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

DISEÑO E IMPLEMENTACIÓN DE UNA ARQUITECTURA BASADA EN BLOCKCHAIN PARA UN SISTEMA DE MANEJO DE REGISTROS ACADÉMICOS DENTRO DE UN EXPEDIENTE ESTUDIANTIL DIGITAL

Ing. Jorge Mario Rubio Vidal
Asesorado por el Ing. y M. C. Edwin Estuardo Zapeta Gómez

Guatemala, noviembre de 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

DISEÑO E IMPLEMENTACIÓN DE UNA ARQUITECTURA BASADA EN BLOCKCHAIN PARA UN SISTEMA DE MANEJO DE REGISTROS ACADÉMICOS DENTRO DE UN EXPEDIENTE ESTUDIANTIL DIGITAL

TRABAJO DE GRADUACIÓN

PRESENTADO A JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

ING. JORGE MARIO RUBIO VIDAL
ASESORADO POR EL ING. Y MC. EDWIN ESTUARDO ZAPETA GÓMEZ

AL CONFERÍRSELE EL TÍTULO DE

MAESTRO EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN

GUATEMALA, NOVIEMBRE DE 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

NÓMINA DE JUNTA DIRECTIVA

DECANA	Ing. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Kevin Vladimir Cruz Lorente
VOCAL V	Br. Fernando José Paz González
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANA	Ing. Aurelia Anabela Cordova Estrada
EXAMINADORA	Ing. Sasha Steffanie Palencia Zetina
EXAMINADOR	Ing. Marlon Antonio Pérez Türk
EXAMINADOR	Ing. Edgar Darío Álvarez Cotí
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:

DISEÑO E IMPLEMENTACIÓN DE UNA ARQUITECTURA DISTRIBUTIDA BASADA EN BLOCKCHAIN PARA UN SISTEMA DE MANEJO DE REGISTROS ACADÉMICOS DENTRO DE UN EXPEDIENTE ESTUDIANTIL DIGITAL

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

Ing. Jorge Mario Rubio Vidal

USAC
TRICENTENARIA
Universidad de San Carlos de Guatemala

Decanato
Facultad de Ingeniería
24189101 – 24189102
secretariadecanato@ingenieria.usac.edu.gt

DTG. 697.2021.

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: **DISEÑO E IMPLEMENTACIÓN DE UNA ARQUITECTURA BASADA EN BLOCKCHAIN PARA UN SISTEMA DE MANEJO DE REGISTROS ACADÉMICOS DENTRO DE UN EXPEDIENTE ESTUDIANTIL DIGITAL,** presentado por el Ingeniero: **Jorge Mario Rubio Vidal,** estudiante de la **Maestría de 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:

Inga. Anabela Cordova Estrada
Decana

Guatemala, noviembre de 2021.

AACE/asga

Guatemala, noviembre de 2021

LNG.EEP.OI.133.2021

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 del asesor, verificar la aprobación del Coordinador de Maestría y la aprobación del Área de Lingüística al trabajo de graduación titulado:

"DISEÑO E IMPLEMENTACIÓN DE UNA ARQUITECTURA BASADA EN BLOCKCHAIN PARA UN SISTEMA DE MANEJO DE REGISTROS ACADÉMICOS DENTRO DE UN EXPEDIENTE ESTUDIANTIL DIGITAL"

presentado por **Jorge Mario Rubio Vidal** quien se identifica con carné **201020415** 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"

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA FACULTAD DE INGENIERÍA DIRECCIÓN ESCUELA DE POSTGRADO

Mtro. Ing. Edgar Darío Álvarez Cotí
Director
Escuela de Estudios de Postgrado
Facultad de Ingeniería

Guatemala, 21 de julio 2020.

M.A. Edgar Darío Álvarez Cotí
Director
Escuela de Estudios de Postgrado
Presente

M.A. Ingeniero Álvarez Cotí:

Por este medio informo que he revisado y aprobado el **TRABAJO DE GRADUACIÓN** titulado: "DISEÑO E IMPLEMENTACIÓN DE UNA ARQUITECTURA BASADA EN BLOCKCHAIN PARA UN SISTEMA DE MANEJO DE REGISTROS ACADÉMICOS DENTRO DE UN EXPEDIENTE ESTUDIANTIL DIGITAL" del estudiante Jorge Mario Rubio Vidal quien se identifica con número de carné 201020415 del programa de **Maestría en Tecnologías de la Información y la Comunicación.**

Con base en la evaluación realizada hago constar que he evaluado la calidad, validez, pertinencia y coherencia de los resultados obtenidos en el trabajo presentado y según lo establecido en el *Normativo de Tesis y Trabajos de Graduación aprobado por Junta Directiva de la Facultad de Ingeniería Punto Sexto inciso 6.10 del Acta 04-2014 de sesión celebrada el 04 de febrero de 2014*. Por lo cual el trabajo evaluado cuenta con mi aprobación.

Agradeciendo su atención y deseándole éxitos en sus actividades profesionales me suscribo.

Atentamente,

MARLON ANTONIO PEREZ TURK
INGENIERO EN CIENCIAS Y SISTEMAS
COLEGIADO No. 4492

MSc. Ing. Marlon Antonio Pérez Türk
Coordinador
Maestría en Tecnologías de la Información y la Comunicación
Escuela de Estudios de Postgrado

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Estudios de Postgrado
Coordinador de Área

Guatemala, 14 de septiembre de 2020.

M.A. Ing. Edgar Dario Álvarez Cotí
Director
Escuela de Estudios de Postgrados
Presente

Estimado M.A. Ing. Álvarez Cotí:

Reciba un cordial y atento saludo, a la vez aprovecho la oportunidad para hacerle de su conocimiento que he revisado y aprobado el trabajo especial de graduación titulado: "DISEÑO E IMPLEMENTACIÓN DE UNA ARQUITECTURA BASADA EN BLOCKCHAIN PARA UN SISTEMA DE MANEJO DE REGISTROS ACADÉMICOS DENTRO DE UN EXPEDIENTE ESTUDIANTIL DIGITAL" del estudiante Jorge Mario Rubio Vidal del Programa de Maestría en Tecnología de la Información y Comunicación, identificado con número de carné: *201020415*.

Agradeciendo su atención y deseándole éxitos en sus actividades profesionales me suscribo.

"Id y enseñad a todos"

MSc. Ing. Edwin Estuardo Zapeta Gómez
Colegiado No. 12767
Asesor

Cc: Archivo/LA

AGRADECIMIENTOS A:

Mis padres	Por el apoyo, amor y paciencia incondicional en esta aventura.
Mi asesor	Ing. Edwin Zapeta, por su tiempo, esfuerzo, orientación y aporte en el desarrollo del trabajo de graduación.
Mis catedráticos	Por compartir su conocimiento y experiencia e ir siempre más allá de la expectativa y contenido en libros.
Universidad de San Carlos de Guatemala	Por ser un nacimiento de conocimiento y oportunidades para nosotros los guatemaltecos.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES

FIGURAS

TABLAS

LISTA DE SÍMBOLOS

Símbolo	Significado
≈	Aproximadamente
JS	Archivo JSON
O(n)	Función O grande de n
KB	Kilobyte
FK	Llave primaria
MB	Megabyte
ms	Milisegundo
n/a	No aplica
#	Numeral
%	Porcentaje
^	Potencia
s	Segundos
Seg	Segundos
Tx	Transacción enviada
tps	Transacciones por segundo

GLOSARIO

ACID	Propiedades de atomicidad, consistencia, aislamiento y durabilidad de un sistema de manejo de base de datos.
ADS	*Academic Data Store.*
API	*Application Programming Interface.*
AR	*Academic Record.*
AWS	Amazon Web Services, colección de servicios de computación en la nube proveída por Amazon.com
Azure	Colección de servicios de computación en la nube proveída por Microsoft.
B2B	*Business to Business.*
B2C	*Business to Consumer.*
CA	*Certificate Authority..*
CLI	*Command Line Interface.*

Cloud9	Entorno de desarrollo integrado basado en la nube que permite escribir, ejecutar y depurar código en la nube a través de un explorador web.
Cloudformation	Servicio en la nube que permite modelar y aprovisionar todos los recursos de infraestructura necesarios a través de código.
Coin – age	Número que representa la relación entre cantidad de monedas y el tiempo que se han tenido consigo. Utilizado como método de selección en algoritmos de consenso del tipo PoS.
Criptografía	Rama de la criptología que se ocupa de las técnicas de cifrado o codificado destinadas a alterar la representación lingüística de un mensaje.
Criptografía asimétrica	Llamado también criptografía de llave pública ya que utiliza un par de llaves (pública y privada) con la cual intercambiar de forma segura mensajes.
Criptografía simétrica	Llamada también criptografía de llave secreta utiliza una única llave para cifrar y descifrar mensajes en el emisor y receptor.

Criptomoneda	Es un medio digital que utiliza criptografía fuerte para asegurar transacciones, crear unidades y verificar la transferencia de activos utilizando tecnologías de registro distribuido.
Cross-Industry	Hace referencia a la aplicación de un concepto en varios tipos de industria, normalmente con giros de negocio diferentes.
CRUD	Término que hace referencia a las operaciones básicas que se pueden realizar en una base de datos (crear, leer, actualizar, borrar).
DBMS	*Database Management System.*
ECDSA	*Elliptic Curve Digital Signature Algorithm.*
ECERT	*Enrollment Certificate.*
Ether	Nombre de la criptomoneda de la *blockchain* Ethereum.

Función O(n)	Notación matemática con la cual se puede evaluar y expresar el tiempo de ejecución de un algoritmo.
GDPR	Es el reglamento europeo relativo a las personas físicas en lo que respecta a como son tratados sus datos personales y la circulación de estos en dominios de internet de su zona.
IBM	*International Business Machine Corporation.*
Inmutabilidad	Es la cualidad de un objeto de no ser susceptible a modificaciones una vez creado. Permite tener certeza de que no existan cambias inesperados para las características ya definidas en su creación.
ISO	*International Standarization Organization.*
MOOC	*Massive Online Open Course.*
MSP	*Membership Service Provider.*
NVM	*Node Version Manager.*
OAIS	*Open Archive Information System.*

Peer to Peer	Red de pares, es una red de nodos en los que todos se comportan como iguales entre sí, actuando simultáneamente como cliente y servidor respecto a los demás nodos de la red.
POS	*Proof Of Stake.*
POW	*Proof Of Work.*
RAM	*Random Access Memory.*
REST	*Representational State Transfer.*
SDK	*Standard Development Kit.*
SHA	*Secure Hash Algorithm.*
Skylake	Nombre código de la arquitectura de microprocesadores Intel lanzados en agosto de 2015.
TCERT	*Transactional Certificate.*
TLS/SSL	Algoritmos de seguridad criptográfica basados en infraestructura de llave pública para la capa de transporte del modelo de red TCP/IP.

VPC	Servicio comercial de Amazon que proporciona al usuario una nube privada virtual.
YAML	Es un formato de serialización de objetos legible por humanos usado ampliamente para archivos de configuración en aplicaciones donde los datos son almacenados o transmitidos.

RESUMEN

Los registros académicos forman parte del expediente estudiantil que pueda llevar cualquier organización educativa y de los datos personales de cada estudiante y, como tales, deben ser verificables y seguros para garantizar su inmutabilidad una vez generados.

El problema con el manejo de los registros académicos es que su existencia aún está ligada a un registro físico que corrobore los datos obtenidos del estudiante. Lo que a su vez limita la capacidad de auditar el origen y la veracidad de los datos, mediante procesos de auditoría manual, procesos que son poco eficientes, ya que se debe de mantener un organizado modelo de archivo de datos que permita garantizar que estos son exactos y que durante el proceso de generación, manejo y preservación estos registros físicos no sufren ninguna alteración.

En este trabajo, se abordó el problema de registrar en un sistema de información los datos académicos que conforman un expediente estudiantil físico y el riesgo de que los datos tengan un origen dudoso. Para resolver este problema, la arquitectura propuesta se basa en el uso de *blockchain* para garantizar la inmutabilidad de los registros académicos en un expediente estudiantil digital.

Para la implementación de la propuesta de una arquitectura distribuida se utilizaron servicios en la nube de AWS, que proporciona una nube privada, en la cual se desplegó la *blockchain* permisiva Hyperledger Fabric, se analizaron los protocolos de cifrado que incluyen funciones de *hash* SHA3 y firma digital ECDSA con llaves de 256 *bits* para el manejo de envío de transacciones, para adicionar a la red y el manejo de usuarios autorizados.

Asimismo, se realizó un *API REST* con Node JS y el *framework Express* para la interacción con la *blockchain* implementada, obteniendo tiempos de latencia muy bajos en el orden de 0.4 s en promedio, para peticiones de adición y consulta dentro de la red.

Luego de realizar simulaciones de carga y consulta de datos académicos en la red *blockchain* implementada se pudo apreciar que los tiempos de respuesta en peticiones de consulta y adición de datos son eficientes y no afectan el rendimiento de los sistemas de información ya implantados en la organización educativa que use esta arquitectura para ofrecer una capa intermedia de auditoría de sus expedientes académicos. También se pudo constatar que es computacionalmente costoso alterar bloques ya generados para adicionar datos falsos a la red *blockchain*.

Como resultado se determinó que los algoritmos de cifrado *SHA3* y *ECDSA* proporcionan un ecosistema de seguridad eficiente y de baja latencia en el marco de un plan de calidad de datos basados en ISO 14721:2012, la cual establece que cada bloque de la red almacenará únicamente datos relevantes que serán inmutables en el tiempo.

PLANTEAMIENTO DEL PROBLEMA Y FORMULACIÓN DE PREGUNTAS ORIENTADORAS

El expediente académico comprende una serie de cursos, diplomados, talleres u otras actividades como foros o investigación. Este historial se forma progresivamente. Cada vez que un logro de esta índole es alcanzado se agrega al expediente académico del estudiante y forma parte de su evolución académica.

Partiendo de la importancia que poseen los datos de los estudiantes dentro de las organizaciones públicas y de la necesidad de brindar los servicios a los que pueden optar de forma eficiente y rápida es de vital importancia garantizar que los datos que el sistema mantiene sean veraces y que su origen y actualizaciones puedan ser auditables. Con nuevos paradigmas como la computación en la nube, es aún más importante la definición de controles estrictos sobre la calidad de los datos incluyendo su origen, integridad, custodia y almacenamiento.

Los débiles controles sobre la custodia y manejo de los datos, así como las arquitecturas de sistemas utilizados propician diferentes puntos de fallo tanto para los servicios y en los datos de los estudiantes, lo que genera una reputación negativa al demostrar que no poseen estándares que garanticen el origen, veracidad e integridad de los datos almacenados.

En el caso particular de la Universidad de San Carlos de Guatemala, los expedientes académicos son formados por actas digitales y sus contrapartes físicas que las hacen verificables. Tanto el registro de notas, así como los comentarios sobre algún tipo de gestión administrativa queda registrada por su

solicitud digital y física. Sin embargo, los débiles controles para los procesos de negocio que gobiernan la recolección de los datos dan lugar a que la integridad, privacidad y origen de los mismos se vea comprometido a través de diferentes puntos dentro de soluciones *on-premise* como las implementaciones en arquitectura monolítica que predominan como implementación más favorable, la cual menos beneficia a la comunidad de estudiantes cuando los servicios que la universidad presta demuestran altos tiempos de espera y atención.

La evolución académica se maneja en sistemas aislados, por lo que la custodia y veracidad de los datos en un expediente académico es un reto constante. La variedad de información que es posible recabar, sin duda, es un tema de interés. La privacidad se vuelve una necesidad y el manejo de datos, cuya precisión y veracidad esté en duda, puede provocar conflictos legales, éticos y sociales. El uso de nuevos paradigmas como computación en la nube permiten que los sistemas sean más escalables y eficientes.

Sin embargo, detrás de estos expedientes estudiantiles debe existir una estrategia para el manejo de los datos que garantice la veracidad e integridad, pero que, a su vez, permitan al usuario que los genera, derechos sobre estos y la privacidad adecuada. Regulaciones como la General Data Protection Regulation (GPDR) aplicada a la privacidad de los datos de las personas que ha desarrollado la Unión Europea empodera al ciudadano, pues le concede derechos sobre la información que genera (Bennet y Webber, 2015). Empresas como Sony han incursionado en diferentes proyectos para el manejo de datos académicos, su solución se basa en tecnología que se aplica al campo de moneda electrónica, *blockchain*, con el fin de establecer un protocolo de intercambio de datos abierto (Comisión Europea, 2018).

Al mismo tiempo que las características de este tipo de almacenamiento distribuido promueven la privacidad del usuario, la veracidad de los datos y un medio eficiente para verificarlos (Sony Global Education, 2016). Existe también el caso en el que los títulos de grado de la Universidad de Nicosia, luego de completar todos los requisitos académicos y datos estudiantiles necesarios a través de una serie de cursos en línea, se han extendido de manera pública en la *blockchain* de *bitcoin* (Universidad de Nicosia, 2014).

La educación se está revolucionando y cada día se reinventan para encontrar nuevos modelos y arquitecturas de datos que se apeguen a un mundo cada día más conectado. Los datos académicos son el activo que más valor dan a una institución de este tipo. En los sistemas de control académico tradicionales, como los utilizados hasta ahora por la Universidad de San Carlos, se debe implementar nuevos protocolos de verificación y manejo que garanticen datos veraces e íntegros.

Por lo anteriormente mencionado surgen las siguientes interrogantes:

- Pregunta central
    - ¿Qué arquitectura de sistemas asegura la inmutabilidad en el tiempo de la información en un sistema de datos distribuido para la gestión de expedientes académicos?
- Preguntas auxiliares
    - ¿Qué protocolos de cifrado para el manejo de la información académica deben de ser implementados?

- ¿Cómo una arquitectura de base de datos basada en *blockchain* garantiza eficazmente mitigar el riesgo de datos académicos de origen dudoso?

- ¿Qué estructuras de datos o formatos de archivo deben considerarse para el desarrollo de un plan de aseguramiento de la calidad de los datos?

OBJETIVOS

General

Implementar una arquitectura distribuida que garantice la inmutabilidad en el tiempo de la información en un sistema de datos distribuido para la gestión de expedientes académicos.

Específicos

- Analizar los diferentes protocolos de cifrado para sistemas distribuidos basados en *blockchain* y determinar el más adecuado para el manejo de la información académica.

- Implementar una capa intermedia de auditoría de datos basada en *blockchain* para mitigar el riesgo de datos académicos de origen dudoso.

- Evaluar y comprobar las estructuras de datos o formatos de archivo, cuya complejidad y tamaño cumplan con el estándar ISO 14721:2012 para el desarrollo de un plan de aseguramiento de la calidad de los datos.

MARCO METODOLÓGICO

- Tipo de investigación

La investigación realizada fue de tipo explicativo, ya que con esta se pretendía comprender un poco más los diferentes campos en los cuales la tecnología de *blockchain* ha incursionado y las condiciones en las que esta se ha adoptado a diferentes sectores.

Fenómenos sociales como la reputación, confianza y la veracidad de la información que consumimos o que generamos seguirá siendo parte de las relaciones sociales y de los sistemas informáticos. Por tal razón, también esta investigación tuvo características de un estudio analítico ya que se recolectaron datos sobre el rendimiento y las características de usabilidad y seguridad que permitieron evaluar la eficacia con la que una arquitectura de registro de datos académicos puede ser manejada y los datos registrados sean inmutables en el tiempo.

- Diseño de investigación

El diseño del estudio es de tipo documental experimental. Los conceptos de *blockchain* han existido desde hace varios años en el sector financiero, lo que ha eliminado a las instituciones intermediarias en las transacciones financieras alrededor del mundo, sin embargo, aún existen retos para esta tecnología en otros ámbitos como la educación y los mismos se han tratado de forma teórica solamente.

Por lo que el diseño documental se orientó a la obtención de los antecedentes de implementaciones de *blockchain* en la educación que permitieron identificar la base idónea del modelo de arquitectura que se utilizó en el desarrollo del prototipo de esta investigación.

Con la experimentación que se realizó, luego de llevar a cabo el desarrollo e implementación del prototipo propuesto se logró validar la factibilidad del registro de datos académicos dentro de un expediente académico digital para determinar si existe un incremento en la confiabilidad de los datos en el tiempo y un decremento en los tiempos de atención al estudiante en sus operaciones académicas como certificaciones de notas, solicitudes administrativas y velocidad de comprobación de los datos por la entidad emisora que es la Escuela de Estudios de Postgrado de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala

- Procedimiento metodológico

A continuación, se detallan las fases realizadas para llegar concretar el desarrollo de la investigación del trabajo de graduación, estas fases describen en qué consisten y que instrumentos de recolección de información se utilizaron para la investigación:

- Fase I: revisión documental

Esta fase consistió en el uso de diversas fuentes como libros, revistas científicas, tesis y publicaciones para definir los conceptos que se utilizaron como base para el diseño e implementación de un prototipo de gestión de archivos para expedientes académicos digitales utilizando *blockchain*.

A través de esta revisión se obtuvo los antecedentes de los diseños de *blockchain* en el ámbito educativo para evaluar sus beneficios e implementarlas en el desarrollo del prototipo de este proyecto. Para lo cual se necesitó la investigación de los temas siguientes:

- *Blockchain*.
- Seguridad de las *blockchain*.
- Tipos de *blockchain*.
- Metodología para la implementación de una *blockchain*.
- Métricas de operación de una *blockchain*.

Asimismo, en el desarrollo del prototipo la unidad de datos a almacenar lo comprenden el expediente académico digital y las transacciones que se generan de estos, por lo que se revisaron publicaciones científicas, tesis, libros sobre:

- Manejo de archivos académicos.
- Estructura de un expediente digital.
- Manejo de expedientes digitales.
- Técnicas de compresión de archivos.

- Fase II: entrevistas

Esta fase consistió en el mapeo y obtención de los datos que la unidad de registro académico de la Escuela de Estudios de Postgrado utiliza para llevar el control académico de los estudiantes. Esto permitió establecer la estructura de datos necesaria para conformar el expediente académico digital considerada en las transacciones.

Para esta fase se llevó a cabo los siguientes pasos:

- Presentación del proyecto a la Escuela de Estudios de Postgrado de la facultad de Ingeniería de la Universidad de San Carlos de Guatemala, así como sus objetivos y alcances.
- Identificación de los actores técnicos y administrativos en el proceso de registro de datos académicos.
- Departamento informático.
- Autoridades administrativas.
- Evaluación de la brecha entre la arquitectura de sistemas actual y la propuesta en el presente proyecto, esta evaluación resultará en una lista de componentes de arquitectura que comprenderá lo siguiente:
    - Componentes existentes
    - Componentes reutilizables
    - Protocolos de comunicación
    - Protocolos de seguridad

Como resultado de esta fase se tuvo un inventario actual de los componentes de arquitectura de sistemas que utiliza el registro de datos académicos, la perspectiva de la parte técnica y administrativa sobre el manejo de los datos, así como la arquitectura de datos utilizada para el almacenamiento de estos. Se realizaron los siguientes diagramas como parte de esta fase:

- Diagrama de caso de uso de registro de estudiante, catedrático y actividad académica.
- Diagrama de paquetes del sistema de registro de estudiantes, catedráticos y actividad académica.
- Diagrama de actividades de registro de estudiantes, catedráticos y actividad académica.

- Documento de definición de objetivos y alcances del sistema de registro académico utilizado actualmente por la Escuela de Estudios de Postgrado.

- Fase III: análisis y diseño del prototipo

Esta fase comprendió el diseño de los módulos que integrarán el prototipo, basado en una arquitectura de microservicios que conformará un sistema secundario de registro de cada transacción académica realizada.

El objetivo de esta fase fue diseñar una arquitectura distribuida y escalable, que considere la arquitectura y el sistema que posee la Escuela de Estudios de Postgrado. Para iniciar con esta fase, fue necesario que la fase de entrevistas hubiera concluido.

El análisis del prototipo abarcó los puntos siguientes:

- Alcances de la arquitectura ya implementada en la Escuela de Estudios de Postgrado.
- Funcionalidades y operaciones ya ofrecidas en el sistema.
- Estructura de datos para el almacenamiento del perfil académico.
- Definición de las operaciones que se le presta al estudiante, catedrático y personal administrativo que involucran la creación, modificación y consulta de registros académicos.

El diseño del prototipo abarcó los siguientes módulos:

- Módulo de transacciones académicas.
- Módulo de usuarios (estudiantes, catedráticos y administrativos).
- Módulo de autenticación.
- Módulo de *ledger* académico.

Para lograr el objetivo de esta fase se generó un documento de especificación de arquitectura con las siguientes especificaciones:

- Definición del sistema, requerimientos funcionales y operativos.
- Descripción de la seguridad entre componentes de la arquitectura.
- Descripción de la arquitectura.
- Diagrama descriptivo de la arquitectura.
- Diagrama de componentes.
- Diagrama de despliegue.
- Descripción de casos funcionales.
- Diagrama de casos de uso.
- Diagrama de actividades.
- Descripción del modelo de datos.
- Diagrama entidad relación.
- Modelo de datos para el expediente académico.
- Descripción de matrices de riesgo y responsables en el desarrollo del prototipo.
- Descripción de los protocolos de comunicación y seguridad implementados.
- Definición de la interfaz de aplicación (*API*) expuesta por cada módulo.

- Fase IV: desarrollo e implementación del prototipo

En esta fase se llevó a cabo el desarrollo del prototipo con base a los requerimientos, herramientas de desarrollo y diseño resultado de la fase anterior.

Se utilizó una metodología ágil basada en técnicas de *scrum* para el desarrollo del prototipo teniendo como resultado un entregable funcional luego de cada iteración. Como resultado de esta fase se tuvo un cronograma de entregas de los componentes del prototipo garantizando con esto la coordinación requerida para la realización de pruebas, así como de retroalimentación necesaria para ajustar el desarrollo de este.

Esta fase contempló las siguientes actividades:

- Preparación de un ambiente de pruebas.
- Identificación del equipo técnico y administrativo que formará parte del soporte al desarrollo del prototipo por parte de la Escuela de Estudios de Postgrado.
- Desarrollo del *software* para los componentes de la arquitectura.
- Desarrollo de pruebas unitarias y de usuario para los componentes desarrollados.

- Fase V: experimentación

Con la fase de desarrollo concluida se realizó la experimentación de resultados con el sistema implementado en un ciclo académico y con todos los programas académicos que se ofrecen por parte de la Escuela de Estudios de Postgrado.

Esta fase se realizó en un ambiente de pruebas integrado para el desarrollo donde se realizó una limpieza de los datos de prueba utilizados durante la fase anterior.

Como objetivos de esta fase se tuvieron:

- Colocar el bloque inicial de datos académicos en el *ledger* central de la arquitectura.
- Analizar la factibilidad del proceso de migración de los datos de la arquitectura legada hacia la arquitectura del prototipo basada en *blockchain*.
- Probar las características de inmutabilidad proveídas por la arquitectura implementada.
- Monitorear los valores de las métricas operativas de la *blockchain* en la que basa su funcionamiento la arquitectura del prototipo.

Entre las actividades que formaron parte de esta fase se encuentran:

- Capacitación a los usuarios técnicos y administrativos de la Escuela de Estudios de Postgrado.
- Configuración inicial de los parámetros de la *blockchain*.
- Carga de datos de prueba individual (datos de los usuarios).
- Carga de datos académicos de forma masiva a través del *API* expuesto por la arquitectura.
- Monitoreo de las métricas de tiempo de respuesta, seguridad de la información y satisfacción del usuario.

- Fase VI: resultados

En esta fase se analizaron tanto cuantitativamente como cualitativamente los resultados obtenidos en la fase de experimentación. Como objetivo principal de esta fase se tuvo la evaluación de las características de manejo de datos que proporciona la arquitectura del prototipo basada en *blockchain* para los datos académicos de los estudiantes de la Escuela de Estudios de Postgrado y la misma institución con las operaciones académicas que más utilizan.

Las actividades que se contemplaron para esta fase fueron las siguientes:

- Análisis de los resultados de monitoreo de variables operativas de la arquitectura del prototipo resultado de la fase de experimentación.
- Análisis de la percepción del usuario ante la nueva arquitectura haciendo uso de encuestas electrónicas para el personal administrativo que haga uso del sistema, así como de la población estudiantil de los diferentes programas de maestría, especialización y doctorado ofrecidos por la Escuela de Estudios de Postgrado.
- Redacción de un informe final con los resultados obtenidos que incluye las siguientes secciones:
    - Diseño de la arquitectura.
    - Definición y diseño del prototipo.
    - Casos de uso.
    - Operaciones implementadas (*API*).
    - Nodos de almacenamiento intermedio.
    - Estructura del ambiente de pruebas.
    - Manual de uso y administración de la arquitectura.
    - Manual de instalación y despliegue de la arquitectura.
    - Conjunto de pruebas realizadas y sus resultados cuantitativos y cualitativos.

- Instrumentos de recolección de información

Para el desarrollo de nuevas soluciones es necesario el soporte documental como base en la cual se sustenten los principios que se quieren perseguir.

Para el desarrollo del trabajo de investigación se utilizarán las siguientes técnicas de recolección de información:

- Fuentes de información primaria
  - Datos académicos resultado del proceso de registro de las transacciones académicas dentro del prototipo de sistema implementado.
  - Encuestas electrónicas a usuarios (académicos, técnicos y administrativos) del prototipo de sistema propuesto a implementarse en la Escuela de Estudios de Postgrado y sus diferentes programas de especialización, maestría y doctorado.

- Fuentes de información secundaria
  - Artículos en revistas científicas sobre *blockchain* y su implementación en la educación.
  - Libros
  - Resultados de publicaciones científicas y en modelos abiertos de datos sobre experimentación de sistemas basados en *blockchain.*

INTRODUCCIÓN

La tecnología ha evolucionado constantemente, se ha buscado un gran número de mejoras como la capacidad de procesamiento, ordenadores más pequeños e incluso la posibilidad de llevar una vida en línea. Todo este abanico de sistemas y aplicaciones se ha basado en diferentes tipos de arquitecturas de sistemas como sistemas monolíticos, distribuidos, en la nube, microservicios, entre otros. Pero siempre ha sido necesario confiar en la entidad detrás de estos para que el concepto o el objetivo del sistema se cumpla.

¿Qué sucedería si fuéramos dueños de nuestra propia información?, debe existir una forma universal de identificarnos y que sea a través de esta que todos nuestros inicios de sesión y registros se fueran almacenando. Los registros académicos son un tipo de información del cual se puede partir, sin embargo, existe cierta dependencia con las entidades educativas ya que son estas las que se encargan de proveer la interfaz de confianza apropiada para que las métricas o notas de promoción sean válidas y reconocidas.

En 2009, el pseudónimo Satoshi Nakamoto cobró notoriedad debido a su publicación de una investigación realizada en el ámbito de lo que se conoce como *blockchain* o cadena de bloques. Un sistema de nodos punto a punto distribuido que a través de un método de consenso hace que los datos registrados sean inmutables en el tiempo.

Su aplicación eminentemente financiera en un inicio dio lugar a grandes especulaciones acerca del valor del dinero. Sin embargo, luego de varios años implementaciones similares como Ethereum han surgido y consigo nuevas

posibilidades de sistemas en los cuales se elimine a las empresas intermediarias y se puedan realizar transacciones punto a punto como en el caso de los registros académicos entre entidad y estudiante, teniendo este último total control sobre los datos que ha generado.

El presente trabajo de investigación se compone de seis capítulos los cuales describen las diferentes etapas de análisis y desarrollo seguidas para el desarrollo de una arquitectura basada en *blockchain* para un sistema de gestión de expedientes académicos digitales.

En el capítulo uno se describen los antecedentes en los cuales se fundamenta el problema planteado y como las soluciones de *blockchain* se adaptan a la educación para permitir datos inmutables en el tiempo.

En el capítulo dos se justifica el desarrollo del presente trabajo de investigación dada la necesidad de crear arquitecturas de sistemas descentralizadas que garanticen la inmutabilidad de los datos académicos.

En el capítulo tres se define el alcance que tendrán las actividades del presente trabajo de investigación desde tres perspectivas. En la perspectiva investigativa se definirá las ventajas de aplicar una capa intermedia de validación para el origen de los datos así como los protocolos de cifrado utilizados en *blockchain* para almacenar una estructura de datos que permita gestionar la calidad de los expedientes académicos.

Desde el punto de vista técnico se diseñará e implementará una capa intermedia para validar los expedientes estudiantiles y su origen definiendo para esto los servicios de la *blockchain* Ethereum que permitan cifrar consultar y operar los mismos. Y desde la perspectiva de resultados se obtendrá un prototipo

de protocolo de comunicación para una arquitectura distribuida basada en *blockchain* que permitirá guardar expedientes académicos en almacenamiento secundario de forma confiable e inmutable en el tiempo.

En el capítulo cuatro se presenta la teoría necesaria para comprender la tecnología que será el fundamento de la arquitectura a implementar y que apoyará el desarrollo del presente trabajo.

En el capítulo cinco se presenta la arquitectura basada en *blockchain* desarrollada para la gestión de expedientes académicos que garantizará la inmutabilidad de estos en el tiempo y proporcionará una forma rápida y confiable de verificar los mismos.

En el capítulo seis se presenta la discusión de los resultados de la implementación de la arquitectura basada en *blockchain* propuesta para la gestión de los expedientes académicos, las condiciones en las cuales esta se puede aplicar, los beneficios que presenta así como los inconvenientes que puede ocasionar.

1. ANTECEDENTES

Los datos son el bien con mayor valor en los sistemas de hoy en día, sean estos manejados en la nube o en arquitecturas *on-premise* (monolítica, cliente-servidor, distribuida, en capas) son la entidad mínima que da lugar a la información y a los sistemas que la manejan. Particularmente, los datos educativos son de vital importancia en sectores como la educación superior, la empresa privada, entidades públicas, así como en toda organización alrededor del mundo. Los datos educativos avalan los logros, así como la experiencia y dominio de conceptos y tecnologías a través de títulos, certificaciones, diplomas, entre otros.

Por esta razón, ha surgido el interés de replantear los sistemas de manejo de información para los expedientes estudiantiles que aún se manejan de forma física o en sistemas centralizados monolíticos que no proporcionan las condiciones de seguridad y privacidad necesarias para preservar los datos en el tiempo, *blockchain* (cadena de bloques) ha sido una herramienta ampliamente difundida en la economía a través de la criptomoneda. Sin embargo, sus aplicaciones se han empezado a expandir hasta el campo de la educación, sin ser aún propuestas concretas.

La sociedad aún es escéptica sobre los beneficios que aporta, pero la transaccionalidad del proceso educativo y su posición dentro de los negocios basados en la confianza y reputación de la institución lo hace una tecnología ideal. Confianza que se compromete al presentar credenciales falsas, o falsas habilidades.

La privacidad y los estándares por los cuales los datos deben regirse contemplan la transparencia de la información en el tiempo y que no pueda modificarse una vez validada. Iniciativas de *blockchain* para la educación como *Learning is Earning* que intenta registrar estudios sobre una materia a través de un *edublock* que equivale a 1 hora de aprendizaje pudiendo ser este de origen formal (universidades, centros de estudios especializados, empresas) e informales como charlas, reuniones, seminarios han sentado las bases para crear historiales transparentes y rápidamente verificables. La extensión de certificados es un área en la cual *blockchain* ha incursionado de forma constante usando la firma de certificados con el registro de estos en una *blockchain* (Watters, 2016).

En la era de la información, la nube juega un papel fundamental. Mucho del procesamiento de datos se realiza en ella, ya sea implementaciones de nube privadas o en servicios como AWS o Azure. Pero aún con estos servicios al alcance las entidades no hacen uso de ellas para almacenar información personal, la preocupación sobre el manejo y gobernanza de los datos es una prioridad. De 2012 a 2015 el sistema educativo del estado de Nueva York en Estados Unidos de América realizó un proyecto para llevar los registros de los estudiantes. Datos académicos, demográficos, de salud y personales hacia una implementación de nube privada.

Los objetivos, ser eficientes con el manejo de la información y realizar análisis de datos para ajustar de forma adecuada los métodos utilizados para la entrega de conocimiento en base a los resultados de los estudiantes. Sin embargo, el mismo tuvo serios inconvenientes debido a la privacidad de los datos, los servicios de almacenamiento en la nube utilizados debido a sus políticas podían acceder a la información y analizar la misma. La comunidad estudiantil, padres de familia, juntas locales de educación analizaron la situación y el proyecto se descontinuó debido a las diferentes visiones que cada parte

involucrada tenía sobre el acceso, manejo y almacenamiento de la información en la nube (Bennet y Webber, 2015).

La extensión de certificados digitales es el campo con mayor desarrollo de *blockchain* dentro de la educación. MIT Media Lab en 2015, dentro del Instituto Tecnológico de Massachusetts inició con proporcionar certificados digitales a través de una infraestructura de llaves pública y privada haciendo uso de *blockchain* para la autenticación de las personas y el registro de las contribuciones de valor generadas para la institución. Esto generaba una reputación positiva para los usuarios que ostentaban los certificados digitales y los empodera para compartir dicha información de forma segura, anónima y rápida con posibles empleadores, otras instituciones educativas y colegas (Schmidt, 2015).

El modelo educativo ha cambiado, la educación ha evolucionado de clases magistrales como el método tradicional y más divulgado entre la educación superior al uso desde los últimos años de cursos masivos en línea *MOOC*, y seguirán desarrollándose plataformas como *Coursera, Udemy, Udacity o EdX* (Online Course Report, 2016). La industria de la educación al igual que la financiera se respalda en la confianza de las instituciones intermediarias que validan que los bienes y sus poseedores son confiables.

En las plataformas digitales para la educación es un punto primordial y en sistemas que hacen uso de *blockchain* se tiene que superar el reto de la forma en la que se mantiene la privacidad de los datos y la identidad, para esto, el uso de criptografía asimétrica con una infraestructura de llaves privadas y públicas garantiza que los datos almacenados sean privados. A través de esta infraestructura y almacenamiento la validación de los datos se realiza de forma

rápida y eficiente, permitiendo un intercambio abierto y seguro de la información académica (Tapscott, 2017).

Se han realizado propuestas de arquitecturas para el manejo de los registros académicos desde diferentes sistemas de manejo de educación como EdX o Coursera, cuya información a través de implementaciones de contratos inteligentes permitiría registrar nuevas instituciones educativas, almacenando cada logro de los estudiantes o información de estos como un hash dentro de la *blockchain*, los contratos inteligentes han demostrado ser una herramienta eficiente para el manejo de privilegios, accesos y procesos de negocios que deben cumplirse garantizando la privacidad, inalterabilidad (Ocheja, Flanagan y Ogata, 2018).

Sin duda alguna las aplicaciones de *blockchain* se irán expandiendo en la educación superior, así como todas sus formas digitales existentes, los certificados firmados digitalmente y confiables serán la llave del éxito profesional debido a los bajos costos de generación e incremento en la confiabilidad de la entidad que los emite.

Llegará el momento en que la movilidad académica se realice en cuestión de minutos con iniciativas de datos académicos abiertos usando *blockchain* como la propuesta por Sony, Blockchain Archive (Sony Global Education 2016), aunque muchas otras como OpenBadges de la fundación Mozilla, Blockcerts del MIT Media Lab o Learning is Earning 2026 seguirán operando y será necesario generar estándares para la seguridad y manejo de la información y el intercambio abierto de la misma a través de nubes públicas.

2. JUSTIFICACIÓN

El presente trabajo tiene como propósito contribuir en el área de administración de tecnologías de la información en la línea de investigación de dispositivos y sistemas para incrementar la seguridad al utilizar tecnologías de la información y comunicación.

Con la aplicación de los conceptos de cadena de bloques (*blockchain*), una red distribuida que permite almacenar datos sin la necesidad de una entidad centralizada, toda la información es almacenada de forma precisa en el tiempo, la inmutabilidad y privacidad es garantizada haciendo que los nodos de la red verifiquen los datos agregados, siendo posible modificaciones posteriores, pero no elimina datos previamente almacenados. Por lo que se identifica una oportunidad en el uso de *blockchain* como medida de seguridad al manejo de la información estudiantil donde la comunidad de estudiantes forma parte activa en garantizar la privacidad y veracidad de su información académica haciendo posible que la verificación de esta sea rápida y eficiente.

Los sistemas de control de información estudiantil basados en documentos físicos no son confiables, el tiempo de procesamiento es alto y sin los controles de verificación adecuados en la información no existe credibilidad. Esta alternativa para el manejo de la información estudiantil busca combinar las características de un sistema distribuido, inmutable en el tiempo y en la cual la confiabilidad de los datos no sea dada por un solo nodo, sino un conjunto de ellos.

Debido a las limitantes de los sistemas de gestión y almacenamiento que hacen uso de sistemas de base de datos relacional centralizados los datos de los expedientes estudiantiles pueden ser alterados por usuarios con acceso a los datos y esto puede dar lugar a casos en los que se generan títulos universitarios sin que los estudiantes hayan siquiera estado en las aulas. Otra limitante son los procesos administrativos al momento de verificar un expediente estudiantil, pues al ser un conjunto de hitos alcanzados durante un período de tiempo promedio de 5 años, los datos se corroboran en actas físicas y estas a su vez no son fiables ya que pudieron sufrir alteraciones en el tiempo.

Los estudiantes y catedráticos se beneficiarían de este sistema, al hacer uso de *blockchain* para la implementación del sistema de gestión del expediente estudiantil las notas serían verificadas de forma pronta y veraz, teniendo como consecuencia que los catedráticos, estudiantes e institución educativa construyan una reputación con alto índice de credibilidad basado en la confiabilidad de los datos aumentando la eficiencia de los procesos administrativos que concluirán con el seguimiento de tareas de titulación soportadas por la seguridad con la que se manejaron los datos que rectifican los logros académicos alcanzados por el estudiante.

3. ALCANCES

3.1. Resultados

- Prototipo de un protocolo de comunicación y estructura de datos para una arquitectura distribuida basada en *blockchain* para el intercambio de datos académicos hacia almacenamiento secundario.
- Prototipo de sistema de gestión de los expedientes académicos, implementado en una arquitectura distribuida con una capa intermedia de validación de datos utilizando *blockchain Ethereum*.
- Definir e implementar un protocolo de comunicación de datos conforme la norma ISO 14721:2012.

3.2. Técnicos

- Mostrar las ventajas de aplicar una capa intermedia en una arquitectura distribuida como mecanismo de validación del origen y variación de los datos.
- Analizar los protocolos de cifrado utilizados por *blockchain* como herramienta para garantizar la inmutabilidad y origen de los datos del expediente académico en el tiempo.
- Evaluar que componentes se requieren para construir una estructura óptima y eficiente para cubrir la norma ISO 14721:2012 que permita gestionar procesos de calidad y confiabilidad de los datos de un expediente académico digital.

3.3. Investigativos

- Diseñar e implementar una capa intermedia para la validación de los expedientes académicos usando *blockchain* para generar una cadena inmutable de transacciones en el tiempo.
- Definir e implementar los servicios *REST* que integren el uso de un protocolo de comunicación de cifrado de datos necesarios para la consulta y operación de las transacciones de los expedientes académicos dentro de la *blockchain*.
- Adecuar los servicios de la *blockchain* Ethereum hacia la gestión de los expedientes académicos y perfiles de estudiante y unidades académicas en estructuras de datos trazables apegadas a la norma ISO 14721:2012.

4. MARCO TEÓRICO

4.1. Sistemas de manejo de base de datos

La información que genera todo sistema informático puede ser susceptible a ser almacenada, dependiendo de los niveles de granularidad los datos pueden ser de valor crítico o de importancia menor. “Los sistemas de manejo de bases de datos se refieren a una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos” (Silberschatz, Korth y Sudarshan, 2002, p. 01).

Estos sistemas a su vez proveen una forma conveniente y efectiva para definir esquemas, concurrencia, seguridad, manejo, control para compartir datos y restauración en caso de siniestros (Gill, 2011).

4.2. Tipos de arquitecturas de sistemas

La estructura de las diferentes arquitecturas de sistemas varía en función de las necesidades del problema y describen una solución de componentes desde los puntos de vista físico y lógico.

4.2.1. Arquitecturas *on-premise*

El término *on-premise* es ampliamente utilizado para describir todo sistema desarrollado por una entidad con el propósito de utilizarlo dentro de la organización. Estos sistemas según sus objetivos pueden ser construidos desde diferentes puntos de vista. El término arquitectura define la estructura o

estructuras de un sistema que comprende a todos los elementos de *software*, las propiedades externas de estos que son visibles y las relaciones entre ellos (Cervantes, Kazman, 2016).

Existen diferentes estilos de arquitectura, estos proporcionan una definición clara de los principios organizacionales para un sistema que han sido probados y bien entendidos previamente. Además, al proporcionar una estructura genérica ampliamente utilizada permite entender las características más importantes que la conforman de forma sencilla y rápida. Estos estilos de arquitectura usualmente se definen en términos de los elementos de arquitectura que lo componen, las restricciones existentes entre estos y las interfaces externas disponibles. Logrando visualizar la forma en que los elementos se distribuyen armoniosamente como un todo (Rozansky, Woods, 2005).

4.2.2. Monolítica

Es el estilo de arquitectura más básica que se puede seguir, los componentes funcionales de las diferentes estructuras identificadas son altamente acoplados y engloban los diferentes aspectos funcionales del sistema (presentación, procesamiento y almacenamiento de datos). Las arquitecturas basadas en este estilo no poseen flexibilidad al cambio debido a su alto grado de acoplamiento y su estructura interna no definida. Su mantenimiento, implantación, ampliación y seguridad de los datos son sus principales desventajas (Cervantes, Kazman, 2016).

4.2.3. Cliente – Servidor

Estilo de arquitectura que comprende dos tipos de elementos principales, un servidor que provee uno o más servicios a través de una bien definida interfaz

externa y un cliente que hace uso de los servicios expuestos como parte de su operación. El cliente y el servidor habitualmente residen en diferentes terminales dentro de una red. Su principal ventaja es la centralización del procesamiento de información importante o sensible. Y su principal desventaja es el punto único de falla de su modelo único punto de entrada (Rozansky, Woods, 2005).

4.2.4. Distribuida

Un estilo de arquitectura de este tipo es un sistema de procesamiento de información que contiene un numero independiente de nodos que cooperan entre sí y se comunican a través de una red para alcanzar un objetivo específico (Puder, Römer y Pilhofer, 2006).

Los sistemas de arquitectura distribuida ofrecen mayor relación costo-beneficio que las arquitecturas monolíticas, debido a que los nodos que conforman la arquitectura pueden ser terminales con especificaciones estándar. La introducción de redundancia incrementa la disponibilidad en caso de que algún punto de la arquitectura falle. Permitiendo un fácil manejo de fallas y de adición de funcionalidad lo cual provee mejor escalabilidad comparado con las arquitecturas centralizadas.

Este tipo de arquitectura posee alto grado de escalabilidad, apertura a estándares, calidad en el servicio y heterogeneidad entre componentes, por lo que se hace necesario un componente intermedio que coordine el trabajo en conjunto de los nodos que se tienen dentro de la arquitectura para gestionar este aspecto subyacente de los nodos, redes, lenguajes de programación y sistema operativo (Coulouris, Dollimore y Kindberg, 2000).

4.2.5. En capas

Esta arquitectura reconoce un solo elemento de sistema. La capa de arquitectura, y lo que hace es organizar en forma de pila las diferentes capas que conforman la arquitectura del sistema.

Cada capa provee servicios a la capa inmediatamente superior y puede solicitar servicios a la capa inmediatamente inferior a ella. Cada capa posee un nivel de abstracción y el orden en la pila es en base a este. En este estilo de arquitectura las capas pueden ser reutilizadas y la abstracción de cada capa a un objetivo en específico le da una gran independencia para que su mantenimiento y escalabilidad sean eficientes gracias a su implementación (Rozansky y Woods, 2005).

4.3. Microservicios

Newman describe el término microservicio como “servicios lo suficientemente pequeños y autónomos que trabajan uno con otro para lograr un objetivo”.

Este concepto hace uso del gran desacople entre servicios para brindar sistemas de alta disponibilidad debido a que pueden utilizar tecnologías heterogéneas, proporcionan resiliencia en su funcionamiento, son fácilmente escalables y desplegables de forma independiente. Esto conduce a una mejora de la productividad al permitir una alineación efectiva entre la arquitectura de software a implementar y la cultura organizacional (Newman, 2015).

4.4. Blockchain

La definición de *blockchain* o cadena de bloques depende del punto de vista desde el cual se estudie. Técnicamente, *blockchain* es un libro contable distribuido entre varios nodos, inmutable y criptográficamente seguro dentro del cual solo se puede añadir o actualizar información a través del consenso de estos (Bashir, 2017).

Desde una perspectiva de negocios *blockchain* se define como una plataforma en el cual los nodos pueden intercambiar bienes y valores a través de transacciones que no necesitan de un intermediario centralizado confiable (Bashir, 2017).

4.5. Principios de *blockchain*

A continuación se explican los principios que se deben tener en cuenta en el diseño de una *blockchain*.

4.5.1. Hashing

Las funciones de *hash* son utilizadas para crear una cadena de salida resumida de una cadena de texto arbitrariamente larga. Se utilizan para proveer integridad de datos a los servicios bajo tres características de seguridad que deben de cumplir: compresión arbitraria de un mensaje hacia una cadena resumida de longitud constante, fácil de calcular y resistencia a colisiones (Bashir, 2017).

4.5.3. *Bitcoin* y Ethereum

Bitcoin es un conjunto de conceptos y tecnologías que forman un ecosistema distribuido *peer-to-peer* de dinero virtual. El intercambio de valor entre los participantes de la red *bitcoin* se consigue a través de las unidades monetarias llamadas "*bitcoins*" y se registran en un libro contable público "*blockchain*". Estos *bitcoins* se crean mediante un proceso llamado "minería", basado en una competencia por encontrar soluciones a un problema matemático complejo que utilizará los recursos del sistema para hallar una solución al mismo tiempo que se procesan transacciones (Antonopoulus, 2015).

Ethereum al igual que *Bitcoin* es una red distribuida *peer-to-peer* de dinero virtual. La unidad monetaria de intercambio de valor es el "*ether*" y posee la capacidad de crear contratos inteligentes para la *blockchain* y aplicaciones descentralizadas (Bashir, 2017).

4.5.4. Bloques

Es el activo más importante, a través de estos se almacenan las transacciones y su estado. El nodo inicial se conoce como nodo génesis, todos los nodos subsecuentes se encuentran relacionados de forma criptográfica haciendo muy costoso cambiar el estado de la cadena y garantizando de esta forma la inmutabilidad. Los bloques son creados por servicio de órdenes y validados y almacenados por cada nodo.

Los bloques se encuentran formados comúnmente por la siguiente estructura:

- Encabezado
  - Número de bloque, entero que inicia con 0 (bloque génesis) e incrementado en 1 cada vez que se agrega un bloque a la cadena.
  - *Hash* de bloque actual, hash de todas las transacciones contenidas en el bloque.
  - *Hash* del bloque anterior.
- Datos: almacenados en forma de un conjunto de atributos clave-valor.
- Metadata: esta sección del bloque contiene el certificado y la firma del creador del bloque que será usado para validarlo en la red. En cualquiera de los casos, esta sección contiene adicionalmente un indicador de validez.

4.6. Tipos de blockchain

Las *blockchain* han evolucionado con el desarrollo de nuevos casos de uso a los cuales aplicar la tecnología, y se han dividido para satisfacer necesidades específicas.

4.6.1. Públicas

Como describe Bashir (2017) este tipo es abierta al público y cualquier persona puede participar como un nodo en el proceso de decisión. Conocidas también como *permission-less blockchain*, cada nodo mantiene una copia de los registros de forma local y haciendo uso de un método de consenso distribuido se logra alcanzar una decisión en cuanto al estado final de la *blockchain*.

4.6.2. Privadas

Como su nombre lo indica son *blockchains* restringidas a un grupo de individuos u organizaciones. También se tienen de tipo semiprivadas, que poseen su parte privada como se describió previamente y su parte pública la cual es abierta a que cualquiera participe. Una variante más dentro de este grupo son las completamente privadas o propietarias, cuyo objetivo de descentralización se ve comprometido por la necesidad de compartir y proveer garantía de que los datos generados son auténticos (Bashir, 2017).

4.7. Métodos de consenso

Bashir (2017) define el consenso como un concepto de computación distribuida que ha sido utilizado en *blockchain* para proporcionar los medios para acordar una única versión de lo que es auténtico entre los nodos que conforman la red. Existen dos tipos comunes de mecanismos de consenso que son:

- Basado en tolerancia a fallo bizantino, en los cuales no se realizan operaciones de cómputo intensivas, sino que se basa en un esquema simple de nodos que publican mensajes firmados constantemente.
- Basado en un líder o en una prueba, en los cuales se requiere de nodos que compitan por ser elegidos líder dentro del conjunto de nodos que conforman la red. El nodo que gana la competencia es el que propone el estado final.

4.7.1. Prueba de trabajo

Este mecanismo PoW *(Proof of Work)* se basa en la necesidad de gastar una cantidad suficiente de recursos computacionales previo a proponer un valor

aceptable por la red con el fin único de disuadir las acciones maliciosas en la red de nodos (Bashir, 2017).

4.7.2. Prueba de participación

Este mecanismo PoS *(Proof of Stake)* se basa en la idea de que un nodo o usuario tiene más poder en la medida que tenga mayor inversión o intereses dentro de la red. El concepto de *coin age* también es importante ya que con mayor *coin age* mayor es la probabilidad de firmar los siguientes bloques de la cadena en la red (Bashir, 2017).

4.7.3. Prueba de importancia

Este mecanismo no solamente confía en el nivel de interés que un usuario tiene en el sistema, también monitorea el uso y los movimientos de los *tokens* de los usuarios para establecer un nivel de confianza e importancia de cada uno de ellos (Bashir, 2017).

4.8. Hyperledger Fabric

Es un proyecto de Linux Foundation creado en 2015 apoyado por IBM y Digital Assets para promover el uso de la tecnología *blockchain* a través de distintos sectores industriales. Es una *blockchain* de carácter privado cuyo enfoque es empresarial ya que permite realizar transacciones privadas.

Esta implementación de *blockchain* se diferencia de *bitcoin* o *ethereum* ya que es una red privada y autorizada, en lugar de un sistema abierto y sin permisos que permite que usuarios desconocidos o ajenos a la red puedan intentar agregar nuevos bloques a la cadena. En lugar de esto, los miembros de una *blockchain*

en Hyperledger Fabric se inscriben a través de un proveedor de servicios de membresía (MSP) autorizado y confiable dentro de la red.

4.8.1. Modelo de un Fabric Ledger

Los siguientes elementos describen las características clave que permiten a *Hyperledger Fabric* ser una solución empresarial, pero al mismo tiempo personalizable.

- Activo: la definición de los activos dentro del modelo permite el intercambio de elementos con valor dentro de la red.
- *Contrato* inteligente *(chaincode)*: es la representación empaquetada de un contrato inteligente en Hyperledger Fabric. Es código que se ejecuta desde aplicaciones externas a la *blockchain* que administra el acceso y modificación a un conjunto de pares clave-valor en el historial de estados de la *blockchain* a través de una transacción.
- Privacidad: el estado de cada activo puede ser compartido en un canal común a todos los pares de la red. Sin embargo, llegado el momento por la necesidad empresarial se pueden crear canales específicos para grupos de participantes específicos.
- Seguridad y servicios de membresía: todos los participantes poseen una identidad conocida. Cada nodo posee un certificado criptográfico generado a través de infraestructura de clave pública, cada uno ligado a un actor dentro la red. Con esta noción de “autorizada” los controles de acceso a datos pueden manejarse de forma general o específicamente a nivel de cada canal. Permitiendo manejar diferentes escenarios en donde la privacidad la confidencialidad son clave.

4.8.3. Tipo de nodo en hyperledger fabric

Committing Peer: cada nodo dentro de un canal es de este tipo. Recibe bloques de las transacciones generadas que son posteriormente validadas antes de ser aceptadas y almacenados al final en la copia de la cadena de estados que posee cada uno de estos.

Endorsing Peer: se denomina así a todo nodo que tenga un contrato inteligente instalado en él. Sin embargo, para que realmente lo sea el contrato debe ser usado por un cliente y generar una respuesta digitalmente firmada.

Leader Peer: cuando una organización tiene múltiples nodos en un canal el nodo líder es el que toma responsabilidad para distribuir las transacciones hacia los *commiting peers* de la red.

Anchor Peer: este nodo es útil cuando se desea que los nodos de múltiples organizaciones puedan comunicarse entre ellos. Es definido en el canal de cada organización y pueden existir cero o más nodos de este tipo.

4.9. Arquitectura de *blockchain*

El patrón de diseño de la arquitectura de *blockchain* da importancia a tres factores importantes: seguridad, descentralización y trazabilidad de los datos. Dadas estas necesidades se evalúan los siguientes conceptos.

4.9.1. Arquitectura *peer-to-peer*

Este tipo de arquitecturas son esencialmente de tipo distribuido, su fundamento es la interconexión de nodos individuales independientes que

comparten sus recursos computacionales entre ellos sin un ente regularizador. Todos los nodos en el sistema realizan las mismas tareas, actúan como proveedores y consumidores de los recursos y servicios proveídos dentro de la red. Sin embargo, existen también arquitecturas *peer-to-peer* centralizadas que mantienen un nodo central para facilitar la interacción entre ellos y darles mantenimiento a los servicios disponibles, aunque esta última variante no es común en la implementación de esta arquitectura (Drescher, 2017).

4.9.2. Autenticación de usuarios y seguridad en las transacciones

La autenticación de usuarios es realizada a través de criptografía, haciendo uso de la criptografía simétrica en la cual se posee una llave para el cifrado, esta se utiliza tanto para el cifrado como el descifrado del mensaje.

Otro método de cifrado utilizado es el de criptografía asimétrica. En esta se tienen dos llaves, una pública y una privada. Los mensajes cifrados con una de estas llaves solamente pueden descifrarse con la otra llave y a la inversa (Drescher, 2017).

4.10. Contratos inteligentes

Szabo (2018) describe los contratos inteligentes como un protocolo de transacción computarizada que ejecuta los términos de un contrato. Su objetivo general es satisfacer los términos contractuales que tengan de común acuerdo los subscriptores minimizando las acciones maliciosas, accidentales y el uso de intermediarios.

Sin embargo, Bashir (2007) define un contrato inteligente como un programa de computadora seguro e imparable que representa un acuerdo ejecutable de forma automática.

4.11. Características de verificación de identidad y propiedad

Las personas dentro de las *blockchain* pueden utilizar su información real o pseudónimos que puedan ser mapeados de forma única, garantizando con esto el derecho a ser propietarios sin servicios de terceros como Facebook y Google. Los frentes en los cuales *blockchain* realiza esfuerzos para la autenticación en ocasiones requieren de nuevo hardware, o de soluciones integrales *B2B* y *programas* cada vez más refinados. Las lecturas biométricas están siendo consideradas para el proceso de autenticación. Aunque como en todo ámbito el cambio de hábitos es un factor crucial en la adopción de este método descentralizado (Mougayar, 2016).

4.12. Condiciones para la implementación de una *blockchain*

Drescher (2017) sugiere una ruta a considerar para identificar e implementar una *blockchain* dentro de una red distribuida de nodos *peer-to-peer.* Como punto inicial de la ruta es definir la titularidad, se refiere a que se puede definir como bienes de los cuales somos titulares dentro de la red y con los cuales se ejecutarán las transacciones.

Este conjunto histórico de transacciones identificará al titular actual. Definida la titularidad de los datos dentro de la red el paso siguiente es proteger la titularidad, y se refiere a prevenir que las personas puedan acceder a los bienes de otras dentro de la red. Este paso tiene tres aspectos importantes que son: la autentificación e identificación de los titulares, la restricción de acceso y el uso

de *hashing* para garantizar los pasos anteriores. Con la protección adecuada de los datos se procede a pensar en almacenar los datos de transacción en una estructura de datos llamada libro contable dado que son el elemento fundamental para determinar la titularidad.

Los libros contables dentro de la red se deben de preparar para ser distribuidos entre los nodos dentro de un ambiente con una figura de autoridad descentralizada, lo que significa que existirán copias en la red del libro contable que deben garantizar la inmutabilidad de las transacciones en el tiempo. Este sistema de control para la distribución de los libros contables dentro de la red de nodos además de controlar la distribución de estos controla la adición de nuevos bloques de transacciones verificando estás a través de un método de consenso previamente establecido.

4.13. Implementación de *blockchain* en la educación

Durante mucho tiempo se han realizado intentos por aplicar el concepto de *blockchain* a diferentes sectores, encontrando en la educación un entorno más eficiente para el desarrollo de nuevos conceptos e ideas que saquen provecho a los beneficios de su aplicación.

4.13.1. La función crucial de la educación en el desarrollo de *blockchain*

Melanie Swan, una teórica destacada de las *blockchain* explica por qué el mejor lugar para educar sobre esta tecnología es sobre la misma *blockchain*. Para que una publicación científica de este tipo sea aprobada y publicada lleva meses.

Sin embargo, liberar este tipo de documentos dentro de la *blockchain* permite una distribución inmediata, recibir críticas en tiempo real y ganarse el crédito necesario o reputación para enfocarse en un público más amplio (Tapscott, 2016).

4.14. Manejo de propiedad intelectual

David Byrne de Talking Heads ha resumido la situación de la propiedad intelectual en internet como un negocio insostenible como medio de apoyo al trabajo creativo, cualquiera que sea su tipo.

En cada punto de la cadena de distribución a través de canales digitales se observa que cada vez son menos las regalías para los productores de este tipo de bienes. Sin embargo, *blockchain* plantea que los artistas sean protagonistas del modelo de negocio, eliminando intermediarios. De manera que no sólo puedan ejercer su derecho de expresión, sino también beneficiarse de él, maximizando el valor del interés moral y material que despierte su propiedad intelectual (Tapscott, 2016).

4.15. Manejo de portafolios digitales

La tecnología *blockchain* y su implementación *bitcoin* y contratos inteligentes de *ethereum* han abierto posibilidades de un nuevo mercado en donde la reputación es parte de cada transacción.

Han proporcionado de características como: plantillas de contratos que respetan al artista a carta cabal como creación de valor, derechos de autor inclusivos que reparten las ganancias equitativamente, registros transparentes distribuidos para que todo el mundo pueda ver la contabilidad de determinados

portafolios, posibilidad de micro contabilizar para distribuir los ingresos de forma inmediata, bases de datos completas que se puedan comunicar entre sí y vinculen los materiales sujetos a derechos de autor en un registro digital que todos pueden ver, un sistema de reputación que permita crear a los artistas así como a sus posibles socios su propia credibilidad (Tapscott, 2016).

5. PRESENTACIÓN DE RESULTADOS

5.1. Análisis de la *blockchain*

A continuación se describe el proceso de análisis realizado para la implementación de la solución.

5.1.1. Selección del tipo de *blockchain*

Las operaciones de almacenamiento y consulta de datos académicos se han regido a un modelo de datos relacional que de forma tradicional respeta las propiedades ACID de un sistema de gestión de base de datos relacional. Sin embargo, las arquitecturas de almacenamiento de datos académicos actuales se pueden considerar silos de información que son validados en la existencia física de actas que respalden los datos.

Para la selección del tipo de *blockchain* se juzgaron las siguientes características:

- Propósito
- Modo de participación de los nodos
- Mecanismo de consenso
- Capacidad de ejecutar contratos inteligentes

Y se consideraron los aspectos siguientes para la arquitectura de sistema de base de datos existente para el manejo de datos académicos:

- Modelo de datos relacional
- Datos almacenados en base de dato relacional

Los datos se validan con su archivo físico firmado y sellado por autoridad competente.

Tabla I. **Comparación de características de implementaciones de *blockchain***

Característica	**Hyperledger Fabric**	**Ethereum**	***Bitcoin***
Propósito	B2B	B2C, criptomonedas	B2C, criptomonedas
Modo de participación	Privada	Privada, pública	Pública
Mecanismo de consenso	PBFT	PoW	PoW
Algoritmo de *Hash*	SHA3 – SHAKE256	Ethash	SHA-256
Lenguaje de programación	Go, Node Js, Java	Solidity	N/A

Fuente: elaboración propia.

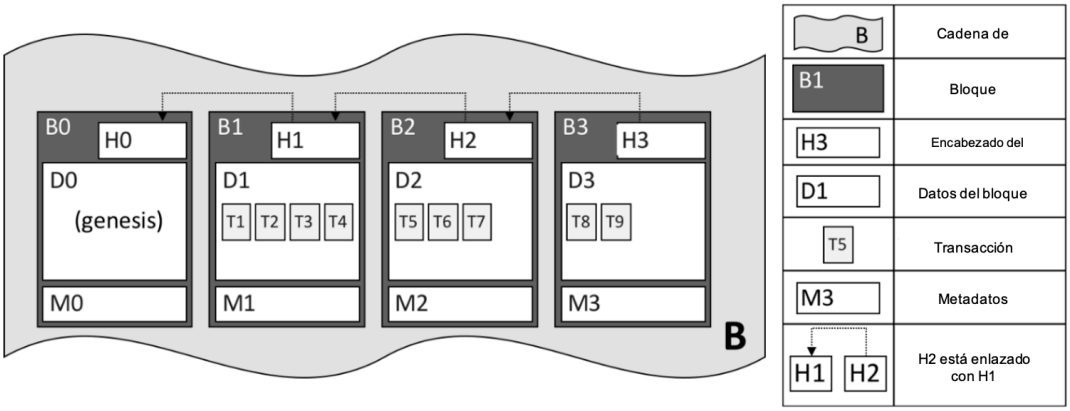
5.1.2. Estructura de un bloque

La definición de la estructura de un bloque supuso identificar aquellos valores y elementos de datos que permitieran identificar transacciones determinísticas. Es decir, que no importando que nodo las evalúe el resultado será siempre el mismo por lo cual siempre se logrará el consenso por parte de los nodos. Una característica inherente a la implementación de los protocolos y mecanismos de *blockchain*.

La selección de los datos que conforman cada bloque se seleccionó con base en la estructura general de un bloque teniendo los siguientes grupos de datos:

- Encabezado del bloque
- Datos del bloque
- Metadata del bloque

Figura 1. **Estructura de la *blockchain* basa en hyperledger fabric**

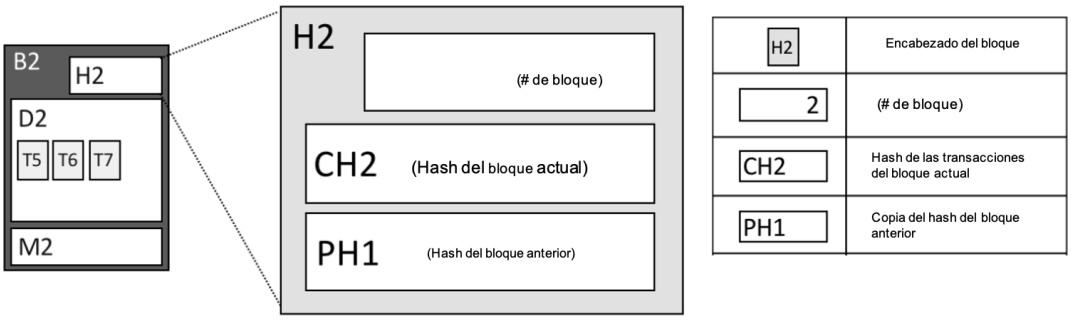


Fuente: *Hyperledger Fabric Read the Docs*. Consultado el 25 de junio de 2020. Recuperado de https://hyperledger-fabric.readthedocs.io/en/release-2.0/ledger/ledger.html#blocks.

La *sección de encabezado* de cada bloque define la información que permite darle trazabilidad al bloque en la cadena. Entre los datos que conforman la sección se definieron los siguientes:

- Número de bloque, definido como un número entero que inicia con 0 siendo este el bloque inicial en la cadena o bloque génesis. Se incrementa en 1 unidad cada vez que se agrega un bloque nuevo a la cadena.
- *Hash* del bloque actual, es el resultado de aplicar una función criptográfica de hash a los datos contenidos en bloque actual.
- *Hash* del bloque anterior, es el *hash* obtenido del bloque anterior y que permitirá anexar el nuevo bloque a la cadena manteniendo la referencia al bloque anterior.

Figura 2. **Estructura de un bloque basado en hyperledger fabric.**



Fuente: *Hyperledger Fabric Read the Docs*. Consultado el 25 de junio de 2020. Recuperado de: https://hyperledger-fabric.readthedocs.io/en/release-2.0/ledger/ledger.html#blocks.

En la figura 2 se observa la composición genérica de un bloque. La sección de datos de cada bloque define los datos que almacenará la cadena. Para la propuesta de arquitectura y prototipo se ha considerado almacenar datos de cada

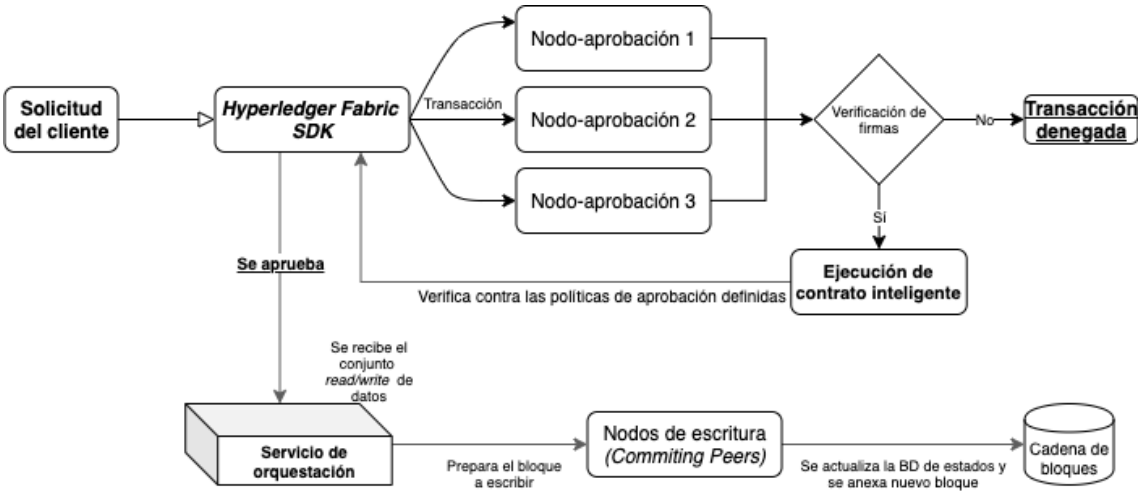
curso y datos de los resultados obtenidos por cada estudiante respetando la lógica del modelo de datos relacional aplicado a la información académica.

La sección de metadatos contiene la firma del creador del bloque que es usada para validar el mismo a través de los demás nodos de la red. Los nodos que validan la transacción adicionan también un indicador de validez/invalidez del bloque y contiene un *hash* acumulativo de la cadena hasta ese punto para detectar posibles alteraciones a los datos.

5.1.3. Selección del método de consenso

Dada la necesidad de garantizar que un dato no pueda alterarse en el tiempo y el planteamiento de una arquitectura distribuida. Surge el requerimiento dentro de la solución propuesta y la selección de *blockchain*. ¿Cómo garantizar el consenso dentro de los nodos y quienes deberían de ser los nodos que aprueban las transacciones? Limitados a los métodos implementados por la solución de *blockchain* seleccionada *Hyperledger Fabric*, tiene como premisa que existe una confianza parcial entre los nodos ya que no es una red pública sino más bien una red privada y permisiva que limita el acceso de quienes pueden originar una transacción, realizar actualizaciones a cambios y consultar datos.

Figura 3. **Flujo de consenso**



Fuente: elaboración propia.

En la figura 3 se puede observar el flujo generalizado que siguieron las transacciones dentro del prototipo y se entiende de la siguiente forma:

- La aplicación cliente envía una nueva transacción.
- El *SDK* nativo de *Hyperledger Fabric* genera la propuesta de la transacción recibida y envía hacía los nodos de aprobación para su validación.
- Los nodos de aprobación validan que:
  - La firma sea válida.
  - No sea una transacción previamente enviada.
  - El cliente tenga permisos para ejecutar la transacción solicitada
  - Los datos y estructura estén bien formados.
- Los datos de la propuesta de transacción son pasados a un contrato inteligente para su manejo.
- EL *SDK* nativo verifica luego la firma de los nodos de aprobación coincidan y que el resultado sea el mismo.

- La aplicación hace un *broadcast* de los conjuntos de datos *read/write* y las firmas para el nuevo bloque al servicio de orquestación.
- Los nodos de organización de la cadena reciben los datos del nuevo bloque y los envían hacia todos los nodos.
- Cada nodo dentro de la red agrega el nuevo bloque a la cadena y actualiza el estado de la base de datos.

Por tal razón la selección del método de consenso no se basó en pruebas de trabajo o pruebas de estado como lo hacen implementaciones de *blockchain* públicas aplicadas a criptomonedas como *bitcoin* o *ethereum* sino más bien basado en la premisa de confianza existente dentro de la red privada considerando los siguientes aspectos: aprobación (*endorsement*), gestión de ordenes (*ordering*), validación (*validation*).

5.2. Análisis de protocolos criptográficos en *blockchain*

Las implementaciones de *blockchain* evaluadas utilizan algoritmos criptográficos para su funcionamiento. Desde los bloques almacenados como los mensajes intercambiados en la red.

5.2.1. Funciones *hash*

En la tabla I se realizó una comparación de las diferentes características de *blockchain*. Uno de ellos fue el algoritmo utilizado para la implementación. Cuya fortaleza a ataques y seguridad se basa en que sean funciones de una sola vía y que sean resistentes a colisiones. Lo primero garantiza que no pueda ser invertida y lo segundo que no existan dos valores de entrada para el mismo valor de salida de la función.

La elección del algoritmo de *hash* usado en la arquitectura debe su importancia a que cada bloque genera un *hash* con base en las transacciones que empaqueta y la referencia al *hash* del bloque anterior. En las funciones de *hashing* SHA y Ethash la complejidad de romper la propiedad de ser una función de una sola vía supone un orden de $O(2^k)$ $\text{donde k} = \text{ tamaño de la llave}$ esto en el supuesto de un ataque de fuerza bruta lo que indica una dificultad muy grande y que al más mínimo cambio el *string* resultado de la función hash sea completamente diferente como se muestra en la figura 4.

Figura 4. **Cálculo de hash para una palabra**

```
SHA - 256
Palabra Clave: ARBlockchain. (12 Bytes)
Resultado: 9131c2667a0c1ec9b0254179914fadf71a54bffd59b7c7dc9badd311dd4386b8

SHA - 256
Palabra Clave: ARBlockchain. (13 Bytes)
Resultado: b1e25bbf1206f7003180a2607e871d0e8a45a75da99c096dcecf0c52716c09b7
```

Fuente: elaboración propia.

Como se observa al más mínimo cambio la salida de la función mantiene su longitud de 64 bytes, pero el resultado es completamente diferente y con esto garantiza que no existe el mismo resultado para dos palabras diferentes en el dominio de la función.

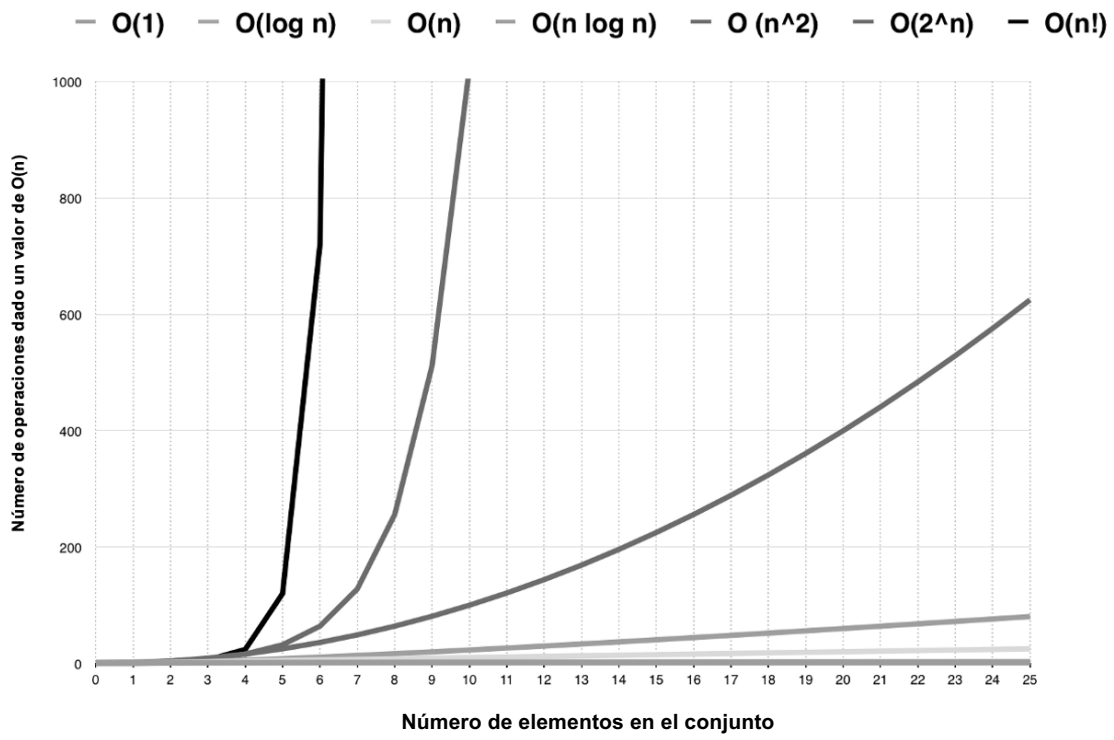
Del mismo modo como se observa en la figura 5, SHA3 SHAKE256 obtiene un *hash* de 64 *bytes* de longitud, pero la implementación del algoritmo al ser diferente proporciona valores distintos a los resultados de la figura 4.

Figura 5. **Cálculo de hash para una palabra**

Fuente: elaboración propia.

Por lo que ambas funciones son suficientemente seguras por la complejidad exponencial encontrada en la función $O(n)$. Se puede observar de la figura 6 que la complejidad se incrementa con el tamaño de la llave por lo que al utilizar una llave de 256 bits podríamos garantizar la seguridad necesaria para las transacciones y por esa razón *SHA3 SHAKE256* fue el algoritmo de *hash* seleccionado para la implementación.

Figura 6. **Ejemplo de comportamiento en O(n)**



Fuente: Yangani, K. *A Begnners Guide to Big O Notation*. Consultado el 25 de junio de 2020. Recuperado de: https://medium.com/free-code-camp/my-first-foray-into-technology-c5b6e83fe8f1.

5.2.2. Certificados de firma digital

Los certificados digitales tomaron un papel muy importante dada la selección de una *blockchain* privada y la implementación de una capa intermedia que permitiera almacenar un log auditable cuyos registros pudieran ser insertados solo por nodos autorizados o con los privilegios administrativos suficientes para generar propuestas de transacciones confiables hacia la red.

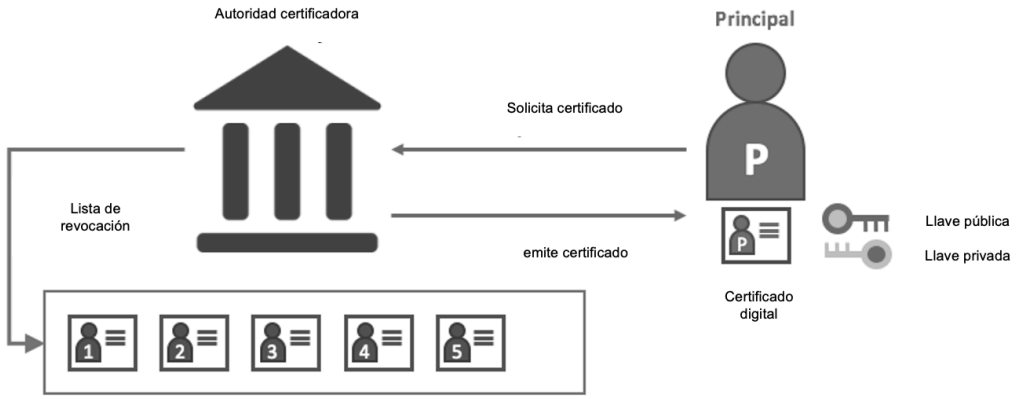
Hyperledger Fabric a través de sus componentes nativos brinda una autoridad certificadora propia o una integración simple con entidades terceras para la emisión de certificados digitales. Para la propuesta de solución para la capa intermedia de auditoría se decidió utilizar *Hyperledger Fabric CA Client* que es la entidad emisora de certificados por defecto. La selección de una entidad interna y propia para la implementación respondió a los siguientes atributos:

- El costo de utilizar el cliente *CA* de *Hyperledger Fabric* no constituía gasto para la emisión de certificados.
- El cliente es capaz de emitir certificados X.509 a través del algoritmo *ECDSA* y llaves de tamaño 256, 384 y 512 *bits* respectivamente. Siendo un estándar criptográficamente fuerte.

Con esta entidad definida dentro de la arquitectura como el proveedor de membresías *(MSP)* e incluido por defecto en *Hyperledger Fabric* se puede garantizar una identidad verificable dentro de la red haciendo uso de los certificados emitidos para ese fin.

El algoritmo de firma digital utilizado ECDSA es un algoritmo que hace uso de la arquitectura de llave pública-privada dentro de la red, y por la dificultad que ofrece para suplantar la identidad de los nodos certificados se utilizó como medio de verificación de identidad para consulta de la información, así como medio de identidad para las transacciones enviadas para su adición a la *blockchain*.

Figura 7. **Flujo de enrolamiento de una nueva identidad en la CA**

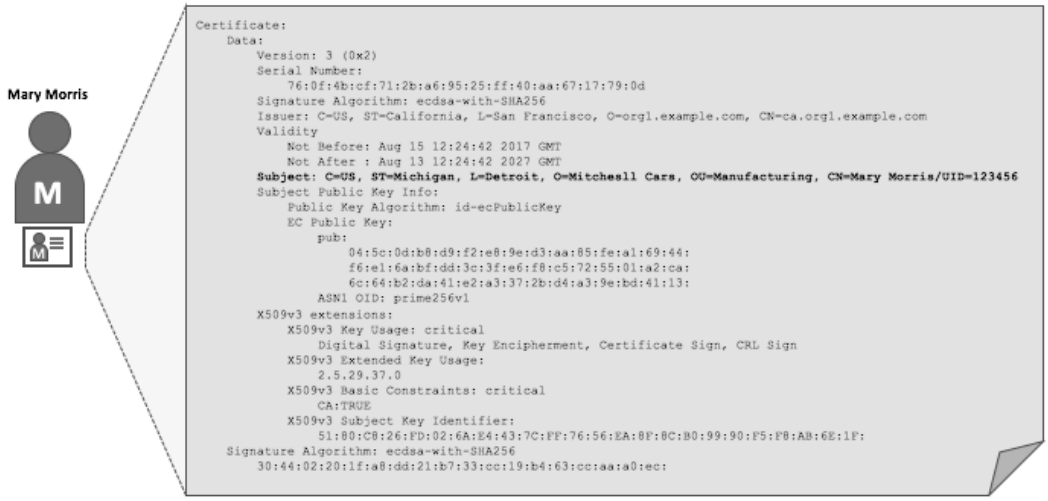


Fuente: *Hyperledger Fabric Read the Docs*. Consultado el 25 de junio de 2020. Recuperado de: https://hyperledger-fabric.readthedocs.io/en/release-2.0/identity/identity.html.

En la figura 7 se observa que la petición para generar un certificado de firma lo puede realizar cualquier nodo de la red. Sin embargo, el certificado extendido puede ser de identidad (ECert para consultas) y de transacciones (Tcert para envío de peticiones de adición). El procesar la solicitud en este caso del principal la *CA* evalúa si puede o no emitir un certificado para la solicitud realizada.

La política de emisión puede responder a diferentes niveles, desde validaciones de correo electrónico como validaciones de puesto, dirección, documentos de identificación que corroboren exactamente la identidad a enrolar, se puede observar también una lista de revocación que es como el típico concepto de lista negra y le indica a la *CA* que las entidades ahí definidas no pueden obtener un certificado por una u otra razón. Luego de que la política de emisión es cumplida la *CA* extiende el certificado correspondiente.

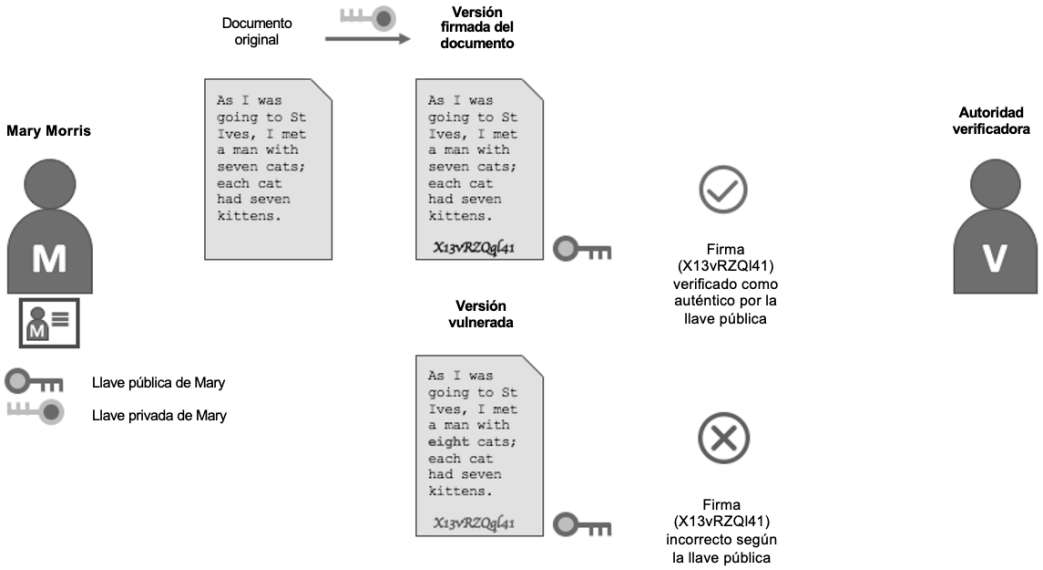
Figura 8. **Estructura de certificado X.509 por cliente CA de Hyperledger Fabric**



Fuente: *Hyperledger Fabric Read the Docs*. Consultado el 25 de junio de 2020. Recuperado de https://hyperledger-fabric.readthedocs.io/en/release-2.0/identity/identity.html.

La figura 8 muestra los atributos que normalmente se pueden encontrar en un certificado X.509. Se especifica el algoritmo de cifrado utilizado, la entidad emisora (*CA*), fechas de validez, la entidad o nodo a la que se le emite y su correspondiente llave pública con la cual firma las transacciones enviadas.

Figura 9. **Autenticación vía PKI**



Fuente: *Hyperledger Fabric Read the Docs*. Consultado el 25 de junio de 2020. Recuperado de https://hyperledger-fabric.readthedocs.io/en/release-2.0/identity/identity.html.

De la figura 9 se puede observar como la utilización del certificado de firma digital garantiza que los mensajes enviados cumplen con la integridad necesaria para enviar transacciones dentro de la red. Todo lo que se firma con la llave privada del emisor puede ser validado con la llave pública asociada. Al más mínimo cambio del mensaje en el medio la validación no se puede completar y por tal razón el receptor puede verificar la integridad del mensaje.

En este tipo de protocolos la complejidad radica en custodiar de forma correcta la llave privada con la cual se cifran las comunicaciones ya que si esta se ve comprometida es preciso generar un par de llaves pública-privada nuevamente para así garantizar que la confidencialidad e integridad de los mensajes enviados por el nodo se mantenga.

Dentro de la propuesta de solución para una capa intermedia de auditoría de datos basada en *blockchain* se tuvo presente la necesidad de garantizar que los datos no puedan ser alterados en el tiempo. Esta característica conocida como inmutabilidad se logró incluir en el diseño de la solución dado que la implementación de *blockchain* utilizada soporta la utilización de protocolos de cifrado fuertes en términos computacionales como es SHA y en protocolos de autenticación como lo es certificados digitales. La elección de una llave de 256 *bits* para el algoritmo SHA 3 utilizado en el *hash* de los bloques se tomó debido a que previo a que una transacción pueda ser enviada a la red el nodo que la envía debe enviarlo firmado con su certificado.

Y si este no se encuentra registrado, la configuración de políticas de seguridad de la *CA* no permitirá la transacción por no ser enviada por una entidad conocida y autorizada para ese fin.

5.3. Diseño e implementación de la arquitectura

Con base en las mejores prácticas para sistemas de alta disponibilidad se consideraron los factores: rendimiento, contratos inteligentes y almacenamiento. Estos fueron potenciados por el objetivo de poder mantener la trazabilidad de los registros académicos y de las identidades de los nodos con las cuales interactúan y garantizar su validez.

5.3.1. Definición de los componentes de la arquitectura

El diseño e implementación de la arquitectura propuesta se llevó a cabo bajo el supuesto de poseer una arquitectura transaccional ya establecida y que realiza las transacciones siguientes en el sistema:

- CRUD de estudiantes.
- CRUD de catedráticos.
- CRUD de cursos.
- Carga de zona, examen final y total para cada estudiante registrado en cada curso.
- Lectura de información académica (filtrada por: estudiante/catedrático/escuela).
- Los componentes dentro del alcance del prototipo de arquitectura son:
    - *Blockchain* manejada – *Hyperledger Fabric* en AWS, componente que suministró un servicio permisivo de nodos con accesos segmentados hacia usuarios autorizados con escalabilidad flexible.
    - Contrato inteligente de la *blockchain*, que permitió ejecutar validaciones con base en los datos recibidos, almacenados y ejecutar reglas de negocio automatizadas (ej. Monitoreo de Tx dudosas, intentos de actualización de información sin privilegios, consulta sin privilegios, entre otros).
    - *API Rest* – Red TxPrimaria, a través de esta se expuso las diferentes interacciones CRUD hacia la *blockchain*.

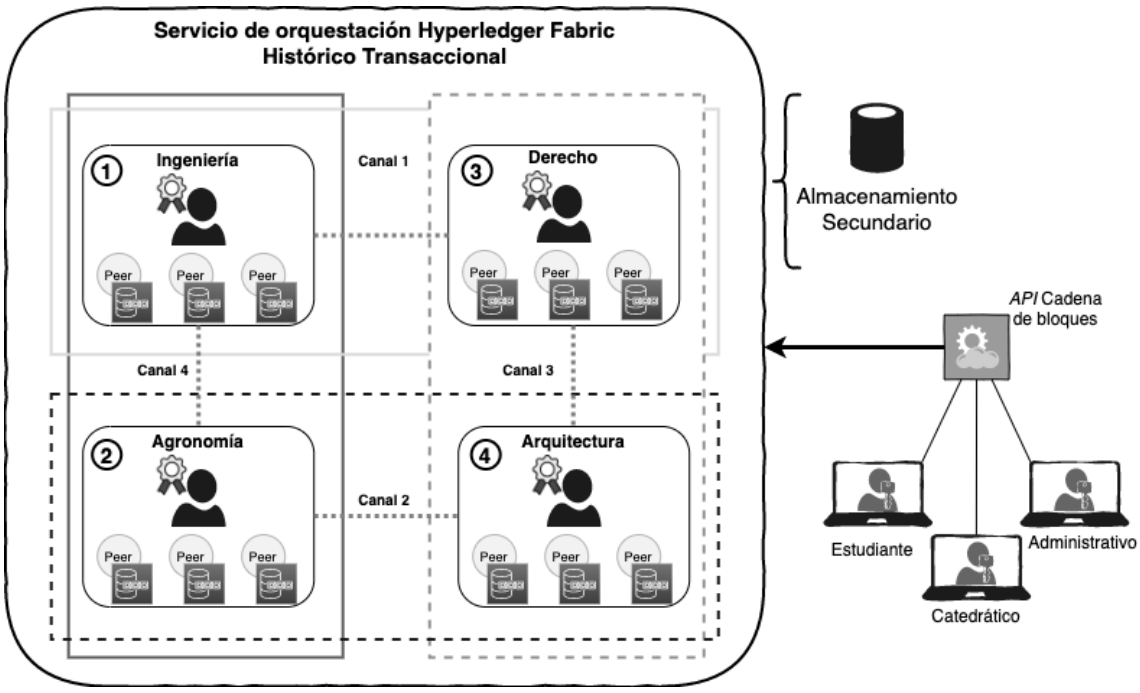
Definición de los documentos utilizados en la capa intermedia de comunicación *blockchain* – aplicación, *blockchain* – API, *blockchain* – Base de datos de estados.

5.3.2. Diagrama de arquitectura

En la figura 10 se muestra la propuesta de arquitectura utilizada. Los componentes principales en orden de prioridad son los siguientes:

- Servicio de orquestación (canales, nodos)
- Autoridad certificadora *(CA)* de cada grupo de nodos
- *API* de la cadena de bloques

Figura 10. **Diagrama de arquitectura**

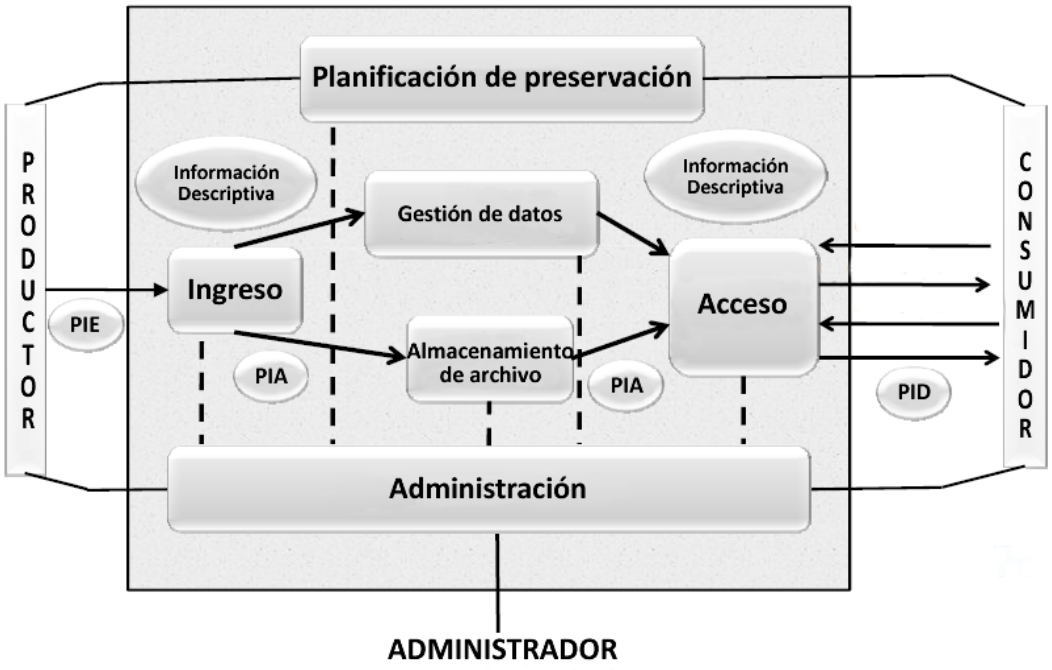


Fuente: elaboración propia.

La gobernanza de los datos e información generada por la entidad educativa se basa en lo propuesto por la norma ISO 14721:2012. La figura 11 ejemplifica el modelo funcional para un sistema de almacenado de datos basado en esa

norma. Cobra especial relevancia la comunicación en ambas vías desde el productor de la información a través de la ingesta de paquetes de información hacia el sistema de manejo de datos que provee de servicios y funcionalidades para mantener y acusar a los paquetes de información por parte de un ente consumidor. El modelo además considera la importancia que los accesos a los datos e información tienen dentro de un sistema que preserva información relevante a través del tiempo.

Figura 11. **Modelo Funcional *ISO14721:212* OAIS**



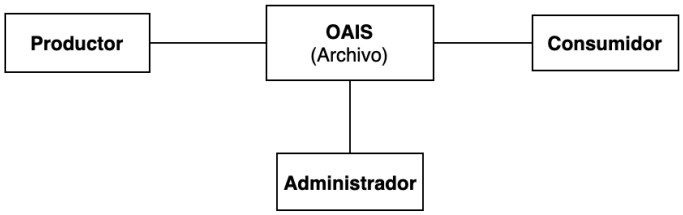
Fuente: The consultative committee for space data systems. *Modelo de referencia OAIS*. Consultado el 26 de junio de 2020. Recuperado de https://bit.ly/3AOJzMY.

5.3.3. Análisis de un plan de calidad de datos basados en ISO 14721:2012

La preservación de los datos permite manejar de forma eficaz la metodología con la cual se reciben datos de los productores y se ponen a disposición los posibles consumidores. Al igual que muchos de los sistemas de manejo de archivos. OAIS plantea el uso de diferentes módulos que garantizan un correcto archivado de datos.

La norma ISO 14721:2012 especifica las bases para un sistema de archivado de datos integral teniendo como base la unidad básica IP *(Information Package)*. En la figura 12 se observa el diseño de alto nivel de un sistema de preservación de archivos OAIS.

Figura 12. **Modelo de un ecosistema OAIS**



Fuente: The consultative committee for space data systems. *Modelo de referencia OAIS*. Consultado el 26 de junio de 2020. Recuperado de https://bit.ly/3AOJzMY.

El modelo de ecosistema OAIS se ve complementado con la implementación de la capa intermedia para auditoría de datos propuesta en la solución. Misma que puede incluirse en el componente de gestión como parte de

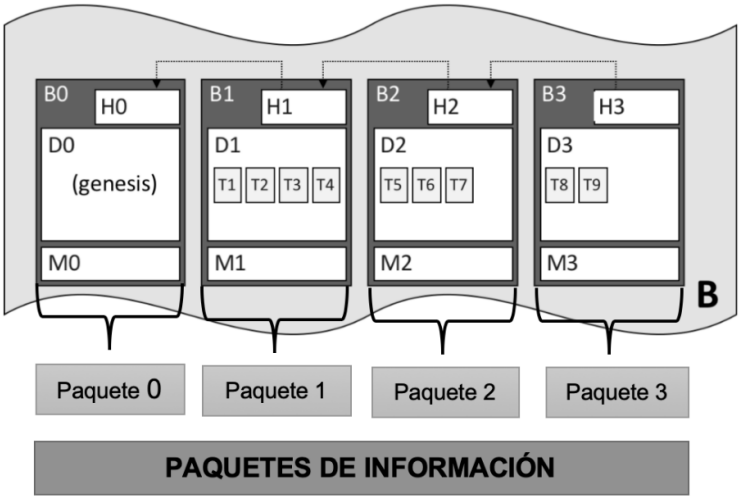
las políticas de acceso y el aseguramiento de que la información remitida para su archivado no ha sido alterada de ninguna manera desde que se originó.

5.3.4. Identificación de los datos

Los datos académicos juegan un papel importante en todos los sentidos de una institución educativa. Se generan, se procesan, se publican y como fin permiten a los que ostentan esos registros el logro de objetivos académicos como un diploma, un título o un grado académico.

Para utilizar ISO 14721:2012 se debe definir con claridad los datos tanto físicos como digitales que serán sujetos de archivado y el proceso de validación, procesamiento y publicación respectivo para garantizar que los datos recibidos y expuestos sean correctos. El que un dato sea correcto significa que el origen puede ser auditable así como el proceso por el cual el dato se obtuvo. Esta característica en los registros académicos de un expediente académico digital se cubre con el uso de la capa intermedia de auditoría de datos dentro de la arquitectura propuesta en este trabajo.

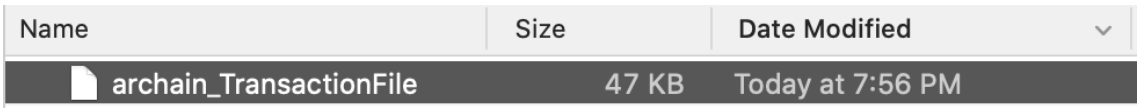
Figura 13. **Conceptualización de un *IP* basado en arquitectura propuesta y OAIS**



Fuente: elaboración propia.

La figura 13 ilustra lo que un paquete de información puede mapear dentro del modelo OAIS. Si bien lo contenido en cada bloque no resulta relevante para ser archivado, le proporciona al medio físico o digital el respaldo de que no han sufrido alteraciones en el tiempo. Así que puede mapearse hacia la metadata que describe el paquete de información siendo viable para un ejemplo de 100 registros de transacciones obtener una parte de metadata de 47Kb que se muestra en la figura 14.

Figura 14. **Archivo de 100 transacciones de 472 *Bytes***



Fuente: elaboración propia.

Cada paquete de información almacena metadata, esta es utilizada para generar un *hash* del bloque actual que será utilizado para preservar la cadena haciendo que el último nodo conocido de la red guarde la referencia de que nodo es el siguiente y el nodo actual la referencia de cuál es el nodo anterior. En este caso el *hash* del bloque actual en el cual se encuentran los datos se apega a la estructura de datos de la figura 15.

Figura 15. **Modelo de estructura de datos para la auditoría de registros académicos**


```
{
   "docType":"arAllocation",
   "arAllocationId":"c5b29e938a19a80c225d30e8327caaf817f76aecd381c868263c4f59b45dgf72-1",
   "arAllocationZona":51,
   "arAllocationExFinal":25,
   "arAllocationDate":"2019-09-25T15:31:29.582Z",
   "arAllocationDescription":"Ingreso Nota Ordinaria - Seminario I",
   "studentId":"FFF6A68D-DB19-4CD3-97B0-01C1A793ED3B",
   "teacherId":"FFF6A48D-DB59-4DD3-97B1-01B1A493E23C",
   "courseScheduleId":"5",
   "ADSRegistrationNumber":"D0194B30-685D-499E-A9CD-2C6DCE5GAG48",
   "arId":"1234"
}
```


Fuente: elaboración propia.

5.3.5. Gestión de los datos dentro de un sistema ISO 14721:2012

La gestión de accesos juega un importante rol en un plan de calidad de los datos, ya que los consumidores esperan obtener datos coherentes, correctos y veraces. Estos atributos pueden ser difíciles de medir y por lo tanto las métricas o definiciones de que es un dato de calidad dependerá de los objetivos que se deseen alcanzar. La capa intermedia para auditoría de datos de la arquitectura propuesta permite cumplir con la característica de veracidad.

Y lo hace a través del registro en la red *blockchain* del conjunto de datos asociado a un estudiante, catedrático, curso y estos a su respectiva unidad académica.

El plan de calidad de datos prioriza la capacidad de validar si un dato es confiable. Y si está registrado en alguna transacción de la red *blockchain* simplemente lo es. El protocolo de cifrado utilizado garantiza que el flujo de datos seguido en la figura 11 se cumpla. Los productores de datos no pueden proveer de insumos sin validación al sistema de almacenamiento, esto porque cada transacción de registro de datos académicos queda registrada con información concerniente al evento que se ejecuta. Si por alguna razón el evento digital requiere de una contraparte física se podrá garantizar que este insumo físico pueda verificarse a través de los protocolos de cifrado implementados en la capa intermedia de auditoría basada en *blockchain*.

Los módulos de planificación para la preservación, administración y acceso que se pueden apreciar en la figura 11 quedan fuera de la planeación para la calidad de los datos dado que no preservamos el registro histórico físico o digital, sino que brindamos un medio auditable de que los datos ahí estipulados son auténticos y válidos.

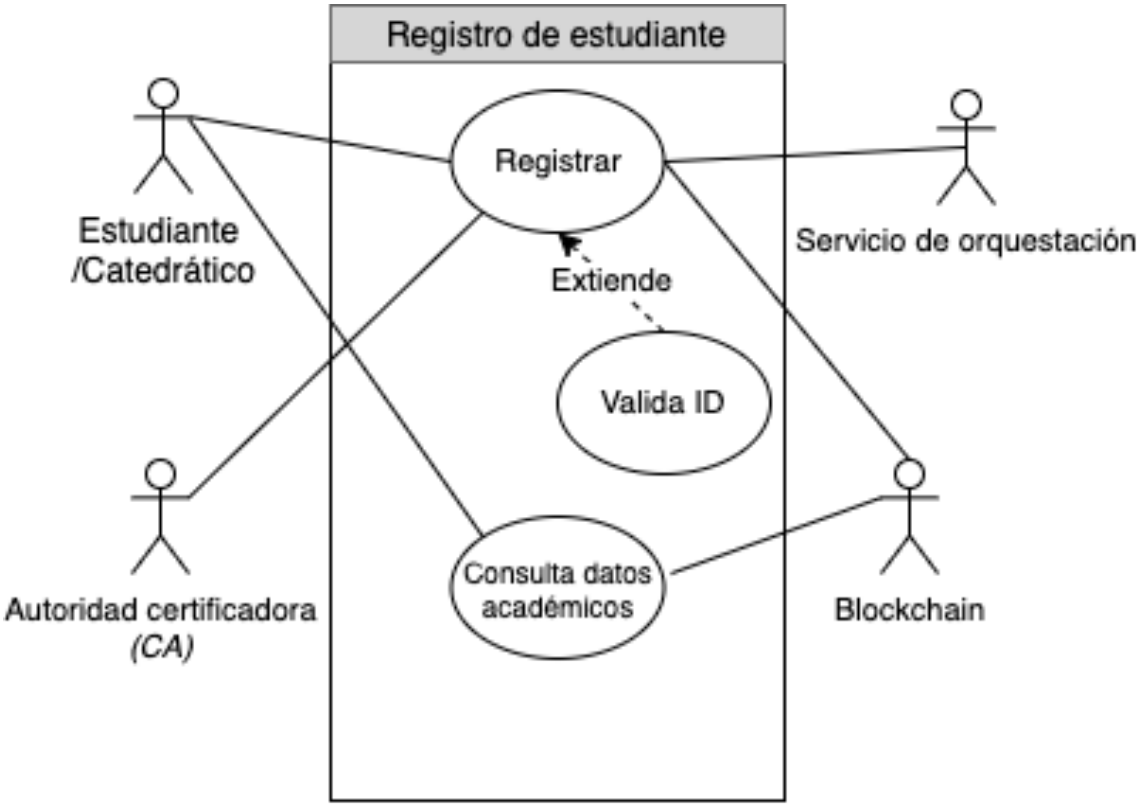
5.3.6. Casos de uso

Para el presente trabajo en donde se validó la auditoría de las transacciones que pudiesen afectar los datos académicos se definieron los casos de uso para manejo de la entidad estudiante, catedrático, curso y las operaciones relacionadas con estas. Los sistemas actuales de manejo de datos académicos pueden variar de acuerdo con su implementación y las necesidades propias de cada unidad académica.

5.3.6.1. Registro de participante

A continuación, en las figura 16 y tabla II se presenta el caso de uso de la creación de un nuevo participante. Este puede ser un estudiante, catedrático o personal administrativo.

Figura 16. **Caso de uso para el registro de nuevo participante**



Fuente: elaboración propia.

Los usuarios de tipo estudiante podrán únicamente enrolarse para consultar y compartir sus datos académicos.

Los usuarios de tipo catedrático podrán enviar peticiones para registrar la información correspondiente a una nueva entidad de notas específicas a un curso o actualizar una entidad existente previamente por algún error de tipo humano o de sistema.

Tabla II. **Definición de caso de uso "Registrar nuevo participante"**

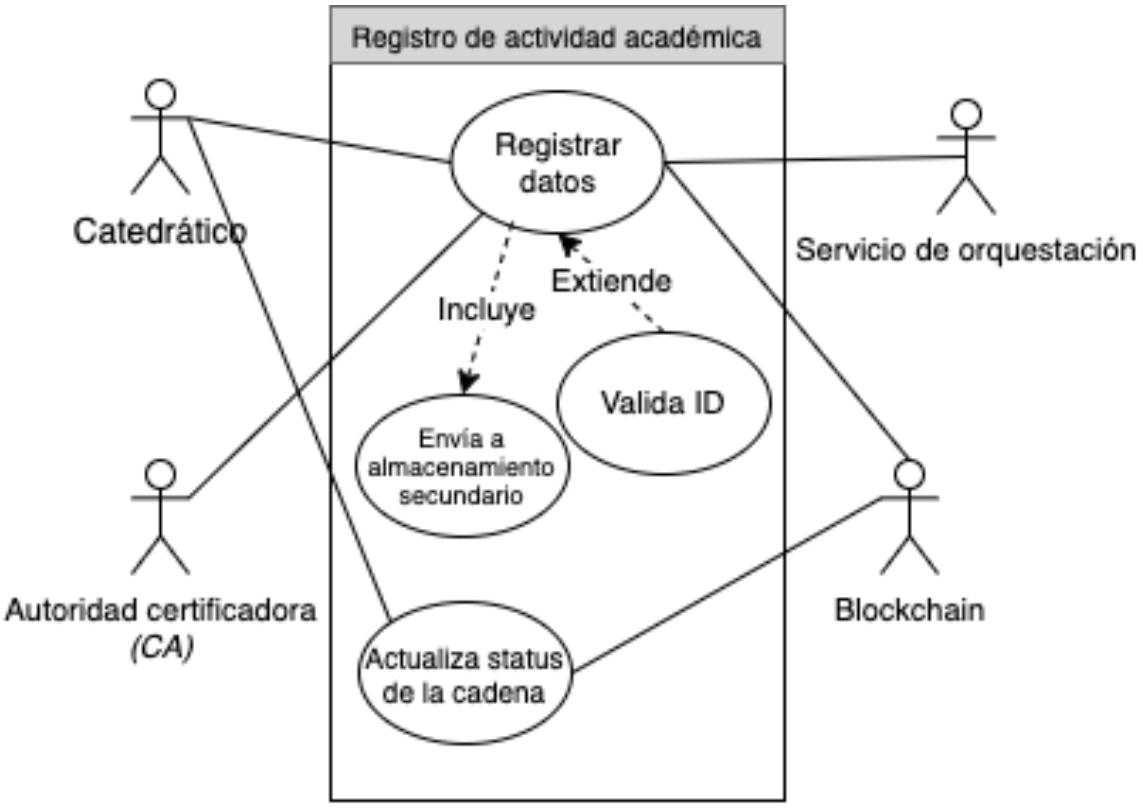
CU-01	**Registro de participante**	
Actor:	Participante, *CA*, Servicio de Orquestación	
Secuencia Normal	Paso	Acción
	1	Participante ingresa datos para la estructura de nuevo participante
	2	Se genera el *request* con los datos hacia el *CA* de la red a la que pertenece (Unidad Facultativa)
	3	*CA* emite cetificado de enrolamiento (Llave publica-privada)
	4	Con el certificado de enrolamiento se genera certificado para transacciones
	5	*CA* emite certificado para firma de transacciones
	6	Usuario almacena su certificado para autenticarse y firma de transacciones
Excepciones	Paso	Acción
	1	Se captura la excepción en caso de un *request* mal formado y se presenta al usuario mensaje con indicador del posible origen.
	3,5	Si el usuario ya existe y está enrolado no se podrá emitir nuevo certificado
Comentarios		

Fuente: elaboración propia.

5.3.6.2. Registro de actividad académica

A continuación, la figura 17 muestra el diagrama de casos de uso del registro de actividad académica y la tabla III el detalle de la funcionalidad. Este puede representar el final de un curso y puede conllevar una nota de aprobación o no aprobación. Cada nota o grupo de notas será cargada por el catedrático correspondiente.

Figura 17. **Caso de uso para registro de nueva actividad académica**



Fuente: elaboración propia.

Tabla III. **Definición de caso de uso Registro de nueva actividad académica**

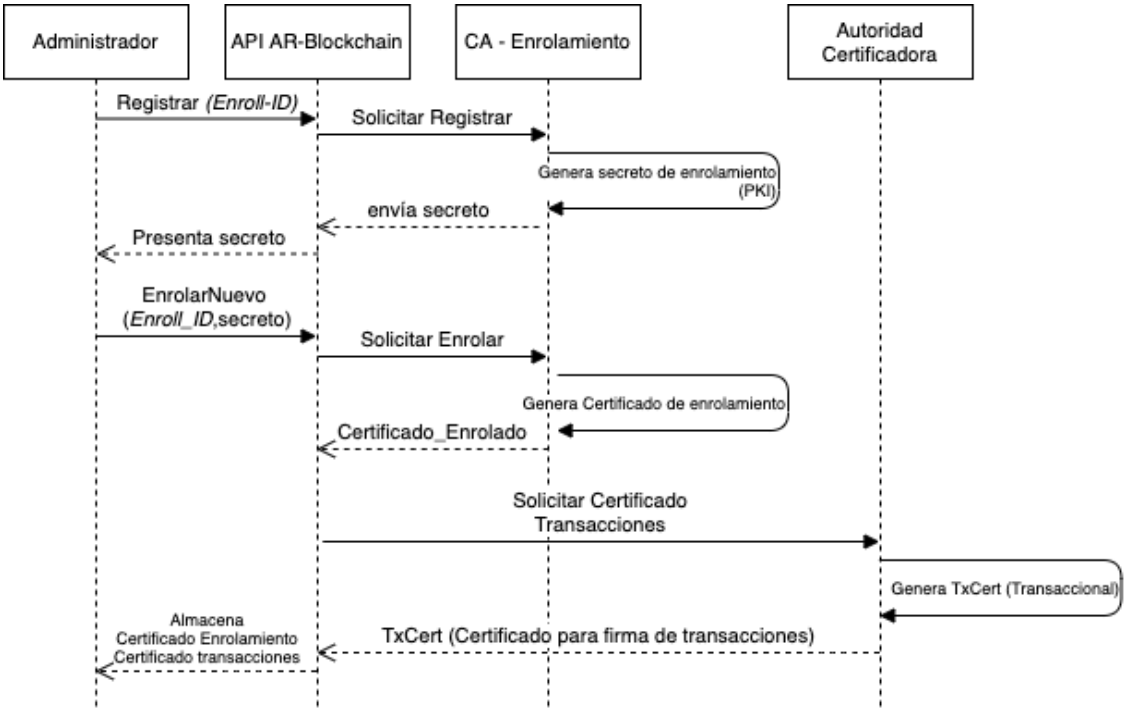
CU-02	**Registro de Actividad Académica**	
Actor:	Catedrático, *CA*, Servicio de Orquestación, *Blockchain*	
Secuencia Normal	Paso	Acción
	1	Participante genera *request* para registro de nueva actividad.
	2	Se envía el *request* en el canal al que pertenece el participante.
	3	Datos enviados son puestos a disposición a través del servicio de orquestación para que la red valide su integridad y verifiquen su autenticidad.
	4	Validado el bloque se generan las tramas *read/write* para escritura en la cadena. El servicio de orquestación hace *broadcast* hacia los nodos de la red.
	5	Cada nodo actualiza su base de datos de estados actuales.
Excepciones	Paso	Acción
	2	El canal no se encuentra habilitado, disponible o aceptando peticiones.
	3	No se cumple la política de consenso y se rechaza la transacción.
	5	El estado actual de la base de datos de estados no se encuentra en un estado consistente respecto al resto de la red de nodos.
Comentarios		

Fuente: elaboración propia.

5.3.7. Diagramas de secuencia

Para entender de manera más detallada los casos de uso presentados anteriormente se muestra las secuencias de eventos para el registro de nuevos participantes, así como el registro de una actividad académica.

Figura 18. **Diagrama de secuencia de registro de usuario**

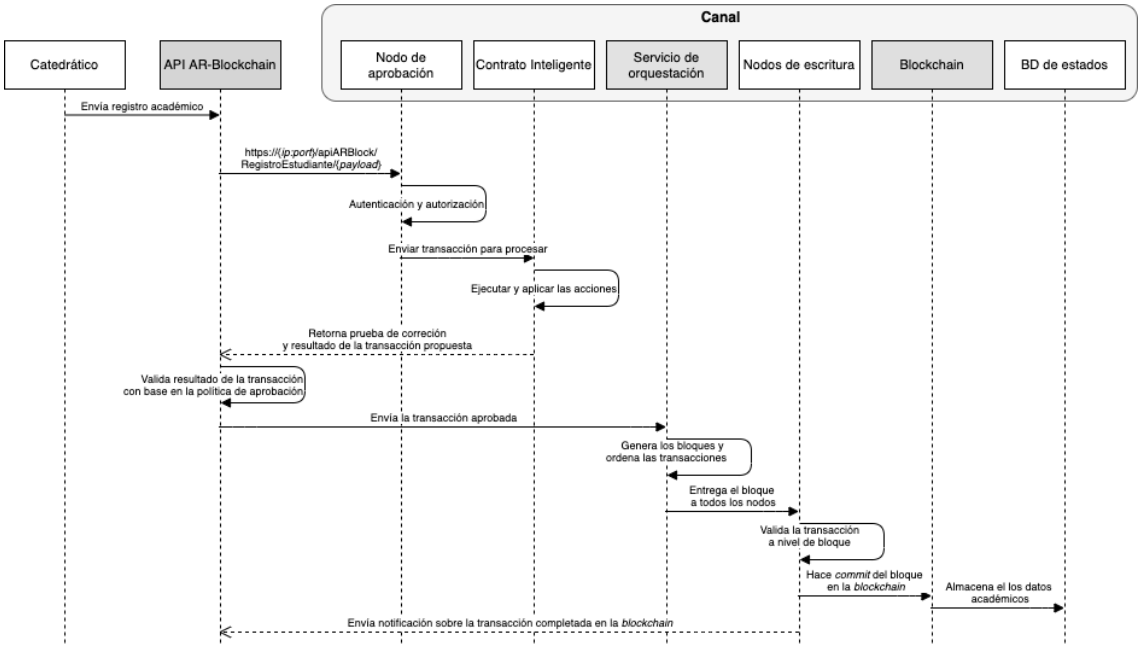


Fuente: elaboración propia.

El registro de nuevos usuarios será realizado por un usuario administrador ya que la red no es pública. Para esto en la arquitectura se establece una autoridad certificadora (*CA*) que se encarga de extender los certificados criptográficos con los cuales los usuarios interactúan con la red. A través de estos

se identifican de forma anónima y se registra a través de su firma criptográfica que cada transacción es de un usuario autorizado.

Figura 19. **Diagrama de secuencia de registro evento académico**



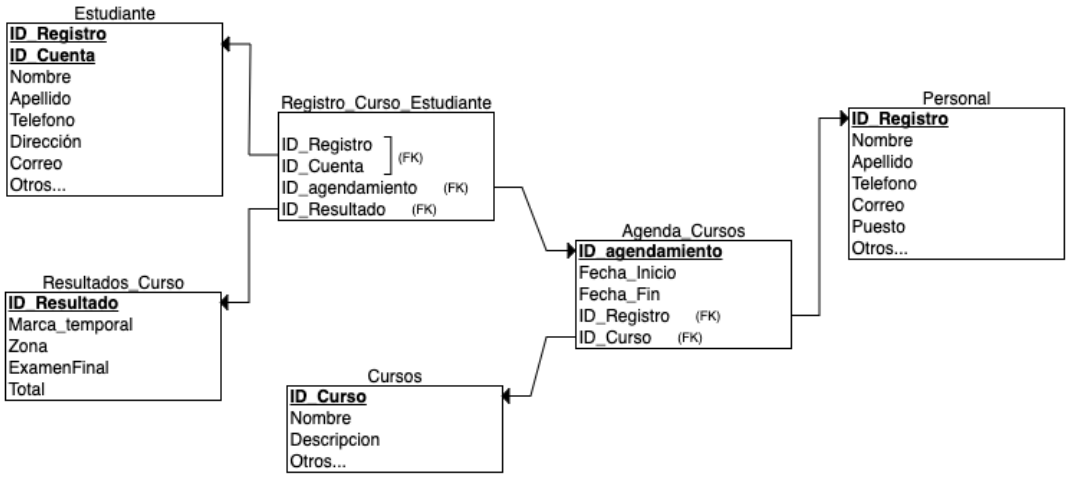
Fuente: elaboración propia.

El registro de actividades académicas se realizará a través de usuarios que tengan privilegios suficientes para enviar transacciones para su validación, aprobación y posterior anexión a la cadena como método de auditoría. La figura 19 muestra la secuencia que sigue el envío de una transacción hacia la red y realiza la ejecución del flujo correspondiente. La auditoría almacenada por la *blockchain* es criptográficamente segura y evitando así una alteración en el transcurso de la secuencia de recepción, aprobación y escritura.

5.3.8. Diagrama entidad – relación

Las estructuras de datos utilizadas por los sistemas de registro de datos académicos tradicionalmente hacen uso del modelo relacional en su patrón maestro – esclavo.

Figura 20. **Modelo entidad - relación convencional**



Fuente: elaboración propia.

5.3.9. Configuración de la blockchain Hyperledger Fabric

Se utilizó el servicio en la nube de Amazon Web Services ya que proveen entre sus servicios la tecnología de *blockchain* en *Hyperledger Fabric*, de forma que la implementación se pudo realizar con mayor velocidad con los siguientes puntos importantes durante la configuración y despliegue.

5.3.9.1. Creación de la *blockchain* en AWS

Se utilizó infraestructura como código para la creación de los nodos a través del servicio *cloudformation* de AWS considerando una sola red y un nodo, así como las restricciones de red necesarias.

Se crearon los servidores o nodos dentro de la zona de disponibilidad que AWS define para el servicio de *blockchain* manejada REGION=*us-east-1*. Las especificaciones de recursos para estos servidores se encuentran en el archivo *Yaml* ejecutado en *cloudformation*.

Figura 21. **Definición de la red blockchain en *template* de AWS *Cloudformation***

```
Parameters:
  NetworkName:
    Description: The name of your Amazon Managed Blockchain network.
    AllowedPattern: "^[0-9a-zA-Z]+$"
    ConstraintDescription: Network name must be alphanumeric and cannot contain spaces.
    Type: String
    Default: ngo
  NetworkDescription:
    Description: An optional description of your network.
    Type: String
    Default: Network for tracking donations made to non-profit organisations
  Framework:
    Description: The blockchain protocol to use, such as Hyperledger Fabric
    Type: String
    Default: HYPERLEDGER_FABRIC
  FrameworkVersion:
    Description: The version of the blockchain protocol to use
    Type: String
    Default: 1.2
  Edition:
    Description: Setting that determines the number of peer nodes per member and the selection of instance types
    ConstraintDescription: Must be STARTER or STANDARD
    Default: STARTER
    AllowedValues:
      - STARTER
      - STANDARD
    Type: String
  MemberName:
    Description: The name of the first member in your Amazon Managed Blockchain network.
    AllowedPattern: "^[0-9a-zA-Z]+$"
    ConstraintDescription: MemberName must be alphanumeric.
    Type: String
```

Fuente: elaboración propia.

Se seleccionó un nodo con pocos recursos de procesador y memoria *RAM* para mantener los costos lo más bajo posible. Siendo la definición del nodo de la *blockchain* la siguiente:

Figura 22. **Creación del nodo donde corre la *blockchain* en el *template* de AWS *Cloudformation***

```
PeerNodeAvailabilityZone:
  Description: The Availability Zone for your first peer node.
  Default: us-east-1a
  Type: String
PeerNodeInstanceType:
  Description: The type of compute instance to use for your peer nodes.
  Default: bc.t3.small
  Type: String
  AllowedValues:
    - bc.t3.small
    - bc.t3.medium
    - bc.t3.large
    - bc.t3.xlarge
    - bc.m5.large
    - bc.m5.xlarge
    - bc.m5.2xlarge
    - bc.m5.4xlarge
    - bc.c5.large
    - bc.c5.xlarge
    - bc.c5.2xlarge
    - bc.c5.4xlarge
  ConstraintDescription: >-
    If Edition is STARTER, then this value must be bc.t3.small
    or bc.t3.medium.
```

Fuente: elaboración propia.

La asignación o manejo de los recursos creados tanto para la red manejada de *Hyperledger Fabric* como de su nodo se ven asociados en la plantilla de creación de los componentes en la siguiente instrucción que se muestra en la figura 25.

Figura 23. **Asignación de la red manejada *Hyperledger Fabric* creada y el primer nodo**

```
Resources:
  Member:
    Type: "AWS::ManagedBlockchain::Member"
    Properties:
      NetworkConfiguration:
        Name: !Ref NetworkName
        Description: !Sub "network-${NetworkName}-${AWS::AccountId}"
        Framework: !Ref Framework
        FrameworkVersion: !Ref FrameworkVersion
        NetworkFrameworkConfiguration:
          NetworkFabricConfiguration:
            Edition: !Ref Edition
        VotingPolicy:
          ApprovalThresholdPolicy:
            ThresholdPercentage: !Ref ThresholdPercentage
            ProposalDurationInHours: !Ref ProposalDurationInHours
            ThresholdComparator: !Ref ThresholdComparator
      MemberConfiguration:
        Name: !Sub "member-${NetworkName}"
        Description: !Sub "member-${NetworkName}-${AWS::AccountId}"
        MemberFrameworkConfiguration:
          MemberFabricConfiguration:
            AdminUsername: !Ref MemberAdminUsername
            AdminPassword: !Ref MemberAdminPassword
  MemberPeerNode:
    Type: "AWS::ManagedBlockchain::Node"
    Properties:
      NetworkId: !GetAtt Member.NetworkId
      MemberId: !GetAtt Member.MemberId
      NodeConfiguration:
        InstanceType: !Ref PeerNodeInstanceType
        AvailabilityZone: !Ref PeerNodeAvailabilityZone
```

Fuente: elaboración propia.

El archivo se ejecutó en la *CLI* proporcionada por AWS con el siguiente conjunto de instrucciones.

Figura 24. **Creación de nodo nuevo**

```
#Ejecución AWS Cloud9 CLI
export REGION=us-east-1
export STACKNAME=ar-chain-amb
cd ~/ar-blockchain/archain-fabric
./amb.sh
```

Fuente: elaboración propia.

Al finalizar la ejecución se validó que los elementos estuvieran con el estatus *"CREATION_COMPLETE"* en la consola visual de AWS. Tanto para la creación de la *blockchain* como el primer nodo.

5.3.9.2. Creación de cliente *fabric* y enrolar identidad

El nodo cliente *fabric* se creó en una *VPC* o segmento de red propio a través de *cloudformation* desde *Cloud9* al igual que la ejecución de la creación del inciso previo.

Figura 25. **Configuración de VPC para cliente *fabric template* AWS *Cloudformation***

```
BlockchainWorkshopVPC:
  Type:                                    AWS::EC2::VPC
  Properties:
    CidrBlock:                             10.0.0.0/16
    EnableDnsSupport:                      True
    EnableDnsHostnames:                    True
    InstanceTenancy:                       default
    Tags:
      - Key:                               ReInventBlockchainWorkshop
        Value:                             VPC

BlockchainWorkshopPublicSubnet:
  Type:                                    AWS::EC2::Subnet
  Properties:
      VpcId:                               !Ref BlockchainWorkshopVPC
      MapPublicIpOnLaunch:                 false
      CidrBlock:                           10.0.0.0/16
      Tags:
      - Key:                               ReInventBlockchainWorkshop
        Value:                             PublicSubnet

BlockchainWorkshopSecurityGroupBase:
      Type:                                AWS::EC2::SecurityGroup
      Properties:
        GroupDescription:                  Base Security Group
        VpcId:                             !Ref BlockchainWorkshopVPC
        SecurityGroupIngress:
        - IpProtocol:                      tcp
          CidrIp:                          0.0.0.0/0
          FromPort:                        22
```

Fuente: elaboración propia.

Los datos relevantes a la creación de la infraestructura se desplegaron en la consola con la siguiente información:

Figura 26. **Salida del *template* del cliente *fabric* AWS *Cloudformation***

```
Outputs:
  VPCID:
    Description:                    VPC ID
    Value:
      !Ref                          BlockchainWorkshopVPC
  PublicSubnetID:
    Description:                    Public Subnet ID
    Value:
      !Ref                          BlockchainWorkshopPublicSubnet
  SecurityGroupID:
    Description:                    Security Group ID
    Value:
      !GetAtt                       BlockchainWorkshopSecurityGroupBase.GroupId
  EC2URL:
    Description:                    Public DNS of the EC2 instance
    Value:
      !GetAtt                       BlockchainWorkshopEC2.PublicDnsName
  ELBDNS:
    Description:                    Public DNS of the ELB
    Value:
      !GetAtt                       BlockchainWorkshopELB.DNSName
  BlockchainVPCEndpoint:
    Description:                    VPC Endpoint ID
    Value:
      !Ref                          BlockchainWorkshopVPCEndpoint
```

Fuente: elaboración propia.

La ejecución de la plantilla de AWS Cloudformation se realizó de la siguiente forma en la línea de comando:

Figura 27. **Ejecución de plantilla de *Cloudformation***

```
export REGION=us-east-1
cd ~/ar-blockchain/archain-fabric
./vpc-client-node.sh
```

Fuente: elaboración propia.

Creada la red *blockchain* de *Hyperledger fabric* y el cliente que interactúa con ella se procedió a enrolarlo y generar una identidad o certificado con el cual pudiera autenticarse ante la *blockchain* y enviar transacciones.

Para esto se accedió en el nodo del cliente *fabric* y se crearon los archivos *fabric-exports.sh* y *template-exports.sh*

Figura 28. **Seteo de variables de entorno necesarias para contexto de comunicación entre el nodo cliente *fabric* y el nodo *blockchain Hyperledger fabric***

```
export REGION=us-east-1
export STACKNAME=$(aws cloudformation describe-stacks --region
export NETWORKNAME=$(aws cloudformation describe-stacks --stack
export MEMBERNAME=$(aws cloudformation describe-stacks --stack-
export NETWORKVERSION=$(aws cloudformation describe-stacks --st
export ADMINUSER=$(aws cloudformation describe-stacks --stack-n
export ADMINPWD=$(aws cloudformation describe-stacks --stack-na
export NETWORKID=$(aws cloudformation describe-stacks --stack-n
export MEMBERID=$(aws cloudformation describe-stacks --stack-na

VpcEndpointServiceName=$(aws managedblockchain get-network --re
OrderingServiceEndpoint=$(aws managedblockchain get-network --r
CaEndpoint=$(aws managedblockchain get-member --region $REGION
nodeID=$(aws managedblockchain list-nodes --region $REGION --ne
peerEndpoint=$(aws managedblockchain get-node --region $REGION
peerEventEndpoint=$(aws managedblockchain get-node --region $RE
export ORDERINGSERVICEENDPOINT=$OrderingServiceEndpoint
export ORDERINGSERVICEENDPOINTNOPORT=${ORDERINGSERVICEENDPOINT:
export VPCENDPOINTSERVICENAME=$VpcEndpointServiceName
export CASERVICEENDPOINT=$CaEndpoint
export PEERNODEID=$nodeID
export PEERSERVICEENDPOINT=$peerEndpoint
export PEERSERVICEENDPOINTNOPORT=${PEERSERVICEENDPOINT::-6}
export PEEREVENTENDPOINT=$peerEventEndpoint
```

Fuente: elaboración propia.

Con las variables necesarias se creó el certificado para la entidad *admin*. Con este administramos la red de *Hyperledger Fabric*, desplegamos los diferentes contratos inteligentes (*chaincode*) y creamos el canal en el cual se comunica el cliente con la red.

Figura 29. **Despliegue de contrato inteligente**

```
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/managedblockchai-tls-chain.pem -M /home/ec2-user/admin-msp
```

Fuente: elaboración propia.

5.3.9.3. Creación de un canal *fabric*

La comunicación de los nodos a través de la *blockchain Hyperledger Fabric* sucede dentro de un canal que es el medio que agrupa a los posibles participantes y cuyo identificador de canal permite enrutar y enviar los mensajes de forma eficiente y segura a través servicio de orquestación de la arquitectura.

Para el establecimiento del canal utilizado se realizó la ejecución de las instrucciones siguientes:

Figura 30. **Creación del canal**

```
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
```

Fuente: elaboración propia.

En este se habilita la comunicación *TLS/SSL* en donde se le pasó como parámetros, el certificado generado para el nodo por el *CA* y la configuración del canal que se exportaron a variables de entorno en el paso anterior. El resultado de esta creación fue la creación del archivo *archannel.block*.

Figura 31. **Validación de creación archivo archannel.block**

```
ls -lt /home/ec2-user/fabric-samples/chaincode/hyperledger/fabric/peer
```

Fuente: elaboración propia.

Generado el canal se procedió con enrolar un *peer node* que conforma la topología, para el efecto el *peer node* sería el correspondiente a cada una de las unidades facultativas que conforman el entorno de la administración académica y poseen operaciones sobre registros académicos. Se realizó desde la terminal del cliente *fabric* creado.

Figura 32. **Enrolar un *peer node* en la topología**

```
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 join -b $CHANNEL.block -o $ORDERER --cafile
$CAFILE -tls
```

Fuente: elaboración propia.

En este punto la red de *blockchain* en *Hyperledger Fabric* se creó de forma satisfactoria, así como su bloque origen que hace referencia a la creación y configuración inicial del canal y el primer *peer node* asociado tanto al nodo *fabric* como al cliente *fabric*.

5.3.9.4. Instalar la *chaincode*

La *chaincode* es conocida como el contrato inteligente. Esto es código que ejecuta reglas de negocio. Por tal razón se considera como un contrato inteligente, su ejecución puede darse para validar las condiciones en las cuales se debe tratar la información dentro de los nodos de la red.

Dentro de la solución propuesta se implementó un contrato inteligente para revisar condiciones dentro del manejo de datos de la *blockchain* con la que se evaluó su conformidad con la lógica de negocio en la que para crear un registro académico debe existir la unidad autorizada encargada de generar un nuevo conjunto de datos (*ADS*).

Figura 33. **Funciones definidas para la creación de una unidad ADS de la red distribuida**

```
/**
 * Creates a new ADS
 *
 * @param {*} stub
 * @param {*} args - JSON as follows:
 * {
 *    "adsRegistrationNumber":"6322",
 *    "adsName":"Facultad Ingeniería",
 *    "adsDescription":"Unidad académica Facultad de Ingeniería",
 *    "address":"Zona 12, Ciudad Universitaria. Guatemala, Guatemala",
 *    "contactNumber":"24188000",
 *    "contactEmail":"ingenieria@usac.edu.gt"
 * }
 */
async createADS(stub, args) {
  console.log('============= START : createADS ===========');
  console.log('##### createADS arguments: ' + JSON.stringify(args));

  // args is passed as a JSON string
  let json = JSON.parse(args);
  let key = 'ads' + json['adsRegistrationNumber'];
  json['docType'] = 'ads';

  console.log('##### createADS payload: ' + JSON.stringify(json));

  // Check if the ADS already exists
  let adsQuery = await stub.getState(key);
  if (adsQuery.toString()) {
    throw new Error('##### createADS - This ADS already exists: ' + json['adsRegistrationNumber']);
  }

  await stub.putState(key, Buffer.from(JSON.stringify(json)));
  console.log('============= END : createADS ===========');
}
```

Fuente: elaboración propia.

La unidad que almacena los registros académicos que bajo la solución de arquitectura propuesta pasó a formar parte de una unidad identificada por la autoridad de certificación dentro de la red además de almacenar los datos dentro de su propio silo de registros electrónicos necesita entidades de las cuales

almacenar registros académicos. Por lo que se generaron funciones para la creación entidades estudiante y catedrático.

Figura 34. **Funciones definidas para validación de estudiante en la *chaincode***

```
/**
 * Creates a new Student
 *
 * @param {*} stub
 * @param {*} args - JSON as follows:
 * {
 *    "registroAcademico":"201220435",
 *    "email":"student@test.com",
 *    "nombres":"Student 1",
 *    "apellidos":"Lastname 1",
 *    "telefono":"22343556",
 *    "celular":"56786777",
 *    "direccion":"16 calle 9-75 z12, guatemala",
 *    "dpi":"2261323250101",
 *    "registeredDate":"2019-10-22T11:52:20.182Z"
 * }
 */
async createStudent(stub, args) {
  console.log('============= START : createStudent ===========');
  console.log('##### createStudent arguments: ' + JSON.stringify(args));

  // args is passed as a JSON string
  let json = JSON.parse(args);
  let key = 'student' + json['studentUserName'];
  json['docType'] = 'student';

  console.log('##### createStudent payload: ' + JSON.stringify(json));

  // Check if the student already exists
  let studentQuery = await stub.getState(key);
  if (studentQuery.toString()) {
    throw new Error('##### createStudent - This student already exists: ' + json['studentUserName']);
  }

  await stub.putState(key, Buffer.from(JSON.stringify(json)));
  console.log('============= END : createStudent ===========');
}
```

Fuente: elaboración propia.

Figura 35. **Funciones definidas para la validación de catedráticos en la *chaincode***

```
/**
* Creates a new Teacher
*
* @param {*} stub
* @param {*} args - JSON as follows:
* {
*    "registroEmpleado":"201220435",
*    "email":"teacher@test.com",
*    "nombres":"Teacher 1",
*    "apellidos":"Lastname 1",
*    "telefono":"22253536",
*    "celular":"58786587",
*    "direccion":"16 calle 6-50 z8, guatemala",
*    "dpi":"2333623250101",
*    "registeredDate":"2019-10-20T11:52:20.182Z"
* }
*/
async createTeacher(stub, args) {
  console.log('============= START : createTeacher ===========');
  console.log('##### createTeacher arguments: ' + JSON.stringify(args));

  // args is passed as a JSON string
  let json = JSON.parse(args);
  let key = 'teacher' + json['registroEmpleado'];
  json['docType'] = 'teacher';

  console.log('##### createTeacher payload: ' + JSON.stringify(json));

  // Check if the teacher already exists
  let donorQuery = await stub.getState(key);
  if (donorQuery.toString()) {
    throw new Error('##### createTeacher - This teacher already exists: ' + json['registroEmpleado']);
  }

  await stub.putState(key, Buffer.from(JSON.stringify(json)));
  console.log('============= END : createTeacher ===========');
}
```

Fuente: elaboración propia.

Con estas entidades de negocio (catedrático, curso, estudiante) se ejecutan las validaciones del contrato inteligente para la creación de un nuevo registro académico (*AR*). Se definió la estructura de datos considerando que previo a la generación de un registro académico se debe tener un estudiante válido, un catedrático válido y una asignación a curso válido.

Figura 36. **Funciones definidas para validación de AR en la *chaincode***

```
 * {
 *    "adsRegistrationNumber":"6322",
 *    "arId":"1234",
 *    "teacherId":"2",
 *    "studentId":"1",
 *    "courseScheduleId":"5",
 *    "arDescription":"Ingreso de Notas Ordinaria",
 *    "arDate":"2019-09-20T12:41:59.582Z",
 *    "arZona":56,
 *    "arExFinal":25,
 * }
 */
async createAcademicRecord(stub, args) {
  console.log('============= START : createAcademicRecord ===========');
  console.log('##### createAcademicRecord arguments: ' + JSON.stringify(args));

  // args is passed as a JSON string
  let json = JSON.parse(args);
  let key = 'ar' + json['arId'];
  json['docType'] = 'ar';

  console.log('##### createAcademicRecord AR: ' + JSON.stringify(json));

  // Confirm the ADS exists
  let ngoKey = 'ads' + json['adsRegistrationNumber'];
  let adsQuery = await stub.getState(adsKey);
  if (!adsQuery.toString()) {
    throw new Error('##### createAcademicRecord - Cannot create academic record as the ADS does not exist: '
  }

  // Check if the AR already exists
  let arQuery = await stub.getState(key);
  if (arQuery.toString()) {
    throw new Error('##### createAcademicRecord- This Academic Record already exists: ' + json['arId']);
  }

  await allocateAcademicRecord(stub, json);
```

Fuente: elaboración propia.

Con la creación de un nuevo registro se procedió a propagar la información en los nodos validando que la distribución se haga de forma correcta.

Figura 37. **Funciones para validar en la *chaincode* la distribución de los registros académicos**

```
* {
*   "docType":"arAllocation",
*   "arAllocationId":"c5b29e938a19a80c225d30e8327caaf817f76aecd381c868263c4f59b45dgf72-1",
*   "arAllocationZona":51,
*   "arAllocationExFinal":25,
*   "arAllocationDate":"2019-09-25T15:31:29.582Z",
*   "arAllocationDescription":"Ingreso Nota Ordinaria - Seminario I",
*   "studentId":"FFF6A68D-DB19-4CD3-97B0-01C1A793ED3B",
*   "teacherId":"FFF6A48D-DB59-4DD3-97B1-01B1A493E23C",
*   "courseScheduleId":"5",
*   "ADSRegistrationNumber":"D0194B30-685D-499E-A9CD-2C6DCE5GAG48",
*   "arId": "1234"
* }
*/

/**
 * Retrieves a specfic arAllocation
 *
 * @param {*} stub
 * @param {*} args
 */
async querySpendAllocation(stub, args) {
  console.log('============= START : queryArAllocation ===========');
  console.log('##### queryArAllocation arguments: ' + JSON.stringify(args));

  // args is passed as a JSON string
  let json = JSON.parse(args);
  let key = 'arAllocation' + json['arAllocationId'];
  console.log('##### queryArAllocation key: ' + key);
  return queryByKey(stub, key);
}
```

Fuente: elaboración propia.

Figura 39. **Funciones de validación en la *chaincode* para la distribución de los registros en la red**

```
/**
 * Retrieves the arAllocation records for a specific Course
 *
 * @param {*} stub
 * @param {*} args
 */
async queryarAllocationForCourse(stub, args) {
  console.log('============= START : queryArAllocationForCourse ===========');
  console.log('##### queryArAllocationForCourse arguments: ' + JSON.stringify(args));

  // args is passed as a JSON string
  let json = JSON.parse(args);
  let queryString = '{"selector": {"docType": "arAllocation", "arId": "' + json['arId'] + '"}}';
  return queryByString(stub, queryString);
}

/**
 * Retrieves the arAllocation records for a specific Course record
 *
 * @param {*} stub
 * @param {*} args
 */
async queryArAllocationForCourse(stub, args) {
  console.log('============= START : queryArAllocationForCourse ===========');
  console.log('##### queryArAllocationForCourse arguments: ' + JSON.stringify(args));

  // args is passed as a JSON string
  let json = JSON.parse(args);
  let queryString = '{"selector": {"docType": "arAllocation", "courseId": "' + json['courseId'] + '"}}';
  return queryByString(stub, queryString);
}
```

Fuente: elaboración propia.

Al mismo tiempo se proporcionó en el contexto del contrato inteligente, una función para obtener valores.

Con las funciones definidas se procedió a la instalación del contrato inteligente en el canal *fabric* creado con anterioridad.

Figura 40. **Instalación del contrato inteligente**

```
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
```

Fuente: elaboración propia.

El resultado de la instalación se tuvo de forma exitosa.

Figura 41. **Resultado de la instalación del contrato inteligente**

```
2019-10-29 19:21:34.004 UTC [chaincodeCmd] install -> INFO 003
Installed remotely response:<status:200 payload:"OK" >
```

Fuente: elaboración propia.

5.3.9.5. Invocar una transacción – *Ordering service*

El envío de transacciones se validó desde el cliente *fabric* a través de la línea de comando. Con esto se garantizó que la arquitectura propuesta y los nodos involucrados estuvieran dentro del mismo canal establecido para la comunicación de la organización y envío de transacciones.

Figura 42. **Ejemplo de transacción de registro**

```
*    "registroAcademico":"201220435",
*    "email":"student@test.com",
*    "nombres":"Student 1",
*    "apellidos":"Lastname 1",
*    "telefono":"22343556",
*    "celular":"56786777",
*    "direccion":"16 calle 9-75 z12, guatemala",
*    "dpi":"2261323250101",
*    "registeredDate":"2019-10-22T11:52:20.182Z"
```

Fuente: elaboración propia.

Figura 43. **Ejecución de envío de transacción vía 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 invoke -C mychannel -n archaincode \
    -c  '{"Args":["createStudent","{\"registroAcademico\":
\"20120435\", \"nombres\": \"Student 1\", \"apellidos\": \"Lastname
1\", \"telefono\": \"22343556\", \"celular\": \"56786777\",
\"direccion\": \"16 calle 9-75 z12, guatemala\", \"dpi\":
\"2261323250101\", \"registeredDate\": \"2019-10-
22T11:52:20.182Z\"}"]}' -o $ORDERER --cafile
/opt/home/managedblockchain-tls-chain.pem --tls
```

Fuente: elaboración propia.

Se pudo constatar la correcta ejecución de la creación de nuevo usuario obteniendo un identificador único de transacción.

Figura 44. **Resultado de transacción exitosa en la *blockchain***

```
{“transactionId”:”674164dc5a1cb8365cef2dc4a563932b845b2c6f56233c1870e8d
35128c2e048”}
```

Fuente: elaboración propia.

De forma análoga se realizó la operación para obtener el dato recién ingresado y validar que pueda recuperarse de la *blockchain*.

Figura 45. **Consulta de transacción en la *blockchain***

```
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 query -C mychannel -n archaincode -c
'{"Args":["queryStudent”,"{\"registroAcademico\": \"20120435\"}"]}'
```

Fuente: elaboración propia.

Al completar estas acciones se concluyó con la implementación de la *blockchain* basada en Hyperledger Fabric. En donde se pudo observar que el despliegue de reglas de negocio, así como de la lógica de consenso y tamaño de llaves para generación de certificados puede ser configurable y puede realizarse de forma asequible y eficaz en la nube a través de servicios como AWS.

5.3.10. Definición del API de la arquitectura

La tabla IV muestra las operaciones expuestas por la interfaz REST API para la comunicación entre cliente y la *blockchain* para el manejo de la trazabilidad de las transacciones que afectan el historial académico de los estudiantes. De forma generalizada el recurso hace referencia a la ruta de la entidad de datos que juntamente con el método enviado representa la acción específica a realizar. Por ejemplo, una invocación a un recurso con el método *GET* puede obtener la información de la entidad de datos definida y puede filtrarse de acuerdo a algún parámetro que se envíe en el *request*. En el caso que se desee crear una nueva entidad de datos se invocaría el recurso con el método *POST* junto con los datos especificados en la definición de la entidad de datos correspondiente.

Tabla IV. **Operaciones API – Blockchain**

HTTP request REST API - *Blockchain*		
Recurso	**Método**	**Acción**
/APIARBlock/Estudiante	GET, POST, PUT, DELETE	Manejo de estudiante
/ APIARBlock /Catedratico	GET, POST, PUT, DELETE	Manejo de catedrático
/ APIARBlock /RegistroEstudiante	GET, POST, PUT, DELETE	Manejo de datos académicos
/ APIARBlock /RegistroCatedratico	GET, POST, PUT, DELETE	Manejo de datos académicos
/ APIARBlock /updateRegistroEstudiante	POST	Actualización de datos académicos
/ APIARBlock /shareRegistro	POST	Comparte datos académicos con otros
/ APIARBlock /generarActaCurso	GET	Obtiene los datos para acta física

Fuente: elaboración propia.

La razón de utilizar la tecnología *REST* es porque permite la utilización de métodos síncronos y asíncronos así como soporte nativo a comunicación de mensajes a través de formato JSON que es un archivo estandarizado similar a *XML* en el intercambio de mensajes *SOAP*.

Para la implementación de la *API REST* se utilizó NodeJS como lenguaje de programación y el *framework Express* para creación de aplicaciones web. La instalación se realizó a través de *nvm* y *npm* para instalar la versión *lts/carbon* de *nodejs* y las dependencias necesarias.

Se realizó la instalación de la versión lts/carbon en el servidor

Figura 46. **Instalación de *Node JS* versión *LTS/Carbon***

```
. ~/.nvm/nvm.sh
nvm install lts/carbon
nvm use lts/carbon
```

Fuente: elaboración propia.

Se instalaron las dependencias necesarias

Figura 47. **Instalación de dependencias de *Node JS***

```
cd ~/AR-blockchain/arBlockchain-rest-API
npm install
```

Fuente: elaboración propia.

A continuación se muestra un extracto de la configuración utilizada para servir y procesar las peticiones *HTTP* enviadas al *API REST arBlockchain.*

Figura 48. **Código fuente para la configuración inicial de escucha del API**

```
hfc.addConfigFile('config.json');
var host = 'localhost';
var port = 3000;
var username = "";
var orgName = "";
var channelName = hfc.getConfigSetting('channelName');
var chaincodeName = hfc.getConfigSetting('chaincodeName');
var peers = hfc.getConfigSetting('peers');

//////////////////////////////////////////////////////////////////////////////
//////////////////////////////// START SERVER ////////////////////////////////
//////////////////////////////////////////////////////////////////////////////
var server = http.createServer(app).listen(port, function() {});
logger.info('****************** SERVER STARTED ************************');
logger.info('***************  Listening on: http://%s:%s  ******************',host,port);
server.timeout = 300000;

function getErrorMessage(field) {
    var response = {
        success: false,
        message: field + ' field is missing or Invalid in the request'
    };
    return response;
}
```

Fuente: elaboración propia.

Como se puede observar en la figura 49 se genera una instancia escuchando al puerto 3000 del *host*. Para tal efecto el nodo *Fabric* es el que sirve como *host*, adicionalmente de servir como punto medular en la orquestación de las transacciones hacia la *blockchain*.

A través de *express* se generaron las rutas del *API REST* para manejar las diferentes peticiones *POST* (envío de alguna operación hacia el *API*), *GET* (obtención de datos desde la *blockchain* a través del *API*). A continuación, se muestra la implementación de un método *POST* y *GET*.

Figura 49. **API REST http GET ruta /APIARBlock/Estudiante/**

```
/**
  *
  * JSON Student struct as follows:
  * {
  *    "registroAcademico":"201220435",
  *    "email":"student@test.com",
  *    "nombres":"Student 1",
  *    "apellidos":"Lastname 1",
  *    "telefono":"22343556",
  *    "celular":"56786777",
  *    "direccion":"16 calle 9-75 z12, guatemala",
  *    "dpi":"2261323250101",
  *    "registeredDate":"2019-10-22T11:52:20.182Z"
  * }
  */
  // GET a specific Student
app.get('/apiARBlock/Estudiante/:registroAcademico', awaitHandler(async (req, res) => {
    logger.info('================ GET on Student by ID');
    logger.info('Student ID : ' + req.params);
    let args = req.params;
    let fcn = "queryStudent";

    logger.info('##### GET on Student by ID - ID : ' + registroAcademico);
    logger.info('##### GET on Student by ID - StudentOrg : ' + orgName);
    logger.info('##### GET on Student by ID - channelName : ' + channelName);
    logger.info('##### GET on Student by ID - chaincodeName : ' + chaincodeName);
    logger.info('##### GET on Student by ID - fcn : ' + fcn);
    logger.info('##### GET on Student by ID - args : ' + JSON.stringify(args));
    logger.info('##### GET on Student by ID - peers : ' + peers);

    let message = await query.queryChaincode(peers, channelName, chaincodeName, args, fcn, registroAcademico, orgName);
    res.send(message);
}));
```

Fuente: elaboración propia.

Figura 50. **API REST http POST ruta /APIARBlock/Estudiante/**

```
/ POST Student
pp.post('/apiARBlock/Estudiante', awaitHandler(async (req, res) => {
   logger.info('================ POST on Student');
   var args = req.body;
   var fcn = "createStudent";

   logger.info('##### POST on Student - ID : ' + username);
   logger.info('##### POST on Student - StudentOrg : ' + orgName);
   logger.info('##### POST on Student - channelName : ' + channelName);
   logger.info('##### POST on Student - chaincodeName : ' + chaincodeName);
   logger.info('##### POST on Student - fcn : ' + fcn);
   logger.info('##### POST on Student - args : ' + JSON.stringify(args));
   logger.info('##### POST on Student - peers : ' + peers);

   let message = await invoke.invokeChaincode(peers, channelName, chaincodeName, args, fcn, username, orgName);
   res.send(message);
));
```

Fuente: elaboración propia.

Como se puede observar se asocia una ruta en la cual servir las peticiones, en este caso relacionadas con el estudiante. Se define una llamada asíncrona que espera la respuesta de la invocación de la consulta a la cadena de bloques a través del método *queryChaincode*. Cabe resaltar que los atributos *channelName, chaincodeName* forman parte de un archivo de configuración.

Figura 51. **Archivo de configuración de la *blockchain***

```
hfc.addConfigFile('config.json');
var channelName = hfc.getConfigSetting('channelName');
var chaincodeName = hfc.getConfigSetting('chaincodeName');
var peers = hfc.getConfigSetting('peers');
```

Fuente: elaboración propia.

La variable *hfc* hace referencia a la librería (*fabric-client*) utilizada dentro de *nodejs* para manejar los objetos a los que un cliente *fabric* puede acceder. Se observa la carga del archivo *config.js*

Figura 52. **Parámetros de configuración cliente *hyperledger fabric* del API REST**

```
{
    "host":"localhost",
    "port":"3000",
    "channelName":"mychannel",
    "chaincodeName":"archaincode",
    "eventWaitTime":"30000",
    "peers":[
        "peer1"
    ],
    "admins":[
        {
            "username":"admin",
            "secret":"Admin123!"
        }
    ]
}
```

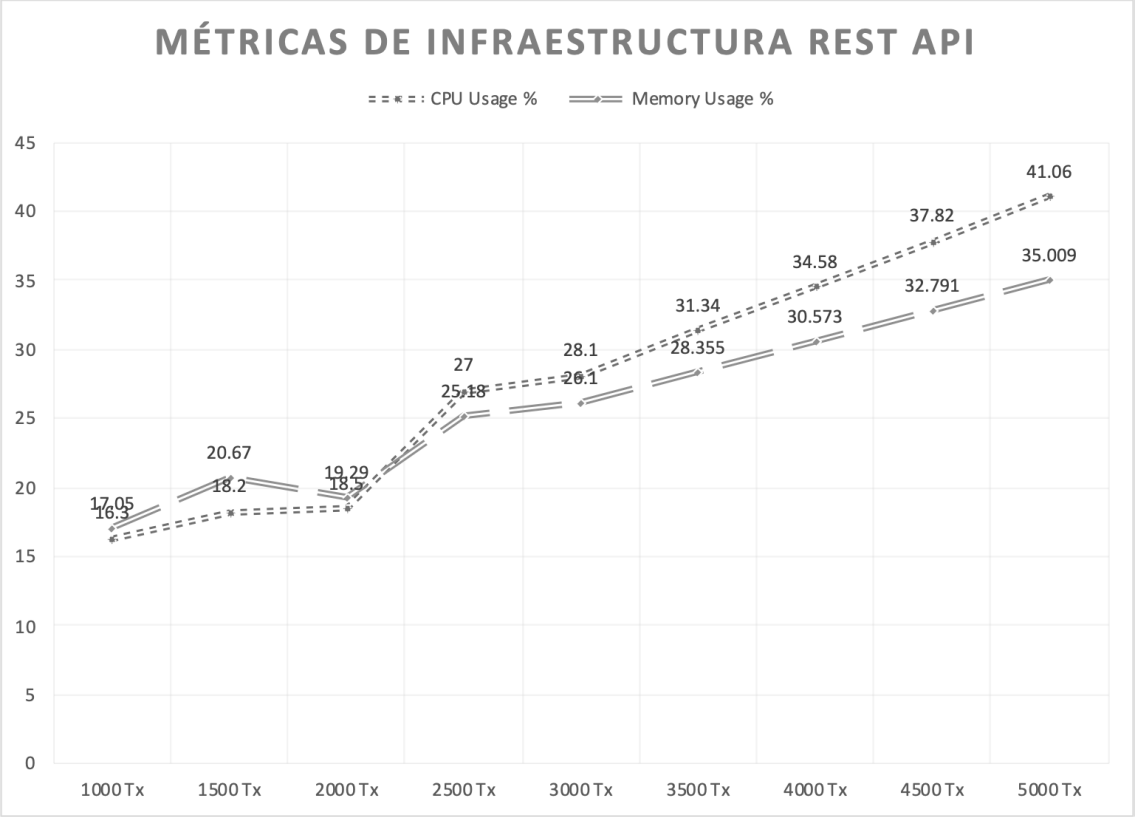
Fuente: elaboración propia.

Se le envía la parametrización de *host*, puerto que utiliza el *API* así como el nombre del canal, el contrato inteligente y nodos configurados para la red *blockchain*. El manejo de errores suscitados en la ejecución de las operaciones de la API fue manejado por *express* devolviendo un código de respuesta *HTTP* 500 de error interno de servidor.

Debido a que la implementación de arquitectura se realiza a través de contenedores manejados por AWS los nodos iniciales de *Hyperledger Fabric* y el peer utilizado pueden escalarse con base en métricas de rendimiento de la infraestructura *CPU* y memoria *RAM*.

Durante la implementación se obtuvieron métricas de monitoreo para nueve pruebas individuales de carga empezando con 1,000 transacciones incrementando en 500 este valor en cada ronda de prueba. Para un total de 27,000 solicitudes hacia el API.

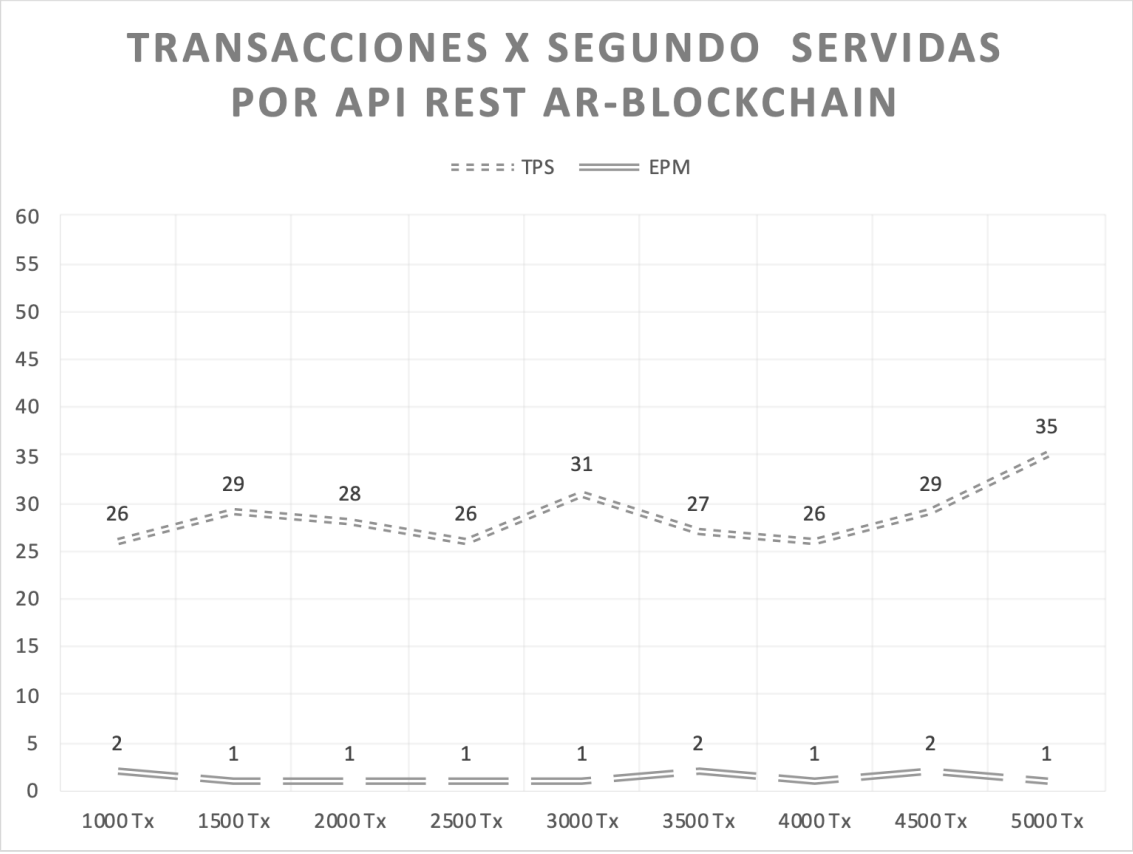
Figura 53. **Gráfica de rendimiento vs número de transacciones**



Fuente: elaboración propia.

Se observó un comportamiento ascendente a medida que la cantidad de transacciones se incrementó. Lo ilustrado en la figura 54 demuestra el comportamiento esperado para la prueba. La instancia en la cual se está corriendo el API también se utilizó para la instancia del servicio de orquestación de *Hyperledger Fabric* pero no se observa una utilización alta, considerando 2 núcleos de *CPU* y 4 *GB* de memoria.

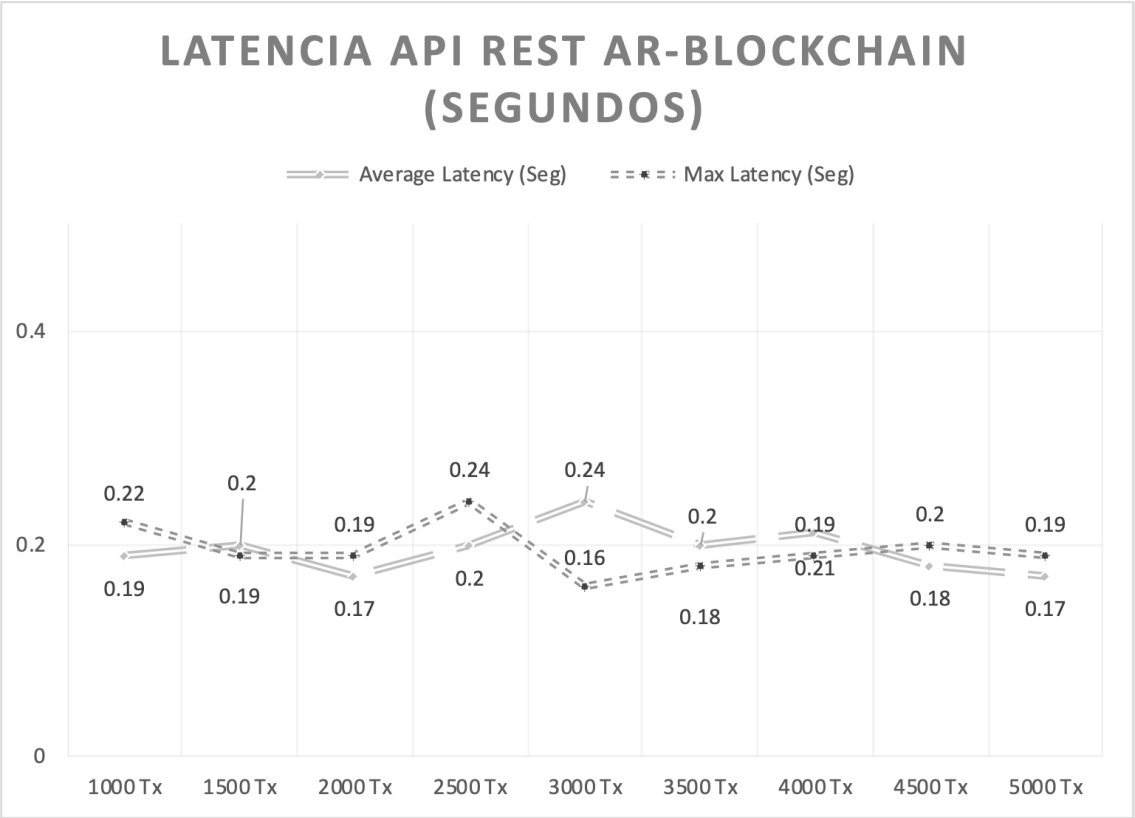
Figura 54. **Gráfica de transacciones servidas por el API**



Fuente: elaboración propia.

Además de las condiciones de infraestructura se monitorizaron las transacciones enviadas por un solo hilo una tras otra hacia el API obteniendo una alta tasa de servicio considerando que el punto más bajo obtenido fue 1,560 transacciones por minuto que equivalen a 26 transacciones por segundo. De la anterior se observa un cumplimiento de transacciones exitosas de aproximadamente 99.9 %.

Figura 55. **Gráfica de latencia**



Fuente: elaboración propia.

En la figura 56 se observa el comportamiento constante con el cual se sirven las peticiones. Estos valores pueden verse afectados de acuerdo con cada implementación por la conectividad de red, configuración de *proxy*, así como políticas de privacidad y encriptación de segmento de red que pueda acceder a la interfaz *API REST* expuesta.

La capa intermedia propuesta dentro de la solución permitió generar un historial auditable, seguro, distribuido, fácilmente escalable y garantizar que cada transacción almacenada no podrá alterarse en el tiempo.

Sea por la dificultad para modificar o eliminar datos ya agregados como para evitar alteraciones o intrusiones de terceros dada la seguridad proporcionada por los protocolos de cifrado utilizados no solo en la autenticación sino del intercambio de mensajes entre los nodos de la red.

6. DISCUSIÓN DE RESULTADOS

Los resultados muestran que una capa intermedia de auditoría de datos no reemplaza a los registros que se envían hacia la base de datos, más bien complementa los mismos como una capa que describe el origen y la cadena de custodia que permite garantizar la exactitud y certeza. Estos atributos permiten que los datos almacenados permanezcan en un grado de confiabilidad aceptable a través del tiempo para su posterior consulta.

6.1. Seguridad y cifrado

Un aspecto importante que considerar en la implementación de una arquitectura basada en *blockchain*, es que implica evaluar diferentes métodos de verificación de identidad, así como de protección a los mensajes en el intercambio entre los diferentes nodos que conforman la red. La infraestructura de llave pública y privada en la que se basa el protocolo de seguridad implementado por la autoridad certificadora garantiza de manera matemática que la complejidad para suplantar la identidad de un nodo o una persona sea muy difícil, mucho más que robar credenciales, como lo son el medio tradicional de usuario y contraseña.

La seguridad en el manejo de registros académicos se ha incrementado con el uso de *MOOC's* y plataformas de educación en línea en donde la custodia de la información de estudiantes, catedráticos y entidades educativas ya no poseen registros históricos de manera física. Pues este registro les plantea un proceso menos eficiente, lento y en ocasiones hasta deficiente si los archivos físicos (actas, notas, memoriales) no se categorizan y preservan adecuadamente.

En la presentación de resultados se observó que la implementación de *blockchain* a través de *Hyperledger Fabric* puede hacerse de forma rápida y eficiente a través de plataformas de servicios en la nube como AWS, pero el aspecto financiero debe considerarse de acuerdo con el alcance y presupuesto.

La inmutabilidad de los datos es un concepto abstracto que desde el punto de vista tecnológico se puede enfrentar a través de protocolos de cifrado cuya complejidad algorítmica sea lo suficientemente grande para que no solamente se necesite capacidad computacional sino también elementos físicos que puedan respaldar los datos académicos digitales. Algoritmos como SHA o ECDSA proporcionan diferentes niveles de personalización de la seguridad a través del tamaño de la llave. Sin embargo, estos protocolos, su selección y uso deben apegarse a los objetivos de cada aplicación.

Como se definió en el diseño de la arquitectura presentado en los resultados la red se definió como permisiva y no pública. Sin embargo, la diferencia con protocolos de implementaciones de *blockchain* públicas como *Bitcoin* o *Ethereum* radica principalmente en el método de consenso y la forma en que se recompensa a los nodos de la red por su trabajo. No se definió un método de consenso basado en una prueba de trabajo o una prueba de estado ya que se tomó como premisa la existencia de un grado de confianza medio – alto dentro de las unidades académicas que integrarían la solución.

Los algoritmos de *hashing* SHA 3 SHAKE 256 y ECDSA 256 vistos en la presentación de resultados forman parte del estándar utilizado para aplicaciones de este tipo en donde se desea mantener un protocolo de cifrado fuerte en términos computacionales.

Tabla V. **Comparación de funciones SHA**

Algoritmo	Variante	Tamaño de Salida (bits)	Tamaño del bloque (bits)	Rondas	Seguridad (en bits) contra ataque de colisiones	Rendimiento en Skylake (cpb)		Publicado en
						Mensaje largo	8 bytes	
SHA-2	SHA-224	224	512	64	112	7.62	84.5	2004
	SHA-256	256			128	7.63	85.25	2001
	SHA-384	384	1024	80	192	5.12	135.75	2001
	SHA-512	512			256	5.06	135.5	
	SHA-512/224	224			112	≈ SHA-384	≈ SHA-384	2012
	SHA-512/256	256			128			
SHA-3	SHA3-224	224	1152	24	112	8.12	154.25	2015
	SHA3-256	256	1088		128	8.59	155.5	
	SHA3-384	384	832		192	11.06	164	
	SHA3-512	512	576		256	15.88	164	
	SHAKE128	d (arbitrary)	1344		min(*d*/2, 128)	7.08	155.25	
	SHAKE256	d (arbitrary)	1088		min(*d*/2, 256)	8.59	155.5	

Fuente: Comparison of SHA functions. Consultado el 3 de julio de 2020. Recuperado de https://en.wikipedia.org/wiki/Template:Comparison_of_SHA_functions.

La tabla V resalta la comparación de las funciones de *hashing* seleccionadas dentro de la implementación para usarse en el protocolo de cifrado. Puede observarse que la diferencia en términos de seguridad computacional es la implementación del algoritmo en su composición interna.

En donde SHA3 SHAKE 256 posee un tamaño de bloque mayor, lo que supone un incremento en la capacidad computacional para vulnerar el algoritmo.

Sin embargo, *SHA2* es mucho más rápido. En términos de rendimiento medido en procesadores Intel con microarquitectura *Skylake SHA2* es aproximadamente un 12 % más rápido al utilizarse. Esto implica una mayor eficiencia en el intercambio de mensajes dentro de la red y menor latencia.

La auditoría tradicional de transacciones en almacenamiento secundario o base de datos hace uso de esquemas para insertar registros cada vez que una acción monitoreada o definida se ejecuta. Esta acción puede ser desde un portal web público, un sitio administrativo privado, entre otros. Sin embargo, el sistema puede ser criptográficamente seguro si se utiliza *HTTPS/SSL* en transferencia de mensajes entre cliente-servidor y servidor-BD. Inclusive si se utiliza una base de datos encriptada la seguridad puede aumentar al almacenar todo registro dentro de la base de datos cifrado.

Este enfoque protege los datos, pero no garantiza que no se hayan alterado en el tiempo ya que el registro de auditoría que debe de ser inmutable no tiene forma de verificarse más que a través de historiales de aplicación, historiales de base de datos y esto recae en la lógica de codificación y la capacidad que los *DMBS* posean para manejar cifrado dentro de su arquitectura de datos.

6.2. Auditoría de datos

Un aspecto importante que considerar es el objetivo para el cual preservar los registros académicos en el tiempo. Y los alcances de los sistemas de información que gestionan la información académica, no solo del estudiante sino de catedráticos, cursos, personal administrativo, en fin. La auditoría de datos puede circunscribirse a todas las áreas de una organización educativa y puede realizarse en diferentes niveles como recursos físicos (sillas, escritorios,

marcadores, libros, actas, entre otros.) y digitales (documentos escaneados, diplomas digitales, verificación de documentos físicos).

La correcta definición de los objetivos y alcances de un plan de auditoría de datos también delimitan a su vez el marco en el cual los sistemas de información se aplican a un mundo cada vez más inmerso en procesos digitales y remotos donde la privacidad es una preocupación. Se observó dentro del análisis de las implementaciones de *blockchain* que existen segmentos específicos como lo son las multinacionales, los gobiernos, los centros de investigación, el sector financiero inclusive que necesitan de un planteamiento propio de los conceptos clave que hacen que *blockchain* sea la tecnología que permite almacenar registros inmutables en el tiempo.

En el presente trabajo no se consideraron tecnologías de cifrado en los *DBMS* para la protección de datos puesto que este tipo de soluciones se encuentran en productos comercialmente costosos y cuyo objetivo no se centra en la inmutabilidad de los datos sino en la protección de estos que son conceptos diferentes pero complementarios. Se asume la existencia de sistemas de base de datos relacionales como lo puede ser Oracle, MySQL, SQLServer, Postgresql y no relacionales como MongoDB, Redis.

Otro aspecto para tomar en consideración es que los servicios en la nube proporcionan una amplia gama de productos complementarios a precios asequibles. Lo que podría significar una oportunidad de innovación o mejora por parte de la organización académica para realizar pruebas de concepto o planes piloto llevando a cabo procesos totalmente digitales que garanticen el origen, el uso y el almacenamiento de los datos.

Tabla VI. **Métricas de monitoreo CPU y memoria obtenidas sobre la implementación Hyperledger Fabric Blockchain**

Type	Name	Memory (Max)	Memory (Avg)	CPU (Max)	CPU (Avg)
Docker	dev-peer1.org1.archain	27.89 MB	17.6 MB	28.80 %	21.20 %
Docker	couchdb.org1.archain	18.33 MB	10.4 MB	23.10 %	11.80 %
Docker	orderer.org1.archain	38.2 MB	26.9 MB	32.70 %	28.50 %

Fuente: elaboración propia.

Tabla VII. **Métricas de monitoreo de tráfico obtenidas sobre la implementación Hyperledger Fabric Blockchain**

Type	Name	Traffic In	Traffic Out
Docker	dev-peer1.org1.archain	3.8 MB	2.7 MB
Docker	couchdb.org1.archain	5.8 MB	9.8 MB
Docker	orderer.org1.archain	4.6 MB	3.8 MB

Fuente: elaboración propia.

Tabla VIII. **Métricas de rendimiento obtenidas sobre la implementación Hyperledger Fabric Blockchain**

Test	Name	Success	Fail	Send Rate	Max Latency
1	Physical Network Access	998	2	26 tps	194 ms
2	Physical Network Access	1499	1	38 tps	328 ms
3	Physical Network Access	1999	1	35 tps	220 ms
4	Physical Network Access	2499	1	51 tps	197 ms
5	Physical Network Access	2999	1	35 tps	258 ms
6	Physical Network Access	3498	2	45 tps	271 ms
7	Physical Network Access	3999	1	26 tps	170 ms
8	Physical Network Access	4498	2	58 tps	209 ms
9	Physical Network Access	4999	1	29 tps	254 ms

Fuente: elaboración propia.

Tabla IX. **Métricas de monitoreo latencia sobre la implementación Hyperledger Fabric Blockchain**

Test	Name	Min Latency	Avg Latency	Throughput
1	Physical Network Access	107 ms	232 ms	26 tps
2	Physical Network Access	209 ms	413 ms	38 tps
3	Physical Network Access	176 ms	305 ms	35 tps
4	Physical Network Access	131 ms	252 ms	51 tps
5	Physical Network Access	252 ms	392 ms	35 tps
6	Physical Network Access	117 ms	298 ms	45 tps
7	Physical Network Access	106 ms	212 ms	26 tps
8	Physical Network Access	265 ms	365 ms	58 tps
9	Physical Network Access	218 ms	363 ms	29 tps

Fuente: elaboración propia.

En las tablas VI y VII se puede observar el rendimiento obtenido a través de la capa intermedia propuesta de la solución. Esto ejemplifica el consumo bajo de recursos de *Hyperledger Fabric* para el manejo de peticiones de los usuarios

autorizados de la red. Además de ser métricas que pueden generar auto escalamiento horizontal adicionando configuración de alta disponibilidad en la implementación. Inclusive puede realizarse escalamiento vertical adicionando mayor cantidad de recursos de *RAM*, disco y red dado que la implementación se realiza a través de contenedores.

En cuanto al ratio de procesamiento de propuestas en la *blockchain* como se observa en las tablas VIII y IX de la misma forma que con la ratio de procesamiento del API REST que se tuvieron propuestas fallidas hacia la red. Estas se dieron debido a que al ejecutar las validaciones dentro de la ejecución del contrato inteligente la transacción provenía de usuarios no autorizados o datos inconsistentes que no cumplían con los requisitos de forma para ingresar la transacción, véase figura 38.

6.3. Marco para un plan de calidad de datos

Tomando en cuenta la norma ISO 14721:2012 se identificaron componentes esenciales para la elaboración de un plan de calidad de datos. Si bien este estándar se aplica al archivado y preservación de archivos tanto físico como digital propone una base sólida para mantener los registros académicos dentro de un expediente digital bajo un formato y procedimientos ampliamente validados por la industria. En la figura 12 se puede observar el ecosistema general propuesto, en donde consumidores de información realizan propuestas de acceso a los archivos hacia el sistema y los productores generan contenido con base en normas de calidad establecidas para su fin.

Con ambos actores en perspectiva se requiere un intermediario que gestione y planifique el intercambio de datos desde el origen hasta el destino.

En el modelo de acceso a la información evaluado para las unidades académicas dentro de la Universidad de San Carlos de Guatemala cada una de ellas posee su propio método de archivado de datos en gran medida este proceso consiste en una secuencia de requisitos como formularios de solicitud, presentación de carné estudiantil o algún otro tipo de documento que permita corroborar la información tanto de productores como emisores.

Basado en el estándar y considerando los procesos ordinarios que deben realizarse de manera física se cree necesario que la planificación de calidad de datos de cada unidad académica incluya las consideraciones siguientes:

- Sistemas de información homologados entre unidades académicas.
- Métodos de gestión de accesos tanto físico como digital para los sitios web, así como físicos en los cuales se almacene o consulten datos académicos.
- Utilización de técnicas de auditoría digital y física que proporcione trazabilidad a lo largo del tiempo sobre el manejo de datos.
- Implementación de protocolos de seguridad y cifrado en el traslado y custodia de los datos entre unidades académicas y unidades administrativas centralizadas.

Dados los puntos anteriores los sistemas de gestión de datos actuales pueden adecuarse para cumplir con el uso de una capa intermedia de auditoría que, si bien no garantiza en un 100 % la calidad de los datos por factores como la naturaleza de los procesos no automatizados, mitiga el riesgo por el cual un sistema no puede ser considerado confiable proveyendo un amplio espectro de puntos de seguridad que deben de ser vulnerados al mismo tiempo para alterarlos.

6.4. Impacto tecnológico

Con el presente trabajo se observó que los sistemas pueden llevarse aún más a procesos con menos intervención física. El ejecutar condiciones vinculantes entre contratante y contratista dentro de un marco de contratos inteligentes proporciona a la entidad académica mayor eficiencia en sus procesos y permite una mayor facilidad de auditar cada interacción para un mejor control.

Una arquitectura distribuida basada en *blockchain* para el manejo de registros académicos en expedientes estudiantiles digitales demuestra que *blockchain* es un concepto que no solo tiene impacto en el sector financiero con criptomonedas sino también en la educación con registros académicos criptográficamente fuertes que permitan una mayor agilidad en el intercambio, consulta y almacenamiento a través del tiempo.

CONCLUSIONES

1. Se implementó satisfactoriamente una arquitectura distribuida en la nube, la cual a través de Hyperledger Fabric y AWS permite auditar digitalmente los registros académicos contenidos dentro de un expediente estudiantil, garantizando la inmutabilidad de los datos y proporcionando fácil escalabilidad y 100 % de trazabilidad de registros digitales y físicos.

2. Se analizaron protocolos de cifrado que incluyen algoritmos de hash SHA2, SHA3 y firma digital con curvas elípticas con llaves de tamaño 256, 512 y 1088 bits. En función de las necesidades de seguridad y complejidad criptográfica el protocolo utilizado fue SHA3 SHAKE256 y ECDSA con llaves de 256 bits de tamaño que implican una alta complejidad para vulnerar la confianza de la arquitectura y los nodos que la conforman, pero lo suficientemente rápida en términos de latencia para procesar transacciones a través de servicios *REST* por parte de la unidad académica dentro de sus procesos de generación de datos dentro de un expediente estudiantil digital.

3. Se realizó el diseño e implementación de una arquitectura distribuida para auditoría con AWS e Hyperledger Fabric con la cual se expuso un *API REST* en Node JS para el envío de datos sobre registros académicos que se almacenan en base de datos. Esta arquitectura compone una capa intermedia que ejecuta reglas de negocio a través de contratos inteligentes y permite identificar transacciones de origen dudoso y descartarlas. De tal cuenta que los registros almacenados en bases de datos tengan trazabilidad dentro de la *blockchain* y a través la implementación de protocolos criptográficos de SHA y ECDSA con llaves de 256 bits mitigando

el riesgo de que los registros académicos generados sean modificados luego de su creación.

4. Se analizó y revisó la norma ISO 14721:2012 para la preservación de archivos, del cual se identificó que el modelo enmarcado como consumidor-administrador-productor hace sentido en un plan de calidad de datos tanto físicos como digitales definiendo un esquema de acceso estricto a los actores que pueden formar parte del rol de consumidor o productor de datos. Dado que el alcance de la arquitectura propuesta es proveer de una capa intermedia de auditoría se propuso el uso de una estructura de datos *JSON* que a su vez proveen una estructura ligera y estándar ampliamente usado a través de protocolos web como el del *API REST* expuesta.

RECOMENDACIONES

1. Documentar, en función de los resultados obtenidos en el diseño e implementación de la solución propuesta en este trabajo, de forma extensa los procesos dentro de la organización educativa para identificar los documentos que pueden generarse tanto de forma física como de forma digital y en función de estos determinar los datos relevantes a ser almacenados dentro de la *blockchain* y mitigar riesgos de registros faltantes, duplicados, erróneos o que posiblemente nunca se generaron.

2. Escalar el uso de servicios en la nube para la implementación de la solución propuesta dado que permite mayor rapidez en el uso de los recursos computacionales de forma horizontal como vertical y acceso a diversos servicios de explotación de datos que podría aprovecharse al tener expedientes académicos digitales y anexar más información relevante tanto cualitativa como cuantitativa.

3. Incrementar la eficiencia de los procesos educativos adaptando los sistemas actuales a esta capa de auditoría que permita gestionar de forma eficaz, segura e inmutable los datos académicos y administrativos pudiendo adaptarse a las necesidades de cada organización educativa. Con esto se podría reducir el posible error humano en los procesos de auditoría, garantizar la cadena de custodia de los datos académicos e incrementar la confianza en los mismos a través de una red descentralizada y criptográficamente segura.

4. Mejorar la auditoría de datos a través de un plan integral de archivado tanto de registros físicos como digitales implementando ISO 14721:2012 que a su vez pueda ser auditado y registrado dentro de la *blockchain* permitiendo de esta forma no solamente garantizar información académica de expedientes estudiantiles sino también la generación y acceso de expedientes digitales para documentos de investigación o trabajos de grado, postgrado y doctorados.

REFERENCIAS

1. A Blockchain Platform for Enterprise. (2020). *Hyperledger Fabric*. Recuperado de: https://hyperledger-fabric.readthedocs.io/en/release-2.0/#

2. Antonopoulus A. (2015). *Mastering Bitcoin: Unlocking digital cryptocurrencies.* California, United States of America: O'Reilly Media, Inc.

3. Bashir I. (2017). *Mastering Blockchain: Distributed ledgers, decentralization and smart contracts explained.* Birmingham, UK: Packt Publishing Ltd.

4. Bennet E., y Webber A. (Mayo, 2020). Cloud computing in New York State education: Case study of failed technology adoption of a statewide longitudinal database for student data. *QScience Connect* 2015(1), 1-19. Recuperado de https://www.qscience.com/content/journals/10.5339/connect.2015.2

5. Cervantes, H., Kazman, R. (2016). *Designing software architectures: a practical approach*. Boston, United States of America: Addison-Wesley.

6. Cook, Merrill. (01 de diciembre, 2016). *State of the MOOC 2016: A year of massive landscape change for massive open online courses* [Mensaje en un blog]. Recuperado de: https://www.onlinecoursereport.com/state-of-the-mooc-2016-a-year-of-massive-landscape-change-for-massive-open-online-courses/

7. Drescher, D. (2017). *Blockchain Basics: A Non-Technical Introduction in 25 Steps*. Nueva York, United States of America: Apress.

8. Gill, P.S. (2011). *Database Management Systems. Second Edition*. New Delhi, India: I.K. International Publishing House Pvt. Ltd.

9. Gräther, W., Kolvenbach, S., Ruland, R., Schütte, J., Torres, C., y Wendland, F. (Mayo, 2018). Blockchain for Education: Lifelong Learning Passport. *ERCIM-blockchain 2018: Blockchain Engineering: Challenges and Opportunities for Computer Science Research.* Congreso llevado a cabo en Amsterdam, Holanda.

10. Lavoie B. (2004). *The Open Archival Information System (OAIS) Reference Model: Introductory Guide 2nd Edition*. Great Britain: Digital Preservation Coalition.

11. Mougayar, W. (2016). *The Business Blockchain: Promise, Practice, and Application of the Next internet Technology.* New Jersey, United States of America: John Wiley y Sons, Inc.

12. Newman S. (2015). *Building Microservices.* California, United States of America: O'Reilly Media, Inc.

13. Ocheja, P., Flanagan, B., y Ogata, H. (Marzo, 2018). Connecting decentralized learning records: a blockchain based learning analytics platform. *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*. Congreso llevado a cabo en Sydney New South Wales, Australia.

14. Richardson C. y Smith F. (2016). *Microservices: From Design to Deployment, a Free Ebook from NGINX.* California, United States of America: NGINX.

15. Rozansky, N., Woods E. (2005). *Software systems architecture: working with stakeholders using viewpoints and perspectives.* New Jersey, United States of America: Pearson Education, Inc.

16. Schmidt, Philipp. (27 de octubre, 2015). *Certificates, Reputation and the Blockchain. Medium: Mit Media Lab*. [Mensaje en un blog]. Recuperado de: https://medium.com/mit-media-lab/certificates-reputation-and-the-blockchain-aee03622426f

17. Sharples, M., de Roock, R., Ferguson, R., Gaved, M., Herodotou, C., Koh, E., Kukulska-Hulme, A., Looi, C., McAndrew, P., Rienties, B., Weller, M., y Lung H. (Diciembre, 2016). Blockchain for learning. *Innovating Pedagogy 2016 Open University Innovation Report 5*, 51-43. Recuperado de: https://bit.ly/3CZmUPO

18. Slawomir Jan Magala. (Marzo, 2018). Equality, creativity, academic merit: for the record or to set records up?. *Journal of Organizational Change Management*, 31(3), 466-467.

19. Sony Global Education. (2017, agosto 09). *Sony Develops System for Authentication, Sharing, and Rights Management Using Blockchain Technology*. [Comunicado de prensa]. Recuperado de https://www.sony.net/SonyInfo/News/Press/201708/17-071E/index.html

20. Swan M. (2015). *Blockchain: Blueprint for a New Economy.* California, United States of America: O'Reilly Media, Inc.

21. Szabo, N. (25 de enero, 2018). Smart Contracts: *Building Blocks for Digital Markets*. [Mensaje en un blog]. Recuperado de http://www.truevaluemetrics.org/DBpdfs/BlockChain/Nick-Szabo-Smart-Contracts-Building-Blocks-for-Digital-Markets-1996-14591.pdf

22. Tapscott, D., Tapscott A. (2016). *Blockchain Revolution: how technology behind the bitcoin is changing money, business and the world.* New York, United States of America: Penguin Random House LLC.

23. Tapscott, Don y Tapscott Alex. (13 de marzo, 2017). *The Blockchain Revolution and Higher* Education. [Mensaje en un blog]. Recuperado de https://er.educause.edu/articles/2017/3/the-blockchain-revolution-and-higher-education

24. University of Nicosia. (2014). *Blockchain Certificates (Academic & Others).* Recuperado de: https://digitalcurrency.unic.ac.cy/free-introductory-mooc/self-verifiable-certificates-on-the-bitcoin-blockchain/academic-certificates-on-the-blockchain/

25. Watters, Audrey. (07 de abril, 2016). *The Blockchain for Education: An Introduction* [Mensaje en un blog]. Recuperado de: http://hackeducation.com/2016/04/07/blockchain-education-guide