



Implementación de la tecnología Blockchain a la cadena de suministros textiles para la empresa

Confecciones Javi.

Villarreal Mendieta, Carlos Ramiro y Yandún Quiroz, Daniel Esteban

Departamento de Ciencias de la Computación

Carrera de Tecnologías de la Información

Trabajo de Integración Curricular, previo a la obtención del título de Ingeniero en Tecnologías de la

Información

Ing. Martínez Espinoza, Mauro Danilo PhD.

13 de julio de 2022

COPYLEAKS

TESIS corregido.docx

Scanned on: 14:5 August 5, 2022 UTC



Identical Words	157
Words with Minor Changes	183
Paraphrased Words	258
Omitted Words	0

MAURO
DANILO
MARTINEZ
ESPINOZA

Digitally signed by
MAURO DANILO
MARTINEZ ESPINOZA
Date: 2022.08.05
10:32:46 -05'00'



Departamento de Ciencias de la Computación

Carrera de Tecnologías de la Información

Certificación

Certifico que el trabajo de integración curricular: **"Implementación de la tecnología Blockchain a la cadena de suministros textiles para la empresa Confecciones Javi"** fue realizado por los señores **Villarreal Mendieta, Carlos Ramiro y Yandún Quiroz Daniel Esteban**, el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizada en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Sangolquí, 06 de agosto de 2022

MAURO
DANILO
MARTINEZ
ESPINOZA

Digitally signed by
MAURO DANILO
MARTINEZ
ESPINOZA
Date: 2022.08.22
14:41:25 -05'00'

Danilo Mauro, Martínez Espinoza

C. C 0603010984



Departamento de Ciencias de la Computación

Carrera de Tecnologías de la Información

Responsabilidad de Autoría

Nosotros, **Villarreal Mendieta Carlos Ramiro**, con cédula de ciudadanía N° 1723226690 y **Yandún Quiroz Daniel Esteban**, con cédula de ciudadanía N° 1714990965, declaramos que el contenido, ideas y criterios del trabajo de integración curricular: **Título: Implementación de la tecnología Blockchain a la cadena de suministros textiles para la empresa Confecciones Javi**, es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 06 de agosto de 2022

.....
Villarreal Mendieta, Carlos Ramiro

C.C.: 1723226690

.....
Yandún Quiroz, Daniel Esteban

C.C.: 1714990965



Departamento de Ciencias de la Computación

Carrera de Tecnologías de la Información

Autorización de Publicación

Nosotros **Villarreal Mendieta, Carlos Ramiro**, con cédula de ciudadanía N° 1723226690 y **Yandún Quiroz, Daniel Esteban**, con cédula de ciudadanía N° 1714990965, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de integración curricular: **Título: Implementación de la tecnología Blockchain a la cadena de suministros textiles para la empresa Confecciones Javi**, en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi/nuestra responsabilidad.

Sangolquí, 06 de agosto de 2022

.....
Villarreal Mendieta, Carlos Ramiro

C.C.: 1723226690

.....
Yandún Quiroz, Daniel Esteban

C.C.: 1714990965

DEDICATORIA

Este trabajo está dedicado a mi familia, en especial a mi madre que siempre soñó verme graduado.

Gracias Blanquita por enseñarme que la adversidad no es pretexto para dejar de estudiar. Mi padre Miguel, desde niño nos trataste de dar la mejor educación y es el mejor regalo que nos has podido dar. A mis hermanos Juan y Tatiana, gracias por confiar en mí, su apoyo fue fundamental para llegar al final de este ciclo.

Si bien mi familia ha sido el estímulo principal para llegar a esta etapa de mi vida, hay personas que han contribuido de cierta manera para alcanzar esta meta. Tal vez no alcance texto para agradecerles a esas personas anónimas, pero si hay espacio suficiente dentro de mi corazón.

Finalmente, este trabajo lo dedico a todas las personas que aún no creen que los sueños pueden lograrse, todo a su debido tiempo y siempre mirada al frente buscando nuevas metas. Una personita que se que va a lograr grandes cosas, recuerda: no es necesario que todo el mundo crea en ti, basta tan solo que una de ellas lo haga para que tú puedas cambiar tu mundo.

Daniel Yandún.

AGRADECIMIENTO

Agradezco a Dios por ponerme en el lugar exacto y en el tiempo exacto, por la nueva oportunidad que me ha regalado.

A mi familia que, a pesar de la distancia ha estado pendiente de mis logros, alegrías y tristezas.

A mis compañeros de Universidad, gracias por aguantar mis malos ratos y también los buenos momentos.

A la Universidad por acogerme y prepararme de la mejor manera para el mundo laboral.

A los ingenieros que contribuyeron con su granito de arena para generar conocimiento en mi persona. Gracias ingeniero Ramiro Delgado por aguantarme estos casi 5 años de Universidad, extrañaré llevarle galletitas o su plato preferido cuycito asado. Al ingeniero Danilo Martínez, por ser nuestro tutor. En este poco tiempo que se pudo compartir con usted de manera virtual fue la pieza clave para llegar a plasmar el conocimiento en el plan de tesis.

A los compañeros que se fueron haciendo amigos: Mafer gracias por no dejarme rendir. Darío, Carlos y Omar poco tiempo, pero grandes recuerdos. Dianita, gracias por ayudarme en un tiempo nuboso de mi vida.

Gracias a la vida por ponerme frente a las personas correctas e incorrectas, he aprendido de las dos partes. Pero sobre todo he aprendido que con dedicación y esfuerzo es posible alcanzar cualquier objetivo, de la mano de Dios y mi familia.

Daniel Yandún.

ÍNDICE DE CONTENIDOS

Resumen	15
Abstract.....	16
Capítulo 1	17
Introducción.....	17
Tema	19
Definición y Justificación del Problema	20
Motivación	22
Objetivos.....	23
<i>Objetivo General</i>	23
<i>Objetivos Específicos</i>	23
Alcance del Proyecto.....	23
Capítulo 2	25
Marco Teórico.....	25
Blockchain.....	25
Cadena de Suministro	26
Trazabilidad.....	27
Smart Contract.....	28
Tecnologías y Herramientas.....	28
<i>Hyperledger Fabric</i>	28

<i>Golang</i>	30
<i>Docker y Docker Compose</i>	30
Visual Studio Code	30
Node JS.....	30
Metodología Investigativa / Modelo de Desarrollo a utilizar	31
<i>Modelo de Kemmis y Carr (1988)</i>	31
<i>Modelo Espiral</i>	32
Áreas del Conocimiento Involucradas	34
<i>Tecnologías Emergentes</i>	34
Cadena de Suministros.....	34
Bases de Datos NoSQL	35
Capítulo 3	36
Diseño de la cadena de suministro	36
Flujo de trabajo de la cadena de suministro.....	37
Definición del Consorcio	38
Ciclo de vida de la Cadena de Suministro	41
<i>Atributos y valores del producto</i>	41
Cambio de estado del producto.....	41
Capítulo 4	45
Creación y ejecución del contrato inteligente o Chaincode	45

	10
Lógica de Transacción	46
Consulta a la Ledger	51
Configuración de la Red de Hyperledger Fabric	53
<i>Configtx</i>	53
Docker Compose	61
Cryptogen.....	67
Creación del Canal de Comunicación entre las Organizaciones	69
Ejecución del Smart Contract.....	70
Capítulo 5	74
Creación del Front End del sistema	74
Servicio EquipmentService.....	76
Servicio QueryService	78
Servicio QueryHystoryByKeyService	80
Realización de Pruebas	81
Proceso de transición.....	85
Capítulo 6	91
Análisis e interpretación de resultados	91
Análisis Cuantitativo de Datos	91
Resultados de la Encuesta.....	92
Capítulo 7	106

Conclusiones, recomendaciones y trabajos futuros	106
Conclusiones	106
Recomendaciones	107
Trabajo Futuro	107
Bibliografía	109

ÍNDICE DE TABLAS

Tabla 1 <i>Organizaciones y usuarios en el consorcio de la red Blockchain</i>	38
Tabla 2 <i>Pregunta 1 (Usuario Inexperto)</i>	96
Tabla 3 <i>Pregunta 2 (Usuario Inexperto)</i>	97
Tabla 4 <i>Pregunta 3 (Usuario Inexperto)</i>	98
Tabla 5 <i>Pregunta 4 (Usuario Inexperto)</i>	99
Tabla 6 <i>Pregunta 5 (Usuario Inexperto)</i>	100
Tabla 7 <i>Pregunta 1 (Usuario Intermedio)</i>	101
Tabla 8 <i>Pregunta 2 (Usuario Intermedio)</i>	102
Tabla 9 <i>Pregunta 3 (Usuario Intermedio)</i>	103
Tabla 10 <i>Pregunta 4 (Usuario Intermedio)</i>	104
Tabla 11 <i>Pregunta 5 (Usuario Intermedio)</i>	105

ÍNDICE DE FIGURAS

Figura 1 <i>Arquitectura Hyperledger Fabric (Roy, 2020)</i>	29
Figura 2 <i>Modelo espiral de ciclos del método de la investigación acción (Naranjo Toro et al., 2019)</i>	31
Figura 3 <i>Modelo espiral (Gutiérrez, 2017)</i>	33
Figura 4 <i>Organizaciones de la red Blockchain</i>	37
Figura 5 <i>Cadena de Suministro</i>	38
Figura 6 <i>Red de Hyperledger Fabric</i>	40
Figura 7 <i>Atributos y valores del producto</i>	41
Figura 8 <i>Producto en estado del fabricante</i>	42
Figura 9 <i>Producto en estado del Distribuidor</i>	43
Figura 10 <i>Producto en estado del Punto de Venta</i>	44
Figura 11 <i>Organizaciones definidas en cryptogen</i>	68
Figura 12 <i>Generación del material criptográfico</i>	68
Figura 13 <i>Creación del bloque Génesis</i>	68
Figura 14 <i>Visualización de la creación del canal</i>	70
Figura 15 <i>Visualización de la creación del AnchorPeer de una organización</i>	70
Figura 16 <i>Ejecución del Smart Contract</i>	73
Figura 17 <i>Estructura de la aplicación para la red de Blockchain</i>	74
Figura 18 <i>Vista principal de la aplicación</i>	82
Figura 19 <i>Acceso a la Aplicación</i>	83
Figura 20 <i>Ingreso a la Aplicación</i>	84
Figura 21 <i>Módulo de Logueo</i>	85
Figura 22 <i>Inicio de contenedores</i>	86
Figura 23 <i>Módulo fabricante</i>	87

Figura 24 <i>Guardar producto</i>	87
Figura 25 <i>Consultar producto</i>	88
Figura 26 <i>Búsqueda histórica del producto</i>	88
Figura 27 <i>Cambio de estado del producto al Distribuidor</i>	89
Figura 28 <i>Cambio de estado del producto al Punto de Venta</i>	90

Resumen

En el mundo actual tanto empresas como consumidores deben garantizar la procedencia de cada producto para de evitar posibles fraudes y garantizar que la distribución se realice de manera confiable. La tecnología Blockchain aplicado a la cadena de suministros es una opción tecnológica que permite tener mayor transparencia y descentralización de los datos, lo cual coadyuva a la colaboración entre las partes involucradas y por ende generar mayor confianza al eliminar al grupo intermediario. El presente proyecto, busca implementar la tecnología Blockchain en la cadena de suministro de una empresa textil. Con la finalidad de poder mejorar la trazabilidad de productos desde su punto de elaboración hasta el consumidor final. La implementación de Blockchain en la empresa busca controlar de manera eficiente cada proceso de manufactura a lo largo de la cadena de suministro. Blockchain permite que los diversos actores que se encuentren siendo parte de la red del sistema puedan interactuar de forma dinámica, es decir, que puedan conocer a cada paso como el producto ha ido cambiando de custodio hasta llegar a su destino, esto favorece a que la confianza entre los diferentes participantes aumente. Finalmente, esta tecnología permite asegurar la integridad e inmutabilidad de los datos mediante la utilización de cadenas de bloques que se crean conforme se interactúa con cada participante de la red.

Palabras Clave: cadena de suministro, trazabilidad, blockchain, chaincode, smart contract.

Abstract

Nowadays, both companies and consumers must guarantee the origin of each product, in order to avoid possible fraud and that distribution is carried out reliably. Blockchain technology applied to the supply chain is a technological option that allows greater transparency and decentralization of data, which contributes to mutual collaboration between the parties involved and therefore generates greater trust by eliminating the intermediary group. This project seeks to implement Blockchain technology within the supply chain in a textile company. In order to be able to improve the traceability of products from their point of manufacture to the final consumer. The implementation of Blockchain within the company seeks to efficiently control each manufacturing process throughout the supply chain. Blockchain allow the various actors that are part of the system network to interact dynamically, that is, they can know at each step how the product has been changing custodian until it reaches its destination, this favors the trust between the different participants increases. Finally, this technology ensures the integrity and immutability of the data using blockchains that are created as each participant in the network interacts. Blockchain allow the various actors that are part of the system network to interact dynamically, that is, they can know at each step how the product has been changing custodian until it reaches its destination, this favors the trust between the different participants increases. Finally, this technology ensures the integrity and immutability of the data through the use of blockchains that are created as each participant in the network interacts. Blockchain will allow the various actors that are part of the system network to interact dynamically, that is, they can know at each step how the product has been changing custodian until it reaches its destination, this favors the trust between the different participants increases. Finally, this technology will ensure the integrity and immutability of the data through the use of blockchains that are created as each participant in the network interacts.

Keywords: supply chain, traceability, blockchain, chaincode, smart contract.

Capítulo 1

Introducción

Según (Lin et al., 2019), los sistemas de trazabilidad tradicionales administran los datos de forma centralizada y por medio de intermediarios, lo que resulta en poca eficiencia y susceptibles a manipulación. (Cocco et al., 2021) manifiesta que, Blockchain es un nuevo paradigma en el manejo de datos de forma distribuida, la cual permite dar soluciones efectivas cuando existen varios actores con los cuales se requiere interactuar. Blockchain es una base de datos distribuida que se caracteriza por la descentralización e inmutabilidad en el manejo de datos, con lo cual aumenta la transparencia, seguridad e integridad de la información al momento de conocer la trazabilidad del producto en cada punto de la cadena de suministro. Su inmutabilidad se basa en la creación de cadenas de bloques mediante algoritmos criptográficos, así como mecanismos de consenso entre las partes involucradas dentro de la red. Además, la integración de otras tecnologías que permiten la creación de contratos inteligentes, confirmación de datos, marcas de tiempo permiten asegurar la integridad de la información sin la necesidad de que exista una entidad intermediaria.

Para (Alkhader et al., 2020), la pandemia del coronavirus reveló falencias, limitaciones a lo largo de la cadena de suministros de insumos médicos, por lo cual manifiesta que es indispensable migrar hacia sistemas descentralizados que faciliten conocer la trazabilidad de cada insumo de manera oportuna y eficiente. La tecnología Blockchain debido a sus características de inmutabilidad, descentralización, seguridad y confianza, resulta ser hoy en día el mecanismo más óptimo para realizar el manejo de datos sin intermediarios y de forma segura por medio de la utilización de claves pública y privada, así como el manejo de firmas digitales.

Según (Shahid et al., 2020), el rastreo de productos mediante mecanismos tradicionales no satisface las exigencias que hoy en día se necesitan, debido a que existe un elevado crecimiento en los

datos, los cuales requieren ser procesados de forma rápida y por ende es necesario buscar cadenas de suministros automatizadas que permitan gestionar de mejor manera la información, para ello es indispensable utilizar sistemas descentralizados que transparenten, auditen y aseguren la información sin necesidad de un tercero. Lo que en contraparte ocurre con sistemas centralizados que pueden ser fácilmente manipulados y carecen de transparencia en cada una de sus transacciones. Para poder gestionar la Cadena de Suministro es necesario seguir un conjunto de procesos, los cuales permiten determinar el ciclo de vida de un producto, desde su fabricación hasta su comercialización. Estos procesos conllevan varias etapas, en las cuales se encuentran involucrados varios actores y el tiempo que se requiere para completar este ciclo a veces puede tardar varios días incluso meses. Por lo cual es necesario poder controlar cada etapa, con la finalidad de poder rastrear un producto que carece de calidad en el menor tiempo posible.

La confianza en el consumidor final es fundamental y por ello es necesario que los registros de la cadena de suministro sean entregados de forma precisa y eficiente. Además, se requiere que a lo largo de la cadena de suministro exista transparencia, eficiencia y calidad en la información entregada. Es por ello que, varios gobiernos han aplicado ciertos estándares para controlar la calidad y veracidad de la información de cada producto. Los códigos de barras o incluso etiquetas son comunes en varios productos, los cuales al ser escaneados otorgan información de dicho artículo. Es por ello que, cada producto tiene su cadena de suministro, el cual debe ser rastreado mediante la trazabilidad del mismo. Este proceso a lo largo del camino va a generar mucha información y por ende la complejidad de los sistemas va incrementándose, produciendo retraso en el procesamiento de datos incluso generando pérdidas de los mismos. Esto generalmente se produce en los sistemas tradicionales centralizados, los cuales carecen de transparencia, veracidad y seguridad. Es por ello que es necesario migrar a sistemas descentralizados, que ayuden a gestionar la enorme cantidad de datos que se generan y la alternativa que ha permitido crear nuevos sistemas de trazabilidad para las Cadenas de Suministro ha sido la

implementación de la tecnología Blockchain, la cual ofrece cada una de las características de una red descentralizada.

Según (Salah et al., 2019), Blockchain es una tecnología disruptiva que permite gestionar la cadena de suministro de forma innovadora, mediante la integración de contratos inteligentes, los cuales controlan cada una de las transacciones dentro de la red. La adquisición de datos generalmente se los realiza mediante lectores de códigos de barras o dispositivos de radio frecuencia (RFID). Además, manifiesta que, la tecnología Blockchain se ha vuelto popular gracias a la transparencia, inmutabilidad y seguridad en cada transacción. Y sobre todo que los diversos actores que intervienen en cada transacción pueden visualizar cada una de ellas, con lo cual se incrementa el nivel de confianza en cada punto dentro de la cadena de suministro.

Cada transacción que se genera dentro de la red se almacena dentro de una cadena de bloques, los cuales se asemejan a un libro público de transacciones, el cual se encuentra disponible solo para los participantes de la red. Cada participante puede realizar el seguimiento de cada transacción dependiendo de la configuración establecida en las reglas de negocio dentro del contrato inteligente. Cada bloque contiene el hash del bloque anterior, si existiese algún cambio dentro del bloque el valor hash cambiaría, con lo cual se garantiza la inmutabilidad de cada transacción. La tecnología Blockchain funciona con éxito en base a tres principios básicos: (i) hash criptográfico, (ii) firma digital (basada en claves públicas y privadas), y (iii) mecanismo de consenso distribuido (minería) (Alkhader et al., 2021).

Tema

Implementación de la tecnología Blockchain a la cadena de suministros textiles para la empresa Confecciones Javi.

Definición y Justificación del Problema

Blockchain es una tecnología emergente que puede brindar soluciones a diversas áreas como: académica, financiera, comercial entre otras. Blockchain en conjunto con otras tecnologías como los RFID (Identificador por Radio Frecuencias) permiten crear nuevas estructuras para el manejo descentralizado de datos. Según (Lin et al., 2019), en su artículo asevera que, las cadenas de suministro tradicionales tienen problemas al momento de manipular la información de manera oportuna, ya que se encuentra gestionada de forma centralizada y por ende existe la posibilidad de que exista pérdida e incluso manipulación de la información. Debido al manejo recurrente de documentos físicos que son susceptibles a errores involuntarios. Esto ocasiona pérdidas de tiempo, las cuales se traducen en un manejo ineficiente de los recursos de la empresa, sin tomar en cuenta que cada documento puede ser alterado y ocasionar varios inconvenientes, como evasión de responsabilidades e incluso pérdidas no declaradas.

Según Avivah Litan, distinguido analista y vicepresidente de investigación de Gartner manifiesta: “Estamos siendo testigos de muchos desarrollos en la tecnología Blockchain que cambiarán el patrón actual. Para 2023, las plataformas Blockchain serán escalables, interoperables y admitirán la portabilidad de contratos inteligentes y la funcionalidad de cadena cruzada. También se van a respaldar transacciones privadas confiables con la confidencialidad de datos requerida. En conjunto, estos avances tecnológicos nos acercarán mucho más a la cadena de bloques convencional y la web descentralizada, también conocida como Web 3.0” (*Gartner 2019 Hype Cycle Shows Most Blockchain Technologies Are Still Five to 10 Years Away From Transformational Impact*, 2019).

Según Meyrick, la implementación de la tecnología Blockchain en la cadena de suministro posibilita crear registros inmutables y duraderos, consiguiendo un ahorro en el procesamiento de documentos de hasta la quinta parte del costo de transporte (Meyrick, 2018). Mediante esta tecnología

es factible conocer el proceso de la cadena de distribución de un producto que ha partido desde la locación A y posteriormente se lo procesó en la locación B. Permitiendo, la elaboración de trazabilidad del producto, evitando de esta manera que existan posibles alteraciones al mismo y a su vez otorgando responsabilidades a cada custodio.

Un ejemplo concreto de los beneficios que proporciona la implementación de la tecnología Blockchain en el manejo de la cadena de suministro es la existencia de alguna enfermedad transmitida por alimentos o medicamentos en mal estado, encontrar su origen demanda mucho tiempo. Es por ello que, teniendo un mejor manejo de la cadena de suministro y conociendo la trazabilidad del producto, se procede a descartar los alimentos en mal estado de manera efectiva y en beneficio de la salud del consumidor. La cadena de alimentos Walmart en EE. UU puso a prueba la implementación de la tecnología Blockchain mediante Hyperledger Fabric, con la finalidad de conocer la trazabilidad de ruta de mangos vendidos en sus tiendas y la importación de carne proveniente de China. Los resultados obtenidos permitieron generar certificados de autenticidad del producto de importación y por ende disponer de una mayor confianza en la calidad del mismo. Mientras tanto, en lo referente a los mangos, se pudo disminuir el rastreo de origen de 7 días a tan solo 2.2 segundos (Hyperledger, 2019).

Blockchain es una tecnología que puede ser de mucha ayuda tanto en la cadena de suministros como en la trazabilidad de productos y la privacidad de datos. En el área de la salud, Blockchain ha permitido detectar suministros médicos adulterados o caducados. En (*Blockchain, clave durante la pandemia*, 2020), se hace una mención especial acerca de los beneficios que ha traído la implementación de Blockchain para el monitoreo de dispositivos médicos y que al aplicarlo se ha reducido el tiempo de entregas de insumos a tan sólo 24 horas, así como también un 90% en la facturación y un 60% en errores cometidos al procesar órdenes de compras.

En el ámbito textil, la tecnología Blockchain puede aportar de manera significativa en el manejo de trazabilidad de productos, permitiendo conocer su origen y proceso de producción hasta llegar al consumidor final. Adicionalmente, Blockchain permitirá que los diversos actores que interactúan dentro de la red puedan interactuar de forma dinámica, lo cual favorece a que exista mayor confianza entre ellos. Finalmente, esta tecnología permitirá asegurar la integridad e inmutabilidad de los datos mediante la utilización de cadenas de bloques que se irán creando conforme se interactúe con cada participante de la red.

Motivación

En el mundo actual tanto empresas como consumidores deben garantizar la procedencia de cada producto, con la finalidad de evitar posibles fraudes y que la distribución se realice de manera confiable. Esta situación no sólo recae en empresas dedicadas a proveer productos alimenticios, de salud o de higiene, sino en cualquier empresa dedicada a dar un servicio. Es por ello, que es indispensable buscar nuevas alternativas con el manejo de tecnologías disruptivas que permitan mejorar el proceso de control de calidad en cada una de las etapas que requiere la cadena de suministros. Es así que, tanto empresas como consumidores deben asegurar la procedencia de cada producto y es aquí donde la tecnología Blockchain aplicado a la cadena de suministros, toma un papel muy importante para implementar una mayor transparencia y descentralización de los datos mediante la gestión de los sistemas de trazabilidad, los cuales generan un número considerable de datos que deben ser procesados de manera rápida y sin pérdidas. Dentro de la red de Blockchain existen diversos colaboradores y por ende partes involucradas, las cuales pueden acceder a las transacciones de manera transparente sin necesidad de grupos intermediarios. Es por ello que la aplicación de la tecnología Blockchain dentro de una empresa textil podría ayudar a gestionar de forma eficiente y segura el proceso de producción del producto terminado hasta llegar al consumidor final.

Objetivos

Objetivo General

Implementar la tecnología Blockchain para supervisar la trazabilidad de la cadena de suministros textiles en la empresa Confecciones Javi.

Objetivos Específicos

- Analizar el funcionamiento tanto de la tecnología Blockchain como de otras herramientas que trabajan en conjunto, mediante un análisis literario de los artículos relacionados, con la finalidad de implementarla en la trazabilidad de la cadena de suministros en la empresa textil Confecciones Javi.
- Implementar un Smart Contract que permita a los participantes conocer la trazabilidad de los productos textiles en la cadena de suministros de la empresa Confecciones Javi, mediante la elaboración del chaincode dentro de la red Blockchain.
- Evaluar la aplicabilidad de la tecnología Blockchain, mediante la realización de una encuesta pre-post al personal de bodega dentro de la empresa textil, con la finalidad de determinar si existe mejora en la trazabilidad de ruta de los productos y afines.

Alcance del Proyecto

El presente proyecto busca facilitar la comprensión de la tecnología Blockchain en la cadena de suministros, permitiendo determinar las ventajas y desventajas de su implementación en el área textil. Esto posibilita conocer su funcionamiento, arquitectura, estructura y relación con otras tecnologías en cada una de las etapas que conlleva su desarrollo, de esta manera la curva de aprendizaje se incrementa, generando en los participantes nuevas experiencias que aportan a su perfil profesional. Una vez que la tecnología y los nuevos conceptos hayan sido comprendidos mediante la realización de

diversas pruebas, se procederá a la implementación de Blockchain en la cadena de suministro. La cual inicia con el planteamiento de las reglas de negocio para cada organización, en las cuales se detalla la función de cada participante dentro de la red. Posteriormente, se crean los contratos inteligentes o chaincode que servirán para la gestión de actividades de cada organización.

La implementación de la tecnología Blockchain en el área de manufactura o textil, con el fin de determinar la trazabilidad de los productos de una pequeña empresa dedicada a la distribución y fabricación de medias, busca evitar varios problemas que pueden surgir en el proceso de elaboración y distribución, tales como: productos con fallas o baja calidad, envíos duplicados, entre otros. Con Blockchain se busca disminuir todos estos riesgos que pueden afectar tanto a una empresa como al consumidor, además de ayudar en el manejo descentralizado de datos y por ende los costos de manejo de documentación se verán disminuidos. Adicionalmente, el sistema de trazabilidad permite conocer en el menor tiempo los orígenes y manejos de cada insumo.

Capítulo 2

Marco Teórico

Blockchain

Blockchain es una tecnología emergente creada en el 2008 por el pseudónimo Satoshi Nakamoto, con la finalidad de descentralizar el manejo de dinero a través de criptomonedas (García, 2020). Según (García, 2020), esta tecnología podría ayudar a mejorar el desenvolvimiento entre etapas de la cadena de suministro, generando mayor confianza entre cada integrante de la misma, ya que durante todo el proceso no existe la intervención de una tercera persona que pueda alterar y producir cambios que lleguen a afectar a la parte interesada.

De acuerdo a (Delgado, 2019) y (Villameriel, 2020), Blockchain está caracterizado por los siguientes factores:

- **Base de datos distribuida:** En Blockchain, no existe una autoridad como tal que posea toda la información y transacciones que se realizan dentro de ella, cada participante es responsable de lo que maneja y todo lo que sucede en ella, estos participantes dentro de Blockchain son conocidos como nodos y cada vez que un nuevo nodo desea integrarse a esta red, todas las transacciones que han sido realizadas, son descargadas a su sistema (Pandey & Litoriya, 2020).
- **Segura:** Sin duda alguna, Blockchain ofrece una total seguridad a la comunidad que participa en ella, debido a que toda la información que se procesa es cifrada y normalmente cada transacción es identificada por un valor hash por lo cual es muy difícil de que se corrompa.
- **P2P (Peer to Peer):** Esto hace referencia a que no existe un intermediario como tal para que se lleve a cabo un proceso, únicamente los participantes o la parte interesada interviene.

- **Transparente:** Este término va a variar dependiendo de la red en la que se esté trabajando, en el caso de ser pública la transparencia de las transacciones en blockchain es total, es decir que, cada usuario tiene acceso a ellas y podrán identificar la procedencia de las mismas, esto se debe a que como se mencionó anteriormente, cada vez que un nuevo usuario se registra obtiene una copia de todas las transacciones que se han realizado con anterioridad.

(Delgado, 2019) y (Villameriel, 2020), también han planteado que la tecnología Blockchain puede ser de tres tipos, los cuales serán mencionados a continuación:

- **Pública:** Una red Blockchain pública, es aquella que todo usuario tiene acceso, al unirse a ella tiene acceso completo a todas las transacciones que se han realizado y hacer uso de ellas. La gran ventaja de una Blockchain pública es que, al ser distribuida, es más difícil de ser atacada debido a que cada nodo tiene un respaldo de todo.
- **Privada:** A diferencia de la pública, en una blockchain privada existe un límite de participantes que puede acceder a ella y también los permisos son distintos para cada uno, además de que estas son controladas por una entidad.
- **Federada:** Una blockchain federada o de consorcio, normalmente pertenece a entidades bancarias, gobiernos u organizaciones que realizan muchas transacciones, las cuales se realizan de forma más rápida y privada. En este tipo de blockchain al igual que en una privada, existe un número limitado de acceso a usuarios.

Cadena de Suministro

Una cadena de suministro o Supply Chain, es un conjunto de diversas actividades, conformada por todas las partes involucradas en el proceso de venta de un producto, con varios propósitos, entre ellos, satisfacer las necesidades y requerimientos de un cliente, evitar pérdidas que pueden afectar tanto a la empresa como al usuario, mejorar y respetar los tiempos de entrega de un producto, entre

otros (Arcia, 2018). Una cadena de suministro, cuenta con 3 fases muy importantes durante todo el proceso, las cuales son (Estaún, 2020):

- **Suministro:** Esta fase consiste en la adquisición de la materia prima, es decir, cómo, cuándo y dónde conseguirla.
- **Fabricación:** Hace referencia a la transformación de la materia prima a un producto final.
- **Distribución:** Una vez que el producto esté listo, se busca distribuirlo a distintos puntos de venta, con el fin de que llegue al consumidor.

Dentro de los elementos fundamentales que abarca la cadena de suministro, para garantizar que todo el proceso sea llevado con total normalidad y sin ningún inconveniente de por medio, se encuentran (Arcia, 2018):

- **Proveedores:** Son los encargados de la distribución de un bien material a distintos puntos de comercialización.
- **Transporte:** Movilizan tanto la materia prima como el producto final a un lugar en específico.
- **Fabricantes:** Transformar la materia prima en un producto que sea consumible.
- **Cliente:** Son la parte más importante de una cadena de suministros, ya que se encargan de evaluar el producto y verificar si cumplen los requerimientos solicitados.
- **Comunicación:** Es un aspecto muy importante ya que, al existir una buena comunicación entre cada parte interesada del proceso, hay un mejor resultado.
- **Tecnología:** Permite una optimización en las actividades que se realicen y así también disminuir el tiempo de cada una.

Trazabilidad

De acuerdo a (Keyence, 2018), la trazabilidad se define como la capacidad de rastrear todo el proceso que conlleva la venta de un producto, desde la adquisición de la materia prima, su

transformación, hasta su consumo, con el fin de conocer cuándo, cómo, dónde y por quién fue elaborado cierto producto. Sin embargo, la norma ISO 9000, define a la trazabilidad como la habilidad de rastrear la historia, aplicación, uso y ubicación de un producto o sus características a través de información almacenada.

Smart Contract

Un Smart Contract o contrato inteligente, son acuerdos que pueden ser desarrollados en distintos lenguajes de programación, que tienen el propósito de facilitar y asegurar que se cumpla un acuerdo entre dos o más partes de manera automatizada y sin la intervención de una tercera persona durante todo el proceso, provocando que existan inconvenientes tal como suele suceder en contratos realizados de forma manual donde las alteraciones de estos documentos se pueden llevar a cabo de manera sencilla. Todo el proceso de autenticación y cumplimiento de lo establecido por las partes involucradas es verificado con blockchain (Pablo et al., 2019).

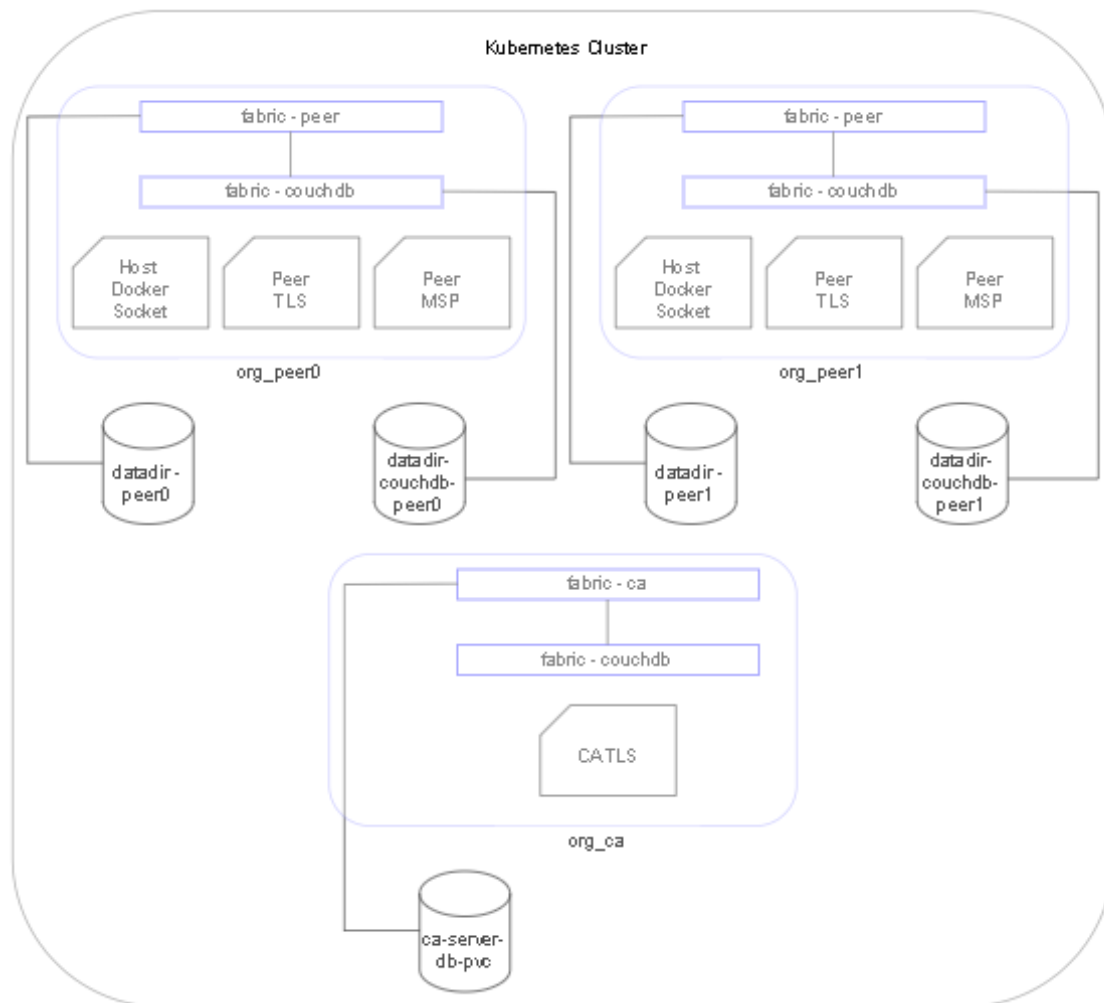
Tecnologías y Herramientas

Hyperledger Fabric

Hyperledger Fabric está diseñado como base para el desarrollo de aplicaciones o soluciones con una arquitectura modular. Hyper Ledger Fabric permite que los componentes, como los servicios de consenso y membresía, sean plug-and-play. Su diseño modular y versátil satisface una amplia gama de casos de uso de la industria. Ofrece un enfoque único de consenso que permite el rendimiento a gran escala y, al mismo tiempo, preserva la privacidad (Hyperledger Fabric, 2020).

Figura 1

Arquitectura Hyperledger Fabric (Roy, 2020)



En la Figura 1, se puede observar la arquitectura básica de Hyperledger. Esta tecnología funciona sobre un conjunto de contenedores, los cuales van a interactuar con cada organización denominada peer. Cada peer tiene sus propias reglas de negocio, Docker, certificados de seguridad que actúan a su vez sobre una base de datos CouchDB. Una vez definidos los contenedores de cada organización estos interactúan entre sí mediante un contrato inteligente o chaincode, en el cual se han establecido previamente los participantes de la red y sus atributos.

Golang

Go es un lenguaje de programación de código abierto que facilita la creación de software simple, confiable y eficiente (Go, 2007).

Hyperledger utiliza el protocolo GRPC de Golang para ejecutar cada transacción, además, mediante GO se crean las funciones para crear los contratos inteligentes o chaincode.

Docker y Docker Compose

Docker es una plataforma de software que permite crear, probar e implementar aplicaciones rápidamente. Docker empaqueta software en unidades estandarizadas llamadas contenedores que incluyen todo lo necesario para que el software se ejecute, incluidas bibliotecas, herramientas de sistema, código y tiempo de ejecución (Amazon AWS, 2020).

Docker Compose es una herramienta para definir y ejecutar aplicaciones Docker de varios contenedores. Compose, usa un archivo YAML para configurar los servicios de su aplicación. Luego, con un solo comando, crea e inicia todos los servicios desde su configuración (Docker, 2020) .

Visual Studio Code

Visual Studio Code es un editor de código fuente ligero pero potente que se ejecuta en su escritorio y está disponible para Windows, macOS y Linux. Viene con soporte incorporado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes (como C++, C#, Java, Python, PHP, Go) y tiempos de ejecución (como .NET y Unity) (Microsoft Azure, 2020).

Node JS

Es un entorno de tiempo de ejecución de JavaScript en tiempo real el cual contiene todo lo necesario para ejecutar un programa que ha sido escrito en JavaScript. Fue desarrollado por los

creadores de JavaScript con el fin de poder ejecutarse en un ordenador y no únicamente en un navegador. Node.js permite la creación de aplicaciones de red rápidas, ya que tiene la capacidad de manejar una gran cantidad de conexiones al mismo tiempo con alto nivel de rendimiento, lo que significa que proporciona una alta escalabilidad (Lucas, 2019).

Metodología Investigativa / Modelo de Desarrollo a utilizar

Modelo de Kemmis y Carr (1988)

El modelo de Kemmis y Carr, fue diseñado anteriormente por Lewin en 1946 y desarrollado posteriormente por Kolb en 1984 (Naranjo Toro et al., 2019), el cual es empleado normalmente para cuestiones escolares, desarrollo profesional, para la mejora de programas y planificación (Serrano, n.d.). Este modelo se encuentra organizado por dos ejes, el primero contiene la acción reflexión y el segundo eje, está compuesto por la planificación y observación, estos cuatro son parte de las fases del mismo, las cuales se repiten en cada ciclo, pero con un diferente propósito. En la Figura 2, se puede observar el diseño del modelo y cómo se encuentran organizadas sus fases:

Figura 2

Modelo espiral de ciclos del método de la investigación acción (Naranjo Toro et al., 2019)



A continuación, se describen las acciones que se deben realizar en cada una de las fases planteadas en el modelo:

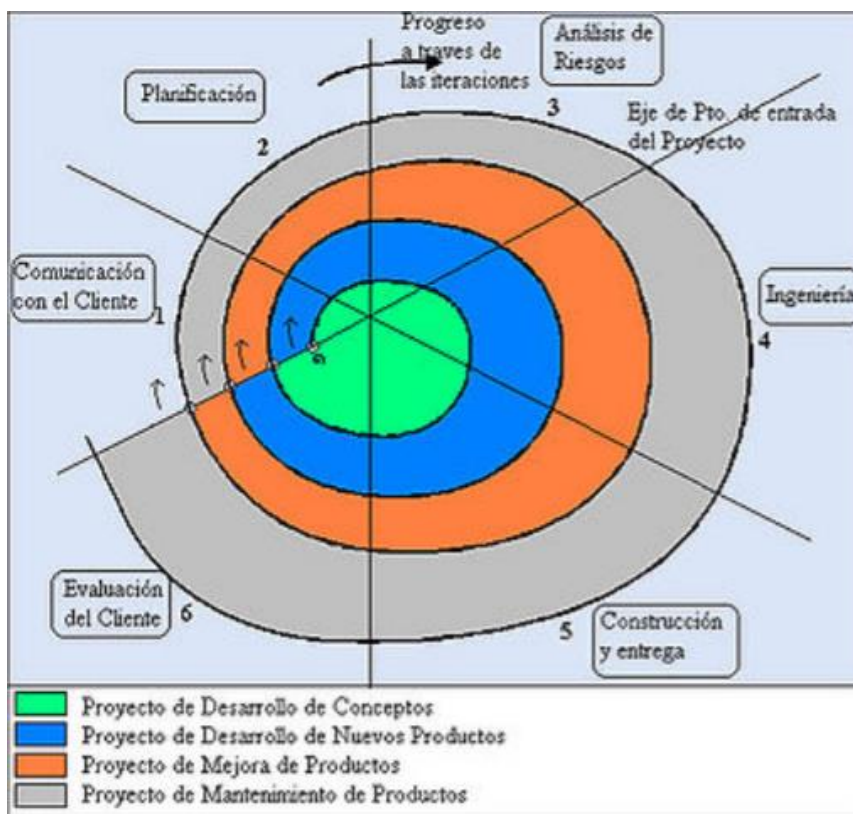
- **Planificar:** Se diagnostica y plantea el problema de investigación a partir de un conocimiento previo del investigador con el propósito de guiar todo el proceso
- **Acción:** Para esta fase, es necesario tener establecido anteriormente un cronograma que permita definir los tiempos necesarios tanto para la revisión literaria como para definir los responsables para realizar cada una de las acciones planteadas.
- **Observación:** El investigador se encarga de documentar todo el proceso de la investigación-acción para la generación y recolección de datos a través de las distintas técnicas e instrumentos cualitativos.
- **Reflexión:** En esta fase realiza una elaboración de informe o se replantea alguna de las fases.

Modelo Espiral

El modelo espiral, propuesto por Barry Boehm en 1986, es una combinación entre el modelo lineal o cascada y el modelo iterativo o basado en prototipos (*El modelo de desarrollo en espiral como mezcla de cascada e iterativo*, 2019). Este modelo está basado en entregas incrementales, en donde los entregables de las primeras iteraciones pueden ser un prototipo de lo que será el producto final o de alguna funcionalidad del mismo y las siguientes iteraciones ya son entregables de un producto más completo (Esparza Maldonado et al., 2015). De igual forma, el modelo está dividido en números de actividades llamados “regiones de tareas”, que por lo general son de 3 a 6, tal como se puede observar en la Figura 3 (Gutiérrez, 2017).

Figura 3

Modelo espiral (Gutiérrez, 2017)



A continuación, se describirán las regiones de tareas, para conocer el proceso que se debe realizar en cada una de ellas:

- En la primera región, se establece una comunicación entre el desarrollador y el cliente para conocer los requerimientos que desea que cumpla el software.
- En la segunda región, se procede con la planificación del proyecto que vendría a ser la definición de recursos (humanos y tecnológicos), el tiempo para realizar cada actividad, entre otros.
- En cuanto a la tercera región, se hace una evaluación para establecer tanto los riesgos técnicos como la gestión del proyecto.

- En la cuarta región se procede a realizar un prototipado con las diferentes funcionalidades que tendrá el software.
- En la región 5, se realizan tareas para la construcción del software, pruebas y para la documentación (manual de usuario).
- Por último, en la sexta región el usuario hace una evaluación del software para verificar que cumpla con los requerimientos y que no exista ningún error.

Áreas del Conocimiento Involucradas

Tecnologías Emergentes

Una tecnología emergente es una herramienta que en un futuro creará un gran impacto en el mercado, estas surgen de alguna necesidad actual, sin embargo, algunas de ellas se encuentran en su fase inicial de desarrollo, debido a que aún necesitan ser pulidas en ciertos aspectos para que puedan llegar a cumplir su función. Aún así, llegando a corregir o implementar dichas características, debido al gran avance de la tecnología estas deben irse adaptando a las mismas. Las tecnologías emergentes están definidas por cinco atributos de acuerdo a (Rotolo et al., 2015), las cuales son: novedad radical, crecimiento rápido, coherencia, impacto destacado e incertidumbre y ambigüedad.

Cadena de Suministros

La cadena de suministros o Supply Chain es un proceso en el cual se desarrollan diferentes actividades que son necesarias para llevar a cabo la venta de un producto sin ningún inconveniente de por medio. Una cadena de suministro tiene como propósito entregar productos de calidad en el menor tiempo y costo posible, para satisfacer todas las necesidades que un cliente requiera (Arcia, 2018).

Bases de Datos NoSQL

Las bases de datos NoSQL proporcionan una variedad de modelos de datos, como clave-valor, documentos y gráficos, que están optimizados para el rendimiento y la escala. Las bases NoSQL están diseñadas para varios patrones de acceso a datos que incluyen aplicaciones de baja latencia, permitiendo realizar análisis sobre datos semiestructurados (AWS, 2018).

Capítulo 3

Diseño de la cadena de suministro

Según (Shahid et al., 2020), el rastreo de productos mediante mecanismos tradicionales no satisface las exigencias que hoy en día se necesitan, debido a que existe un elevado crecimiento en los datos, los cuales requieren ser procesados de forma rápida y por ende es necesario buscar cadenas de suministros automatizadas que permitan gestionar de mejor manera la información, para ello es indispensable utilizar sistemas descentralizados que transparenten, auditen y aseguren la información sin necesidad de un tercero. Lo que en contraparte ocurre con sistemas centralizados que pueden ser fácilmente manipulados y carecen de transparencia en cada una de sus transacciones. Para poder gestionar la Cadena de Suministro es necesario seguir un conjunto de procesos, los cuales permiten determinar el ciclo de vida de un producto, desde su fabricación hasta su comercialización. Estos procesos conllevan varias etapas, en las cuales se encuentran involucrados varios actores y el tiempo que se requiere para completar este ciclo a veces puede tardar varios días incluso meses. Por lo cual es necesario poder controlar cada etapa, con la finalidad de poder rastrear un producto que carece de calidad en el menor tiempo posible.

La confianza en el consumidor final es fundamental y por ello es necesario que los registros de la cadena de suministro sean entregados de forma precisa y eficiente. Además, se requiere que a lo largo de la cadena de suministro exista transparencia, eficiencia y calidad en la información entregada. Es por ello que, varios gobiernos han aplicado ciertos estándares para controlar la calidad y veracidad de la información de cada producto. Los códigos de barras o incluso etiquetas son comunes en varios productos, los cuales al ser escaneados otorgan información de dicho artículo. Es por ello que, cada producto tiene su cadena de suministro, el cual debe ser rastreado mediante la trazabilidad del mismo. Este proceso a lo largo del camino va a generar mucha información y por ende la complejidad de los

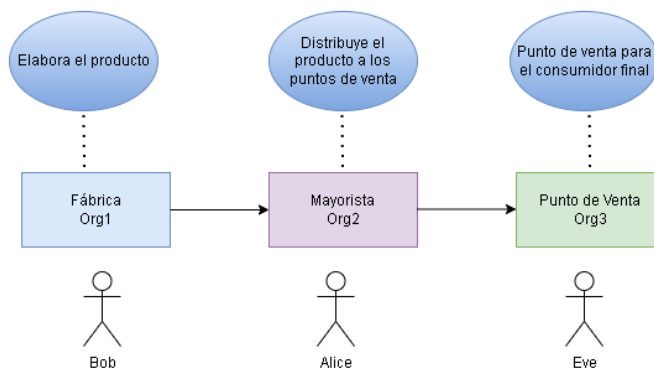
sistemas va incrementándose, produciendo retraso en el procesamiento de datos incluso generando pérdidas de los mismos. Esto generalmente se produce en los sistemas tradicionales centralizados, los cuales carecen de transparencia, veracidad y seguridad. Es por ello que es necesario migrar a sistemas descentralizados, que ayuden a gestionar la enorme cantidad de datos que se generan y la alternativa que ha permitido crear nuevos sistemas de trazabilidad para las Cadenas de Suministro ha sido la implementación de la tecnología Blockchain, la cual ofrece características de transparencia, flexibilidad, seguridad en entornos donde existen varios actores involucrados garantizando que las transacciones sean inviolables y a la vista de todos.

Flujo de trabajo de la cadena de suministro

En la Figura 4, se muestra las organizaciones que participan dentro de la red Blockchain. Cada una de las cuales tiene definido su rol dentro del proceso de manufactura del producto textil hasta llegar al consumidor final.

Figura 4

Organizaciones de la red Blockchain

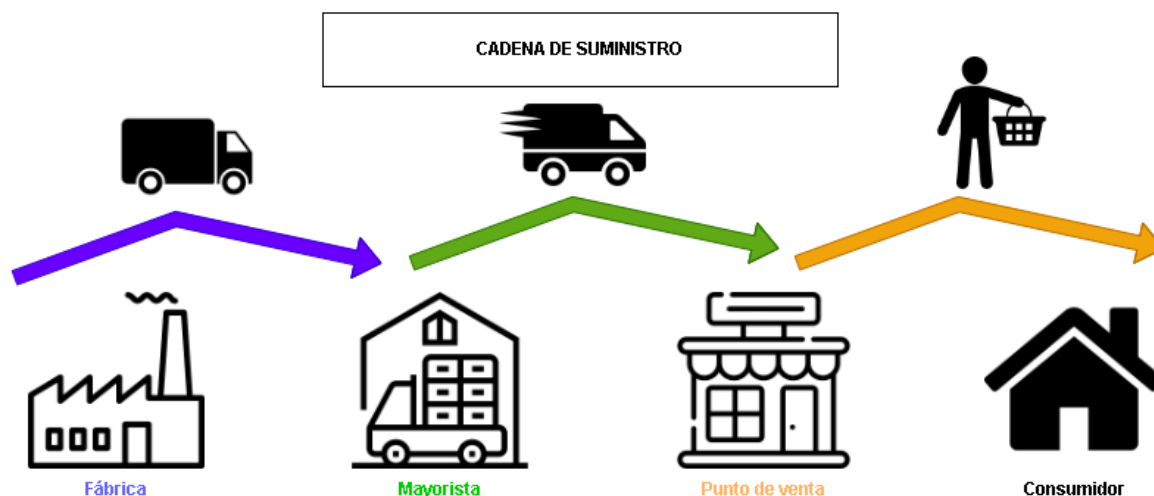


En este proceso se encuentran involucrados tres actores (ver Figura 5): fabricante, mayorista y punto de venta. El fabricante produce el producto y es enviado al mayorista. El mayorista recibe dicho

producto lo registra y envía a los diferentes puntos de venta. Finalmente, el punto de venta cierra el ciclo de vida de la cadena de suministro.

Figura 5

Cadena de Suministro



Definición del Consorcio

En la sección anterior se ha determinado la existencia de tres participantes: fabricante, mayorista y punto de venta. Las cuales van a interactuar entre sí para generar la cadena de suministro. Los participantes o miembros de la red en su conjunto formarán un consorcio. Cada miembro podrá realizar transacciones definidas en el contrato inteligente. En la Tabla 1, se muestran las organizaciones y usuarios del consorcio de la red.

Tabla 1

Organizaciones y usuarios en el consorcio de la red Blockchain

Nombre de la Organización	Usuario	MSP	Peer
Fábrica	Bob	Org1MSP	peer0.org1.example.com

Nombre de la Organización	Usuario	MSP	Peer
Distribuidor	Alice	Org2MSP	peer0.org2.example.com
Punto de Venta	Eve	Org3MSP	peer0.org3.example.com

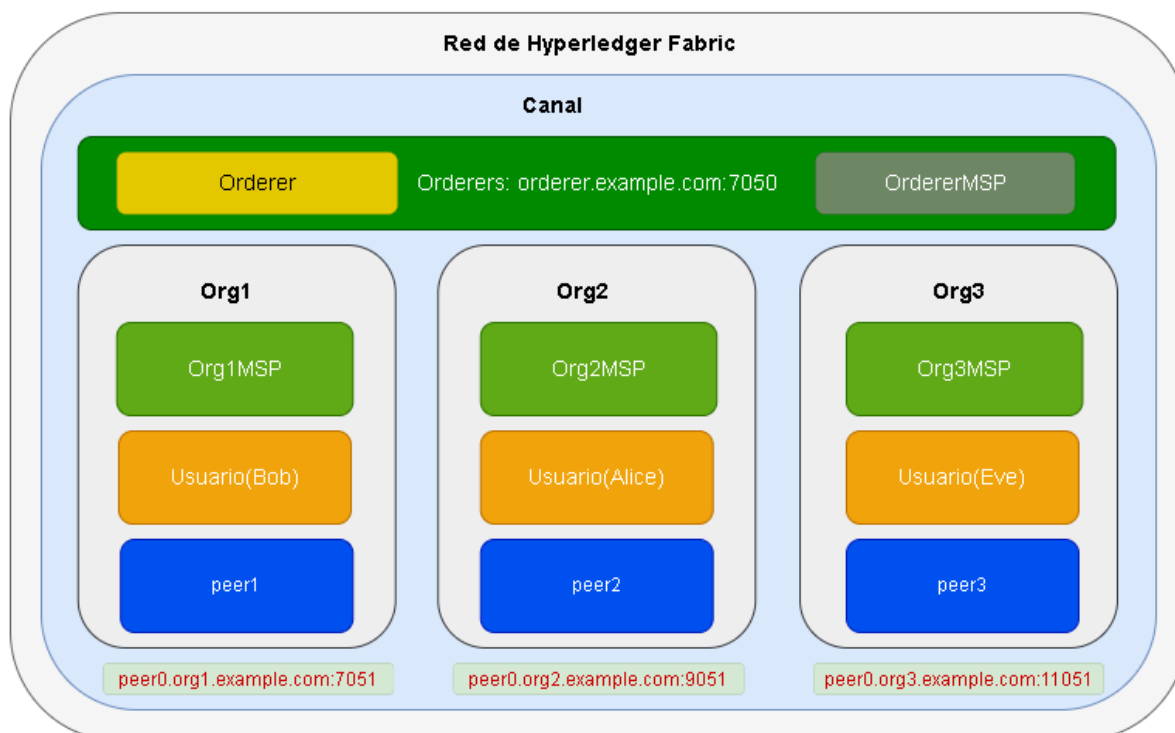
Cada una de las organizaciones tendrá un usuario, un MSP y un peer. Para la organización del fabricante se tiene que su usuario podría ser Bob, su ID para definir los servicios de membresía es Org1MSP y finalmente, se requerirá el nombre del host que está definido por peer0.org1.example.com. Para el caso del mayorista, el usuario es Alice, su ID para definir los servicios de membresía es Org2MSP y finalmente, se requerirá el nombre del host que está definido por peer0.org2.example.com. Para el caso del punto de venta, el usuario es Eve, su ID para definir los servicios de membresía es Org3MSP y finalmente, se requerirá el nombre del host que está definido por peer0.org3.example.com.

En la Figura 6, se muestra la red de Hyperledger Fabric. Esta se encuentra compuesta por cuatro organizaciones: Org1, Org2, Org3 y Orderer; en su conjunto van a conformar el consorcio de la red. Cada organización deberá ser autorizada y controlada dentro de la red, para ello es necesario que cada participante obtenga un certificado de autenticación emitido por la autoridad certificadora, que en este caso es Membership Service Provider(MSP). Una vez obtenidos cada uno de los certificados los usuarios de cada organización podrán interactuar con la red mediante el peer creado para cada una de ellas(peer0.org1.example.com:7051). En la sección de Orderer se especifica el mecanismo de consenso, en nuestro caso se lo realizó vía Raft. Debido a que el consenso vía SOLO es utilizado solo para la realización en ambientes de pruebas y KAFKA requiere de archivos adicionales, lo que lo hace más pesado para implementarse. Además, se especifica la ruta donde se encuentran los certificados de transacción tanto para el cliente como para el servidor.

Los servicios que se requieren son: orderer y un peer para cada organización. Para el caso del Orderer se indica el nombre del contenedor(orderer.example.com) y la imagen de Docker que se utiliza para levantar ese contenedor, haciendo uso de los recursos predefinidos de Hyperledger Fabric. Para que las diferentes organizaciones se puedan identificar dentro de la red, es necesario que tengan un certificado digital. Las autoridades certificadoras (CA) se encargan de gestionar dichos certificados X.509, para el presente proyecto se utiliza Cryptogen, que es la herramienta proporcionada por Hyperledger Fabric para crear los certificados para cada organización, así como se encarga de generar el material criptográfico. Es necesario un canal de comunicación entre los diferentes peers y organizaciones, para ello se utilizó la herramienta configtxgen proporcionada por Hyperledger Fabric. Y a su vez se crearon las transacciones de los AnchorPeer, que son los peer que se comunican con cada organización, por lo cual cada una de ellas debe tener su AnchorPeer dentro de red.

Figura 6

Red de Hyperledger Fabric



Ciclo de vida de la Cadena de Suministro

Atributos y valores del producto

Cada producto dentro de la Cadena de Suministro debe tener atributos y valores, los cuales permiten identificarlos de forma rápida y oportuna. En la Figura 7, se puede observar las especificaciones de cierto producto: ID, nombre, propietario, propietario actual, propietario previo, fecha de elaboración y fecha de actualización. Esta última hace referencia a la fecha de la última modificación.

Figura 7

Atributos y valores del producto

Id Búsqueda	78722344
Productor	Confecciones Javi
Id Producto	78722344
Nombre Producto	medias nk
Propietario	Confecciones Javi
Tipo de Propietario Previo	fabricante
Tipo de Propietario Actual	fabricante
Fecha de Creacion	7/22/2022, 10:27:26 AM
Fecha de Actualizacion	7/22/2022, 10:27:26 AM

Cambio de estado del producto

El producto desde su etapa inicial(fabricante) hasta su etapa final (punto de venta), va a cambiar de estado o propietario en cada una de las etapas dentro de la Cadena de Suministro. En la Figura 8 se muestra que, el fabricante ha elaborado un producto, el cual tiene un ID: 78722344. El nombre del fabricante es Confecciones Javi y por ende es el actual propietario o custodio del producto. En el campo Tipo de Propietario Previo y Tipo de Propietario Actual tienen como valor al fabricante, ya que este es quien ha iniciado el proceso de producción del producto. Además, se observa que el producto tiene un nombre descriptivo (medias nk) y una fecha de creación, la cual es automáticamente ingresada por el sistema al guardar el estado del producto. Esta fecha permanecerá inalterable una vez guardados los

cambios realizados en la ledger. Si existiera alguna modificación dentro de los campos previamente guardados referentes a ese producto, la fecha anterior permanecerá intacta y en el campo de Fecha de Actualización se verá reflejado la fecha posterior a la Fecha de Creación.

Figura 8

Producto en estado del fabricante



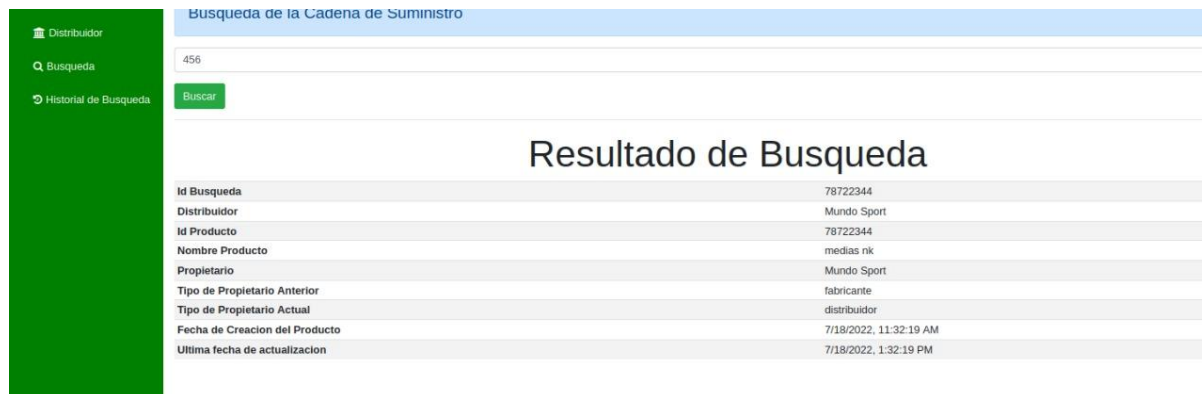
The screenshot shows a web application interface for a factory (Fabrica). The sidebar on the left contains navigation options: 'Producto', 'Buscar', and 'Historial de Búsqueda'. The main content area is titled 'Consulta de datos de la Cadena de Suministro'. A search bar contains the ID '78722344' and a 'Buscar' button. Below the search bar, the results are displayed in a table.

Resultados de Búsqueda	
Id Búsqueda	78722344
Productor	Confecciones Javi
Id Producto	78722344
Nombre Producto	medias nk
Propietario	Confecciones Javi
Tipo de Propietario Previo	fabricante
Tipo de Propietario Actual	fabricante
Fecha de Creacion	7/22/2022, 10:27:26 AM
Fecha de Actualizacion	7/22/2022, 10:27:26 AM

Una vez que el producto ha estado en propiedad del fabricante, este debe cambiar de estado o propietario hacia el Distribuidor. En la Figura 9 se observa que, ahora el nuevo propietario es Mundo Sport. El tipo de propietario anterior era el fabricante y el tipo de propietario actual es ahora el distribuidor, en este caso Mundo Sport. Tanto la fecha de creación del producto como la fecha actual serán registradas en el sistema al momento de guardar los cambios respectivos. Cabe mencionar que, el campo de Id Producto se mantiene invariable, ya que se trata del mismo producto y sólo ha cambiado de propietario.

Figura 9

Producto en estado del Distribuidor



Id Busqueda	78722344
Distribuidor	Mundo Sport
Id Producto	78722344
Nombre Producto	medias nk
Propietario	Mundo Sport
Tipo de Propietario Anterior	fabricante
Tipo de Propietario Actual	distribuidor
Fecha de Creacion del Producto	7/18/2022, 11:32:19 AM
Ultima fecha de actualizacion	7/18/2022, 1:32:19 PM

Una vez que el producto ha estado en propiedad del distribuidor, este debe cambiar de estado o propietario hacia el Punto de Venta. En la Figura 10 se observa que, ahora el nuevo propietario es Store JF. El tipo de propietario anterior era el distribuidor y el tipo de propietario actual es ahora el punto de venta, en este caso Store JF. Tanto la fecha de creación del producto como la fecha actual fueron registradas en el sistema al momento de guardar los cambios respectivos. Cabe mencionar que, el campo de Id Producto se mantiene invariable, ya que se trata del mismo producto y sólo ha cambiado de propietario.

Figura 10

Producto en estado del Punto de Venta

The screenshot shows a web application interface for 'Punto de Venta'. On the left is a green sidebar with navigation options: 'Añadir', 'Recibidor', 'Busqueda', and 'Historial de Busqueda'. The main content area is titled 'Busqueda de la Cadena de Suministro' and contains a search input field with the value '78722344'. Below the input is a green 'Busqueda' button. The search results are displayed under the heading 'Resultado de Busqueda' and are organized into a table with two columns: a label and a value.

Id Busqueda	78722344
Productor	Store JF
Id Producto	78722344
Nombre Producto	medias nk
Propietario	Store JF
Tipo de propietario anterior	distribuidor
Tipo de propietario actual	punto de venta
Fecha de creacion	7/18/2022, 11:32:19 AM
Fecha de modificacion	7/18/2022, 5:32:19 PM

Capítulo 4

Creación y ejecución del contrato inteligente o Chaincode

Un Smart Contract o Contrato Inteligente es un programa que contiene la lógica de negocio, en este se establecen las acciones que podrá realizar cada participante de la red. Hyperledger Fabric permite crear Smart Contract en diferentes lenguajes de programación, para el presente proyecto se utilizará JavaScript.

Para la implementación del Smart Contract se utilizó la librería de Hyperledger Fabric fabric-contract-api, la cual proporciona la interfaz necesaria para crear los Smart Contract requeridos. Esta interfaz requiere de dos clases: Contract y Context. La clase Contract permite crear instancias para el chaincode o contrato inteligente, mientras que la clase Context proporciona el contexto para cada transacción.

```
const { Contract, Context } = require('fabric-contract-api');
```

Definimos un nuevo contrato mediante la clase PharmaLedgerContract que se extiende de la clase Contract, cabe señalar que es aconsejable utilizar nombres distintivos que faciliten la comprensión de cada función. Se creó el objeto de la clase PharmaLedgerContract, el cual es el nombre que recibe nuestro Smart Contract, en este caso org.pln.PharmaLedgerContract.

```
class PharmaLedgerContract extends Contract {
```

```
  constructor() {
```

```
    super('org.pln.PharmaLedgerContract');
```

```
  }
```

Lógica de Transacción

Se define las funciones que rigen la lógica del negocio:

1. Función para crear el producto e introducir esa información en la ledger (makeEquipment)
2. Función para especificar que, el producto ha sido entregado al distribuidor
(wholesalerDistribute)
3. Función que permita indicar que el punto de venta ha recibido el producto (pharmaReceived)

```
async makeEquipment(ctx, manufacturer, equipmentNumber, equipmentName, ownerName)
    async wholesalerDistribute(ctx, equipmentNumber, ownerName)
    async pharmacyReceived(ctx, equipmentNumber, ownerName)
```

Se creó una función asíncrona denominada makeEquipment, la cual requiere de cinco parámetros:

- ctx: hace referencia al contexto de la transacción, se la utiliza para enviar los datos de cada transacción al ledger de la red de Hyperledger Fabric.
- manufacturer: hace referencia al fabricante del producto
- equipmentNumber: es el número identificativo del producto
- equipmentName: nombre del producto
- ownerName: hace referencia al propietario actual del producto

```
async makeEquipment(ctx, manufacturer, equipmentNumber, equipmentName, ownerName) {
    let dt = new Date().toString();

    const equipment = {
        equipmentNumber,
        manufacturer,
        equipmentName,
```

```

    ownerName,
    previousOwnerType: 'MANUFACTURER',
    currentOwnerType: 'MANUFACTURER',
    createDateTime: dt,
    lastUpdated: dt
  };

  await ctx.stub.putState(equipmentNumber, Buffer.from(JSON.stringify(equipment)));
}

```

Mediante la función de JavaScript Date(), se almacenó la fecha del sistema dentro de la variable dt, se la transformó en un String mediante la función toString(). Posteriormente se creó una constante equipment, la cual contiene la información referente al producto: equipmentNumber, manufacturer, equipmentName, ownerName, previousOwnerType, currentOwnerType, createDateTime, lastUpdated. Se observa que en la propiedad de previousOwnerType y currentOwnerType tienen un valor de MANUFACTURER. Esto se debe a que, el fabricante(Manufacturer) es quien empieza la cadena de suministro y por ende el propietario actual(currentOwnerType) y previo(previousOwnerType) es el mismo fabricante. La fecha de creación(createDateTime) y fecha de actualización(lastUpdate) recogen el valor de la variable dt establecida previamente. Finalmente, se hizo uso de la expresión await por ser una función asíncrona y se guarda la transacción utilizando la función putState(), la cual permite almacenar en la ledger la información descrita dentro del paréntesis. En la ledger se almacena en un formato clave valor. Siendo la clave el ID del producto(equipmentNumber) y el valor lo que se encuentre dentro de la constante equipment, a continuación, se transforma los datos a formato JSON para transmitirlos por el Buffer.

Se creó una función asíncrona denominada wholesalerDistribute que permitió actualizar el nuevo propietario del producto, ante lo cual requiere de tres parámetros:

- ctx: hace referencia al contexto de la transacción, se la utiliza para enviar los datos de cada transacción al ledger de la red de Hyperledger Fabric.
- equipmentNumber: es el número identificativo del producto
- ownerName: hace referencia al propietario actual del producto

```
async wholesalerDistribute(ctx, equipmentNumber, ownerName) {  
  
  const equipmentAsBytes = await ctx.stub.getState(equipmentNumber);  
  
  if (!equipmentAsBytes || equipmentAsBytes.length === 0) {  
    throw new Error(`${equipmentNumber} does not exist`);  
  }  
  
  let dt = new Date().toString();  
  
  const strValue = Buffer.from(equipmentAsBytes).toString('utf8');  
  
  let record;  
  
  try {  
  
    record = JSON.parse(strValue);  
  
    if(record.currentOwnerType !== 'MANUFACTURER') {  
      throw new Error(`equipment - ${equipmentNumber} owner must be MANUFACTURER`);  
    }  
  
    record.previousOwnerType= record.currentOwnerType;  
  
    record.currentOwnerType = 'WHOLESALE';  
  
    record.ownerName = ownerName;  
  
    record.lastUpdated = dt;  
  
  } catch (err) {  
  
    console.log(err);  
  
    throw new Error(`equipment ${equipmentNumber} data can't be processed`);  
  }  
}
```



```

    }

    await ctx.stub.putState(equipmentNumber, Buffer.from(JSON.stringify(record)));

}

```

Se creó la constante `equipmentAsBytes` para poder consultar a la ledger si existen datos referentes a un producto determinado, para ello es necesario utilizar la función `getState()` y pasar por parámetro en ID del producto. El siguiente paso fue crear un condicional con la finalidad de conocer si existe o no dicho producto. En caso de no existir se genera una excepción, indicando que dicho producto no existe. Caso contrario, se establece la variable `dt` para almacenar la fecha de registro. De igual manera se creó la constante `strValue` con la información retornada por la función `getState()` y se realizó su conversión a `String` mediante la función `toString()`. La información extraída de la ledger se convirtió a formato JSON (`JSON.parse(strValue)`) y se creó un condicional para determinar si el propietario anterior es el fabricante (`Manufacturer`). Si la condición establecida se cumple en la variable `record` se guardará en `previousOwnerType` el valor del `currentOwnerType` que en este caso será el Distribuidor (`WHOLESALE`), y la fecha de actualización (`lastUpdate`) será la actual. Finalmente, se hizo uso de la expresión `await` por ser una función asíncrona y se guardó la transacción utilizando la función `putState()`, la cual permitió almacenar en la ledger la información actualizada de la variable `record` y se procedió a transformar los datos a formato JSON mediante la función.

A continuación, se creó una función asíncrona denominada `pharmacyReceived` que permitió actualizar el nuevo propietario del producto, ante lo cual requiere de tres parámetros:

- `ctx`: hace referencia al contexto de la transacción, se la utiliza para enviar los datos de cada transacción al ledger de la red de Hyperledger Fabric.
- `equipmentNumber`: es el número identificativo del producto
- `ownerName`: hace referencia al propietario actual del producto

```
async pharmacyReceived(ctx, equipmentNumber, ownerName) {  
  
  const equipmentAsBytes = await ctx.stub.getState(equipmentNumber);  
  
  if (!equipmentAsBytes || equipmentAsBytes.length === 0) {  
    throw new Error(`${equipmentNumber} does not exist`);  
  }  
  
  let dt = new Date().toString();  
  
  const strValue = Buffer.from(equipmentAsBytes).toString('utf8');  
  
  let record;  
  
  try {  
    record = JSON.parse(strValue);  
  
    if(record.currentOwnerType !== 'WHOLESALER') {  
      throw new Error(`equipment - ${equipmentNumber} owner must be WHOLESALER`);  
    }  
  
    record.previousOwnerType= record.currentOwnerType;  
    record.currentOwnerType = 'PHARMACY';  
    record.ownerName = ownerName;  
    record.lastUpdated = dt;  
  } catch (err) {  
    console.log(err);  
    throw new Error(`equipment ${equipmentNumber} data can't be processed`);  
  }  
  
  await ctx.stub.putState(equipmentNumber, Buffer.from(JSON.stringify(record)));  
}
```

Se creó la constante `equipmentAsBytes` para poder consultar a la ledger si existen datos referentes a un producto determinado, para ello es necesario utilizar la función `getState()` y pasar por parámetro en ID del producto. Se utilizó un condicional con la finalidad de conocer si existe o no dicho producto. En caso de no existir se generó una excepción, indicando que dicho producto no existe. Caso contrario, se estableció la variable `dt` para almacenar la fecha de registro. De igual manera se creó la constante `strValue` con la información retornada por la función `getState()` y se realizó su conversión a `String` mediante la función `toString()`. La información extraída de la ledger se transforma a formato JSON (`JSON.parse(strValue)`) y se creó un condicional para determinar si el propietario anterior es el distribuidor (`Whosaler`). Si la condición establecida se cumple en la variable `record` se guardó en `previousOwnerType` el valor del `currentOwnerType` que en este caso será el Punto de Venta (`PHARMACY`), y la fecha de actualización (`lastUpdate`) será la actual. Finalmente, se hace uso de la expresión `await` por ser una función asíncrona y se guarda la transacción utilizando la función `putState()`, la cual permite almacenar en la ledger la información actualizada de la variable `record` y se procede a transformar los datos a formato JSON mediante la función.

Consulta a la Ledger

La función `queryByKey()` permite consultar a la ledger mediante el paso de un ID como parámetro del estado actual de un producto. En la variable `value` se guardó la información extraída de la ledger mediante la función `getState()`, a continuación, se creó la constante `strValue` para setear la información extraída y convertirla a `String` mediante la función `toString()`. Creamos la variable `record` para transformar la información de la constante `srtValue` a formato JSON. Finalmente, mediante la función `JSON.stringify()` se creó un objeto de tipo JSON, el cual tendrá como clave el ID del producto y cuyo valor será la información obtenida de la variable `record`.

```
async queryByKey(ctx, key) {  
    let value = await ctx.stub.getState(key);
```

```

const strValue = Buffer.from(value).toString('utf8');

let record;

  try {

    record = JSON.parse(strValue);

  } catch (err) {

    console.log(err);

    record = strValue;

  }

return JSON.stringify({

  Key: key, Record: record

});

}

```

La función `queryHistoryByKey()` devuelve el historial de todas las transacciones relacionadas a un determinado producto, para lo cual es necesario especificar el ID del producto como parámetro.

Mediante la función `stub.getHistoryForKey()` se puede obtener toda la información de un producto mediante el paso de parámetro de su ID y se la guardó dentro de la variable `iterator`. Se crea un bucle con la finalidad de obtener la información completa de un determinado producto, una vez obtenida se la convierte a formato JSON y se la guardó dentro del array declarado posteriormente como `result[]`. Finalmente, se retorna el array como un objeto JSON mediante la función `JSON.stringify()`.

```

async queryHistoryByKey(ctx, key) {

  console.info('getting history for key: ' + key);

  let iterator = await ctx.stub.getHistoryForKey(key);

  let result = [];

  let res = await iterator.next();

```

```
while (!res.done) {  
  if (res.value) {  
    const obj = JSON.parse(res.value.value.toString('utf8'));  
    result.push(obj);  
  }  
  res = await iterator.next();  
}  
await iterator.close();  
console.info(result);  
return JSON.stringify(result);  
}
```

Configuración de la Red de Hyperledger Fabric

Configtx

Este archivo contiene la configuración de las diferentes organizaciones de la red. Para el presente proyecto se tienen 4 organizaciones: fabricante, distribuidor, punto de venta y orderer. Para todas las organizaciones se especifica un Name, ID, MSPDir, Policies. MSPDir hace referencia al directorio donde se encuentra el material criptográfico generado por la herramienta criptogen, el cual permite interactuar con las diferentes organizaciones dentro de la red. Se establecen políticas de escritura (Writers), lectura (Readers) y administración (Admin). Así como, el punto de acceso a cada una de ellas (OrdererEndpoints).

- **&OrdererOrg**

Name: OrdererOrg

ID: OrdererMSP

MSPDir: ../organizations/ordererOrganizations/example.com/msp

Policies:

Readers:

Type: Signature

Rule: "OR('OrdererMSP.member')"

Writers:

Type: Signature

Rule: "OR('OrdererMSP.member')"

Admins:

Type: Signature

Rule: "OR('OrdererMSP.admin')"

OrdererEndpoints:

- orderer.example.com:7050

En la organización 1(Org1) se observa que se especifica un nombre (Name), ID, MSPDir, políticas (Policies) y punto de acceso (OrdererEndPoint). En lo referente a las políticas esta organización tiene permisos de lectura (Readers), escritura (Writers) y endosamiento (Endorsement) la cual permite respaldar las transacciones. En esta organización podemos ver que la política de lectura es de tipo (Type) firma (Signature), la regla (Rule) es que sólo la podrán ejecutar el administrador (admin) de la organización 1, el peer de la organización 1 y el cliente de la organización 1. En la política de escritura sólo podrá realizar esta acción el admin de la organización 1 y el cliente de la organización 1. En políticas de administración solo el admin de la organización 1 podrá hacerlo. En la política de endosamiento o

quien se encarga de respaldar las transacciones sólo lo realizará el peer. A continuación, se define un AnchorPeer que actúa como punto de acceso principal a la red y por ende es por donde se van a estar comunicando las organizaciones.

- &Org1

Name: Org1MSP

ID: Org1MSP

MSPDir: ../organizations/peerOrganizations/org1.example.com/msp

Policies:

Readers:

Type: Signature

Rule: "OR('Org1MSP.admin', 'Org1MSP.peer', 'Org1MSP.client')"

Writers:

Type: Signature

Rule: "OR('Org1MSP.admin', 'Org1MSP.client')"

Admins:

Type: Signature

Rule: "OR('Org1MSP.admin')"

Endorsement:

Type: Signature

Rule: "OR('Org1MSP.peer')"

AnchorPeers:

- Host: peer0.org1.example.com

Port: 7051

Ahora fue necesario definir las reglas del consorcio, donde se van a configurar la red mediante las Capabilities: Channel, Orderer y Application. Las Capabilities permiten que los nodos de la red que se ejecutan en diferentes versiones se mantengan actualizados, con la finalidad de exista compatibilidad y consistencia en la configuración del canal.

Capabilities:

Channel: `&ChannelCapabilities`

V2_0: true

Orderer: `&OrdererCapabilities`

V2_0: true

Application: `&ApplicationCapabilities`

V2_0: true

En la sección de Aplicación se especifican políticas(policies) que son un conjunto de reglas que permiten al consorcio tomar decisiones y poder lograr objetivos específicos. En esta sección, se puede observar que se debe especificar políticas de: lectura (Readers), escritura (Writers), administración (Admin), endosamiento (Endorsement).

Application: `&ApplicationDefaults`

Organizations:

Policies:**Readers:**

Type: ImplicitMeta

Rule: "ANY Readers"

Writers:

Type: ImplicitMeta

Rule: "ANY Writers"

Admins:

Type: ImplicitMeta

Rule: "MAJORITY Admins"

LifecycleEndorsement:

Type: ImplicitMeta

Rule: "MAJORITY Endorsement"

Endorsement:

Type: ImplicitMeta

Rule: "MAJORITY Endorsement"

Capabilities:

<<: *ApplicationCapabilities

En la sección de Orderer se especifica el mecanismo de consenso, se lo realizó vía Raft. Debido a que el consenso vía SOLO es utilizado solo para la realización en ambientes de pruebas y KAFKA requiere de archivos adicionales, lo que lo hace más pesado para implementarse. Además, se especifica la ruta donde se encuentran los certificados de transacción tanto para el cliente como para el servidor. El BatchTimeout sirve para indicar el tiempo que se deberá demorar para crear un batch, en este caso será

de 2 segundos. Otros parámetros requeridos son: el tamaño del batch(BatchSize), máximo número de mensajes dentro del batch(MaxMessageCount), tamaño de serialización(AbsoluteMaxBytes), máximo número de bytes que soporta la serialización dentro del batch(PreferredMaxBytes).

Orderer: &OrdererDefaults

OrdererType: etcdraft

EtcdRaft:

Consenters:

- Host: orderer.example.com

Port: 7050

ClientTLSCert: ../organizations/ordererOrganizations/example.com/orderers/orderer.example.com/tls/server.crt

ServerTLSCert: ../organizations/ordererOrganizations/example.com/orderers/orderer.example.com/tls/server.crt

BatchTimeout: 2s

BatchSize:

MaxMessageCount: 10

AbsoluteMaxBytes: 99 MB

PreferredMaxBytes: 512 KB

Organizations:

Policies:

Readers:

Type: ImplicitMeta

Rule: "ANY Readers"

Writers:

Type: ImplicitMeta

Rule: "ANY Writers"

Admins:

Type: ImplicitMeta

Rule: "MAJORITY Admins"

BlockValidation:

Type: ImplicitMeta

Rule: "ANY Writers"

En el apartado Profile se realizó una configuración de resumen de cómo está configurada la red, aquí se especifica a cada uno de los peers que conforman el consorcio, el canal que será implementado, las autoridades de certificación.

Profiles:

PharmaLedgerOrdererGenesis:

<<: *ChannelDefaults

Orderer:

<<: *OrdererDefaults

Organizations:

- *OrdererOrg

Capabilities:

<<: *OrdererCapabilities

Consortiums:

PharmaLedgerConsortium:

Organizations:

- *Org1

- *Org2

- *Org3

PharmaLedgerChannel:

Consortium: PharmaLedgerConsortium

<<: *ChannelDefaults

Application:

<<: *ApplicationDefaults

Organizations:

- *Org1

- *Org2

- *Org3

Capabilities:

<<: *ApplicationCapabilities

Docker Compose

Docker compose contiene los archivos que se van a utilizar para iniciar el entorno de ejecución de fabric. Aquí se definen los diferentes servicios, volúmenes, variables de entorno, TLS, archivos criptográficos, certificados de las llaves privadas de Hyperledger Fabric. Docker va a levantar diversos contenedores para cada servicio y organización. Se puede observar que para cada organización se ha creado una autoridad de certificación, así como: se especifica el puerto, material criptográfico, nombre del contenedor.

```
version: '2'
```

```
networks:
```

```
  pln:
```

```
services:
```

```
  ca_org1:
```

```
    image: hyperledger/fabric-ca:$IMAGE_TAG
```

```
    environment:
```

```
      - FABRIC_CA_HOME=/etc/hyperledger/fabric-ca-server
```

```
      - FABRIC_CA_SERVER_CA_NAME=ca-org1
```

```
      - FABRIC_CA_SERVER_TLS_ENABLED=true
```

```
      - FABRIC_CA_SERVER_PORT=7054
```

```
    ports:
```

```
      - "7054:7054"
```

```
    command: sh -c 'fabric-ca-server start -b admin:adminpw -d'
```

volumes:

- ../organizations/fabric-ca/org1:/etc/hyperledger/fabric-ca-server

container_name: ca_org1

networks:

- pln

ca_org2:

image: hyperledger/fabric-ca:\$IMAGE_TAG

environment:

- FABRIC_CA_HOME=/etc/hyperledger/fabric-ca-server

- FABRIC_CA_SERVER_CA_NAME=ca-org2

- FABRIC_CA_SERVER_TLS_ENABLED=true

- FABRIC_CA_SERVER_PORT=8054

ports:

- "8054:8054"

command: sh -c 'fabric-ca-server start -b admin:adminpw -d'

volumes:

- ../organizations/fabric-ca/org2:/etc/hyperledger/fabric-ca-server

container_name: ca_org2

networks:

- pln

ca_org3:

image: hyperledger/fabric-ca:\$IMAGE_TAG

environment:

- FABRIC_CA_HOME=/etc/hyperledger/fabric-ca-server
- FABRIC_CA_SERVER_CA_NAME=ca-org3
- FABRIC_CA_SERVER_TLS_ENABLED=true
- FABRIC_CA_SERVER_PORT=9054

ports:

- "9054:9054"

command: sh -c 'fabric-ca-server start -b admin:adminpw -d'

volumes:

- ../organizations/fabric-ca/org3:/etc/hyperledger/fabric-ca-server

container_name: ca_org3

networks:

- pln

ca_orderer:

image: hyperledger/fabric-ca:\$IMAGE_TAG

environment:

- FABRIC_CA_HOME=/etc/hyperledger/fabric-ca-server
- FABRIC_CA_SERVER_CA_NAME=ca-orderer
- FABRIC_CA_SERVER_TLS_ENABLED=true
- FABRIC_CA_SERVER_PORT=11054

ports:

- "11054:11054"

command: sh -c 'fabric-ca-server start -b admin:adminpw -d'

volumes:

- ../organizations/fabric-ca/ordererOrg:/etc/hyperledger/fabric-ca-server

container_name: ca_orderer

networks:

- pln

Los servicios que se levantarán son: orderer y un peer para cada organización. Para el caso del Orderer se indica el nombre del contenedor(orderer.example.com) y la imagen de Docker que se utilizó para levantar ese contenedor (hyperledger/fabric-orderer: \$IMAGE_TAG) haciendo uso de los recursos predefinidos de Hyperledger Fabric. Las variables de entorno(enviroment) que se definen son las relacionadas al contenedor. En la parte de volúmenes se copia la dirección de los archivos de configuración hacia el contenedor, por ejemplo, el archivo system-genesis-block/genesis.block será copiado en la dirección del contenedor /var/hyperledger/orderer/orderer.genesis.block.

services:**orderer.example.com:**

container_name: orderer.example.com

image: hyperledger/fabric-orderer:\$IMAGE_TAG

environment:

- FABRIC_LOGGING_SPEC=INFO

- ORDERER_GENERAL_LISTENADDRESS=0.0.0.0

- ORDERER_GENERAL_LISTENPORT=7050

- ORDERER_GENERAL_GENESIMETHOD=file

- ORDERER_GENERAL_GENESISFILE=/var/hyperledger/orderer/orderer.genesis.block

- ORDERER_GENERAL_LOCALMSPID=OrdererMSP
- ORDERER_GENERAL_LOCALMSPDIR=/var/hyperledger/orderer/msp
- ORDERER_GENERAL_TLS_ENABLED=true
- ORDERER_GENERAL_TLS_PRIVATEKEY=/var/hyperledger/orderer/tls/server.key
- ORDERER_GENERAL_TLS_CERTIFICATE=/var/hyperledger/orderer/tls/server.crt
- ORDERER_GENERAL_TLS_ROOTCAS=[/var/hyperledger/orderer/tls/ca.crt]
- ORDERER_KAFKA_TOPIC_REPLICATIONFACTOR=1
- ORDERER_KAFKA_VERBOSE=true
- ORDERER_GENERAL_CLUSTER_CLIENTCERTIFICATE=/var/hyperledger/orderer/tls/server.crt
- ORDERER_GENERAL_CLUSTER_CLIENTPRIVATEKEY=/var/hyperledger/orderer/tls/server.key
- ORDERER_GENERAL_CLUSTER_ROOTCAS=[/var/hyperledger/orderer/tls/ca.crt]

working_dir: /opt/gopath/src/github.com/hyperledger/fabric

command: orderer

volumes:

- ../system-genesis-block/genesis.block:/var/hyperledger/orderer/orderer.genesis.block
- ../organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp:/var/hy

perledger/orderer/msp

- ../organizations/ordererOrganizations/example.com/orderers/orderer.example.com/tls:/var/hyp

erledger/orderer/tls

- orderer.example.com:/var/hyperledger/production/orderer

ports:

- 7050:7050

networks:

- pln

Para los peer se realizó el mismo procedimiento que en el servicio Orderer, especificando el nombre del contenedor, la ubicación de la imagen, cada una de las variables de entorno, volúmenes, puertos. Este proceso se lo realiza para cada una de las organizaciones dentro de la red.

[peer0.org1.example.com](#):

`container_name`: peer0.org1.example.com

`image`: hyperledger/fabric-peer:\$IMAGE_TAG

`environment`:

- CORE_VM_ENDPOINT=unix:///host/var/run/docker.sock
- CORE_VM_DOCKER_HOSTCONFIG_NETWORKMODE=\${COMPOSE_PROJECT_NAME}_pln
- FABRIC_LOGGING_SPEC=INFO
- CORE_PEER_TLS_ENABLED=true
- CORE_PEER_GOSSIP_USELEADERELECTION=true
- CORE_PEER_GOSSIP_ORGLEADER=false
- CORE_PEER_PROFILE_ENABLED=true
- CORE_PEER_TLS_CERT_FILE=/etc/hyperledger/fabric/tls/server.crt
- CORE_PEER_TLS_KEY_FILE=/etc/hyperledger/fabric/tls/server.key
- CORE_PEER_TLS_ROOTCERT_FILE=/etc/hyperledger/fabric/tls/ca.crt
- CORE_PEER_ID=peer0.org1.example.com
- CORE_PEER_ADDRESS=peer0.org1.example.com:7051
- CORE_PEER_LISTENADDRESS=0.0.0.0:7051
- CORE_PEER_CHAINCODEADDRESS=peer0.org1.example.com:7052
- CORE_PEER_CHAINCODELISTENADDRESS=0.0.0.0:7052
- CORE_PEER_GOSSIP_BOOTSTRAP=peer0.org1.example.com:7051
- CORE_PEER_GOSSIP_EXTERNALENDPOINT=peer0.org1.example.com:7051

- CORE_PEER_LOCALMSPID=Org1MSP

volumes:

- /var/run:/host/var/run/

- ../organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/msp:/etc/

hyperledger/fabric/msp

- ../organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls:/etc/hy

perledger/fabric/tls

- peer0.org1.example.com:/var/hyperledger/production

working_dir: /opt/gopath/src/github.com/hyperledger/fabric/peer

command: peer node start

ports:

- 7051:7051

networks:

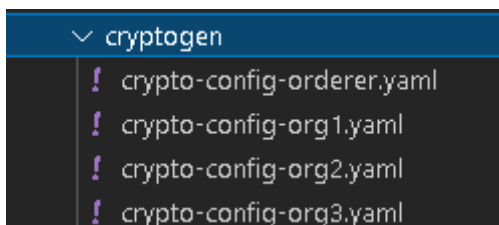
- pln

Cryptogen

Para que las diferentes organizaciones se puedan identificar dentro de la red, es necesario que tengan un certificado digital. Las autoridades certificadoras (CA) se encargan de gestionar dichos certificados X.509, para el presente proyecto se utilizó Cryptogen que es la herramienta proporcionada por Hyperledger Fabric para crear los certificados para cada organización, así como se encarga de generar el material criptográfico. En la Figura 11, se observa que se tienen cuatro organizaciones: orderer, org1(fabricante), org2(distribuidor) y org3(punto de venta).

Figura 11

Organizaciones definidas en cryptogen



En la Figura 12, se muestra como por medio de cryptogen se generó el material criptográfico asociado a la organización 1, se indica cual es el archivo que tiene que tomar como referencia y la salida del material criptográfico. Este procedimiento se lo realizó para cada organización.

Figura 12

Generación del material criptográfico

```
cryptogen generate --config=./organizations/cryptogen/crypto-config-org1.yaml --output="organizations"
```

Ahora, fue necesario crear el bloque Génesis como se puede observar en la Figura 13. El cual es un bloque de configuración más no de datos, que permite determinar cómo va a funcionar la red de Hyperledger Fabric. Se especifica el profile PharmaLedgerOrdererGenesis, el ID del canal system-channel y la carpeta de salida ./system-genesis-block/genesis.block.

Figura 13

Creación del bloque Génesis

```
##### Generating Orderer Genesis block #####
+ configtxgen -profile PharmaLedgerOrdererGenesis -channelID system-channel -outputBlock ./system-genesis-block/genesis.block
2022-06-17 13:43:25.443 -05 [common.tools.configtxgen] main -> INFO 001 Loading configuration
2022-06-17 13:43:26.006 -05 [common.tools.configtxgen.localconfig] completeInitialization -> INFO 002 Orderer.Addresses unset,
setting to [127.0.0.1:7050]
2022-06-17 13:43:26.008 -05 [common.tools.configtxgen.localconfig] completeInitialization -> INFO 003 orderer type: etcdraft
2022-06-17 13:43:26.009 -05 [common.tools.configtxgen.localconfig] completeInitialization -> INFO 004 Orderer.EtcdRaft.Options
unset, setting to tick_interval:"500ms" election_tick:10 heartbeat_tick:1 max_inflight_blocks:5 snapshot_interval_size:167772
16
```

Creación del Canal de Comunicación entre las Organizaciones

Es necesario un canal de comunicación entre los diferentes peers y organizaciones, mediante la herramienta configtxgen, se indica un perfil o profile PharmaLedgerChannel, seguido de la especificación de la carpeta en donde se va a guardar la información ./channel-artifacts/\${CHANNEL_NAME}.tx.

```
configtxgen -profile PharmaLedgerChannel -outputCreateChannelTx ./channel-
artifacts/${CHANNEL_NAME}.tx -channelID $CHANNEL_NAME
```

A continuación, se crearon las transacciones de los AnchorPeer, que son los peer que se comunican con cada organización, por lo cual cada una de ellas debe tener su AnchorPeer. Se debe especificar el profile(PharmaLedgerChannel), el archivo donde se va a guardar la información y el ID del canal.

```
configtxgen -profile PharmaLedgerChannel -outputAnchorPeersUpdate ./channel-
artifacts/${orgmsp}anchors.tx -channelID $CHANNEL_NAME -asOrg ${orgmsp}
```

Finalmente, se creó el canal mediante el comando peer channel create. Los parámetros que se necesita especificar son: el nombre del canal(\$CHANNEL_NAME), servicio de ordenamiento(orderer), archivo donde se encuentra la información de transacción del canal, se especifica la carpeta donde se guardará la información del canal creado, protocolo de transporte(TLS) y la autoridad certificadora(CA).

```
peer channel create -o localhost:7050 -c $CHANNEL_NAME --
ordererTLSHostnameOverride orderer.example.com -f ./channel-artifacts/${CHANNEL_NAME}.tx --
outputBlock ./channel-artifacts/${CHANNEL_NAME}.block --tls $CORE_PEER_TLS_ENABLED --
cafile $ORDERER_CA >&log.txt
```

En la Figura 14, se observa que el canal se ha creado de forma correcta.

Figura 14

Visualización de la creación del canal

```
+ configtxgen -profile PharmaLedgerChannel -outputCreateChannelTx ./channel-artifacts/plnchannel.tx -ch
annelID plnchannel
2022-06-21 21:16:05.552 -05 [common.tools.configtxgen] main -> INFO 001 Loading configuration
2022-06-21 21:16:05.724 -05 [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded configuratio
n: /home/daniel/HF Dev Final Lab- Updated on August 21 2020/pharma-ledger-network/configtx/configtx.yam
l
```

Si bien, se ha creado el canal de comunicación que utilizará cada peer de la organización, es necesario anexar al canal a cada una de ellas. Esto se lo realizó mediante el comando `peer channel join`, en el cual se especificó la dirección del archivo de configuración creado anteriormente. Solo las organizaciones que se encuentran registradas en este archivo podrán anexarse al canal. Se debe actualizar los `AnchorPeers` de cada organización, indicando el nombre del canal, la dirección de cada peer, así como los archivos donde se encuentran los certificados de autorización generados anteriormente.

```
peer channel join -b ./channel-artifacts/$CHANNEL_NAME.block >&log.txt
```

En la Figura 15, se observa la ejecución y creación del `AnchorPeer` de la Organización 3.

Figura 15

Visualización de la creación del AnchorPeer de una organización

```
==== Generating anchor peer update transaction for Org3MSP ====
+ configtxgen -profile PharmaLedgerChannel -outputAnchorPeersUpdate ./channel-artifacts/Org3MSPanchors.
tx -channelID plnchannel -asOrg Org3MSP
2022-06-21 21:26:33.945 -05 [common.tools.configtxgen] main -> INFO 001 Loading configuration
2022-06-21 21:26:34.139 -05 [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded configuratio
n: /home/daniel/HF Dev Final Lab- Updated on August 21 2020/pharma-ledger-network/configtx/configtx.yam
l
2022-06-21 21:26:34.147 -05 [common.tools.configtxgen] doOutputAnchorPeersUpdate -> INFO 003 Generating
anchor peer update
2022-06-21 21:26:34.156 -05 [common.tools.configtxgen] doOutputAnchorPeersUpdate -> INFO 004 Writing an
chor peer update
```

Ejecución del Smart Contract

Una vez creado el Smart Contract es hora de incorporarlo en el canal, para ello se necesita especificar ciertos parámetros como: nombre del canal(`CHANNEL_NAME`), tipo de lenguaje de

programación(CC_SRC_LANGUAGE), ubicación del contrato o smart contract(CC_SRC_PATH), nombre del contrato(CHAINCODE_NAME).

```
CHANNEL_NAME="$1"
CC_SRC_LANGUAGE="$2"
VERSION="$3"
DELAY="$4"
MAX_RETRY="$5"
VERBOSE="$6"

: ${CHANNEL_NAME:="plnchannel"}
: ${CC_SRC_LANGUAGE:="javascript"}
: ${VERSION:="1"}
: ${DELAY:="3"}
: ${MAX_RETRY:="5"}
: ${VERBOSE:="false"}

CC_SRC_LANGUAGE=`echo "$CC_SRC_LANGUAGE" | tr [:upper:] [:lower:]`
CC_RUNTIME_LANGUAGE=node # chaincode runtime language is node.js
CC_SRC_PATH="organizations/manufacturer/contract/"
CHAINCODE_NAME="pharmaLedgerContract"
FABRIC_CFG_PATH=$PWD/./config/
```

Ahora fue necesario empaquetar el smart contract. Utilizamos el comando `peer lifecycle chaincode package`, en el cual se especificó el nombre del contrato(CHAINCODE_NAME), la ubicación del archivo a empaquetar(CC_SRC_PATH), el tipo de lenguaje de programación(CC_RUNTIME_LANUAJE), se puede etiquetar dando un nombre descriptivo(CHAINCODE_NAME) y la versión(VERSION).

```
peer lifecycle chaincode package ${CHINCODE_NAME}.tar.gz --path ${CC_SRC_PATH} --
lang ${CC_RUNTIME_LANGUAGE} --label ${CHINCODE_NAME}_${VERSION} >&log.txt
```

Cada organización debe tener el smart contract, es por ello que se especificó el peer de la organización con el respectivo nombre del contrato(CHINCODE_NAME) y se procedió a instalarlo mediante el comando install. Este proceso se lo realizó con cada miembro de la organización.

```
peer lifecycle chaincode install ${CHINCODE_NAME}.tar.gz >&log.txt
```

Para poder saber si el contrato se ha instalado en cada organización, hicimos uso de la sentencia queryinstalled. Como resultado devolverá un identificador del paquete(PACKAGE_ID) , el cual debe ser igual para cada una de las organizaciones.

```
peer lifecycle chaincode queryinstalled >&log.txt
```

El siguiente paso es, cada organización debe aprobar el chaincode o contrato instalado. Es decir, las políticas de negocio establecidas fueron revisadas y aprobadas por cada organización, para ello se hizo uso del comando approveformyorg. Especificando el puerto utilizado por cada peer(7050), el nombre de la organización, la entidad certificadora(ORDERER_CA), ID del canal(CHANNEL_ID), nombre del canal(CHANNEL_NAME), nombre del chaincode o contrato inteligente(CHINCODE_NAME), version(VERSION), ID del paquete que previamente fue empaquetado(PACKAGE_ID).

```
peer lifecycle chaincode approveformyorg -o localhost:7050 --
ordererTLSHostnameOverride orderer.example.com --tls $CORE_PEER_TLS_ENABLED --
cafile $ORDERER_CA --channelID $CHANNEL_NAME --name ${CHINCODE_NAME} --version ${VERSION} -
-package-id ${PACKAGE_ID} --sequence ${VERSION} >&log.txt
```

Se requirió conocer cual organización ha aprobado el contrato y para ello hacemos uso de la sentencia checkcommitreadiness, especificaremos el ID del canal, nombre del canal, nombre del

contrato, versión. Si la organización reconoce que el contrato se ha validado, retornará un true caso contrario un false.

```
peer lifecycle chaincode checkcommitreadiness --channelID $CHANNEL_NAME --
name ${CHINCODE_NAME} --version ${VERSION} --sequence ${VERSION} --output json >&log.txt
```

Una vez que las organizaciones han instalado y validado el smart contract, se realizó un commit. Especificamos el puerto del peer, servicio de ordenamiento, protocolo de seguridad(TLS), autoridad certificadora(CA), ID del canal, nombre del canal, nombre del contrato, entre otros.

```
peer lifecycle chaincode commit -o localhost:7050 --
ordererTLSHostnameOverride orderer.example.com --tls $CORE_PEER_TLS_ENABLED --
cafile $ORDERER_CA --channelID $CHANNEL_NAME --
name ${CHINCODE_NAME} $PEER_CONN_PARMS --version ${VERSION} --
sequence ${VERSION} >&log.txt
```

Se puede observar la ejecución del Smart Contract una vez realizados cada uno de los pasos anteriores. Ver Figura 16.

Figura 16

Ejecución del Smart Contract

```
Using organization 1
**** [Step: 1]: start call packageChaincode on peer: peer0.org1, channelID: plnchannel, smartcontract:
  pharmaLedgerContract, version 1, sequence 1 ****
+ peer lifecycle chaincode package pharmaLedgerContract.tar.gz --path organizations/manufacturer/contract/ --lang node --label pharmaLedgerContract 1
```

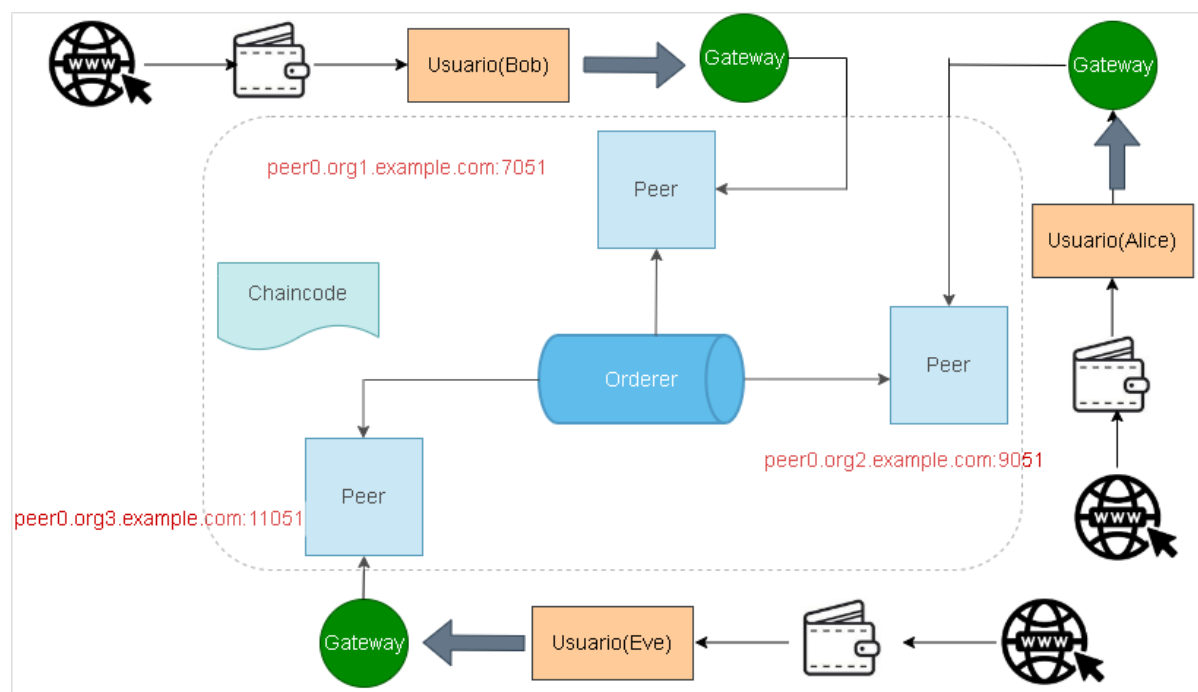
Capítulo 5

Creación del Front End del sistema

Una vez realizado el backEnd de la red Blockchain dentro de la Cadena de Suministros, es necesario crear el frontEnd, el cual ayudará a los usuarios a interactuar con el sistema de forma visual, transparente, ágil y sin la utilización de comandos. Para cada una de las organizaciones se debe crear una billetera(wallet) por cada usuario de entrada, el cual se relacionará con cada uno de los certificados de autenticación(CA) que contienen las llaves privadas de usuario de cada organización.

Figura 17

Estructura de la aplicación para la red de Blockchain



La aplicación se ha desarrollado en Express para poder encontrar las rutas que nos lleven a cada uno de los servicios de la app. En la Figura 17, se puede observar que se importa cada uno de los servicios creados previamente dentro de la red de Blockchain, con sus respectivas ubicaciones. Cada uno

de los servicios son instanciados para cada organización, con la finalidad de poder interactuar con cada uno de ellos.

```
const EquipmentService = require('./services/equipmentService.js');
const QueryService = require( "./services/queryService.js" );
const QueryHistoryService = require( "./services/queryHistoryService.js" );
const WalletService = require( "./services/walletsService.js" );
const equipmentSvcInstance = new EquipmentService();
const querySvcInstance = new QueryService();
const queryHistorySvcInstance = new QueryHistoryService();
const walletSvcInstance = new WalletService();
```

Cada usuario de la organización debe tener una Wallet(Billetera), la misma que permite acceder a cada uno de los servicios creados. Para ello, fue necesario crear un nuevo usuario dentro de la wallet(addWallet(user)) y se lo guardó dentro de una constante wallet, la misma que inserta al usuario dentro de la billetera mediante el comando Wallets.newFileSystemWallet().

```
const wallet = await Wallets.newFileSystemWallet('./identity/user/'+userName+'/wallet');
```

Se indicó las credenciales que se guardarán en esa wallet, que son los certificados de autenticación, llaves privadas, path que hacen referencia a cada organización. Se puede observar que se han creado tres constantes donde se almacenan las rutas de las variables especificadas.

```
const credPath = path.join(fixture, '/peerOrganizations/org1.example.com/users/User1@org1.example.com');
const certificate = fs.readFileSync(path.join(credPath, '/msp/signcerts/User1@org1.example.com-cert.pem')).toString();
const privateKey = fs.readFileSync(path.join(credPath, '/msp/keystore/priv_sk')).toString();
```

Se creó una constante donde se guarde la información del usuario, especificando sus credenciales, llave privada y certificado de autenticación. Estos servirán de parámetros para añadir la información del usuario mediante la función `wallet.put()`.

```
const identityLabel = user;

const identity = {
  credentials: {
    certificate,
    privateKey
  },
  mspId: 'Org1MSP',
  type: 'X.509'
}

const response = await wallet.put(identityLabel, identity);
```

Servicio EquipmentService

El servicio para el fabricante `equipmentService` requiere de la información de `red(fabric-network)`, así como de la ubicación del contrato o `chaincode`.

```
const { Wallets, Gateway } = require('fabric-network');
const PharmaLedgerContract = require('../contract/lib/pharmaledgercontract.js');
```

Es necesario en la función que dará el servicio para el fabricante pasar como parámetro el nombre del usuario (`userName`). Especificando, la ubicación del mismo dentro de la wallet mediante el comando `Wallets.newFileSystemWallet()`.

```
async makeEquipment(ctx, manufacturer, equipmentNumber, equipmentName, ownerName)
```

```
const wallet = await Wallets.newFileSystemWallet('../identity/user/'+user+'/wallet');
```

A continuación, se creó la puerta de acceso(Gateway) para que el usuario se pueda comunicar con el peer. Esta clase debe ser pasada como parámetro previo al invocar el contrato fabric-network.

```
const gateway = new Gateway();
```

Se realizó la conexión al perfil(profile) de la organización mediante el comando `yaml.safeLoad()`, el cual leerá la información de la organización respectiva para posteriormente conectarse a la red mediante el uso de las credenciales del usuario creado.

```
let connectionProfile = yaml.safeLoad(fs.readFileSync('../organizations/peerOrganizations/org1.example.com/connection-org1.json', 'utf8'));
```

Tanto el nombre del usuario como la información guardada dentro de la wallet fueron recogidas por una nueva variable(connectionOptions), la cual sirvió de parámetro para la conexión con la puerta de enlace(gateway) que requiere cada peer.

```
let connectionOptions = {
  identity: userName,
  wallet: wallet,
  discovery: { enabled:true, asLocalhost: true }
};

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

Se requiere acceder al canal mediante el comando `gateway.getNetwork()`. E igualmente, se requiere acceder al contrato, para ello se hace uso de la sentencia `network.getContract()`. Especificando el nombre del contrato y la organización que va a acceder al mismo.

```
const network = await gateway.getNetwork('plnchannel');
const contract = await network.getContract('pharmaLedgerContract', 'org.pln.PharmaLedgerContract');
```

Servicio QueryService

El servicio para el fabricante queryService hace referencia a la consulta en la ledger de un determinado producto. Para ello se requiere de la información de red(fabric-network), así como de la ubicación del contrato o chaincode. Los pasos a seguir son similares a los descritos en el servicio equipmentService.

```
const { Wallets, Gateway } = require('fabric-network');
const PharmaLedgerContract = require('../..../contract/lib/pharmaledgercontract.js');
```

Es necesario en la función que dará el servicio de consulta de un producto pasar como parámetro el nombre del usuario(userName), el cual fue insertado dentro de la wallet. Adicionalmente, es necesario especificar el ID del producto del cual se desea obtener la información.

```
async queryHistoryByKey(userName, key)
```

A continuación, se creó la puerta de acceso(Gateway) para que el usuario se pueda comunicar con el peer. Esta clase debe ser pasada como parámetro previo al invocar el contrato fabric-network.

```
const gateway = new Gateway();
```

Se realizó la conexión al perfil(profile) de la organización mediante el comando yaml.safeLoad(), el cual leerá la información de la organización respectiva para posteriormente conectarse a la red mediante el uso de las credenciales del usuario creado.

```
let connectionProfile = yaml.safeLoad(fs.readFileSync('../..../organizations/peerOrganizations/org1.example.com/connection-org1.json', 'utf8'));
```

Tanto el nombre del usuario como la información guardada dentro de la wallet serán recogidas por una nueva variable(connectionOptions), la cual sirve de parámetro para la conexión con la puerta de enlace(gateway) que requiere cada peer.

```
let connectionOptions = {
  identity: userName,
  wallet: wallet,
  discovery: { enabled:true, asLocalhost: true }
};

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

Se accedió al canal mediante el comando gateway.getNetwork(). E igualmente, se necesitó acceder al contrato, para ello se hizo uso de la sentencia network.getContract(). Especificando el nombre del contrato y la organización que va a acceder al mismo.

```
const network = await gateway.getNetwork('plnchannel');
const contract = await network.getContract('pharmaLedgerContract', 'org.pln.PharmaLedgerContract');
```

Finalmente, se realizó la transacción de consulta a la ledger mediante el comando contract.submitTransaction(). En la cual, se especificó la función definida en el contrato o chaincode que permite realizar la consulta a la ledge(queryByKey) y el ID(key) respectivo del producto. La información retornada será convertida a formato JSON mediante la función JSON.parse(response).

```
const response = await contract.submitTransaction('queryHistoryByKey', key);

return response?JSON.parse(response):response;
```

Servicio QueryHystoryByKeyService

El servicio para el fabricante queryHystoryByKeyService hace referencia a la consulta en la ledger del historial generado a lo largo de la cadena de suministro un determinado producto. Para ello se requiere de la información de red(fabric-network), así como de la ubicación del contrato o chaincode. Los pasos a seguir son similares a los descritos en el servicio queryService.

```
const { Wallets, Gateway } = require('fabric-network');
const PharmaLedgerContract = require('../contract/lib/pharmaledgercontract.js');
```

Es necesario en la función que dará el servicio de consulta de un producto pasar como parámetro el nombre del usuario(userName), el cual fue insertado dentro de la wallet. Adicionalmente, es necesario especificar el ID del producto del cual se desea obtener la información.

```
async queryHistoryByKey(userName, key)
```

A continuación, se creó la puerta de acceso(Gateway) para que el usuario se pueda comunicar con el peer. Esta clase debe ser pasada como parámetro previo al invocar el contrato fabric-network.

```
const gateway = new Gateway();
```

Se realizó la conexión al perfil(profile) de la organización mediante el comando yaml.safeLoad(), el cual leerá la información de la organización respectiva para posteriormente conectarse a la red mediante el uso de las credenciales del usuario creado.

```
let connectionProfile = yaml.safeLoad(fs.readFileSync('../organizations/peerOrganizations/org1.example.com/connection-org1.json', 'utf8'));
```

Tanto el nombre del usuario como la información guardada dentro de la wallet serán recogidas por una nueva variable(connectionOptions), la cual servirá de parámetro para la conexión con la puerta de enlace(gateway) que requiere cada peer.


```

let connectionOptions = {
  identity: userName,
  wallet: wallet,
  discovery: { enabled:true, asLocalhost: true }
};

await gateway.connect(connectionProfile, connectionOptions);

```

Se requirió acceder al canal mediante el comando `gateway.getNetwork()`. E igualmente, se requiere acceder al contrato, para ello se hace uso de la sentencia `network.getContract()`. Especificando el nombre del contrato y la organización que va a acceder al mismo.

```

const network = await gateway.getNetwork('plnchannel');
const contract = await network.getContract('pharmaLedgerContract', 'org.pln.PharmaLedgerContract');

```

Finalmente, se realizó la transacción de consulta a la ledger mediante el comando `contract.submitTransaction()`. En la cual, se especifica la función definida en el contrato o chaincode que permite realizar la consulta a la ledger (`queryHistoryByKey`) y el ID (`key`) respectivo del producto. La información retornada será convertida a formato JSON mediante la función `JSON.parse(response)`.

```

const response = await contract.submitTransaction('queryHistoryByKey', key);

return response?JSON.parse(response):response;

```

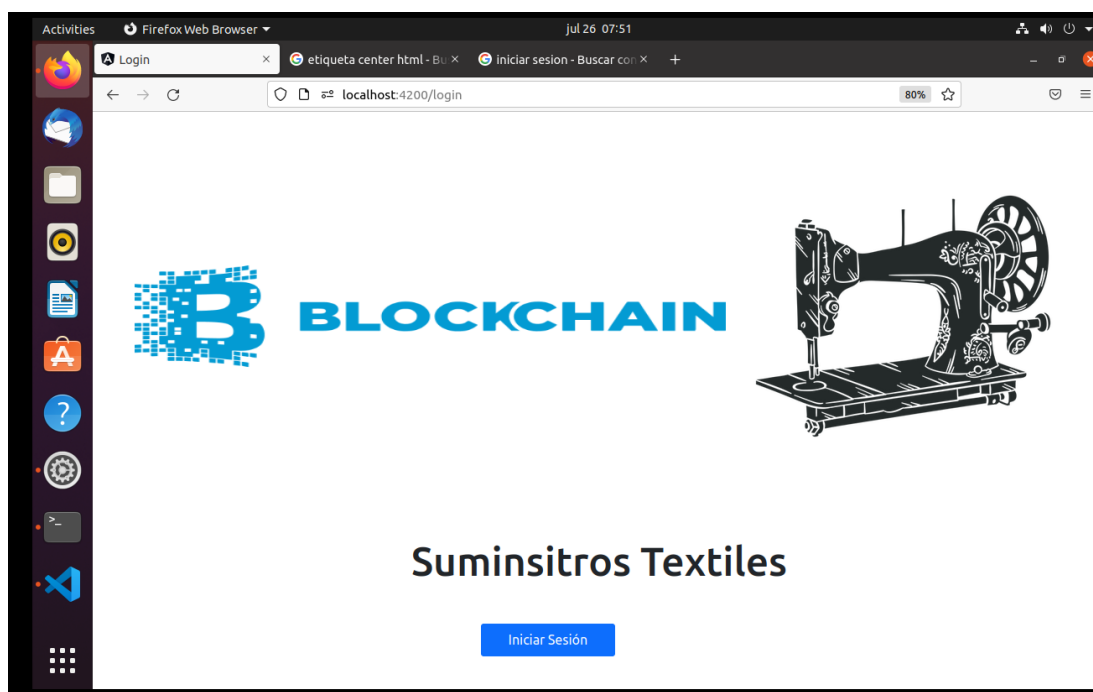
Realización de Pruebas

Una vez que se ha realizado la configuración de cada servicio dentro de la aplicación, con la finalidad de que se presente al usuario un ambiente web más sencillo de manipular. Se observa en la Figura 18, la página principal de la aplicación. Como primera instancia aparece un inicio de sesión el cual

está autenticado con los servicios de Google, que se encarga de gestionar a los usuarios, únicamente el usuario debe tener creada una cuenta en Gmail para poder acceder a la aplicación así al momento de ingresar se registrará el correo electrónico y su usuario para poder agregarlo a la wallet de la aplicación.

Figura 18

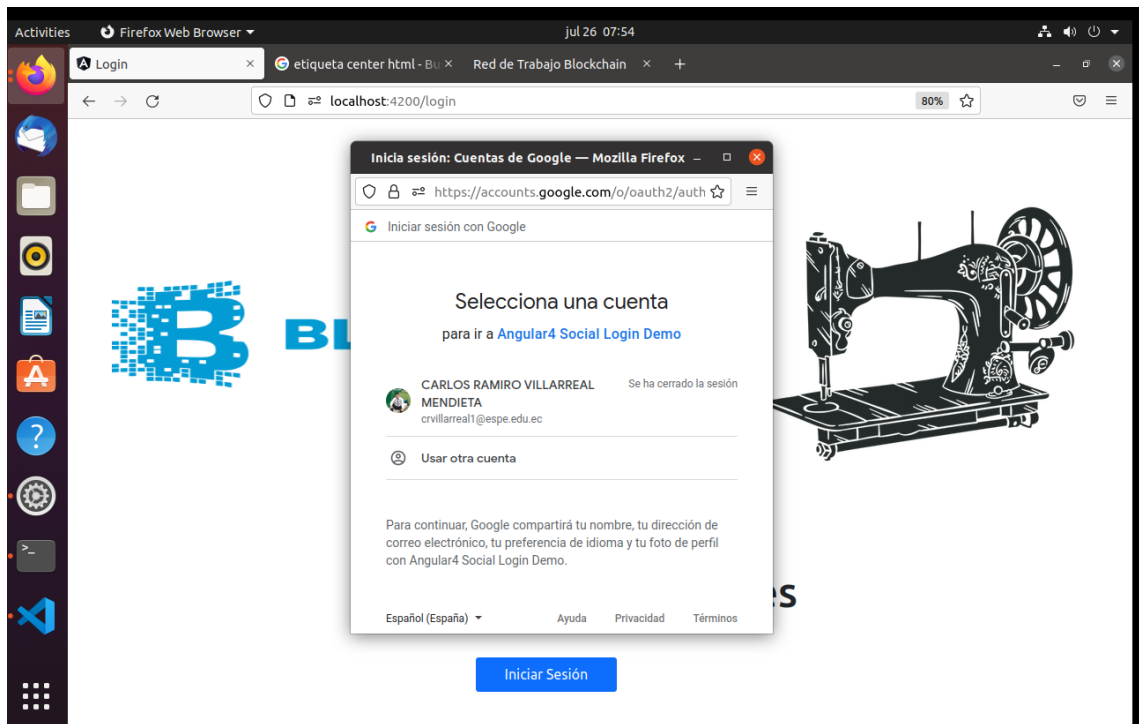
Vista principal de la aplicación



Al momento de iniciar la sesión se debe elegir la cuenta con la que queremos autenticarnos, como se observa en la Figura 19, a continuación, tenemos una cuenta ya registrada en el sistema en caso de no tener cuentas registradas nos pedirá el correo electrónico y la contraseña de nuestra cuenta en Google.

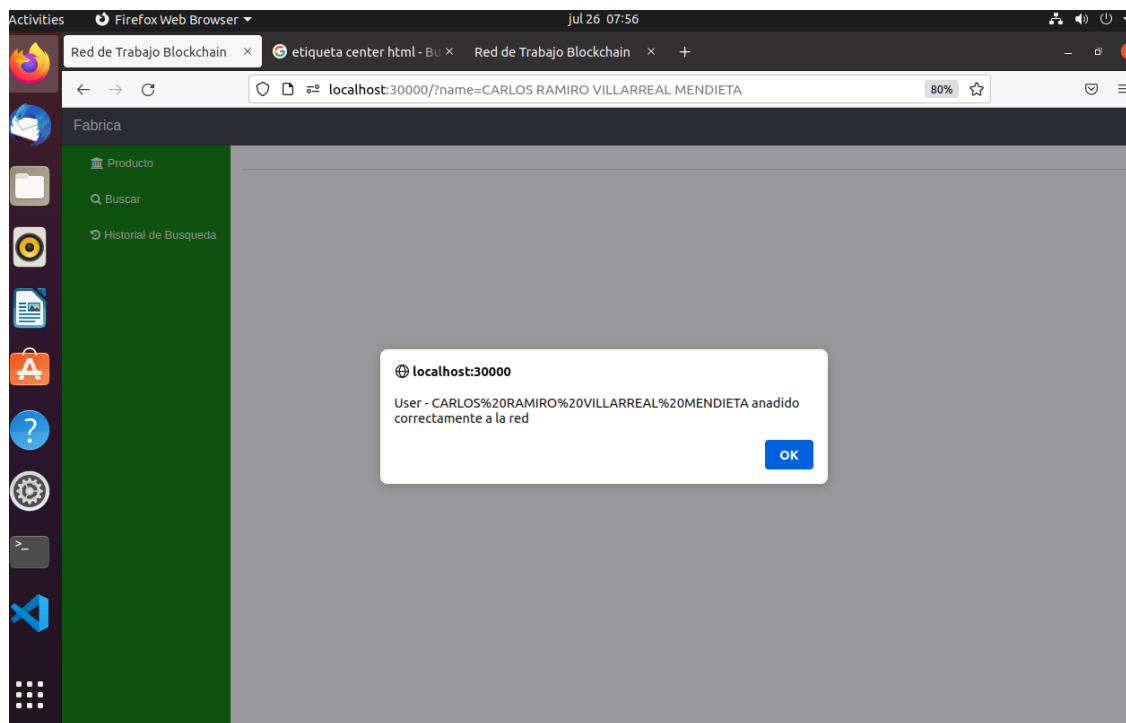
Figura 19

Acceso a la Aplicación



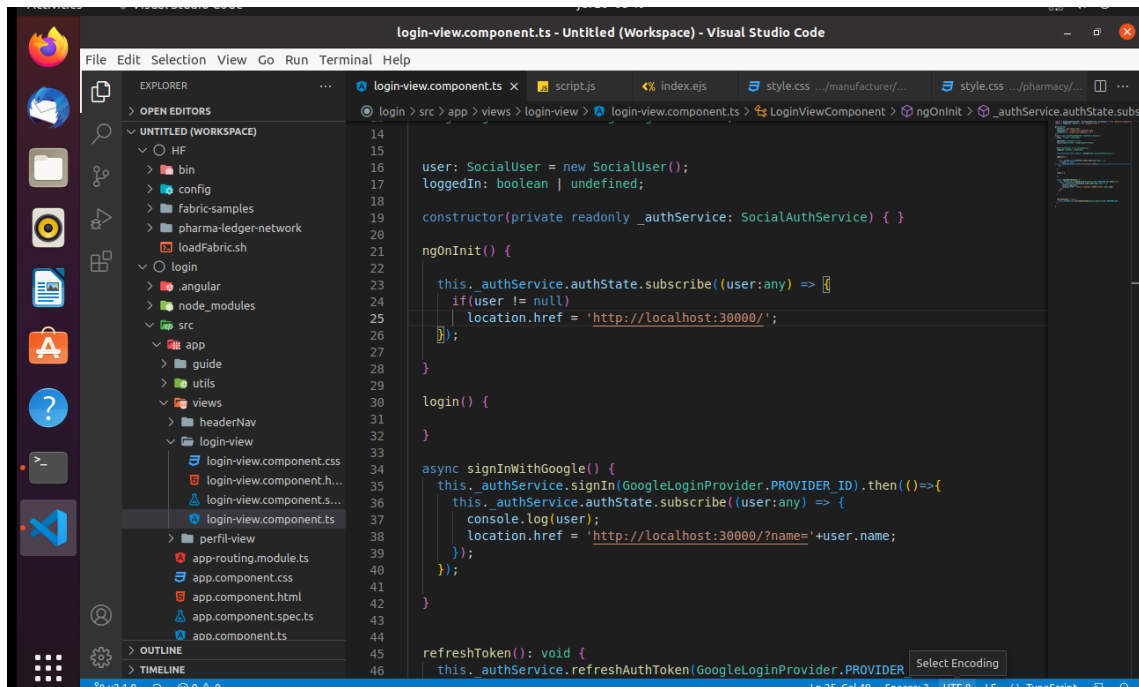
Una vez logueado correctamente como podemos observar en la Figura 20, nos aparece el nombre del usuario que accedió al sistema permitiendo registrar todos los procesos o transacciones que haga ese usuario ya que se grabara en la wallet asignada al mismo.

Figura 20
Ingreso a la Aplicación



Es importante recalcar que solo podrá acceder al sistema teniendo una cuenta creada en Google ya que la aplicación utiliza servicios de AuTh0 para el manejo y gestión de usuarios, otra razón para la implementación de este tipo de autenticación es la seguridad que brinda a la confidencialidad de los datos. En esta sección podemos observar la codificación del módulo de logueo en cual el cual se usa los servicios de Google, ver Figura 21.

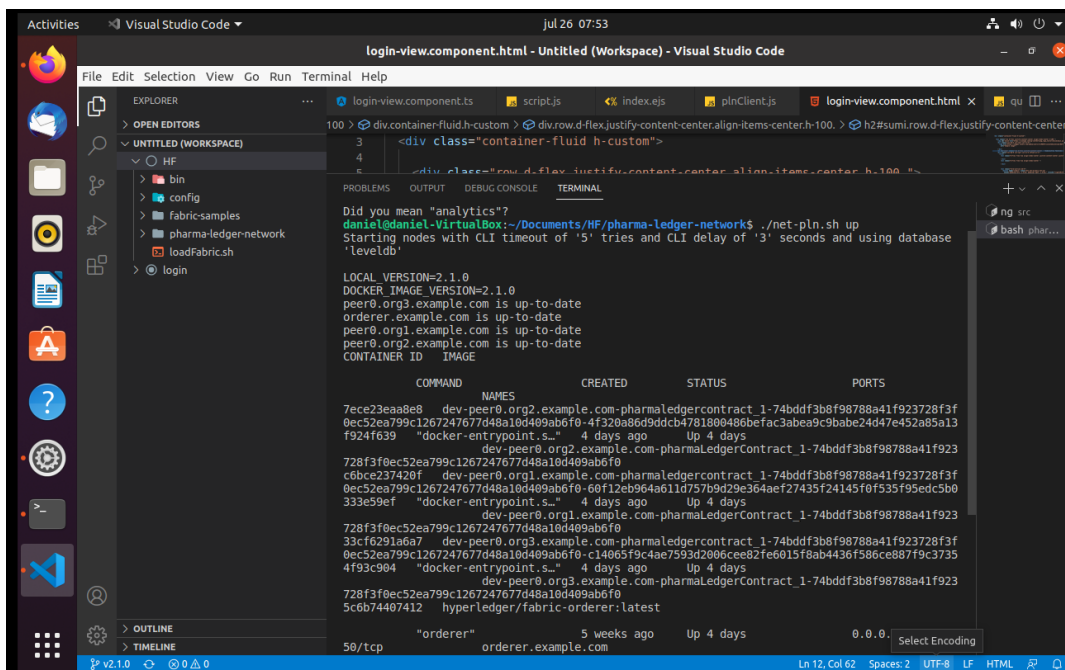
Figura 21

Módulo de Logueo**Proceso de transición**

En esta sección se debe asegurar que la red esta levantada y corriendo para poder ejecutar cualquier registro de los productos, en la siguiente Figura 22, se puede apreciar cómo se ha inicializado todos los contenedores que tiene nuestra red y su estatus además nos indica los puertos que está utilizando.

Figura 22

Inicio de contenedores



The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal displays the output of a Docker command, including the following text:

```

Did you mean "analytics"?
daniel@daniel-VirtualBox:~/Documents/HF/pharma-ledger-networks$ ./net-pln.sh up
Starting nodes with CLI timeout of '5' tries and CLI delay of '3' seconds and using database
'leveldb'

LOCAL VERSION=2.1.0
DOCKER_IMAGE_VERSION=2.1.0
peer0.org3.example.com is up-to-date
orderer.example.com is up-to-date
peer0.org1.example.com is up-to-date
peer0.org2.example.com is up-to-date
CONTAINER ID        IMAGE

```

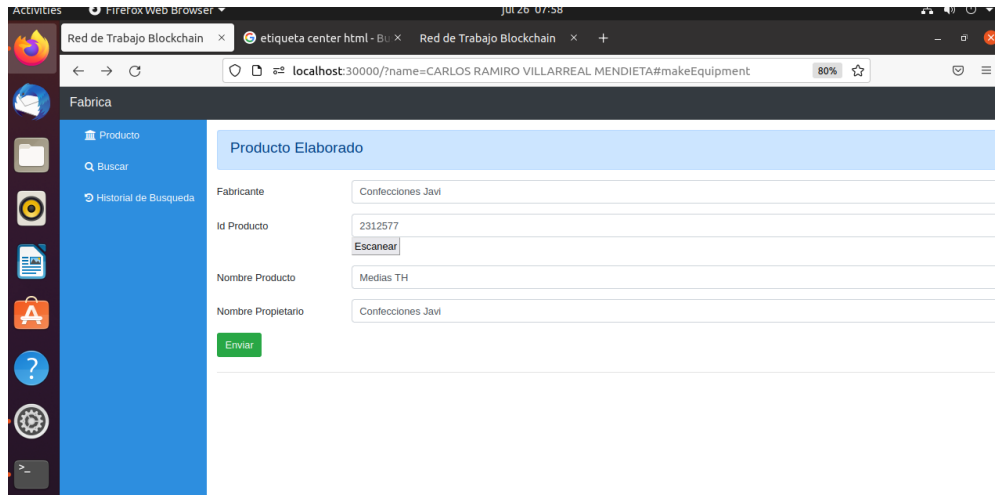
Below the text, a table lists the running containers:

COMMAND	NAMES	CREATED	STATUS	PORTS				
	7ece23eaa8e8 dev-peer0.org2.example.com-pharmalidgercontract	1-74bddf3b8f98788a41f923728f3f0ec52ea799c1267247677d48a10d409ab6f0-4f320a86d9ddcb4781800486befac3abea9c9babe24d47e452a85a13f924f639	4 days ago	Up 4 days				
	728f3f0ec52ea799c1267247677d48a10d409ab6f0	dev-peer0.org2.example.com-pharmalidgercontract	1-74bddf3b8f98788a41f923c6bce237420f	dev-peer0.org1.example.com-pharmalidgercontract	1-74bddf3b8f98788a41f923728f3f0ec52ea799c1267247677d48a10d409ab6f0-60f12eb964a611d757b9d29e364aef27435f24145f0f535f95edc5b0333e59ef	4 days ago	Up 4 days	
	728f3f0ec52ea799c1267247677d48a10d409ab6f0	dev-peer0.org1.example.com-pharmalidgercontract	1-74bddf3b8f98788a41f92333cf6291a6a7	dev-peer0.org3.example.com-pharmalidgercontract	1-74bddf3b8f98788a41f923728f3f0ec52ea799c1267247677d48a10d409ab6f0-c14065f9c4ae7593d2006cee82fe6015f8ab4436f586ce88779c37354f93c904	4 days ago	Up 4 days	
	728f3f0ec52ea799c1267247677d48a10d409ab6f0	dev-peer0.org3.example.com-pharmalidgercontract	1-74bddf3b8f98788a41f9235c6b74407412	hyperledger/fabric-orderer:latest				
	"orderer"	5 weeks ago	Up 4 days	0.0.0:	Select Encoding			

En primera instancia tenemos el módulo del fabricante (ver Figura 23) en el cual queda registrado el usuario que accede a la plataforma, el usuario de la empresa Confecciones Javi puede registrar un producto en donde le pedirá ciertos parámetros que deberán ser llenados, uno de esos es el id del producto el cual puede ser ingresado manualmente o escaneando el código de barras de cada producto.

Figura 23

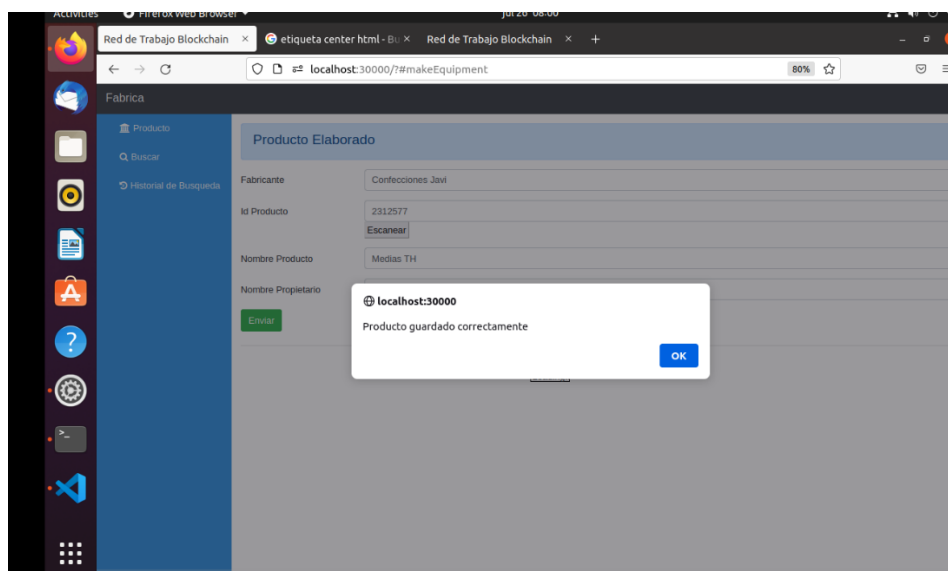
Módulo fabricante



Al dar enviar el sistema verificará que el usuario se encuentre creado en la wallet, en caso de que no se registre se envía un mensaje de error por otra parte si el usuario fue correctamente registrado el producto se guarda satisfactoriamente y todo el registro queda en la wallet encriptado (ver Figura 24).

Figura 24

Guardar producto



Se puede consultar los datos de la cadena de suministro, solo es necesario ingresar el id del producto, después se despliega todos los datos relacionados con ese producto entre los más importantes se puede observar el propietario actual u el anterior para poder así ver el monitoreo del mismo. Ver Figura 25.

Figura 25

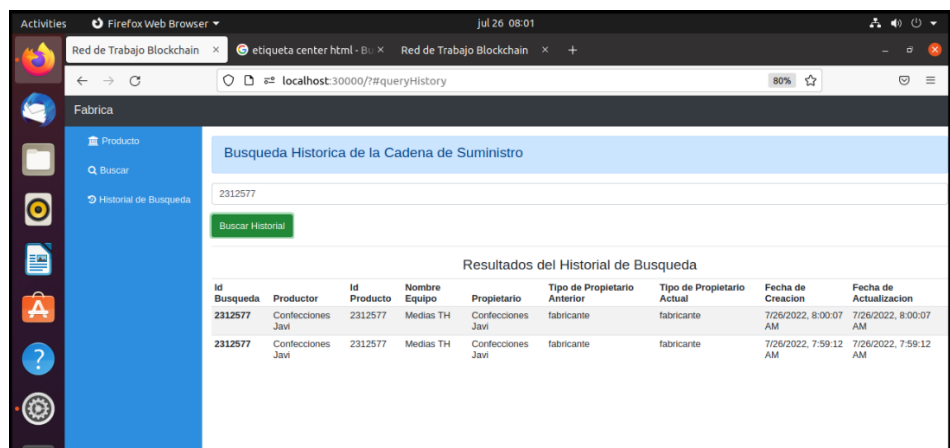
Consultar producto



Otra parte del menú del fabricante es la búsqueda histórica de la cadena de suministro, de igual manera tan solo con el id del producto se puede observar todo el historial del producto. Ver Figura 26.

Figura 26

Búsqueda histórica del producto



En la Figura 27, a continuación, se muestra el módulo del distribuidor de igual manera se puede añadir el producto a la wallet, en este caso se agrega el mismo producto para poder observar la trazabilidad del mismo y como se puede identificar, cambia automáticamente de responsabilidad es decir ahora el dueño actual pasa a ser el distribuidor y la responsabilidad o custodio recae en él.

Figura 27

Cambio de estado del producto al Distribuidor



Resultado de Busqueda	
Id Busqueda	2312577
Productor	Confecciones Javi
Id Producto	2312577
Nombre Producto	Medias TH
Propietario	Confecciones Javi
Tipo de Propietario Anterior	Fabricante
Tipo de Propietario Actual	Distribuidor
Fecha de Creacion del Producto	7/26/2022, 8:00:07 AM
Ultima fecha de actualizacion	7/26/2022, 9:00:07 AM

En la Figura 28, se muestra el módulo del punto de venta de igual manera se puede añadir el producto a la wallet, en este caso se agrega el mismo producto para poder observar la trazabilidad del mismo y como se puede identificar cambia automáticamente de responsabilidad es decir ahora el dueño actual pasa a ser el distribuidor y la responsabilidad carga sobre él.

Figura 28

Cambio de estado del producto al Punto de Venta



The screenshot displays a web application interface for a 'Punto de Venta' (Point of Sale) system. On the left, a vertical navigation menu contains icons for 'Añadir', 'Recibidor', 'Busqueda', 'Historial de Busqueda', and other functions. The main content area is titled 'Busqueda de la Cadena de Suministro' (Supply Chain Search). A search input field contains the value '2312577', and a green 'Busqueda' button is positioned below it. The search results are displayed under the heading 'Resultado de Busqueda' (Search Result) and are presented in a table format.

Id Busqueda	2312577
Productor	Confecciones Javi
Id Producto	2312577
Nombre Producto	Medias TH
Propietario	Confecciones Javi
Tipo de propietario anterior	Distribuidor
Tipo de propietario actual	Punto de Venta
Fecha de creacion	Tue Jul 26 2022 13:00:07 GMT+0000 (Coordinated Universal Time)
Fecha de modificacion	Tue Jul 26 2022 13:00:07 GMT+0000 (Coordinated Universal Time)

Capítulo 6

Análisis e interpretación de resultados

Análisis Cuantitativo de Datos

En este capítulo, se presentan los resultados de evaluación de la aplicabilidad de la tecnología Blockchain en la Cadena de Suministro en la empresa Confecciones Javi, mediante la realización de una encuesta pre-post al personal de bodega dentro de la empresa textil, con la finalidad de determinar si existe mejora en la trazabilidad de ruta de sus productos y afines. Se ha escogido a la encuesta como herramienta de investigación. Debido al número reducido de trabajadores de la empresa, la encuesta se ha aplicado a los 10 empleados entre operarios, recepcionistas y personal de venta. Las preguntas realizadas fueron elaboradas por las personas a cargo de la elaboración del presente proyecto. Cada pregunta forma parte de un cuestionario, que permitió recoger información mediante las respuestas proporcionadas por cada encuestado.

Antes de desarrollar la encuesta fue necesario identificar el contexto, el cual es las personas que utilizarán el sistema dentro de la empresa. Es por ello que, se ha clasificado al personal en dos grupos: Usuarios Intermedios y Usuarios Inexpertos. Los Usuarios Intermedios son aquellos que tienen conocimientos moderados en el manejo de aplicaciones web o móviles y a la vez son quienes frecuentemente utilizan un dispositivo tecnológico dentro de la empresa (generalmente personal de oficina). Mientras tanto, los Usuarios Inexpertos son aquellos que tienen conocimientos básicos de informática y por ende la manipulación de aplicaciones se les hace dificultoso (generalmente personal de bodega y operarios). Cada uno de los participantes respondió a 5 preguntas diferentes, cuya escala va de 1 a 5, donde 1 significa que el usuario se encuentra totalmente en desacuerdo y 5 es la calificación más alta, lo que significa que el usuario está totalmente de acuerdo en el uso de la aplicación. Los

cuestionarios se encuentran elaborados en Google Forms, por lo cual se enviará a cada participante el link correspondiente para la elaboración de la encuesta.

El modelo de Análisis de Varianza (ANOVA) permitió comparar las varianzas entre las medias de los dos grupos. Para el presente proyecto se ha dividido la población en dos grupos: Usuarios Intermedios y Usuarios Inexpertos. A cada uno de ellos, se le ha realizado una encuesta individual con 5 preguntas para medir el nivel de complejidad del sistema dentro de la empresa textil. El modelo ANOVA permitió comparar las medias de los grupos para determinar si existen varianzas estadísticas, esto lo realiza mediante el uso de la Distribución F, la cual permite concluir si la hipótesis nula es aceptada o rechazada.

Resultados de la Encuesta

Usuarios Inexpertos

Pregunta 1.

¿Cuál ha sido la primera impresión al utilizar el sistema?

En la encuesta realizada a 5 personas referente a la pregunta 1, se determina que la primera impresión al sistema fue favorable debido a que la hipótesis nula se acepta, los resultados obtenidos son: el valor de la F teórico con 4 y 10 grados de libertad, a un nivel de confianza del 95% es 5.96. Como $F_{0.05}(4,10) = 7.50$ ($F_{teórica}$) entonces, como $F_{real} < F_{teórica}$ $5.96 < 7.50$, por consiguiente, se acepta la hipótesis nula y se concluye que los resultados de las encuestas son similares. Ver resultados en la Tabla 2 de Usuarios Inexpertos.

Pregunta 2.***¿Le gustaría utilizar el sistema con frecuencia?***

En la encuesta realizada a 5 personas referente a la pregunta 2, se determina que el uso frecuente del sistema sería aceptado por los empleados de la empresa ya que la hipótesis nula se acepta, los resultados obtenidos son: el valor de la F teórico con 4 y 10 grados de libertad, a un nivel de confianza del 95% es 5.96. Como $F_{0.05}(4,10) = 7.52$ ($F_{teórica}$) entonces, como $F_{real} < F_{teórica}$ $5.96 < 7.52$, por consiguiente, se acepta la hipótesis nula y se concluye que los resultados de las encuestas son similares. Ver resultados en la Tabla 3 de Usuarios Inexpertos.

Pregunta 3.***¿La búsqueda de información en el sistema es simple?***

En la encuesta realizada a 5 personas referente a la pregunta 3, debido a que la hipótesis nula se acepta, entonces se puede aseverar que, la búsqueda de información al utilizar el sistema es fácil de realizar. Los resultados obtenidos son: el valor de la F teórico con 4 y 6 grados de libertad, a un nivel de confianza del 95% es 6.16. Como $F_{0.05}(4,6) = 5.61$ ($F_{teórica}$) entonces, como $F_{real} > F_{teórica}$ $6.16 > 5.61$, por consiguiente, se rechaza la hipótesis nula y se concluye que los resultados de las encuestas presentan diferencias significativas. Ver resultados en la Tabla 4 de Usuarios Inexpertos.

Pregunta 4.***¿Cuál ha sido su grado de aceptación en la manipulación del sistema?***

En la encuesta realizada a 5 personas referente a la pregunta 4, la hipótesis nula se acepta y por ende el grado de aceptación del sistema es favorable. Los resultados obtenidos son: el valor de la F teórico con 4 y 6 grados de libertad, a un nivel de confianza del 95% es 6.16. Como $F_{0.05}(4,6) = 2.86$ ($F_{teórica}$) entonces, como $F_{real} > F_{teórica}$ $6.16 < 2.86$, por consiguiente, se rechaza la hipótesis

nula y se concluye que los resultados de las encuestas presentan diferencias significativas. Ver resultados en la Tabla 5 de Usuarios Inexpertos.

Pregunta 5.

¿Cree que necesitaría el apoyo de un técnico para utilizar el sistema?

En la encuesta realizada a 5 personas referente a la pregunta 5, la hipótesis nula se acepta y por ende el personal no requiere de gran ayuda técnica para manipular el sistema. Los resultados obtenidos son: el valor de la F teórico con 4 y 161 grados de libertad, a un nivel de confianza del 95% es 5.66. Como $F_{0.05}(4,161) = 16.56$ ($F_{teórica}$) entonces, como $F_{real} < F_{teórica}$ $5.66 < 16.56$, por consiguiente, se acepta la hipótesis nula y se concluye que los resultados de las encuestas son similares. Ver resultados en la Tabla 6 de Usuarios Inexpertos.

Usuarios Intermedios

Pregunta 1.

¿Encontré el sistema innecesariamente complejo?

En la encuesta realizada a 5 personas referente a la pregunta 1, la hipótesis nula se acepta y se puede aseverar que el manejo del sistema no es complejo. Los resultados obtenidos son: el valor de la F teórico con 4 y 6 grados de libertad, a un nivel de confianza del 95% es 4.87. Como $F_{0.05}(4,6) = 6.16$ ($F_{teórica}$) entonces, como $F_{real} < F_{teórica}$ $4.87 < 6.16$, por consiguiente, se acepta la hipótesis nula y se concluye que los resultados de las encuestas son similares. Ver resultados en la Tabla 7 de Usuarios Intermedios.

Pregunta 2.***¿Encontré que las diversas funciones del sistema estaban bien integradas?***

En la encuesta realizada a 5 personas referente a la pregunta 2, la hipótesis nula se acepta y se concluye que, la disposición de cada función dentro del sistema mantiene un grado aceptable de aceptación por parte del personal de la empresa. Los resultados obtenidos son: el valor de la F teórico con 4 y 6 grados de libertad, a un nivel de confianza del 95% es 5.42. Como $F_{0.05}(4,6) = 6.16$ ($F_{teórica}$) entonces, como $F_{real} < F_{teórica}$ $5.42 < 6.16$, por consiguiente, se acepta la hipótesis nula y se concluye que los resultados de las encuestas son similares. Ver resultados en la Tabla 8 Usuarios Intermedios.

Pregunta 3.***¿Cree que el sistema ayudaría a minimizar el manejo de documentos físicos?***

En la encuesta realizada a 5 personas referente a la pregunta 3, la hipótesis nula se acepta y se concluye que, el sistema ayuda de forma significativa a la reducción de la manipulación de documentos físicos. Los resultados obtenidos son: el valor de la F teórico con 4 y 4 grados de libertad, a un nivel de confianza del 95% es 2.07. Como $F_{0.05}(4,4) = 6.39$ ($F_{teórica}$) entonces, como $F_{real} < F_{teórica}$ $2.07 < 6.39$, por consiguiente, se acepta la hipótesis nula y se concluye que los resultados de las encuestas son similares. Ver resultados en la Tabla 9 Usuarios Intermedios.

Pregunta 4.***¿Me gustaría utilizar el sistema con frecuencia?***

En la encuesta realizada a 5 personas referente a la pregunta 4, la hipótesis nula se acepta y por ende se concluye que, los encargados de utilizar el sistema lo harían con frecuencia. Los resultados obtenidos son: el valor de la F teórico con 4 y 5 grados de libertad, a un nivel de confianza del 95% es

3.98. Como $F_{0.05}(4,5) = 6.26$ ($F_{teórica}$) entonces, como $F_{real} < F_{teórica}$ $3.98 < 6.26$, por consiguiente, se acepta la hipótesis nula y se concluye que los resultados de las encuestas son similares. Ver resultados en la Tabla 10 Usuarios Intermedios.

Pregunta 5.

¿Necesito aprender muchas cosas antes de poder manipular el sistema?

En la encuesta realizada a 5 personas referente a la pregunta 5, la hipótesis nula se acepta y se concluye que, la manipulación del sistema por parte de los empleados es relativamente fácil y su aprendizaje de igual manera. Los resultados obtenidos son: el valor de la F teórico con 5 y 8 grados de libertad, a un nivel de confianza del 95% es 3.5. Como $F_{0.05}(5,8) = 4.82$ ($F_{teórica}$) entonces, como $F_{real} < F_{teórica}$ $3.5 > 4.82$, por consiguiente, se acepta la hipótesis nula y se concluye que los resultados de las encuestas son similares. Ver resultados en la Tabla 11 Usuarios Intermedios.

Tabla 2

Pregunta 1 (Usuario Inexperto)

Nivel	Total	Media	Diferencia de medias entre grupos	Varianza entre grupos	Diferencia de medias dentro de los grupos	Varianza dentro de los grupos
1	0	0	-1,60	12,80	0	0
2	0	0	-1,60	12,80	0	0
3	12	4,00	2,40	28,80	8,00	64,00
4	4	4,00	2,40	28,80	0	0

Nivel	Total	Media	Diferencia de medias entre grupos	Varianza entre grupos	Diferencia de medias dentro de los grupos	Varianza dentro de los grupos
5	0	0	-1,60	12,80	0	0
Promedio		1,60	Q_E	96,00	Q_R	64,00

De acuerdo a los resultados obtenidos de la variabilidad entre grupos y dentro de los grupos, se puede encontrar los siguientes valores: $S_E^2 = 24,00$; $S_R^2 = 3,20$ y $F_{exp} = 7,50$.

Tabla 3

Pregunta 2 (Usuario Inexperto)

Nivel	Total	Media	Diferencia de medias entre grupos	Varianza entre grupos	Diferencia de medias dentro de los grupos	Varianza dentro de los grupos
1	0	0	-1,70	14,45	0	0
2	0	0	-1,70	14,45	0	0
3	9	3,00	1,30	8,45	6,00	36,00
4	8	4,00	2,30	26,45	4,00	16,00
5	0	0	-1,70	14,45	0,00	0
Promedio		1,70	Q_E	78,25	Q_R	52,00

De acuerdo a los resultados obtenidos de la variabilidad entre grupos y dentro de los grupos, se puede encontrar los siguientes valores: $S_E^2 = 19,56$; $S_R^2 = 2,60$ y $F_{exp} = 7,52$.

Tabla 4

Pregunta 3 (Usuario Inexperto)

Nivel	Total	Media	Diferencia de medias entre grupos	Varianza entre grupos	Diferencia de medias dentro de los grupos	Varianza dentro de los grupos
1	0	0	-2,30	26,45	0	0
2	0	0	-2,30	26,45	0	0
3	0	0	-2,30	26,45	0	0
4	8	4,00	1,70	14,45	4,00	16,00
5	15	5,00	2,70	36,45	10,00	100,00
	Promedio	2,30	Q_E	130,25	Q_R	116,00

De acuerdo a los resultados obtenidos de la variabilidad entre grupos y dentro de los grupos, se puede encontrar los siguientes valores: $S_E^2 = 32,56$; $S_R^2 = 5,80$ y $F_{exp} = 5,61$.

Tabla 5

Pregunta 4 (Usuario Inexperto)

Nivel	Total	Media	Diferencia de medias entre grupos	Varianza entre grupos	Diferencia de medias dentro de los grupos	Varianza dentro de los grupos
1	0	0	0	26,45	0	0
2	0	0	0	26,45	0	0
3	3	3,00	3,00	26,45	0	0
4	16	4,00	4,00	14,45	12,00	144,00
5	0	0	0	36,45	0	0
Promedio		1,90	Q_E	82,25	Q_R	144,00

De acuerdo a los resultados obtenidos de la variabilidad entre grupos y dentro de los grupos, se puede encontrar los siguientes valores: $S_E^2 = 20,56$; $S_R^2 = 7,20$ y $F_{exp} = 2,86$.

Tabla 6

Pregunta 5 (Usuario Inexperto)

Nivel	Total	Media	Diferencia de medias entre grupos	Varianza entre grupos	Diferencia de medias dentro de los grupos	Varianza dentro de los grupos
1	0	0	-1,5	11,25	0	0
2	4	2,00	0,50	1,25	2	4,00
3	3	3,00	1,50	11,25	0	0
4	8	4,00	2,50	31,25	4,00	16,00
5	0	0	-1,50	11,25	0	0
Promedio		1,90	Q_E	66,25	Q_R	20,00

De acuerdo a los resultados obtenidos de la variabilidad entre grupos y dentro de los grupos, se puede encontrar los siguientes valores: $S_E^2 = 16,56$; $S_R^2 = 1,00$ y $F_{exp} = 16,56$.

Tabla 7

Pregunta 1 (Usuario Intermedio)

Nivel	Total	Media	Diferencia de medias entre grupos	Varianza entre grupos	Diferencia de medias dentro de los grupos	Varianza dentro de los grupos
1	0	0	-2,2	33,88	0	0
2	10	2,00	-0,2	0,28	8,00	64,00
3	0	0	-2,2	33,28	0	0
4	12	4,00	1,80	22,68	8,00	64,00
5	0	0	-2,2	33,88	0	0
Promedio		2,20	Q_E	124,60	Q_R	128,00

De acuerdo a los resultados obtenidos de la variabilidad entre grupos y dentro de los grupos, se puede encontrar los siguientes valores: $S_E^2 = 31,15$; $S_R^2 = 6,40$ y $F_{exp} = 4,87$.

Tabla 8

Pregunta 2 (Usuario Intermedio)

Nivel	Total	Media	Diferencia de medias entre grupos	Varianza entre grupos	Diferencia de medias dentro de los grupos	Varianza dentro de los grupos
1	0	0	-2,90	58,87	0	0
2	0	0	-2,90	58,87	0	0
3	12	3,00	0,10	0,07	9,00	81,00
4	12	4,00	1,10	8,47	8,00	64,00
5	5	5,00	2,10	30,87	0	0
Promedio		2,90	Q_E	157,15	Q_R	145,00

De acuerdo a los resultados obtenidos de la variabilidad entre grupos y dentro de los grupos, se puede encontrar los siguientes valores: $S_E^2 = 39,29$; $S_R^2 = 7,25$ y $F_{exp} = 5,42$.

Tabla 9

Pregunta 3 (Usuario Intermedio)

Nivel	Total	Media	Diferencia de medias entre grupos	Varianza entre grupos	Diferencia de medias dentro de los grupos	Varianza dentro de los grupos
1	0	0	-2,70	51,03	0	0
2	0	0	-2,70	51,03	0	0
3	3	3,00	0,30	0,63	0	0
4	24	4,00	1,30	11,83	20,00	400,00
5	0	0	-2,70	51,03	0	0
Promedio		2,70	Q_E	165,55	Q_R	400,00

De acuerdo a los resultados obtenidos de la variabilidad entre grupos y dentro de los grupos, se puede encontrar los siguientes valores: $S_E^2 = 41,39$; $S_R^2 = 20,00$ y $F_{exp} = 2,07$.

Tabla 10

Pregunta 4 (Usuario Intermedio)

Nivel	Total	Media	Diferencia de medias entre grupos	Varianza entre grupos	Diferencia de medias dentro de los grupos	Varianza dentro de los grupos
1	0	0	-2,70	51,03	0	0
2	0	0	-2,70	51,03	0	0
3	15	3,00	0,30	0,63	12,00	144,00
4	12	4,00	1,30	11,83	8,00	64,00
5	0	0	-2,70	51,03	0	0
Promedio		2,70	Q_E	165,55	Q_R	208,00

De acuerdo a los resultados obtenidos de la variabilidad entre grupos y dentro de los grupos, se puede encontrar los siguientes valores: $S_E^2 = 41,39$; $S_R^2 = 10,40$ y $F_{exp} = 3,98$.

Tabla 11

Pregunta 5 (Usuario Intermedio)

Nivel	Total	Media	Diferencia de medias entre grupos	Varianza entre grupos	Diferencia de medias dentro de los grupos	Varianza dentro de los grupos
1	2	1,00	-0,80	4,48	1,00	1,00
2	4	2,00	0,20	0,28	2,00	4,00
3	12	3,00	1,20	10,08	9,00	9,00
4	0	0	-1,80	22,68	0	0
5	0	0	-1,80	22,68	0	0
Promedio		1,80	Q_E	60,20	Q_R	86,00

De acuerdo a los resultados obtenidos de la variabilidad entre grupos y dentro de los grupos, se puede encontrar los siguientes valores: $S_E^2 = 15,05$; $S_R^2 = 4,30$ y $F_{exp} = 3,50$.

Capítulo 7

Conclusiones, recomendaciones y trabajos futuros

Conclusiones

- A lo largo de la Implementación de Blockchain dentro de la Cadena de Suministros, se pudo comprobar que esta tecnología se relaciona de manera directa con otras, en especial en el área de seguridad y base de datos. Cada una de las cuales aporta al fortalecimiento de la cadena de bloques y por ende se convierte en un aliado estratégico para la industria 4.0.
- El proyecto fue enfocado al área textil, ya que Blockchain podría ser implementado en una diversidad de escenarios. En la encuesta realizada al personal de la empresa, respecto a la pregunta: ¿Cree que el sistema ayudaría a minimizar el manejo de documentos físicos?. Mediante la utilización del Modelo Anova se pudo determinar que en la Distribución F con un nivel de confianza del 95% es 2.07 y el valor teórico es 6.39. Por lo tanto, la F teórica(6.39) es mayor al valor F real(2.07) y por consiguiente se acepta la hipótesis.
- Si bien Blockchain permite descentralizar la información, varias empresas no están preparadas para dar este gran cambio hacia la implementación de una nueva tecnología. Ya sea por temor, desconocimiento o simplemente porque disponen de su propio activo fijo. Con lo cual resulta complicado evaluar de manera directa las herramientas o tecnologías de una empresa por motivos de confidencialidad. Pero, por la experiencia adquirida y la investigación realizada, se podría asegurar que en su mayoría disponen de sistemas centralizados. Con lo cual, se ha podido identificar un gran campo de emprendimiento e investigación hacia la creación de nuevos modelos tecnológicos basados en Blockchain.
- La clave de usar la tecnología Blockchain es usar los contratos inteligentes, ya que ayudan a validar los términos y condiciones en las cuales un producto está regido bajo normas de

procesamiento, recalcando nuevamente en la transparencia y seguridad, provocando una reacción en el cliente de satisfacción ya que podrá saber el estado del producto debido a que las transacciones están registradas y se actualizan con cada operación en un tiempo instantáneo.

Recomendaciones

- Las tecnologías disruptivas como Blockchain han permitido el apareamiento de nuevos productos y formas de abordar escenarios que anteriormente carecían de seguridad, si bien cada tecnología tiene su propósito. El estudio de este tipo de nuevas tecnologías debería ser constante, ya que su aplicación tiene un enorme potencial y campo de acción dentro de cualquier empresa o emprendimiento.
- Al aplicar la tecnología Blockchain dentro de un escenario, es recomendable conocer las reglas de negocio que rigen dicho escenario. Con la finalidad de poder crear e implementar de forma eficiente los requerimientos previstos y estos plasmarlos dentro del chaincode o contrato inteligente.
- Si bien se ha evaluado la usabilidad del sistema mediante la elaboración de un cuestionario al personal de la empresa, es recomendable realizar la obtención de datos cualitativos de los usuarios que utilizaron el sistema mediante herramientas como GazeRecorder. Esta permitirá obtener el mapa de calor de las regiones visitadas por cada usuario dentro de la pantalla, con lo cual se podrá obtener información para mejorar el diseño de interfaz de usuario.

Trabajo Futuro

La implementación de la tecnología Blockchain dentro del área textil podría ser implementada en varios aspectos dentro de la empresa Confecciones Javi. Uno de ellos es automatizar el control de producción y personal. Si bien disponen de dispositivos que controlan su ingreso y producción, estos se ven afectados al momento de realizar el mantenimiento del equipo, produciendo que los registros

puedan ser eliminados involuntariamente por el personal de mantenimiento. Si a futuro, cada uno de estos registros fueran obtenidos por la red Blockchain y por ende se almacenaran dentro de la ledger o libro mayor, permanecerían inmutables y se guardaría información detallada de cada proceso productivo, así como del personal. Esto permitiría a la empresa, ayudar a la toma de decisiones en beneficio de la misma.

Bibliografía

- Alkhader, W., Alkaabi, N., Salah, K., Jayaraman, R., Arshad, J., & Omar, M. (2020). Blockchain-Based Traceability and Management for Additive Manufacturing. *IEEE Access*, 8, 188363-188377. <https://doi.org/10.1109/ACCESS.2020.3031536>
- Alkhader, W., Salah, K., Sleptchenko, A., Jayaraman, R., Yaqoob, I., & Omar, M. (2021). Blockchain-Based Decentralized Digital Manufacturing and Supply for COVID-19 Medical Devices and Supplies. *IEEE Access*, 9, 137923-137940. <https://doi.org/10.1109/ACCESS.2021.3118085>
- Amazon AWS. (2020). *Contenedores de Docker | ¿Qué es Docker? | AWS*. Amazon Web Services, Inc. <https://aws.amazon.com/es/docker/>
- Apache CouchDB. (2013, September 29). 1. Introduction—Apache CouchDB® 3.1 Documentation. <https://docs.couchdb.org/en/stable/intro/index.html>
- Arcia, M. (2018, July 17). *Cadena de suministro, qué es y cómo funciona*. Entrepreneur. <https://www.entrepreneur.com/article/316908>
- AWS. (2018, June 21). *Bases de datos no relacionales | Bases de datos de gráficos | AWS*. Amazon Web Services, Inc. <https://aws.amazon.com/es/nosql/>
- Blockchain, clave durante la pandemia: Así se usa para monitorear el abastecimiento de dispositivos médicos*. (2020, June 12). <https://www.iproup.com/economia-digital/14474-blockchain-y-medicina-por-que-crece-su-uso-durante-la-pandemia>. <https://www.iproup.com/economia-digital/14474-blockchain-y-medicina-por-que-crece-su-uso-durante-la-pandemia>
- Cocco, L., Mannaro, K., Tonelli, R., Mariani, L., Lodi, M. B., Melis, A., Simone, M., & Fanti, A. (2021). A Blockchain-Based Traceability System in Agri-Food SME: Case Study of a Traditional Bakery. *IEEE Access*, 9, 62899-62915. <https://doi.org/10.1109/ACCESS.2021.3074874>
- Delgado, D. T. (2019). PROPUESTA USO DE LA TECNOLOGÍA BLOCKCHAIN EN LA CADENA DE SUMINISTRO DE MEDICAMENTOS. 55.

Docker. (2020, December 18). *Overview of Docker Compose*. Docker Documentation.

<https://docs.docker.com/compose/>

El modelo de desarrollo en espiral como mezcla de cascada e iterativo. (2019, April 5). ASPgems.

<https://aspgems.com/metodologia-de-desarrollo-de-software-iii-modelo-en-espiral/>

Esparza Maldonado, A., Rodriguez, F., & Ponce, J. (2015, October 26). *Modelo del Proceso del Software para la creación de Aplicaciones Móviles utilizando Técnicas de Gamificación para el Desarrollo de Psicomotricidad Fina Viso-Manual*.

Estaún, M. (2020). La Cadena de Gestión de Suministro (SCM): Qué es y cuáles son las ventajas que ofrece. *Thinking for Innovation*. <https://www.iebschool.com/blog/cadena-gestion-suministro-negocios-internacionales/>

García, H. C. E. (2020). Tecnología blockchain en cadenas de suministro; mito, realidad o proyección.

Revista de la Facultad de Ciencias Económicas, 24(1), 173–188.

<https://doi.org/10.30972/rfce.2414366>

Gartner 2019 Hype Cycle Shows Most Blockchain Technologies Are Still Five to 10 Years Away From Transformational Impact. (2019, October 8). Gartner.

<https://www.gartner.com/en/newsroom/press-releases/2019-10-08-gartner-2019-hype-cycle-shows-most-blockchain-technologies-are-still-five-to-10-years-away-from-transformational-impact>

Go. (2007, October 18). *The Go Programming Language*. <https://golang.org/>

González, A. G. (2015, June 15). ¿Qué es Maven y para qué se utiliza? *Panama Hitek*.

<http://panamahitek.com/que-es-maven-y-para-que-se-utiliza/>

Gutiérrez, A. (2017). *Ingeniería de Software Avanzada*.

<https://cursos.aiu.edu/Ingenieria%20de%20Software%20Avanzada/PDF/Tema%202.pdf>

Hyperledger. (2019, February 21). Walmart Case Study. *Hyperledger*.

- <https://www.hyperledger.org/learn/publications/walmart-case-study>
- Hyperledger Fabric. (2020, December 22). Hyperledger Fabric. *Hyperledger*.
- <https://www.hyperledger.org/use/fabric>
- Keyence. (2018, October 2). *¿Qué es la trazabilidad? | Principios de trazabilidad | KEYENCE México*.
- https://www.keyence.com.mx/ss/products/marketing/traceability/basic_about.jsp
- Lin, Q., Wang, H., Pei, X., & Wang, J. (2019). Food Safety Traceability System Based on Blockchain and EPCIS. *IEEE Access*, 7, 20698-20707. <https://doi.org/10.1109/ACCESS.2019.2897792>
- Lucas, J. (2019, September 4). *Qué es NodeJS y para qué sirve*. OpenWebinars.net.
- <https://openwebinars.net/blog/que-es-nodejs/>
- Meyrick, L. (2018, April 4). *How Blockchain Can Build Stronger Supply Chains*. The One Brief.
- <https://theonebrief.com/blockchain-can-build-stronger-supply-chains/>
- Microsoft Azure. (2020, May 19). *Visual Studio Code – Editor de código | Microsoft Azure*.
- <https://azure.microsoft.com/es-es/products/visual-studio-code/>
- Naranjo Toro, M., Trujillo, C., Lomas Tapia, K., & Milton, M. (2019). *Investigación Cualitativa*.
- https://www.researchgate.net/publication/330683601_Investigacion_Cualitativa
- Pablo, J., Ramírez, V., & Valencia-Ramirez, J. (2019). *CONTRATOS INTELIGENTES SMART CONTRACTS*.
- Pandey, P., & Litoriya, R. (2020). Securing E-health Networks from Counterfeit Medicine Penetration Using Blockchain. *Wireless Personal Communications*. <https://doi.org/10.1007/s11277-020-07041-7>
- Portainer.io. (2020, December 9). *Portainer | Open Source Container Management for Kubernetes, Docker, Swarm*. <https://www.portainer.io>
- Rotolo, D., Hicks, D., & Martin, B. R. (2015). What is an emerging technology? *Research Policy*, 44(10), 1827–1843. <https://doi.org/10.1016/j.respol.2015.06.006>
- Roy, S. (2020, January 30). *Hyperledger Fabric Architecture Reference—Blockchain Automation*

Framework 0.4.0 documentation. <https://blockchain-automation-framework.readthedocs.io/en/latest/architecture/hyperledger-fabric.html>

Salah, K., Nizamuddin, N., Jayaraman, R., & Omar, M. (2019). Blockchain-Based Soybean Traceability in Agricultural Supply Chain. *IEEE Access*, 7, 73295-73305.
<https://doi.org/10.1109/ACCESS.2019.2918000>

Serrano, G. P. (n.d.). *La investigación-acción en la educación formal y no formal*. 22.

Shahid, A., Almogren, A., Javaid, N., Al-Zahrani, F. A., Zuair, M., & Alam, M. (2020). Blockchain-Based Agri-Food Supply Chain: A Complete Solution. *IEEE Access*, 8, 69230-69243.
<https://doi.org/10.1109/ACCESS.2020.2986257>

Villameriel, I. (2020). *Blockchain y Criptomonedas*.

<https://uvadoc.uva.es/bitstream/handle/10324/38340/TFG-J-95.pdf?sequence=1&isAllowed=y>

VMware Cloud on AWS. (2016, October 13). Amazon Web Services, Inc.

<https://aws.amazon.com/es/vmware/>

Wshaffer. (2019, December 5). *Documentación de VMware Cloud on AWS*.

<https://docs.vmware.com/es/VMware-Cloud-on-AWS/index.html>