Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Estudios de Postgrado
Maestría en Tecnologías de la Información y Comunicación

PROTOTIPO DE UNA PLATAFORMA DE GESTIÓN DE DATOS DISTRIBUIDA, PERMANENTE Y SEGURA UTILIZANDO TECNOLOGÍA *BLOCKCHAIN* PARA LOS DATOS DE CIENTÍFICOS Y PROYECTOS DE CIENCIA DE LA SECRETARÍA NACIONAL DE CIENCIA Y TECNOLOGÍA

Ing. Kevin Adiel Lajpop Ajpacajá
Asesorado por el MSc. Ing. Yuri Asusena Castro Estrada

Guatemala, noviembre de 2020

Decanato
Facultad de Ingeniería
24189101 - 24189102

DTG. 395.2020.

La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Estudios de Postgrado, al Trabajo de Graduación titulado: **PROTOTIPO DE UNA PLATAFORMA DE GESTIÓN DE DATOS DISTRIBUIDA, PERMANENTE Y SEGURA UTILIZANDO TECNOLOGÍA *BLOCKCHAIN* PARA LOS DATOS DE CIENTÍFICOS Y PROYECTOS DE CIENCIA DE LA SECRETARÍA NACIONAL DE CIENCIA Y TECNOLOGÍA,** presentado por el Ingeniero **Kevin Adiel Lajpop Ajpacajá,** estudiante de la **Maestría en Tecnologías de la Información y Comunicación** y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DECANA
FACULTAD DE INGENIERÍA

Inga. Anabela Cordova Estrada
Decana

Guatemala, noviembre de 2020.

AACE/asga

Guatemala, Noviembre de 2020

EEPFI-1507-2020

En mi calidad de Director de la Escuela de Estudios de Postgrado de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen y verificar la aprobación del Revisor y la aprobación del Área de Lingüística al Trabajo de Graduación titulado: **"PROTOTIPO DE UNA PLATAFORMA DE GESTIÓN DE DATOS DISTRIBUIDA, PERMANENTE Y SEGURA UTILIZANDO TECNOLOGÍA BLOCKCHAIN PARA LOS DATOS DE CIENTÍFICOS Y PROYECTOS DE CIENCIA DE LA SECRETARÍA NACIONAL DE CIENCIA Y TECNOLOGÍA"** presentado por el Ingeniero **Kevin Adiel Lajpop Ajpacajá** identifica con Carné **201020724** correspondiente al programa de **Maestría en Artes en Tecnologías de la Información y la Comunicación** apruebo y autorizo el mismo.

Atentamente,

"Id y Enseñad a Todos"

DIRECCIÓN

Mtro. Ing. Edgar Darío Álvarez Cotí
Director
Escuela de Estudios de Postgrado
Facultad de Ingeniería
Universidad de San Carlos de Guatemala

Guatemala, Noviembre de 2020

Como Coordinador de la **Maestría en Artes en Tecnologías de la Información y Comunicación** doy el aval correspondiente para la aprobación del Trabajo de Graduación titulado: **"PROTOTIPO DE UNA PLATAFORMA DE GESTIÓN DE DATOS DISTRIBUIDA, PERMANENTE Y SEGURA UTILIZANDO TECNOLOGÍA BLOCKCHAIN PARA LOS DATOS DE CIENTÍFICOS Y PROYECTOS DE CIENCIA DE LA SECRETARÍA NACIONAL DE CIENCIA Y TECNOLOGÍA"** presentado por el Ingeniero **Kevin Adiel Lajpop Ajpacajá** quien se identifica con Carné **201020724**

Atentamente,

"Id y Enseñad a Todos"

Mtro. Ing. Marlon Antonio Pérez Türk
Coordinador de Maestría
Escuela de Estudios de Postgrado
Facultad de Ingeniería
Universidad de San Carlos de Guatemala

Guatemala, Noviembre de 2020

EEPFI-1508-2020

En mi calidad como Asesora del Ingeniero **Kevin Adiel Lajpop Ajpacajá** quien se identifica con Carné **201020724** procedo a dar el aval correspondiente para la aprobación del Trabajo de Graduación titulado: **"PROTOTIPO DE UNA PLATAFORMA DE GESTIÓN DE DATOS DISTRIBUIDA, PERMANENTE Y SEGURA UTILIZANDO TECNOLOGÍA BLOCKCHAIN PARA LOS DATOS DE CIENTÍFICOS Y PROYECTOS DE CIENCIA DE LA SECRETARÍA NACIONAL DE CIENCIA Y TECNOLOGÍA"** quien se encuentra en el programa de **Maestría en Artes en Tecnologías de la Información y la Comunicación** en la Escuela de Estudios de Postgrado de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala.

Atentamente,

"Id y Enseñad a Todos"

MSc. Yuri Asusena Castro Estrada
Asesora

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

PROTOTIPO DE UNA PLATAFORMA DE GESTIÓN DE DATOS DISTRIBUIDA, PERMANENTE Y SEGURA UTILIZANDO TECNOLOGÍA *BLOCKCHAIN* PARA LOS DATOS DE CIENTÍFICOS Y PROYECTOS DE CIENCIA DE LA SECRETARÍA NACIONAL DE CIENCIA Y TECNOLOGÍA

TRABAJO DE GRADUACIÓN

PRESENTADO A JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

ING. KEVIN ADIEL LAJPOP AJPACAJÁ
ASESORADO POR EL MSC. ING. YURI ASUSENA CASTRO ESTRADA

AL CONFERÍRSELE EL TÍTULO DE

MAESTRO EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN

GUATEMALA, NOVIEMBRE DE 2020

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

NÓMINA DE JUNTA DIRECTIVA

DECANA	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Goméz Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Christian Moisés de la Cruz Leal
VOCAL V	Br. Kevin Armando Cruz Lorente
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANA	Inga. Aurelia Anabela Cordova Estrada
DIRECTOR	Ing. Edgar Darío Álvarez Cotí
EXAMINADOR	Ing. Marlon Antonio Peréz Türk
EXAMINADOR	Ing. Edwin Estuardo Zapeta Gómez
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

PROTOTIPO DE UNA PLATAFORMA DE GESTIÓN DE DATOS DISTRIBUIDA, PERMANENTE Y SEGURA UTILIZANDO TECNOLOGÍA *BLOCKCHAIN* PARA LOS DATOS DE CIENTÍFICOS Y PROYECTOS DE CIENCIA DE LA SECRETARÍA NACIONAL DE CIENCIA Y TECNOLOGÍA

Tema que me fuera asignado por la Dirección de la Escuela de Estudios de Posgrado, con fecha 30 marzo de 2019.

Ing. Kevin Adiel Lajpop Ajpacajá

ACTO QUE DEDICO A:

Dios	Por su amor, su misericordia y sus bendiciones que me tienen vivo hasta hoy y por toda la inteligencia y sabiduría que me ha dado.
Mi mamá	Por darme la vida en la tierra, por sus enseñanzas y por perdonarme mis errores y aun así con amor darme sus bendiciones día con día.
Mi papá	Por darme el ejemplo de trabajo duro, de lucha diaria por los sueños y por el perdón de mis errores y confiar en mí a pesar de lo que soy.
Mis hermanos	Por estar en todos los momentos conmigo, por vivir conmigo, por mostrarme los pasos a seguir, por tantas vivencias y aventuras juntos.
Mi tío	Luis Guillermo Ajpacajá Vasquéz (q. e. p. d.) y que su memoria y legado quede por los tiempos de la tierra, de la familia y del país.

AGRADECIMIENTOS A:

Dios	A Dios por darme la inteligencia, la sabiduría y los recursos financieros para culminar los estudios de postgrado.
Mamá	Por ser mi vital inspiración día con día y con paciencia entender mis diferencias con la vida.
Papá	Por su entendimiento en cada meta que concluyo y no separarse de mi a pesar de lo que he sido.
Hermanos	Por todo el apoyo que le han brindado a su hermano menor.
Seres amados	Por estar conmigo lado a lado, en cada desvelada, en cada tarea, en cada noche, por su apoyo en todo momento.
Universidad de San Carlos de Guatemala	Por ser la casa de estudios que me abrió las puertas a una educación pública y de calidad.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES

FIGURAS

TABLAS

GLOSARIO

Amazon AWS	Colección de servicios en la nube que abarca infraestructura, plataforma y software. Dichos servicios son accedidos vía internet.
AWS	Amazon Web Services.
Arquitectura	Es la estructura lógica de un sistema de información, la cual contiene patrones, abstracciones, modelos lógicos y mentales que proporcionan un marco de desarrollo en todas las escalas.
AWS Lambda	Plataforma o servicio proporcionado por Amazon AWS que busca la independencia de infraestructura dado que las funciones son sin servidor y las implementaciones o despliegues están orientados a eventos.
API	Se le denomina así al conjunto de funciones, procedimientos y en sí, *backend* que esta encapsulado en forma de biblioteca para que pueda ser utilizado por otro software.
Backup	Resguardo de datos que alternativo a la fuente original y es obtenido en un determinado momento y está listo para ser utilizado en un momento posterior.

Chaincode	Es el software que va instalado en cada nodo del *blockchain* y es el encargado de interactuar con el *blockchain*.
Consenso	Acuerdo concedido entre un grupo de elementos que quieren llegar a un fin.
Cloud9	IDE en línea de desarrollo de Amazon AWS.
Docker	Proyecto de código abierto que facilita el despliegue de software, pero esto dentro contendedores, brindando un alto nivel de abstracción y encapsulamiento.
DNI	Directorio Nacional de Investigadores.
Encapsulamiento	Es la propiedad de ocultar el estado de un componente de software y estado quiere decir, funcionalidad, datos y políticas, brindando únicamente acceso a lo que se desee.
EC2	*Amazon Elastic Compute Cloud.*
Fonacyt	Fondo Nacional de Ciencia y Tecnología.
Frontend	Es la interfaz en donde se realiza la presentación de datos al usuario u otro sistema, puede ser presentación web, escritorio o formato de datos

	dependiendo de quién sea el consumidor de la información.
GCC	*GNU Compiler Collection.*
HTTP	*Hypertext Transfer Protocol.*
HyperLedger	Es el proyecto de la organización *Umbrella* que implementa el software a código abierto de para el *blockchain*.
IDE	*Integrated Development Environment.*
JSON	Es un formato de texto que sirve para el intercambio de datos en el proceso de comunicación entre dos sistemas.
On Premise	Solución de software que está orientado a una infraestructura o ambiente local, impulsando el uso de servidores físicos yendo en contra de las implementaciones en la nube.
Prototipo	Versión preliminar de un producto o software en la cual van características iniciales, pero no contempla todo el un gran alcance debido al tiempo de implementación o uso que se le desea realizar.
REST	*Representational State Transfer.*

RESTful	Es una interfaz que permite la comunicación entre sistemas de información, dicha comunicación se realiza bajo el protocolo HTTP, dicha interfaz carece de estado y está orientada a la manipulación de la información.
Serverless	Es un modelo de arquitectura el cual propone la computación sin servidor, es decir, tener sistemas de información sin servidores físicos, esto impulsa la implementación de la nube todas sus escalas.
Senacyt	Secretaría Nacional de Ciencia y Tecnología.
SSH	*Secure Shell.*
Transacción	Trato o acuerdo entre dos partes en las cuales ambas partes tienen un interés en particular personal.
URL	*Uniform Resource Locator.*

RESUMEN

En esta era digital, la base de todo son los datos, por lo que cuidar de los mismos también es una tarea básica en la actualidad, cuando se habla de cuidar de estos, se pueden hablar de *backups*, técnicas de manipulación e incluso gobernanza sobre los datos; pero todo esto sigue con un margen de error sobre los datos, margen de error que muchas veces cae en el error humano, vulnerabilidades sobre los sistemas de información e incluso sobre las políticas de manejo de los datos, sin descartar también los ataques, robo de identidades o *crackeo* sobre los sistemas de información e infraestructura.

De toda esta problemática no excluye al sector público, es más, por ser un tema gubernamental puede llegar a tener un mayor interés de los piratas cibernéticos, para ello se realizó un prototipo de *blockchain* para resguardar las transacciones de científicos inscritos en la Secretaría Nacional de Ciencia y Tecnología, Senacyt, así como las transacciones de los proyectos de ciencia, tecnología e innovación que administra la misma secretaría. El *blockchain* es una tecnología que funciona como un libro mayor de transacciones y la característica que tiene es que es descentralizado o federal; es decir, no existe un ente centralizador de los datos de igual manera también lo respalda su alto nivel de criptografía a la hora de almacenar los datos o transacciones.

El prototipo de *blockchain* implementado para la Senacyt fue realizado en la nube de Amazon AWS utilizando HyperLedger como plataforma de *blockchain*, en dicha implementación, se realizó la configuración de la sub red dentro del *blockchain* mundial de Amazon AWS, de igual manera también se implementó la agregación de un nodo a la red, en este caso el nodo de la Senacyt, en el cual

se instaló el software de *HyperLedger* para interactuar con la cadena como tal, todo esto bajo las políticas de cincuenta votos más uno como consenso de una petición de unión a la cadena y veinticuatro horas para agregar un nodo a la misma.

Con la infraestructura montada a nivel de red y con el software de *HyperLedge*r instalado en el nodo, se realizó una especie de *frontend* del *blockchain*; para ello se implementó una API bajo el protocolo *RESTful*, el cual brinda un encapsulamiento de la funcionalidad del *blockchain*, posterior a la creación de la API, se le dio una salida publica para que pueda ser consumido por cualquier plataforma o sistema de información. Para esto lo que se implemento fue el complemento de Amazon AWS, Lambda, el cual facilitó bajo la arquitectura *serverless* únicamente una función en la cual esta encapsulada el API, para que sea consumida por la plataforma de Senacyt.

Al momento de implementar un *blockchain* le provee a Senacyt un nivel mayor de transparencia en sus transacciones y por ende en sus procesos que maneja, dado que el *blockchain* maneja como base la inmutabilidad de sus transacciones, así como los datos están asegurados porque uno de los principios del *blockchain* es la descentralización del resguardo de los datos, y el no depender de un órgano central de almacenamiento hace inquebrantable la alteración de una transacción.

PLANTEAMIENTO DEL PROBLEMA Y FORMULACIÓN DE PREGUNTAS ORIENTADORAS

En la medida que crecen los sistemas de información así también es el ritmo del software y técnicas de *hacking*, teniendo en cuenta que toda entidad o empresa que tenga datos puede ser vulnerable a un ataque de este tipo. En Guatemala, en el año 2011 existió una serie de ataques a varios sitios de medios de comunicación, estos ataques no fueron del tipo de denegación de servicio, sino fueron ataques en los cuales existió acceso a servidores web porque hubo modificaciones en las páginas de los servicios brindados.

En el año 2013 existió otro ataque a gran escala en el país, afectando a entidades gubernamentales y el ataque no fue sencillo porque se modificaron sitios e incluso material multimedia disponible en la web de los sitios afectados. En el año 2013 la Secretaría Nacional de Ciencia y Tecnología (Senacyt) sufrió uno de los ataques informáticos más fuertes que se recuerdan en la institución, en aquella ocasión los atacantes se filtraron a la red de servidores modificando los sistemas con los que cuenta la Senacyt, inyectándoles código malicioso en las cabeceras de los archivos del código fuente haciendo fallar a toda la funcionalidad de los sistemas.

Después de este ataque masivo, los ataques han quedado en intentos a menor escala, pero con cierta frecuencia. El riesgo que atacantes vuelvan a filtrarse a la red y logren crackear algún servidor permanece, poniendo en riesgo los sistemas de información con los que cuenta la Senacyt, sobre todo se pone en riesgo la información y la razón de ser de la Senacyt, que son los proyectos de investigación científica así como información personal y documental de los

científicos guatemaltecos o entidades que se dediquen a la ciencia, en los logs de los servidores se observa que los hackers constantemente andan en busca de puertos abiertos de distintos motores de bases de datos por lo que da la pauta a que existe el interés en los datos de la Senacyt.

Los ataques no se pueden evitar, pero sí se puede reducir el riesgo de estos, la poca capacitación y no contar con personal designado a la seguridad de bases de datos, contar con pobres estándares de desarrollo de sistemas provoca que la administración y almacenamiento sea centralizada y muy pobre, es decir, se depende de un componente centralizador para estas tareas, si este componente es vulnerado o tiene algún fallo todos los sistemas de información que dependen de los datos se verían afectados y por ende la institución. Lo ideal sería descentralizar ambas partes, la administración y el almacenamiento, muchas veces se ha logrado descentralizar el almacenamiento, pero todavía persiste la administración centralizadora cayendo al mismo problema de la centralización, por otra parte, que el almacenamiento sea distribuido no significa que los datos no puedan ser vulnerados por lo que se busca es una solución que gestione los datos de manera distribuida, permanente y segura.

La información y documentación de todos los científicos y entidades dedicadas a la ciencia es de suma importancia para el país dado que son personas que generan investigaciones científicas para incursionar a la sociedad guatemalteca y el país en general en temas de ciencia, tecnología e innovación, por lo que esta información y documentación debe ser tratada con cuidado siendo información confidencial no solo para la persona o entidad sino para el país en general.

Ante la problemática identificada la investigación busca responder las siguientes preguntas:

Pregunta central:

¿Cómo implementar un sistema de gestión de datos que sea distribuido, permanente y seguro utilizando *blockchain* como tecnología de almacenamiento?

Preguntas auxiliares:

- ¿Cómo funciona el algoritmo de consenso que se utiliza para realizar el almacenamiento de datos por medio de un *blockchain*?

- ¿Cómo funciona el proceso de consulta, inserción, modificación y eliminación de datos en un *blockchain*?

- ¿Cómo utilizar la nube como infraestructura para aprovechar eficientemente la tecnología *blockchain*?

OBJETIVOS

General

Implementar un prototipo de una plataforma de gestión de datos distribuida, permanente y segura utilizando tecnología *blockchain* para los datos de científicos y proyectos de la Secretaría Nacional de Ciencia y Tecnología, Senacyt.

Específicos

- Implementar un algoritmo de consenso para el almacenamiento en un *blockchain* de los datos de científicos y proyectos de la Senacyt.

- Diseñar y desarrollar el proceso de altas, bajas, cambios y consulta de datos almacenados en el *blockchain*.

- Implementar la plataforma de gestión de datos distribuida, permanente y segura (*blockchain*) en la nube.

MARCO METODOLÓGICO

- Tipo de investigación

El tipo de estudio es una investigación mixta teniendo como base las investigaciones iniciales y los fundamentos del *blockchain* con esto se implementará un *blockchain* privado y se adaptará para que almacene los datos y documentos de los investigadores y proyectos científicos. Con esto se logra implementar un *blockchain* privado. La investigación mixta tiene como variables cuantitativas el almacenamiento distribuido, que representa la cantidad de nodos de la cadena, también el almacenamiento seguro; esta es la cantidad de intentos para vulnerar los datos y el sitio y como única variable cualitativa, el almacenamiento permanente, que representa la característica de la permanencia de los archivos cuando se dispongan.

- Diseño de investigación

El diseño de la investigación es experimental, se analizará principalmente el resultado del experimento diseñado, se analiza el tiempo de inserción y recuperación de los datos del *blockchain* tanto como la nativamente, así como un sistema de información externo.

- Procedimiento metodológico

Para la presente investigación se cuenta un método que consta de seis fases siendo estas separadas por fase de implementación, experimentación, evaluación de resultados y presentación e interpretación de los resultados.

- Fase documental: la primera fase del estudio es la investigación documental que gira en torno a *blockchain*, la fase documental tiene como objetivo responder las preguntas:

    - ¿Qué es *blockchain*?

    - ¿Cómo se comporta *blockchain* (altas, bajas y cambios en la cadena)?

    - ¿Qué aplicaciones y usos tiene *blockchain*?

    - Posibles tecnologías para utilizar tanto en el *blockchain* como la nube.

- Fase de análisis y diseño del *blockchain*: en la fase de análisis y diseño se profundizó en la evaluación de las tecnologías para la implementación del *blockchain*; se evalúan las potenciales tecnologías a utilizar, ethereum y hyperledger; se evalúan la flexibilidad para la implementación, la documentación y los foros disponibles. De igual manera se evaluó la infraestructura en la nube que se utilizó para el despliegue de la plataforma. También se realizó la selección de los diferentes conjuntos de datos (bases de datos relacionales) y de archivos (bases de datos documentales) que se almacenarán en la cadena, estos datos deben de ser todos los que giran en torno a un investigador y sus actividades o proyectos que ha trabajado con la Senacyt.

- Fase de experimentación: en la fase de experimentación se instaló la plataforma primero de forma local y se experimenta la carga de datos relacionales y documentales al *blockchain*, y se evaluó el comportamiento

de la plataforma, comparando el rendimiento de la plataforma con respecto a la carga anterior de datos. Posteriormente se instaló la plataforma en la infraestructura en la nube. La carga de datos y documentos se realizará de manera incremental, de 10 % hasta llegar al 50 % de los datos y documentos de manera local y al 100 % en la nube. El objetivo de hacer la carga incremental es la mejora continua (depurar errores) y así poder presentar una plataforma confiable.

- Fase de implementación del *blockchain*: luego de la fase de experimentación, la plataforma está certificada para su implementación, para lo cual se realizó la preparación de la nube tal y como quedó en la última fase de la experimentación, posteriormente se realizó la carga del 100 % de los datos e investigadores.

- Fase de evaluación de resultados: en esta fase se evaluó el resultado de rendimiento de la plataforma, se evaluó lo siguiente:

    - Tiempo de respuesta en consultas
    - Tiempo de respuesta en inserciones
    - Número de peticiones concurrentes aceptadas

  Esto comparándolo contra los sistemas de información y el almacenamiento tradicional.

- Fase de evaluación de resultados: la última fase de la investigación es la interpretación del resultado de los datos, explicando el porqué de estos, esto para fundamentar las conclusiones de la investigación, así como las recomendaciones para futuras investigaciones que sigan con

la presente. También, se redacta el documento final de la investigación, en donde se plasmará todo lo que respalda a la investigación.

- Instrumentos de recolección de información

  Las técnicas que se utilizaron para este estudio son las siguientes:

  - Recolección de datos: es una fuente de datos primaria, la cual se realizó por medio de herramientas para el análisis del comportamiento de la solución, midiendo herramientas que permitan medir las tres subvariables identificadas esto por medio de pruebas de rendimiento, para luego estudiar los resultados obtenidos a raíz de esta recolección de datos.

  - Documental: es una fuente de datos secundaria, la cual se realizará bajo el estudio de libros y artículos publicados en revistas científicas.

INTRODUCCIÓN

Lo principal de todos los sistemas de información es la información en sí, por lo que compartir datos vía internet o algún sistema de información es algo común en la actualidad, pero esto da la pauta de que en el momento que exista un compartimiento de información; dicha información puede ya no ser propietaria en su totalidad dado que se concede la administración de la misma por lo que ¿se podrá lograr la total privacidad, seguridad y disponibilidad de la información en un sistema sin depender de una entidad o sistema administrador que pueda vulnerar nuestra información?

La Secretaría Nacional de Ciencia y Tecnología, Senacyt, administra toda la red científica del país, por lo que es de vital importancia que la información relacionada a temas científicos del país cumpla con ser privados, estén seguros y disponibles. *blockchain* es una tecnología que busca cumplir con estos tres pilares de la información (privado, seguro y disponible). Con la implementación de un *blockchain* se busca que la información siempre sea propiedad del investigador, que sea segura por la descentralización y por ende siempre esté disponible.

En el presente documento se expone la investigación que se llevó a cabo para elaborar un *blockchain* de tipo privado que permita el almacenamiento de los datos de tipo relacionales y documentales de los investigadores, así como sus proyectos científicos. El desarrollo de la investigación se presenta en los siguientes capítulos:

- Capítulo uno: se presentan los antecedentes de la problemática, las vulnerabilidades de contar con un sistema de información centralizador de

todas las capas de información, el riesgo permanente que existe al momento de hablar de información y el resguardo de esta.

- Capítulo dos: se presentan estudios realizados con anterioridad, casos similares que se han presentado o soluciones con la misma temática, recalcando la importancia de utilizar *blockchain* para poder solucionar la problemática identificada.

- Capítulo tres: en este capítulo se presentan los alcances identificados para el estudio, separándolos en alcance de tipo investigativo, técnico y resultados que se espera de la investigación, esto con el fin de poder delimitar la investigación, tener un marco que permita el buen desempeño de esta.

- Capítulo cuatro: se expone todo el marco teórico que engloba una solución de *blockchain*, terminología y aspectos necesarios de *blockchain*. Se tocan temas como definición de *blockchain*, arquitectura de un *blockchain*, tecnología a utilizar y arquitectura de software.

- Capítulo cinco: en este capítulo se presentan los resultados obtenidos, analizados mediante las técnicas propuestas para el análisis de datos e información, siendo estos el método inductivo y deductivo.

- Capítulo seis: en este capítulo se presenta la discusión e interpretación de los resultados para evaluar lo que pasó con la investigación, deduciendo lo que pasó con los experimentos y en qué medida fueron alcanzados los objetivos y metas de la investigación.

1. ANTECEDENTES

Una de las incertidumbres que ha creado la tecnología del internet ha sido la pérdida de información, la pérdida de información puede llegar a ser catastrófico tal y como lo plantea Keeton (2004) en su artículo *Designing for Disasters*; el autor plantea que tal y como el hardware puede ser vulnerable así también el software puede llegar a ser vulnerable ya sea por un ataque, un virus o un mal manejo que se traduce en un error y se puede llegar a plantear cuánto valen los datos; o en otro orden de palabras, cuánto puede llegar a costar el no tener un respaldo de los datos; lo mejor sería que esto no sucediera, pero lo único que se puede realizar es una prevención de desastres.

Keeton (2004) también plantea que tener una copia de los datos no es una solución al problema porque de igual manera dicha copia puede llegar a ser vulnerada, tener una política de *backups* o resguardo de los datos es de suma importancia, porque no solo es realizar una copia de los datos y que eso sea el respaldo; al contrario, se debe tener en cuenta que existen políticas de *backup* tal y como se adopta en la actualidad ya sean los patrones de diseño de software o la arquitectura de un sistema de información.

En la actual era, todo ha sido potencializado gracias al internet, reduciendo costos al momento de investigar, colaborar o compartir información, Tapscott (2017) en su libro *La revolución Blockchain* habla acerca de cómo el internet ha cambiado todos estos paradigmas, pero también dando lugar a un vacío más grande: la fiabilidad de la identidad y la información, a muchos usuarios de internet les ha sucedido que se encuentran con identidades o información falsas. Tapscott (2017) pone el ejemplo bancario: para hacer viable un intercambio

monetario entre dos personas o entidades, debe existir un regulador que pueda dar la veracidad de la transacción y centralice la información; en este caso son los bancos que cumplen esta función. La intervención de terceros para la administración de los datos la posibilidad al mal manejo de estos y el autor hace una aseveración muy acertada “La prosperidad que la tecnología crea, ya no es mayor que la intimidad que destruye” (Tapscott, 2017, p.24).

El uso del *blockchain* en la actualidad ha ido en aumento y esto por la flexibilidad que brinda esta tecnología, tal y como lo menciona Navarro (2017) en su publicación llamada *Blockchain y sus aplicaciones*, en donde se expone el siguiente enunciado: “Todo sistema en el que haya algún tipo de compartición está sujeto a que una tecnología como *blockchain* pueda aplicarse” (p.3). Esto da lugar a que prácticamente casi cualquier sistema de información se puede implementar un *blockchain* esto debido a que en la actualidad son pocos los sistemas que se crean bajo un enfoque monolítico. En dicho artículo el autor menciona la correlación que puede llegar a tener un *blockchain* con respecto situaciones cotidianas de la tecnología.

No todas las aplicaciones de un *blockchain* deben de ser criptomonedas, por ejemplo, el voto electrónico, el autor expone que haciendo sistemas de información de la manera tradicional no se puede asegurar que los datos no puede llegar a ser vulnerados o alterados al momento de un conteo, con un *blockchain* público este problema podría terminarse porque el voto se convertiría en una transacción que es parte de un *blockchain* con un conteo fiable de votos, por lo que los ciudadanos podrán hacer el conteo de votos y sería inalterable por el método de consenso que existe para realizar alguna modificación en la cadena; de igual manera, se mantendría la confidencialidad del votante con la ayuda del sistema criptográfico del *blockchain*.

Lo interpolable que puede llegar a ser un *blockchain* da oportunidad a un sinfín de posibles aplicaciones apoyándose en la descentralización y la criptografía, McConaghy (2016), en su artículo *BigchainDB: A Scalable Blockchain Database*, habla acerca de la aplicación de un *blockchain* como un manejador de bases de datos escalable, se hace ver de que el *blockchain* no fue creado para ser una base de datos distribuidas tal y como muchos piensan, por ejemplo el tiempo en el que una transacción de bitcoin se pueda realizar puede llegar a tener un tiempo muy grande a la par del tiempo en el que una transacción en una base de datos distribuida (por ejemplo, una NoSQL) pero una base de datos distribuida no tiene tres claves que el *blockchain* si tiene: control descentralizado, inmutabilidad y la creación y movimiento de activos digitales.

El *blockchain* da estas ventajas sobre un uso común de ficheros o de bases de datos distribuidas porque dichas tecnologías siempre recaen en la administración centralizada, en lo fácil que puede ser alterar los datos o lo complicado que puede ser el movimiento de datos entre nodos de un sistema distribuido clásico. Ante todo, esto el *blockchain* puede ser la respuesta ante la problemática del sistema distribuido.

Zyskind (2015) en el artículo *Decentralizing Privacy: Using Blockchain to Protect Personal Data* habla, de igual manera que Tapscott (2017), acerca de la desconfianza que genera otorgar permisos de administración de los archivos personales a terceros, similar a lo que se realizan en las aplicaciones móviles al momento de instalarlas. El autor hace referencia a la diversidad de estudios que se han realizado para mantener la confidencialidad de los datos, pero han fallado en un punto en común: la administración centralizada. Que la información esté centralizada en un punto es un eslabón de seguridad, porque siempre dependerá de un manejo centralizado ya sea de permisos o acceso a los mismos datos o archivos.

La solución al manejo centralizado para la seguridad de la información es tan simple como decir la descentralización de esta, es allí en donde entra en acción el *blockchain*; un *blockchain* certifica la descentralización de la administración de los datos. El *blockchain*, según el autor, protege contra los problemas comunes de la privacidad de la información:

- Propiedad de los datos: al usuario lo que principalmente le interesa es que él tenga la propiedad total sobre sus datos y no una entidad rectora o centralizadora de la información y él puede o no brindar permisos para acceder a sus datos.

- Transparencia y auditabilidad de datos: los usuarios deben saber qué y cuándo se realiza alguna acción sobre sus datos.

- Control de acceso de granularidad: los permisos de quiénes pueden ver o incluso editar los datos es vital, pero, sobre todo, cuándo y cómo revocar estos permisos; en aplicaciones móviles un permiso puede perdurar incluso cuando se cambia de móvil siendo este una desventaja para el usuario, por lo que tener control a un nivel minúsculo es de suma importancia para el usuario y dueño de los datos.

Blockchain cubre las necesidades de privacidad de la información que dejan muchos sistemas de resguardo de datos, descentralizando los datos y sobre todo dándole al usuario el control total sobre sus datos.

Tanto Keeton (2004) como Zyskind (2015), llegan a la misma conclusión, para que el resguardo de los datos sea eficiente se debe tener en cuenta dos cosas: las políticas de *backup* y la tecnología para emplearlas, porque se pueden tener excelentes políticos pero la tecnología no puede acoplarse o se puede tener

la mejor tecnología, pero si no se cuenta con políticas de *backup* será de balde el uso de la tecnología.

2. JUSTIFICACIÓN

El trabajo se acopla a la línea de investigación: dispositivos y sistemas para incrementar la seguridad al utilizar tecnología de la información y comunicación, proponiendo un prototipo de una plataforma de gestión de datos distribuida, permanente y segura utilizando tecnología *blockchain* para los datos de científicos y proyectos de la Secretaría Nacional de Ciencia y Tecnología, Senacyt.

Las bases de datos son vitales para cualquier sistema de información por lo que resguardar las bases de datos es de suma importancia para cualquier entidad, así mismo también es importante las técnicas y tecnologías empleadas para realizar un respaldo de los datos. En el caso de la Secretaría Nacional de Ciencia y Tecnología, Senacyt, se cuenta con la información de investigadores nacionales, entidades que se dedican a la ciencia y así como proyectos científicos que se realizan por parte de estos investigadores que se deben de resguardar.

El hecho que se realice *backup* de las distintas bases de datos relacionales y documentales no asegura que estos sean totalmente confiables y sobre todo no asegura que el lugar donde se encuentra el *backup* no pueda ser crackeado; si no se cuenta con políticas definidas y sobre todo técnicas y métodos para la realización de *backup* confiables y confidenciales, los *backup* o los mismos datos estarán expuestos ante cualquier eventualidad negativa.

Los manejadores de bases de datos ya sean relacionales o documentales generan *backup* e incluso se pueden programar y alinear a una política, pero su alcance es corto, porque no pueden ver más allá de la seguridad de los datos, el

acceso a los mismos ni la prevención de cualquier uso indebido del *backup* generado.

Generar los *backup* de bases de datos relacionales y documentales de la Senacyt no es una tarea difícil, pero no se asegura que los datos y documentos de los investigadores puedan ser accesibles o robados por medio de un acceso a un *backup*. El problema se puede mitigar con un almacenamiento distribuido de los *backup*, pero el problema no se elimina del todo por la misma razón del acceso a la ubicación del *backup* en el almacenamiento distribuido, dando lugar al problema de *backup* incompletos.

Por otra parte, en la actualidad, las criptomonedas han tenido un gran auge, una criptomoneda se basa en su confidencialidad y su alta seguridad a alteraciones, pero esto lo logra gracias al método base: el *blockchain*. *Blockchain* es una estructura de datos en la cual se puede almacenar todo tipo de datos agrupando estos datos es su unidad básica, el bloque, dichos bloques se encuentran descentralizados, con esto se logra la replicación de los datos a lo largo de la red, para acceder a los datos todos los miembros del *blockchain* (los bloques) deben de estar ‘de acuerdo’, lo que se le conoce como consenso, si uno de los bloques responde de manera negativa al consenso no se brinda acceso a los datos que están encriptados, esto convierte al *blockchain* un sistema sólido a nivel de confianza y confidencialidad.

El uso de un *blockchain* no se limita solamente a criptomonedas, todo lo contrario, en un bloque se puede almacenar todo tipo de datos, dando lugar a un abanico de aplicaciones de un *blockchain*. Utilizar un *blockchain* como método y técnica para almacenar bases de datos relacionales y documentales de la Senacyt es la mejor opción, porque se aprovecha los recursos distribuidos con

los que se cuenta y se asegura la confidencialidad de los datos al contar con el método de consenso, reduciendo el riesgo de contar con *backup* incompletos.

Con el uso de un *blockchain* como sistema de gestión de bases de datos relacionales y documentales la Senacyt, se tiene como beneficio el resguardo de manera segura los datos y documentos de los investigadores nacionales, las entidades que se dedican a la ciencia en el país y de los proyectos científicos de los investigadores.

3. ALCANCES

3.1. Resultados

- Algoritmo de consenso criptográfico que asegure los datos y el funcionamiento del *blockchain*.

- Aplicaciones apegadas a los procesos de altas, bajas, cambios y consulta de datos, almacenarlos en el *blockchain*.

- Prototipo de *blockchain* privado que almacene datos transaccionales y documentales de los investigadores y proyectos de los sistemas de información de la Secretaría Nacional de Ciencia y Tecnología; dicho prototipo contará con una infraestructura de del prototipo implementada en la nube.

3.2. Técnicos

El desarrollo de un prototipo de *blockchain* se realizará utilizando Hyperledger, una plataforma de código abierto para realizar *blockchain* de tipo privados, con lo cual posteriormente se vinculó a un sistema transaccional con toda su infraestructura en la nube.

- Implementar un algoritmo de consenso y de criptografía para el manejo de la seguridad de la plataforma.

- Conectar los sistemas de información de la Senacyt con el *blockchain* implementando los procesos de altas, bajas, cambios y consulta de datos.

- Implementar Hyperledger Composer y Hyperledger Factory para el *blockchain* privado utilizando contenedores de Docker e infraestructura en el cloud.

3.3. Investigativos

Con el desarrollo del prototipo de una plataforma de gestión de datos distribuida, permanente y segura por medio de un *blockchain* privado para el resguardo de los datos de científicos y proyectos de la Senacyt, se definen los siguientes alcances investigativos:

- Investigar y comprender los tipos de algoritmos de pruebas de trabajo y consenso que existen y son más utilizados.

- Investigar sobre el funcionamiento completo, como es el proceso de altas, bajas, cambios y consultas de datos de un *blockchain* privado.

- Comprender e investigar toda la lógica del funcionamiento de *blockchain*, la distribución, almacenamiento y seguridad de los datos.

4. MARCO TEÓRICO

4.1. *Blockchain*

Es una estructura de datos la cual tiene forma de una lista enlazada que utiliza punteros hash en lugar de punteros normales como se utilizarían en listas normales, los punteros hash sirven para enlazar al bloque anterior. *Blockchain* funciona como un libro mayor de transacciones, dichas transacciones se agrupan en bloques, en la actualidad cada organización maneja su fuente de datos, mientras que en *blockchain* el poder distribuido tendría la única fuente verdadera de datos (Bashir, 2017).

4.1.1. Tipos de *blockchain*

Esta tecnología puede ser aplicable a cualquier tipo de ámbito y cualquier naturaleza por lo que se puede tener *blockchain*s públicos o privados, dependiendo del tipo de acceso a los datos (Navarro, 2017).

4.1.1.1. Público

Un *blockchain* público es aquel que no tiene ninguna restricción para leer los datos del *blockchain* y de igual manera es fácil entrar y salir del *blockchain*, siendo las transacciones transparentes. Un ejemplo de ellas son las criptomonedas como Bitcoin o Ethereum.

4.1.1.2. Privado

Un *blockchain* privado tiene la característica que solo el propietario posee permisos para acceder a los datos o registrar transacciones.

4.1.2. Principios de *blockchain*

Al ser una tecnología, debe contar con principios, patrones o características básicas y mínimas para que se pueda asegurar un funcionamiento mínimo, algunos patrones básicos según Tapscott, (2017):

4.1.2.1. Integridad

La integridad está a lo largo de la cadena y nodos que forman parte del *blockchain* esto quiere decir que no depende de un componente aislado, la clave de esto es que toda la integridad está cifrada a lo largo de la cadena. El interés que existe en la información que hay dentro de la cadena hace que exista una confianza de integridad entre los nodos de la cadena.

4.1.2.2. Poder distribuido

El poder está distribuido a todos los nodos de la cadena, no existe un poder centralizador, *blockchain* es público para todos los nodos, por lo que si existe algún movimiento extraño dentro de la cadena todos lo verían dejando el poder de decisión a todos los nodos miembros de la cadena, lo que en *blockchain* se conoce como consenso.

4.1.2.3. Seguridad

La seguridad de un *blockchain* está centrada en el uso de criptografía, es decir, todos los nodos miembros deben de utilizar criptografía para formar parte de un *blockchain*, caso contrario no podrán formar parte de uno, no existe puntos intermedios ni excepciones.

4.1.2.4. Privacidad

Los datos de una persona o entidad dentro de un *blockchain* son únicamente de las personas, no más, no menos, respetando la privacidad de estos, no existe un sistema centralizador de los datos el cual puede llegar a tener cierta potestad sobre los mismos.

4.1.2.5. Derechos preservados

Los miembros de un *blockchain* tienen derechos y estos derechos se respetan, tal es el caso del derecho a privacidad, es decir que en mundo digital del *blockchain* se respetan los derechos.

4.1.3. Componentes básicos de un *blockchain*

Blockchain cuenta con varios elementos, pero se pueden mencionar tres elementos que son básicos para que un *blockchain* sea funcional y confiable (Navarro, 2017).

4.1.3.1. Bloques

Toda la información que entra a un *blockchain* es almacenada en un bloque, y este bloque añadido a la cadena, el bloque es la unidad básica de almacenamiento del *blockchain*. Cada bloque debe tener como mínimo la siguiente información:

- Código que enlaza con el bloque anterior
- La información como tal que se almacena
- Código que enlaza con el bloque siguiente

4.1.3.2. Mineros

Un minero es una máquina que ayuda en el proceso de validación de transacciones que se producen, autorizando la agregación de nuevos bloques. Un minero sigue los siguientes pasos:

- Los mineros reciben las nuevas transacciones.
- Los mineros recogen la nueva transacción y la almacenan en un bloque.
- Cada minero busca una prueba de trabajo para el bloque.
- Cuando termina de encontrar la prueba de trabajo transfiere el bloque a todos los nodos.
- Cada nodo acepta la transacción solo si las transacciones son válidas.
- Los nodos aceptan el bloque creando el bloque siguiente colocando el hash anterior en el nuevo hash.

4.1.3.3. Nodos

El nodo es una máquina conectada a una red, cuenta con un software que permite el almacenamiento y la distribución de una copia real del *blockchain*. Se distribuye un nodo cuando es validado y este se añade a la cadena y cada nodo almacena el cambio

4.1.4. Ventajas de *blockchain*

Utilizar *blockchain* da muchas ventajas sobre un modelo tradicional de almacenamiento transaccional, siendo estas las siguientes (Gupta, 2017):

4.1.4.1. Libro contable

Blockchain trabaja en base a los principios de un libro de contabilidad, solamente que es distribuido y compartido, es decir, un registro inmutable del libro está en toda la red y todos los participantes de la red pueden acceder. El libro contable cuenta con las siguientes características:

- Registra todas las transacciones en la red, la única fuente de verdad.
- Se comparte entre todos los participantes en la red a través de la replicación, cada participante tiene una copia duplicada del libro contable.
- Los participantes solo ven las transacciones que están autorizados a ver.
- Los participantes tienen identidades que los vinculan con las transacciones, pero pueden elegir la información de la transacción que otros participantes están autorizados a ver (Gupta, 2017).

4.1.4.2. Permisos

En un *blockchain*, cada participante tiene una identidad única, que permite el uso de políticas para restringir la participación en la red y el acceso a las

transacciones. Con la capacidad de restringir el acceso a la red también son más efectivas para controlar la consistencia de los datos que se adjuntan al *blockchain*.

Con la capacidad de restringir el acceso a los detalles de la transacción, se pueden almacenar más detalles de la transacción en el *blockchain*, y los participantes pueden especificar la información de la transacción que están dispuestos a permitir que otros vean. Con un *blockchain* pública, el nivel de detalle de la transacción puede estar limitado para proteger la confidencialidad y el anonimato (Gupta, 2017).

4.1.4.3. Consenso

En una red de bloques donde los participantes son conocidos y confiables, las transacciones pueden verificarse y comprometerse con el libro de contabilidad a través de varios medios de acuerdos (consenso), cuyas características clave son las siguientes:

- Prueba de participación: valida las transacciones, los validadores deben tener un cierto porcentaje del valor total de la red. La prueba de participación podría proporcionar una mayor protección contra cualquier tipo de ataque que pueda sufrir la red.

- Firma múltiple: muchos validadores (por ejemplo, la mitad más uno) deben aceptar que una transacción es válida.

- Tolerancia de falla bizantina práctica (PBFT, por sus siglas en inglés): es un algoritmo diseñado para resolver disputas entre los nodos (en este caso participantes de la red) cuando un nodo en un conjunto de nodos (en este

caso blokchain) genera una salida diferente de los otros en el conjunto (Gupta, 2017).

4.1.4.4. Contratos inteligentes

Un contrato inteligente es un acuerdo o conjunto de reglas que rigen una transacción; un contrato inteligente se almacena en el *blockchain* y se ejecuta automáticamente como parte de una transacción. Los contratos inteligentes pueden tener muchas cláusulas contractuales que podrían ser parcial o totalmente autoejecutables. Su propósito es proporcionar una seguridad superior a la ley de contratos tradicional al mismo tiempo que reduce los costos y retrasos asociados con los contratos tradicionales (Gupta, 2017).

4.1.4.5. Roles y participantes

Blockchain tiene diversos participantes y tipos de roles, los cuales, según Gupta (2017), son los siguientes:

- Usuario *blockchain*: un participante con permisos para unirse a la red de *blockchain* y realizar transacciones con otros participantes de la misma red. La tecnología *Blockchain* opera en segundo plano, por lo que el usuario de *blockchain* no tiene conocimiento de ello. En una red de *blockchain* existen muchos usuarios de este tipo (Gupta, 2017).

- Regulador: un usuario de *blockchain* con permisos especiales para supervisar las transacciones que ocurren dentro de la red. Hay que tener en cuenta que los reguladores pueden tener prohibido realizar transacciones, es decir, no cuentan con permisos de alteración sobre la cadena (Gupta, 2017).

- Operador de red de *blockchain*: personas que tienen permisos especiales y autoridad para definir, crear, administrar y monitorear la red de *blockchain*. Cada entidad que forma parte de la red de *blockchain* debe tener un operador de red de *blockchain* (Gupta, 2017).

- Plataformas de procesamiento tradicionales: sistemas de información existentes que pueden ser utilizados por el *blockchain* para aumentar el poder de procesamiento. Este sistema también puede interactuar vía solicitudes en el *blockchain* (Gupta, 2017).

- Fuentes de datos tradicionales: los sistemas de datos existentes pueden proporcionar datos para influir en el comportamiento de los contratos inteligentes y ayudar a definir cómo se producirán las comunicaciones y la transferencia de datos entre las aplicaciones y datos tradicionales con el *blockchain* (Gupta, 2017).

- Autoridad de certificación: una persona o entidad que emite y administra los diferentes tipos de certificados requeridos para ejecutar un *blockchain* con permisos (Gupta, 2017).

4.2. Arquitectura de microservicios

Los microservicios es una arquitectura que está teniendo auge en los últimos años debido a su nivel de desacoplamiento que maneja.

4.2.1. ¿Qué es una aplicación descentralizada y una distribuida?

Antes de comprender la definición de una arquitectura de microservicios hay que tener en cuenta qué es una aplicación descentralizada y una distribuida. Una

aplicación distribuida es aquella que extiende a través de una red de múltiples nodos ordenados en vez de tener uno solo. Una aplicación descentralizada es aquella que carece de un orden en específico y ningún nodo tiene la potestad de ordenar a otro nodo, básicamente con este principio se acelera el poder de procesamiento y reducir la latencia de datos (Raval, 2016).

Ante esto, se plantea la siguiente pregunta: ¿se puede tener una aplicación descentralizada y a la vez distribuida? Sí se puede, un claro ejemplo de este tipo de aplicaciones son las criptomonedas como Bitcoin o Ethereum, dado que su registro de transacciones (*blockchain*) reside en varias computadoras siendo esto distribuido, pero, si uno de los nodos falla la red funciona todavía por lo que también es descentralizado (Raval, 2016).

4.2.2. Definición

Esta arquitectura reduce en su mínima expresión las funcionalidades del software para que esas mínimas expresiones se conviertan en servicios o, mejor dicho, microservicios (Nadareishvili, 2016).

Otra definición indica que los microservicios son pequeños servicios autónomos, que trabajan juntos entre sí para lograr un objetivo en común (Newman, 2015).

4.2.3. Características y modelado de una arquitectura de microservicios

Existen diferentes factores que se deben considerar al construir una arquitectura de microservicios; estos se basan en los siguientes principios (Newman, 2015):

- Bajo acoplamiento: cuando un servicio está débilmente acoplado, un cambio en dicho servicio no debe repercutir en el funcionamiento de los demás servicios.

- Alta cohesión: lo que se busca con una alta cohesión es que las funcionalidades relacionadas estén juntas y las que no estén relacionadas se encuentren separadas, definiendo límites entre funcionalidades, pero sin olvidar la comunicación entre ambos límites.

- Contexto limitado: un contexto limitado es cuando un servicio se dedica a una sola funcionalidad en específico.

- Modelos compartidos y ocultos: cuando se ha contextualizado o separado según funcionalidad, hay que tener en cuenta que debe existir una comunicación entre servicios o conjunto de servicios, pero, no todos los servicios y funcionalidades estarán disponibles existen ciertas funcionalidades que son de carácter interno y sirve para la funcionalidad del servicio.

- Descomposición prematura: la descomposición de un sistema o servicio es aquella propiedad que separa en funcionalidades más pequeñas, pero la descomposición prematura es pensar que teniendo un sistema actual solo será de pasar a servicios y esta idea puede llegar a ser errónea.

- Capacidades empresariales: al momento de hablar de microservicios hay que tener en cuenta que la empresa debe tener la capacidad de hasta reinventar su modelo de datos para que se pueda adecuar a la arquitectura de microservicios, bajo el principio del bajo acoplamiento.

- La comunicación en términos de conceptos de negocio: cualquier arquitectura que se implemente debe estar amarrada a la lógica de

negocio y más los microservicios que son totalmente independientes, la comunicación entre tecnología y lógica de negocio debe ser clara.

- El límite técnico: hay que tener en cuenta que para utilizar una arquitectura de microservicios se debe considerar la infraestructura, personal, tecnologías y todo lo que engloba un proyecto de software porque al ser un paradigma relativamente nuevo de arquitectura hay que tener en cuenta todo lo que conlleva a nivel técnico.

Con base en los principios anteriormente mencionados se pueden plantear características que deben de tener los servicios parte formar parte de una arquitectura de microservicios Nadareishvili, (2016)

- Totalmente autónomos o independientes: los microservicios no dependen de algún otro componente y mantienen su propio conjunto de datos.

- Tienen un alcance funcional pequeño y limitado: el alcance de un microservicio no debe ser grande, una pequeña funcionalidad a la vez.

- No requieren participación en transacciones distribuidas: al momento de ser autónomos y de tener su propio conjunto de datos, no pueden participar en transacciones que sean interpolables entre microservicios.

- Giran en torno a las capacidades del negocio o equipos de trabajo: es decir que los microservicios a diseñar estarán divididos entre los cuatro principios del *blockchain* y de esa manera también se puede dividir un equipo de trabajo.

- No dependen de invocaciones síncronas: los microservicios son asíncronos, porque no dependen de algún otro componente, con esto se da lugar a que los servicios sean de tipo REST.

- No son parte de composiciones complejas: el ecosistema de un microservicio no es un entorno complejo, por su segmentación (característica 4) no pueden ser parte de un ecosistema complejo.

4.2.4. Estilo coreográfico

Lo que se necesita es la descentralización de la funcionalidad, por lo que no se puede tener un estilo orquestado porque se necesitaría de un componente centralizador, por esta razón el estilo coreográfico encaja a la perfección, cada microservicio sabrá cuándo y cómo actuar en las diferentes situaciones a las que estará expuesto (Nadareishvili, 2016).

4.3. Tecnología aplicada

Para implementar un *blockchain* privado se debe tener en cuenta que la tecnología a utilizar debe acoplarse a la arquitectura, descentralización y distribución.

4.3.1. *Blockchain*: HyperLedger

Hyperledger es un proyecto que cuenta con más de 50 participantes (empresas, universidades y centros de investigación) y su principal objetivo es tener una herramienta Open Source para soluciones que utilizan *blockchain* (Cachin, 2016).

4.3.1.1. HyperLedger Fabric

Hyperledger Fabric es una implementación de un *blockchain* de código abierto, que permite la creación de contratos inteligentes, utilizando tecnologías

conocidas y comprobadas, con una arquitectura modular que permite el moldeo a la medida según sea la necesidad con la que se tenga (Cachin, 2016).

Hyperledger Fabric trabaja con un protocolo que es ejecutado por *peers*, de los cuales tiene dos tipos: el primer peer es el encargado de la validación en la red, de la operación de consenso dentro del *blockchain*, de validar las transacciones y mantener los registros de transacciones y por otro lado el segundo peer que no verifica ninguna transacción, sino tiene la función de proxy dentro de la cadena, validando conectividad entre nodos (Cachin, 2016).

Hyperledger se creó para ayudar a avanzar en las tecnologías del *blockchain*. Es una colaboración global de código abierto que involucra a líderes de numerosas industrias (Gaur, 2018).

Algunas funcionalidades de Hyperledger Fabric son las siguientes (Cachin, 2016):

- Implementa contratos inteligentes.
    - Código de la cadena definida por el usuario es encapsulada en un contender de Docker.
    - El código de la cadena se ejecuta paralelamente que el peer.
- Utiliza un algoritmo de consenso basado en la tolerancia bizantina.
- Soporte a seguridad por medio de certificados.

- Almacenamiento persistente por medio de una base de datos clave-valor de RocksDB.

- SDK para el cliente (Node.JS) para interactuar directamente con Fabric.

- Soporte para API REST y CLIs.

4.3.1.2. HyperLedger Fabric: arquitectura de referencia

Hyperledger sigue una arquitectura modular, teniendo en cuenta los siguientes módulos de la arquitectura:

- Módulo de servicios de membresía: este módulo es esencialmente un módulo de permisos y actúa como un vínculo para establecer un enlace de confianza durante la creación de la red de *blockchain*, pero esto también es fundamental para asegurar y administrar la identidad de los miembros de la misma red. Los servicios de membresía son esencialmente una autoridad de certificación, así como elementos utilizados de la infraestructura de clave pública para cosas como la distribución de claves, la administración y el establecimiento de la confianza federada a medida que la red crece (Gaur, 2018).

- Módulo de transacciones: una transacción es una solicitud al *blockchain* para ejecutar una función en el libro contable. La función es implementada por un código de cadena denominado chaincode. La criptografía garantiza la integridad de las transacciones al vincular la transacción con los bloques anteriores y garantizar la integridad transaccional, si está protegida, al

vincular el criptograma o hash de los bloques previamente vinculados (Gaur, 2018).

- Contratos inteligentes: el código de cadena o chaincode es un código de nivel de aplicación almacenado en el libro contable como parte de una transacción. Chaincode ejecuta transacciones que pueden modificar el estado global de la cadena. La lógica de transacción se escribe como código de cadena (en los lenguajes Go o JavaScript) y se ejecuta en contenedores seguros de Docker. La transacción transforma los datos, abarcados por el código de cadena en el canal desde el que opera (Gaur, 2018).

4.3.2. Contenedores: Docker

Docker es una plataforma que permite el despliegue de aplicaciones en contenedores proporcionando el uso óptimo de los recursos, porque utilizará solamente lo necesario (Preeth, 2015).

4.3.3. Comunicación entre microservicios: REST

Rest es un estilo arquitectónico inspirado en la Web, respetando los principios de la arquitectura de microservicios, principios tales como la independencia o la autonomía, el protocolo de comunicación debe ser REST, porque es asíncrono, no depende de una llamada a algún servicio o similar (Newman, 2015).

4.3.4. Infraestructura: Cloud Computing

El Cloud Computing se puede definir como un método que proporciona una serie de recursos informáticos compartidos que incluye entre ellas plataformas para aplicaciones, almacenamiento, redes, desarrollo y despliegue. Pero no se limita a eso únicamente, también se tienen componentes como procesos empresariales. El Cloud Computing convierte los activos informáticos en silos tradicionales en grupos de recursos compartidos que se basan en una base de Internet subyacente (Hurwitz, 2012).

4.3.4.1. Características del Cloud Computing

El Instituto Nacional de Estándares y Tecnología (NIST, por sus siglas en inglés) define cinco características clave del Cloud Computing (Rountree, 2014):

- On-Demand Self-Service: esta característica habla acerca de la independencia de un administrador de parte del proveedor, el cual se encargue de prestar el servicio, en el Cloud, si es necesario un servicio nuevo el mismo usuario de la nube lo puede solicitar al instante, sin tener un servicio manual por parte del proveedor.

- Broad Network Access: esta característica es acerca del acceso sencillo hacía la nube esto a raíz de que lo único a lo que se necesita para acceder a la nube es una conexión de red básica, por lo regular este acceso se realiza vía internet, hay que tener en cuenta que las conexiones internas son mucho más rápidas que una conexión a internet, por lo que los proveedores de Cloud se han adaptado a esto.

- Resource Pooling: esta característica dicta que los recursos que se tengan disponibles son los necesarios, es decir, no se tendrá la misma cantidad de recursos siempre, porque no siempre se estará ocupando y así estos recursos pueden ser utilizados por otro cliente que si necesite los recursos.

- Rapid Elasticity: esta característica es acerca de la habilidad de poner a disposición los recursos necesarios según sea la necesidad o demanda, inmediatamente, es decir, si se requiere más recursos para atender una grande manda, el Cloud lo provee de manera inmediata.

- Measured Service: bajo la premisa de 'lo que es medible es mejorable', esta característica indica de las diferentes medidas que se pueden tener en el Cloud, por ejemplo, tiempo de uso, banda de ancha datos usados, entre otros.

4.3.5. Amazon AWS

Amazon es un repertorio de servicios de cualquier índole, es decir, servicios que van desde infraestructura hasta funciones, Amazon AWS, dependiendo del producto, así es como obtiene sus ganancias, ya sea por ancho de banda utilizado, peticiones realizadas o espacio utilizado. Lo servicios de Amazon AWS se comportan de manera independiente, es decir, que se pueden utilizar de manera independiente uno del otro (Shao, 2012).

4.3.6. Amazon AWS y HyperLedger Fabric: Amazon Managed Blockchain

Amazon tiene una solución o servicio que permite la creación de redes *blockchain* y es denominada Amazon Managed Blockchain, esta solución crea y a administra los componentes necesarios para montar una solución *blockchain*,

siendo estos, la red como tal, miembros, nodos, bades de datos descentralizadas y todo esto encapsulado, para que a la hora de implementar la solución el desarrollador o el encargado de implementar la red no tenga que ver componentes muy técnicos haciendo así, la curva de aprendizaje más pequeña.

5. PRESENTACIÓN DE RESULTADOS

5.1. Análisis y diseño del *blockchain*

Para comprender hacer más sencilla la implementación de la solución se tomó dos perspectivas, la arquitectura del *blockchain* y la arquitectura de la integración final

5.1.1. Arquitectura final de *blockchain*

La arquitectura del *blockchain* se ve subdividida en temas de red y una arquitectura de funcionalidad, las cuales son las siguientes:

5.1.1.1. Red

La arquitectura del *blockchain* a nivel de red es donde se ubica todo el *core* de la funcionalidad del *blockchain*, porque en dicha arquitectura van los siguientes componentes:

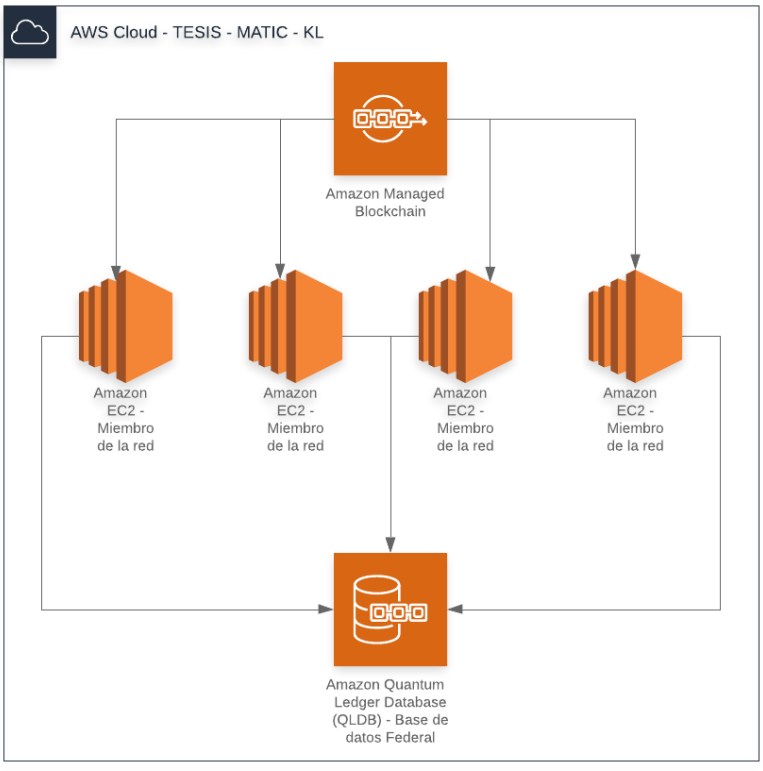
- Amazon Managed *Blockchain*: es el nodo central, orquestador y miembro de la red principal de HyperLedger de Amazon, es el miembro representante dentro de la red y forma parte del consenso para que exista una inserción dentro de la cadena.

- Amazon EC2: son los nodos miembros de la red interna que se crea, se crea un nodo por miembro, para este prototipo se ha creado un solo miembro, si en dado caso se unen más entidades gubernamentales a la

red, se debe crear una instancia Amazon EC2. Cada instancia de Amazon EC2 de preferencia debe tener un Ubuntu instalado, dependiendo de la transaccionalidad que va a manejar el servicio, para nuestro caso es la instancia básica de Amazon EC2.

- *Amazon Quantum Ledger Database*: este elemento va implícito en el nodo de Amazon Managed *Blockchain*, pero es importante mencionarlo, es la base de datos de tipo federal en la cual se almacenan las transacciones que se realizan en la cadena.

Figura 1. **Arquitectura del *blockchain***



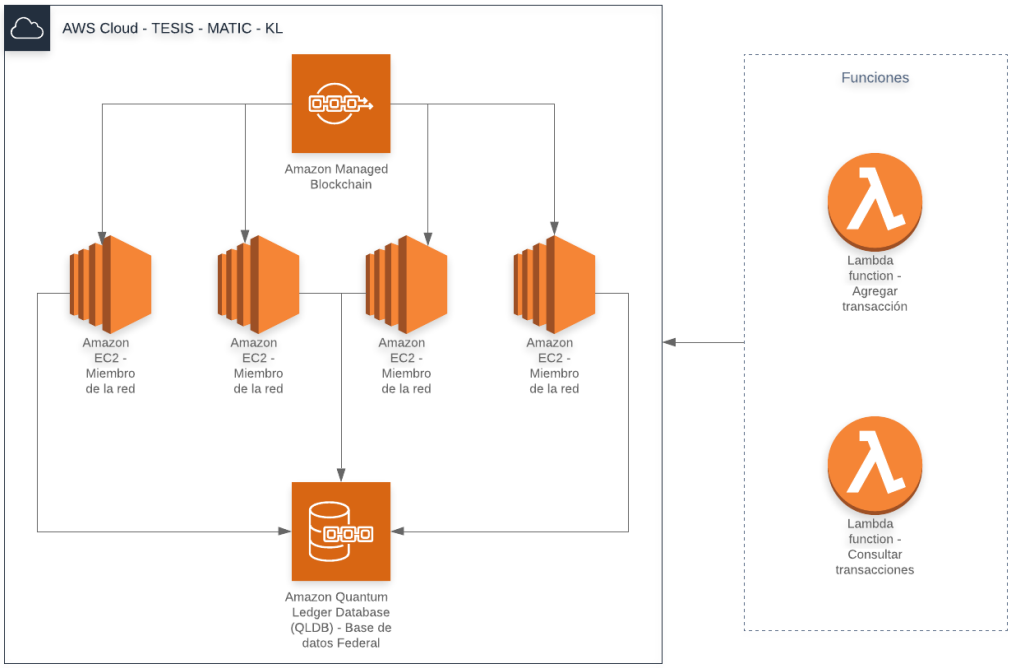
Fuente: elaboración propia, utilizando el programa Lucidchart.

5.1.1.2. Lambda

Los clientes como tal no tienen acceso directo a los nodos miembros de red interna y mucho menos al miembro general de la red mundial, sino lo que se realiza es darle entrada mediante una capa intermedia, en este caso la capa a utilizar la extensión de Amazon denominada Lambda.

En el caso de Lambda, contendrá tres funciones: crear una transacción, consultar una transacción y consultar todas las transacciones, las únicas operaciones que se pueden realizar en un *blockchain*.

Figura 2. **Arquitectura del *blockchain* con funciones Lambda**

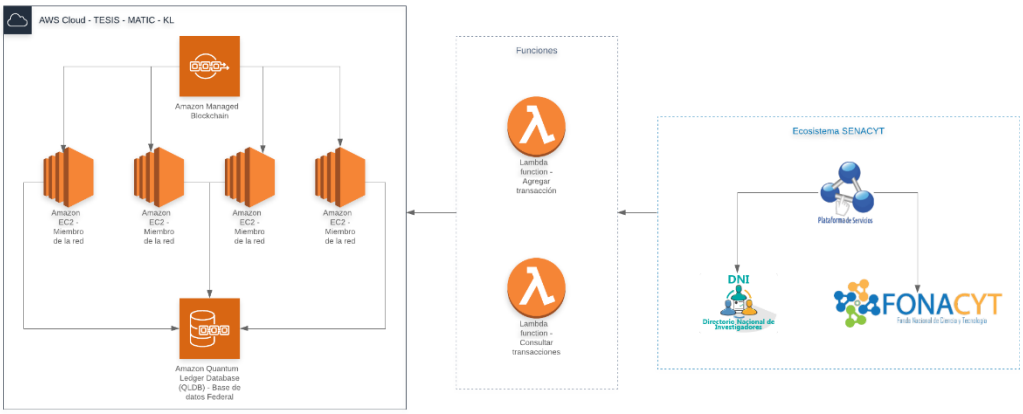


Fuente: elaboración propia, utilizando el programa Lucidchart.

5.1.2. Arquitectura integrada del *blockchain* y la plataforma Senacyt

Al final, la arquitectura que se desarrolló fue un ecosistema en el que la puerta de enlace es las funciones Lambda, es decir, la plataforma de Senacyt, plataforma de servicios en línea; se conecta directamente a las funciones Lambda, sin interactuar con los nodos directamente, tal y como se ve en la figura 3. En la que se observa que la puerta de entrada al *blockchain* es las funciones Lambda, cabe mencionar que será por medio del protocolo RESTful, explicado más adelante.

Figura 3. **Arquitectura del final con integración a Senacyt**



Fuente: elaboración propia, utilizando el programa Lucidchart.

5.2. Implementación del *blockchain*

La plataforma se implementó desde Amazon AWS en su servicio Managed Blokchain, en la cual se configuró toda la funcionalidad del *blockchain*. Como prerrequisito se debió configurar una instancia de Amazon AWS Cloud9 el cual

es un entorno de desarrollo especializado para todos los servicios de Amazon para ello se seleccionó un tipo *t2.small*.

Posterior a eso se procede a la clonación de las plantillas de Amazon que sirvieron para crear el ambiente de desarrollo de Cloud9.

cd ~
git clone https://github.com/aws-samples/non-profit-blockchain.git

Posterior se actualizó el ambiente de Cloud9.

sudo pip install awscli --upgrade

5.2.1. Implementación de la red

Posterior a ello se realizó la configuración de la red, para ello se configuró la región del *blockchain* de la siguiente manera:

export REGION=us-east-1
export STACKNAME=non-profit-amb
cd ~/non-profit-blockchain/ngo-fabric
./amb.sh

Luego de realizar la región de la cadena se configura el nodo parte de la cadena o red, para ello se realiza el nodo dentro de la consola de Amazon Managed Blokchain. Luego se ejecutó el script para crearle la llave publica para ingresar al nodo recién creado, de la siguiente manera:

export REGION=us-east-1

cd ~/non-profit-blockchain/ngo-fabric
./vpc-client-node.sh

Cuando se realizó la clave publica se procedió a configurar el nodo creado anteriormente, se inicia sesión en el nodo creado por vía SSH y con la llave creada anteriormente. En este punto lo que se realizó fue la clonación del repositorio de plantilla tal y como se realizó en la tabla 1.

Se realizó la configuración del nodo, exportando todas las variables necesarias para el funcionamiento del nodo.

export REGION=us-east-1
cd ~/non-profit-blockchain/ngo-fabric
cp templates/exports-template.sh fabric-exports.sh
source fabric-exports.sh
source ~/peer-exports.sh

Se actualiza la última versión de Managed *Blockchain* en el nodo recientemente configurado:

aws s3 cp s3://us-east-1.managedblockchain/etc/managedblockchain-tls-chain.pem /home/ec2-user/managedblockchain-tls-chain.pem

Posterior a ello se debe enrolar a la entidad criticadora, en este caso, es la red recién creada:

export PATH=$PATH:/home/ec2-user/go/src/github.com/hyperledger/fabric-ca/bin
cd ~

fabric-ca-client enroll -u https://$ADMINUSER:$ADMINPWD@$CASERVICEENDPOINT --tls.certfiles /home/ec2-user/managedblockchain-tls-chain.pem -M /home/ec2-user/admin-msp

Por último, se copiaron los certificados de acceso y membresía del nodo:

mkdir -p /home/ec2-user/admin-msp/admincerts
cp ~/admin-msp/signcerts/* ~/admin-msp/admincerts/
cd ~/non-profit-blockchain/ngo-fabric

En este punto ya estaba configurado el nodo, y cabe destacar que por cada miembro que se quiera que forme parte del *blockchain*, se debe realizar la anterior configuración por nodo, como se mencionó, si hay otras entidades públicas que quieran formar parte se deben de configurar de esta forma.

5.2.2. Configuración de los canales de comunicación

En este punto, el nodo creado ya tiene instalado todo el software necesario que se necesita para poder votar en el consenso dentro de la red internacional de Amazon Managend Blockchain.

Para ello se realizó la configuración de *configtx* que es el servicio web de HyperLedger Fabric, se inició actualizando los repositorios de *configtx.*

cp ~/non-profit-blockchain/ngo-fabric/configtx.yaml ~
sed -i "s|__MEMBERID__|$MEMBERID|g" ~/configtx.yaml

Posterior a ello se precedió a correr el contenedor con el servicio de *confitx*, esto es para poder crear y ejecutar el bloque génesis o bloque número cero, de la siguiente manera:

docker exec cli configtxgen -outputCreateChannelTx /opt/home/$CHANNEL.pb -profile

Posteriormente se tuvo que crear el canal de HyperLedger Fabric:

docker exec -e "CORE_PEER_TLS_ENABLED=true" -e "CORE_PEER_TLS_ROOTCERT_FILE=/opt/home/managedblockchain-tls-chain.pem" \ -e "CORE_PEER_ADDRESS=$PEER" -e "CORE_PEER_LOCALMSPID=$MSP" -e "CORE_PEER_MSPCONFIGPATH=$MSP_PATH" \ cli peer channel create -c $CHANNEL -f /opt/home/$CHANNEL.pb -o $ORDERER --cafile $CAFILE --tls --timeout 900s

Luego, se creó el *chaincode* en el nodo, el *chaincode* permite la interacción con el *blockchain* y la inserción de transacciones dentro de la cadena.

docker exec -e "CORE_PEER_TLS_ENABLED=true" -e "CORE_PEER_TLS_ROOTCERT_FILE=/opt/home/managedblockchain-tls-chain.pem" \ -e "CORE_PEER_ADDRESS=$PEER" -e "CORE_PEER_LOCALMSPID=$MSP" -e "CORE_PEER_MSPCONFIGPATH=$MSP_PATH" \ cli peer chaincode install -n $CHAINCODENAME -v $CHAINCODEVERSION -p $CHAINCODEDIR

Por último, ejecutar todo lo necesario para que el servicio funcione con el siguiente comando:

docker exec -e "CORE_PEER_TLS_ENABLED=true" -e "CORE_PEER_TLS_ROOTCERT_FILE=/opt/home/managedblockchain-tls-chain.pem" \ -e "CORE_PEER_ADDRESS=$PEER" -e "CORE_PEER_LOCALMSPID=$MSP" -e "CORE_PEER_MSPCONFIGPATH=$MSP_PATH" \ cli peer chaincode install -n $CHAINCODENAME -v $CHAINCODEVERSION -p $CHAINCODEDIR

5.2.3. Creación del *frontend* del *blockchain* por medio de una API *RESTful*

Luego se creó el *frontend* o darle salida publica se debió de construir una API REST, en la cual están los métodos para interactuar directamente con el *blockchain*.

Para ello se instaló en el nodo creado anteriormente, miembro del *blockchain*, todo lo necesario para implementar las API REST, para ello se procedió a instalar gcc, g++ y Node.js.

Se procedió generar el perfil de conexión en el nodo, se necesita un perfil para poder conectarse, es decir un usuario de conexión en la cual será el que proporcionará la información a la aplicación dónde estará montada el API REST, para ello se generó de la siguiente manera:

cd ~/non-profit-blockchain/ngo-rest-api/connection-profile
./gen-connection-profile.sh
cd ~/non-profit-blockchain/tmp/connection-profile/
cat ngo-connection-profile.yaml

En este punto ya estaba registrado la conexión, lo único que queda es iniciar el servicio de *blockchain*, y las APIs:

cd ~/non-profit-blockchain/ngo-rest-api
nvm use lts/carbon
node app.js

Con el servicio corriendo ya se podía realizar experimentos, dado que para realizar experimentos o pruebas no es necesario que tenga salida publica o estable las funciones.

5.2.4. Integración de Lambda

El último paso para poder implementar el *blockchain* es darle salida pública bajo el concepto de *serverless* es decir, acceder a las funciones o APIs sin la necesidad de tener que acceder al servidor, para ello se utiliza Amazon Lambda.

Lo que se buscaba en este punto es implementar la arquitectura de la figura 3. en la cual ya existe la integración de con lambda sin la necesidad de ingresar directamente al servidor o nodo de la cadena, para ello se realizó la creación del usuario pero que utilizará lambda, que es muy diferente al usuario que utiliza la conexión (usuario creado con anterioridad).

export FABRICUSER=userSENACYT
export FABRICUSERPASSWORD=passSENACYT
~/non-profit-blockchain/ngo-lambda/createFabricUser.sh

Posteriormente, se implementa como tal la salida de lambda, bajo el protocolo REST, para ello se ejecutaron los siguientes comandos:

export BUCKETNAME=`echo "ngo-fabric-lambda-$(date +%N)" | tr '[:upper:]' '[:lower:]'`
export LAMBDANAME=`echo "$NETWORKNAME-fabric-lambda" | tr '[:upper:]' '[:lower:]'`
~/non-profit-blockchain/ngo-lambda/createLambda.sh

El resultado de toda esta implementación fue lo siguiente:

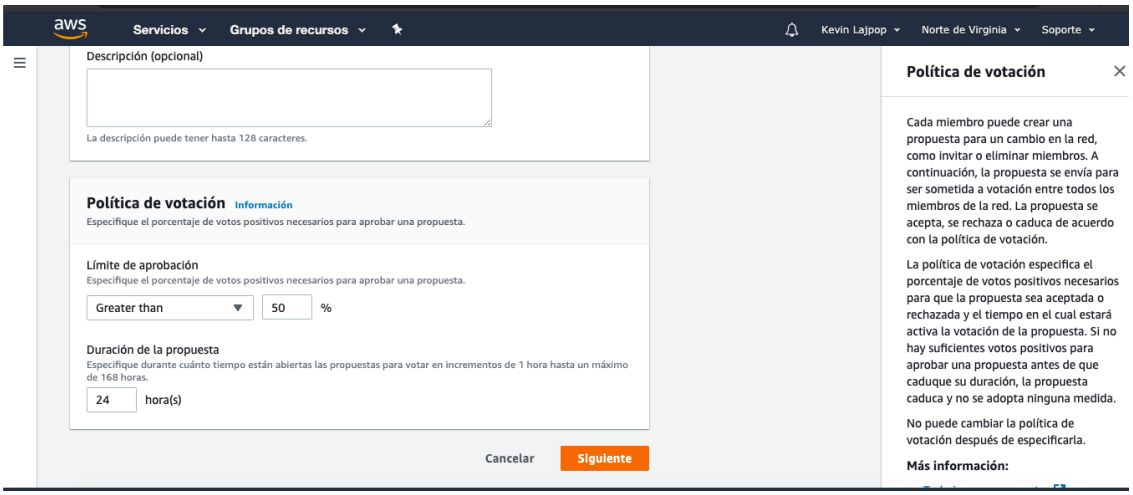
Lambda creation completed. API Gateway is active at:
https://sqskuz0731.execute-api.us-east-1.amazonaws.com/dev/

Que a la larga es la URL dónde se encuentra alojada nuestra función lambda, es decir, el resultado de la implementación del *blockchain* se resume en una sola URL que contiene las funciones para poder interactuar con el *blockchain*.

5.2.5. Consenso del *Blockchain*

Al momento de hablar de consenso en el *blockchain* en Amazon AWS, a lo que se tienen alcance es a configurar la política de votación de los miembros de la red, como se ve en la figura 4.

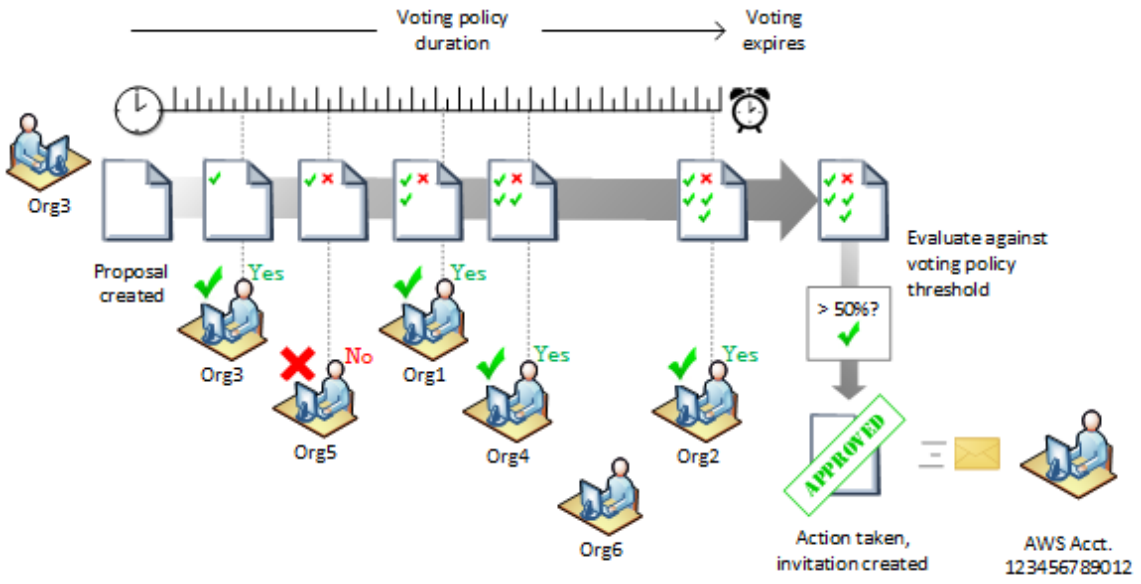
Figura 4. **Configuración de política de votación**



Fuente: elaboración propia, utilizando el programa Lucidchart.

Se configuraron dos cosas en la política de votación, el límite de votación y la duración de la propuesta, y estas reglas aplican al momento de que exista la acción de agregar un miembro a la red o eliminar un miembro de la red. El momento de hablar de límite de votación se habla sobre cuántos votos son necesarios para aceptar la propuesta ya sea de agregar un miembro o eliminarlo del a red, para este caso se aplicó la política estándar, que es 50 votos más 1 (50% + 1); para ese caso, se puede ver el funcionamiento en la siguiente figura:

Figura 5. **Concenso de 50 votos más 1**

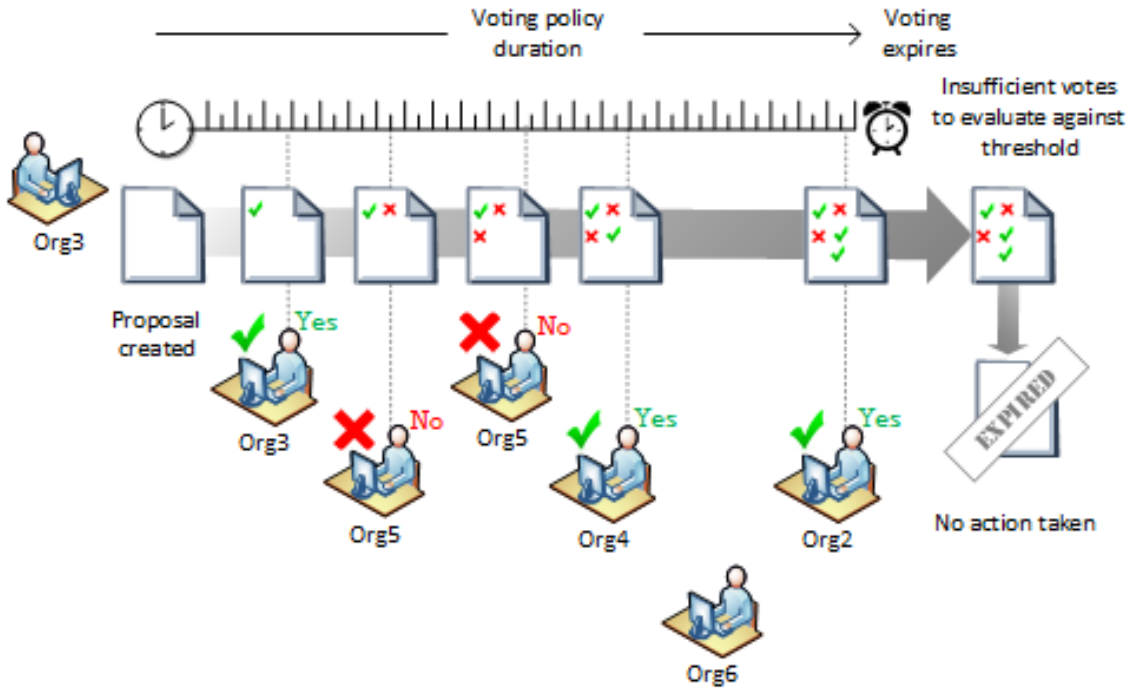


Fuente: Amazon AWS. *Work with Proposals*. Consultado el 18 de abril de 2020. Recuperado de https://docs.aws.amazon.com/managed-blockchain/latest/managementguide/managed-blockchain-proposals.html

El otro criterio inmerso en la política de votación es el tiempo el que dura la propuesta, para el prototipo se configuró el tiempo estándar que son veinticuatro horas (24 hrs) para la duración máxima de una propuesta, es decir que al término de estas veinticuatro horas y no se han recolectado la totalidad de votos no se procede a realizar la creación del miembro o eliminación de este, como se visualiza en la figura 6.

Cabe destacar que al momento de implementar el *blockchain* en la nube, el alcance que se tiene a la configuración de este es limitado, tal y como se observa en la figura 4, pero es necesario comprender que realiza cada campo configurado, es decir el límite de aprobación y la duración de la propuesta.

Figura 6. **Tiempo de expiración agotado para una propuesta**



Fuente: Amazon AWS. *Work with Proposals*. Consultado el 18 de abril de 2020. Recuperado de https://docs.aws.amazon.com/managed-blockchain/latest/managementguide/managed-blockchain-proposals.html

5.2.6. Integración de la plataforma

El resultado de la implementación del *blockchain* a nivel de software, fue una API REST en la cual se pueden realizar las inserciones de transacciones y consultas de esta. Es importante enmarcar que la inmutabilidad del *blockchain* hace que no exista un método de bajas y cambios, esa es la razón del porqué solamente existen métodos de inserciones y consultas de las transacciones.

Partiendo de esa aclaración, para la integración de la plataforma de Senacyt, se procedió a realizar un simple consumo de una API, lo único que se realizó para la integración del *blockchain* en la plataforma de Senacyt fue la agregación de la clase que se visualiza en la figura 7.

Esto porque a nivel de aplicación o plataforma se realiza únicamente la invocación a la API dado que toda la lógica del negocio o de la cadena como tal está inmersa en la nube, lo único que se debe realizar es la invocación, con lo que se desee realizar: inserción o consulta.

Figura 7. **Integración y consumo de funciones POST**

main.php | blockchain.php | blockchainController.php

```php
<?php

namespace Blockchain\Model;

class blockchain{

    function __construct(){

    }

    function getDonors(){
        $cURLConnection = curl_init();
        curl_setopt($cURLConnection, CURLOPT_URL, 'https://sqskuz0731.execute-api.us-east-1.amazonaws.com/dev/donors');
        curl_setopt($cURLConnection, CURLOPT_RETURNTRANSFER, true);
        $phoneList = curl_exec($cURLConnection);
        curl_close($cURLConnection);

        return $phoneList;
    }

    function postDonor($donor, $email = "senacyt@senacyt.gob.gt", $data){
        $hoy = date("Y-m-d H:i:s");
        $cURLConnection = curl_init();
        curl_setopt($cURLConnection, CURLOPT_URL, 'https://sqskuz0731.execute-api.us-east-1.amazonaws.com/dev/donors');
        curl_setopt($cURLConnection, CURLOPT_POST, 1);
        curl_setopt($cURLConnection, CURLOPT_POSTFIELDS,
            "donorUserName=$donor&email=$email&registeredDate=$hoy&data=$data"
       );
        curl_setopt($cURLConnection, CURLOPT_RETURNTRANSFER, true);
        $phoneList = curl_exec($cURLConnection);
        curl_close($cURLConnection);

        return $phoneList;
    }
}
```

Fuente: elaboración propia.

5.2.6.1. Método getDonors

El método *getDonors* que se visualiza en la figura 7 es, como su nombre lo indica, para obtener las transacciones de que se han registrado en la red, para ello se realiza una petición al API: https://sqskuz0731.execute-api.us-east-1.amazonaws.com/dev/donors. Esto es lo único que se debe realizar para obtener las transacciones de la red.

5.2.6.2. Método *postDonor*

El método *getDonors* que se visualiza en la figura 7 es, como su nombre lo indica, para registrar una transacción en la red, para ello se realiza una petición al API: https://sqskuz0731.execute-api.us-east-1.amazonaws.com/dev/donors, a esto se le suma que se le debe enviar unos parámetros a la cabecera de la petición, siendo estos los siguientes:

- DonorUserName: representa el nombre o usuario de la persona que está realizando la transacción.

- Email: representa el correo electrónico de la persona que está realizando la transacción:

- RegisteredDate: fecha en la que se realiza la transacción.

- Data: datos que se registran o almacenan en el *blockchain*, en este caso, se realiza todo lo que dicta los procesos de investigadores y proyectos de la Senacyt.

Al momento de registrar una transacción con sus cabeceras respectivas es inmutable, es decir, no hay forma de que pueda ser eliminada o modificada, por lo que, un método de eliminación o modificación no existe, como se mencionó anteriormente.

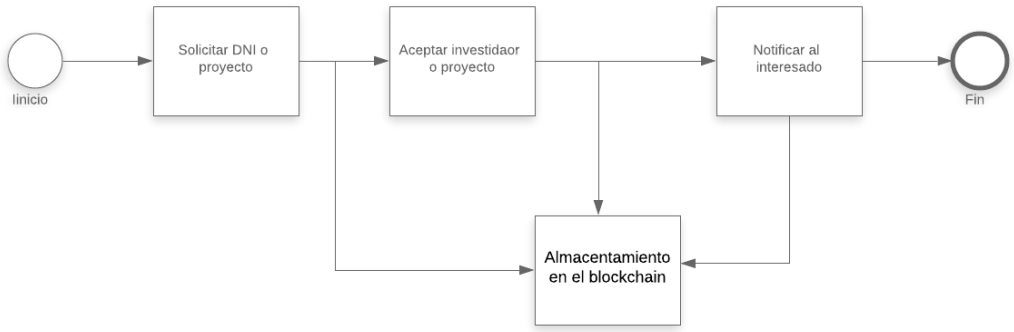
5.3. Pruebas e integración final

Al momento de hablar de pruebas e integración es necesario saber que flujo se va a seguir para hacer las pruebas e integración, así como los flujos o procesos específicos donde se realizaran las pruebas y las integraciones.

5.3.1. Flujo de la funcionalidad

En el caso del flujo de la funcionalidad, se tiene lo siguiente:

Figura 8. **Proceso general del funcionamiento**



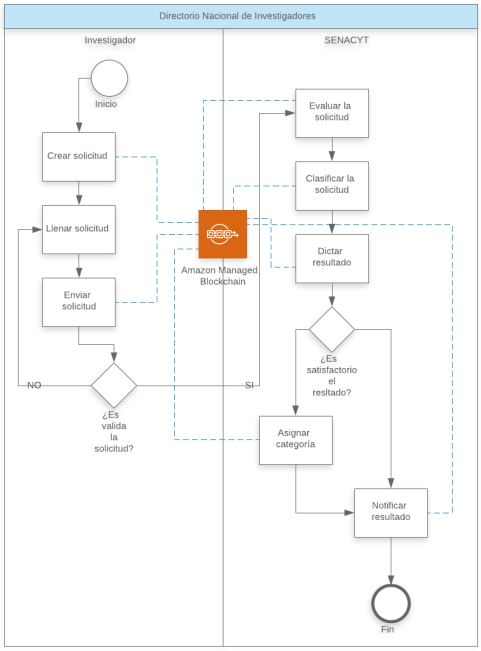
Fuente: elaboración propia, utilizando el programa Lucidchart.

Como se observa en el diagrama de flujo general de funcionamiento o pruebas, por cada acción que tiene el proceso de solicitud de investigador o de apoyo financiero para un proyecto, se invoca a una especie de subproceso o acción alterna, la cual consistirá en el registro de las transacciones en la cadena.

5.3.2. Proceso de inserción de una transacción y el Directorio Nacional de Investigadores

El Directorio Nacional de Investigadores, DNI, es el registro donde se encuentran, valga la redundancia todos los investigadores guatemaltecos, para pertenecer al DNI hay que crear un proceso de admisión vía la plataforma de la Senacyt.

Figura 9. **Proceso general del Directorio Nacional de Investigadores**



Fuente: elaboración propia, utilizando el programa Lucidchart.

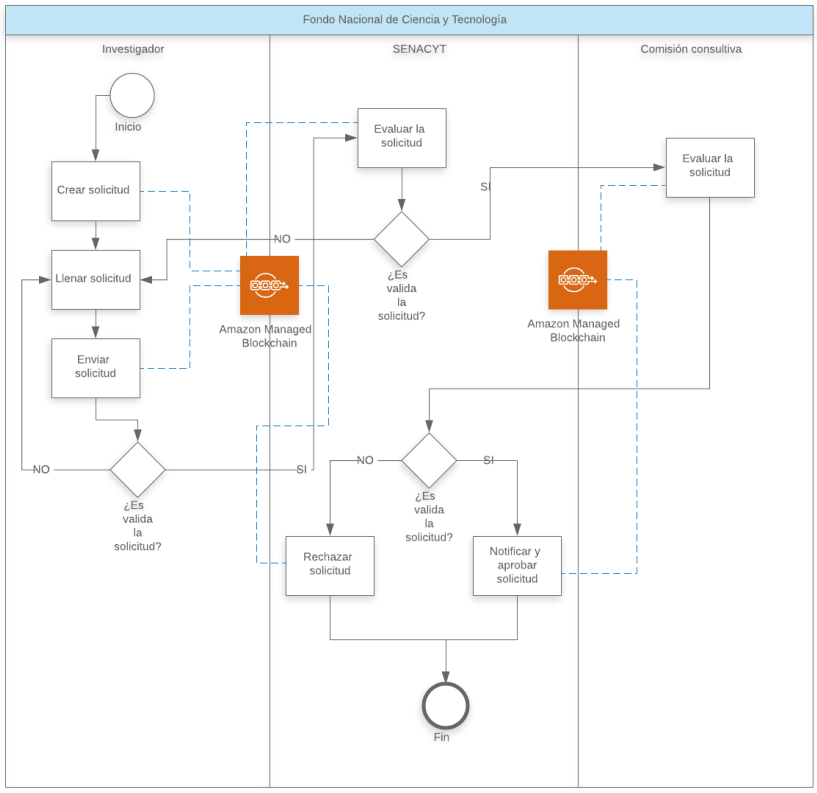
En la figura 9 se tiene el proceso detallado de como interactúa el proceso general de inscripción de una persona en el Directorio Nacional de Investigadores con el *blockchain*.

Se observa que a largo del proceso hay interacción con el *blockchain*, solamente que es de manera asíncrona, es decir, que el proceso de inscripción de una persona al Directorio Nacional de Investigadores no se ve afectado por la interacción con el *blockchain*. Se observa que cada fin de transacción se acude a realizar una inserción en el *blockchain*, con ello se le está dando cierta independencia a ambos sistemas, al transaccional de Senacyt como al *blockchain*.

5.3.3. Proceso de inserción en el Fondo Nacional de Ciencia y Tecnología

Tal y como pasó en el Directorio Nacional de Investigadores, el proceso de inserción de datos en el Fondo Nacional de Ciencia y Tecnología, FONACYT, es similar, tal y como se aprecia en el diagrama de procesos visualizado en la figura 10, el proceso es de manera asíncrona, es decir que no es necesario que exista una respuesta inmediata o que servicio como tal del *blockchain* esté arriba, sino únicamente al momento de que exista una transacción existe una inserción en el *blockchain*, sin ser vital en el proceso como tal la inserción en la cadena.

Figura 10. **Proceso general del Fondo Nacional de Ciencia y Tecnología**



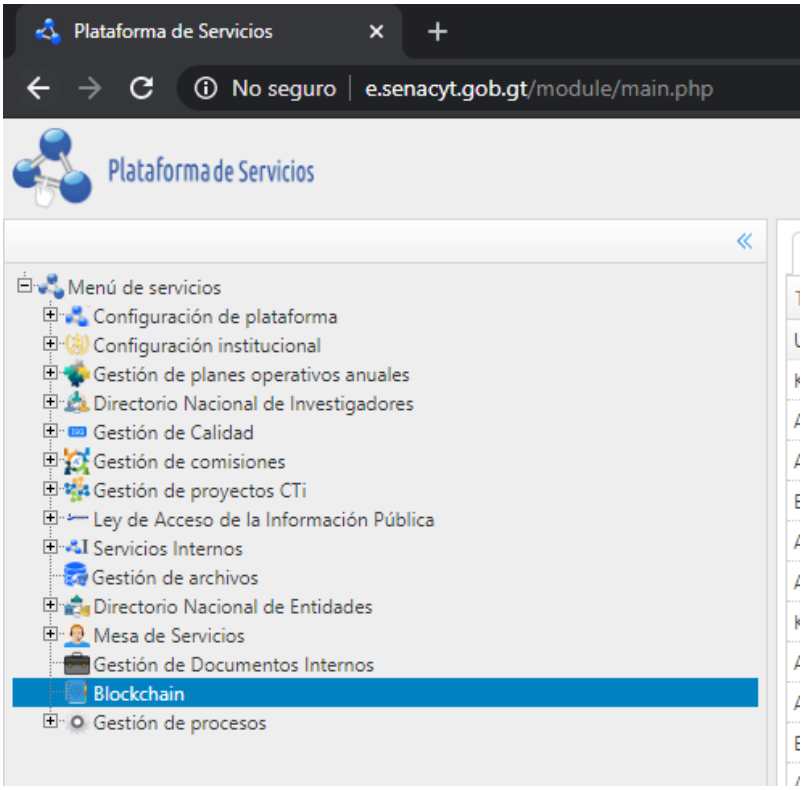
Fuente: elaboración propia, utilizando el programa Lucidchart.

5.3.4. Consulta de datos ingresados en la cadena

Todo lo expuesto con anterioridad desemboca en el poder de transparencia e inmutabilidad que brinda el *blockchain*, pero para esto también es necesario que las transacciones se puedan consultar, para ello se debe seguir con la arquitectura final de la figura 3, en la cual ya se observa la interacción con la Plataforma de Servicios en Línea de la Senacyt.

El proceso de consultas es tan sencillo como se observa en la figura 3, y por el momento solo ese consumo solamente se realiza por medio de la plataforma de servicios en línea de Senacyt, para realizar esa vinculación, se registró en los módulos de dicha plataforma. Para que posteriormente asignarle permisos al perfil de administrador general el permiso de visualizar el *blockchain*; como se observa en la figura 11 ya se encuentra cargado el módulo de *blockchain*.

Figura 11. **Módulo *Blockchain* en la Plataforma de Servicios en Línea**

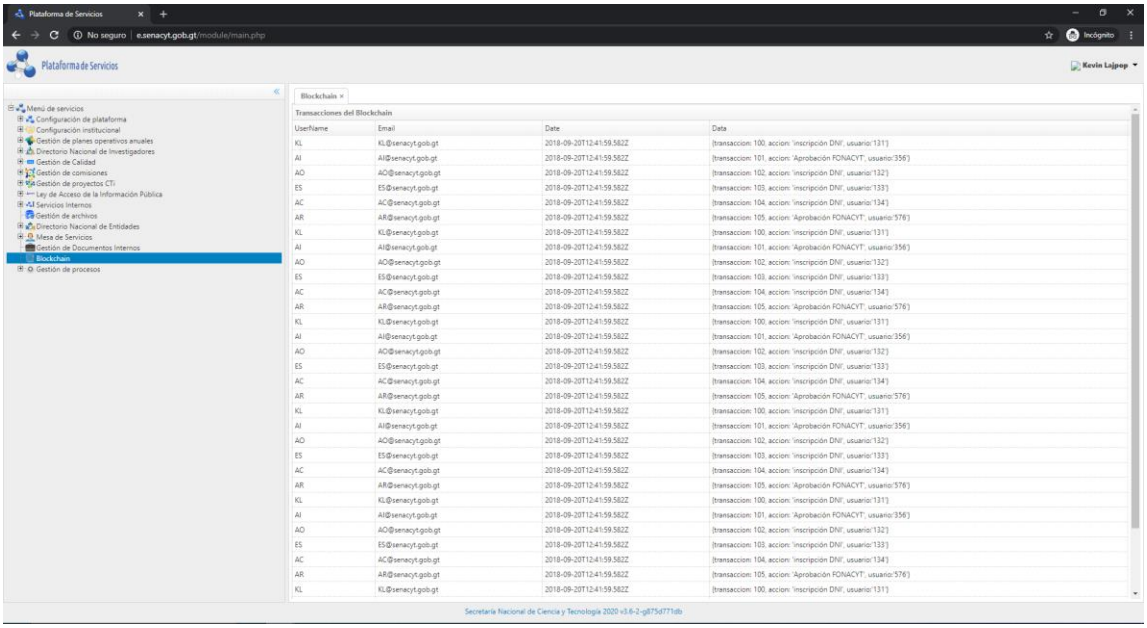


Fuente: elaboración propia.

Luego de ingresar al módulo de *blockchain*, se observa la pantalla que se muestra en la figura 12, recordando que las transacciones únicamente tienen

consulta, es por eso de que la vista del *blockchain* es muy sencilla, dado que no tiene más acciones, y cabe recordar que las inserciones en la cadena se realizan a nivel de código dentro del funcionamiento transaccional.

Figura 12. **Plataforma de Servicios en Línea de Senacy**



Fuente: elaboración propia.

6. DISCUSIÓN DE RESULTADOS

6.1. *Blockchain* implementado en la nube y *blockchain on premise*

Al momento de implementar un *blockchain*, se tiene una gama de opciones, que van desde herramientas, tipos y sobre todo infraestructura; al momento de hablar de infraestructura es importante ver que existen ventajas y desventajas a la hora de implementar un *blockchain* en la nube u *on premise* como comúnmente se le conoce. En la presente investigación se realizó una implementación de infraestructura en la nube, por lo que se observan las siguientes ventajas y desventajas contra una implementación *on premise*.

Tabla I. **Comparativa de un *blockchain* implementado en la nube versus uno implementado on premise**

Característica	En la nube	*On premise*
Facilidad de implementación a nivel de red	Fácil	Muy difícil
Control sobre el algoritmo y regla del consenso	Nulo	Control total
Facilidad de implementación a nivel lógico	Fácil	Muy difícil

Continuación de la tabla I.

Nivel de confianza	Muy alta	Media
Escalable a nivel de infraestructura	Sí	No
Balanceador de carga nativamente	Sí	No
Nivel de encapsulamiento	Muy alto	Bajo
Precio	Muy alto (alrededor de US$ 200.00 al mes)	Bajo

Fuente: elaboración propia.

En la tabla I se observan varias características de la implementación del prototipo, al momento de ser una solución en la nube, la facilidad de implementarlo se multiplica, porque la nube presta todo, los recursos, la documentación sólida paso a paso, es una ventaja sobre la solución *on premise* porque básicamente se olvida uno de la puesta de una infraestructura física es un ahorro grande en tiempo y esfuerzo.

Aunque el hecho de que se deje el control total de la infraestructura en la nube también tiene una desventaja y es que no se tiene ningún control sobre el algoritmo de consenso, esto da la pauta de que no se puede realizar ningún cambio, no se puede desarrollar a medida o como los requerimientos necesiten y este es el caso del consenso, al momento de implementar el *blockchain* en la nube no se tiene control del algoritmo de consenso únicamente se puede decir cuánto tiempo y cuánto es la taza de decisión para una transacción.

Pero no se puede implementar ni se tiene control sobre el algoritmo que se utiliza, no existe un desarrollo como tal en esa área, esto puede ser bueno o malo dependiendo de la necesidad que se tenga porque si se desea implementar un algoritmo en específico la nube no lo permitirá, en el caso de que se desee implementar un *blockchain* sin importar en algoritmo de consenso la opción de la nube es bastante factible.

A nivel de confianza la solución en la nube goza de una estabilidad bastante buena, porque es una solución internacional que es utilizada a nivel de producción, por lo que la confianza que provee una solución de Amazon AWS es bastante asegurada contra una solución creada *on premise*.

El *blockchain* en la nube tiene todas las bondades de la nube, en este caso se habla sobre la escalabilidad que se tiene a nivel de infraestructura, a los nodos que son parte del *blockchain* se le puede aumentar los recursos en ámbitos de producción dado que la solución de Amazon AWS lo permite, mientras que una solución *on premise* esta escalabilidad o elasticidad no es tan sencilla de implementarlo dado que se necesita un proceso de más cuidado sobre este crecimiento.

Nativamente la solución tiene un balanceador de carga, mientras que *on premise* no se tiene esa opción nativamente y también esto habla sobre el nivel de encapsulamiento que tiene dado que no se ve todo lo que está detrás de este tipo de solución, dado que provee únicamente una API para que se maneje toda la interacción, sin necesidad de contar con otro tipo de accesos o soluciones adjuntas.

Por último, se tiene el tema monetario, el costo de implementar el *blockchain* es bastante alto, dado que se paga por cada máquina que es parte de este,

también se paga por hora (US$ 0.30) la membresía de pertenecer al *blockchain* internacional, eso hace un precio más de US$ 200 mensualmente, un precio bastante elevado para una entidad pública lo que es una desventaja clara a la hora de compararlo contra una solución *on premise*.

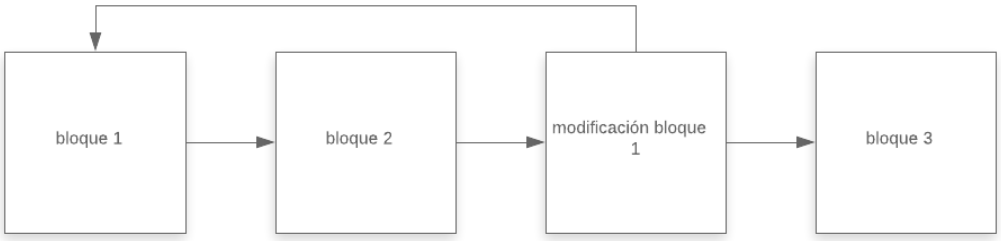
6.2. Altas, bajas, cambios y consultas del *blockchain*

Una de las bases del *blockchain* es la inmutabilidad que tiene en sus datos ingresados en a la cadena, esto porque lo que naturalmente existe es un proceso de altas únicamente, una transacción ingresada en el *blockchain* se queda para la eternidad dentro de la cadena, parte de la inmutabilidad es que no exista en su esencia bajas ni cambios, únicamente altas y consultas como tal de las transacciones.

En el caso de las altas el proceso es transparente, porque únicamente es una inserción limpia dentro de la cadena, para las consultas, dependiendo de cómo se quiera consultar, no existe una gran gama de consultas, sino únicamente es obtener toda las transacciones totales o las transacciones por usuario.

Para los cambios como tal, no se modifica el bloque o transacción insertada en la cadena, bajo el principio de inmutabilidad del *blockchain* lo que se realiza es la inserción de un nuevo nodo a la cadena referenciando al nodo que se desea modificar, es decir, únicamente se notifica que la nueva transacción es una actualización al nodo o transacción referenciada, pero no existe una modificación física como tal, la modificación o cambios se manejan de la siguiente manera (ver figura 13).

Figura 13. **Proceso de actualización de un bloque o transacción**



Fuente: elaboración propia, utilizando el programa Lucidchart.

Como se ha mencionado en reiteradas ocasiones, no existe un proceso como tal de bajas o eliminación como tal, porque es el principio del *blockchain*, el principio de inmutabilidad y dicta que para la eliminación de una transacción deberían estar todos los nodos o miembros de la cadena a favor de ello, y en su principio, ningún nodo ni la plataforma como tal van a estar de acuerdo con una eliminación.

El proceso de consultas es bastante básico y únicamente es consumir el API indicando si se quiere consumir todo por completo o únicamente las transacciones de un usuario en específico, cabe destacar que ante la personas o población en general ellos contaran únicamente con la opción de visualización o consulta de las transacciones, mientras que la plataforma de Senacyt cuenta con la opción de crear las transacciones como tal.

6.3. Bondades y ventajas del *blockchain* en Senacyt

Una institución pública está siempre presta para auditoría social, entes de auditoría y de otra índole, auditando el funcionamiento como tal de la institución

y muchas veces no basta con tener información al día, sino contar con mecanismos que respalden la información y los procesos como tal para poder brindar la transparencia que se espera por parte de una institución pública.

Lo que brinda un *blockchain* bajo su principio de inmutabilidad es la transparencia, el beneficio es que se brinda una herramienta de auditoría para la población y los entes de auditoría en el cual se podrá visualizar todas las transacciones que realiza la Senacyt de igual manera el hecho de contar con una herramienta de este tipo permitirá la automatización de todo un flujo de procesos, es decir desde su concepción hasta su auditoría.

De igual manera con la implementación de esta herramienta la Senacyt sería pionera a nivel regional centroamericana en implementar una solución de este tipo, siendo innovadora y agregando valor y presencia a la Senacyt a nivel nacional y regional, por ser una secretaria automatizada e innovadora, poniéndose a nivel de gobiernos como el de Argentina y Chile que cuentan con este tipo de herramienta para auditar e incluso para contratos.

CONCLUSIONES

1. Se implementó un prototipo de una plataforma de gestión de datos distribuida, permanente y segura utilizando tecnología *blockchain* de Amazon AWS para los datos de científicos y proyectos de la Senacyt. La implementación de un *blockchain* en la Senacyt es un avance tecnológico de gran importancia por lo innovador y transparente que es la tecnología esto viene a ser de ayuda a una institución gubernamental por el tema de transparencia que deben de manejar.

2. Se implementó un algoritmo de consenso en la nube de Amazon AWS para el almacenamiento de *blockchain* de los datos científicos y proyectos de la Senacyt. El algoritmo de consenso, cuando se implementa el *blockchain* en la nube de Amazon AWS, está completamente encapsulada; es decir, no se tiene una interacción real sobre este; así mismo, tampoco se tiene injerencia en la implementación del algoritmo de consenso, por lo que a la hora de implementar la solución en la nube de Amazon AWS lleva por defecto el algoritmo de consenso, que deja como único alcance la configuración de las políticas de votación y tiempo de duración de las propuestas para agregar o eliminar un miembro de la red.

3. Se diseñó y desarrolló el proceso de altas y consultas de datos almacenados en el *blockchain* implementado en Amazon AWS, dicho proceso de altas y consultas de transacciones son las bases para dicha tecnología, dado que, para que exista datos, hay que crear la transacción y darla de alta. Por otra parte, el proceso de consultas es necesario para el funcionamiento y transparencia que lo precede, los procesos de bajas y

cambios no existen en la esencia del *blockchain*, porque eso estaría rompiendo la inmutabilidad, al momento de existir una baja o un cambio en una transacción perdería toda inmutabilidad y todo sentido de su lógica, por lo que se puede concluir que bajas y cambios no existen en la esencia y construcción del *blockchain*. El prototipo construido en este trabajo implementa los procesos de cambios y bajas utilizando transacciones que generan un alta de información que corresponde a la información actualizada.

4. Se implementó la plataforma de gestión de datos distribuida, permanente y segura en la nube de Amazon AWS y se pudo observar que implementar un *blockchain* en la nube de Amazon AWS es una solución robusta, completa y fácil de implementar en comparación con Google Cloud que tiene una solución en construcción sin ninguna consola de administración o Microsoft Azure que la solución en *blockchain* viene con paquetes extra que no era indispensable para la solución, de igual manera se debe contar con una solidez financiera para una solución de esta índole, dado que el precio por la membresía dentro de Amazon AWS es de US$ 0.30 por hora, haciendo un cobro por día de US$ 7.20 sin tomar en cuenta el precio de los nodos que depende de las necesidades que se requieran.

RECOMENDACIONES

1. El uso del *blockchain* en la Senacyt pase de ser un prototipo a una plataforma en producción, dado que agrega transparencia a las acciones de la institución a raíz del algoritmo de consenso que tiene la tecnología; de igual manera, agrega seguridad a los datos de interés de la secretaría, por los principios de inmutabilidad que tiene el *blockchain*; por último, agrega innovación a la institución y a la administración pública guatemalteca por ser pionera en el sector público y en la región en cuanto al tipo de soluciones basadas en *blockchain*.

2. Cuando se implemente el *blockchain* y se tenga que definir la política de votación, el límite de aprobación de votos debe ser de cincuenta (50) votos más uno (1), es decir, la mayoría que decide; si bien es cierto se puede tener la libertad de elegir cualquier comportamiento de la política de votación, lo más recomendable, ético y transparente es el voto de cincuenta (50) más uno (1). En el caso de la duración de las propuestas que entren en consenso se recomienda que sea por veinticuatro (24) horas como máximo para que todas las partes o nodos de la cadena puedan votar.

3. El hecho que en *blockchain* no exista un proceso de bajas ni cambios (nativos) hace que sea necesario estudiar bien el ambiente en dónde será implementada la solución, dado por su carácter inmutable; no todos los sistemas de información pueden hacer uso de una tecnología como esta; de ser así, se debe tener en consideración una estrategia para asociar el

giro del negocio a la inmutabilidad porque carece de un proceso de bajas y cambios como tal.

4. Este tipo de soluciones se implemente en ambientes y aplicaciones que en realidad lo requieran, tecnológicas y monetarias; de esa forma, se le podrá dar un uso eficiente de toda la arquitectura y los servicios que propone Amazon AWS; esto porque al momento de implementar un *blockchain* el precio mensual de la membresía es bastante alto.

REFERENCIAS

1. Bashir, I. (2018). *Mastering Blockchain: Distributed ledger technology, decentralization, and smart contracts explained.* Birmingham, UK: Packt Publishing Ltd.

2. Cachin, C. (julio, 2016). Architecture of the hyperledger *blockchain* fabric. *Workshop on Distributed Cryptocurrencies and Consensus Ledgers, 310(4)*. Recuperado de https://www.zurich.ibm.com/dccl/papers/cachin_dccl.pdf

3. Han, J., Haihong, E., Le, G., y Du, J. (octubre, 2011). Survey on NoSQL database. *Pervasive computing and applications (ICPCA), 2011 6th international conference,* 1(1), (pp. 363-366).

4. Gaur, N. (2018). *Hands-On Blockchain with Hyperledger: Building decentralized applications with Hyperledger Fabric and Composer*. Birmingham, UK: Packt Publishing Ltd.

5. Gupta, M. (2017). *Blockchain for dummies*. Hoboken, USA: John Wiley & Sons.

6. Hurwitz, J. (2012). *Cloud Services for dummies*. Hoboken, USA: John Wiley & Sons.

7. Keeton, K., Santos, C. A., Beyer, D., Chase, J. S., y Wilkes, J. (marzo, 2004). Designing for Disasters. *FAST*, 4(1), (59-62).

8. McConaghy, T., Marques, R., Müller, A., De Jonghe, D., McConaghy, T., McMullen, G., ... y Granzotto, A. (2016). BigchainDB: a scalable *blockchain* database*. white paper BigChainDB*, 1(1), (1-64). Recuperado de https://mycourses.aalto.fi/pluginfile.php/378362/mod_resource/content/1/bigchaindb-whitepaper.pdf

9. Nadareishvili, I., Mitra, R., McLarty, M., y Amundsen, M. (2016). *Microservice architecture: aligning principles, practices, and culture*. Sebastopol, USA: O'Reilly Media, Inc.

10. Navarro, B. (2017). *Blockchain y sus aplicaciones*, Asunción, Paraguay: Universidad Católica Nuestra Señora de la Asunción. Recuperado de http://jeuazarru.com/wp-content/uploads/2017/11/Blockchain.pdf

11. Newman, S. (2015). *Building microservices: designing fine-grained systems*. Sebastopol, USA: O'Reilly Media, Inc.

12. Raval, S. (2016). *Decentralized Applications: Harnessing Bitcoin's Blockchain Technology*. Sebastopol, USA: O'Reilly Media, Inc.

13. Rountree, D., y Castrillo, I. (2013). *The basics of cloud computing: Understanding the fundamentals of cloud computing in theory and practice*. Waltham, USA: Elsevier.

14. Shao, Y., Di, L., Bai, Y., Guo, B., y Gong, J. (Agosto, 2012). *Geoprocessing on the Amazon cloud computing platform—AWS*.

2012 first international conference on agro-geoinformatics (agro-geoinformatics), 1(1), (pp. 1-6).

15. Swan, M. (2015). *Blockchain: Blueprint for a new economy*. Sebastopol, USA: O'Reilly Media, Inc.

16. Tapscott, D., y Tapscott, A. (2017). *La revolución blockchain. Descubre cómo esta nueva tecnología transformará la economía global*. Barcelona, España: Ediciones Deusto.

17. Van Rossum, G. (junio, 2007). Python Programming Language. *USENIX Annual Technical Conference*, 41(1), (36).

18. Zyskind, G., y Nathan, O. (mayo, 2015). Decentralizing privacy: Using *blockchain* to protect personal data. *2015 IEEE Security and Privacy Workshops*, 1(1), (180-184).