código de cadena para un propósito comercial específico, se puede tener tantos canales de aplicación como sea posible. El canal de sistema es en el que se almacena la configuración del consorcio[36]. Siempre que haya una actualización sobre el consorcio, como una nueva organización que se une al consorcio, necesitamos actualizar este canal del sistema. Los canales están definidos por un bloque-configuración.

Bloque-configuración: Contiene los datos de configuración que definen organizaciones miembros y políticas para un canal (servicio de ordenamiento). Cualquier modificación de configuración a un canal o red general (por ejemplo, un miembro que se va o se une) dará como

[35] **Go:** Lenguaje de programación concurrente y compilado inspirado en la sintaxis de C, que intenta ser dinámico como Python y con el rendimiento de C o C++.

[36] **Consorcio**: conjunto de organizaciones que consensan entre si para validar una transacción en BlockChain

resultado un nuevo bloque de configuración que se agregará a la cadena correspondiente. Este bloque contendrá el contenido del bloque de génesis[37], más el delta[38] .

Docker: Es un proyecto de código abierto que permite crear contenedores Linux los cuales optimizar recursos a la hora de desplegar aplicaciones. Permite la entrega de código en forma eficiente y rápida. Con Docker se virtualizan un sin número de instancias que tienen un objeto común de administración (Docker, 2021).

Docker Compose: Es una herramienta que permite simplificar el uso de Docker. A partir de archivos YAML es más sencillo crear contendores, conectarlos, habilitar puertos, volúmenes.(Docker, 2021). Con Compose se puede crear diferentes contenedores y al mismo tiempo, en cada contenedor, diferentes servicios, unirlos a un volumen común, iniciarlos y apagarlos. Es un componente fundamental para poder construir aplicaciones y microservicios.

CouchDB: Es un gestor de bases de datos de código abierto, cuyo foco está puesto en la facilidad de su uso y en ser "una base de datos que asume la web de manera completa". Se trata de una base de datos NoSQL que emplea JSON para almacenar sus transacciones, CouchDB es la base de datos principal de BlockChain en el cual se alojan las transacciones y su libro diario también conocido como Ledger (ApacheCon, 2021; Nao Nishima, 2021).

Hyperledger Fabric CA o Autoridad Certificadora: Es una entidad de confianza, responsable de emitir y revocar los certificados digitales o certificados utilizados en las transacciones de BlockChain. Es la autoridad de certificación predeterminada, que emite certificados basados en PKI[39] a las organizaciones miembros de la red y sus usuarios. La CA emite un certificado raíz (rootCert[40]) a cada miembro y un certificado de inscripción (ECert[41]) a cada usuario autorizado. Esta autoridad es la encargada de validar que el certificado que se ha enviado es el correcto y mediante un consenso aprueba o desaprueba una transacción de un ordenante.

---

[37] **Bloque Génesis:** Es el primer bloque de la BlockChain

[38] **Bloque Delta:** son los bloques subsecuentes al bloque génesis.

[39] **PKI (Public Key Infrastructure**): es la combinación de hardware, software, un grupo de políticas y procedimientos de seguridad, garantiza el cifrado de datos.

[40] **rootCert:** Certificado raíz certificado de clave pública sin firma o auto firmado que identifica la autoridad de certificación

[41] **Ecert**: se almacena una credencial, para firmar y autorizar transacciones.

Ordenante: Su función es ordenar las transacciones en un bloque y luego distribuye los bloques a los pares conectados para su validación y confirmación. El servicio de ordenamiento existe independientemente de los procesos de pares y las transacciones de pedidos se basan en el orden de llegada para todos los canales de la red. Está diseñado para admitir implementaciones conectables más allá de las variedades Kafka [42]y Raft[43] listas para usar. Es un enlace común para toda la red; contiene el material de identidad criptográfico vinculado a cada Miembro u Organización (Nao Nishima, 2021).

Organización: También conocidas como «miembros», las organizaciones están invitadas a unirse a la red BlockChain por un proveedor de red BlockChain. Una organización se une a una red agregando su Proveedor de servicios de membresía (MSP) a la red. El MSP define cómo otros miembros de la red pueden verificar que las firmas (como las de las transacciones) fueron generadas por una identidad válida, emitida por esa organización. Los derechos de acceso particulares de las identidades dentro de un MSP se rigen por políticas que también se acuerdan cuando la organización se une a la red. Una organización puede ser tan grande como una corporación multinacional o tan pequeña como un individuo. El punto final de la transacción de una organización es un Peer. Una colección de organizaciones forma un Consorcio. Si bien todas las organizaciones de una red son miembros, no todas las organizaciones formarán parte de un consorcio (Nao Nishima, 2021).

Protocolo gRPC: Este protocolo se define como un tipo de framework[44], es un sistema moderno que llama procedimientos remotos que procesan la comunicación en estructuras cliente-servidor distribuidas o peer-to-peer desde la capa de aplicación de una manera eficiente gracias a una innovadora ingeniería de procesos. Permitiendo mensajería bidireccional basadas en flujo (IONOS, 2020).

Protocolo de Consenso RAFT: Este protocolo utiliza un modelo de "líder y seguidor", en el que un líder se elige dinámicamente entre los nodos ordenantes de en un canal (esta colección de nodos se conoce como el "conjunto de consentimiento"), y ese líder replica los mensajes a los nodos

---

[42] **Kafka**: protocolo de consenso de Hyperledger.

[43] **Raft:** es un algoritmo de consenso que selecciona a un líder en forma dinámica de tal manera que si alguno de los nodos no se encuentra disponible, el algoritmo rápidamente busca otro nodo reemplazo para validad las transacciones enviadas..

[44] **Framework:** es una estructura real o conceptual destinada a servir de soporte o guía para la construcción de algo que expande la estructura en algo útil.

seguidores. Debido a que el sistema puede soportar la pérdida de nodos, incluidos los nodos líderes, siempre que quede la mayoría de los nodos de pedido (lo que se conoce como "quórum"), se dice que RAFT es "tolerante a fallas de bloqueo" (byzantine fault tolerance[45]). En otras palabras, si hay tres nodos en un canal, puede soportar la pérdida de un nodo (dejando dos Restantes). Si tiene cinco nodos en un canal, puede perder dos nodos (dejando tres nodos Restantes). Esta característica de un servicio de pedidos de RAFT es un factor en el establecimiento de una estrategia de alta disponibilidad para su servicio de pedidos. Este protocolo es ideal para un entorno de producción y redes amplias entonces si uno de los nodos fallas la red BlockChain puede seguir operando con normalidad (Nao Nishima, 2021).

Elementos del Experimento – Entorno Front End

Apache Tomcat: Apache es un servidor web HTTP de código abierto, para plataformas Unix, Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 (ApacheOrg, 2021).

WordPress: s un sistema de gestión de contenidos lanzado el 27 de mayo de 2003, enfocado a la creación de cualquier tipo de página web, micrositio, aplicación compatible con cualquier dispositivo tecnológico.

Plugins WordPress: es un fragmento de código que se conecta a tu sitio web de WordPress. En pocas palabras, es una extensión a tu sitio que modifica y mejora las funciones principales de WordPress.

Boostrap: Es una biblioteca que permite crear un marco de interfaz gráfica para sitios y aplicaciones web, el cual se constituye de varias plantillas en HTML y CSS, permitiendo crear elementos gráficos como formularios, botones, marcos que mejoran la interfaz de un usuario final.

SOLUCIÓN PLANTEADA

El objetivo de esta investigación es mejorar la seguridad y privacidad de los datos de un dispositivo IoT en la capa de aplicación, por lo tanto, se propone que BlockChain es una tecnología de

[45] **Byzantine fault tolerance:** Es la resistencia de un sistema informático tolerante a faltas, en particular los sistemas informáticos distribuidos, a fallas de componentes electrónicos donde hay información imperfecta sobre si un componente falla.

convergencia que ayudar a mitigar y atenuar la inseguridad de los dispositivos IoT, evitar perdida de datos, modificación y eliminación de estos.

Con el fin de verificar la validez de esta investigación se empleó un experimento en donde se integran varios elementos y entornos los cuales se muestran en la Figura 8. Arquitectura Experimento.

Herramientas de Desarrollo

A continuación, se detallan las herramientas de desarrollo que se emplean en el experimento y sus versiones.

- Tasmota 9.5.0
- Servidor Mosquito MQTT 1.6.7
- Hyperledger Fabric 2.2.0
- Docker 3.9
- Docker Compose.3.3.3
- Node JS 14.17.1
- Go, versiones 1.14.x y 1.15.x.
- JDK 8.
- Git. 2.32.0
- Python 3.8
- WordPress 5.7.2
- Servidor Virtual Ubuntu 18.

ARQUITECTURA.

La arquitectura empleada para este experimento integra varios elementos que permiten crear un entorno que intenta ser seguro apoyados en una red BlockChain, la misma que interactúa por medio de contratos inteligentes para la inserción y recuperación de datos leídos desde un dispositivo IoT, pero sin la facultad de modificaciones o eliminaciones, protegiendo la información almacenada.

La Figura 8, muestra los entornos que interactúan en este experimento: por una parte el entorno IoT que es el que genera la información que será enviada a la BlockChain mediante una lectura de datos de varios sensores conectados a una placa LoLin new NodeMCU V3 y transmitidos por un protocolo MQTT hacia un módulo Python el cual interactúan con el Entorno API de servicios que almacena los microservicios para enviar datos de cada sensor, validando que los usuarios registrados sean los correctos para luego mediante un canal enviar los datos hacia el Entorno la red BlockChain que contiene los Chaincode para cada sensor que permiten almacenar los datos y

por último el entorno Front - End que muestra los datos leídos de los sensores y guardados en la BlockChain, cada entorno trabaja totalmente separado pero depende de forma indirecta del otro para almacenar transacciones.

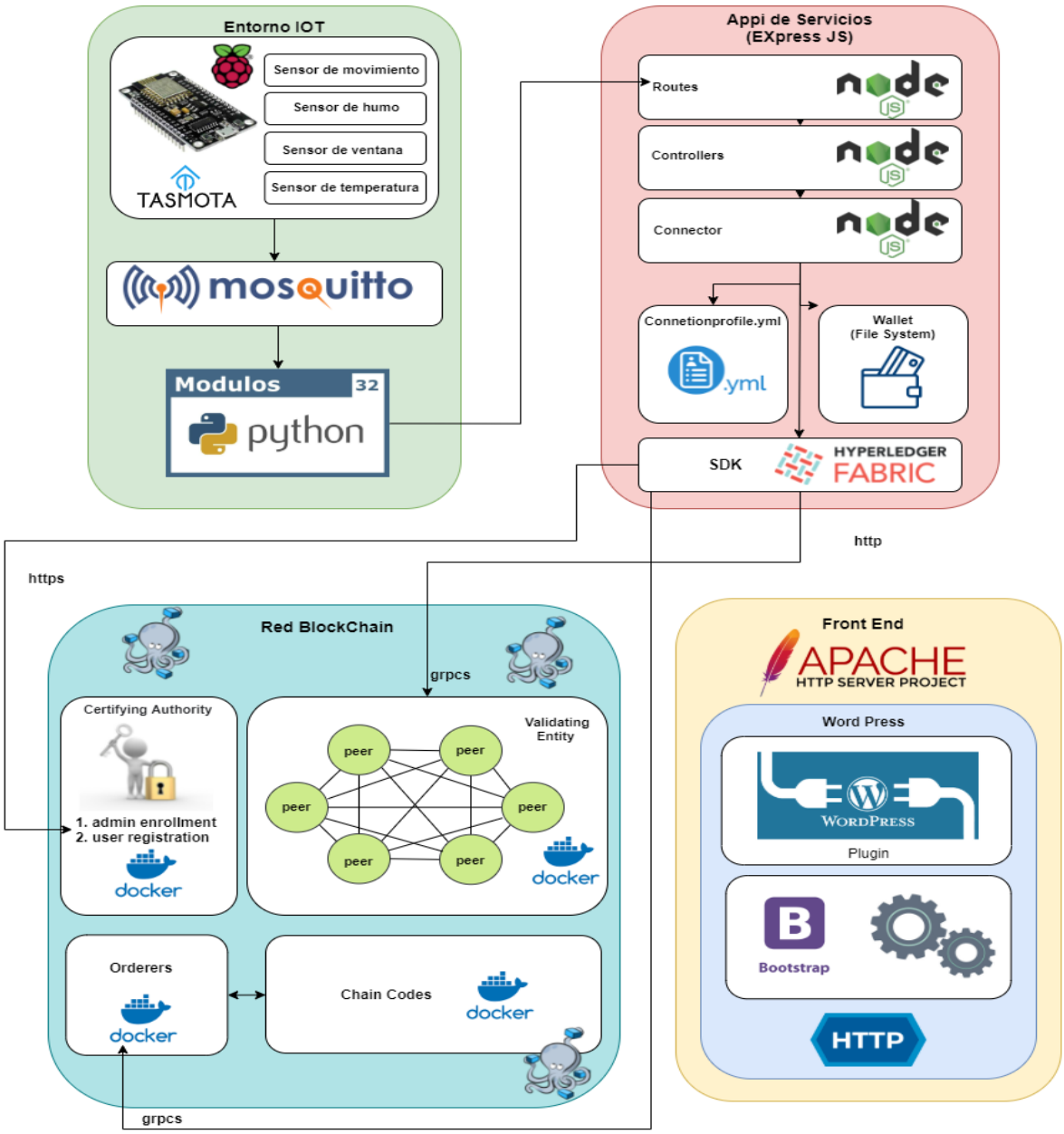


Figura 8. Arquitectura IoT BlockChain

Fuente: Investigador

A continuación, se describe cada entorno que compone la arquitectura planteada.

Entorno IoT

El Internet de las cosas ha permitido generar un sin número de soluciones que permiten producir datos para ser evaluados y analizados con la finalidad de dar soluciones y mejorar la calidad de vida de las personas.

Para este caso de uso se desarrolló un prototipo de casa inteligente que almacena los datos de 5 sensores. La Figura 9, muestra el circuito IoT ensamblado en una placa LoLin new NodeMCU V3 como dispositivo de control que se conecta fácilmente a una red Wifi, esta placa ha sido programada con la ayuda de Arduino[46] y flasheada[47] con el firmware Tasmota permitiendo crear un servidor web llamado WebUI. Cuando cualquier dispositivo conectado accede a este servidor web, la LoLin new NodeMCU V3 lee los sensores conectados y los muestra en una interfaz gráfica bastante amigable, los sensores conectados son:

- la temperatura, humedad, punto de rocío con el sensor DHT22.
- el movimiento detectado con el sensor PIR infrarrojo HC-SR505 que emite un valor booleano (ON, OF).
- Un sensor de humo metano butano MQ2, que emite un valor booleano (ON, OF).
- Un sensor de ventana, que emite un valor booleano (ON, OF).

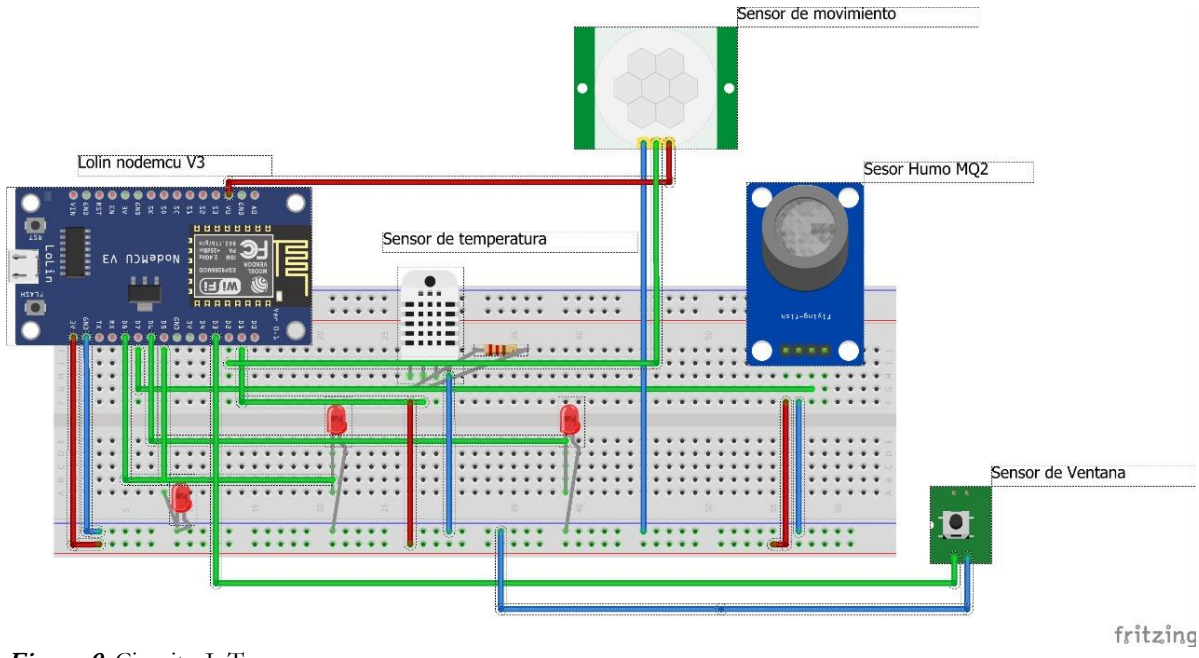


Figura 9. Circuito IoT

Fuente: Investigador

Herramientas para programar el Circuito IoT

1. Descargar instalador de Arduino

[46] **Arduino:** es una plataforma abierta que facilita la programación de un microcontrolador.

[47] **Flashear:** Instalación de un firmware en un dispositivo electrónico.

https://www.arduino.cc/en/Main/Software

2. Instalando Plugin del ESP8266 para Arduino.
   https://naylampmechatronics.com/blog/56_usando-esp8266-con-el-ide-de-arduino.html
3. Instalar el CHIP USB-SERIAL CH340
   http://en.doit.am/CH341SER.zip
4. Descarga Tasmota y cargar en la placa base.
   https://github.com/arendst/Tasmota
5. Configurar el servidor de Tasmota

   Como antes se mencionó Tasmota es un firmware alternativo que permite levantar un servidor web liviano, fácil de administrar que permite alojar varios sensores de diversos tipos y configurar su transmisión de datos a través de un protocolo MQTT, en forma local o remota. En la Figura 10, se visualiza el servidor WebUI de Tasmota y sus configuraciones principales para la integración de la Placa LoLin new NodeMCU V3 con los sensores conectados.

   1. **Web Server Tasmota WebUI (Main Menú):** Aquí se puede visualizar los datos leídos de los sensores: AM2301 temperatura, Humedad y punto de rocío, adicionalmente se visualiza el estado de los sensores de movimiento, ventana y humo.
   2. **Configure Module:** Tasmota al ser un firmware gratuito que permite integrar varios módulos de tarjetas integradas de tipo MCU (75) la cuales se acoplan a las necesidades del servicio IoT que se requiere trabajar, para este caso de uso se empleó un módulo Genérico de 18 pines el más recomendable dentro de la familia Tasmota, debido a que permite habilitar todos los pines GPIO (entrada y salidas digitales).
   3. **Configuración MQTT:** Una vez que tenga un agente MQTT funcionando como es el caso del Servidor Mosquitto, se debe configurar a Tasmota para que se comunique con él para la suscripción de los datos enviados por los sensores.

      Vaya a **Configuración -> Configurar otro** y asegúrese de que la casilla **"Habilitar MQTT"** esté marcada.

      Una vez que MQTT está habilitado, se debe configurar usando **Configuración -> Configurar MQTT**, aquí se colocará la IP donde se encuentra publicado el servicio de Mosquitto MQTT, el usuario y clave.

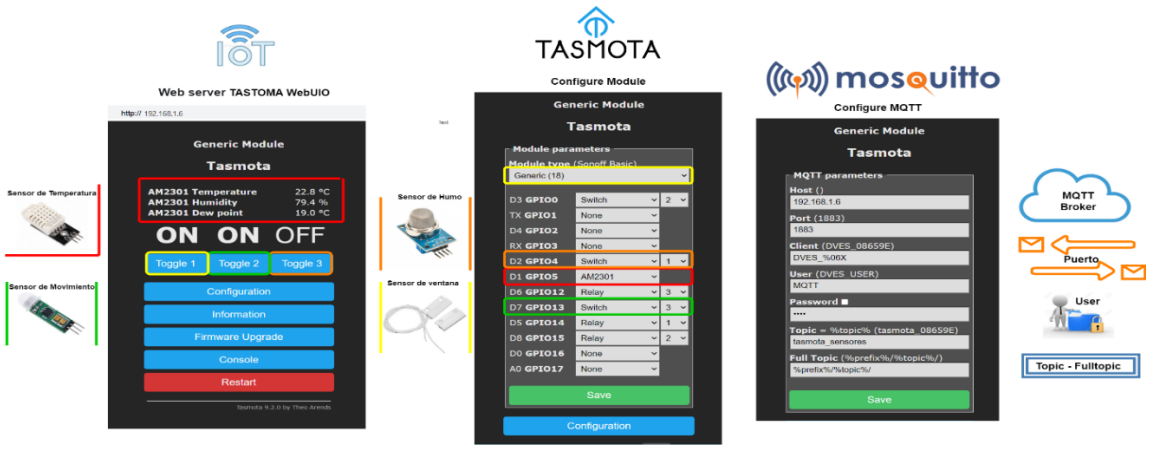


Figura 10. Servidor WebUI Tasmota

Fuente: Investigador

Instalación del Servidor de Mosquitto.

Como ya se habló en la sección: Elementos del Experimento en el Entorno IoT, Mosquitto es un broker para el protocolo MQTT que trabaja bajo el modelo publicación/suscripción y debido a su ligereza al emplear pocos recursos de procesador y memoria en ideal para mensajería de dispositivos inteligentes como IoT.

Para este proyecto se empleó un servidor Mosquitto en Windows, este servidor usa dos clientes desarrollados en C y se los ejecuta por medio de líneas de comandos: **mosquitto_pub** y **mosquitto_sub,** los cuales permiten publicar el servidor Mosquitto y suscribir mensajes.

A continuación, se detalla como configurar el servidor Mosquito.

1. Descargar Servidor Mosquitto
   https://mosquitto.org/download/
2. Ingresar a la carpeta de mosquito mediante la línea de comandos.
   `cd "C:\Program Files\mosquitto"`
3. Levantar el suscriptor
   `mosquitto_sub -h localhost -v -t # -u MQTT -P MQTT`
   Este servicio se levanta en el puerto 1883 no encriptado, y encriptado 1884, para este experimento se decidió correrlo en el puerto 1883, para garantizar la seguridad con BlockChain.

Módulo Python.

"Python es un lenguaje de programación bastante robusto, se trata de un lenguaje de programación multiparadigma, debido a que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional, lo que lo hace un lenguaje fuerte y eficiente el mismo que se ajusta a cualquier necesidad." (Python, 2021).

Este lenguaje permitió desarrollar un módulo integrador entre el entorno IoT y las APIS de servicios diseñadas para la comunicación con BlockChain mediante el protocolo MQTT.

En la Figura 11. se detalla la lógica del módulo Python y como interactúa con el Servidor Mosquitto y la API de Servicios IoT. El módulo desarrollado requiere varias librerías como:

- Paho: que permite detectar a un servidor de MQTT, convirtiendo al módulo Python en un cliente MQTT.
- ReQuest: Permite el acceso a toda la información que pasa desde el navegador del cliente al servidor.
- Python time y date time: permite crear un objeto de fecha, hora y tiempo.

Estas librerías permiten interactuar con el broker MQTT Mosquitto, el mismo que trabaja con canales virtuales llamados tópicos los que permiten identificar el cliente suscriptor (sensores), estos mensajes son publicados y enviados por el módulo Python hacia la API de servicios, esta API contiene dos métodos: GET (obtener) y POST (enviar) estos datos son enviados y recibidos en un objeto JSON y a partir de ellos se llega a la red BlockChain y almacenar los datos en la base de datos de estado CouchDB.

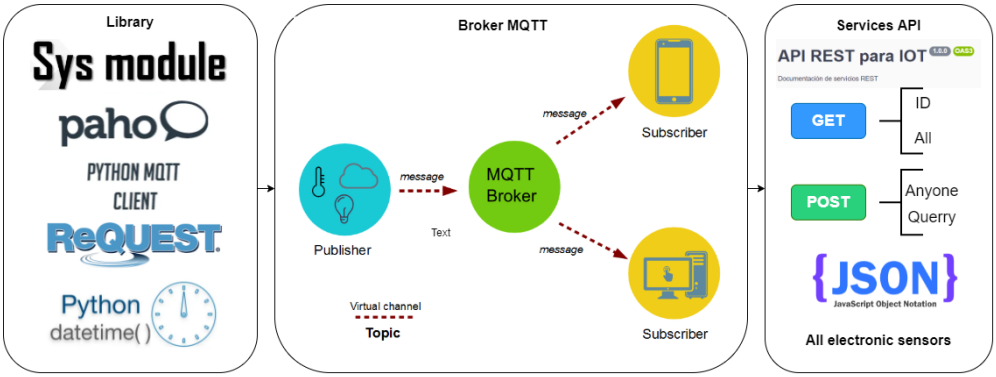


Figura 11. Elementos Módulo Python

Fuente: Investigador

A continuación, se detalla las partes claves del código del módulo Python. Si se desea revisar todo el módulo Python acceder a la ruta: https://gitlab.com/vivianaburgosyar/modulo-python-blockcain.git

- Librerías

```
import paho.mqtt.client as mqtt # libreria MQTT
import sys
import requests # libreria request
import datetime # libreria  fecha
import time # libreria hora
import pymysql #libreria mysql
```

- Conexión broker MQTT: la conexión está formada por la IP del servidor Mosquitto MQTT, el puerto y el topic principal con el que se conecta la placa LoLin new NodeMCU V3.

```
#conexión Broker MQTT
broker_address = "192.168.1.6" #ip servidor Mosquitto
broker_port = 1883
topic = "+/tasmota_sensores/#" # topic principal, placa LoLin
```

- Canales Virtuales o sub tópicos: los sub tópicos permiten identificar cuál es el cliente que envía los datos hacia la red BlockChain.

```
#IDENTIFICACIÓN DE SENSORES CON EL SUB TOPICO
SENSOR1="tele/tasmota_sensores/SENSOR" # sensor temperatura (HUMEDAD, GRADOS CENTIGRADOS, PORCENTAJE%)
SENSOR2="stat/tasmota_sensores/POWER3" # sensor humo (ON, OF)
SENSOR3="stat/tasmota_sensores/POWER2" #sensor ventana (ON, OF)
SENSOR4="stat/tasmota_sensores/POWER1" #sensor de movimiento (ON, OF)
SENSOR5="tele/tasmota_sensores/STATE" #estado de tasmota
```

- Uso de la API Rest IoT para el envío de los datos: se usan los métodos GET y POST de los datos recuperados de la LoLin new NodeMCU V3.

```
response=requests.get('http://35.208.212.182:3001/api/sensor/temperatura/ST1')
lectura_grados_centigrados=lectura.split('"Temperature":')
grados_centigrados=lectura_grados_centigrados[1]
grados_centigrados=grados_centigrados[0:4]
data={"datetime": obtener_fecha(), "location": broker_address, "record": grados_centigrados, "sensorid":idunic , "sensortype"
print (data)
response=requests.post("http://35.208.212.182:3001/api/sensor/temperatura", data=data )
# FIN DE DATOS TEMPERATURA
```

API Rest PARA IoT o API DE SERVICIOS

La API Rest de servicios permite mantener una comunicación entre la red BlockChain y el módulo Python de la IoT, cada sensor se comunica con la red BlockChain gracias a la API de servicios que

interactúa con los miembros de la BlockChain, los usuarios que se registran en la Wallet, una vez autenticados, la API informa que se dese hacer en la BlockChain, los Chaicode o contratos inteligentes son quienes al final insertan o recuperan datos en la red BlockChain.

Par ver la instalación de la API Rest para la IoT ver Anexo B. Instalación API Servicios.

Con la finalidad de tener un entorno mucho más atractivo que permita visualizar los métodos de la API de servicios se ha usado Swagger UI[48] el cual utiliza un documento JSON o YAML y lo hace interactivo, crea una plataforma que ordena cada método (GET, POST) y categoriza las operaciones de la APPI. Cada uno de los métodos es expandible, y en ellos se puede encontrar un listado completo de los parámetros con sus respectivos ejemplos. Incluso podemos probar valores de llamada hacia los 4 métodos elaborados:

1. **GET ID:** obtiene un dato con el id del sensor.
2. **GET ALL:** obtiene una lista de todos los datos almacenados por sensor.
3. **POST:** Inserta un dato en la BlockChain, con un formato especifico de sensor
4. **POST QUERY**: muestra un grupo de datos por un rango fecha.

La API Rest IoT de servicio se muestra gráficamente en la ruta http://35.208.212.182:3001/api-docs/

La Figura 12, muestra la API Rest de IoT y los métodos GET y POST para cada uno de los sensores (humedad, humo, movimiento, punto de rocío, temperatura, ventana).

[48] **Swagger UI:** es un conjunto de herramientas de software de código abierto para diseñar, construir, documentar, y utilizar servicios web RESTful

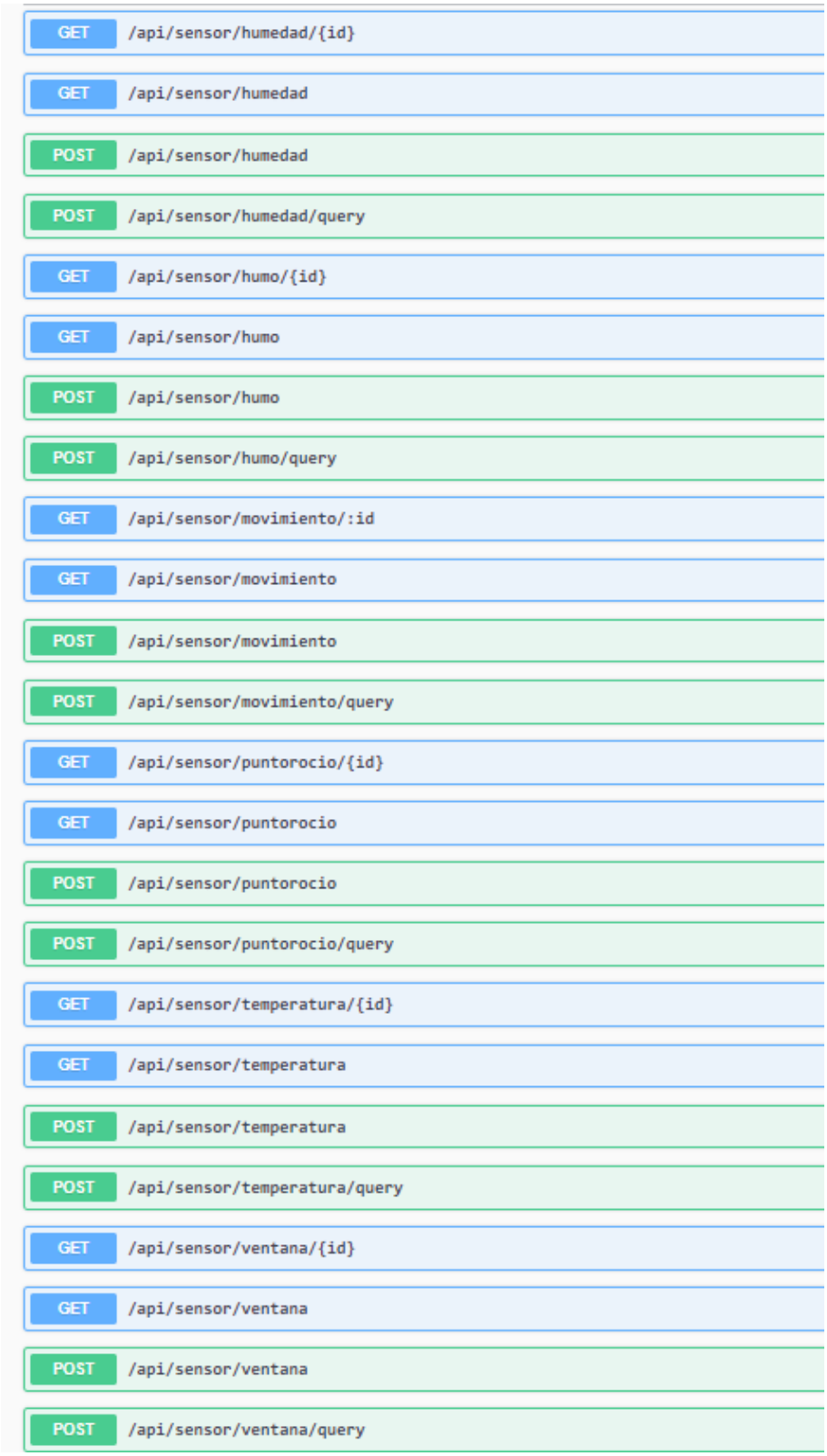


Figura 12. API Rest para IoT con swagger
Fuente: Investigador

RED BLOCKCHAIN HYPERLEDGER FABRIC

Para ver la instalación de la red BlockChain y sus elementos consultar el Anexo A: DESPLEGAR RED HEPERLEDGER FABRIC 2.2.0

Hyperledger Fabric a diferencia de otras plataformas de BlockChain, está autorizada y es una red privada (Rodriguez, 2019). No acepta participantes desconocidos, debido a que cada nodo debe ser autenticado y registrado para hacer sus transacciones, crea una función de membresía conocida como Autoridad Certificadora y permite a aquellos miembros que están autorizados y registrados en la Wallet también conocida como proveedor de servicios de membresía (MSP) el envío de transacciones para un consenso con la red, para este caso de uso como lo indica la Figura 13. La red está compuesta por tres organizaciones que participan en el consenso para el registro de las transacciones en el libro mayor de la red.

- Organización 1 (Org1MSP): tiene permisos de escritura, contiene el bloque génesis[49] del libro mayor y puede ordenar una transacción.
- Organización 2 (Org2MSP): El ordenante no está en estado activo, por lo tanto, esta organización solo participa en el consenso.
- Organización 3 (Org3MSP): tiene permisos de escritura y puede ordenar una transacción.
- Protocolo de consenso La red BlockChain está compuesta por tres organizaciones montada en una red de contenedores Docker y virtualizados por Docker Compose, a continuación, se detallan las URLS más importantes que permiten revisar el funcionamiento de los nodos de la red.

[49] Bloque Génesis: es primer bloque de cada BlockChain.

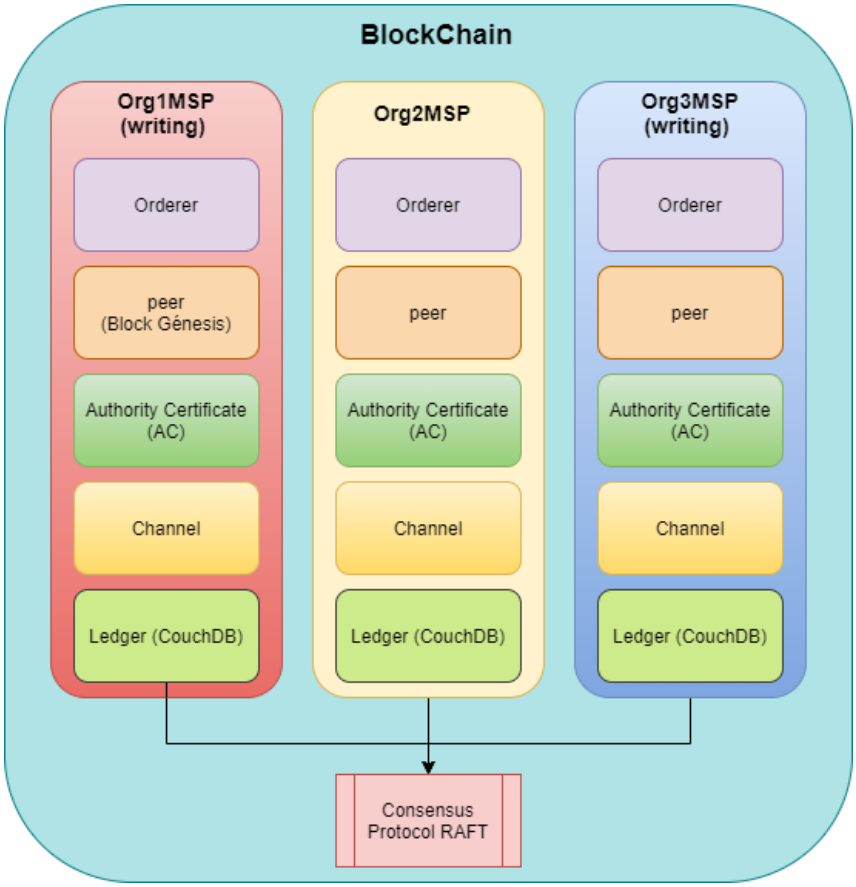


Figura 13. Estructura del contenedor Docker Compose
Fuente: Investigador

a) Administrador de contenedores Dockers:

Docker Compose es una herramienta para diseñar y ejecutar aplicaciones que se virtualizan en contenedores Linux, estos servicios se configuran por medio de un archivo. YAML, el cual guarda todas las instrucciones para iniciar, parar o pausar los servicios.

Para este caso de prueba se levanta la red BlockChain con tres organizaciones y una entidad certificadora, esta virtualización de ambientes separados en un mismo host será la red BlockChain que permitirá transaccionar con los datos de los sensores. La tabla 5. Contiene la URL de Docker Compose para acceder a su panel y verificar los componentes de la red BlockChain.

Tabla 3. URL de administración de los contenedores de la red BlockChain

URL:	http://35.208.212.182:9000/
Usuario:	Admin
Clave:	adminadmin

Fuente: Investigador

La Figura 13. detalla el número de redes virtualizadas, los volúmenes levantados y el número de contenedores empleados para este experimento, cada contenedor es un ambiente distinto.

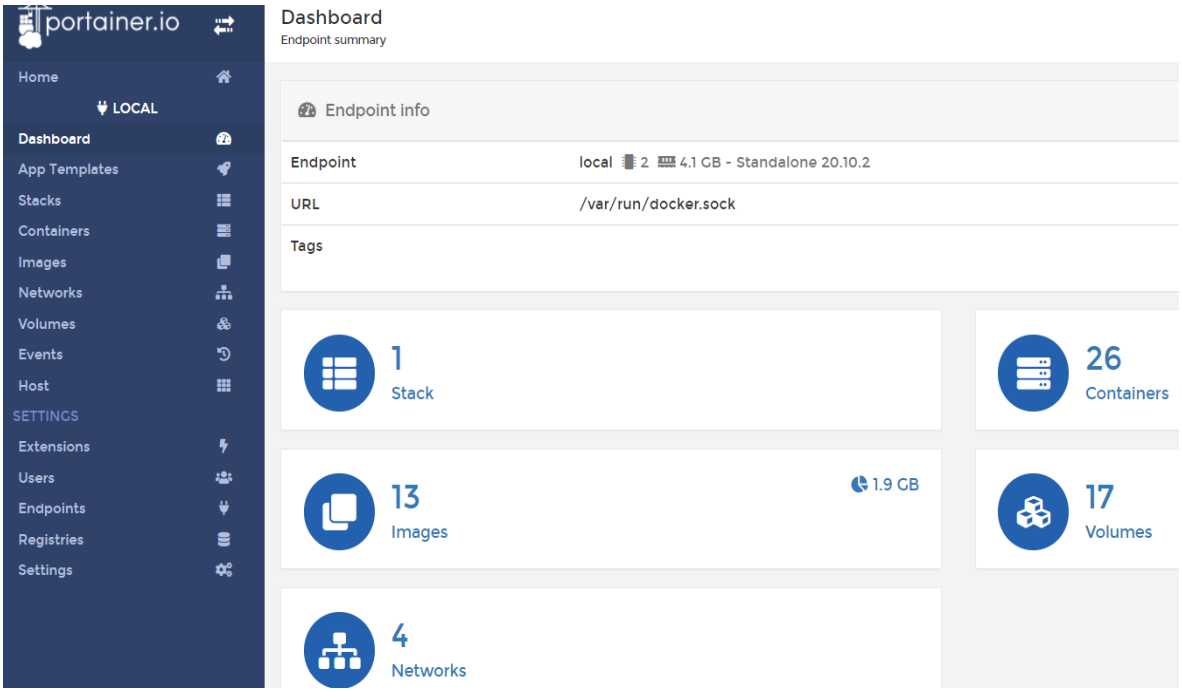


Figura 14. Estructura del contenedor Docker Compose

Fuente: Investigador

La Figura 14. muestra visualmente el inventario de la red BlockChain. Para comprender la estructura de Docker Compose basta con conocer los elementos de la pila o Stack, en ella se encuentran los compontes más importantes de la red BlockChain: clientes (organizaciones que la componen), la base de datos de estado CouchDB (almacena el libro mayor o Ledger) esta base de datos está presente en cada una de las organizaciones participantes de la red, una autoridad certificadora intermedia, ordenantes para el envío de transacciones. Los canales se aseguran por medio de un certificado intermedio y el certificado raíz. (Ver sección: Elementos del Entorno BlockChain).

Figura 15. Stack Docker Compose

Fuente: Investigador

b) Consenso BlockChain

Para este experimento la red BlockChain fue construida con tres organizaciones las mismas que interactúan entre sí, mediante los Chaincode y los protocolos de consenso.

El consenso en Hyperledger Fabric está compuesto de tres fases:

- Aprobación
- Orden
- Validación

Las transacciones enviadas a la red BlockChain sigue los siguientes pasos (Rodríguez, 2021):

- Un cliente hace la petición para ejecutar una transacción. Estos clientes están registrados en la Wallet, si algún peticionario que no está registrado solicita una transacción, esta es rechazada.
- La aplicación SDK de Hyperledger generará una propuesta transaccional y la réplica para todos los nodos participantes de la red BlockChain.
- Los nodos miembros de la red verifican si la transacción enviada sigue el formato estándar, contiene la firma del cliente, tiene las características necesarias y si es válida o no.
- Luego de validar la transacción, se pasa al contrato inteligente o Chaincode u otras aplicaciones SDK.
- Luego, la API verificará las firmas y comparará las respuestas de la propuesta.
- Posterior a ello se transmitirá la transacción con conjuntos de escritura / lectura y firmas a los nodos que ordenan.
- Los nodos que ordenan preparan el bloque de una forma ordenada y matemática para luego ser trasmitido a todos los miembros.
- Luego los pares obtienen el bloque actualizado agregado a la cadena, y el bloque también se agrega a la base de datos de estado CouchDB.

Luego de este proceso el libro mayor de la BlockChain se actualizará, para este experimento y validar lo mencionado en la Tabla 6. Contiene los accesos al panel de CouchDB de las tres organizaciones que forman parte de la red BlockChain, las mismas que gracias a la lógica de la red son idénticas y cuando se procesa una transacción se replican. Ver sección: RED BLOCKCHAIN HYPERLEDGER FABRIC, Ver Figura 14, en estas secciones se explica la estructura de la red y como CouchDB es parte de las organizaciones participantes.

Tabla 4. URL's de administración de la base de datos CouchDB BlockChain

BASES DE DATOS DE LA RED BLOCKCHAIN			
	Primera Organización	**Segunda Organización**	**Tercera Organización**
URL:	http://35.208.212.182:5984/_utils/	http://35.208.212.182:5985/_utils/	http://35.208.212.182:5986/_utils/
Usuario:	admin	admin	admin
Clave:	adminpw	adminpw	adminpw

Fuente: Investigador

CouchDB proporciona una ventaja interesante, tiene una arquitectura distribuida con replicación, esta base de datos se ajusta perfectamente a BlockChain pues permite que todas las organizaciones participantes en el consenso una vez aceptada una transacción puedan ser sincronizadas de un

modo automático y simple, de tal manera que cada organización que forma parte de la red BlockChain cuando una transacción es almacenada todas las bases de datos son replicadas. (CouchDB_Org, 2021). En la Figura 16. se muestra el panel de administración de CouchDB y sus elementos, este panel contiene el acceso a los datos almacenados en la BlockChain, los usuarios y la forma de replicar la base de datos.

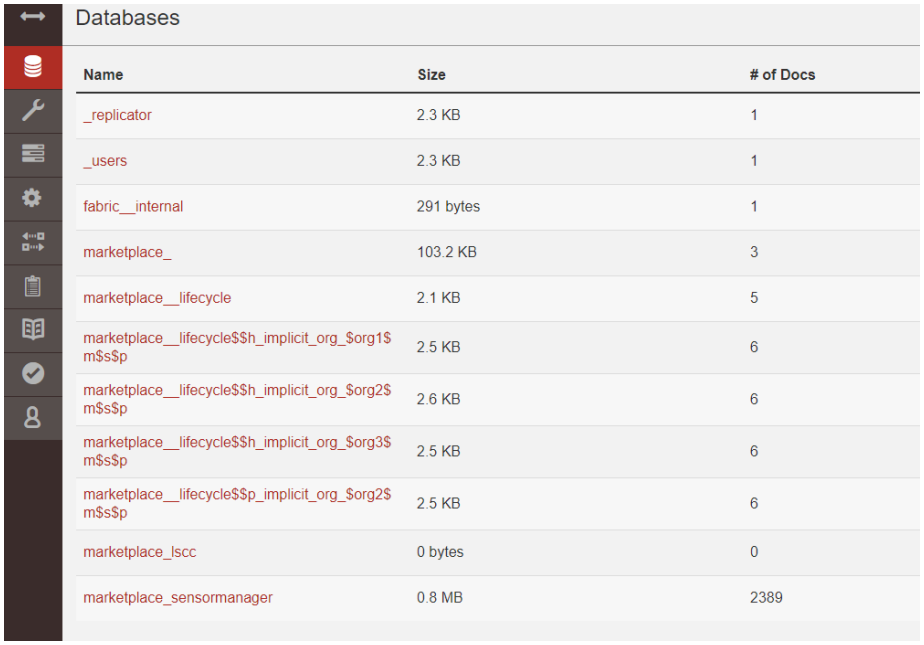


Figura 16. CouchDB de la BlockChain

Fuente: Investigador

FRONT – END PARA DISPOSITIVO MÓVIL

Para este experimento se ha diseñado un Front – End con WordPress (ANEXO 3: Instalación WordPress) que es totalmente compatible con dispositivos móviles.

Con la finalidad de interactuar entre la BlockChain y la API de servicios para la recuperación de datos se programó un plugin compatible con WordPress para mostrar los datos de los sensores que se almacenan en la red BlockChain.

El código fuente del plugin se encuentra en el siguiente link: https://gitlab.com/vivianaburgosyar/IoT-front-end.git

Para acceder al Front – End se lo puede hacerlo con los siguientes datos detallados en la Tabla 7.

Tabla 5. Accesos Front - End

FRONT - END	
URL:	http://35.208.212.182/wp-login.php
Usuario:	admin
Clave:	AdMin

Fuente: Investigator

El Front – End para este experimento permite interactuar entre la API Rest de Servicios IoT y la BlockChain protegiendo la capa de aplicación para evitar la sustracción de los datos pasando por un canal seguro usando el protocolo gRPC , éste protocolo se define como un tipo de framework, es un sistema moderno que llama procedimientos remotos que procesan la comunicación en estructuras cliente-servidor distribuidas o peer-to-peer desde la capa de aplicación de una manera eficiente gracias a una innovadora ingeniería de procesos (IONOS, 2020).

Un elemento característico de la comunicación entre procesos mediante gRPC es el principio de transparencia: la colaboración entre instancias distanciadas (protocolo multilingüe debido a que está basado en Protocol Buffer[50]) es tan estrecha y sencilla que no se percibe ninguna diferencia en comparación con una comunicación local entre procesos internos de una máquina virtual lejana. (Gesior, 2020). Además, es un protocolo multilingüe permitiendo trabajar con varios lenguajes de desarrollo.

Este protocolo fue diseñado y desarrollado por Google en el año 2015, hoy es la Cloud Native Computing Foundation la encargada de su distribución y desarrollo, empleada por varios proyectos de código abierto como Hyperledger Fabric (IONOS, 2020).

El "Protocol Buffer" sirve como el formato de intercambio de mensajes que determina las estructuras, los tipos y los objetos de los mensajes. Son los elementos que hacen que el cliente y el servidor se "entiendan" y funcione de una manera eficiente incluso a grandes distancias.

Para el transporte de ida y vuelta de los flujos de datos entre máquinas (peer to peer) (Proto Request y Proto Response) se integra HTTP/2 en protocolos de red específicos como TCP/IP o UDP/IP. Los flujos transmiten datos binarios compactos que surgen en la

[50] **Protocol Buffer:** biblioteca multiplataforma gratuita y de código abierto que se utiliza para serializar datos estructurados. Es útil en el desarrollo de programas para comunicarse entre sí a través de una red o para almacenar datos.

serialización (Marshaling[51]) típica de las llamadas a procedimientos remotos. Para procesar estas estructuras de datos, totalmente abstractas, en los pasos siguientes en el lado del servidor y en el lado del cliente, el contenido transmitido se vuelve a deserializar[52] (unmarshalling) (IONOS, 2020).

Este procedimiento hace que el envío de los datos desde el dispositivo IoT se cifre de una manera segura pasando por varios entornos que permiten cifrar lo datos tanto en el momento del almacenamiento como en el momento de la recuperación.

Ver Anexo D., para conocer más sobre cómo funciona el Front – End.

RESUMEN DE TRABAJOS

En Tabla 2, se presenta un resumen varios estudios desarrollados con IoT y BlockChain los cuales fueron revisados y analizados para este estudio, en la tabla que manifiestan detalles de la tecnología BlockChain empleada en varias industria de IOT. Para cada caso de estudio, se discute el problema abordado y la solución propuesta, también se colocan algunos temas técnicos como los algoritmos empleados.

Tabla 6. Resumen de Estudos Similares

Ref.	Problema Abordado	Solución Propuesta	Consenso Empleado	Usa Smart Contract?	Platafor ma	Datos Transaccionales	Tipo de evaluación
(XIA et al., 2017)	Minimización de Riesgos en la privacidad	Smart Contract con blocques	DNA*	SI	BlockChain privada	El lector de datos identifica, solicita, identifica, sensibilidad de los datos	Número de solicitantes activos entre 5 y 100 usuarios, JMeter, rendimiento
(Institute, n.d.)	Integridad de los datos de las cadenas de suministro farmacéuticas	IoT con Smart Contracts	PoC/ proof-of-stake	SI	Ethereum	Datos de Temperatura, humedad y numeros seriales	Evaluación Teorico Practica
(Homayoun et al., 2019)	Intercambio de datos	BlockChain con estándares	PoC	SI	Ethereum	Transacción comercial, actividad	Funcionamiento de los Contracts

[51] **Marshaling**: es el proceso de transformación de la representación de datos a un formato adecuado, realizado según unas normas específicas y que su objetivo más primordial es la transferencia de estos a través de la red

[52]**Deserializar:** Volver un objeto a su estado natural.

	entre empresas con seguridad e integridad	de la industria para el intercambio de datos (GS1, IOT, etc)				física o digital en la cadena de suministro de activos	
(Mellek & Kaya, 2020)	Fallo de un solo punto y ataque malicioso en el sistema IoT	Gráfico acíclico dirigido DAG estructurado BlockChain	Creditos basados en ocnsenso Mechanico	NO	Consorci' on BlockChaim	Lectura de sensores	Evaluación de rendimiento Aplicación. Tiempo de respuesta
(Wickström et al., 2020)	A Protocol Based on BlockChain Smart Contracts for Secure and Automated IoT Deployments	Smart Contract Deterministas para asegurar el ciclo de vida del dispositivo IoT	proof-of-stake	SI	Ethereum	Utilizan la tecnología de bloques basados en criptografía, para asegurar que las redes distribuidas de IoT se mantengan seguras y fiables a lo largo de su ciclo de vida	Evaluación de Rendimiento
(Andersen John Kolb Kaifei Chen Gabriel Fierro David E Culler Raluca Ada Popa et al., 2017)	Inseguridad en los datos transmitidos en IoT	presentan un sistema de autorización basado en BlockChain llamado WAVE (Wide Area Verifi ed Exchange) para dispositivos IoT	proof-of-stake	SI	Ethereum	Se transmite los datos de dispositivos IoT, servidores y puertas de enlace en la capa de superposición.	Evaluación Teórica
Fuente: (Alladi Tejasvi et al., 2019; Wickström et al., 2020)							

PRUEBAS

Con la finalidad de probar que este experimento cumpla con el objetivo planteado: "Desarrollar un cifrado de datos usando BlockChain como tecnología de convergencia para dispositivos móviles asociados con IoT, en la capa de aplicación del modelo de capas de IoT"

Aplicando la investigación inductiva, experimental se realizó dos pruebas para luego obtener un razonamiento que lleva a conclusiones que parten de los resultados encontrados como válidos. Los resultados se obtienen basados sus premisas: 1) observación, 2) formulación de hipótesis, 3) verificación, 4) tesis, 5) ley y teoría.

Ambiente Inseguro.

Este un ambiente convencional que se usa en la mayoría de los casos en un proyecto de desarrollo, por una parte, se tiene al dispositivo IoT que lee y envía datos y por otra parte se tiene una Base de Datos que recibe transacciones la misma que se puede acceder con un usuario y una clave. La Figura 16, muestra el dispositivo IoT ensamblado en la LoLin New NodeMCU V3, este es un ambiente convencional que cifra los datos únicamente con el servidor Mosquitto MQTT. Mediante un módulo Python se envía los datos leídos desde el dispositivo IoT directamente a una base de datos MySQL, los cuales pueden ser modificados en el momento del envío o directamente desde la base de datos de forma manual sin tener ninguna alerta.

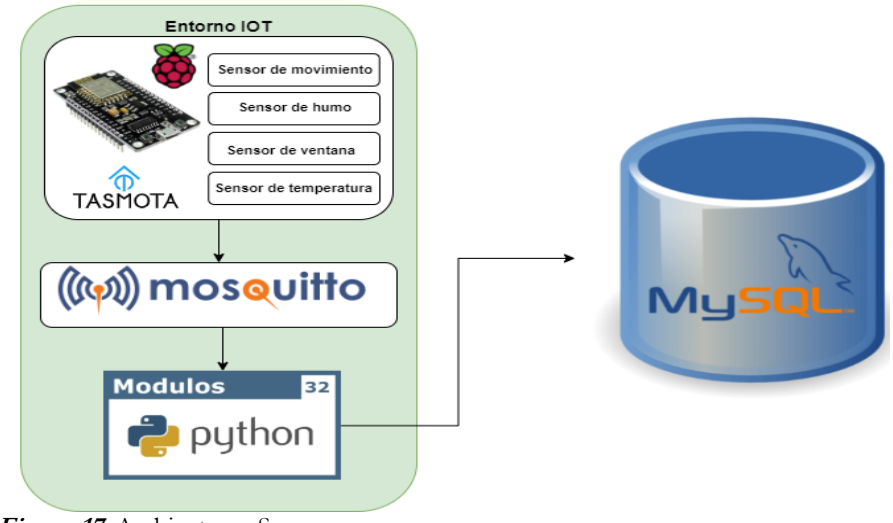


Figura 17. Ambiente no Seguro

Fuente: Investigador

Ambiente Seguro.

Este ambiente ha sido el motivo de la investigación durante todo este documento, las pruebas que se ejecutaron, se las realizará en la base de datos de estado de la red BlockChain: CouchDB que

almacena las transacciones que han sido aceptadas por los miembros de la red , esta base de datos es parte de la red, la misma que no dejará insertar datos si no es validada la transacción por la autoridad certificadora y si a más de ellos es un cliente registrados en la Wallet, en el caso de que deje insertar datos cuando la BlockChain compara los libros mayores de la red, nota una diferencia y el dato encontrado como diferente es rechazado.

Para comprobar esta teoría se harán dos pruebas:

1. Panel de CouchDB: por medio del panel de administración de CouchDB se intentará modificar, eliminar, e insertar un dato.
2. Módulo Python: se intentará insertar, modificar y eliminar un dato hacia la base de datos de estado CouchDB.

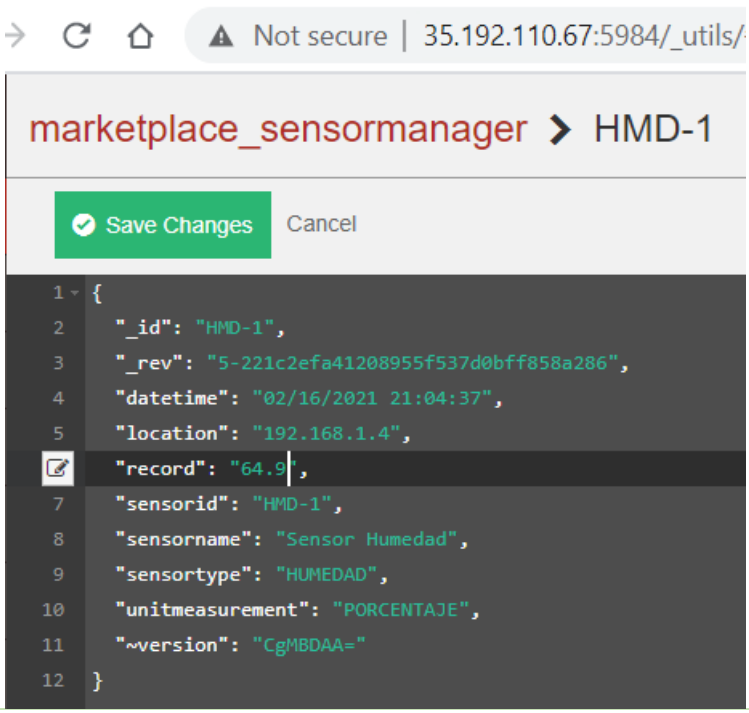
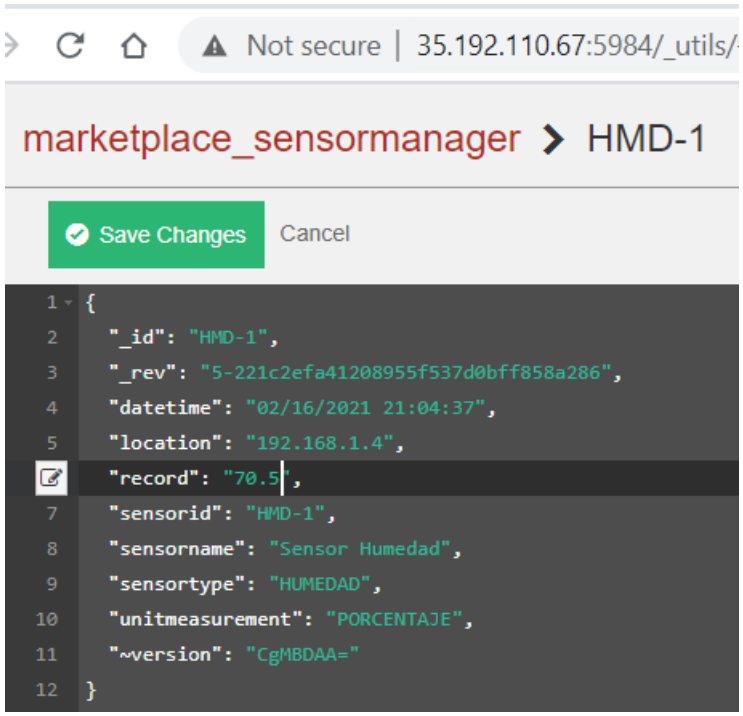
PRUEBA UNO:

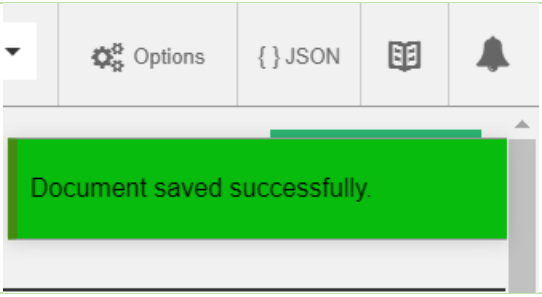
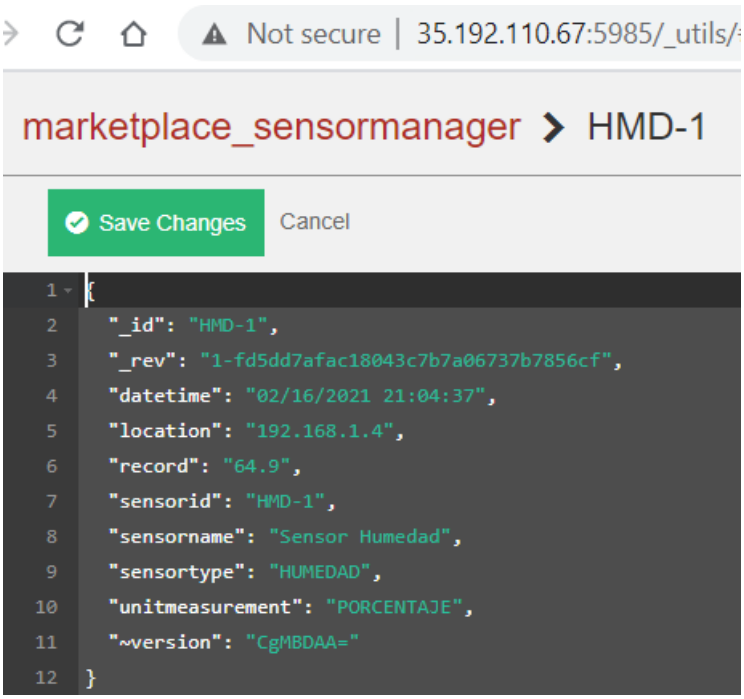
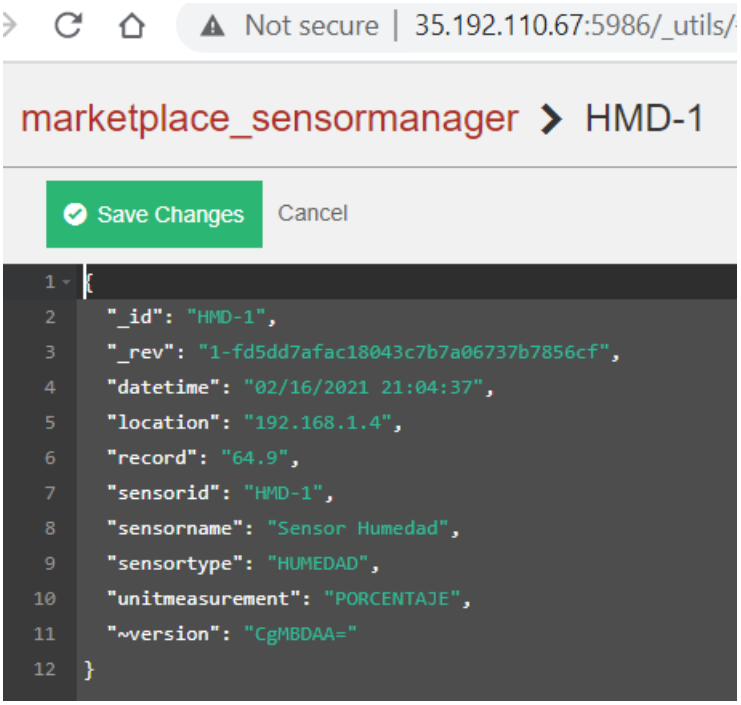
INSERTAR DATOS DESDE EL FRONTAL DE COUCHDB.

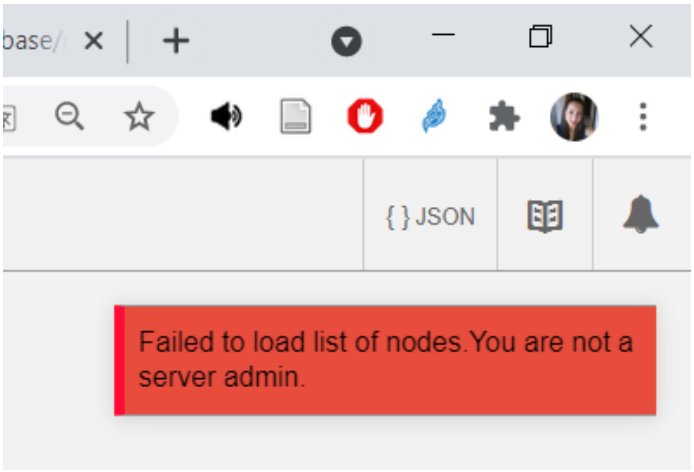
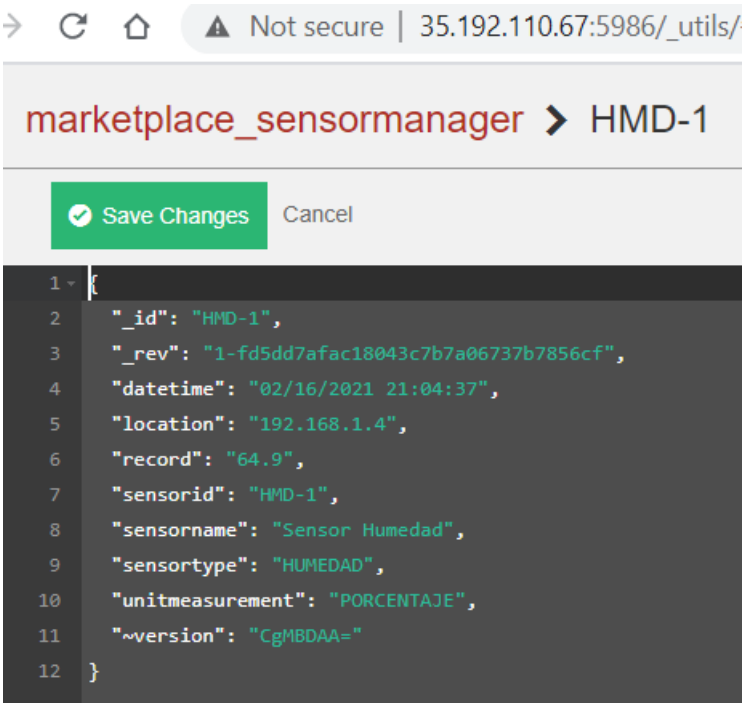
PRIMERA ORGANIZACIÓN: http://35.192.110.67:5984/_utils/

Tabla 7. Resultados de la Prueba uno.

DESCRIPCIÓN	ACCIÓN Y RESULTADO
Base de datos: **Marketplace_sensormanager**	Databases Name / Size / # of Docs _replicator — 2.3 KB — 1 _users — 2.3 KB — 1 fabric__internal — 291 bytes — 1 marketplace_ — 103.1 KB — 3 marketplace__lifecycle — 2.1 KB — 5 marketplace__lifecycle$$h_implicit_org_$org1$msp — 2.5 KB — 6 marketplace__lifecycle$$h_implicit_org_$org2$msp — 2.6 KB — 6 marketplace__lifecycle$$h_implicit_org_$org3$msp — 2.5 KB — 6 marketplace__lifecycle$$p_implicit_org_$org1$msp — 2.5 KB — 6 marketplace_lscc — 0 bytes — 0 marketplace_sensormanager — 0.8 MB — 2388

Dato: HMD-1	
Cambio: record 70.5	
Resultado: Guardar	**Document saved successfully**

	
Vista de la base de Datos de la Organización dos	
Vista de Base de Datos de la Organización tres	

Regresamos a la Base de Datos de la Organización uno	Cuando volvemos a la base de datos de CouchDB de la organización uno, se produce un error de sincronización, ya que se ha producido un fallo de consenso, la base de datos demorará unos minutos en sincronizar y Restaurará nuevamente los valores debido a que no hay un consenso entre las otras organizaciónes para ese valor, por lo tanto lo detacta como un intruzo y una falla de seguridad para desechar el dato modificado. 
Regresamos a los datos de la Organización uno	

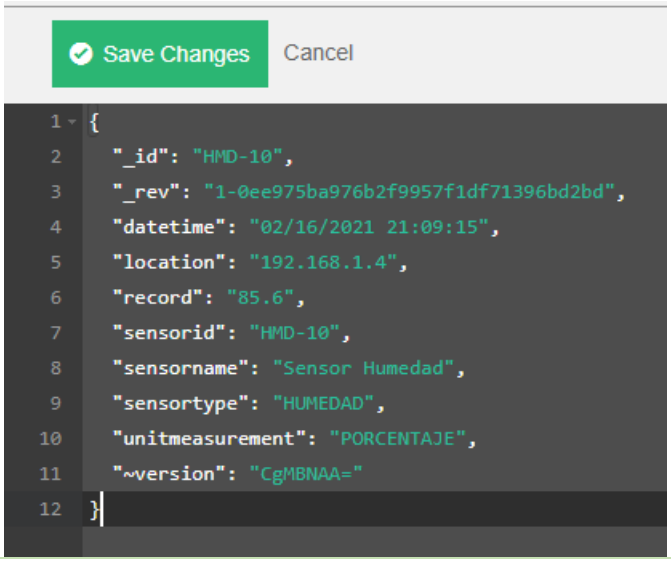
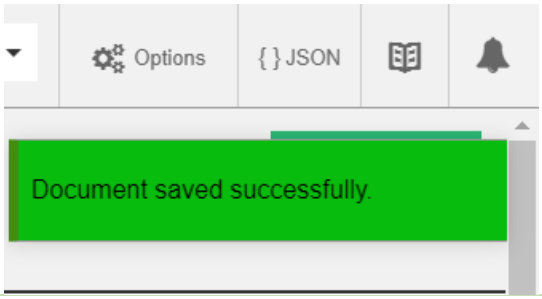
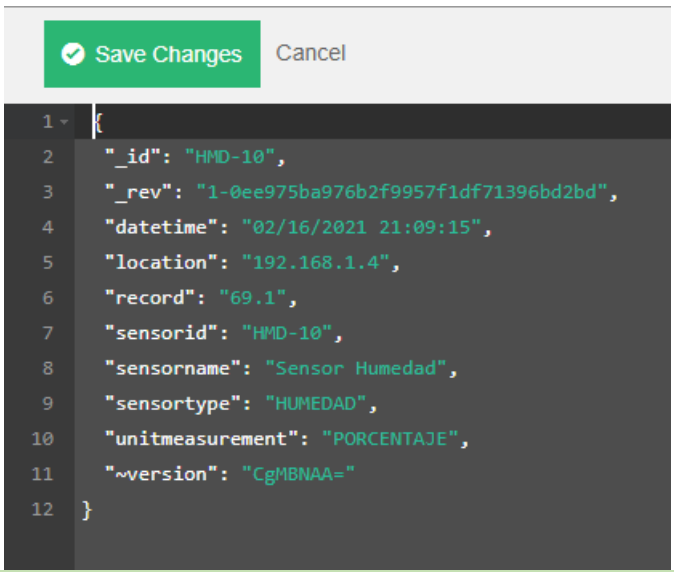
Fuente: Investigador

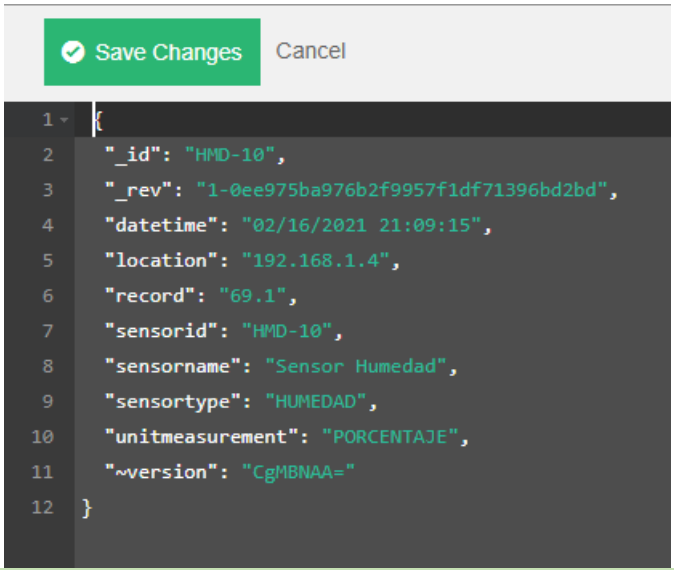
SEGUNDA ORGANIZACIÓN: http://35.192.110.67:5985/_utils/

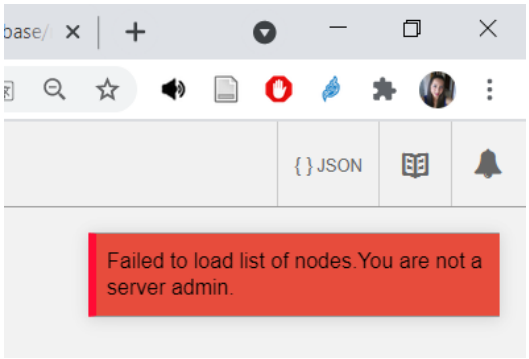
Tabla 8. Resultados de la Prueba uno, organización dos

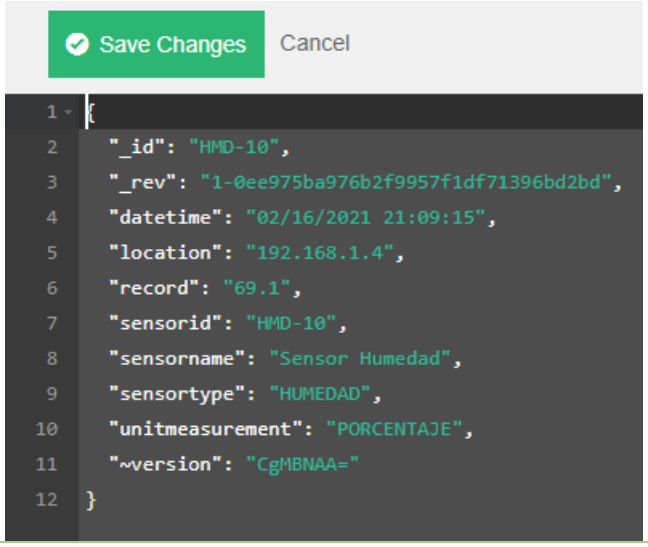
DESCRIPCIÓN	ACCIÓN Y RESULTADO
Base de datos: **Marketplace_sensormanager**	Databases Name / Size / # of _replicator — 2.3 KB — 1 _users — 2.3 KB — 1 fabric__internal — 291 bytes — 1 marketplace_ — 103.1 KB — 3 marketplace__lifecycle — 2.1 KB — 5 marketplace__lifecycle$$h_implicit_org_$org1$msp — 2.5 KB — 6 marketplace__lifecycle$$h_implicit_org_$org2$msp — 2.6 KB — 6 marketplace__lifecycle$$h_implicit_org_$org3$msp — 2.5 KB — 6 marketplace__lifecycle$$p_implicit_org_$org1$msp — 2.5 KB — 6 marketplace_lscc — 0 bytes — 0 marketplace_sensormanager — 0.8 MB — 2388

Dato: HMD-10	Save Changes Cancel
	
	`{`
	`"_id": "HMD-10",`
	`"_rev": "1-0ee975ba976b2f9957f1df71396bd2bd",`
	`"datetime": "02/16/2021 21:09:15",`
	`"location": "192.168.1.4",`
	`"record": "69.1",`
	`"sensorid": "HMD-10",`
	`"sensorname": "Sensor Humedad",`
	`"sensortype": "HUMEDAD",`
	`"unitmeasurement": "PORCENTAJE",`
	`"~version": "CgMBNAA="`
	`}`

Cambio: record 85.6	
Resultado: Guardar	**Document saved successfully** 
Vista de la base de Datos de la Organización tres	

Vista de Base de Datos de la Organización tres	

Regresamos a la Base de Datos de la Organización dos	Cuando volvemos a la base de datos de CouchDB de la organización dos, se produce un error de sincronización, ya que se ha producido un fallo de consenso, la base de datos demorará unos minutos en sincronizar y Restaurará nuevamente los valores debido a que no hay un consenso entre las otras organizaciónes para ese valor, por lo tanto lo detacta como un intruzo y lo desecha.
	 
	// 20210510211613
	// http://35.192.110.67:5985/marketplace_sensormanager/_all
	{
	"error": "unauthorized",
	"reason": "You are not authorized to access this db."
	}

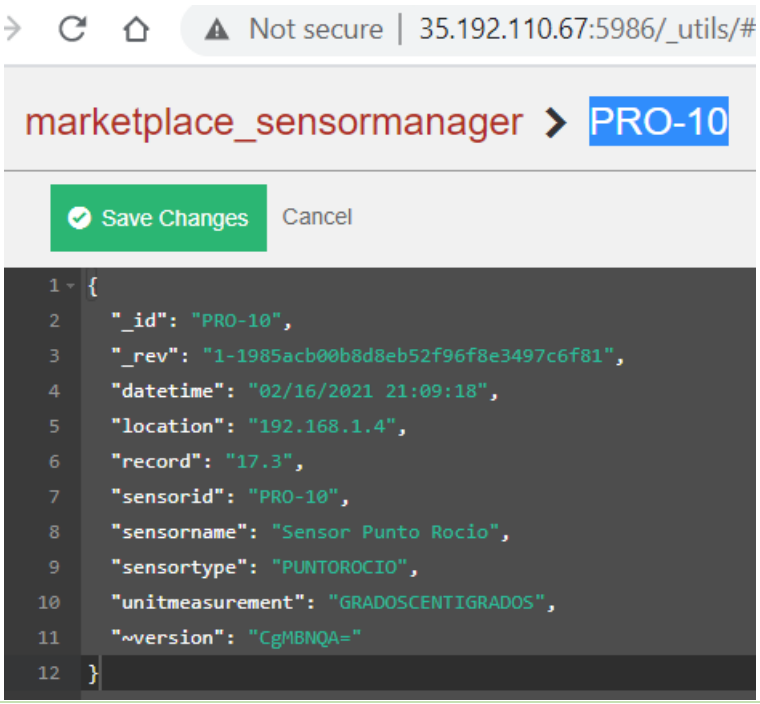
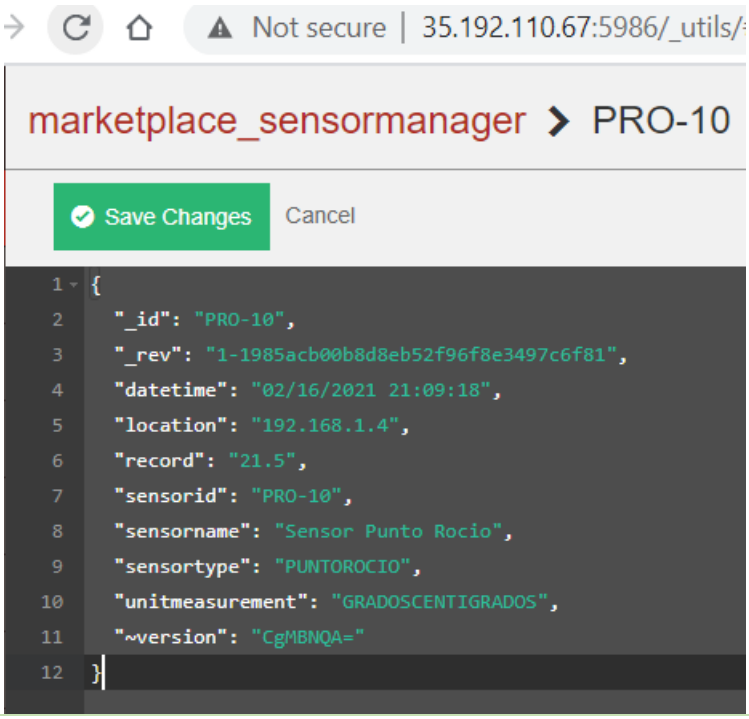
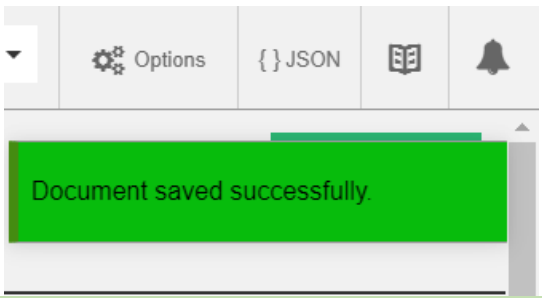
Regresamos a los datos de la Organización dos	

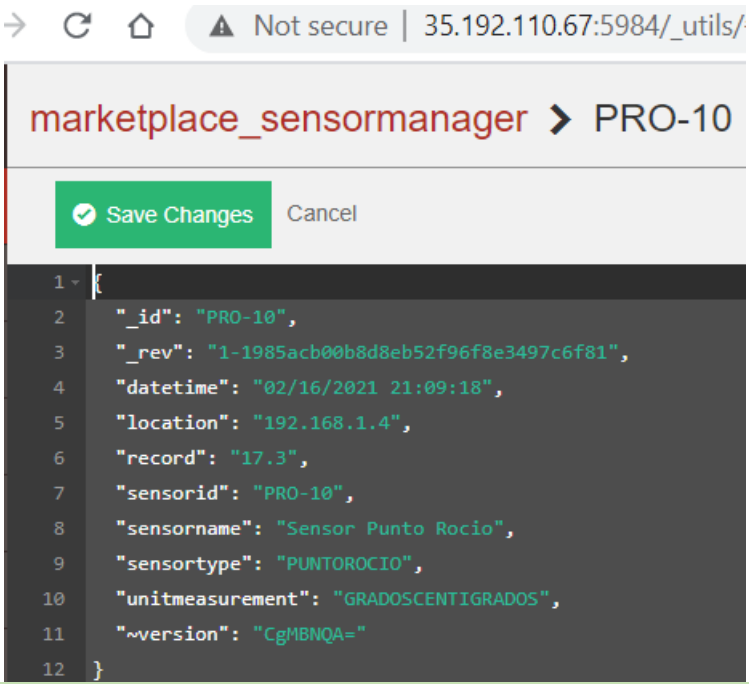
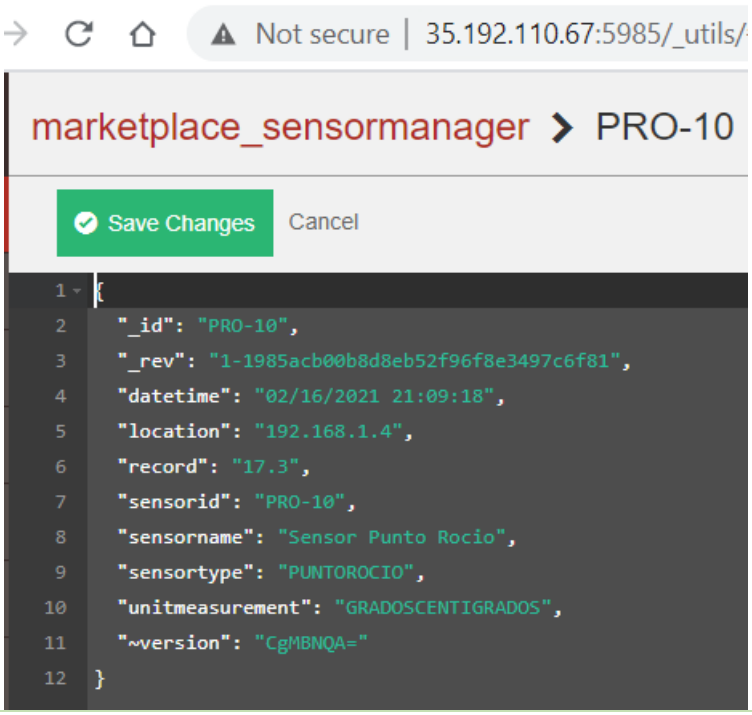
Fuente: Investigador

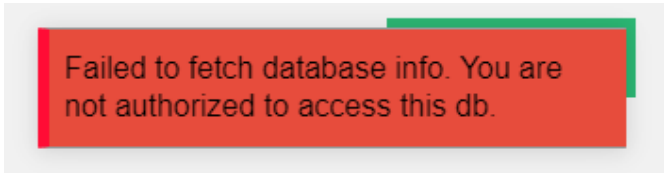
TERCERA ORGANIZACIÓN: http://35.192.110.67:5986/_utils

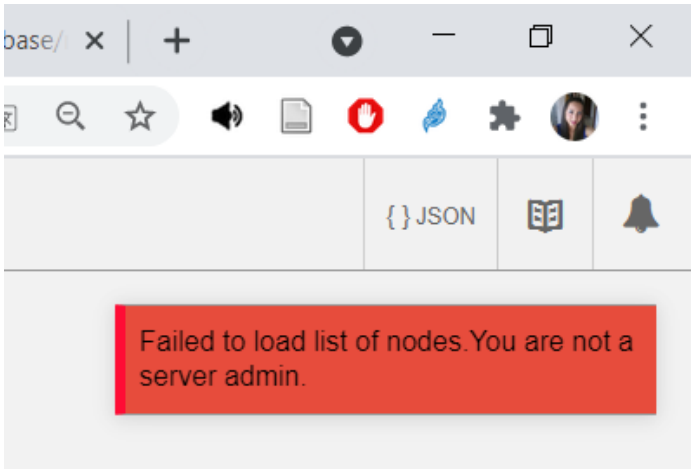
Tabla 9. Resultados de la Prueba uno, organización tres

DESCRIPCIÓN	ACCIÓN Y RESULTADO
Base de datos: **Marketplace_sensormanager**	Databases Name / Size / # of Docs _replicator — 2.3 KB — 1 _users — 2.3 KB — 1 fabric__internal — 291 bytes — 1 marketplace_ — 103.1 KB — 3 marketplace__lifecycle — 2.1 KB — 5 marketplace__lifecycle$$h_implicit_org_$org1$msp — 2.5 KB — 6 marketplace__lifecycle$$h_implicit_org_$org2$msp — 2.6 KB — 6 marketplace__lifecycle$$h_implicit_org_$org3$msp — 2.5 KB — 6 marketplace__lifecycle$$p_implicit_org_$org1$msp — 2.5 KB — 6 marketplace_lscc — 0 bytes — 0 marketplace_sensormanager — 0.8 MB — 2388

Dato: PRO-10	
Cambio: record 21.5	
Resultado: Guardar	**Document saved successfully** 

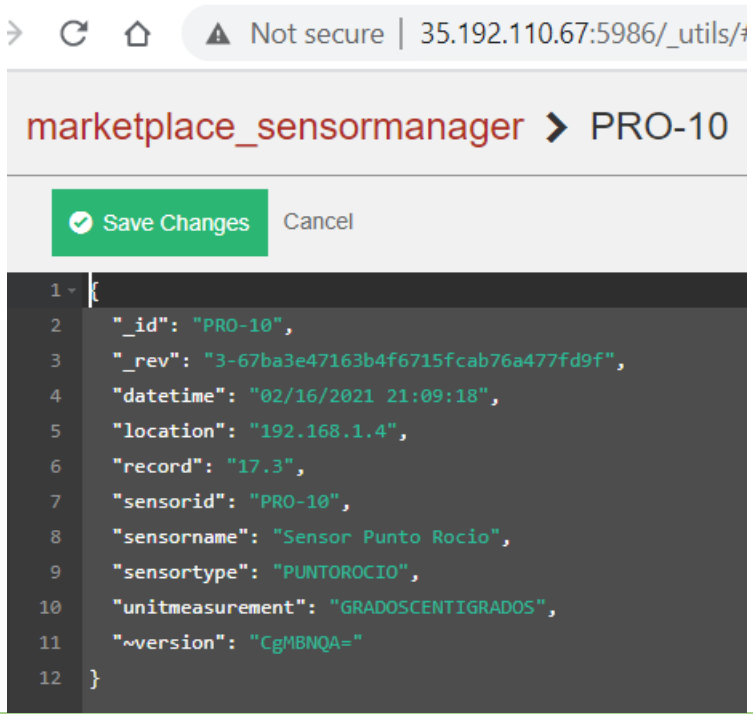
Vista de la base de Datos de la Organización uno	
Vista de Base de Datos de la Organización dos	
Regresamos a la Base de Datos de la Organización tres	Cuando volvemos a la base de datos de CouchDB de la organización dos, se produce un error de sincronización, ya que se ha producido un fallo de consenso, la base de datos demorará unos minutos en sincronizar y Restaurará nuevamente los valores debido a que no hay un consenso entre las otras organizaciones para ese valor, por lo tanto, lo detecta como un intruso y lo desecha.






```
// 20210510210905
// http://35.192.110.67:5986/_all_dbs

{
  "error": "unauthorized",
  "reason": "You are not a server admin."
}
```

Regresamos a los datos de la Organización tres




```
{
  "_id": "PRO-10",
  "_rev": "3-67ba3e47163b4f6715fcab76a477fd9f",
  "datetime": "02/16/2021 21:09:18",
  "location": "192.168.1.4",
  "record": "17.3",
  "sensorid": "PRO-10",
  "sensorname": "Sensor Punto Rocio",
  "sensortype": "PUNTOROCIO",
  "unitmeasurement": "GRADOSCENTIGRADOS",
  "~version": "CgMBNQA="
}
```

Fuente: Investigador

PRUEBA NUMERO DOS:

INYECCIÓN DE CÓDIGO HACIA COUCHDB.

Con la finalidad de ejecutar una nueva prueba y evidenciar que la base de datos de BlockChain no es modificable haremos una inyección de código hacia la Base de Datos de las tres organizaciones participantes en la red, esta inyección de código la haremos con Python y la librería CouchDB.

Esta prueba tiene la finalidad de recuperar un id de la Base de Datos de CouchDB, que pertenece a la red BlockChain, este id engloba a un documento que se encuentra en formato JSON, una vez obtenido el documento se lo alterará modificando datos para posteriormente hacer el update en la base de datos CouchDB.


```
{
  "_id": "HMD-1",
  "_rev": "1-fd5dd7afac18043c7b7a06737b7856cf",
  "datetime": "02/16/2021 21:04:37",
  "location": "192.168.1.4",
  "record": "64.9",
  "sensorid": "HMD-1",
  "sensorname": "Sensor Humedad",
  "sensortype": "HUMEDAD",
  "unitmeasurement": "PORCENTAJE",
  "~version": "CgMBDAA="
}
```


Figura 18. Formato de JSON, que se almacena en CouchDB
Fuenta: Investigador

El código Python empleado para esta prueba se detalla a continuación.

```
15      if dbname in couch:
16          db = couch[dbname]
17          print(db)
18          ## obtener un dato desde couchdb
19          doc_id = "TEM-266"
20          doc = db [ doc_id ]
21          print (doc)
22          # datos
23          try:
24              doc1 ={
25    "_id": "TEM-266",
26    "_rev": "1-35c4edf3793a3a823f58a7569af86b9d",
27    "datetime": "02/17/2021 00:41:16",
28    "location": "192.168.1.6",
29    "record": "22.9",
30    "sensorid": "TEM-266",
31    "sensorname": "Sensor Temperatura",
32    "sensortype": "TEMPERATURA",
33    "unitmeasurement": "GRADOSCENTIGRADOS",
34    "~version": "CgQCA/YA"
```

Figura 19. Código Fuente - inyección de código
Fuente: Investigador

Con el código anterior se ejecutará varias pruebas para borrar datos (delete), modificar datos (update), e insertar (save), ninguna de las acciones se ejecuta con éxito, debido a que la Base de datos de CouchDB, espera que un grupo de organizaciones garanticen el consenso y validen los datos que quieren ser almacenados, rechazando los cambios, esta prueba se realiza en las tres bases de datos de las organizaciones dando el mismo resultado.

Organización 1: http://35.208.212.182:5984/_utils/

```
ython 3.8.3 (default, Jul  2 2020, 17:30:36) [MSC v.1916 64 bit (AMD64)]
ype "copyright", "credits" or "license" for more information.

Python 7.16.1 -- An enhanced Interactive Python.

n [1]: runfile('C:/Users/vivia/OneDrive/Documentos/Viviana/TESIS/APLICATIVO/python/
houcha_db_prueba.py', wdir='C:/Users/vivia/OneDrive/Documentos/Viviana/TESIS/APLICATIVO/python')
onexion exitosa
Database 'marketplace_sensormanager'>
Document 'TEM-266'@'1-35c4edf3793a3a823f58a7569af86b9d' {'datetime': '02/17/2021 00:41:16', 'location':
192.168.1.6', 'record': '22.9', 'sensorid': 'TEM-266', 'sensorname': 'Sensor Temperatura', 'sensortype':
TEMPERATURA', 'unitmeasurement': 'GRADOSCENTIGRADOS', '~version': 'CgQCA/YA'}>
lo se pudo insertar los datos  'Database' object has no attribute 'put'
```

Organización 2: http://35.208.212.182:5984:5985/_utils/

```
ython 3.8.3 (default, Jul  2 2020, 17:30:36) [MSC v.1916 64 bit (AMD64)]
ype "copyright", "credits" or "license" for more information.

Python 7.16.1 -- An enhanced Interactive Python.

n [1]: runfile('C:/Users/vivia/OneDrive/Documentos/Viviana/TESIS/APLICATIVO/python/
houcha_db_prueba.py', wdir='C:/Users/vivia/OneDrive/Documentos/Viviana/TESIS/APLICATIVO/python')
onexion exitosa
Database 'marketplace_sensormanager'>
Document 'TEM-266'@'1-35c4edf3793a3a823f58a7569af86b9d' {'datetime': '02/17/2021 00:41:16', 'location':
192.168.1.6', 'record': '22.9', 'sensorid': 'TEM-266', 'sensorname': 'Sensor Temperatura', 'sensortype':
TEMPERATURA', 'unitmeasurement': 'GRADOSCENTIGRADOS', '~version': 'CgQCA/YA'}>
lo se pudo insertar los datos  'Database' object has no attribute 'put'
```

Organización 3: http:// 35.208.212.182:5986/_utils

```
ython 3.8.3 (default, Jul  2 2020, 17:30:36) [MSC v.1916 64 bit (AMD64)]
ype "copyright", "credits" or "license" for more information.

Python 7.16.1 -- An enhanced Interactive Python.

n [1]: runfile('C:/Users/vivia/OneDrive/Documentos/Viviana/TESIS/APLICATIVO/python/
houcha_db_prueba.py', wdir='C:/Users/vivia/OneDrive/Documentos/Viviana/TESIS/APLICATIVO/python')
onexion exitosa
Database 'marketplace_sensormanager'>
Document 'TEM-266'@'1-35c4edf3793a3a823f58a7569af86b9d' {'datetime': '02/17/2021 00:41:16', 'location':
192.168.1.6', 'record': '22.9', 'sensorid': 'TEM-266', 'sensorname': 'Sensor Temperatura', 'sensortype':
TEMPERATURA', 'unitmeasurement': 'GRADOSCENTIGRADOS', '~version': 'CgQCA/YA'}>
lo se pudo insertar los datos  'Database' object has no attribute 'put'
```

SEGURIDAD EN HYPERLEDGER FABRIC

Como se ha hablado en otros capítulos, la tecnología IoT debido a su arquitectura y ligereza se ha convertido en una presa fácil para la inseguridad en el momento de transmitir datos afectando a todas sus capas: desde el dispositivo físico hasta la capa de aplicación. El objetivo de esta investigación es crear una "metodología para cifrar datos usando Cadena de Bloques como tecnología de convergencia para dispositivos móviles asociados con Internet de las Cosas, en la capa de aplicación del modelo de capas IoT". Una vez realizadas las pruebas se evidencia que no es posible modificar los datos del libro mayor de la BlockChain mediante su base de datos de estado CouchDB.

Para reafirmar que Hyperledger Fabric es una BlockChain robusta y segura se investigó literatura que evidencie algún tipo de hackeo o alguna vulnerabilidad que permita acceder lo cual no se encontró. Por lo tanto, se afirma que Hyperledger Fabric un sistema distribuido seguro que permite garantizar la seguridad de los datos de los dispositivos IoT.

Además, utilizando el protocolo de consenso RAFT se garantiza el normal funcionamiento de la BlockChain, de tal manera que si un nodo de la red falla se sigue operando con la misma seguridad con el resto de los nodos.

Hyperledger Fabric por sí sola no aporta seguridad y privacidad a los datos, sin embargo, todos los elementos complementarios de la red cuentan con funcionalidades que aportan a la seguridad de los datos, la seguridad más básica es el protocolo TLS, el cual funciona de manera bidireccional y dentro de la red BlockChain, asegurando la comunicación entre los nodos. Además se cuenta con certificados digitales proveídos por el nodo génesis que aloja a la AC (Authority Certificate) (Reyes Delgado, 2018).

Hyperledger Fabric realiza una validación de accesos mediante un Chaincode, que almacena un grupo de ACL también conocido como archivo de lista de control, gracias a este archivo la red BlockChain sabe que usuarios pertenecen a su red y que miembros pueden validar una transacción, ejecutarla e insertarla, esto permite tener un nivel de seguridad incluso en las funciones programadas en los Chaincode de la lógica de negocio (Reyes Delgado, 2018).

Otra seguridad empleada por Hyperledger Fabric es cifrar los datos convirtiéndolos en confidenciales al momento de transmitirlos mediante una clave publica la cual está poseída por los nodos ordenantes y validada por la AC.

La red BlockChain de Hyperledger dispone de un almacenamiento descentralizado, cada miembro de la red tiene una copia exacta de este espacio de tal manera que cuando se envían transacciones a la red, se convierten en bloques con cabeceras en los cuales se almacena el código HASH del bloque y el código de su bloque previo (Akhil, 2018).

Por otra parte, Hyperledger Fabric al ser una red privada permite un control total de todos los miembros que participan de la red, disminuyendo la probabilidad de que exista un nodo malicioso o desconocido que quiera atacar la red (Hyperledger, 2020).

En cuanto a la seguridad del entorno de la API Rest IoT, permite proteger la capa de transporte mediante una conexión segura con cifrado TLS, gRPC y con autenticación vía el SDK de Hyperledger Fabric para realizar consultas a la cadena de bloques.

Finalmente se puede concluir que la red BlockChain construida con Hyperledger Fabric gracias a sus mecanismos de control criptográficos, certificados de seguridad, autoridades certificadoras, protocolos seguros, crea un ambiente robusto, seguro y descentralizado al momento de almacenar información.

CAPITULO VI: CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

Con la implementación de este experimento se puede concluir que la tecnología IoT y BlockChain son compatibles y permiten construir una infraestructura robusta para la protección de datos, esta compatibilidad entre las dos tecnologías se logra mediante una red Hyperledger Fabric.

El prototipo diseñado como caso de prueba permitió la validación de los objetivos planteados para esta investigación, comenzando por el estudio de las diferentes tecnologías, análisis de algoritmos de cifrado de datos para IoT, la implementación de una red BlockChain compatible con IoT, validar la pertinencia entre las tecnologías: IoT, BlockChain y dispositivos móviles, y por último la validación de la seguridad y privacidad de los datos almacenados en la cadena de bloques.

Tasmota (IoT), Hyperledger Fabric, Docker Compose fueron los elementos que permitieron crear una sinergia para integra las tecnologías: IoT y BlockChain las mismas que han sido levantada en una infraestructura liviana a bajo costo y con altas prestaciones. La red BlockChain corre en una máquina virtualizada, junto a seis contratos inteligentes, que plasman la lógica de negocio que es la lectura y recuperación de datos de los dispositivos IoT. Por otro lado, se desarrolló una API Rest IoT que interactúa con los contratos inteligentes al momento de ejecutar transacciones en la red BlockChain, la API se conecta con un interfaz de usuario final diseñada en WordPress que permite visualizar toda la información que está almacenada en la cadena de bloques. Por último, se ensambló y programó un dispositivo IoT en una placa LoLin New NodeMCU V3 el cual envía lecturas de varios sensores: ventana, humo, movimiento, punto rocío, humedad, temperatura a un servidor Mosquitto MQTT, gracias a un módulo programado con Python estos datos se los recupera para enviarlos a la red BlockChain por medio de la API Rest IoT y procesarlos para un almacenamiento seguro en la cadena de bloques.

Con la investigación realizada y basados en la hipótesis: ¿Es posible implementar el cifrado de datos con BlockChain como tecnología de convergencia en las aplicaciones IoT por medio de dispositivos móviles, protegiendo la capa de aplicación del modelo de capas IoT?. Se puede concluir que las tecnologías IoT y BlockChain son compatibles y se complementan una a la otra cuando de crear un entorno seguro se trata, BlockChain atenúa y disminuye los problemas de seguridad y privacidad de los datos transmitidos por los dispositivos IoT en la capa de aplicación.

Luego de realizar las pruebas de seguridad se recomienda emplear Hyperledger Fabric debido a sus elementos anexos que elevan el nivel de seguridad, por lo que esta herramienta puede ser empleada para soluciones empresariales a gran escala, con la finalidad de mantener datos seguros (Iago Tudela Díaz, 2019).

IoT

La tecnología IoT, al ser una tecnología liviana y de bajo consumo sufre de varias vulnerabilidades en sus diferentes, para este caso de prueba se puede aseverar que Hyperledger Fabric añade una seguridad extra en la capa de aplicación donde los datos son procesados y almacenados mediante aplicaciones y funciones. Las pruebas de seguridad permiten aseverar que la red BlockChain es una solución efectiva para garantizar la seguridad y privacidad de los datos en la capa de aplicación, debido a que de ninguna manera se pudo modificar los datos almacenados en la BlockChain.

Hyperledger Fabric.

Hyperledger Fabric es un framework sencillo, escalable, potente, capaz de montar una red BlockChain robusta en poco tiempo, tiene controles de privacidad avanzados, por lo que solo los datos que se desea compartir se comparten entre los participantes de la red "con permisos". Permite crear la lógica de negocio mediante contratos inteligentes escritos en líneas de código, además de los términos de ejecución. El código y los acuerdos contenidos en él contrato inteligente existen en toda la red BlockChain. Las transacciones son rastreables e irreversibles, por lo que crean confianza entre las organizaciones participantes.(Benedict et al., 2019). Posee un consenso más simplificado que cualquier otra red BlockChain pública, lo que ocasiona una optimización en los recursos de infraestructura, manteniéndose como una solución liviana, elástica, económica y escalable. Debido a estas características se puede levantar una red BlockChain privada en una máquina virtual bastante asequible. Para este caso de uso se empleó una máquina virtual: de 4GB de RAM y 2vCPU y 80 GB de almacenamiento a un costo mensual de USD $29.80, la misma que soporta transacciones cada 10 segundos las 24/7/365 días.

Hyperledger Fabric se viene desarrollando desde el año 2015, se ha convertido en un pionero en redes BlockChain empresariales, con una comunidad a nivel mundial, gracias a ello se generar actualizaciones de versión de manera frecuente. Esta investigación, se implementó bajo la versión 2.0.0 de Hyperledger Fabric y la versión 3.3.3 de Docker, sin embargo a pesar contar con una gran comunidad apoyando el proyecto existe escasa información en español por lo que la mayoría de

documentos y artículos encontrados que apoyar a esta investigación fueron en inglés, además se denota la variedad de trabajos realizados con Hyperledger Fabric.(Hyperledger, 2020)

Seguridad de los datos

La seguridad y privacidad de BlockChain como tecnología distribuida, fue mediante dos casos de prueba que intentaron cambiar los datos en la base de datos de estado de Hyperledger Fabric lo cual no fue posible debido a que la base de datos de CouchDB es parte integral de la red. Por lo cual se concluye que BlockChain no permite cambios en su base de datos y su libro mayor si no es de un peer que pertenece a la red y tiene permisos de escritura.

Además, Hyperledger emplea altos estándares de seguridad que permiten mantener la integridad de los datos mediante el uso de protocolos como TLS y gRPC, listas de control de accesos (ACL), encriptado de datos, uso de hash en las cabeceras de los bloques con el algoritmo matemático SHA256, una API Rest IoT asegurada con: protocolo, tokens, certificados SSL, canales de comunicación privados y encriptados entre pares y sus nodos de confianza. En conclusión, todas las características mencionadas agregan seguridad, confidencialidad y privacidad a la cadena de bloques de Hyperledger Fabric, garantizando datos seguros e inmutables.

RECOMENDACIONES

IoT

Los dispositivos IoT sufren varias vulnerabilidades en todas sus capas, por lo que se recomiendo estudiar otras soluciones que permitan atenuar y eliminar la inseguridad de los datos de una forma definitiva en el Resto de las instancias.

Hyperledger Fabric

Para este caso de estudio se usó un protocolo de consenso RAFT, simulando un ambiente de producción el cual está listo para una falla bizantina o un fallo de un nodo, lo que permite seguir transaccionando con el Resto de los nodos. La red BlockChain se la levanto en un solo host con Docker Compose que simula tener tres hosts por separado, por lo que se recomienda que para próximos estudios se lo haga con nodos separados e integrados con Compose.

Seguridad de los datos

Se recomienda investigar protocolos de comunicación que usan encriptación simétrica con, que permitan proteger a los dispositivos de IoT desde el origen de los datos, pasando por todas sus capas hasta el envió de datos a la Red BlockChain de Hyperledger.

TRABAJOS FUTUROS

Después de haber desarrollado este experimento se recomienda a los futuros investigadores basarse a los siguientes posibles trabajos futuros:

- La plataforma escogida para esta investigación fue Hyperledger Fabric debido a la gran cantidad de documentación disponible, sin embargo, existen varias plataformas de BlockChain (Bitcoin, Ethereum, R3 Corda, Ripple, Quorum) que pueden ser empleadas, se propone que se desarrolle una comparativa entre estas plataformas o al menos dos de ellas para evaluar su rendimiento en cuanto a consumo de recursos, disponibilidad, rendimiento y facilidad de implementación.
- Este estudio fue diseñado en Docker Compose, se propone emplear Kubernets para evaluar su rendimiento y seguridad.
- Diseñar la red BlockChain en servidores virtuales separados, para este caso de uso se virtualizó la red en un solo servidor, se recomienda emplear al menos tres servidores por separado para sus miembros.
- Se propone estudiar posibles fallas de seguridad que pudiesen suscitarse en los Smart Contracts, la Wallet, canales de comunicación y la autoridad Certificadora
- Implementar Hyperledger Fabric haciendo uso de una tarjeta Raspberry, instanciando un contenedor Peer.
- Levantar una BlockChain Hyperledger Fabric con Kubernets.
- Analizar protocolos de comunicación que usan encriptación simétrica, con el fin de asegurar los datos de los sensores IoT desde el dispositivo hacia la red BlockChain de Hyperledger asegurando todas las capas del dispositivo IoT

REFERENCIAS

Abbas, Q. E., & Sung-Bong, J. (2019). 2. A Survey of Blockchain and Its Applications. *1st International Conference on Artificial Intelligence in Information and Communication, ICAIIC 2019*, 1–3. https://doi.org/10.1109/ICAIIC.2019.8669067

Academy, B. (2020). *El mundo de la BlockChain*.

Akhil, D. (2018). Managing IoT Data on Hyperledger Blockchain. *Journal of Chemical Information and Modeling*, *53*(9), 1689–1699.

Alexander Preukschat, Carlos Kuchkovsky, Gonzalo Gómez Lardies, D. D. G. e I. M. (2017). Blockchain._La_revolucion_industrial_de. *Blockchain. La Revolución Industrial de Internet*, *1*, 397.

Alladi Tejasvi, C., M., P. R., & Raymond, C. K. K. (2019). Blockchain Applications for Industry 4.0 and Industrial IoT: A Review. *IEEE Access*, *7*, 176935–176951. https://doi.org/10.1109/ACCESS.2019.2956748

Allliance Online TRust. (2018). *Marco de confianza y confidencialidad de IoT v2.5*. *2.5*, 7.

Andersen John Kolb Kaifei Chen Gabriel Fierro David E Culler Raluca Ada Popa, M. P., Andersen, M. P., Kolb, J., Chen, K., Fierro, G., Culler, D. E., & Ada Popa, R. (2017). *WAVE: A Decentralized Authorization System for IoT via Blockchain Smart Contracts*.

Androulaki, E., Barger, A., Bortnikov, V., Muralidharan, S., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Murthy, C., Ferris, C., Laventman, G., Manevich, Y., Nguyen, B., Sethi, M., Singh, G., Smith, K., Sorniotti, A., Stathakopoulou, C., Vukolić, M., … Yellick, J. (2018). Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. *Proceedings of the 13th EuroSys Conference, EuroSys 2018*, *2018-January*. https://doi.org/10.1145/3190508.3190538

ApacheCon. (2021). *CouchDB relax*. The Apache Software Foundation. https://couchdb.apache.org/

ApacheOrg. (2021). *Apache Software Foundation*. Apache Software Foundation. https://www.apache.org/

Arends., T. (n.d.). *TASMOTA*. https://tasmota.github.io/docs/About/

Bahga, A., & Madisetti, V. K. (2016). Blockchain Platform for Industrial Internet of Things. *Journal of Software Engineering and Applications*, *09*(10), 533–546.

https://doi.org/10.4236/jsea.2016.910036

Baz Arturo, Ferreira Irene, Álvarez María, G. R. (2011). Dispositivos móviles. *Ingeniería de Telecomunicación*, 1–12.

Benedict, S., Rumaise, P., & Kaur, J. (2019). IoT Blockchain Solution for Air Quality Monitoring in SmartCities. *International Symposium on Advanced Networks and Telecommunication Systems, ANTS*, *2019-Decem*. https://doi.org/10.1109/ANTS47819.2019.9118148

Bernal, C. (2012). *Metodologia de la Investigación* (O. F. Palma (ed.); Tercera).

Biswas, S., Sharif, K., Li, F., Nour, B., & Wang, Y. (2019). A scalable blockchain framework for secure transactions in IoT. *IEEE Internet of Things Journal*, *6*(3), 4650–4659. https://doi.org/10.1109/JIOT.2018.2874095

Carle, G. (2003). *Chapter 7 Cryptographic Protocols*. 255–292.

Carlos Kuchkovsky, Gonzalo Gomez, Lardies, Daniel Diéz Garcia, I. M. (2017). Block Chain: la Revolución Industrial de Internet. *Alex Preukschat*, 397.

Cázarez Ayala, G., Duarte Valenzuela, A., Castillo Meza, H., Rodríguez Beltrán, A., Lugo Zavala, S., & Ramírez Montenegro, M. (2014). Sistema de sensores inalámbricos para la implementación de n espacios inteligentes. *Ra Ximhai*, 15–26. https://doi.org/10.35197/rx.10.01.e.2014.02.gc

Christidis, K., & Devetsikiotis, M. (2016). Blockchains and Smart Contracts for the Internet of Things. In *IEEE Access* (Vol. 4). https://doi.org/10.1109/ACCESS.2016.2566339

CISCO. (2019). *Securing the Internet of Things: A Proposed Framework*. https://tools.cisco.com/security/center/resources/secure_iot_proposed_framework

Dahlqvist, F., Patel, M., Rajko, A., & Shulman, J. (2019). Growing-opportunities-in-the-Internet-of-Things-v5.pdf. *McKinsey & Company*, *July*.

Dávila Newman, G. (2006). El razonamiento inductivo y deductivo dentro del proceso investigativo en ciencias experimentales y sociales. *Journal of Medical Genetics*, *13*(6), 20. https://doi.org/10.1136/jmg.13.6.469

Diego Cárdenas Quintero, Exel Ropero Silva, Karla Puerto López, Karla, Sanchez Mojica Sergio, Castro Casadiego Jhon, R. M. (2020). Vulnerabilidad en la seguridad del internet de las cosas.

Mundo FESC, *10*(19), 162–179. https://www.fesc.edu.co/Revistas/OJS/index.php/mundofesc/article/download/542/572/

Docker. (2021). *Docker Docs*. Docker Inc. https://docs.docker.com/

Fabiano, N. (2017). The Internet of Things ecosystem: The blockchain and privacy issues. the challenge for a global privacy standard. *Internet of Things for the Global Community, IoTGC 2017 - Proceedings*, *July 2017*. https://doi.org/10.1109/IoTGC.2017.8008970

Fiorentino, G., Occhipinti, C., Corsi, A., Moro, E., Davies, J., & Duke, A. (2020). Blockchain: Enabling Trust on the Internet of Things. *The Internet of Things*, 141–157. https://doi.org/10.1002/9781119545293.ch11

Francisco Javier Balmaseda Aranda. (2018). Aseguramiento de Dispositivos IoT con Blockchain e Infraestructura de Clave Pública. *Universidad Internacional de La Rioja Máster Universitario En Seguridad Informática*, *1*.

Gallagher, S. (2021). *NODE JS*. OpenJS Foundation. https://nodejs.org/es/

Ganz, N. B., Domínguez, F. A., Ares, A. E., & Kuna, H. D. (2018). Avances en selección de biomateriales utilizados en implantes dentales aplicando técnicas de minería de datos. In *XX Workshop de Investigadores en Ciencias de la Computación (WICC)*.

Gélvez Rodríguez, L. F., & Santos Jaimes, L. M. (2020). Internet de las Cosas: una revisión sobre los retos de seguridad y sus contramedidas. In *Revista Ingenio* (Vol. 17, Issue 1, pp. 56–64). https://doi.org/10.22463/2011642x.2370

Generalitat Valenciana, Unio Europea, C. S. T. de la C. V. (2014). Seguridad en Internet de las cosas. *Documento Pública*. http://es.wikipedia.org/w/index.php?title=Internet_de_las_cosas&oldid=76451697

Germán, M., Mariana, T. C., Angelo, B. D., & Sergio, G. (2019). *Modelo de Seguridad IoT*. 1288–1296.

Hang, L., & Kim, D. H. (2019). Design and implementation of an integrated iot blockchain platform for sensing data integrity. *Sensors (Switzerland)*, *19*(10). https://doi.org/10.3390/s19102228

Homayoun, S., Dehghantanha, A., Parizi, R. M., & Choo, K.-K. R. (2019). 1. A Blockchain-based

Framework for Detecting Malicious Mobile Applications in App Stores. *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*, 1–4. https://doi.org/10.1109/CCECE.2019.8861782

Hyperledger. (2020). *Hyperledge Frabic*. The Linux Fundation Projects.

Iago Tudela Díaz. (2019). *Arquitectura Blockchain Para La Securización De Dispositivos Iot Mediante Smart Contracts*. 2. http://castor.det.uvigo.es:8080/xmlui/bitstream/handle/123456789/345/TFG Iago Tudela Díaz.pdf?sequence=1

Institute, B. R. (n.d.). *Medicalchain*. https://medicalchain.com/en/

IoT Security Foundation. (2018). *IoT Security Compliance Framework*. *December*, 46. https://www.iotsecurityfoundation.org/wp-content/uploads/2018/12/IoTSF-IoT-Security-Compliance-Framework-Release-2.0-December-2018.pdf

Jorge Alberto Virguez. (2019). IoT: La Evolución de la Seguridad en el Internet de las Cosas. *Universidad Piloto de Colombia*, 10.

José Daniel, C. F., & Leticia, Bertha, G. B. (n.d.). Objetos de aprendizaje : Una Investigación Bibliográfica y Compilación. *Objects, Learning*, *2012*, 1–24.

Jose Maldonado. (2019). *Protocolos de Consenso – Pilar en la Seguridad Blockchain*. Bitcobie.

Juan Gómez Ortega. (2019). *Blockchain en la Universidad*. https://tic.crue.org/publicaciones/#tendencias

Karen Rose, Scott Eldridge, L. C. (2015). RESEÑA, LA INTERNET DE LAS COSAS— UNA BREVE Conectado, Para entender mejor los problemas y desafíos de un mundo más conectado. *Methodologies and Techniques for Advanced Maintenance*, 63–112. https://doi.org/10.1007/978-0-85729-103-5_5

Khan, R., Khan, S. U., Zaheer, R., & Khan, S. (2012). Future internet: The internet of things architecture, possible applications and key challenges. *Proceedings - 10th International Conference on Frontiers of Information Technology, FIT 2012*, 257–260. https://doi.org/10.1109/FIT.2012.53

Leonardo J. Caballero G. (2019). *Python*. https://entrenamiento-python-basico.readthedocs.io/es/latest/index.html

Litan Avivah. (2019). *Adopción de blockchain combinada con la adopción de IoT*. Gartner.

https://blogs.gartner.com/avivah-litan/2019/12/05/iot-integration-sweet-spot-blockchain-per-gartner-survey/?_ga=2.21066034.841834332.1595386697-528099158.1595386697

Mahapatra, D. (2020). A Study of Blockchain Solutions for IoT Issues. *International Journal for Research in Applied Science and Engineering Technology*, *8*(5), 986–990. https://doi.org/10.22214/ijraset.2020.5156

Marcos Allende. (2018). Blockchain Cómo desarrollar confianza en entornos complejos para generar valor de impacto social . *Banco Interamericano de Desarrollo*, 1–50.

María Burzaco Samper. (2020). Reglamento (Ue) 2016/679 Del Parlamento Europeo Y Del Consejo De 27 De Abril De 2016, Relativo a La Protección De Las Personas Físicas En Lo Que Respecta Al Tratamiento De Datos Personales Y a La Libre Circulación De Estos Datos Y Por El Que Se Deroga La. *Protección de Datos Personales*, *2014*, 17–144. https://doi.org/10.2307/j.ctv17hm980.4

Mellek, Y., & Kaya, A. (2020). *A case Study for BlockChaim in Manufacturing: FabRec: A Prototype for Peer to Peer*.

Mitrokotsa, Aikaterini & Rieback, Melanie & Tanenbaum, A. (2008). *Classification of RFID Attacks*.

Mosquito.org. (n.d.). *Mosquitto MQTT*. https://mosquitto.org/

Nao Nishima. (2021). *Hyperledger Fabric*. https://hyperledger-fabric.readthedocs.io

Norma Beatriz Perez, Miguel Alfredo Bustos, Dr. Mario M. Berón, P. R. H. (2018). *Análisis Sistemático De La Seguridad En Internet of Things*. *2*, 1066–1071.

Oscar Novo. (2019). Scalable access management in IoT using blockchain: A performance evaluation. *IEEE Internet of Things Journal*, *6*(3), 4694–4701. https://doi.org/10.1109/JIOT.2018.2879679

Prometec . Net. (2019). *Prometec . Net*. PROGRAMANDO NODEMCU CON ARDUINO IDE. https://www.prometec.net/nodemcu-arduino-ide/

QUEMBA MARTINEZ, L. A. (2020). CIFRADO DE LA INFORMACIÓN Y SU INCIDENCIA ACTUAL EN LA SEGURIDAD DE LA INFORMACIÓN PARA PEQUEÑAS EMPRESAS PYMES EN COLOMBIA. *(UNAD) Universidad Nacional Abierta a Distancias*, *1*, 99. https://doi.org/10.1017/CBO9781107415324.004

Rashid, A., & Siddique, M. J. (2019). Smart Contracts Integration between Blockchain and Internet of Things: Opportunities and Challenges. *2019 2nd International Conference on Advancements in Computational Sciences, ICACS 2019*. https://doi.org/10.23919/ICACS.2019.8689132

Red Hat. (2021). *Red Hat*. https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces

René Bástian Silva. (2019). DESARROLLO DE APLICACIÓN BLOCKCHAIN PARA PROYECTOS DE GENERACIÓN DISTRIBUIDA EN CHILE. *UNIVERSIDAD DE CHILE*, 63.

Restuccia, F., Kanhere, S. D. andSalil S., Melodia, T., & Das, S. K. (2019). *Blockchain for the Internet of Things: Present and Future*. *1*(1), 1–8.

Reyes Delgado, D. F. (2018). APLICACIÓN DE BLOCKCHAIN PARA LA SEGURIDAD DE LOS DATOS DEL INTERNET OF THINGS. *Cyber Resilience of Systems and Networks*, 76. https://doi.org/10.1007/978-3-319-77492-3_16

Rocha Robson. (2020). *BLOCKCHAIN E IOT: BENEFICIOS Y DESAFIOS*.

Rodríguez, N. (2021). *Hyperledger Fabric: Un Pionero De Blockchain*. 101 BlockChain,. https://101blockchains.com/

RODRÍGUEZ, N. (2019). *Blockchain e IoT: El Dúo Dinámico*. 101BlockChain.

Saldana Gustavo. (2017). *Kaspersky Lab detectó más 7,000 muestras de malware en dispositivos IoT a principios de año*. Kapersky Daily.

Samaniego, M., & Deters, R. (2017). Blockchain as a Service for IoT. *Proceedings - 2016 IEEE International Conference on Internet of Things; IEEE Green Computing and Communications; IEEE Cyber, Physical, and Social Computing; IEEE Smart Data, IThings-GreenCom-CPSCom-Smart Data 2016*, 433–436. https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData.2016.102

Satshi Nakamoto. (2008). Bitcoin: A Peer to Peer Electronic Cash System. *Journal for General Philosophy of Science*, *39*(1), 53–67. https://doi.org/10.1007/s10838-008-9062-0

Security, C. for I. (2015). *Internet of Things Security Companion to the CIS Critical Security Controls (Version 6). October*, 1–21.

SignalIoT. (2020). *Gartner: Blockchain es la combinación perfecta para IoT*. SignalIoT.

Steffen Sorrell. (2017). The internet of things for security providers: Opportunities, strategies, & market leaders 2016-2021. *Juniper*. https://www.juniperresearch.com/press/internet-of-things-botnets-unmanageable-risk

Tool, O. (n.d.). *Online Tool*. Keccak-256.

VALDÉS, R. B. S. (2019). DESARROLLO DE APLICACIÓN BLOCKCHAIN PARA PROYECTOS DE GENERACIÓN DISTRIBUIDA EN CHILE MEMORIA. *UNIVERSIDAD DE CHILE FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS DEPARTAMENTO DE INGENIERÍA ELÉCTRICA*.

Varela, R. (2008). Innovación Empresarial Tercera Edición. In *Innovación Empresarial Tercera Edición*.

Welivesecurity. (2014). *Cámaras IP y contraseñas por defecto: haz lo que yo digo, pero no lo que yo hago*. ESET.

Wickström, J., Westerlund, M., & Pulkkis, G. (2020). *Rethinking IoT Security: A Protocol Based on Blockchain Smart Contracts for Secure and Automated IoT Deployments*. 1–8.

XIA, Q., SIFAH, E. B., KWAME OMONO ASAMOAH, J. G., DU, X., & GUIZANI, M. (2017). MeDShare : Trust-less Medical Data Sharing Among. *IEEE Access*, *5*, 1–10.

BlockChain, J. (21 de 11 de 2017). *Antecedentes de BlockChain en España*. Obtenido de 2017: https://judiciaryblockchain.org/2017/08/31/antecedentes-de-blockchain-en-espana/

Cendón, B. (16 de 01 de 2017). *Bruno Cendón*. Obtenido de Bruno Cendón: http://www.bcendon.com/el-origen-del-iot/

CouchDB_Org. (02 de 05 de 2021). *Guía de CochDB*. Obtenido de Guía de CochDB: https://guide.couchdb.org/

Dai, W. (21 de 2019 de 2018). *Bmoney. Retrieved*. Obtenido de Bmoney. Retrieved: 11

Davies, A. (19 de 02 de 2019). DevTeam. Space. *Cómo asegurar el Internet de las cosas (IoT) con Blockchain*.

Eclipse. (21 de Marzo de 2021). *https://mosquitto.org/*. Obtenido de https://mosquitto.org/: https://mosquitto.org/

enRed. (12 de 12 de 2019). *enRed*. Obtenido de enRed: https://www.en-red.mx/las-ultimas-tecnologias-para-la-optimizacion-de-sistemas-de-ti/

Gesior, T. (14 de 02 de 2020). *https://brightinventions.pl/*. Obtenido de https://brightinventions.pl/: https://brightinventions.pl/blog/p2p-in-hyperledger-fabric

hyperledger.org. (27 de 02 de 2021). *Hyperledger Fabric*. Obtenido de https://hyperledger-fabric.readthedocs.io/: https://hyperledger-fabric.readthedocs.io/en/release-2.2/blockchain.html#what-is-hyperledger-fabric

IONOS, D. G. (13 de 07 de 2020). *https://www.ionos.es/*. Obtenido de Digital Guide IONOS: 13.07

IoT, S. (13 de 05 de 2019). *Señales IoT*. Obtenido de Señales IoT: https://signalsiot.com/conexiones-celulares-iot-representan-13-del-total-de-accesos-moviles-en-2018/

Kaspersky. (18 de 09 de 2018). *Secure List*. Obtenido de Secure List: https://securelist.lat/new-trends-in-the-world-of-iot-threats/87948/

Navarro, W. (2018). *Historia del blockchain, la solución a un problema*. Obtenido de Addalia: https://www.addalia.com/historia-del-blockchain-la-solucion-problema/

Python. (24 de marzo de 2021). *Python Org*. Obtenido de https://www.python.org/: https://www.python.org/

Rodriguez, N. (30 de 10 de 2019). *101blockchains*. Obtenido de https://101blockchains.com/: https://101blockchains.com/es/hyperledger-fabric-blockchain/#:~:text=Hyperledger%20Fabric%20es%20un%20proyecto,a%20administrar%20todas%20sus%20transacciones.

Santos, D. (2019). *Seguridad del IoT: por qué se preocupan los expertos y qué puedes hacer para protegerte*. Iberia: HubSpot.

ANEXOS A:

DESPLEGAR RED HEPERLEDGER FABRIC 2.2.0

PRERREQUISITOS.

Instalar Portainer

```
# DOCKER PORTAINER

sudo docker volume create portainer_data

sudo docker run -d -p 8000:8000 -p 9000:9000 -v /var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer
```

REQUISITOS PREVIOS.

- Docker y Docker Compose.
- Lenguaje de programación Go, según la documentación oficial de Hyperledger Fabric, se recomienda instalar Golang en la versión 1.12.x pero se encontraron problemas para instalar los ejecutables de peer en el equipo local (fuera de cli de Fabric), estas herramientas se probaron con las versiones 1.14.x y 1.15.x.
- JDK 8.
- Git.
- Gradle o Maven.

CLONAR EL PROYECTO.

```
git clone https://github.com/burgosyar/fabric-BlockChain-network
```

El proyecto provee un conjunto de shell scripts con lo necesario para instalar la red BlockChain con Hyperledger Fabric y una base de datos CouchDB

```
acme-network/scripts
```

VERSIÓN DE LA RED.

La versión de la red se encuentra definida en el archivo acme-network/.env en el ejemplo se tienen las versiones:

```
IMAGE_TAG=2.2.0
FABRIC_CA_IMAGE_TAG=1.4.8
```

El CA mantiene la última versión y no es necesario mantener la misma versión que las imágenes de Fabric.

INSTALAR PRERREQUISITOS

Instalar librerías (binarios)

La instalación de los contratos inteligentes requiere de diferentes librerías, ésta por lo general viene como parte del contenedor cliente. Sin embargo, es posible instalar estos binarios en la máquina local para posibilitar la instalación del contrato inteligente para ello, siga las siguientes instrucciones.

Ejecutar el script.

```
./prerequisites.sh
```

El shell tiene los siguientes comando:

```
sudo apt install gcc
go get -u github.com/hyperledger/fabric/cmd/configtxgen
go get -u github.com/hyperledger/fabric/cmd/configtxlator
go get -u github.com/hyperledger/fabric/cmd/peer
go get -u github.com/hyperledger/fabric-ca/cmd/fabric-ca-client
export PATH=$PATH:$GOPATH/bin
```

INICIAR LA RED.

Ingresamos a la ruta de scripts con el comando:

```
cd fabric-BlockChain-network/acme-network/scripts
```

Iniciar la red ejecutando el archivo:

```
./up.sh
```

Este comando iniciará los CA raíz luego procede a registrar y enrolar los usuarios de los CA intermedios y generar sus certificados, con los certificados generados se procede a levantar los CA intermedios y generar los certificados de los usuarios y certificados tls para comunicación; posteriormente se crean los canales, se con todo el material criptográfico generado se inicia la red de Fabric y se hace el ingreso de las organizaciones en el canal creado.

Nota: Para eliminar la red se puede ejecutar el comando `./down.sh` que se encuentra en el directorio scripts.

El comando ./down.sh se debe usar con precaución, ya que este elimina completamente la red y los certificados de la misma.

CONFIGURAR EL CANAL DE LA RED (CHANNEL)

Ejecutar el script channel.sh el cual une las organizaciones de la red al canal "Marketplace" de la red BlockChain, esto es indispensable para un nodo pueda participar en las transacciones enviadas a la BlockChain, dado que hyperledger fabric crea redes BlockChain privadas que requieren la autorización para que los nodos de las organizaciones puedan ver y ejecutar transacciones.

```
./channels.sh
```

INSTALAR EL CHAINCODE FLEDGER

Primero debemos copiar el código del Chaincode (contrato inteligente) dentro del directorio de la red BlockChain designado para la instalación de Chaincode.

```
cp -r /opt/IoT/fledger-IoT-services/Chaincode/IoTmanager /opt/BlockChain/curso-hyperledger-fabric/Chaincode/
```

Para instalar un Chaincode debemos ubicarnos en el directorio que contiene el código fuente de nuestro proyecto.

```
cd fabric-BlockChain-network/Chaincode/IoTmanager
```

En esta ubicación ejecutamos los siguientes comandos:

```
export FABRIC_CFG_PATH=$(cd ../../acme-network && pwd)
export CC_NAME=sensormanager
export CC_VERSION=v1.0
#IMPORTANTE! el valor CC_SEQUENCE tomar un valor incremental dependiendo de las veces que se instala (actualiza el mismo Chaincode)
export CC_SEQUENCE=1
export CHANNEL_NAME=marketplace
export CORE_PEER_TLS_ENABLED=true

export CORE_PEER_ADDRESS=localhost:7051
export CORE_PEER_LOCALMSPID=Org1MSP
export CORE_PEER_TLS_ROOTCERT_FILE=$(cd ../../acme-network && echo $PWD/fabric-ca/org1.acme.com/peers/peer0.org1.acme.com/tls/ca.crt)
export CORE_PEER_MSPCONFIGPATH=$(cd ../../acme-network && echo $PWD/fabric-ca/org1.acme.com/users/admin@org1.acme.com/msp)
export ORDERER_CA=$(cd ../../acme-network && echo $PWD/fabric-ca/org1.acme.com/orderers/orderer.org1.acme.com/tls/ca.crt)
```

```
export ORDERER_ADDRESS=localhost:7050
```

Con estas variables definimos la configuración de nuestro Chaincode, como el nombre, la versión, la secuencia de las políticas de aprobación, el usuario que realiza la instalación. Una vez configurado el Chaincode procedemos a empaquetar el mismo con el comando:

```
peer lifecycle Chaincode package ../../acme-network/channel-artifacts/$CC_NAME$CC_VERSION.tar.gz --path . --lang golang --label $CC_NAME$CC_VERSION
```

Al terminar podemos realizar la instalación en todas las organizaciones, para instalar en la primera organización ejecutamos el comando:

```
peer lifecycle Chaincode install ../../acme-network/channel-artifacts/$CC_NAME$CC_VERSION.tar.gz --peerAddresses $CORE_PEER_ADDRESS --tls $CORE_PEER_TLS_ENABLED --tlsRootCertFiles $CORE_PEER_TLS_ROOTCERT_FILE
```

Al terminar la instalación nos retorna un identificador como el de la siguiente imagen.

Ese identificador es necesario para poder instalar en las otras organizaciones por eso lo asignamos en una variable de configuración, en el ejemplo:

```
export CC_PACKAGE_ID=sensormanagerv1.0:bfdaa1f20aed4cfcbbd32d86cf21713b8923385a92f71f7038078df770b76884
```

AGREGAMOS LAS POLÍTICAS.

```
peer lifecycle Chaincode approveformyorg -o $ORDERER_ADDRESS --tls --cafile $ORDERER_CA --channelID $CHANNEL_NAME --name $CC_NAME --version $CC_VERSION --package-id $CC_PACKAGE_ID --sequence $CC_SEQUENCE --init-required --tlsRootCertFiles $CORE_PEER_TLS_ROOTCERT_FILE --peerAddresses $CORE_PEER_ADDRESS --signature-policy "OUTOF(2, 'Org1MSP.peer','Org2MSP.peer','Org3MSP.peer')"
```

Repetimos los pasos para la organización 2:

```
export CORE_PEER_ADDRESS=localhost:8051
export CORE_PEER_LOCALMSPID=Org2MSP
export CORE_PEER_TLS_ROOTCERT_FILE=$(cd ../../acme-network && echo $PWD/fabric-ca/org2.acme.com/peers/peer0.org2.acme.com/tls/ca.crt)
export CORE_PEER_MSPCONFIGPATH=$(cd ../../acme-network && echo $PWD/fabric-ca/org2.acme.com/users/admin@org2.acme.com/msp)
export ORDERER_CA=$(cd ../../acme-network && echo $PWD/fabric-ca/org2.acme.com/orderers/orderer.org2.acme.com/tls/ca.crt)
export ORDERER_ADDRESS=localhost:8050
```

```
peer lifecycle Chaincode install ../../acme-network/channel-artifacts/$CC_NAME$CC_VERSION.tar.gz --peerAddresses $CORE_PEER_ADDRESS --tls $CORE_PEER_TLS_ENABLED --tlsRootCertFiles $CORE_PEER_TLS_ROOTCERT_FILE
peer lifecycle Chaincode approveformyorg -o $ORDERER_ADDRESS --tls --cafile $ORDERER_CA --channelID $CHANNEL_NAME --name $CC_NAME --version $CC_VERSION --package-id $CC_PACKAGE_ID --sequence $CC_SEQUENCE --init-required --tlsRootCertFiles $CORE_PEER_TLS_ROOTCERT_FILE --peerAddresses $CORE_PEER_ADDRESS --signature-policy "OUTOF(2, 'Org1MSP.peer','Org2MSP.peer','Org3MSP.peer')"
```

Repetimos los pasos para la organización 3:

```
export CORE_PEER_ADDRESS=localhost:9051
export CORE_PEER_LOCALMSPID=Org3MSP
export CORE_PEER_TLS_ROOTCERT_FILE=$(cd ../../acme-network && echo $PWD/fabric-ca/org3.acme.com/peers/peer0.org3.acme.com/tls/ca.crt)
export CORE_PEER_MSPCONFIGPATH=$(cd ../../acme-network && echo $PWD/fabric-ca/org3.acme.com/users/admin@org3.acme.com/msp)
export ORDERER_CA=$(cd ../../acme-network && echo $PWD/fabric-ca/org3.acme.com/orderers/orderer.org3.acme.com/tls/ca.crt)
export ORDERER_ADDRESS=localhost:9050

peer lifecycle Chaincode install ../../acme-network/channel-artifacts/$CC_NAME$CC_VERSION.tar.gz --peerAddresses $CORE_PEER_ADDRESS --tls $CORE_PEER_TLS_ENABLED --tlsRootCertFiles $CORE_PEER_TLS_ROOTCERT_FILE
peer lifecycle Chaincode approveformyorg -o $ORDERER_ADDRESS --tls --cafile $ORDERER_CA --channelID $CHANNEL_NAME --name $CC_NAME --version $CC_VERSION --package-id $CC_PACKAGE_ID --sequence $CC_SEQUENCE --init-required --tlsRootCertFiles $CORE_PEER_TLS_ROOTCERT_FILE --peerAddresses $CORE_PEER_ADDRESS --signature-policy "OUTOF(2, 'Org1MSP.peer','Org2MSP.peer','Org3MSP.peer')"
```

Una vez instalado en todas las organizaciones procedemos a confirmar la instalación con el comando:

```
peer lifecycle Chaincode commit -o $ORDERER_ADDRESS --channelID $CHANNEL_NAME --name $CC_NAME --version $CC_VERSION --sequence $CC_SEQUENCE --init-required --tls --cafile $ORDERER_CA --peerAddresses $CORE_PEER_ADDRESS --peerAddresses localhost:7051 --peerAddresses localhost:8051 --tlsRootCertFiles $CORE_PEER_TLS_ROOTCERT_FILE --tlsRootCertFiles $(cd ../../acme-network && echo $PWD/fabric-ca/org1.acme.com/peers/peer0.org1.acme.com/tls/ca.crt) --tlsRootCertFiles $(cd ../../acme-network && echo $PWD/fabric-ca/org2.acme.com/peers/peer0.org2.acme.com/tls/ca.crt) --signature-policy "OUTOF(2, 'Org1MSP.peer','Org2MSP.peer','Org3MSP.peer')"
```

Ahora es necesario inicializar el Chaincode.

```
peer Chaincode invoke -o $ORDERER_ADDRESS --tls --cafile $ORDERER_CA --peerAddresses $CORE_PEER_ADDRESS --peerAddresses localhost:7051 --peerAddresses localhost:8051 --tlsRootCertFiles $CORE_PEER_TLS_ROOTCERT_FILE --tlsRootCertFiles $(cd ../../acme-network && echo $PWD/fabric-ca/org1.acme.com/peers/peer0.org1.acme.com/tls/ca.crt) --tlsRootCertFiles $(cd ../../acme-network && echo
```

```
$PWD/fabric-ca/org2.acme.com/peers/peer0.org2.acme.com/tls/ca.crt) -C $CHANNEL_NAME -n $CC_NAME --isInit -c '{"Args":[]}'
```

Ahora el Chaincode está completamente instalado e inicializado, para ello hacemos una prueba ingresando un primer registro.

```
peer Chaincode invoke -o $ORDERER_ADDRESS --tls --cafile $ORDERER_CA -C $CHANNEL_NAME -n $CC_NAME -c '{ "Args": ["Record", "SENSOR1", "sensor test","ON","2020/12/22","EC","BINARIO","HUMO"] }' --peerAddresses $CORE_PEER_ADDRESS --peerAddresses localhost:8051 --tlsRootCertFiles $CORE_PEER_TLS_ROOTCERT_FILE --tlsRootCertFiles $(cd ../../acme-network && echo $PWD/fabric-ca/org2.acme.com/peers/peer0.org2.acme.com/tls/ca.crt)
```

Dicho registro se puede consultar de la siguiente manera:

```
peer Chaincode invoke -o $ORDERER_ADDRESS --tls --cafile $ORDERER_CA -C $CHANNEL_NAME -n $CC_NAME -c '{ "Args": ["Query", "SENSOR1"] }' --peerAddresses $CORE_PEER_ADDRESS --peerAddresses localhost:8051 --tlsRootCertFiles $CORE_PEER_TLS_ROOTCERT_FILE --tlsRootCertFiles $(cd ../../acme-network && echo $PWD/fabric-ca/org2.acme.com/peers/peer0.org2.acme.com/tls/ca.crt)
```

Al finalizar este proceso ya se puede encontrar el Chaincode instalado en nuestra red con un primer registro.

PERFIL DE CONEXIÓN

Un perfil de conexión describe un conjunto de componentes, incluidos pares, ordenadores y autoridades de certificación en una red BlockChain de Hyperledger Fabric. También contiene información sobre el canal y la organización relacionada con estos componentes. Una aplicación usa principalmente un perfil de conexión para configurar una puerta de enlace que maneja todas las interacciones de red, lo que le permite centrarse en la lógica empresarial. Un perfil de conexión normalmente lo crea un administrador que comprende la topología de la red.

Un perfil de conexión contiene una descripción de una vista de red, expresada en una sintaxis técnica, que puede ser JSON o YAML. En este tema, usamos la representación YAML, ya que es más fácil de leer. Las puertas de enlace estáticas necesitan más información que las puertas de enlace dinámicas porque estas últimas pueden utilizar el descubrimiento de servicios para aumentar dinámicamente la información en un perfil de conexión.

Un perfil de conexión no debe ser una descripción exhaustiva de un canal de red; solo necesita contener suficiente información para una puerta de enlace que lo esté usando. En la red anterior, el perfil de conexión 1 debe contener al menos las organizaciones que respaldan y los pares para

la transacción, así como identificar a los pares que notificarán a la puerta de enlace cuando la transacción se haya comprometido en el libro mayor.

Es más fácil pensar que un perfil de conexión describe una vista de la red. Podría ser una vista completa, pero eso no es realista por algunas razones:

- Los pares, ordenadores, autoridades de certificación, canales y organizaciones se agregan y eliminan según la demanda.
- Los componentes pueden iniciarse y detenerse o fallar inesperadamente (por ejemplo, corte de energía).
- Una puerta de enlace no necesita una vista de toda la red, solo lo que es necesario para manejar con éxito el envío de transacciones o la notificación de eventos, por ejemplo.
- Service Discovery puede aumentar la información en un perfil de conexión. Específicamente, las puertas de enlace dinámicas se pueden configurar con información mínima de topología de Fabric; el Resto se puede descubrir.

Un perfil de conexión estática normalmente lo crea un administrador que comprende la topología de la red en detalle. Esto se debe a que un perfil estático puede contener mucha información y un administrador debe capturar esto en el perfil de conexión correspondiente. Por el contrario, los perfiles dinámicos minimizan la cantidad de definición requerida y, por lo tanto, pueden ser una mejor opción para los desarrolladores que desean comenzar rápidamente o los administradores que desean crear una puerta de enlace con mayor capacidad de respuesta. Los perfiles de conexión se crean en formato YAML o JSON utilizando un editor de su elección.

USO DEL PERFIL DE CONEXIÓN.

Dicho lo anterior este perfil de conexión puede ser usado por aplicaciones clientes, como es el caso de este proyecto, se usa por la aplicación de servicios para IoT la cual está escrita en nodejs. La implementación está escrita en el archivo:

/opt/IoT/fledger-IoT-services/api/main/src/BlockChain/conector/conector.js

Ejemplo simplificado:

```
const yaml = require('js-yaml');
const { Gateway } = require('fabric-network');
```

```
const connectionProfile =
yaml.safeLoad(fs.readFileSync('/opt/fledger/api/hfabric2/connectionprofile.yml',
'utf8'));

const gateway = new Gateway();

await gateway.connect(connectionProfile, connectionOptions);
```

Después de cargar algunas clases requeridas, vea cómo el archivo connectionprogile .yaml de la puerta de enlace (gateway) se carga desde el sistema de archivos, se convierte en un objeto JSON usando el yaml.safeLoad() método y se usa para configurar una puerta de enlace usando su connect() método.

Al configurar una puerta de enlace con este perfil de conexión, la aplicación de problemas proporciona a la puerta de enlace la topología de red relevante que debe utilizar para procesar transacciones. Esto se debe a que el perfil de conexión contiene información suficiente sobre los canales, organizaciones, pares, ordenadores y CA de para garantizar que las transacciones se puedan procesar correctamente.

La red cuenta con 3 usuarios administradores para poder transaccionar, estos son admin@org1.acme.com, admin@org2.acme.com, admin@org3.acme.com y 3 organizaciones, Org1MSP, Org2MSP, Org3MSP.

Archivo connectionprofile en formato yaml.

```
name: acme-network-local
x-type: "hlfv1"
version: "v1.0"
client:
  organization: Org1MSP
  connection:
    timeout:
      peer:
        endorser: "30000"
  credentialStore:
    path: tmp/hfc-kvs/org3
    cryptoStore:
      path: tmp/hfc-kvs/org3
  clientPrivateKey:
    path: /opt/BlockChain/fabric-BlockChain-network/acme-network/fabric-
ca/org3.acme.com/users/client@org3.acme.com/msp/keystore/priv.key
  clientSignedCert:
    path: /opt/BlockChain/fabric-BlockChain-network/acme-network/fabric-
ca/org3.acme.com/users/client@org3.acme.com/msp/signcerts/cert.pem
```

```
organizations:
  Org1MSP:
    mspid: Org1MSP
    peers:
      - peer0.org1.acme.com
    certificateAuthorities:
      - int.ca.org1.acme.com
    adminPrivateKey:
      path: /opt/BlockChain/fabric-BlockChain-network/acme-network/fabric-
ca/org1.acme.com/users/admin@org1.acme.com/msp/keystore/priv.key
    signedCert:
      path: /opt/BlockChain/fabric-BlockChain-network/acme-network/fabric-
ca/org1.acme.com/users/admin@org1.acme.com/msp/signcerts/cert.pem
  Org2MSP:
    mspid: Org2MSP
    peers:
      - peer0.org2.acme.com
    certificateAuthorities:
      - int.ca.org2.acme.com
    adminPrivateKey:
      path: /opt/BlockChain/fabric-BlockChain-network/acme-network/fabric-
ca/org2.acme.com/users/admin@org2.acme.com/msp/keystore/priv.key
    signedCert:
      path: /opt/BlockChain/fabric-BlockChain-network/acme-network/fabric-
ca/org2.acme.com/users/admin@org2.acme.com/msp/signcerts/cert.pem
  Org3MSP:
    mspid: Org3MSP
    peers:
      - peer0.org3.acme.com
    certificateAuthorities:
      - int.ca.org3.acme.com
    adminPrivateKey:
      path: /opt/BlockChain/fabric-BlockChain-network/acme-network/fabric-
ca/org3.acme.com/users/admin@org3.acme.com/msp/keystore/priv.key
    signedCert:
      path: /opt/BlockChain/fabric-BlockChain-network/acme-network/fabric-
ca/org3.acme.com/users/admin@org3.acme.com/msp/signcerts/cert.pem
channels:
  marketplace:
    orderers:
      - orderer.org1.acme.com
      - orderer.org2.acme.com
      - orderer.org3.acme.com
    peers:
      peer0.org1.acme.com:
```

```
        endorsingPeer: true
        ChaincodeQuery: true
        ledgerQuery: true
        eventSource: true
      peer0.org2.acme.com:
        endorsingPeer: false
        ChaincodeQuery: true
        ledgerQuery: true
        eventSource: false
      peer0.org3.acme.com:
        endorsingPeer: true
        ChaincodeQuery: true
        ledgerQuery: true
        eventSource: false
clients:
  client@org1.acme.com:
    client:
      organization: Org1MSP
      connection:
        timeout:
          peer:
            endorser: "30000"
      credentialStore:
        path: tmp/hfc-kvs/org1
        cryptoStore:
          path: tmp/hfc-kvs/org1
      clientPrivateKey:
        path: /opt/BlockChain/fabric-BlockChain-network/acme-network/fabric-
ca/org1.acme.com/users/client@org1.acme.com/msp/keystore/priv.key
      clientSignedCert:
        path: /opt/BlockChain/fabric-BlockChain-network/acme-network/fabric-
ca/org1.acme.com/users/client@org1.acme.com/msp/signcerts/cert.pem
  client@org2.acme.com:
    client:
      organization: Org2MSP
      connection:
        timeout:
          peer:
            endorser: "30000"
      credentialStore:
        path: tmp/hfc-kvs/org2
        cryptoStore:
          path: tmp/hfc-kvs/org2
      clientPrivateKey:
```

```
      path: /opt/BlockChain/fabric-BlockChain-network/acme-network/fabric-
ca/org2.acme.com/users/client@org2.acme.com/msp/keystore/priv.key
    clientSignedCert:
      path: /opt/BlockChain/fabric-BlockChain-network/acme-network/fabric-
ca/org2.acme.com/users/client@org2.acme.com/msp/signcerts/cert.pem
  client@org3.acme.com:
    client:
      organization: Org3MSP
      connection:
        timeout:
          peer:
            endorser: "30000"
      credentialStore:
        path: tmp/hfc-kvs/org3
        cryptoStore:
          path: tmp/hfc-kvs/org3
      clientPrivateKey:
        path: /opt/BlockChain/fabric-BlockChain-network/acme-network/fabric-
ca/org3.acme.com/users/client@org3.acme.com/msp/keystore/priv.key
      clientSignedCert:
        path: /opt/BlockChain/fabric-BlockChain-network/acme-network/fabric-
ca/org3.acme.com/users/client@org3.acme.com/msp/signcerts/cert.pem
peers:
  peer0.org1.acme.com:
    url: grpcs://peer0.org1.acme.com:7051
    tlsCACerts:
      path: /opt/BlockChain/fabric-BlockChain-network/acme-network/fabric-ca/org1.acme.com/tls-int/ca-chain.pem
    grpcOptions:
      ssl-target-name-override: peer0.org1.acme.com
      hostnameOverride: peer0.org1.acme.com
      grpc.keepalive_time_ms: 600000
  peer0.org2.acme.com:
    url: grpcs://peer0.org2.acme.com:8051
    tlsCACerts:
      path: /opt/BlockChain/fabric-BlockChain-network/acme-network/fabric-ca/org2.acme.com/tls-int/ca-chain.pem
    grpcOptions:
      ssl-target-name-override: peer0.org2.acme.com
      hostnameOverride: peer0.org2.acme.com
      grpc.keepalive_time_ms: 600000
  peer0.org3.acme.com:
    url: grpcs://peer0.org3.acme.com:9051
    tlsCACerts:
      path: /opt/BlockChain/fabric-BlockChain-network/acme-network/fabric-ca/org3.acme.com/tls-int/ca-chain.pem
    grpcOptions:
      ssl-target-name-override: peer0.org3.acme.com
```

```
      hostnameOverride: peer0.org3.acme.com
      grpc.keepalive_time_ms: 600000
orderers:
  orderer.org1.acme.com:
    url: grpcs://orderer.org1.acme.com:7050
    grpcOptions:
      ssl-target-name-override: orderer.org1.acme.com
    tlsCACerts:
      path: /opt/BlockChain/fabric-BlockChain-network/acme-network/fabric-ca/org1.acme.com/tls-int/ca-chain.pem
  orderer.org2.acme.com:
    url: grpcs://orderer.org2.acme.com:8050
    grpcOptions:
      ssl-target-name-override: orderer.org2.acme.com
    tlsCACerts:
      path: /opt/BlockChain/fabric-BlockChain-network/acme-network/fabric-ca/org2.acme.com/tls-int/ca-chain.pem
  orderer.org3.acme.com:
    url: grpcs://orderer.org3.acme.com:9050
    grpcOptions:
      ssl-target-name-override: orderer.org3.acme.com
    tlsCACerts:
      path: /opt/BlockChain/fabric-BlockChain-network/acme-network/fabric-ca/org3.acme.com/tls-int/ca-chain.pem
certificateAuthorities:
 int.ca.org1.acme.com:
   url: http://int.ca.org1.acme.com:7056
   caName: int.ca.org1.acme.com
   tlsCACerts:
      path: /opt/BlockChain/fabric-BlockChain-network/acme-network/fabric-ca/org1.acme.com/int/ca-chain.pem
   httpOptions:
     verify: false
   registrar:
      - enrollId: admin
        enrollSecret: adminpw
 int.ca.org2.acme.com:
   url: http://int.ca.org2.acme.com:8056
   caName: int.ca.org2.acme.com
   tlsCACerts:
      path: /opt/BlockChain/fabric-BlockChain-network/acme-network/fabric-ca/org2.acme.com/int/ca-chain.pem
   httpOptions:
     verify: false
   registrar:
      - enrollId: admin
        enrollSecret: adminpw
 int.ca.org3.acme.com:
   url: http://int.ca.org3.acme.com:9056
   caName: int.ca.org3.acme.com
```

```
tlsCACerts:
  path: /opt/BlockChain/fabric-BlockChain-network/acme-network/fabric-ca/org3.acme.com/int/ca-chain.pem
httpOptions:
 verify: false
registrar:
  - enrollId: admin
    enrollSecret: adminpw
```

La red cuenta con tres organizaciones cada una con un nodo de ordenamiento, una CA intermedia y un nodo peer, en el archivo de configuración se encuentran definida la configuración para las tres organizaciones.

ANEXO B:

API Rest IoT

Requisitos:

- Node js v12+
- npm v4
- pm2

Instalación

1. Clonar proyecto

```
git clone https://gitlab.com/vivianaburgosyar/API-Rest-IoT-BlockChain.git
```

2. Configurar archivo host máquina: /etc/host

```
127.0.0.1 peer0.org1.acme.com peer0.org2.acme.com peer0.org3.acme.com orderer.acme.com
orderer.org1.acme.com orderer.org2.acme.com orderer.org3.acme.com int.ca.org1.acme.com acme.com
int.ca.org3.acme.com int.ca.org2.acme.com
```

3. Ir al directorio del proyecto:

```
cd api/main
```

4. Ejecutar:

```
sudo npm install
```

5. Si no existen las Wallets de admin y de user:

   5.1. Ir al directorio

```
cd api/main/src/BlockChain
```

   5.2. Crear Wallet administrador:

```
node enrollAdmin.js
```

   5.3. Crear Wallet usuario:

```
node registerUser.js
```

6. En el directorio api/main/src ejecutar:

```
sudo pm2 start index.js --name IoT-services
```

ANEXO C:

INSTALACIÓN WORDPRESS Y PLUGIN IoT

Antes de comenzar con las instalaciones y configuraciones, asegúrese de estar conectado en el servidor por medio de ssh, pues todas las configuraciones se harán por medio de comandos.

PRE-REQUISITOS

- Servidor Virtual con Sistema Operativo Ubuntu 18 o Debian.
- Clave de acceso SSH

INSTALAR APACHE Y PHP EN GCP - GOOGLE CLOUD PLATFORM.

Primero actualizamos los repositorios para evitar inconvenientes posteriormente.

```
sudo apt-get update -y
```

Instalar Apache.

Apache será el servidor de aplicaciones con el que el Servicio de WordPress funcionará. Escribir el comando de instalación:

```
sudo apt-get install apache2 -y
```

Iniciar apache.

```
sudo systemctl start apache2.service
```

Activar servicios de apache.

El siguiente comando activa el servicio de apache, para luego reiniciarlos.

```
sudo systemctl enable apache2.service
sudo systemctl Restart apache2.service
```

Verificar el estado del servidor.

```
sudo systemctl status apache2.service
```

Enseguida se muestra el estado del servidor, el mismo que se encuentra en estado "activo".

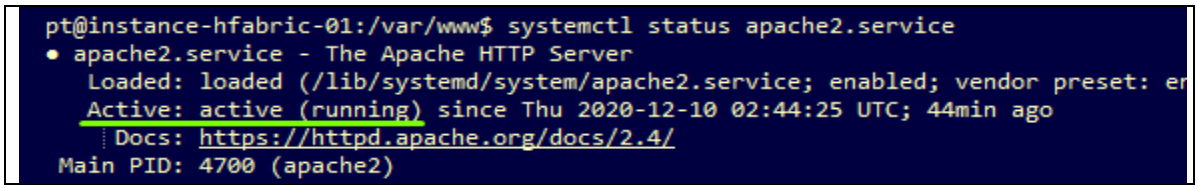


Figura 20. Servicio activo de Apache

Fuente: Investigador

Instalar PHP.

PHP el lenguaje interprete que usa WordPress por lo tanto es necesario instalar. Ejecute el siguiente comando.

```
sudo apt-get install php -y
```

Al finalizar la instalación, descargar e instalamos algunas de las extensiones de PHP necesarias para su buen desempeño escribir lo siguiente:

```
sudo apt-get install -y php-{bcmath,bz2,intl,gd,mbstring,mcrypt,mysql,
zip} && sudo apt-get install libapache2-mod-php -y
```

Descargar la versión de MySQL y MariaDB para Debian

En la versión reciente de Debian, MySQL se reemplaza con MariaDB, que es una bifurcación de MySQL, para descargarla utilizar el comando:

```
   sudo apt-get install default-mysql-server
```

Luego de detallar los paquetes y librerías que serán instaladas, se pide una confirmación para continuar con la instalación. Ingresar la letra "**y**" para continuar. La instalación continuará inmediatamente.

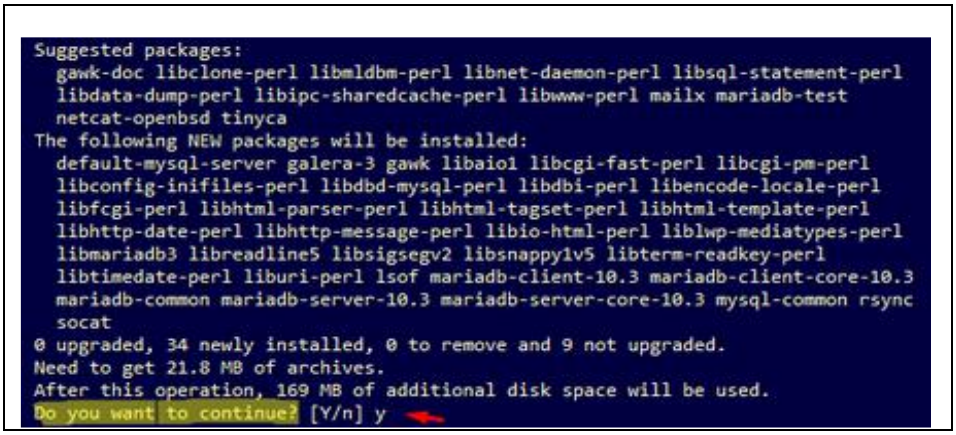


Figura 21. Instalación de Paquetes MariaDB

Fuente: Investigador

La instalación finaliza, verificamos la versión instalada, utilizando el comando:

```
mysql --version
```

Si la instalación es correcta se mostrará la versión de MySQL que fue instalada.

```
mysql  Ver 15.1 Distrib 10.3.27-MariaDB, for debian-linux-gnu (x86_64) using readline 5.2
```

ACCEDER A MARIADB.

Usar el cliente de consola *mysql* para crear el usuario que manejará esta base de datos:

```
mysql -u root -p
```

Enseguida pedirá la contraseña, por defecto **no tiene contraseña**, por lo tanto, presionar la tecla **ENTER** para ingresar. Se mostrará unmensaje de bienvenida.

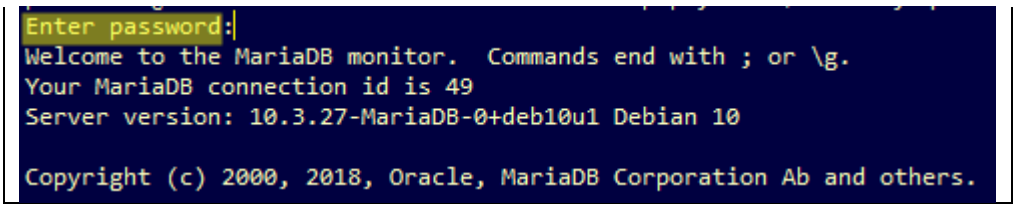


Figura 22. Pantalla de Password de MariaDB

Fuente: Investigador

CREAR BASE DE DATOS PARA WORDPRESS.

Para crear una nueva base de datos que se dedicará a la instancia de la página en WordPress:

```
create database wordpress;
```

Una vez creada la base de datos, verificar en el listado de MariaDB/ MySQL.

```
show databases;
```

Listado de Bases de Datos en MariaDB.

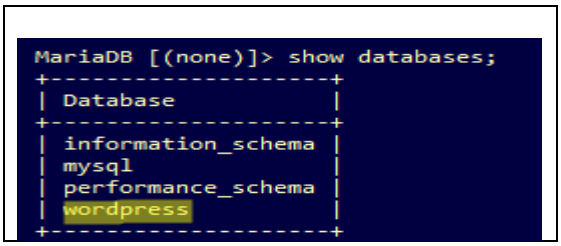


Figura 23. Listado de Bases de Datos en MariaDB

Fuente: Investigador

CREAR UN USUARIO CON PRIVILEGIOS.

Para crear un usuario con uso exclusivo de la base de datos creada en el paso anterior y que se pueda conectar desde cualquier equipo, seguir los siguientes pasos.

```
create user 'usuario'@'%' identified by 'P@ssw0rd12';
```

Dar todos los permisos al usuario anterior sobre la base de datos creada anteriormente:

```
grant all privileges on dbwordpresstest.* TO 'usuario'@'%';
```

En caso de que se necesite que el usuario acceda solo por una dirección IP, se puede ajustar la dirección desde donde se hará la conexión,sustituir el símbolo % para la dirección IP. Por ejemplo, para permitir las conexiones sólo desde el equipo 192.168.0.3:

```
create user 'usuario'@'192.168.0.3' identified by 'P@ssw0rd12';
```

Actualizar los permisos de los usuarios en el servidor, es muy recomendable para borrar las memorias caché de permisos:

```
flush privileges;
```

Cerrar el cliente MySQL.

```
> exit
```

INSTALAR WORDPRESS.

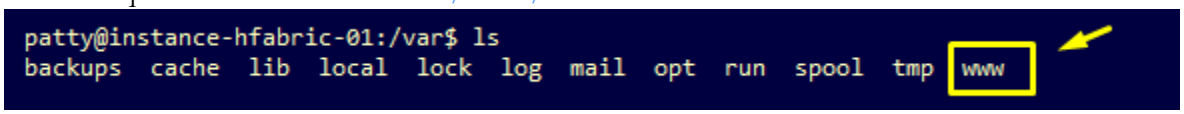
Para la instalación WordPress ubicarse en el directorio /var/

```
cd /var/
```

Listar el contenido del directorio

```
ls
```

Verificar que exista el subdirectorio /www/




```
sudo mkdir www
```

Verificar que se ha creado, listando nuevamente el contenido del directorio /var/

```
ls
```

DESCARGA DE WORDPRESS

Descargar WordPress desde la página oficial: https://wordpress.org/download/

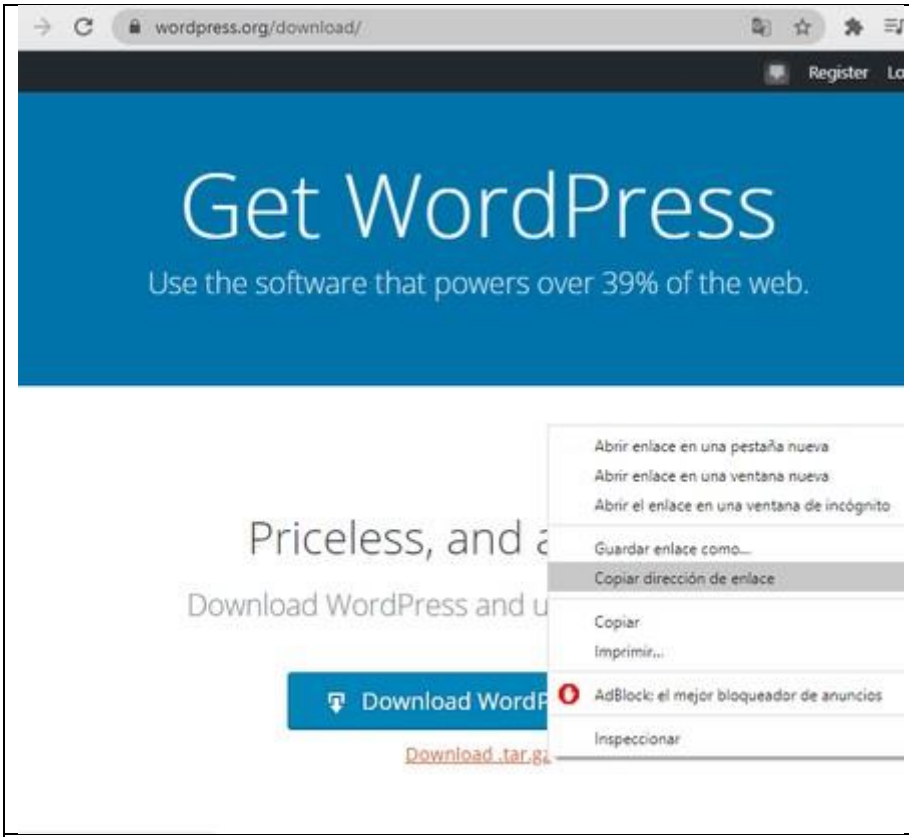


Figura 24. Página Oficial de WordPress

Fuente: Investigador

Para copiar la ruta de descargar clic derecho en el enlace "**Download .tar.gz**", y seleccionamos la opción "Copiar dirección de enlace".

Ubicarse en el directorio **/var/www**, aquí se instalará WordPress, para ello se usa el comando **wget** y dar clic derecho para que el enlacede descarga se copie automáticamente en la consola de comandos.

```
sudo wget https://wordpress.org/latest.tar.gz
```

La descarga empezará enseguida. Cuando finalice, verificar que se encuentre en el directorio. Utilizar el comando ls para ver el contenido de la **/www**

```
ls
```

Instalación de WordPress

Descomprimir el archivo comprimido, para continuar con la instalación.

```
sudo tar -xzf latest.tar.gz
```

Verificar que el archivo se haya comprimido.

```
ls
```

```
instance-hfabric-01:/var/www$ ls
  latest.tar.gz  wordpress
```

Ubicarse en el directorio **/var/www/wordpress** y verificar su contenido con el comando:

```
ls
```

El contenido debe ser similar al que se muestra en la **Figura 19.**

```
@instance-hfabric-01:/var/www/wordpress$ ls
index.php    wp-activate.php     wp-comments-post.php  wp-cron.php        wp-load.php   wp-settings.
license.txt  wp-admin            wp-config-sample.php  wp-includes        wp-login.php  wp-signup.ph
readme.html  wp-blog-header.php  wp-content            wp-links-opml.php  wp-mail.php   wp-trackback
```

Figura 25. Contenido de la Carpeta WordPress
Fuente: Investigador

Para continuar con la instalación, dar todos los privilegios a la carpeta wordpres al usuario y al grupo **www-data**. Este usuario es el propietario del servidor web de Apache por lo que podrá leer y escribir archivos de WordPress ubicados en la carpeta www.

Cambiar los permisos de la siguiente manera:

```
sudo chown -R www-data:www-data wordpress/
```

Para verificar que se hayan cambiado los permisos del directorio **/car/www/wordpress** y su contenido ejecutar.

```
ls -l
```

Carpetas con permisos de lectura y escritura

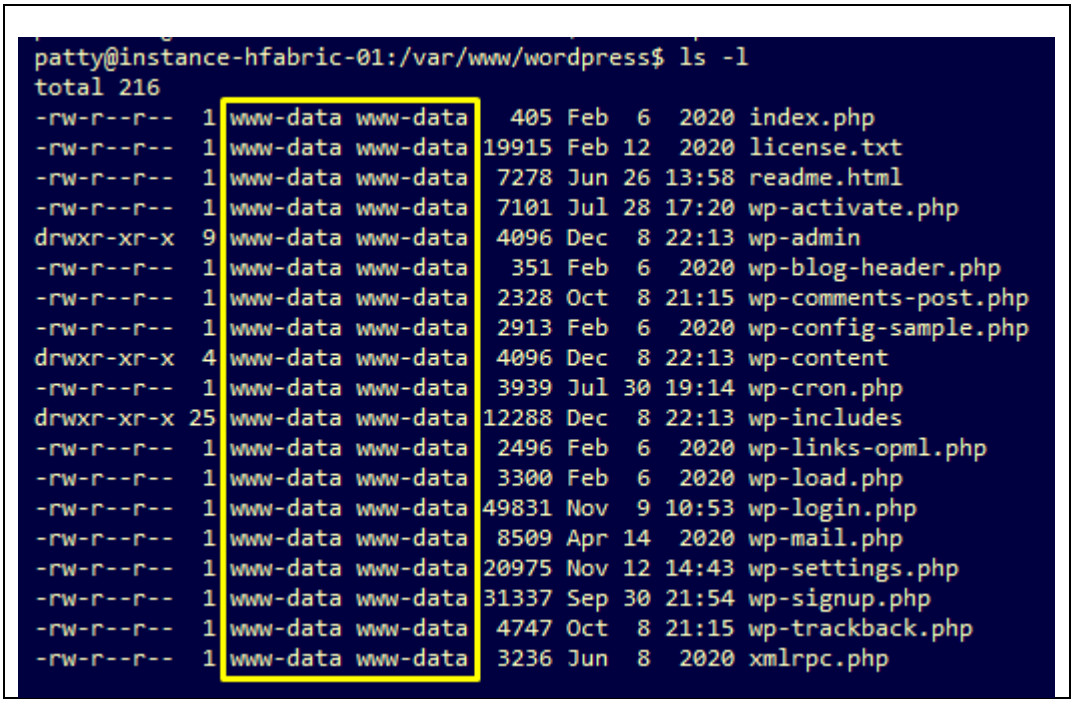


Figura 26. Permisos de lectura y escritura carpeta WordPress

Fuente: Investigador

Configuración de Apache.

Con la configuración por defecto el archivo .htaccess tiene Restricciones para escritura y modificación del mismos. WordPress y muchos de sus complementos utilizan estos archivos de forma exhaustiva para realizar ajustes de comportamiento del servidor web dentro del directorio, por ello es importante liberar estos permisos.

Ubicarse en el directorio **/etc/apache2**

```
cd /etc/apache2
```

Crear un archivo para habilitar la disponibilidad de wordpress en el servidor.

```
sudo touch /etc/apache2/sites-available/wordpress.conf
```

El archivo creado está en blanco. Editarlo con el siguiente comando:

```
sudo nano /etc/apache2/sites-available/wordpress.conf
```

Copie y pegue el contenido a continuación en el archivo:

```
<VirtualHost *:80> ServerName

127.0.0.1

DirectoryIndex index.html index.phpDocumentRoot /var/www/wordpress

</VirtualHost>
```

Editor Nano, archivo de configuración **wordpress.conf**

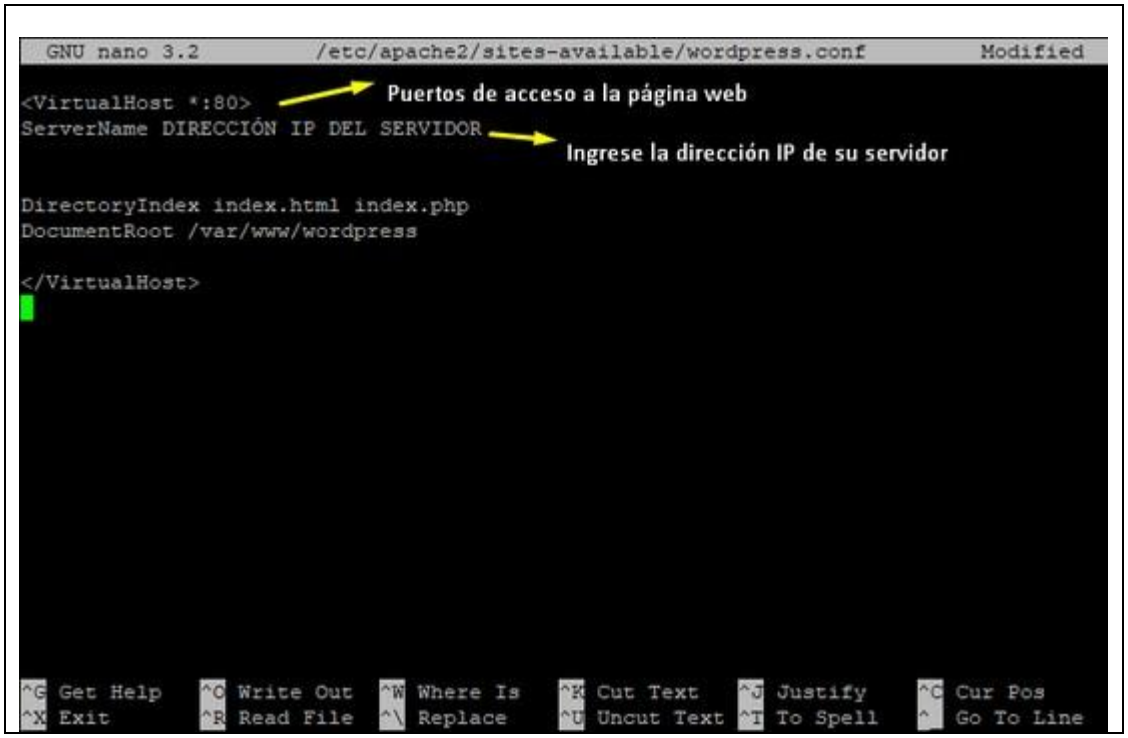


Figura 27. Editor Nano archivo worpress.conf

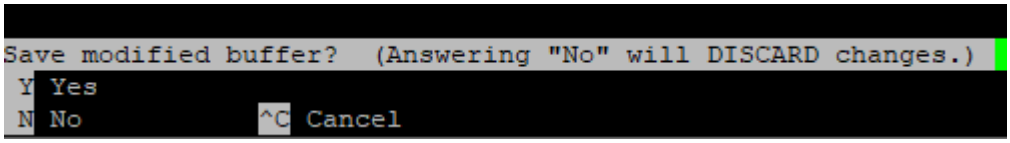
Fuente: Investigador

Para utilizar un dominio en lugar de una dirección IP, configure en '**ServerName**' el dominio deseado. Guíese en el código de referencia acontinuación:

```
<VirtualHost *:80>

ServerName solvetic.lan ServerAdmin webmaster@localhost

DirectoryIndex index.html index.php
```

```
    DocumentRoot /var/www/html/solvetic.lan

    </VirtualHost>
```

Para salir presione las teclas “Ctrl+X“. Luego confirme presionando la letra “Y“



Agregar el sitio para que esté disponible en el servidor Apache.

```
    sudo a2ensite wordpress.conf
```

Aparece un mensaje que indica la activación del sitio WordPress. Para que la configuración sea efectiva se pide reiniciar el servicio de Apache.

```
patty@instance-hfabric-01:/etc/apache2$ sudo a2ensite wordpress.conf
Enabling site wordpress.
To activate the new configuration, you need to run:
  systemctl reload apache2
```

Reiniciar el servicio de Apache, ejecutando:

```
    sudo systemctl reload apache2
```

Comprobamos que el sitio se encuentre entre los sitios disponibles:

```
    ls /etc/apache2/sites-available/
```

```
patty@instance-hfabric-01:/etc/apache2$ ls /etc/apache2/sites-available/
000-default.conf  default-ssl.conf  wordpress.conf
```

En los sitios disponibles hay un sitio por defecto, el mismo que debe ser eliminado.

```
    sudo a2dissite 000-default.conf
```

Cargar el servidor para que actualice los cambios reiniciando el servicio.

```
    sudo systemctl reload apache2
```

Comprobar la configuración del servidor.

```
sudo apachectl -S
```

```
patty@instance-hfabric-01:/var$ sudo apachectl -S
VirtualHost configuration:
*:80                   IP-SERVIDOR (/etc/apache2/sites-enabled/wordpress.conf:1)
ServerRoot: "/etc/apache2"
Main DocumentRoot: "/var/www/html"
Main ErrorLog: "/var/log/apache2/error.log"
Mutex default: dir="/var/run/apache2/" mechanism=default
Mutex mpm-accept: using_defaults
Mutex watchdog-callback: using_defaults
PidFile: "/var/run/apache2/apache2.pid"
Define: DUMP_VHOSTS
Define: DUMP_RUN_CFG
User: name="www-data" id=33
Group: name="www-data" id=33
```

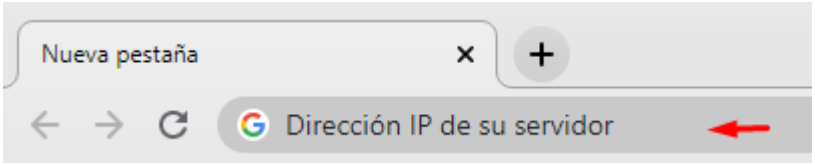
Figura 28. Configuraciones Apache

Fuente: Investigador

IP-SERVIDOR, deberá indicar su dirección IP. Ejemplo: 192.168.0.1

Completar la instalación a través de la interfaz web

Abrir el navegador y colocar la dirección IP del servidor para continuar la instalación de WordPress. Ejemplo de IP: 192.168.0.1



Elegir el idioma del sitio web.

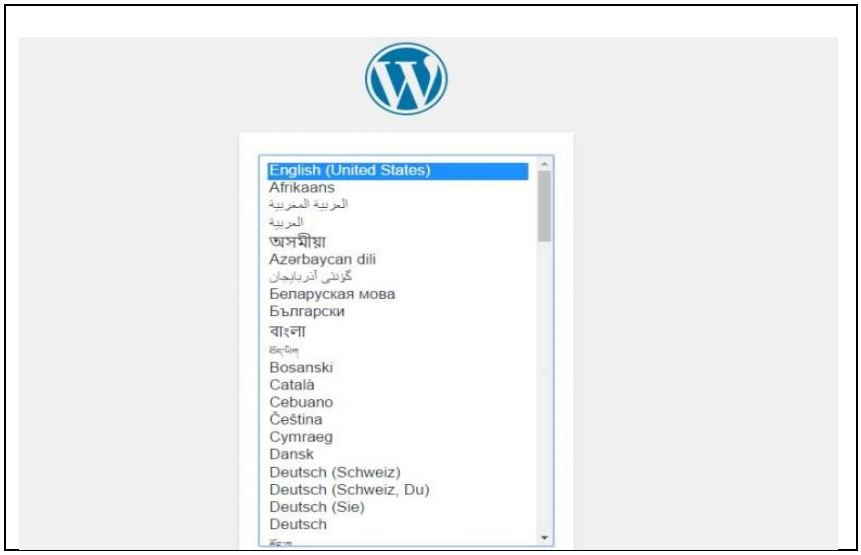


Figura 29. Idioma de WordPress

Fuente: Investigador

Se mostrará una pantalla de bienvenida. Damos clic en el botón "¡Vamos a ello!"

Figura 30. Instalación WordPress

Fuente: Investigador

Ingrese los datos del usuario al que se otorgó los permisos de la base de datos "wordpress". Los datos serán los que usted configuróprevіamente en la sección ***Crear usuario con privilegios para la base de datos.***

Exactamente iguales, sin comillas.

Figura 31. Configuración Base de Datos y Usuario de WorPress
Fuente: Investigador

Si todos los datos son correctos, aparecerá un mensaje para continuar con la instalación.

Figura 32. Ejecutar Instalación WordPress
Fuente: Investigador

En la siguiente pantalla se pide el nombre o título para mostrar en su página web, y se pide un usuario, una contraseña y un correo electrónicopara acceder a la misma.

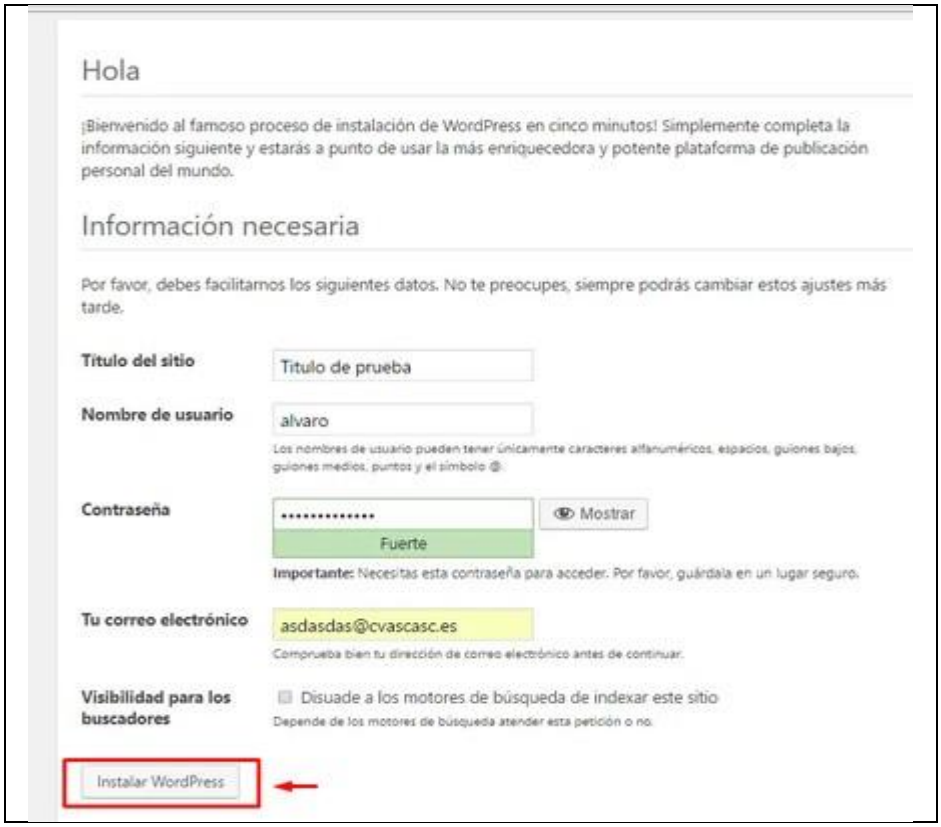


Figura 33. Información del Sitio WordPress
Fuente: Investigador

Luego de ingresar los datos necesarios para el portal principal, dar clic en el botón "Instalar Wordpress". Una vez concluida instalada dar clic en **Acceder.**

Finalmente, damos clic en el botón "Acceder", y se mostrará la página de login, en la cual ingresaremos el usuario y la contraseña queacabamos de crear.

Figura 34. Login WordPress
Fuente: Investigador

INSTALACIÓN PLUGGIN - FRONT END

Una vez logueado se podrá acceder al panel de administrador de WordPress .En la parte izquierda, opción "Plugins". Damos clic

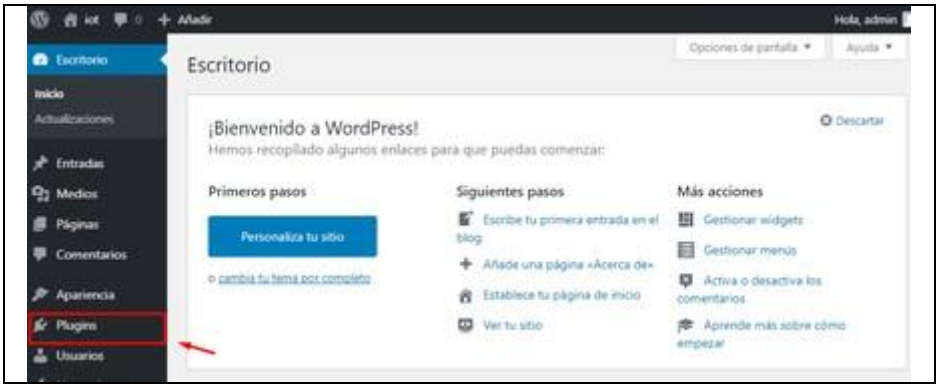


Figura 35. Plugins WordPress
Fuente: Investigador

Junto al título "Plugins" dar clic en el botón "**Añadir nuevo**".

Figura 36. Añadir Plugin

Fuente: Investigador

Damos clic en el botón "Subir plugin".

Figura 37. Subir Plugin

Fuente: investigador

Clic en "Seleccionar archivo", elegir el plugin comprimid, dar clic en Abrir y luego en "Instalar Ahora"

Para descargarse el plugin: https://gitlab.com/vivianaburgosyar/IoT-front-end.git

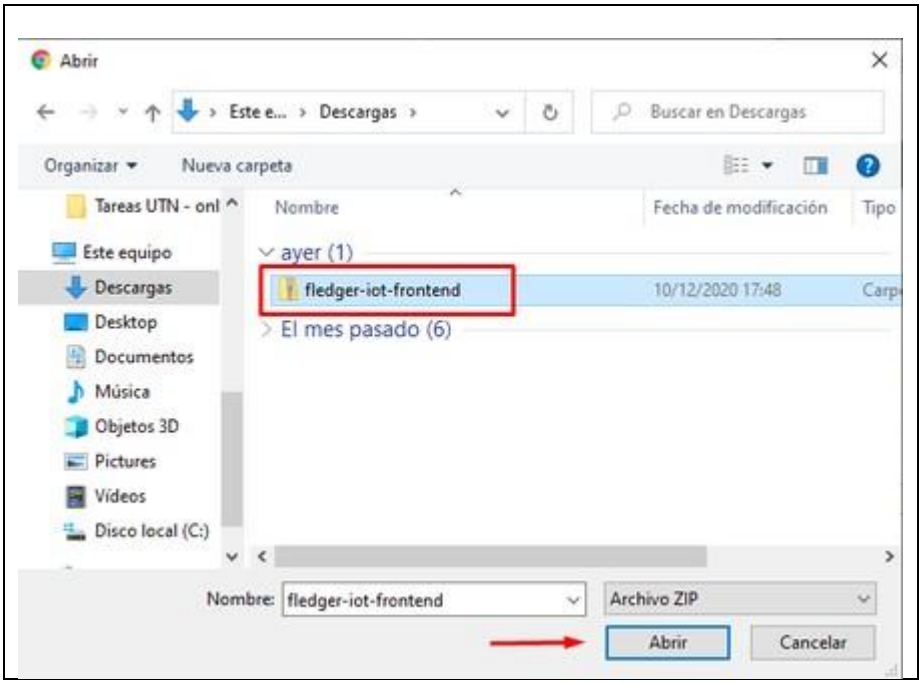


Figura 38. Cargar proyecto Front - End

Fuente: Investigador

Esperar que el plugin se instale, una vez concluida la instalación dar clic en la opción "Activar Plugin"

Volver a la página de "Plugins," un mensaje indica que el plugin ha sido activado, y se mostrará en el panel de configuración.

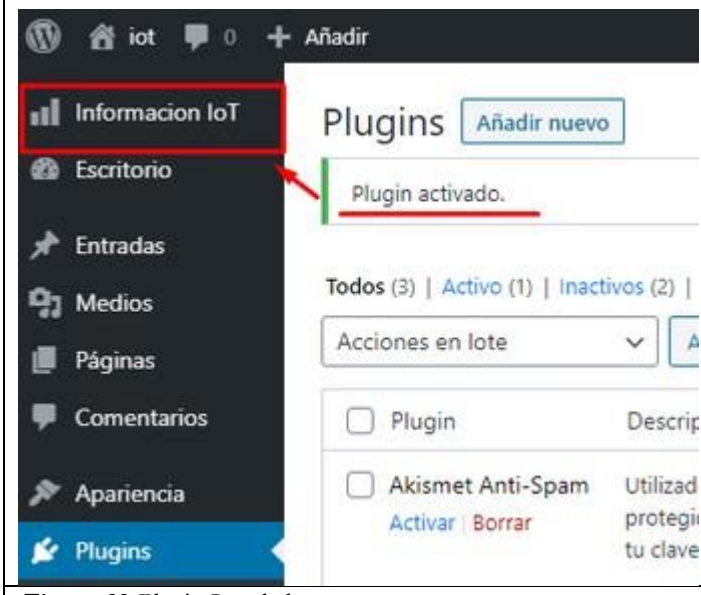


Figura 39. Plugin Instalado

Fuente: Investigador

Finalmente se podrá acceder al Front – End diseñado para mostrar los datos almacenado en la BlockChain en el objeto "**Información IoT**".

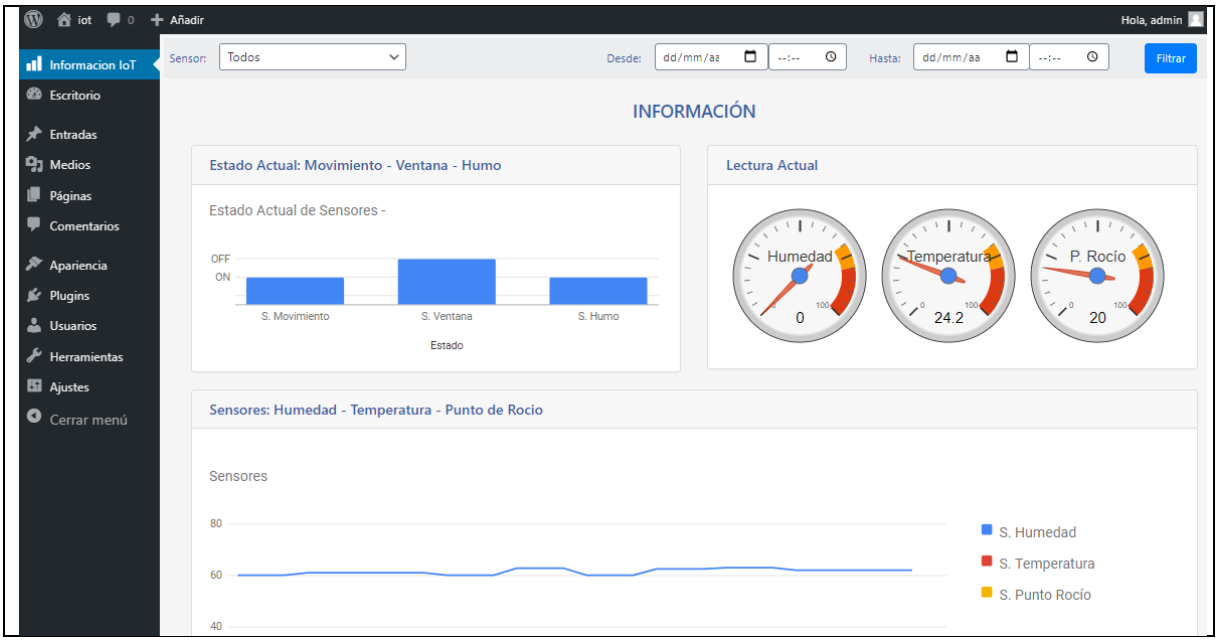


Figura 40. Front - End IoT

Fuente: Investigador

ANEXO D

MANUAL DE USARIO FRONT -END IoT

PANTALLA INFORMACIÓN

La pantalla "Información", contiene estadísticas generales de los sensores.

En el caso de los dos primeros recuadros, los gráficos se generan dinámicamente según el último registro obtenido, se muestra también la hora de la lectura. El gráfico inferior, se genera en correspondencia con las lecturas registradas en la última hora del día.

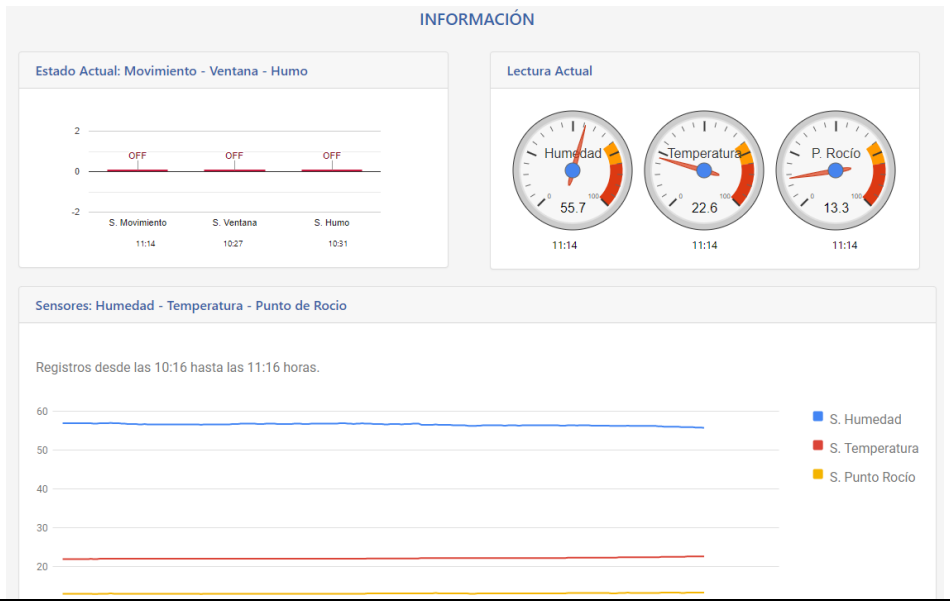


Figura 41. Panel principal del Front – End IoT

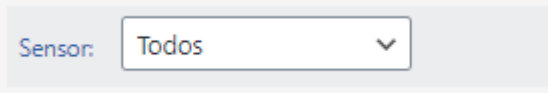
Fuente: Investigador

DESCRIPCIÓN DE LOS COMPONENTES

Cabecera para filtrar información

Se encuentra en la parte superior de la página y permite seleccionar la información a visualizar, la cual puede filtrarse según el sensor, fecha y/o hora según se necesite.

En primera instancia se muestra de la siguiente manera, indicando que las estadísticas mostradas se refieren a todos los sensores.



Si se cambia la opción despegable "sensor" la barra cambia de la siguiente manera:

Información gráfica

La pantalla "información" se divide en tres componentes gráficos, Esta información no puede ser filtrada, ya que su objetivo es presentar información actualizada.

Estado Actual: Sensor Movimiento - Ventana - Humo

Muestra último estado registrado de los sensores movimiento, ventana y humo. Estos estados pueden ser: ON (encendido) y OFF (apagado).

Lectura Actual: Sensor Humedad - Temperatura - Punto de Rocío

Muestra la última lectura registrada de los sensores movimiento, ventana y humo.

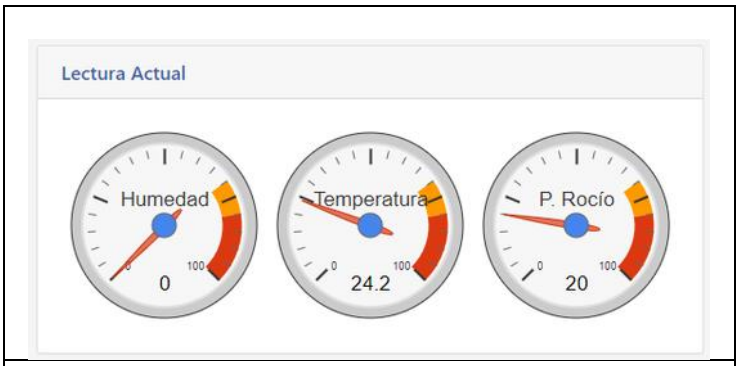
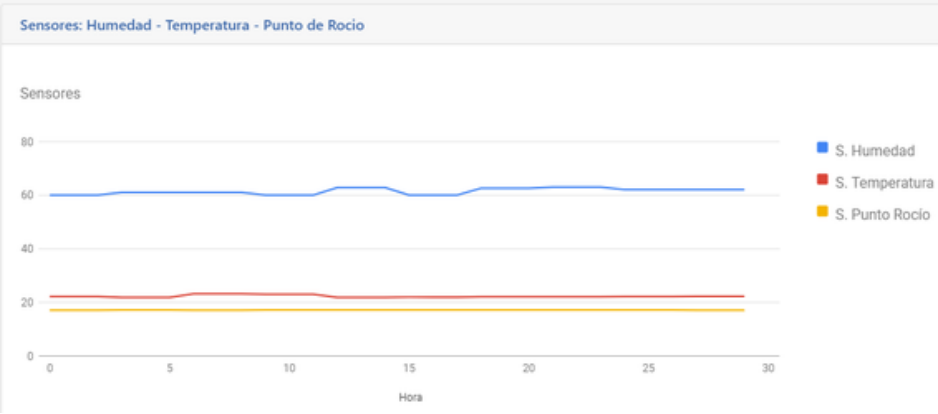


Figura 42. Lecturas sensor Humedad - Temperatura - Punto Rocío

Fuente: Investigador

Sensores: Humedad - Temperatura - Punto de Rocío

Muestra los valores registrados de los sensores humedad, temperatura y punto de rocío en la última hora.



Funcionamiento

Gráfico: Sensores Humedad - Temperatura - Punto de Rocío

Puede visualizar el valor exacto de cualquier lectura registrada. Para ello puede:

1. Seleccionar el nombre del sensor en la parte derecha.
2. La línea de lecturas que lo representan será resaltada sobre las otras.
3. Ubicar el registro (punto) deseado
4. Hacer clic sobre él, para obtener el valor registrado.

Descripción de los componentes por sensor:

- Promedio Hoy: Indica el promedio de todas las lecturas del día hasta el momento de su consulta.
- Valor Máx. Hoy: Indica la lectura máxima del día, obtenida hasta el momento de su consulta.
- Valor Min. Hoy: Indica la lectura mínima del día, obtenida hasta el momento de su consulta.

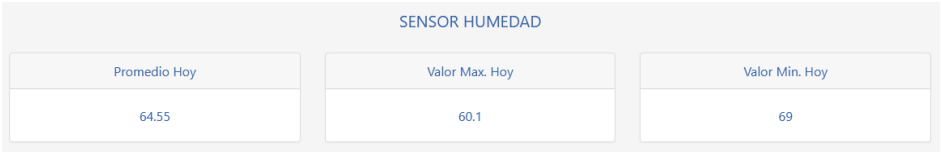
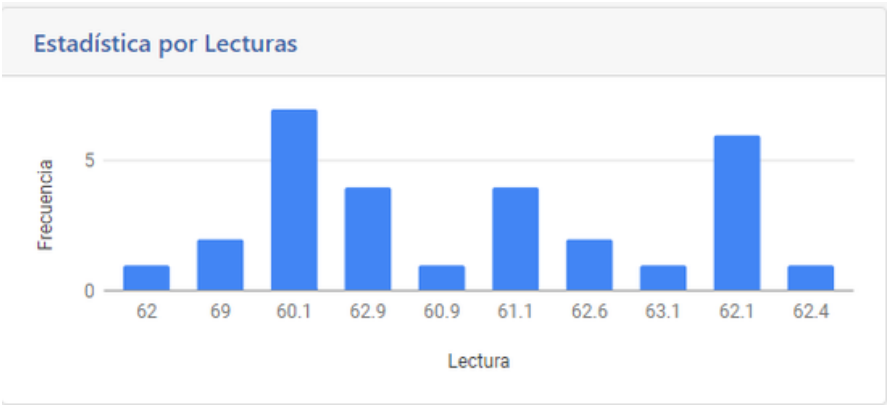


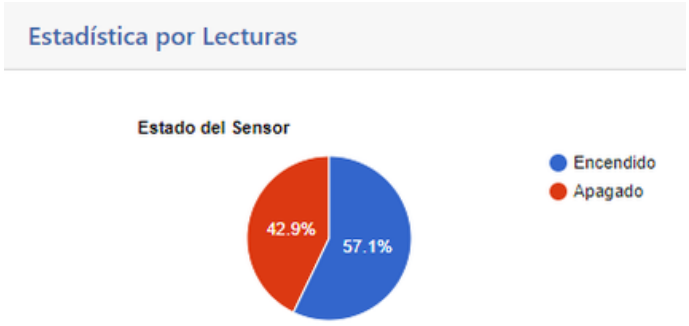
Gráfico "Estadísticas", se genera de acuerdo a las últimas lecturas del sensor.

El gráfico "Estadística por Lecturas", contabiliza las veces que cada lectura ha sido registrada. Por ejemplo: El valor 60.1 se ha registrado 6 veces en la última hora.

Este gráfico se genera para los sensores: humedad, temperatura y punto de rocío.



El gráfico tipo pastel se muestra en los sensores: movimiento, ventana y humo. A diferencia del gráfico anterior, presenta el porcentaje de estado 'encendido/apagado' del sensor según las últimas lecturas registradas.



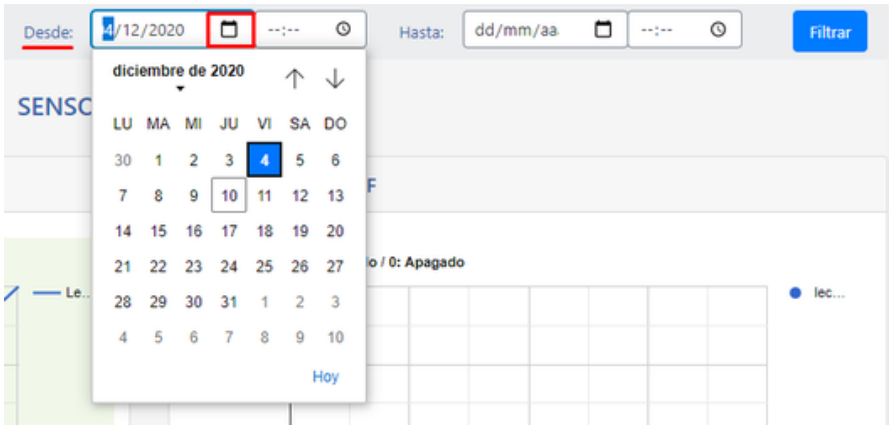
Filtrar información por sensor

1. Diríjase a la cabecera y seleccione el sensor del que requiera visualizar su información.

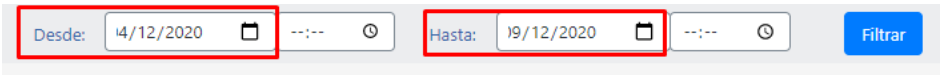
Automáticamente se presentará una pantalla según el sensor seleccionado. Al diferencia de la pantalla "información", los gráficos son generados según las lecturas registradas en las últimas tres horas.

Filtrar información por fecha

Se debe seleccionar una fecha de inicio y una fecha límite para el filtro de información. La fecha puede ingresarse directamente o seleccionarse del calendario.



Ingrese una fecha inicial y una final, verificando que las fechas ingresadas sean correctas. Para continuar haga clic en el botón "Filtrar".



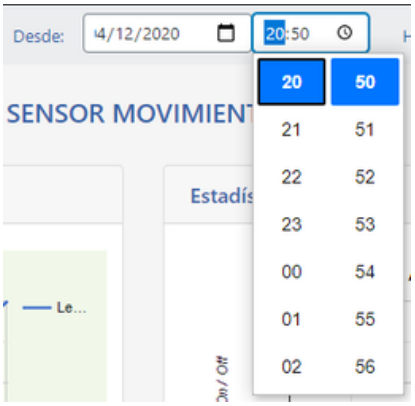
Se mostrarán los resultados según las lecturas registradas en el intervalo de fechas indicadas.

Nota: Cuando la hora no es especificada correspondientemente, se considera la hora de inicio 00:00, y la hora final 23:59.

Filtrar información por fecha y hora

Se debe indicar el intervalo de tiempo, tanto en fecha y hora. Para ello tenga en cuenta que la fecha inicial y la fecha final deben estar acompañadas de una hora respectivamente.

El campo para ingreso de horas puede se llenado manualmente o haciendo clic en el. El formato es 24 horas.



Verifique que los campos de fecha y hora tengan los datos correctos.