UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO

FACULTAD DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA, INFORMÁTICA Y MECÁNICA

ESCUELA PROFESIONAL DE INGENIERÍA INFORMÁTICA Y DE SISTEMAS

TESIS

BLOCKCHAIN APLICADO A LA SEGURIDAD PARA LA GESTIÓN DE INFRACCIONES DE TRÁNSITO EN LA MUNICIPALIDAD PROVINCIAL DEL CUSCO

PRESENTADO POR:

Br. VICTOR ABEL CHOQUEVILCA QUISPE

Br. ERIKA ALEXANDRA MORALES VALENCIA

PARA OPTAR EL TITULO PROFESIONAL DE INGENIERO INFORMÁTICO Y DE SISTEMAS

ASESOR:

Dr. RONY VILLAFUERTE SERNA

CUSCO - PERÚ

2024

UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO
VICE RECTORADO DE INVESTIGACIÓN

INFORME DE ORIGINALIDAD

(Aprobado por Resolución Nro.CU-303-2020-UNSAAC)

El que suscribe, asesor del trabajo de investigación titulado: "**BLOCKCHAIN APLICADO A LA SEGURIDAD PARA LA GESTIÓN DE INFRACCIONES DE TRÁNSITO EN LA MUNICIPALIDAD PROVINCIAL DEL CUSCO**". Presentado por los bachilleres:

- Victor Abel Choquevilca Quispe con DNI nro: 41485563
- Erika Alexandra Morales Valencia con DNI nro: 45787898

Para optar al Título Profesional de Ingeniero Informático y de Sistemas, informo que el trabajo de investigación ha sido sometido a revisión por **3 VECES**, mediante el software anti plagio, conforme al Artículo 6° del **Reglamento para Uso de Sistema Anti plagio de la UNSAAC** y de la evaluación de originalidad se tiene un porcentaje de 3% (tres por ciento).

Evaluación y acciones del reporte de coincidencia para trabajos de investigación conducentes a grado académico o título profesional, tesis

PORCENTAJE	EVALUACIÓN Y ACCIONES	MARQUE CON UNA X
Del 1 al 10 %	No se considera plagio.	X
Del 11 al 30%	Devolver al usuario para las correcciones.	
Mayores a 31 %	El responsable de la revisión del documento emite un informe al inmediato jerárquico, quien a su vez eleva el informe a la autoridad académica para que tome las acciones correspondientes. Sin perjuicio de las sanciones administrativas que correspondan de acuerdo a ley.	

Por tanto, en mi condición de Asesor, firmo el presente informe en señal de conformidad y adjunto la primera hoja del reporte del software anti plagio.

Cusco, 19 de febrero del 2024

Dr. Rony Villafuerte Serna
DNI: 23957778
ORCID: 0000-0003-4607-522X

Se adjunta:

1. Reporte generado por el sistema anti plagio.
2. Enlace del reporte generado por el sistema anti plagio: OID: 27259:333552643

turnitin **Identificación de reporte de similitud:** oid:27259:333552643

NOMBRE DEL TRABAJO
BlockchainV3

AUTOR
Victor & Erika Choquevilca & Morales

Conforma

RECUENTO DE PALABRAS
17548 Words

RECUENTO DE CARACTERES
106190 Characters

RECUENTO DE PÁGINAS
105 Pages

TAMAÑO DEL ARCHIVO
1.9MB

FECHA DE ENTREGA
Feb 19, 2024 10:36 PM GMT-5

FECHA DEL INFORME
Feb 19, 2024 10:37 PM GMT-5

3% de similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para cada base de datos.

- 3% Base de datos de Internet
- 0% Base de datos de publicaciones
- Base de datos de Crossref
- Base de datos de contenido publicado de Crossref
- 2% Base de datos de trabajos entregados

Excluir del Reporte de Similitud

- Material bibliográfico
- Material citado
- Coincidencia baja (menos de 20 palabras)
- Bloques de texto excluidos manualmente

Resumen

AGRADECIMIENTO

A nuestro asesor Dr. Rony Villafuerte Serna por el tiempo, conocimiento y apoyo profesional que nos brindó durante todo el proceso de investigación para la culminación de este proyecto.

A nuestros dictaminantes, el Ing. Robert Wilbert Alzamora Paredes y el Ing. José Mauro Pillco Quispe, los cuales gentilmente, nos brindaron aportes para nuestro proyecto.

A nuestros profesores, por todo el conocimiento y aportes que nos brindaron durante los años que estuvimos estudiando en la Universidad Nacional San Antonio Abad del Cusco.

A la Gerencia de Tránsito, Viabilidad y Transito de la Municipal Provincial del Cusco por su valioso tiempo y apoyo hacia nuestro proyecto.

DEDICATORIA

A Dios por haberme acompañado y guiado a lo largo de mi carrera, por los aprendizajes y experiencias que me ha dado.

A mis padres, por su apoyo en todo momento y por el gran ejemplo de vida que he tenido.

A mi esposo e hija, por su amor, motivación y compañía en este proceso.

A mis hermanos, familia y amigos, quienes me han acompañado en este camino, por su paciencia, consejos y apoyo.

Por todo ello, muchas gracias.

Erika Alexandra Morales Valencia

A mis padres quienes me dieron la vida, educación y apoyo incansable.

A mis hermanos por el cariño y confianza depositada en mí.

A mis amigos por el apoyo y exigencia.

Por todo ello, muchas gracias.

Victor Abel Choquevilca Quispe

RESUMEN

La gestión de infracciones de tránsito en la Municipalidad Provincial del Cusco enfrenta desafíos sustanciales en cuanto a seguridad, integridad y transparencia de la información, debido a vulnerabilidades inherentes en los sistemas tradicionales. Este trabajo de investigación se centra en la identificación de limitaciones en los métodos convencionales y propone explorar la aplicabilidad de la tecnología blockchain como solución innovadora para mejorar la seguridad en este sistema.

El estudio inicia con un análisis de los requisitos para el uso de la tecnología blockchain en la gestión de infracciones de tránsito. Posteriormente, se propone la implementación de servicios web que consuman datos de entidades como el Registro Nacional de Identificación y Estado Civil (RENIEC), la Superintendencia Nacional de Registros Públicos (SUNARP) y el Ministerio de Transportes y Comunicaciones (MTC), para garantizar la veracidad de la información.

La adaptación de la tecnología blockchain en la gestión de infracciones vehiculares se presenta como el siguiente paso, destacando su capacidad para ofrecer un almacenamiento inmutable y seguro de datos, abordando así las vulnerabilidades presentes en los métodos tradicionales. La investigación subraya la importancia de la transparencia, trazabilidad y seguridad de los registros para fortalecer la confianza en el sistema, aspectos críticos para la eficiente gestión de infracciones.

Finalmente, las conclusiones respaldan la aplicación exitosa de la tecnología blockchain en la gestión de infracciones de tránsito en la Municipalidad Provincial del Cusco. Se destaca que esta tecnología mejora la seguridad, transparencia e inmutabilidad en el proceso de gestión de

infracciones de tránsito, consolidándose como una herramienta para fortalecer el sistema y avanzar hacia soluciones innovadoras y sostenibles en el campo de la seguridad datos.

Palabras clave: Infracciones de tránsito, Servicio Web, Blockchain.

ABSTRACT

The management of traffic violations in the Provincial Municipality of Cusco faces substantial challenges in terms of security, integrity and transparency of information, due to inherent vulnerabilities in traditional systems. This research work focuses on the identification of limitations in conventional methods and proposes to explore the applicability of blockchain technology as an innovative solution to improve security in this system.

The study begins with an analysis of the requirements for the use of blockchain technology in the management of traffic violations. Subsequently, the implementation of web services that consume data from entities such as the National Registry of Identification and Civil Status (RENIEC), the National Superintendence of Public Registries (SUNARP) and the Ministry of Transportation and Communications (MTC) is proposed, to guarantee the veracity of the information.

The adaptation of blockchain technology in the management of vehicle violations is presented as the next step, highlighting its ability to offer immutable and secure storage of data, thus addressing the vulnerabilities present in traditional methods. The research highlights the importance of transparency, traceability and security of records to strengthen trust in the system, critical aspects for the efficient management of violations.

Finally, the conclusions support the successful application of blockchain technology in the management of traffic violations in the Provincial Municipality of Cusco. It is highlighted that this technology improves security, transparency and immutability in the traffic violation management process, consolidating itself as a tool to strengthen the system and move towards innovative and sustainable solutions in the field of data security

Keywords: Transit infractions, Web Service, Blockchain.

INTRODUCCION

En la era digital actual, la tecnología Blockchain ha emergido como una herramienta revolucionaria con el potencial de transformar numerosos aspectos de nuestra sociedad. Su aplicación va más allá de las criptomonedas y se extiende a diversas áreas, incluida la seguridad y la gestión de datos. En este contexto, el presente trabajo de investigación se enfoca en explorar y analizar la aplicabilidad de la tecnología Blockchain para mejorar la seguridad en el sistema de infracciones de tránsito de la Municipalidad Provincial del Cusco.

La gestión eficiente de las infracciones de tránsito es fundamental para garantizar el cumplimiento de las normativas de tránsito. Sin embargo, el sistema actual enfrenta desafíos relacionados con la integridad, la transparencia y la seguridad de los datos. Los registros de infracciones, almacenados en bases de datos relacional, pueden ser susceptibles a manipulaciones, fraudes o pérdida de información, lo que socava la confianza en el sistema.

La tecnología Blockchain, conocida por su capacidad para ofrecer un registro inmutable y descentralizado, surge como una solución prometedora para abordar estos problemas. Al aplicar la tecnología Blockchain al sistema de infracciones de tránsito, se busca mejorar la confiabilidad de los registros, garantizar la transparencia en el proceso y fortalecer la seguridad de la información asociada con las infracciones de tránsito.

A lo largo de este trabajo de investigación, se explorarán los principios fundamentales de la tecnología Blockchain, sus características distintivas y cómo puede integrarse de manera efectiva en el sistema de gestión de infracciones de tránsito. Además, se realizará el consumo de datos de los servicios web de las instituciones como: el Registro Nacional de Identificación y Estado Civil (RENIEC), el Ministerio de Transportes y Comunicaciones (MTC) y de la Superintendencia Nacional de Registros Públicos (SUNARP).

En última instancia, este trabajo de investigación tiene como objetivo ofrecer una perspectiva integral sobre cómo la aplicación de la tecnología Blockchain puede contribuir a fortalecer la seguridad y la confianza en el sistema de infracciones de tránsito, brindando beneficios tanto para las autoridades de tránsito como para los ciudadanos.

LISTA DE ACRONIMOS

- AES : Advanced Encryption Standard (Estándar de cifrado avanzado)
- API : Application Programming Interface (Interfaz de Programación de Aplicaciones)
- DNI : Documento Nacional de Identidad
- DPoS : Delegated Proof of Stake (Prueba de Participación Delegada)
- HTTP : Hypertext Transfer Protocol (Protocolo de transferencia de hipertexto)
- MSMQ : Microsoft Message Queuing (Cola de mensajes de Microsoft)
- MTC : Ministerio de Transportes y Comunicaciones
- P2P : Peer to Peer (Igual a Igual)
- PBFT : Practical Byzantine Fault Tolezrance (Tolerancia Práctica de Fallos Bizantinos)
- PoS : Proof of Stake (Prueba de Participación)
- PoW : Proof Of Work (Prueba de Trabajo)
- RENIEC : Registro Nacional de Identificación y Estado Civil
- SNIP : Sistema Nacional de Inversión Pública
- SOA : Service Oriented Architecture (Arquitectura Orientada a Servicios)
- SOAP : Simple Object Access Protocol (Protocolo Simple de Acceso a Objetos)
- SUNARP : Superintendencia Nacional de Registros Públicos
- TCP : Transmission Control Protocol (Protocolo de control de transmisión)
- TI : Tecnología de la Información
- UDDI : Universal Description Discovery and Integration (Descripción Universal, Descubrimiento e Integración)
- URL : Uniform Resource Locator (Localizador de Recursos Uniforme)

- WAS : Windows Process Activation Service (Servicio de activación de procesos de Windows)
- WCF : Windows Communication Foundation (Fundación de Comunicación de Windows)
- WSDL : Web Services Description Language (Lenguaje de Descripción de Servicios Web)
- WWW : World Wide Web (Red Informática Mundial)
- XML : Extensible Markup Language (Lenguaje de Marcado Extensible)
- XSD : XML Schema Definition (Definición de Esquema XML)

INDICE GENERAL

INDICE DE FIGURAS

INDICE DE TABLAS

INDICE DE ANEXOS

CAPÍTULO I

ASPECTOS GENERALES

1.1 Planteamiento del problema

La Gerencia de Tránsito, Viabilidad y Transporte de la Municipalidad Provincial del Cusco es responsable de gestionar las infracciones de tránsito vehicular. El personal de la Gerencia de Tránsito, Viabilidad y Transporte recibe las infracciones físicas emitidas por el policía de tránsito y las registra manualmente en el sistema de la Municipalidad Provincial del Cusco, ingresando una serie de datos de la persona infractora. La información registrada del infractor incluye, el número de DNI, nombres y apellidos completos, dirección, número, clase y categoría de la licencia de conducir, datos del vehículo, como el número de placa, marca del vehículo, color, modelo, número de DNI, apellidos y nombres del propietario o propietarios y dirección; así como también datos de la infracción, como la descripción, lugar, tipo de infracción y tipo de reglamento.

Dado que este proceso de registro se realiza de forma manual y se almacena en una base de datos SQL Server, es susceptible a errores de digitación que puede comprometer la veracidad y la actualización oportuna de los datos de las personas infractoras y de los vehículos, ha esto se suma la poca seguridad en el almacenamiento de datos, lo que permite la posibilidad de manipulación de la información, ya sea intencional o accidental.

Este problema puede causar errores en los procesos subsiguientes, como son la cobranza de infracciones, notificaciones y emisión de resoluciones; además, estas deficiencias en la gestión de infracciones pueden resultar en el uso excesivo de recursos y una pérdida de

ingresos debido a la falta pago de las infracciones de tránsito vehicular (Gerencia de Transito Viabilidad y Transporte, 2018).

1.2 Descripción del problema

1.2.1 Problema general

La gestión de infracciones de tránsito en la Municipalidad Provincial del Cusco enfrenta desafíos significativos relacionados con la seguridad, integridad y transparencia de la información. El sistema tradicional presenta vulnerabilidades que pueden comprometer la eficiencia y confiabilidad del proceso.

1.2.2 Problemas específicos

- Inconsistencia de la información por el registro manual de las infracciones de tránsito.
- Falta de seguridad en la información registrada por ser vulnerable a la manipulación.

1.3 Objetivos

1.3.1 Objetivo general

Aplicar la tecnología Blockchain para la seguridad en la gestión de infracciones de tránsito en la Municipalidad Provincial del Cusco.

1.3.2 Objetivos específicos

1. Analizar los requisitos del uso de la tecnología Blockchain para la seguridad en la gestión de infracciones de tránsito de la Municipalidad Provincial de Cusco.

2. Implementar servicios web que consuman datos del Registro Nacional de Identificación y Estado Civil (RENIEC), de la Superintendencia Nacional de Registros Públicos (SUNARP) y del Ministerio de Transportes y Comunicaciones (MTC), para la veracidad de la información.
3. Adaptar la tecnología Blockchain para la seguridad en la gestión de infracciones de tránsito vehicular en la Municipalidad Provincial del Cusco.

1.4 Justificación

Este trabajo de investigación radica en la identificación de las limitaciones y desafíos actuales en los sistemas convencionales de registro de infracciones de tránsito. Los métodos tradicionales, pueden ser susceptibles a registro de datos irreales, manipulaciones y fraudes, comprometiendo la integridad de la información y socavando la confianza tanto de las autoridades como de los ciudadanos.

La tecnología blockchain, al ofrecer un enfoque inmutable para el almacenamiento y gestión de datos, promete abordar estas vulnerabilidades. La implementación de blockchain en el sistema de infracciones de tránsito tiene el potencial de garantizar la transparencia, la trazabilidad y la seguridad de los registros, lo que contribuirá a fortalecer la confianza en el sistema y a mejorar la eficiencia en la gestión de infracciones.

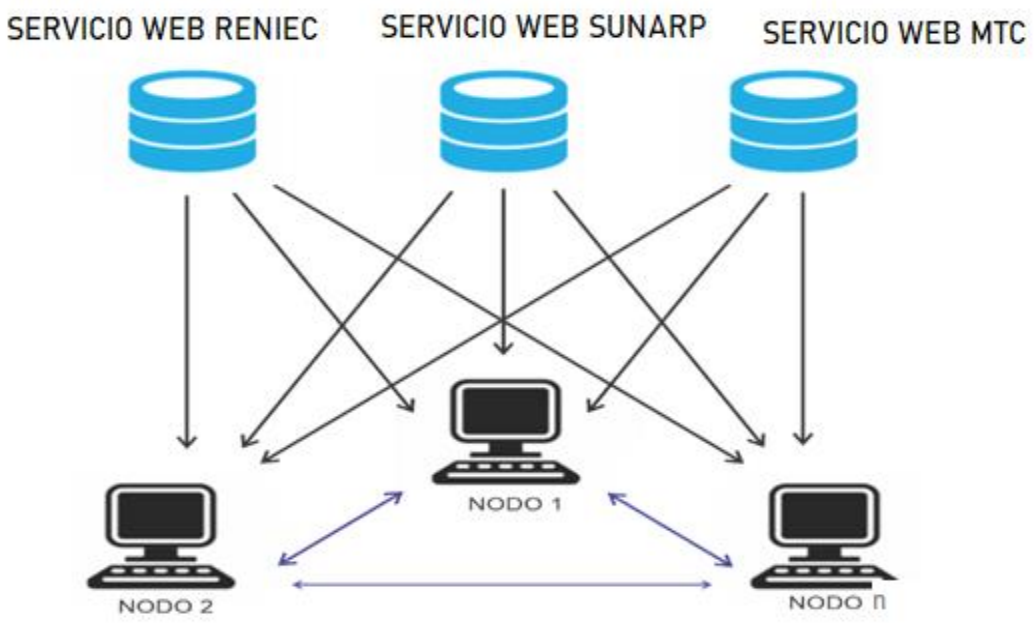
Además, la justificación se sustenta en la creciente adopción de blockchain en diversos sectores, destacando la necesidad de explorar y evaluar su viabilidad en el ámbito de la seguridad vial. La investigación propuesta busca, por tanto, llenar un vacío en la literatura científica al analizar críticamente la aplicabilidad de esta tecnología emergente en el contexto específico de las infracciones de tránsito.

En última instancia, este trabajo aspira a proporcionar una base teórica y empírica sólida que respalde la implementación de blockchain como una medida efectiva para mejorar la seguridad en el sistema de infracciones de tránsito, contribuyendo así al desarrollo de soluciones innovadoras y sostenibles en el campo de la seguridad vial.

1.5 Alcances

Este proyecto se realizará para la Gerencia de Tránsito, Viabilidad y Transporte de la Municipalidad Provincial del Cusco, con una propuesta de arquitectura distribuida.

Figura 1
Propuesta de Arquitectura



En la **Figura 1** se puede observar la arquitectura distribuida propuesta, siendo esto una característica del Blockchain, lo que permitirá que en cada nodo se cuente con la misma información, ya que esta se encontrará sincronizada. Así mismo, se encuentra conectada

mediante Servicios Web a la base de datos del Registro Nacional de Identificación y Estado Civil (RENIEC), de la Superintendencia Nacional de Registros Públicos (SUNARP) y del Ministerio de Transportes y Comunicaciones (MTC), de donde se obtendrán los datos de la persona, licencia de conducir y del vehículo.

1.6 Limitaciones

- Política de Reserva de Información: La Municipalidad Provincial del Cusco mantiene una política de reserva de información que restringe la divulgación completa de los datos almacenados en su base de datos. Esta política limita el acceso y la difusión de ciertos datos, lo que puede afectar la transparencia y la capacidad de realizar un análisis exhaustivo.
- Actualización de Datos: La actualización de los datos almacenados en la base de datos es un proceso que requiere tiempo. Dada la naturaleza progresiva de este proceso, que depende de la cantidad de datos que se agregan diariamente, la actualización completa de los datos con el sistema que se desarrollará puede llevar un tiempo considerable.

1.7 Metodología

La metodología de este proyecto se divide en dos componentes principales: la metodología de investigación y la metodología de desarrollo de software.

1.7.1 Metodología de investigación

La metodología de investigación se clasifica de la siguiente manera:

a) Por la forma en que la investigación es usada: El siguiente proyecto desea dar solución a los problemas que presenta la falta de seguridad en la gestión de

infracciones de tránsito en la Municipalidad Provincial del Cusco. Por lo tanto, es considerado como una investigación APLICADA. Siendo la definición: “La investigación Aplicada o Técnica tiende a la resolución de problemas o al desarrollo de ideas, dirigidas a conseguir innovaciones, mejoras de procesos o productos, etc.” (Sanchez, 2011).

b) Por el propósito del estudio: El presente proyecto realizará la identificación de las características más sobresalientes de la implementación de un Servicio Web con Blockchain, destacando los aspectos más sobresalientes para la mejora en la seguridad de gestión de infracciones de tránsito en la Municipalidad Provincial del Cusco. Por lo tanto, es considerada como DESCRIPTIVA. Siendo la definición: “Los estudios descriptivos buscan especificar las propiedades, las características y los aspectos importantes del fenómeno que se somete a análisis” (Gomez, 2006).

1.7.2 Metodología de desarrollo de software

Se aplicará la metodología de desarrollo “*Extreme Programming*”, que es una metodología ágil muy exitosa porque hace hincapié en la satisfacción del cliente y permite a los desarrolladores responder con confianza a las necesidades cambiantes de los clientes, incluso al final del ciclo de vida.

El *Extreme Programming*, que es un diagrama general que abarca todas las fases del proyecto, el cual tiene como inicio la administración, donde se encuentran las tareas necesarias de coordinación para la dotación del espacio de trabajo para el equipo en la organización y de supervisión de la aplicación en los parámetros de la metodología. A continuación, la fase de planeamiento en donde se recogen los requerimientos del

software o sistema informático. Luego la fase de diseño en la cual se realizará el modelamiento de la base de datos y las interfaces; y pasar a la fase de codificación en la cual se entregará al cliente la última versión del software desarrollado (Gamboa Manzaba, 2014).

CAPÍTULO II

MARCO TEORICO

2.1 Antecedentes

- Anicama López, Fernando Cristóbal (2019) *"Modelo de Blockchain para Mejorar la Toma de Decisiones en las Sentencias Fiscales del Ministerio Publico Lima 2019-2022"* (Tesis de postgrado) Universidad Nacional Federico Villarreal, Lima Perú (Anicama Lopez, 2019)**.**

  Conclusiones:

  - Se concluye que la implementación de un modelo Blockchain mejora la toma de decisiones de las sentencias fiscales del Ministerio Público y los tiempos de espera.
  - El proceso de estudio optimizado del modelo Blockchain mejora en la toma de decisiones de las sentencias fiscales del Ministerio Publico y los tiempos de espera.
  - La fiabilidad de la información del modelo Blockchain mejora en la toma de decisiones de las sentencias fiscales del Ministerio Publico.
  - Los gastos fiscales reducidos del modelo Blockchain mejora en la toma de decisiones de las sentencias fiscales del Ministerio Publico.

  Comentario:

  Este proyecto tuvo como principal objetivo implementar un modelo de Blockchain para la mejor toma de decisiones, determinando cuan fiable es el uso de este modelo, realizaron una comparación entre el sistema actual que manejaba el Ministerio Publico y el cambio propuesto en base a la utilización del Blockchain, permitiendo demostrar un mejor rendimiento del sistema que usa el Blockchain, ya que se reduce gastos y mejora los tiempos de espera, así mismo dicha tesis nos muestra los formularios principales del

sistema implementado así como también nos describe la implementación de la API, la cual se realizó mediante consultas POST al servidor que permite interactuar con el Blockchain a través de una serie de rutas.

- Espíritu Aranda, Walter Augusto y Machuca Nieva, Christian Fernando (2021) *"Modelo de Referencia para la Gestión de la Seguridad de Datos de Salud soportado en una Plataforma Blockchain"* (Para optar el título profesional de Ingeniero de Sistemas de Información) Universidad Peruana de Ciencias Aplicadas (Espíritu Aranda & Machuca Nieva, 2021).

  Conclusiones:

  - Los resultados obtenidos indican que los costos de implementar controles mitigantes en los centros de salud son elevados a comparación de utilizar la tecnología Blockchain la cual minimiza en su mayoría las brechas de seguridad, con relación a la cantidad de riesgos y vulnerabilidades encontrados en los sistemas de la clínica que albergan los datos de salud de pacientes.
  - Con la implementación del modelo de referencia, los centros de salud tienen una visión detallada y específica acerca de los posibles riesgos que podrían ocurrir, evitando problemas legales o sanciones económicas por parte del ente regulador.
  - El uso del modelo de referencia tiene un impacto positivo para la gestión de la seguridad en los centros de salud debido a que permite realizar un diagnóstico sobre los activos de información que tiene como objetivo conocer la criticidad de cada uno de ellos.

- Los resultados obtenidos indican que hay una disminución de un 26% en el nivel de riesgo con el uso de la tecnología Blockchain a comparación de usar un sistema tradicional.

Comentario:

En este proyecto de investigación proponen un modelo referencial para ser utilizados en instituciones de salud, ya que permite la gestión de la seguridad de datos sensibles y confidenciales apoyado en una plataforma Blockchain que integra el estándar ISO/IEC 27799, así mismo, mediante una evaluación del riesgo concluyen que el uso de la tecnología Blockchain disminuye el nivel del riesgo.

- Sebastián Andrés Sánchez Herrera (2021) *"Sistema de voto electrónico basado en Blockchain"* (Tesis Para optar por el Título de Ingeniero Informático) Pontificia Universidad Católica del Perú, Lima Perú (Sanchez Herrera, 2021)**.**

Conclusiones:

- El primer objetivo específico fue implementar un sistema de voto electrónico de código abierto que gestione la información del proceso electoral de forma descentralizada para los actores del proceso electoral. En base a la arquitectura diseñada resultante de las fases de análisis y diseño, se consiguió implementar un sistema de voto electrónico para elecciones generales en el Perú compuesto por una capa front-end en React Js y una capa de back-end en la tecnología Blockchain con un servicio de envío de correos desarrollado en Spring. Una vez escogida la arquitectura del sistema a utilizar, se definieron tres módulos que permitieron conseguir este primer objetivo específico. Estos tres módulos fueron el módulo de emisión de votos, escrutinio de los votos y mantener un repositorio del proyecto con

acceso libre para la comunidad. En tal sentido, se pudo definir toda la configuración del proceso electoral y la forma de interacción con el sistema mediante billeteras electrónicas que se basan en la red de Ethereum y se detallaron el uso de las transacciones en la aplicación descentralizada que permitían mantener un control de auditoría sobre el sistema. Así mismo, el concepto de cifra repartidora fue vital para el flujo del proceso de escrutinio ya que dentro del alcance se encontraba lo que es lista de votación con candidatos múltiples aplicado a elecciones congresales. Adicionalmente, cabe mencionar que el uso de "modifiers" fue de gran utilidad debido a que permitía validar que billetera podía interactuar con el sistema.

- Posterior a ello, el segundo objetivo específico fue implementar un algoritmo de cifrado que provea la integridad de los datos, en la cual se utilizó el algoritmo de cifrado ElGamal. Una característica muy importante para incrementar la seguridad que se quiso agregar en el sistema propuesto. Este tipo de algoritmo es muy utilizado en sistemas de procesos electorales y posee muchas variaciones. Para el uso del presente algoritmo se tuvo que crear la existencia de un nuevo rol, el de "autoridad de escrutinio", la cual cumple una función principal en el sistema para la encriptación y desencriptación del conteo de votos. Se concluye que el algoritmo a utilizar agrega seguridad al sistema y juntamente con el uso de la Blockchain permite darle al sistema una seguridad E2E ya que se tiene un control de auditoría de movimientos en el sistema y además la información de los votos registradas en la Blockchain se encuentran encriptados. Cabe mencionar que se pueden utilizar "n" autoridades de escrutinio y mientras más se incremente más robusto será la encriptación de los votos, sin embargo; está el otro punto en contra el cual es la cantidad de recursos

computacional que le toma al sistema procesar el algoritmo. Este último punto, es fundamental considerarlo ya que debido a que mientras más procesamiento computacional se realice más monedas electrónicas se utiliza en el sistema y esto puede incrementar en gran manera los costos del uso del sistema.

- El tercer objetivo específico fue utilizar estándares legales y técnicos en las fases de emisión, escrutinio y auditoría. En tal sentido, un resultado alcanzado fundamental fue la creación del catálogo de requerimientos, el cual constituye la base para la definición del nivel de seguridad que posee el sistema. Es por ello, que el catálogo de requerimientos sufrió tres versiones a lo largo del proyecto de tesis. Así mismo, se implementó el módulo de verificación individual de los votos y funcionalidades que permitan auditar el sistema. Con estas últimas características se pudo incrementar y finalizar el nivel de seguridad del sistema propuesto. En este objetivo más enfocado al control de cambios registrados en el sistema lo que permitían un nivel alto de auditoría del sistema por parte del elector como por parte del auditor.
- Finalmente, el gobierno electrónico en el cual se implementó el sistema fue las elecciones generales para procesos electores en el Perú. En tal sentido, el alcance del proyecto es a nivel nacional; sin embargo, es factible poder implementarlo en un gobierno electrónico distrital, donde se debería tener en consideración el lugar de residencia del elector, o en un nivel de gobierno electrónico de colegio de profesionales en el Perú en donde el uso de sistema de voto electrónico no presencial es más utilizado. Inclusive se podría agregar mayor nivel en la fase de configuración del proceso electoral para aceptar otros tipos de procesos electorales, tales como el referéndum.

Comentario:

Este proyecto propone el análisis, diseño e implementación de un sistema de voto electrónico para procesos electorales bajo estándares legales y técnicos que brinden transparencia y robustez en las fases de preparación, registro, votación, emisión de voto, escrutinio y auditoría aplicado a las elecciones generales en el Perú; para esto realizaron la implementación de un algoritmo de cifrado.

2.2 Marco teórico

2.2.1 Servicio web

2.2.1.1 Definición de un servicio web

- Un Servicio Web es un sistema que permite el intercambio de información entre distintos servidores por medio de la red, usando mensajes que cumplen un estándar (SOAP) basado en XML5.
- Un servicio Web es un sistema informático con el diseño para soportar la interoperabilidad institucional a través de una red de interacción. Con una interfaz definida en un formato procesable-máquina (específicamente WSDL).
- Un servicio Web es un conjunto de protocolos y estándares cuya función es intercambiar datos entre las distintas aplicaciones de software desarrolladas en los diferentes lenguajes de programación, y ejecutadas sobre cualquier plataforma (Grupo de Investigacion Davincis, 2021).

2.2.1.2 Ventajas de los servicios web

Dentro de las ventajas más importantes que se obtienen de los Servicios Web se pueden citar:

- Ofrecen de manera óptima la tecnología distribuida de componentes.
- Evitan problemas relacionados con las restricciones de firewalls, debido a que SOAP usa como protocolo de comunicación HTTP.
- Permiten recurrir de manera sencilla a los métodos, mediante SOAP.
- Los consumidores de servicios pueden estar desarrollados en cualquier plataforma (con tal que pueda soportar XML/SOAP, o ser reemplazado en su defecto con SOAP por HTTP)
- Permiten la centralización de datos, de manera independiente de que si los Servicio Webs sean distribuidos o no (Besteiro & Rodriguez, 2021).

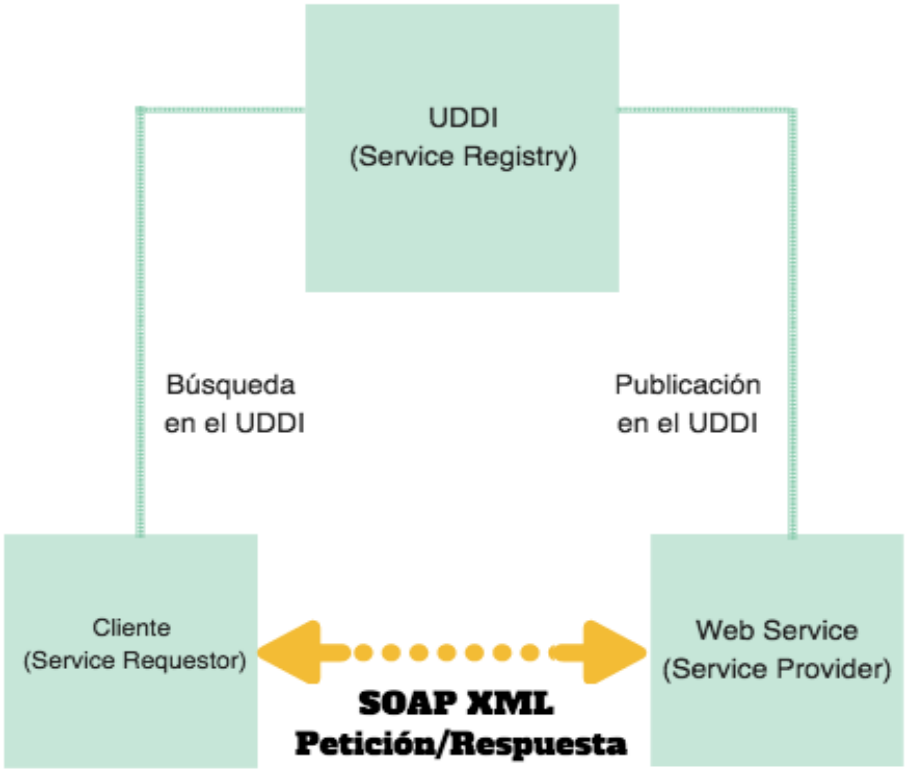
2.2.1.3 Componentes de los servicios web.

Los servicios web se ejecutan con los siguientes componentes, los cuales se pueden visualizar en la **Figura 2**:

- SOAP (Simple Object Access Protocol): Es un protocolo diseñado en XML. Es un formato para enviar mensajes, diseñado para servir de comunicación en red, pudiendo extender los HTTP. Permite definir qué información y como se envía mediante XML.
- WSDL (Web Services Description Language): Lenguaje basado en XML, Es el formato estándar que describe un Servicio Web y nos permite conocer el acceso a ellos. fue diseñado por Microsoft e IBM.

- UDDI (Universal Description Discovery and Integration): Es un estándar XML que nos permite desarrollar, publicar y localizar servicios web. Es un directorio que permite a las instituciones registrar y buscar servicios web. Contenedor de interfaces con servicios web descritos en WSDL los cuales se comunican mediante SOAP (Lazaro, 2021).

Figura 2
Componentes de los Servicios Web



Nota. Imagen tomada de Introducción a los Web Services, por Diego Lázaro, 2021

2.2.1.4 Arquitectura del servicio web

- Service Discovery. Centraliza los servicios web en un único directorio de registro, provee una funcionalidad de publicar y buscar de manera sencilla. UDDI es el encargado de Service Discovery.

- Service Description. Una de las características más destacada de los servicios web es que estos se auto describen. Esto quiere decir que, una vez localizado el Servicio Web, este nos brinda información sobre las operaciones que soporta y la forma de activarlo. Esto se ejecuta mediante el Web Service Description Language (WSDL).
- Service Invocation. La ejecución de un Servicio Web comprende el tráfico de mensajes entre el cliente y el servidor. SOAP nos precisa el cómo se debería formatear los mensajes request para el servidor, y la forma en que el servidor deberá formatear sus mensajes de respuesta.
- Transport. Los mensajes se transmiten de manera particular entre el servidor y el cliente. El protocolo elegido para la transmisión es HTTP. También se pueden utilizar los distintos protocolos, pero HTTP es actualmente el más usado (Lazaro, 2021).

2.2.1.5 Funcionamiento de un servicio web.

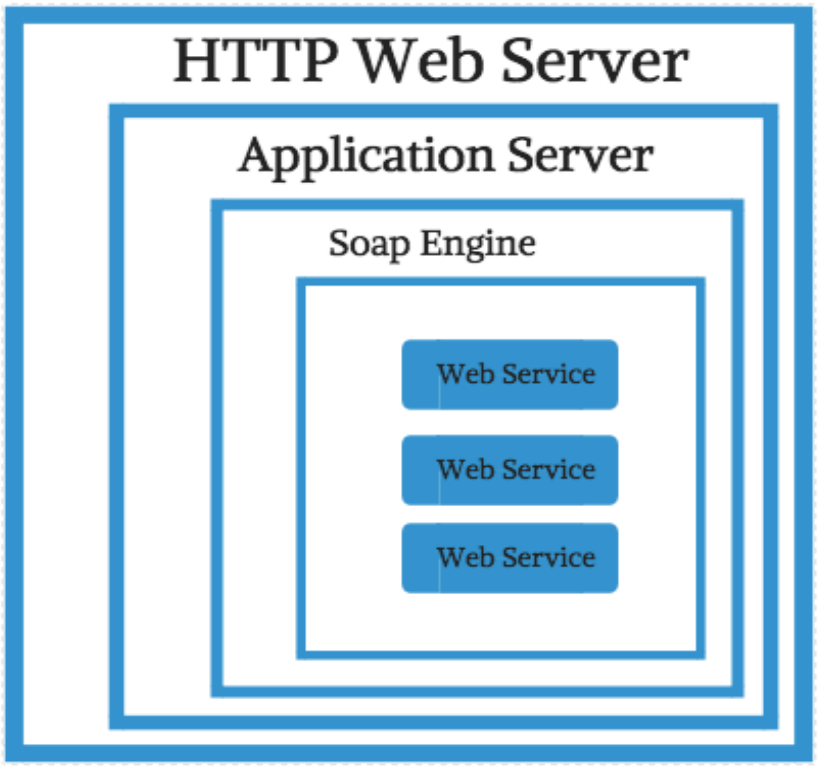
1. El proveedor de servicio genera el WSDL este determina el Servicio Web y registra el WSDL en el directorio UDDI.
2. La aplicación del cliente o solicitante del servicio requiere de un Servicio Web, entonces este se pone en contacto con el UDDI para ubicar el Servicio Web.
3. El cliente, se basa en la descripción del WSDL, envía una petición de servicio al Servicio Web oyente, el cual se encarga de recibir y enviar los mensajes en formato SOAP.

4. El Servicio Web examina el mensaje SOAP del pedido, luego invoca una operación en la aplicación para procesar dicho pedido. Este resultado se escribe nuevamente en SOAP en forma de respuesta para luego ser enviada al cliente.
5. El cliente examina el mensaje de respuesta SOAP y finalmente lo interpreta o de lo contrario genera un error de haber alguno (Lazaro, 2021).

2.2.1.6 Componentes de los servidores para las aplicaciones de servicio web

En la **Figura 3** se puede observar los componentes de los servidores en una Aplicación Web, los cuales se detallarán a continuación:

Figura 3
Componentes de los servidores en una Aplicación Servicio Web



Nota. Imagen tomada de Introducción a los Web Services, por Diego Lázaro, 2021

- Servicio Web. Es la aplicación o componente encargada de realizar las operaciones. Los clientes invocarán dichas operaciones mediante mensajes SOAP.
- Motor SOAP. El Servicio Web no es capaz de interpretar SOAP respuesta ni crear SOAP respuesta. Para realizar esta tarea hace falta un motor SOAP, un software encargado del manejo de estos mensajes.
- Servidor de aplicaciones. Para realizar la función de un servidor que puede recibir respuestas desde distintos clientes, el motor SOAP normalmente funciona dentro de un servidor de aplicaciones. Este es un software distinto cuya tarea es proporcionar un espacio libre para las aplicaciones que serán utilizadas por muchos clientes. El motor SOAP se ejecuta como una aplicación dentro del servidor de aplicaciones. Se tiene como ejemplo: Apache Tomcat server, Java Servlet y JSP container.
- El servidor HTTP. En algunos casos el servidor de aplicaciones incluye funcionalidades HTTP, por lo que se pueden tener Servicios Web funcionando, instalando simplemente un motor SOAP dentro del servidor de aplicaciones. Sin embargo, cuando un servidor de aplicaciones carece de funcionalidad HTTP es necesario también un servidor HTTP, Es un software que nos permite el manejo de mensajes HTTP. Los dos más populares en la actualidad son Apache HTTP Server y Nginx (Lazaro, 2021).

2.2.1.7 Windows communication foundation (WCF)

Windows Communication Foundation es un marco de trabajo para aplicaciones orientadas a servicios. WCF es un producto Microsoft que se incluye en el .NET Framework desde la versión 3.0. Está diseñado bajo un modelo unificado de programación y tiene por objetivo que los desarrollares puedan trabajar con distintos tipos de aplicaciones distribuidas sin tener que aprender distintos tipos de programación. Todo ello, es posible gracias a que WCF tiene una arquitectura orientada a servicios (SOA), la cual logra unificar varios modelos de comunicación disponibles en versiones anteriores de .NET (Servicios web, DCOM+, Remoting, MSMQ, ASMX, WSE). Asimismo, podemos agregar que WCF es un conjunto de librerías que provee el .NET Framework para la construcción de aplicaciones orientadas a servicios (TechClub, 2020).

WCF está compuesto por:

- **Clientes:** Aplicaciones que inician la comunicación.
- **Servicios:** Son aplicaciones que están a la espera de mensajes de los clientes y responden a los mismos. Estos mensajes son enviados entre endpoints (lugar donde un mensaje es enviado o recibido).

  Un servicio expone uno o más application endpoints y un cliente genera un endpoint compatible con uno de los endpoints del servicio dado.

  Esta combinación entre servicio y cliente compatible conforman un communication stack.

  Para poder crear un servicio se debe seguir los siguientes 5 pasos:

- Definir el contrato.
- Implementar el contrato.
- Configurar el servicio.
- Diseñar una aplicación hosting del servicio.
- Diseñar una aplicación cliente del servicio

Arquitectura Windows Communication Foundation

En la **Figura 4** se muestran las capas principales de la arquitectura Windows Communication Foundation (WCF) (Microsoft, 2020).

1. Contratos y descripciones:

Los contratos definen varios aspectos del sistema de mensajes. El contrato de datos describe cada parámetro que constituye cada mensaje que un servicio puede crear o utilizar. Los documentos de Lenguaje de definición de esquemas XML (XSD) definen los parámetros de mensaje, permitiendo a cualquier sistema que entienda XML procesar los documentos. El contrato del mensaje define partes específicas del mensaje utilizando los protocolos SOAP y permite el control más fino sobre las partes del mensaje, cuando la interoperabilidad exige tal precisión. El contrato de servicios especifica las firmas de método actuales del servicio y se distribuye como una interfaz en uno de los lenguajes de programación compatibles, como Visual Basic o Visual C#.

Las directivas y enlaces estipulan las condiciones exigidas para comunicarse con un servicio, como por ejemplo se debe especificar el transporte utilizado (HTTP o TCP) y una codificación (Microsoft, 2020).

Figura 4

Arquitectura de WCF

Nota. Imagen tomada de Arquitectura de Windows Communication Foundation, Microsoft, 2022

2. Tiempo de ejecución del servicio:

La capa del tiempo de ejecución del servicio contiene los comportamientos que solo se producen durante la operación actual del servicio, es decir, los

comportamientos en tiempo de ejecución del servicio. La limitación de peticiones controla cuántos mensajes se procesan que puede variar si la demanda para el servicio crece a un límite preestablecido. Un comportamiento de error especifica lo que sucede cuando se produce un error interno en el servicio, por ejemplo, controlando qué información se comunica al cliente. El comportamiento de la instancia especifica cuántas instancias del servicio se pueden. El comportamiento de la transacción habilita la recuperación de operaciones de transacción si se produce un error. El comportamiento de distribución es el control de cómo procesa un mensaje la infraestructura de WCF (Microsoft, 2020).

3. Mensajería:

La capa de mensajería se compone de canales. Un canal es un componente que procesa un mensaje de alguna manera, por ejemplo, autenticando un mensaje. Un conjunto de canales también se denomina pila de canales. Los canales funcionan en los mensajes y encabezados del mensaje. Esto es diferente de la capa en tiempo de ejecución del servicio, que se ocupa principalmente de procesar el contenido de los cuerpos de los mensajes.
Hay dos tipos de canales: canales de transporte y canales de protocolo.
Los canales de transporte leen y escriben mensajes de la red. Algunos transportes utilizan un codificador para convertir los mensajes hacia y desde la representación de la secuencia de bytes utilizada por la red. Son ejemplos de transportes HTTP, canalizaciones con nombre, TCP y MSMQ. Son ejemplos de codificaciones XML y binario optimizado.

Los canales de protocolo implementan protocolos de procesamiento de mensajes, a menudo leyendo o escribiendo encabezados adicionales en el mensaje. Los ejemplos de tales protocolos incluyen WS-Security y WS-Reliability.

La capa de la mensajería muestra los posibles formatos y patrones de intercambio de los datos. WS-Security es una implementación de la especificación WS-Security que habilita la seguridad en la capa del mensaje. El canal de mensajería WS-Reliable habilita la garantía de entrega del mensaje. Los codificadores presentan una variedad de codificaciones que se pueden utilizar para satisfacer las necesidades del mensaje. El canal HTTP especifica que el Protocolo de transporte de hipertexto se utiliza para la entrega del mensaje. El canal TCP especifica de manera similar el protocolo TCP. El canal de flujo de transacciones rige los patrones de mensajes de transacción. El canal de la canalización con nombre habilita la comunicación entre procesos. El canal de MSMQ habilita la interoperación con aplicaciones MSMQ (Microsoft, 2020).

4. Alojamiento y activación:

En su forma final, un servicio es un programa. Como otros programas, un servicio se debe ejecutar en un ejecutable. Esto se conoce como servicio auto hospedado.

Los servicios también se pueden hospedar o ejecutar en un ejecutable administrado por un agente externo, como IIS o el servicio de activación de Windows (WAS). WAS permite que las aplicaciones WCF se activen

automáticamente cuando se implementan en un equipo que ejecuta WAS. Los servicios también se pueden ejecutar manualmente como ejecutables (archivos .exe). Un servicio también se puede ejecutar automáticamente como un servicio de Windows. Los componentes de COM+ también se pueden hospedar como servicios WCF (Microsoft, 2020).

2.2.2 Blockchain

2.2.2.1 Definición de blockchain

Es una cadena de bloques equivalente a un libro mayor contable (“ledger”) o una base de datos en la que se registran muchas operaciones transaccionales. Tiene la característica de ser distribuido, ya que se encuentra dividido en varias máquinas, en donde la información esta reproducida exactamente igual. Dicha cadena trabaja en una red P2P («peer-to-peer», es decir «igual-a-igual») sin la participación de terceros. Inicialmente se encriptan todos los datos, asignando un "hash “que es una forma de firma codificada basada en el contenido. La cadena contiene la información de cada nueva operación que se realice, originando un nuevo bloque, el que se replica en todas las maquinas en las que se encuentra incluido (“nodos”) y al instante. Cada vez que se crea un nuevo bloque, la cadena se renueva y transfiere la totalidad de la información, esta no se puede cambiar ni eliminar (ni repetir, ya que esta sería una nueva transacción) gracias a la encriptación y el sistema de firmas digitales: El nuevo bloque debe ser validado (lo cual puede hacerse de diversas maneras) y solo pasa a formar parte de la cadena después de su validación. En un sistema basado en Blockchain, la manipulación o

falsificación de datos no es posible debido a que los hashes no coincidirían; se demuestra siempre matemáticamente la integridad de los datos. En breve:

1. Todo el contenido es encriptado y verificado.
2. Una vez aceptado, es imposible modificar un bloque sin invalidar toda la cadena.
3. Una copia de la cadena se conserva automáticamente en los computadores de quienes la utilizan (pero en el futuro podría haber copias parciales, para evitar un crecimiento constante) (Dr. Raymond Colle, 2021).

2.2.2.2 Principales características de blockchain

1. Inmutabilidad: Es una de las características claves de la tecnología; significa que algo no puede ser cambiado o alterado, lo que nos garantiza que la tecnología continúe igual: una red permanente e inalterable. El Blockchain funciona relativamente distinto al sistema bancario tradicional, ya que esta garantiza las características del Blockchain a través de una colección de nodos. Dicho proceso consiste en que cada nodo contiene una copia del registro digital. Y para cada que se quiera adicionar una transacción, todos los nodos verificaran su validez. Si el mayor porcentaje entiende que es correcto, recién se adicionara el registro. Promoviendo la transparencia e inalterable.
2. Descentralización: Esto significa que ninguna persona manda o controle totalmente el proceso, todo lo contrario, un conjunto de nodos preserva la red descentralizada.

Como el sistema no es controlado por una única persona, los usuarios pueden acceder directamente desde la web y almacenar activos; desde criptomonedas, documentos importantes, contratos u otros activos digitales importantes. Y ya que se cuenta con una clave primaria en Blockchain, se tendrá control directo sobre la información, dándole poder a los usuarios sobre sus activos. Lo cual genera:

- **Menos fallos:** Blockchain está completamente organizado y no depende de cálculos humanos, lo que lo hace tolerante a fallos.
- **Control de usuario:** Con la descentralización, los usuarios ahora pueden controlar sus activos sin la necesidad de depender terceros.
- **Menos propenso a averías:** La descentralización resulta más difícil de vulnerar, ya que atacar el sistema puede ser costoso para los hackers y difícil de solucionar.
- **Sin terceros:** La descentralización de la tecnología nos da la ventaja de no depender de empresas terceras.
- **Cero estafas:** No existe la posibilidad de que las personas puedan ser estafadas ya que el sistema es implementado con algoritmos.
- **Transparencia:** Debido a que la tecnología es descentralizada se tiene un perfil transparente de cada usuario, los cambios en el Blockchain son visibles y más concretos.
- **Naturaleza auténtica:** Este sistema es único y puede ser utilizado por todo tipo de personas, así mismo a los hackers se les hará difícil descifrarlo.

3. Seguridad mejorada: La utilización del cifrado nos proporciona otra capa de seguridad para el sistema, garantizando más protección a la información de los usuarios. La criptografía es un algoritmo matemático más complejo que sirve como un firewall para ataques. Toda la información contenida en el Blockchain es cifrada criptográficamente. Para este proceso, cualquier dato de entrada pasa por un algoritmo matemático que produce un tipo diferente de valor; pero manteniendo la longitud siempre fija. Podrías considerarlo como una identificación única para cada dato. Todos los bloques en el registro vienen con un hash único y contienen el hash del bloque anterior. Por lo tanto, cambiar o intentar manipular los datos significa cambiar todas las ID de hash. Y hacerlo con la tecnología existente, resulta imposible. Tendrás una clave privada para acceder a los datos; pero también tendrás una clave pública para realizar transacciones (Valinsky, 2019).
4. Encriptación simétrica: En este tipo de encriptación es utilizada la misma clave para encriptar y desencriptar el mensaje (Anderson, Narus, Narayandas, & Seshadri, 2011). Debido a su mayor velocidad, la encriptación simétrica es empleada de forma generalizada para la protección de información en muchos sistemas de computación modernos.

En la **Figura 5** tenemos un ejemplo de la Encriptación Simétrica.

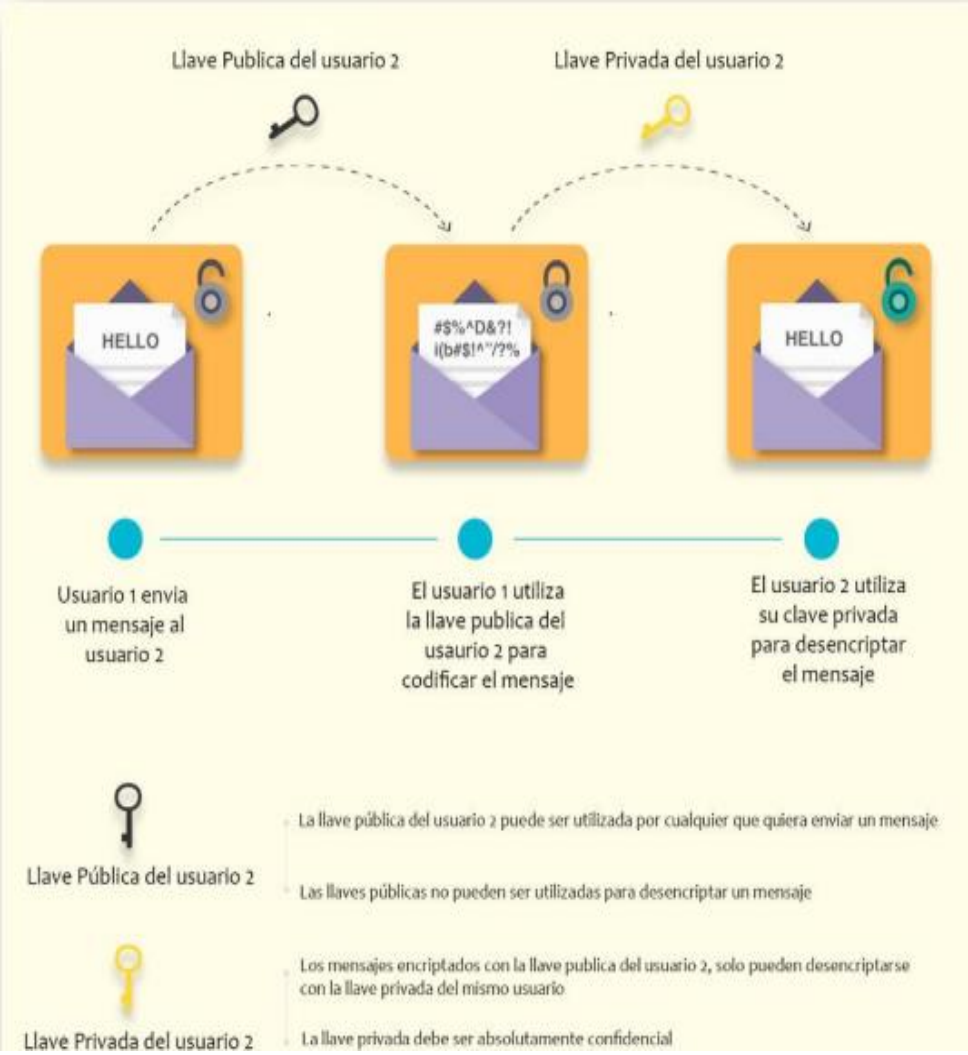
Figura 5
Encriptación Simétrica

Nota. Imagen tomada de La identidad digital en la internet futura con Blockchain, Dr. Raymond Colle, 2021

5. Encriptación asimétrica: Se utilizan distintas claves (públicas y privadas), que sirven para encriptar y desencriptar. Este tipo de encriptación se aplica en sistemas en los que los usuarios requieren encriptar y desencriptar mensajes o conjuntos de datos, en especial cuando la velocidad y la potencia computacional no son prioridad.

   En la **Figura 6** se tiene un ejemplo de la Encriptación asimétrica.

Figura 6
Encriptación Asimétrica



Nota. Imagen tomada de La identidad digital en la internet futura con Blockchain, Dr. Raymond Colle, 2021

6. Irreversible: Los hashes son complejos, y difíciles de modificar o revertir. No es posible crear una clave privada tomando como referencia la clave pública, ya que un cambio en la entrada nos llevaría a un ID muy diferente. Si alguien quiere corromper la red, deberá modificar todos los datos almacenados en cada nodo de la red.

7. Registros distribuidos: Por lo común, un registro público nos proporciona información de una transacción y del participante. Aunque en el Blockchain esto es un poco diferente, en algunos casos algunas personas pueden ver lo que está pasando en el registro, ya que este se mantiene registrado en la red. Distribuyendo el poder computacional para garantizar un mejor resultado, el cual siempre será un registro más eficiente. El registro distribuido responde bien ante alguna actividad sospechosa o intento de manipulación. Como no es posible modificar el registro y este se actualiza muy rápido, es fácil realizar el seguimiento de lo que sucede en el registro.
8. Verificación de propiedad: En esta parte, los nodos se comportan como verificadores del registro, es decir, si un usuario quiere adicionar un nuevo bloque, otro tiene que verificar la transacción y dar el visto bueno, proporcionando una participación justa. Nadie en la red puede obtener favores especiales de la red, todos tienen que seguir el mismo proceso para agregar sus bloques.
9. Consenso: El consenso es un proceso de toma de decisiones para los nodos activos de la red, su arquitectura fue diseñada inteligentemente y son los algoritmos el centro de esta arquitectura. Los nodos llegan a un acuerdo de forma sencilla y más rápida, ya que cuando estos están validando una transacción, es necesario un consenso para que el sistema funcione sin problemas. Puede suceder que los nodos no confíen entre sí; pero si confían en los algoritmos que son ejecutados en su núcleo.

2.2.2.3 Tipos de blockchain

Los tipos de Blockchain se pueden clasificar en función del acceso a los datos, la distinción entre los tipos de Blockchain es el esquema del libro distribuido y quién puede participar en el sistema (Viriyasitavat & Hoonsopon, 2018).

En la **Figura 7** podemos observar las principales características del Blockchain según el tipo.

Blockchain Públicas: son de tipo abierto, en el que cualquiera puede participar. Todos los participantes pueden acceder libremente a datos y realizar transacciones, pero dado que numerosos usuarios no verificados están participando, se necesita cifrado y verificación avanzada, por lo tanto, la expansión de la red se torna lenta y difícil. Además, el Blockchain público forma una perfecta estructura distribuida, y los participantes de la red son pseudoanónimos, por lo tanto, no es apropiado para los servicios financieros que necesitan ser controlados por la información centralizada de los sistemas de gestión (Oh & Shong, 2017). También permiten que cualquiera acceda y mantenga el libro mayor distribuido con permisos para validar la integridad ejecutando un mecanismo de consenso. Una red pública de Blockchain está completamente abierta y distribuida; cualquiera puede unirse, participar y abandonar el sistema libremente (Viriyasitavat & Hoonsopon, 2018).

Blockchain Privadas: en ella el propietario genera y maneja el Blockchain. Esto es apropiado si el propietario desea administrar la Blockchain como el sistema centralizado (Oh & Shong, 2017). Los libros contables son

compartidos y validados por un grupo predefinido de nodos. El sistema requiere iniciación o validación a los nodos que desean ser parte del sistema. Los nodos autorizados son responsables de mantener el consenso. Blockchain privadas son adecuadas para sistemas cerrados, donde todos los nodos son completamente confiables. En definitiva, es el propietario quien tiene la máxima autoridad para controlar el acceso a nodos autorizados (Viriyasitavat & Hoonsopon, 2018).

Blockchain Hibridas (Consorcio): es el tipo intermedio de Blockchain pública y privada. A diferencia de Blockchain Privadas en el que el propietario tiene la autoridad, son los nodos preestablecidos quienes tienen la autoridad en este tipo de Blockchain. Por lo tanto, Blockchain Hibridas mantienen una estructura distribuida al mismo tiempo que fortalece la seguridad mediante una participación limitada, y resuelve el problema de la lenta velocidad de transacción y los problemas de escalabilidad de la red planteados en Blockchain Pública. Por lo tanto, Blockchain Hibridas podrían ser utilizadas para transacciones entre instituciones financieras (Oh & Shong, 2017).

La Blockchain hibrida es adecuada para sistemas semicerrados compuestos por unas pocas empresas, a menudo organizadas en forma de consorcio. El grado de apertura de los datos varía, por lo general con controles de acceso, definidos por el consorcio, para controlar el acceso en ambos participantes y la información dentro de Blockchain. A pesar de que el sistema no está

completamente abierto, los beneficios de la descentralización se pueden obtener parcialmente (Oh & Shong, 2017).

Figura 7

Características según tipo de Blockchain

Tipos de Blockchain	Blockchain Publica	Blockchain Híbrida	Blockchain Privada
Entidad gestora	Todos los participantes (descentralización)	Participantes que pertenezcan al consorcio.	Una institución central tiene toda la autoridad.
Gobernanza	Es muy difícil cambiar la regla que se ha hecho.	Las reglas podrían cambiarse con relativa facilidad de acuerdo con el acuerdo entre los participantes del consorcio.	Las reglas podrían cambiarse fácilmente de acuerdo con la decisión tomada por la institución central.
Velocidad de transacción	Es difícil expandir la red y la velocidad de transacción es lenta.	Es fácil expandir la red y la velocidad de transacción es rápida.	Es muy fácil ampliar la red y la velocidad de transacción es rápida.
Acceso a los datos	Cualquiera puede acceder	Solo usuarios autorizados pueden acceder	Solo usuarios autorizados pueden acceder
Identificabilidad	Seudo-anónimo	Identificable	Identificable
Prueba de transacción	La entidad para la prueba de la transacción se decide mediante algoritmos como PoW y PoS, y no se puede conocer de antemano.	La entidad para la prueba de la transacción se conoce a través de la autenticación, y la verificación de la transacción y generación de bloques se realizan de acuerdo con las reglas acordadas de antemano.	La prueba de transacción es realizada por la institución central.
Casos de utilización	Bitcoin	R3CEV	Linq, una plataforma de mercado bursátil para compañías sin cotización NASDAQ

Nota. Imagen tomada de A case study on business model innovations using Blockchain, Dr. Raymond Colle, 2021

A diferencia de Blockchain Publicas, que proporciona pseudoanonimato, es posible identificar el sujeto en Blockchain Privadas. Las transacciones se manejan rápidamente, la expansión de la red es fácil y podría ser modificado de la manera que el usuario desee, por lo tanto, es adecuado para servicios financieros. Por lo tanto, está recibiendo atención de las compañías e instituciones (Oh & Shong, 2017).

2.2.2.4 Funcionamiento del blockchain

Las cadenas de bloques pueden ser privadas (“sujetas a permiso”), públicas, es decir de acceso abierto (“permissionless”) o híbridas. Si una cadena es privada, cada usuario autorizado tendrá una identificación única que le dará acceso a la información detallada, lo que eleva aún más el nivel de protección de datos. El sistema también permite que se creen accesos diferenciados, permitiendo a algunos ver parte de la información que otros no podrán ver: Si A transfiere algo a B, ambos pueden ver todos los detalles, pero C podría tener permiso para ver que hubo una operación, pero no para observar los demás datos (como son el monto, moneda, etc.).

Figura 8

Como funciona Blockchain

Nota. Imagen tomada de la identidad digital en la internet futura con Blockchain, Dr. Raymond Colle, 2021

En la **Figura 8** se ve el proceso del funcionamiento del Blockchain. Si bien la tecnología fue creada pensando en monedas virtuales (inicialmente el

Bitcoin), se ha descubierto su utilidad en numerosos otros sectores. Hoy, se utiliza para todo: para pagos globales, registrar y compartir música, seguir las ventas y la cadena logística, sistemas electorales, etc. Por todo esto el Blockchain tiene la capacidad de ser utilizado de diferentes formas: ya que puede colocarse de todo en una Blockchain (Marvin, 2017). Y se dice que podrá llegar a ser la base de una nueva internet" (Dr. Raymond Colle, 2021).

2.2.2.5 Plataforma blockchain

Las plataformas Blockchain pueden ser sin permiso o con permiso. En una cadena de Blockchain pública, sin permisos, como Bitcoin, todos los nodos de la red pueden realizar transacciones y participar en el proceso de consenso. En una cadena privada, permitida como Multichain, cada nodo podría ser capaz de realizar transacciones, pero la participación en el proceso de consenso se limita a un número restringido de nodos aprobados (Rouse & TechTarget, 2021).

2.2.2.6 Algoritmos de consenso/validación de blockchain

Escoger el algoritmo de consenso que se debe usar es lo principal en la elección de una plataforma de la cadena de bloques. Se cuenta con cuatro métodos estándares para la aplicación del Blockchain y diversas plataformas de bases de datos distribuidas que nos permita alcanzar un acuerdo. Por lo general, las plataformas públicas escogen algoritmos como 'Prueba de trabajo'. Sin embargo, existen otros nodos de red que permiten comprobar con menos esfuerzo, como son: (Rouse & TechTarget, 2021)

- Algoritmo de prueba de trabajo (PoW)

- Algoritmo práctico de tolerancia a errores bizantinos (PBFT)
- Algoritmo de prueba de participación (PoS)
- Algoritmo de prueba de participación delegado (DPoS)

2.2.2.7 Ventajas y desventajas del blockchain

Los expertos citan varios beneficios clave para usar Blockchain. La seguridad es considerada una de las principales ventajas de esta tecnología. Es casi imposible corromper una cadena bloqueada porque la información es compartida y continuamente reconciliada por miles, incluso millones de computadoras, y Blockchain no tiene un solo punto de falla. Si un nodo se cae, no es un problema porque todos los otros nodos tienen una copia.

Por otro lado, los expertos dicen que Blockchain también tiene posibles inconvenientes, riesgos y desafíos. Con las cadenas de bloques públicas, hay preguntas sobre la confianza y quién es responsable en caso de que surja un problema. Con las cadenas de bloqueo privadas, hay preguntas sobre si las organizaciones son capaces o están dispuestas a invertir en la infraestructura para devolución de cargo de TI, una estrategia contable que aplicaría los costos de los servicios de TI, como las transacciones de base de datos, a la unidad de negocios en la que se utilizan (Rouse & TechTarget, 2021).

2.2.3 Gestión de Riesgo

2.2.3.1 Riesgo

Se entiende por riesgo de seguridad informática, toda amenaza que explote alguna vulnerabilidad de uno o varios activos y pueda afectar el funcionamiento de un sistema, teniendo en cuenta la probabilidad que ocurra

el evento y el impacto en caso de materializarse, en alguna de las tres características principales de la seguridad de la informática, los cuales son: Integridad, Confidencialidad y Disponibilidad (Pinzon, s.f.).

2.2.3.2 Riesgos identificados

- Ingreso errado de datos por parte del personal encargado: Este riesgo identificado se refiere a que el personal de la Gerencia de Tránsito, Viabilidad y Transporte ingresa información al sistema de forma manual, lo que puede ocasionar que algún dato de la papeleta este errado, como puede ser el DNI, número de placa, etc.
- Fuga de información por parte del personal con acceso al sistema: Ya que actualmente el personal de la Gerencia de Tránsito, Viabilidad y Transporte tiene acceso a la información del sistema, se corre el riesgo de que algún personal deshonesto pueda divulgar de forma intensional algún detalle de los infractores o de los vehículos.
- Manipulación y alteración de datos: Debido a que actualmente el personal puede acceder a la información de la base de datos, se tiene el riesgo de que alguna persona deshonesta manipule o altere algún dato registrado de las infracciones, afectando la integridad de la información.
- Caída del servicio e indisponibilidad del sistema: La caída del servicio no solo provocara la afectación de las operaciones propias, sino también el corte de servicio a los usuarios externos, esto también puede ocasionar ataques al sistema informático de la Gerencia de Tránsito, Viabilidad y Transporte.

2.2.4 La Institución: Municipalidad provincial del Cusco

2.2.4.1 Misión institucional

Brindar servicios con eficiencia, eficacia, transparencia y tecnología beneficiando al ciudadano, de esta manera lograr un desarrollo integral y sostenible de la ciudad del Cusco, a través de una gestión participativa y renovadora (Cusco, 2019).

2.2.4.2 Visión institucional

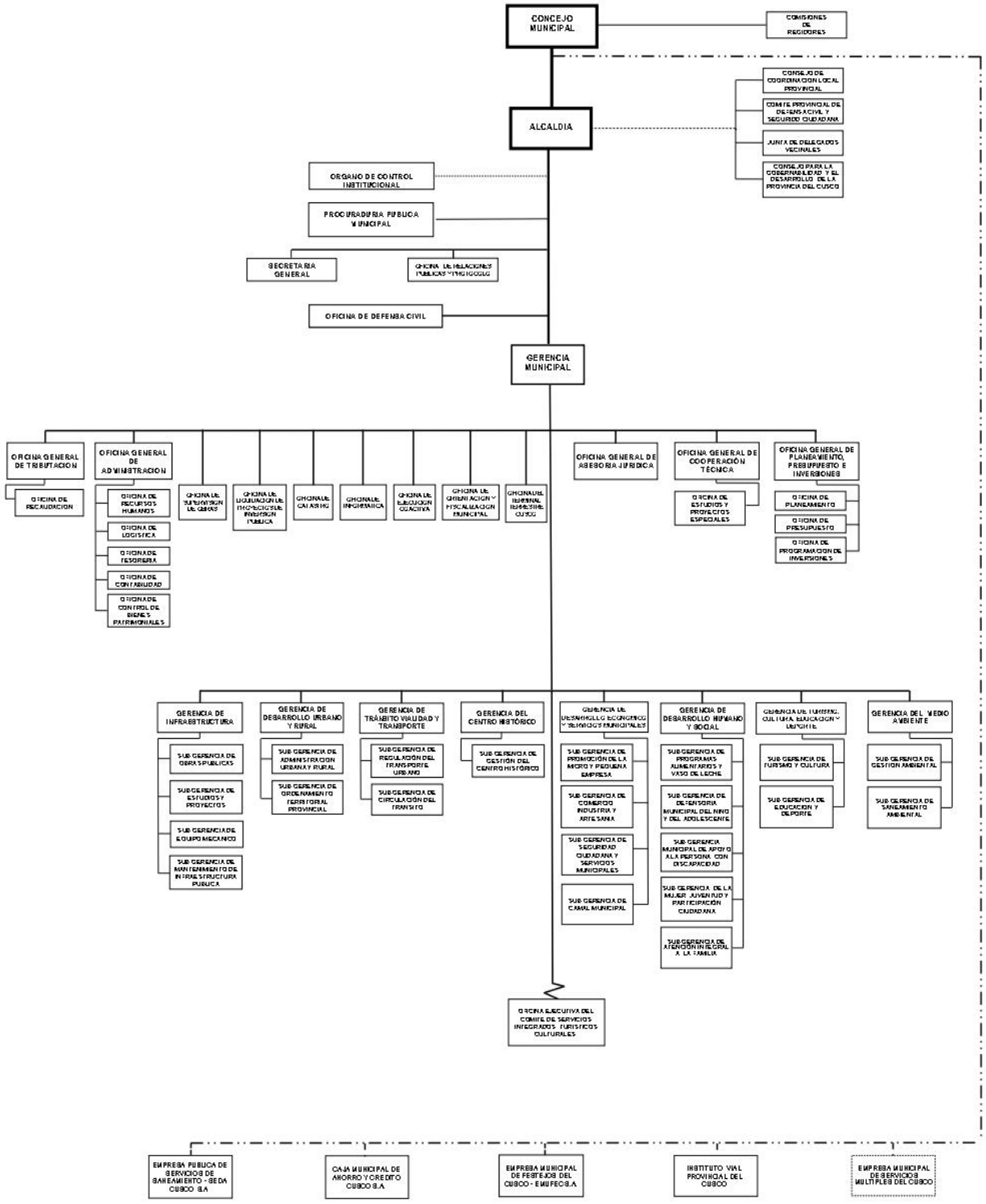
Ser una Municipalidad modelo que impulsa el desarrollo integral de la comunidad, con una gestión eficiente, transparente y con una activa participación ciudadana, posicionando a Cusco como una ciudad saludable, segura, ordenada e inclusiva donde se fomente la cultura (Cusco, 2019).

2.2.4.3 Organigrama

En la **Figura 9** tenemos el Organigrama Estructural de la Municipalidad Provincial del Cusco.

Figura 9

Organigrama Estructural de la Municipalidad Provincial del Cusco



Nota. Imagen tomada de Gobierno Municipal del Cusco, de Municipalidad del Cusco, 2019

2.2.4.4 Gerencia de tránsito vialidad y transporte

La Gerencia de Tránsito Vialidad y Transporte, es un órgano de línea de segundo nivel organizacional, responsable de supervisar, conducir, planificar y administrar los procesos de regulación del transporte terrestre urbano; así como la circulación del tránsito, dentro del ámbito de su jurisdicción. Está a cargo de un Gerente, quien depende del Gerente Municipal (Cusco, 2019). Son funciones y atribuciones de la Gerencia de Tránsito Vialidad y Transporte las siguientes:

1. Normar, regular y planificar el transporte terrestre urbano.
2. Proponer lineamientos, objetivos, reglamentos y planes de acción en materia de tránsito y transporte, para el desarrollo de las actividades de su competencia en concordancia con las disposiciones legales vigentes.
3. Formular, ejecutar, evaluar su Plan Operativo y el Cuadro de Necesidades.
4. Formular, la memoria anual de la dependencia, de acuerdo a la normatividad vigente.
5. Formular su cronograma de gastos anual mensualizado, de acuerdo a la normatividad vigente.
6. Planificar y administrar la gestión del transporte terrestre urbano e interurbano, tránsito urbano de peatones y vehículos.
7. Ejecutar, controlar y/o actualizar el Plan Regulador de Rutas de la Provincia del Cusco.

8. Programar, dirigir, ejecutar y supervisar el proceso de administración de infracciones y sanciones, dentro de la jurisdicción de la provincia.
9. Elaboración de informes mediante el área correspondiente, respecto a infracciones al Registro Nacional de Tránsito.
10. Dirigir y administrar el Depósito Municipal Vehicular, reglamentando sus funciones y acciones afines.
11. Programar, dirigir, ejecutar, coordinar y controlar las actividades referidas al otorgamiento de concesiones o autorizaciones para la prestación del servicio regular de transporte público urbano de pasajeros.
12. Regular el servicio público de transporte urbano de la Provincia del Cusco, de conformidad con las leyes, reglamentos nacionales y ordenanzas sobre la materia.
13. Regular, organizar, administrar y mantener los sistemas de señalización y semaforización para el tránsito urbano de peatones y vehículos.
14. Regular, organizar, administrar y mantener los sistemas viales para el tránsito urbano de peatones y vehículos.
15. Regular el transporte público urbano y otorgar las correspondientes autorizaciones o concesiones de rutas en sus diferentes modalidades, previo informe Técnico y Legal de su dependencia.
16. Ejercer la función de supervisión del servicio público de transporte urbano provincial, contando con el apoyo de la Policía Nacional asignada al control del tránsito.

17. Planificar y organizar el proceso de instalación, mantenimiento y renovación de los sistemas de señalización de tránsito en su jurisdicción.
18. Calificar y revisar l| as infracciones de tránsito vehicular.
19. Orientar e incentivar la participación de los inversionistas en el mejoramiento del servicio de transporte urbano en sus diferentes modalidades.
20. Expedir autorizaciones para la circulación de vehículos para el Transporte público, escolar, turismo, taxis, carga y descarga.
21. Emitir Resoluciones de sanción, en los procedimientos de fiscalización implementados de conformidad a las disposiciones legales y municipales pertinentes, previo informe Técnico y Legal de su dependencia.
22. Mantener actualizado y sistematizado el archivo del acervo documentario a su cargo y de la información básica y estadística de la Gerencia.
23. Emitir Resoluciones Gerenciales de acuerdo a sus funciones y competencias, previo informe Técnico y Legal de su dependencia.
24. Revisar los expedientes técnicos a ser ejecutados por su Gerencia, a través de sus respectivos órganos estructurales, de acuerdo a la normatividad del Sistema Nacional de Inversión Pública – SNIP.
25. Otras funciones que le asigne la Gerencia Municipal.

La Gerencia de Tránsito Vialidad y Transporte, para el cumplimiento de sus objetivos y funciones tiene establecido la estructura siguiente:

- Sub Gerencia de Regulación del Transporte Urbano.
- Sub Gerencia de Circulación de Tránsito.

2.2.4.5 Texto único ordenado del reglamento nacional de tránsito – código de transito

Artículo 1.- Objeto y ámbito. El presente Reglamento establece normas que regulan el uso de las vías públicas terrestres, aplicables a los desplazamientos de personas, vehículos y animales y a las actividades vinculadas con el transporte y el medio ambiente, en cuanto se relacionan con el tránsito. Rige en todo el territorio de la República (SUTRAN, 2014).

Artículo 3.- Autoridades competentes. Son Autoridades competentes en materia de tránsito terrestre:

1) El Ministerio de Transportes y Comunicaciones.

2) Las Municipalidades Provinciales.

3) Las Municipalidades Distritales.

4) La Policía Nacional del Perú.

5) El Instituto Nacional de Defensa de la Competencia y de la Protección de la Propiedad Intelectual – INDECOPI (SUTRAN, 2014).

Artículo 5.- Competencias de las municipalidades provinciales. En materia de tránsito terrestre, las Municipalidades Provinciales en su respectiva jurisdicción y de conformidad con el presente Reglamento tienen las siguientes competencias:

1) Competencias normativas: Emitir normas y disposición complementarias necesarias para la aplicación del presente Reglamento dentro de su respectivo ámbito territorial (SUTRAN, 2014).

2) Competencias de gestión

a) Administrar el tránsito de acuerdo al presente Reglamento y las normas nacionales complementarias;

b) Implementar y administrar los registros que el presente Reglamento establece;

c) Recaudar y administrar los recursos provenientes del pago de multas por infracciones de tránsito;

d) Instalar, mantener y renovar los sistemas de señalización de tránsito en su jurisdicción, conforme al presente Reglamento.

3) Competencia de fiscalización

a) Supervisar, detectar infracciones e imponer sanciones por el incumplimiento de las disposiciones del presente Reglamento y sus normas complementarias

b) Mantener actualizado el Registro Nacional de Sanciones por Infracciones al Tránsito Terrestre, en el ámbito de su competencia, conforme a lo dispuesto en el presente Reglamento (SUTRAN, 2014).

Artículo 6.- Competencias de las municipalidades distritales. Las Municipalidades Distritales en materia de tránsito terrestre, ejercen funciones de gestión y fiscalización, en el ámbito de su jurisdicción, en concordancia con las disposiciones que emita la Municipalidad Provincial respectiva y las previstas en el presente Reglamento. En materia de vialidad, la instalación, mantenimiento y renovación de los sistemas de señalización de tránsito en su jurisdicción, conforme al Reglamento correspondiente (SUTRAN, 2014).

CAPÍTULO III

IMPLEMENTACION DEL SERVICIO WEB

3.1 Requerimientos del sistema

3.1.1 Requerimientos funcionales

- El módulo permitirá el ingreso de la placa del vehículo.
- El módulo permitirá el ingreso de la licencia de conducir o el número de DNI del conductor.
- El módulo permitirá el ingreso del código policía o nombre competo del policía que impuso la infracción.
- El módulo permitirá el ingreso de la siguiente información sobre la infracción: nombre de la infracción, grupo de la infracción, reglamento, tipo, código infracción, vía y observaciones.
- El sistema, obtendrá los datos del vehículo (marca, modelo, clase, color, DNI del propietario) consumiendo datos, desde el Servicio Web de la Superintendencia nacional de registros públicos (SUNARP).
- El sistema, obtendrá los datos del propietario (nombres completos, dirección) consumiendo datos, desde el Servicio Web del registro nacional de identificación y estado civil (RENIEC).
- El sistema, obtendrá los datos del conductor (nombres completos, clase licencia, categoría, dirección) consumiendo datos, desde el Servicio Web del Ministerio de Transportes y Comunicaciones (MTC).

3.2 Implementación del servicio web

3.2.1 Herramientas

Se utilizó el lenguaje C# .NET, sobre la plataforma *Windows Communication Foundation* (WCF) para su desarrollo, debido a las ventajas que ofrece.

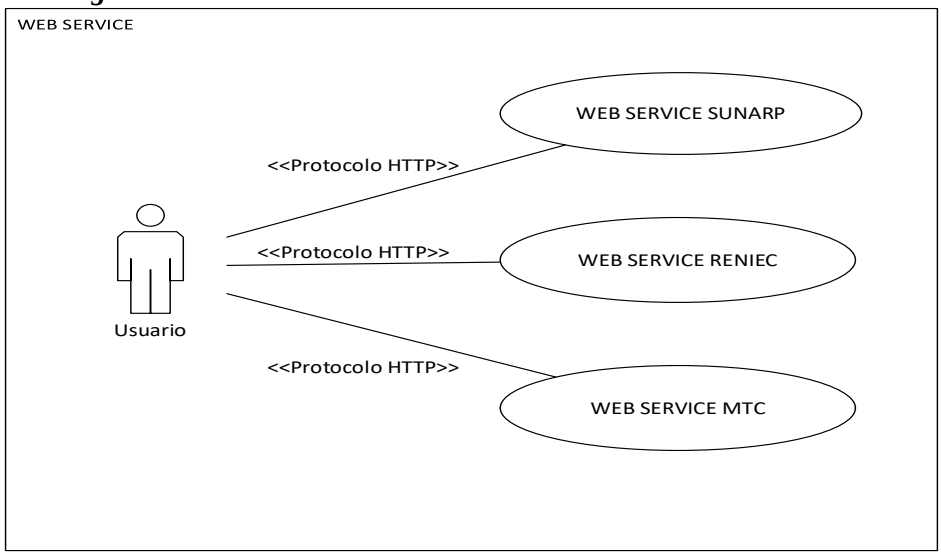
3.2.2 Análisis

Se detalla las especificaciones, usando las herramientas adecuadas para la implementación del Servicio Web.

1. Diagrama de casos de uso

En la **Figura 10** se muestra el diagrama de Casos de Uso del Servicio Web.

Figura 10
Diagrama de Casos de Uso



- El Web Service SUNARP, ofrece las siguientes operaciones: ingreso de la matricula (placa) del vehículo y la obtención de la marca del vehículo, modelo del vehículo, clase del vehículo, color del vehículo, número de DNI y nombres del propietario.

- El Web Service MTC, ofrece las siguientes operaciones: ingreso del número de licencia o número de DNI del conductor y la obtención de nombres del conductor, clase de licencia y categoría de licencia.
- El Web Service RENIEC, ofrece las siguientes operaciones: ingreso del número de DNI de la persona (propietario y conductor) y la obtención de los nombres completos y la dirección de la persona

3.2.3 Diseño

Para el diseño se considera como base el análisis y los requerimientos para el desarrollo del Servicio Web, en concreto se toma la arquitectura de *Windows Communication Foundation* (WCF) para el desarrollo.

Métodos del servicio web

Se implementan los siguientes métodos para la obtención de datos mediante el Servicio Web.

- Método recuperar persona.

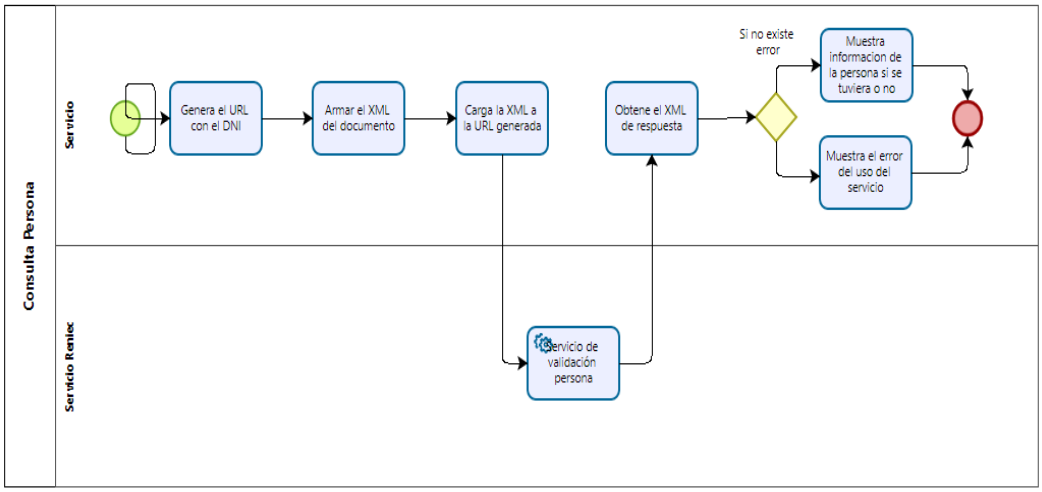
  Este método nos permite recuperar los datos de la persona desde el servicio web de la RENIEC, mediante una cadena de texto correspondiente al objeto persona concatenando el número de DNI, luego se realiza la carga del documento XML desde la cadena de texto generada, una vez cargado el documento se realiza la validación de los datos recuperados, de resultar nula la búsqueda el método no nos devuelve valor alguno, caso contrario. Se recuperan los datos como: el apellido paterno, el apellido materno, los nombres, la dirección, el número de ubigeo y la restricción en caso la tuviera.

private static string

urlPersona= "http://gmczonasegura.cusco.gob.pe/wuqueries/dniquery.php";

En la **Figura 11** podemos observar el flujo del Método Recuperar Persona.

Figura 11
Método Recuperar Persona



- Método recuperar vehículo.

El siguiente método, nos permite recuperar los datos del vehículo desde el servicio web de la SUNARP. mediante una cadena de texto correspondiente al objeto vehículo adicionando el número de la placa o número de matrícula del vehículo, previa transformación a mayúsculas del parámetro placa. Luego generamos una lista de todas las zonas registrales, así como de las oficinas de cada zona registral de la SUNARP. el método realiza la carga del documento XML desde la cadena de texto generada, una vez cargado el documento se realiza la validación de los datos recuperados, cabe mencionar que la búsqueda se realiza en cada una de las oficinas de las zonas registrales, en caso de que la búsqueda resulte nula o vacía se realiza la búsqueda en la siguiente zona registral caso contrario. Se recuperan

los datos del vehículo, como: el número de placa, el color, la serie del vehículo, el número VIN, el número o serie del motor, la marca del vehículo, el estado, la sede donde se registró el vehículo y la lista de propietarios del vehículo.

- Método recuperar propietario.

  Este método nos permite recuperar los datos de propietario del vehículo desde el servicio web de la SUNARP, mediante una cadena correspondiente al objeto propietario concatenando el tipo de propietario (este puede ser jurídico o natural), el apellido paterno, el apellido materno, los nombres y la razón social del propietario, luego se realiza la carga del documento XML desde la cadena de texto generada, esto nos puede devolver uno más ítems, debido a que un propietario puede tener varios vehículos registrados, para resolver esto se genera una lista de los vehículos registrados por el propietario, luego recorremos dicha lista buscando el vehículo requerido comparando por el número de placa, una vez encontrado el vehículo obtenemos el número ítem, finalmente mediante el número ítem se procede a recuperar los datos del propietario, como son: el tipo de documento, el número de documento y el número de placa del vehículo.

  Luego con estos datos recuperados se procede con utilizar el módulo de recuperar persona, para obtener los datos de cada propietario.

- Método recuperar conductor.

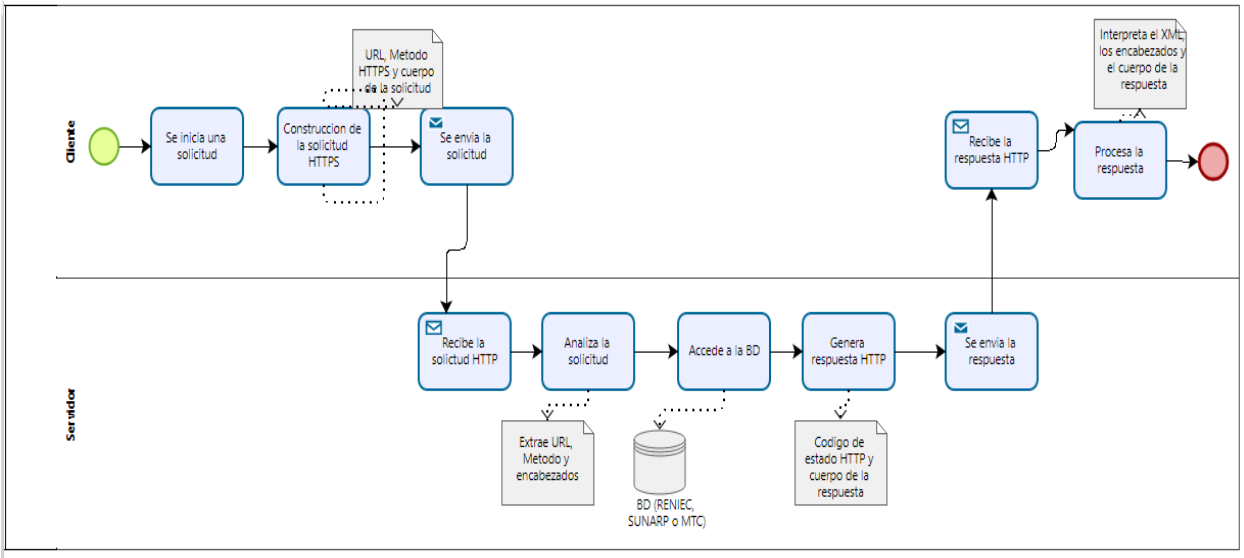
  Este método nos permite recuperar los datos del conductor del vehículo desde el servicio web del ministerio de transportes y comunicaciones, mediante una cadena de texto correspondiente al objeto conductor concatenando el número de documento, luego se realiza la carga del documento XML desde la cadena de texto

generada, se procede a validar que el número de documento ingresa exista, en caso no exista nos retorna nulo, caso contrario obtenemos los datos del conductor, como son: el número de DNI, el número de licencia de conducir, la categoría de la licencia de conducir, el apellido paterno, el apellido materno, los nombres, el departamento, la provincia, el distrito, la fecha de expedición de la licencia, la fecha de revalidación de la licencia, el estado de la licencia, la dirección del conductor y el correlato.

3.2.4 Funcionamiento general del Servicio Web

En la **Figura 12**, podemos observar todo el flujo del funcionamiento del servicio web.

Figura 12
Funcionamiento del Servicio Web



- Solicitud: Mediante la aplicación, se inicia una solicitud al servidor para acceder al recurso, la aplicación como solicitante del servicio requiere de un Servicio Web, entonces este se pone en contacto con el UDDI para ubicar el Servicio Web.

- Construcción de la solicitud HTTPS: Se construye la solicitud HTTPS, la cual incluye:
    1. URL: Identificación del recurso solicitado.
    2. Método HTTPS: Indica la acción a realizar (GET, POST, PUT, DELETE, etc.).
    3. Cuerpo de la Solicitud: se envía los parámetros de formulario para nuestro caso, enviamos para la consulta RENIEC, el número de DNI, para la consulta SUNARP la placa del vehículo y para la consulta al MTC enviamos el número de licencia de conducir o el número de DNI

  La solicitud se envía al servidor a través de la red, utilizando el protocolo en su versión segura, HTTPS.
- Recepción y Procesamiento en el Servidor:
    1. El servidor recibe la solicitud HTTP.
    2. El servidor analiza la solicitud, extrayendo la información clave como la URL, el método y los encabezados.
    3. Basándose en la información de la solicitud, el servidor realiza las operaciones necesarias, para nuestro caso accede a bases de datos.
- Respuesta HTTP:

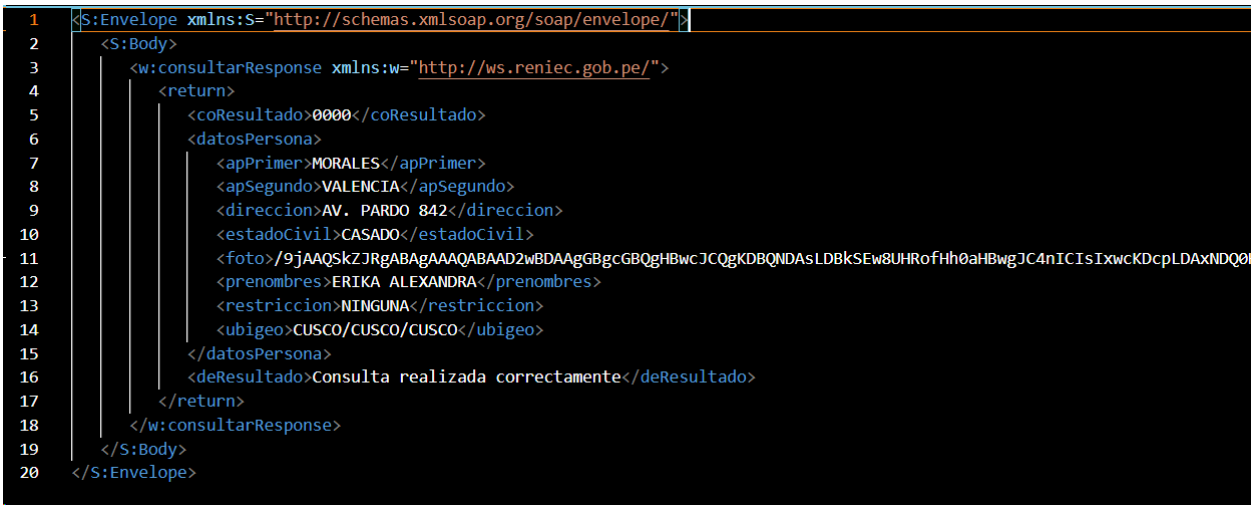
  El servidor genera una respuesta HTTP que incluye:
    1. Código de Estado HTTP: Indica el resultado de la operación (200 OK, 404 Not Found, etc.).
    2. Cuerpo de la Respuesta: Contiene los datos solicitados o información sobre el resultado de la operación en formato XML.

3. La respuesta se envía al cliente a través de la red.

- Recepción y Procesamiento en el Cliente:
  1. El cliente recibe la respuesta HTTPS.
  2. El cliente procesa la respuesta, interpretando el código de estado .xml, analizando los encabezados y extrayendo el cuerpo de la respuesta.
- Fin de la Comunicación: La conexión se cierra después de cada solicitud y respuesta

En la **Figura 13** se observa el resultado generado al momento de realizar la consulta a la RENIEC.

Figura 13
Resultado de la consulta a la RENIEC



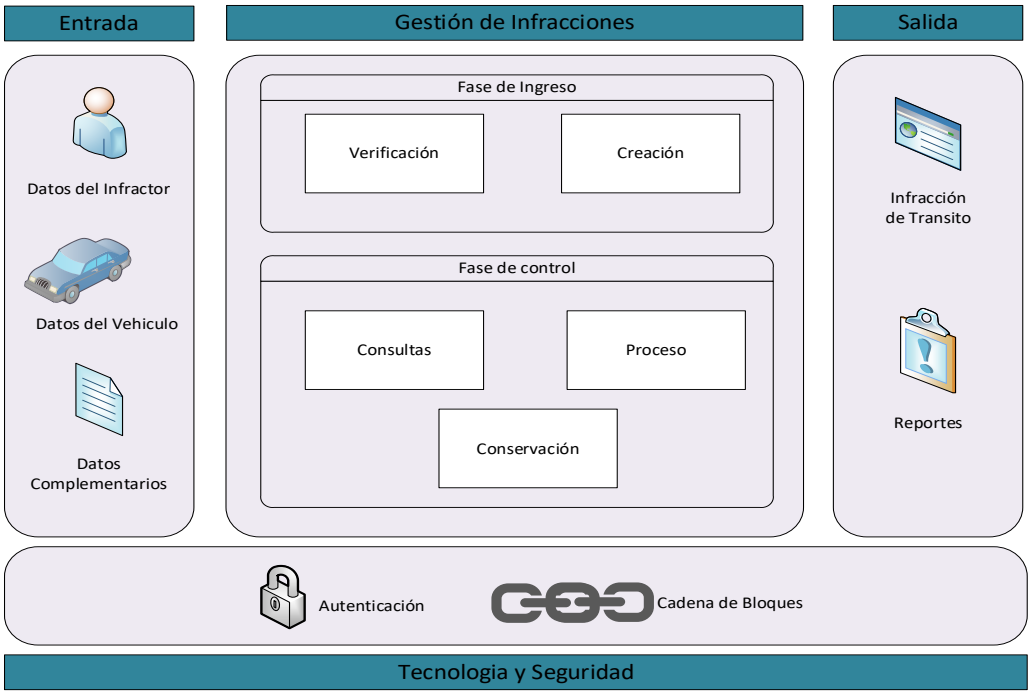
CAPÍTULO IV

IMPLEMENTACION DEL BLOCKCHAIN

4.1 Diseño del modelo

Representando diferentes modelos y arquitecturas técnicas gubernamentales, se realizó el análisis para el manejo de las infracciones de tránsito como contexto y técnica. Siendo así se desarrolló el siguiente modelo técnico.

Figura 14
Componentes del modelo



El modelo planteado en la **Figura 14** se divide en 4 componentes, que permite la gestión de infracciones de tránsito.

A continuación, se detalla cada uno de los componentes del modelo.

4.1.1 Entrada.

En este componente se detalla la forma de ingreso de los datos al modelo tecnológico con relación a la capa de gestión de infracciones.

Específicamente este tiene inicio con la recepción física de las infracciones de tránsito, luego estas son ingresadas al sistema por el personal de tránsito:

- Primero: la placa del vehículo infractor, se ingresa la placa del vehículo infractor y con el uso del servicio web se obtiene los datos del vehículo, como. Marca, modelo, clase color. Así como los datos del o los propietarios, como son: nombres y apellidos y/o razón social, número de DNI y/o número de ruc y la dirección.
- Segundo: el número de DNI o número de licencia del conductor del vehículo, se ingresa el número de DNI o número de licencia del conductor y con el uso del servicio web se obtienen los siguientes datos. Nombres y apellidos, dirección, clase licencia y la categoría.
- Tercero: el número CIP del efectivo policial que aplica la infracción de tránsito, la unidad y los nombres y apellidos.
- Cuarto: los datos propios de la infracción, como son: el número de infracción, grupo de infracción, tipo de infracción, categoría de la infracción, vía o dirección y observaciones.

4.1.2 Gestión de infracciones.

Este componente se encarga de la gestión de la infracción de tránsito, donde se interactúa directamente con la información contenida en cada infracción de tránsito, en la que se debe garantizar la integridad y seguridad de la información, Y el manejo adecuado en el ciclo correspondiente de dichas infracciones.

Fase de ingreso.

- Verificación: Realizada con la previa revisión de cada infracción de tránsito física, que estas tengan todos los datos correspondientes, luego con la existencia de la

placa del vehículo y el número de DNI o número de licencia del conductor, previa consulta en el sistema de información.

- Creación: Una vez realizada la verificación de los datos propios de la infracción, el usuario ingresa la infracción de tránsito al sistema.

Fase de control

- Consultas: Las infracciones de tránsito, luego del registro correspondiente, estarán a disposición de las áreas involucradas según sus privilegios y bajo las medidas de seguridad necesaria, para que puedan realizar consultas y/o reportes según se requiera.
- Proceso: Las infracciones de tránsito, luego del registro correspondiente, estarán a disposición de las áreas involucradas según sus privilegios y bajo las medidas de seguridad necesaria, para que estas puedan ser utilizadas, ya sea para realizar el pago total o parcial de dichas infracciones.
- Conservación: Las infracciones de tránsito, por ser un activo de la municipalidad provincial del cusco estas se conservarán en el tiempo, para la consulta correspondiente y el análisis para la toma de decisiones futuras.

4.1.3 Tecnología.

Este componente contiene una gran tecnología que soporta el modelo tecnológico, que por su característica aporta en cubrir las oportunidades de mejoras propuestas en el problema. La tecnología de cadena de bloques nos garantiza la integridad y seguridad de la información contenida en la base de datos de infracciones de tránsito.

Por las medidas de seguridad, características y políticas que nos proporciona la mencionada tecnología.

4.1.4 Salida.

En esta capa, se tiene como resultado final la infracción de tránsito, brindando la disponibilidad cuando esta sea requerida.

4.2 Recursos para la implementación

Para la implementación del Blockchain, se sugiere que la Municipalidad Provincial del Cusco, cuente con los siguientes recursos mínimos.

- **Base de Datos SQL Server:**
  - Versión de SQL Server: Para el proyecto se debe instalar SQL Server en la versión 2019, utilizando el hash criptográfico SHA256.
  - Esquema de Base de Datos: Se realizo el diseño del esquema de la base de datos SQL Server de acuerdo con los requisitos del sistema de cadena de bloques que se desea implementar.
- **Seguridad:**
  - Gestión de Claves Privadas: La base de datos deberá permitir la implementación de prácticas de seguridad para manejar las claves privadas y asegurar la integridad y confidencialidad de los datos.
  - Cifrado y Autenticación: La base de datos deberá permitir la utilización de cifrado y autenticación para proteger la comunicación y garantizar la seguridad de la información.

Adicional a ello, se desarrolló interfaces de usuario que permitan a los usuarios interactuar con el sistema de cadena de bloques y acceder a la información almacenada en la base de datos.

4.3 Diseño de la solución

Se desarrolló el modelo de Blockchain para el almacenamiento y gestión de infracciones de tránsito, para lo cual se diseñó la base de datos hasta su implementación, utilizando la tecnología representada en el modelo.

4.3.1 Arquitectura de la solución

La solución consta de tres componentes principales: las bases de datos desarrolladas en SQL Server, la red de blockchain y la interfaz del usuario. La base de datos almacenara los datos relacionados a las infracciones de tránsito, como son: datos de los vehículos, datos de los propietarios, datos de los infractores, datos del personal policial y datos propios de la infracción. Mientras que la red blockchain garantizará la inmutabilidad, seguridad y transparencia de los datos registrados de la infracción, como son: infracción, código vehículo, código conductor, código propietario, código infracción, grupo infracción, tipo infracción, fecha infracción, vía, código PNP, operador y observaciones. La interfaz del usuario permitirá a los usuarios interactuar con el sistema.

4.3.2 Base de datos SQL Server

Se implemento las siguientes tablas en la base de datos:

- Persona.Documento: IdDocumento, NombreDocumento, EstadoDocumento, UsuarioCrea, UsuarioModifica, FechaCreacion, FechaModificacion.
- Persona.Persona: IdPersona, IdDocumento, NumeroDocumento, ApellidoPaterno, ApellidoMaterno, Nombres, RazonSocial, Direccion, Celular, CorreoElectronico, UsuarioCrea, UsuarioModifica, FechaCreacion, FechaModificacion.

- Persona.Propietario: IdPropietario, IdPersona, IdVehiculo, TarjetaPropiedad, UsuarioCrea, UsuarioModifica, FechaCreacion, FechaModificacion.
- PersonaConductor: IdConductor, LicenciaNumero, LicenciaClase, LicenciaCategoria, FechaLicenciaExpira, fechaLicenciaRevalida, UsuarioCrea, UsuarioModifica, FechaCreacion, FechaModificacion.
- Persona.Policia: IdPolicia, IdPersona, CIP, Unidad, UsuarioCrea, UsuarioModifica, FechaCreacion, FechaModificacion.
- Licencia.Clase: IdClase, NombreClase, Descripcion, UsuarioCrea, UsuarioModifica, FechaCreacion, FechaModificacion.
- Licencia.Categoria: IdCategoria, IdClase, NombreCategoria, UsuarioCrea, UsuarioModifica, FechaCreacion, FechaModificacion.
- Vehiculo.Vehiculo: IdVehiculo, NumeroPlaca, IdMarca, IdModelo, IdClase, IdColor, SerieMotor, NumeroAsientos, UsuarioCrea, UsuarioModifica, FechaCreacion, FechaModificacion.
- Vehiculo.Marca: IdMarca, NombreMarca, Estado, UsuarioCrea, UsuarioModifica, FechaCreacion, FechaModificacion.
- Vehiculo.Modelo: IdModelo, IdMarca, NombreModelo, Estado, UsuarioCrea, UsuarioModifica, FechaCreacion, FechaModificacion.
- Vehiculo.Clase: IdClase, NombreClase, Estado, UsuarioCrea, UsuarioModifica, FechaCreacion, FechaModificacion.
- Vehiculo.Color: IdColor, NombreColor, Estado, UsuarioCrea, UsuarioModifica, FechaCreacion, FechaModificacion.

- Infraccion.Estado: IdEstado, IdInfraccion, NombreEstado, UsuarioCrea, UsuarioModifica, FechaCreacion, FechaModificacion.
- Infraccion.Infraccion: IdBlock, KeyDNI, DataInfraccion, PrevKey, FechaInsert.

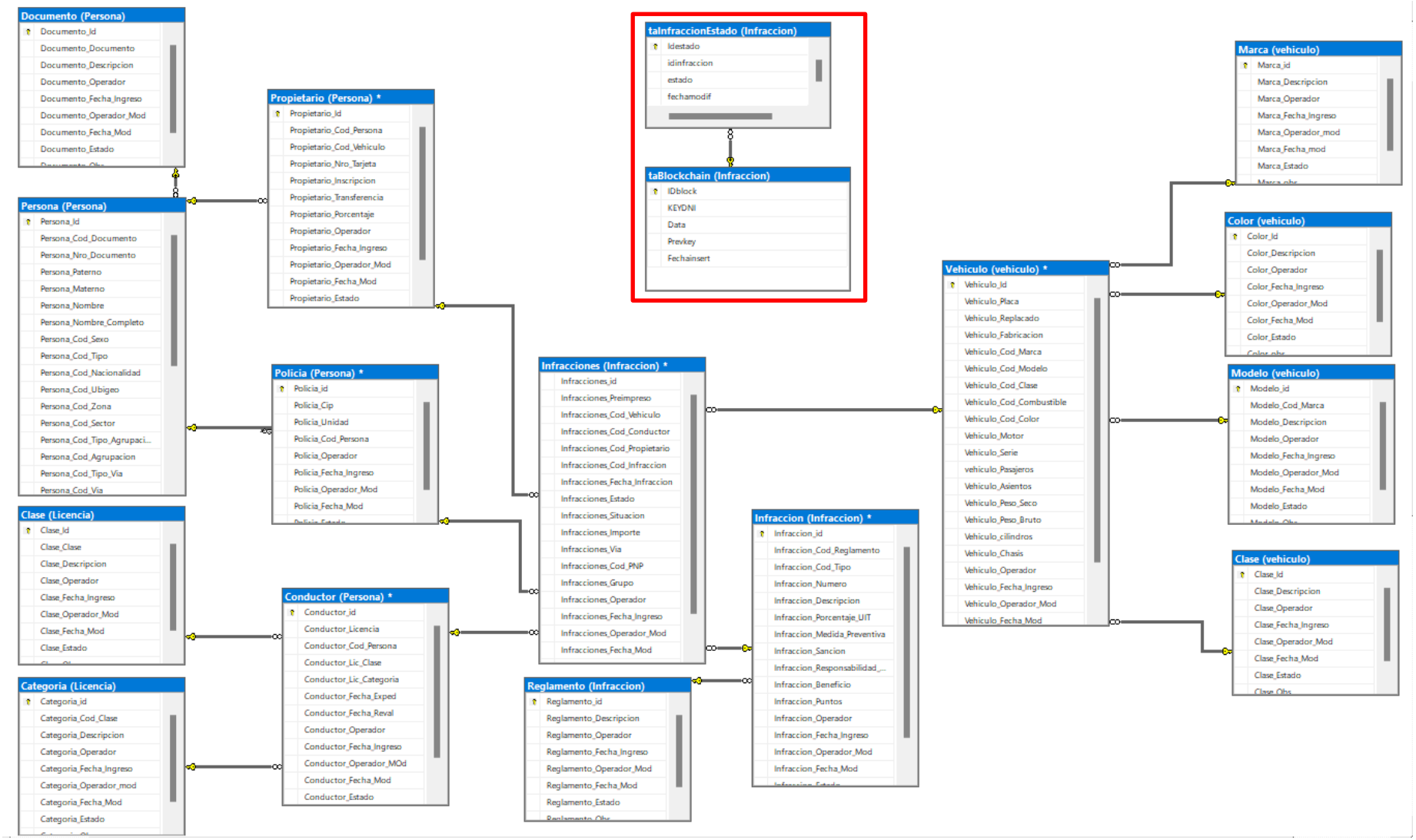
4.3.3 Modelado de Datos.

Para una base de datos relacional se admiten en el concepto de manejo de datos las opciones de insertar, modificar, eliminar, así como la opción de lectura. A diferencia de una red blockchain que solo admite como concepto de manejo de datos, las opciones de lectura y escritura. Con base en este concepto, se desarrolló el modelo de datos agregando una tabla “taInfraccionEstado”, que nos permitirá la modificación o actualización del estado de una infracción de tránsito. Y una tabla “taBlockchain”, donde se almacenará el detalle de una infracción de tránsito en modo cadena, aplicando el concepto con el cifrado de una red blockchain. Las tablas mencionadas solo guardarán relación entre sí, por el campo de identificador (idBlock); mas no estarán relacionadas con el resto de tablas de la base de datos, debido a que los campos restantes de la tabla taBlockchain estarán cifrados, cumpliendo con el concepto de la red blockchain.

De esta manera se desarrolló el siguiente modelo de datos que se muestra en la **Figura 15**, donde se observa que la parte remarcada de color rojo son las tablas nuevas que se implementaron y el resto pertenecen a la base de datos original.

Figura 15

Diagrama de Base de Datos



4.3.4 Modelado de procesos

Los procesos involucrados en la solución del modelo de gestión de infracciones de tránsito son:

- Registro de personal digitador.
- Registro de infracciones de tránsito.
- Consulta de infracciones de tránsito.
- Pago de infracciones de tránsito.

Los actores involucrados en los procesos, son los siguientes:

Tabla 1
Actores del Proceso

Rol del proceso	Responsabilidad
|	Aplicar la infracción de tránsito a los conductores infractores.
Policía de tránsito	Entrega de las infracciones de tránsito en la municipalidad provincial del cusco.
Digitador	Validar los datos de las infracciones de tránsito. Registro de las infracciones de tránsito en el sistema.
Personal de caja	Visualizar el monto correspondiente a la infracción de tránsito. Realizar el cobro de la multa aplicado a la infracción de tránsito.

A continuación de describen los procesos:

- Registro de personal digitador.

  Actores involucrados:

  - Administrador.
  - Digitador.
  - Sistema informático

  El proceso inicia con la necesidad de contar con personal responsable encargado de la validación y registro de las infracciones de tránsito. El digitador podrá acceder al sistema par al validación y registro correspondiente de las infracciones de tránsito. El administrador ingresa los datos del digitador en el sistema con los privilegios correspondientes (**Figura 16**).

  Figura 16
  Caso de Uso: Registro de Personal Digitador

- Registro de infracciones de tránsito.

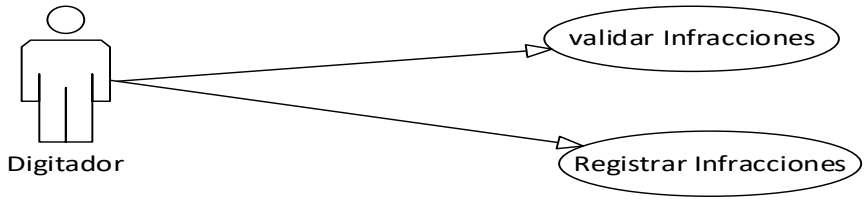
  Actores involucrados:

  - Digitador.
  - Sistema informático.

  El proceso consiste en registrar una nueva infracción de tránsito en el sistema, el personal que tiene como funcionalidad la de realizar esta actividad es el digitador. Previa verificación de la veracidad de los datos como son. Datos del vehículo y

datos del infractor, el digitador procede con ingresar o registrar la infracción de tránsito al sistema (**Figura 17**).

Figura 17
Caso de Uso: Registro de Infracciones de Transito



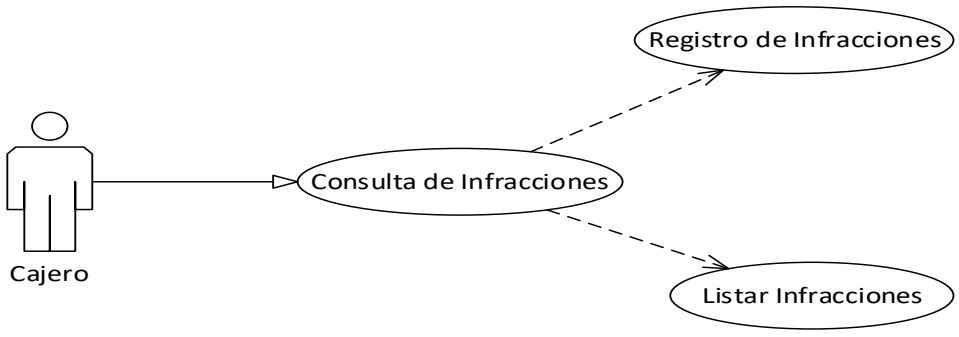
- Consulta de infracciones de tránsito.

  Actores involucrados:

  - Personal administrativo.
  - Sistema informático.

  El proceso consiste en la consulta o verificación de los datos de una infracción en específico o un listado de infracciones, el personal administrativo encargado realiza la consulta con fines propios de su encargatura (**Figura 18**).

Figura 18
Caso de Uso: Consulta de Infracciones de Transito



- Pago de infracciones de tránsito.

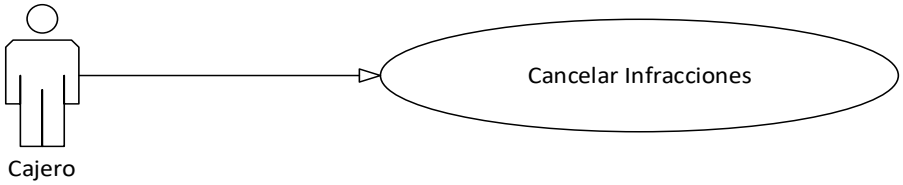
  Actores involucrados:

  - Personal de caja.

- Sistema informático.

El proceso consiste en realizar la cobranza de la multa correspondiente a la infracción de tránsito. El personal encargado para dicha función es el personal de caja, cuya función es la de verificar la infracción de tránsito en el sistema y realizar la cobranza correspondiente, cancelando la deuda de la infracción en el sistema informático (**Figura 19**).

Figura 19
Caso de Uso: Pago de Infracciones de Transito



4.3.5 Requisitos

Los requisitos se establecen para satisfacer las necesidades de la empresa. Estos tipos de requisitos describen los requisitos funcionales y no funcionales del sistema en un lenguaje formal. De esta manera, estos requisitos se crean de una manera fácil de entender.

4.3.5.1 Funcionales

Teniendo en cuenta los requisitos del sistema, se decidió utilizar una red Blockchain para almacenar el historial de infracciones de tránsito de manera electrónica. De esta forma será posible satisfacer las necesidades de la forma más completa posible. Antes de profundizar en los requisitos no funcionales, una razón preliminar es que las redes de cadena de bloques brindan a los usuarios un control total sobre cómo se administran sus datos, y esos datos siempre están disponibles.

Un requisito funcional es una definición de funcionalidad que el sistema debe de tener. Estos requisitos surgen de los requisitos del usuario y los casos de uso. Por otro lado, se complementan con requisitos no funcionales. Esta definición satisface la siguiente tabla:

Tabla 2
Matriz de Requisitos Funcionales

Código	Descripción
RF01	El sistema debe permitir iniciar sesión a los usuarios con las credenciales correspondientes, estas son: un ID alfanumérico y una contraseña.
RF02	El sistema debe mostrar el módulo con las funcionalidades correspondientes al rol del usuario que inicio sesión.
RF03	El sistema solo debe permitir el registro del personal por el administrador del sistema. (digitadores y personal de caja)
RF04	El sistema debe permitir el registro de las infracciones de tránsito, como son: datos del vehículo, datos del infractor, datos complementarios a la infracción de tránsito.
RF05	El sistema debe funcionar de manera distribuida para todos los nodos que interactúan con el sistema
RF06	El sistema no debe permitir la modificación o alteración de una infracción de tránsito registrada en el sistema.
RF07	El sistema debe permitir la búsqueda de infracciones de tránsito por DNI (8 dígitos) del infractor.

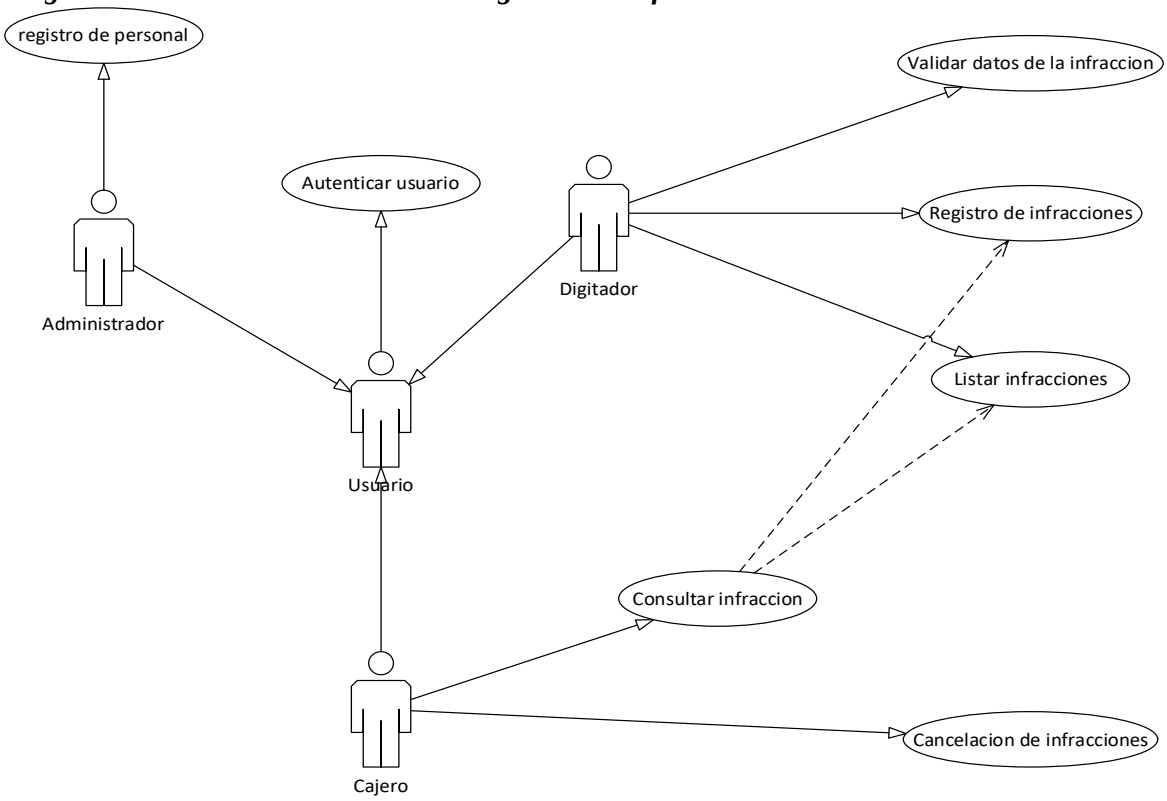
RF08	El sistema debe permitir la búsqueda de infracciones de tránsito dentro de un rango de fechas seleccionadas.
RF09	El sistema debe permitir la cancelación de las multas correspondientes a las infracciones de tránsito.

4.3.6 Diagrama de casos de uso

El sistema de gestión de infracciones de tránsito, tiene tres actores que interactúan con las funcionalidades con el sistema. Esto según los requisitos funcionales, a continuación, se presenta el diagrama de clases de uso, donde se puede observar la relación de los actores con las funcionalidades del sistema (**Figura 20**).

Figura 20

Diagrama de casos de uso del sistema de gestión de infracciones de tránsito



4.4 Implementación de la red Blockchain en la base de datos.

4.4.1 Herramientas.

La integración se realizó en la plataforma de SQL Server versión 2019, donde se utilizó el hash criptográfico SHA256.

- SHA256: es un algoritmo de HASH seguro. Es una de varias funciones criptográficas. El hash criptográfico funciona como una firma para un conjunto de datos, es como las huellas dactilares de los datos, incluso si se llega a cambiar solo un símbolo, el algoritmo producirá un valor hash diferente.

  El algoritmo SHA256 genera un hash casi único de tamaño fijo de 256 bits (32 bytes). El Hash se llama una función unidireccional. Esto lo hace adecuado para verificar la integridad de sus datos, desafiar la autenticación hash, anti manipulación, firmas digitales, Blockchain.

- Merkle Tree: Un árbol Merkle, es una estructura de datos divida en varias capas que tiene como finalidad relacionar cada nodo con una raíz única asociada a los mismos. Para lograr esto, cada nodo debe tener un identificador único (hash). Estos nodos iniciales, llamados nodos hijos (hojas), se asocian luego con un nodo superior llamado nodo padre (rama). El nodo padre, tendrá un identificador único resultado del hash de sus nodos hijos. Esta estructura se repite hasta llegar al nodo raíz o raíz Merkle (Merkle Root), cuya impronta está asociada a todos los nodos del árbol.

4.4.2 Implementación

Se generó la estructura de la base de datos del sistema de infracciones con las tablas correspondientes, a esto se añadió una tabla nombrada ‘taBlockchain’ donde se

almacena la cadena de bloques con todos los datos que contiene la infracción de tránsito, adicional a ello se implementó la tabla ‘taInfraccionEstado’, esta nos permite saber el estado de la infracción que también nos permite modificar el estado de las infracciones almacenadas en la tabla taBlockchain.

En la tabla taBlockchain se generó 5 columnas:

- IDBlock: Identificador de nuestra tabla o primary key. Este dato se guarda como un numero entero el cual tiene una secuencia según el ingreso de las infracciones que a la vez nos permite saber el número de infracción registrada en la base de datos.
- KEYDNI: Número de DNI del infractor, se almacena en un formato de hash (resumen criptográfico) y adopta la función de la llave publica de la cadena de bloques
- Data: Campo donde se almacena todos los datos de la infracción, se almacena en formato hash (resumen criptográfico) de los datos de la cadena de bloques, este hash actúa como una huella digital, que nos permite salvaguardar la integridad de los datos en la base de datos
- PrevKey: se almacena en formato hash (resumen criptográfico). Número de DNI que apunta al registro anterior (apunta al KEYDNI del registro anterior), excepto el primer registro que hace referencia a un NULL, (el primer bloque también llamado bloque génesis)
- FechaInsert: Fecha en que se realizó el registro de la infracción o bloque.

Cuando se genera una infracción en el sistema, este se genera con los campos ya mencionados, así como en sus formatos correspondientes.

4.5 Funcionamiento general

Se encontró la viabilidad de aplicar Blockchain como seguridad sobre la base de datos en SQL Server, ya que esta base de datos tiene todas las versiones criptográficas. Estas funciones admiten firma digital, validación de firma digital, cifrado y descifrado. Así como también cifrado y descifrado simétrico, cifrado y descifrado asimétrico y hash de cifrado.

Se consiguió hacer uso de la tecnología Blockchain aplicado a la seguridad, con un nuevo modelamiento de datos, implementando dos nuevas tablas las cuales nos permiten la aplicación correcta de Blockchain.

Se implementó la tabla taBlockchain, con 5 columnas, dicha tabla nos permitirá almacenar los datos de la infracción de manera conveniente. En la columna IDblock de tipo entero almacenamos el id del bloque agregado, en la columna KEYDNI de tipo varbinary se almacenará el número de DNI del infractor, este dato se almacena en forma cifrada.

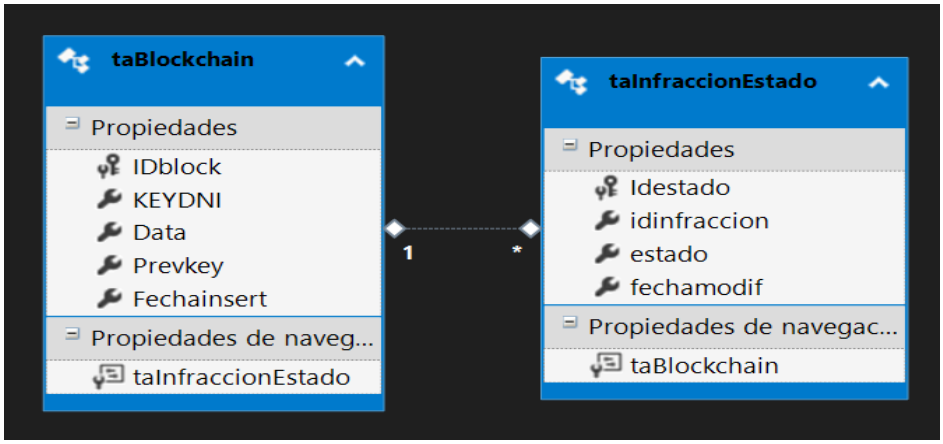
En la columna “data” de tipo varbinary se almacenará los datos de la infracción, los cuales son: Infracción, código vehículo, código conductor, código propietario, código infracción, grupo infracción, tipo infracción, fecha infracción, vía, código PNP, operador, observaciones; este dato se almacena de forma cifrada. Luego tenemos la columna Prevkey de tipo varbinary donde almacenaremos la clave previa del bloque, es decir el KEYDNI del registro anterior que también se almacenara de manera cifrada. Y por último tenemos la columna FechaInsert donde almacenaremos la fecha de inserción del registro.

Luego tenemos implementada la tabla taInfraccionEstado con el campo idEstado de tipo entero donde se almacenará el identificador. Luego tenemos el campo idInfraccion de tipo entero donde se relaciona con la tabla taBlockchain. Luego tenemos el campo Estado, donde se almacenará el estado de la infracción, si esta está registrada, pagada, cancelada o

extornada. Y finalmente tenemos el campo fechamodificacion, de tipo date time, donde almacenaremos la fecha en que se modificó el registro.

En la **Figura 21** podemos observar la tabla de almacenamiento del Blockchain

Figura 21
Tablas de almacenamiento de blockchain



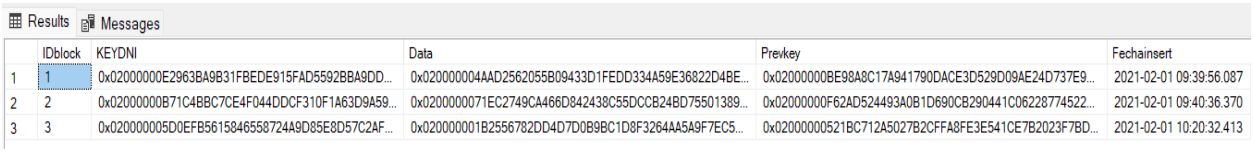
Para la implementación de los servicios web, se eligió la arquitectura SOAP, debido a que la herramienta de uso nos brindó facilidades en la implementación, con el Framework exclusivo para servicios y con la existencia de librerías propias para el proyecto. Se utilizó el lenguaje de programación .NET C# con el Framework Windows Communication Foundation (WCF).

Para la implementación de Blockchain aplicado a la seguridad se optó por el uso del algoritmo de Hash seguro de 256 bits SHA-256, debido a que este nos brinda toda la seguridad criptográfica requerida para el proyecto, cabe mencionar que el Hash viene incorporado en SQL Server a partir de la versión 2016.

Al guardar cada infracción esta se almacenará de manera cifrada según los campos establecidos previamente como son: Infracción, código vehículo, código conductor, código

propietario, código infracción, grupo infracción, tipo infracción, fecha infracción, vía, código PNP, operador, observaciones, lo cual se observa en la **Figura 22**.

Figura 22
Cifrado en almacenamiento de datos.



Results Messages

	IDblock	KEYDNI	Data	Prevkey	Fechainsert
1	1	0x02000000E2963BA9B31FBEDE915FAD5592BBA9DD...	0x020000004AAD2562055B09433D1FEDD334A59E36822D4BE...	0x02000000BE98A8C17A941790DACE3D529D09AE24D737E9...	2021-02-01 09:39:56.087
2	2	0x02000000B71C4BBC7CE4F044DDCF310F1A63D9A59...	0x0200000071EC2749CA466D842438C55DCCB24BD75501389...	0x02000000F62AD524493A0B1D690CB290441C06228774522...	2021-02-01 09:40:36.370
3	3	0x020000005D0EFB56158466558724A9D85E8D57C2AF...	0x020000001B2556782DD4D7D0B9BC1D8F3264AA5A9F7EC5...	0x02000000521BC712A5027B2CFFA8FE3E541CE7B2023F7BD...	2021-02-01 10:20:32.413

Debido a la estructura del Blockchain, para su funcionamiento se deberá realizar una copia de la Base de datos en los equipos de los usuarios que hacen uso del sistema, como son: el administrador, el digitador, el cajero y otros actores involucrados que tengan acceso al prototipo planteado para su auditoria.

CAPÍTULO V

ANALISIS DE RESULTADOS

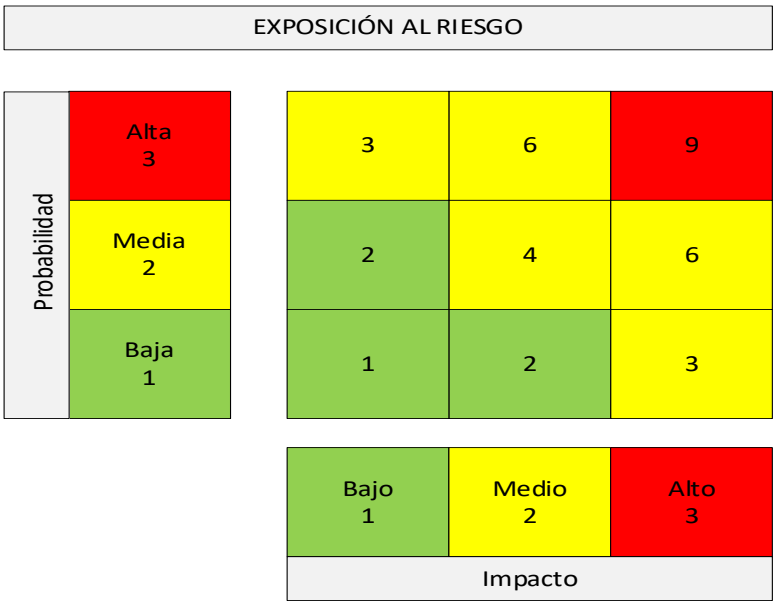
5.1 Análisis de resultados

Se realiza el siguiente análisis en comparación al uso de la tecnología Blockchain. Para ello se elaboró una matriz de riesgos y vulnerabilidades del modelo tradicional del sistema de infracciones versus el modelo planteado aplicando blockchain.

La matriz de exposición al riesgo o también conocido como mapa de calor, generalmente se compone de dos ejes: uno que representa la probabilidad de que ocurra un riesgo y otro que refleja el impacto que dicho riesgo tendría en la organización en caso de materializarse.

Estos dos ejes se dividen en categorías, que van desde "bajo" hasta "alto" en función de la probabilidad e impacto, este cuadro ayuda a visualizar como los riesgos pueden perjudicar las operaciones en otras áreas de la municipalidad.

Tabla 3
Exposición al riesgo



Nota. Fuente: elaboración propia.

Para ello se elabora un cuadro de exposición al riesgo de 3 x 3 en donde se clasifican los riesgos que serían encontrados más adelante en la validación de datos.

Los valores obtenidos en las celdas son el resultado de la multiplicación de la posibilidad de contingencia con el efecto del riesgo, indicando los valores más altos catalogándolos como riesgos más críticos y los más bajos, menos relevantes.

Para poder entender mejor los resultados se elaboró una tabla con los valores que se obtuvieron como resultado de la multiplicación.

Tabla 4
Riesgo = probabilidad x impacto

Probabilidad	**Impacto**	**Valor de riesgo**	**Nivel de riesgo**
Baja	Bajo	1	Bajo
Baja	Medio	2	Bajo
Baja	Alto	3	Medio
Medio	Bajo	2	Bajo
Medio	Medio	4	Medio
Medio	Alto	6	Medio
Alto	Bajo	3	Medio
Alto	Medio	6	Medio
Alto	Alto	9	Alto

Nota. Fuente: elaboración propia.

El primer paso en la creación de una matriz de exposición al riesgo es identificar y enumerar todos los riesgos a los que la municipalidad está expuesta. Estos riesgos pueden incluir factores económicos, financieros, operativos, legales, ambientales y más.

- Probabilidad: En la matriz, se asigna a cada riesgo una puntuación de probabilidad que indica cuán probable es que ocurra. Esto puede basarse en análisis cualitativos o cuantitativos.
- Impacto: Se otorga a cada riesgo una puntuación de impacto que mide las consecuencias potenciales que tendría en la organización si llegara a materializarse. El impacto puede variar desde un impacto mínimo hasta un impacto crítico.

Para el siguiente cuadro, la matriz de exposición al riesgo se compone de dos ejes:
Un eje vertical en donde se establecen los valores de probabilidad (1 = Baja, 2 = Media, 3 = Alta) y un eje horizontal en donde se establecen los valores del impacto (en donde 1 es considerado como Bajo, 2 es considerado Medio y 3 es considerado Alto).
La matriz de riesgos elaborada cuenta con 9 columnas que se describen a continuación:

- ✓ Código del riesgo.
- ✓ Nombre del riesgo.
- ✓ Activo.
- ✓ Amenaza.
- ✓ Vulnerabilidad.
- ✓ Probabilidad.
- ✓ Impacto.
- ✓ Valor del riesgo.
- ✓ Nivel del riesgo.

A continuación, se muestra la matriz de riesgos elaborada con base al análisis del modelo del sistema de infracciones de tránsito tradicional.

Tabla 5
Matriz de riesgos con base al análisis del modelo de tránsito tradicional

Código	Riesgo	Activo	Amenaza	Vulnerabilidad	Probabilidad	Impacto	Valor de riesgo	Nivel de riesgo
A	Ingreso errado de datos por parte del personal encargado	Sistema de gestión de infracciones.	Posibilidad de mala acción del trabajador	Actitud deshonesta del trabajador	2	3	6	Medio
B	Fuga de información por parte del personal con acceso al sistema.	Sistema de gestión de infracciones.	Posibilidad de divulgación de información	Actitud deshonesta del trabajador	2	3	6	Medio
C	Manipulación y alteración de datos	Base de datos.	Posibilidad de mala acción del trabajador	Actitud deshonesta del trabajador	2	3	6	Medio
D	Caída del servicio e indisponibilidad del sistema.	Dispositivos de red, servidor, sistema de gestión de infracciones.	Ataques de hacking externo / interno a los dispositivos de red, caída de servidores, entre otros	Vulnerabilidades identificadas no implementadas oportunamente.	2	3	6	Medio

A continuación, se muestra la matriz de riesgos elaborada con base al análisis del modelo del sistema de infracciones de tránsito aplicando tecnología blockchain.

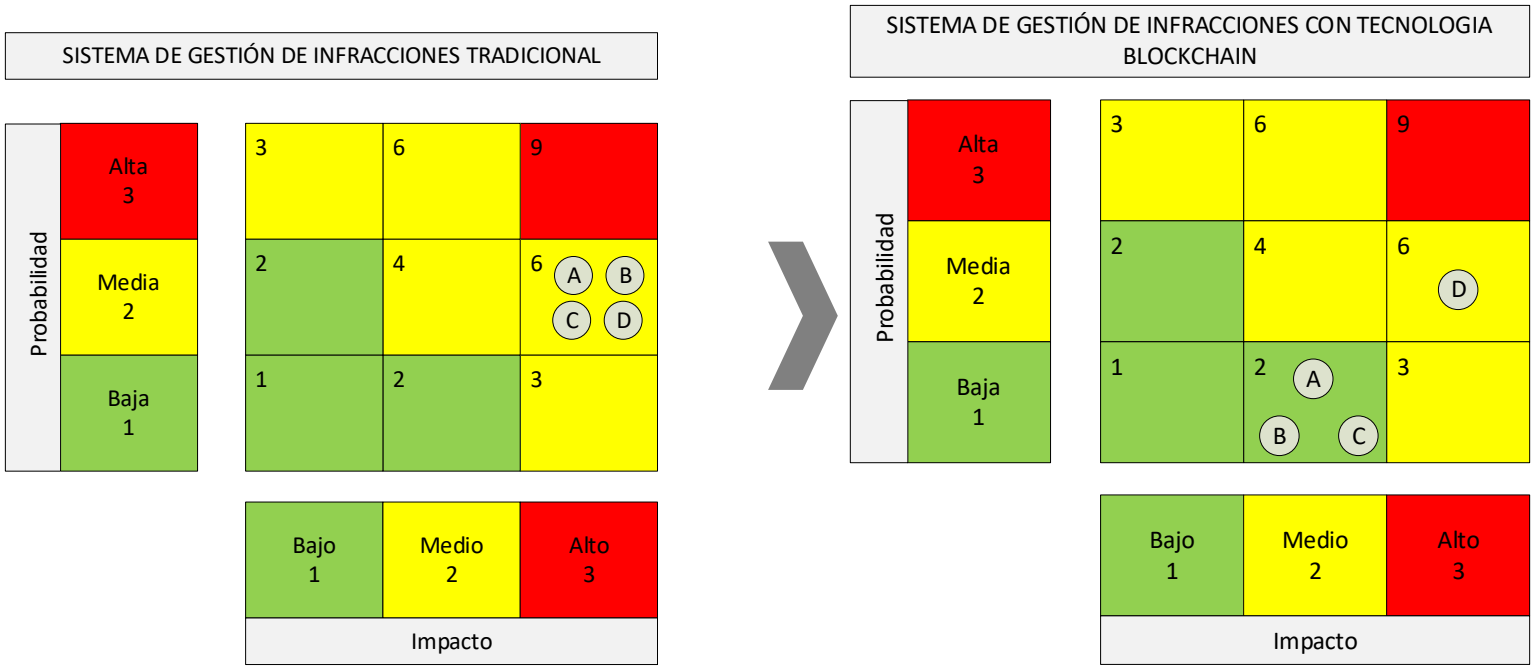
Tabla 6
Matriz de riesgos con base al análisis del modelo de tecnología blockchain

Código	Riesgo	Activo	Amenaza	Vulnerabilidad	Probabilidad	Impacto	Valor de riesgo	Nivel de riesgo
A	Ingreso errado de datos por parte del personal encargado	Sistema de gestión de infracciones.	Posibilidad de mala acción del trabajador	Con el consumo de los servicios web de las entidades nacionales, el error o mal intención del trabajador se reduce	1	2	2	Bajo
B	Fuga de información por parte del personal con acceso al sistema.	Sistema de gestión de infracciones.	Posibilidad de divulgación de información	Con el uso de la tecnología blockchain no se tiene acceso a los datos desde la base de datos.	1	2	2	Bajo
C	Manipulación y alteración de datos	Base de datos.	Posibilidad de mala acción del trabajador	Con el uso de la tecnología blockchain no es posible modificar los datos.	1	2	2	Bajo
D	Caída del servicio e indisponibilidad del sistema.	Dispositivos de red, servidor, sistema de gestión de infracciones.	Ataques de hacking externo / interno a los dispositivos de red, caída de servidores, entre otros	Vulnerabilidades identificadas no implementadas oportunamente.	2	3	6	Medio

Para la validación del análisis de riesgos, realizamos la comparación de la exposición al riesgo del sistema de infracciones de tránsito tradicional frente a otro escenario, utilizando la tecnología de blockchain. A continuación, se muestran los cuadros de riesgos obtenidos.

Tabla 7

Comparación del análisis de riesgos.



La comparación entre la matriz de riesgos presentes en ambos escenarios se vuelve más evidente al analizar el sistema de infracciones de tránsito. Es especialmente notorio cómo el uso de la tecnología Blockchain tiene menor exposición al riesgo en comparación con un enfoque tradicional.

Además, podemos concluir que se cumple con los estándares de la norma ISO 27005, la cual establece pautas para la gestión de riesgos en seguridad de la información. Esta norma respalda los conceptos generales especificados en la norma ISO/IEC 27001 y está diseñada para facilitar la implementación efectiva de la seguridad de la información a través de un enfoque de gestión de riesgos.

CONCLUSIONES

1. La aplicación de la tecnología Blockchain en la gestión de infracciones de tránsito vehicular en la Municipalidad Provincial del Cusco ha proporcionado resultados positivos. Se concluye que Blockchain mejora la seguridad, la transparencia y la trazabilidad en el proceso de gestión de infracciones, consolidándose como una solución efectiva para enfrentar los desafíos previamente identificados.

   En conjunto, las conclusiones derivadas de la investigación respaldan la aplicación de la tecnología Blockchain como una herramienta valiosa para fortalecer la seguridad en la gestión de infracciones de tránsito, proporcionando un marco confiable y eficiente para el sistema de infracciones.

2. Análisis de requisitos y viabilidad: El examen detallado de los requisitos y la viabilidad del uso de la tecnología Blockchain para la seguridad en la gestión de infracciones de tránsito en la Municipalidad Provincial de Cusco reveló la idoneidad de esta solución. Se concluye que la aplicación de la tecnología Blockchain puede abordar las limitaciones identificadas en el sistema convencional, proporcionando una plataforma segura, transparente y resistente a manipulaciones para el registro y la gestión de infracciones de tránsito.

3. Implementación de Servicios Web: La implementación de Servicios Web que permiten consumir datos del Registro Nacional de Identificación y Estado Civil (RENIEC), la Superintendencia Nacional de Registros Públicos (SUNARP) y el Ministerio de Transportes y Comunicaciones (MTC) ha sido esencial para enriquecer la información asociada con las infracciones de tránsito. La interoperabilidad de estos servicios ha

mejorado la calidad de los datos y ha fortalecido la base de información para la gestión eficiente de las infracciones.

4. Adaptabilidad de la Base de Datos: Se realizó una adaptación de la base de datos para facilitar la implementación de Blockchain en la seguridad de la gestión de infracciones de tránsito en la Municipalidad Provincial del Cusco. Este proceso implicó la adaptabilidad de la base de datos y la implementación de nuevas tablas, sin presentarse inconsistencias en la base de datos original.

RECOMENDACIONES

1. Plataforma Web para Usuarios: Se recomienda la implementación de una plataforma web que permita a los usuarios realizar consultas y seguimientos de las infracciones de tránsito. Esta plataforma mejoraría la accesibilidad y la transparencia de la gestión de infracciones de tránsito.
2. Implementación de Blockchain con Base de Datos No SQL: Se sugiere considerar una propuesta de implementación de Blockchain con una base de datos No SQL (no relacional). Dada su escalabilidad y flexibilidad, las bases de datos No SQL pueden ser más adecuadas para el desarrollo de tecnología Blockchain.
3. Uso de Blockchain en Instituciones Públicas y Privadas: Se recomienda la aplicación de Blockchain para la seguridad de la información en diversas instituciones, tanto del sector público como privado. La tecnología Blockchain proporciona una estructura de datos con características de seguridad inherentes, basada en principios de criptografía, descentralización y consenso, que garantizan la confiabilidad de las transacciones.
4. Implementación de un manual para el proceso de la gestión de infracciones en la Municipalidad Provincial del Cusco.
5. Se recomienda la implementación de un módulo para que pueda ser utilizado por el Policía de Tránsito, quien es el actor inicial en el proceso de la emisión de papeletas de tránsito.

REFERENCIAS Y CITAS BIBLIOGRÁFICAS

Anicama Lopez, F. C. (2019). *MODELO DE BLOCKCHAIN PARA MEJORAR LA TOMA DE DECISIONES EN LAS SENTENCIAS FISCALES DEL MINISTERIO PÚBLICO LIMA 2019-2022.* Lima.

Besteiro, M., & Rodriguez, M. (29 de Junio de 2021). *Web Services*. Obtenido de http://www.ehu.eus/mrodriguez/archivos/csharppdf/ServiciosWeb/WebServices.pdf

Bravo Vecorena, A. (2011). *Diseño e Implementacion de una Bliblioteca Digital Distribuida Basada en Web Services para el Sector Educacion.* Lima: Universidad Nacional de Ingenieria.

Criptoeconomia. (23 de Enero de 2019). *¿Qué es Blockchain o la tecnología cadena de bloques?* Obtenido de https://www.criptoeconomia.com/que-es-blockchain-o-cadena-de-bloques/

Cusco, M. d. (15 de Octubre de 2019). *Municipalidad del Cusco*. Obtenido de https://sumamosdesign.com/cusco/nosotros/

Dr. Raymond Colle. (28 de Junio de 2021). *La identidad digital en la internet futura con blockchain*. Obtenido de https://www.academia.edu/36613066/La_identidad_digital_en_la_internet_futura_con_blockchain

Espíritu Aranda, W. A., & Machuca Nieva, C. F. (2021). *Modelo de Referencia para la Gestión de la Seguridad de Datos de Salud.* Lima: Universidad Peruana de Ciuencias Aplicadas.

Gamboa Manzaba, J. (2014). Aumento de la Productividad en la Gestión de Proyectos, Utilizando una Metodología Àgil Aplicada en una Fábrica de Software en la Ciudad de Guayaquil. *Revista Tecnológica ESPOL*, 36.

Gerencia de Transito Viabilidad y Transporte. (2018). Reporte Infracciones Pendientes de Pago. Cusco, Cusco, Cusco.

Gomez, M. M. (2006). *Introducción a la metodología de la investigación científica.* Córdoba - Argentina.

Grupo de Investigacion Davincis. (26 de Junio de 2021). *Los Web Services y Caracteristicas de Calidad*. Obtenido de http://www.unilibre.edu.co/revistaavances/avances_10/r10_art7.pdf

JaeShup Oh, I. S. (04 de Diciembre de 2017). *A case study on business model innovations using Blockchain: focusing on financial institutions*. Obtenido de https://www.emerald.com/insight/content/doi/10.1108/APJIE-12-2017-038/full/html

Lazaro, D. (24 de Junio de 2021). *Introducción a los Web Services*. Obtenido de Introducción a los Web Services: https://diego.com.es/introduccion-a-los-web-services

Loaiza Peña, E. W. (20 de Setiembre de 2018). *Gobierno Municipal del Cusco*. Obtenido de https://www.cusco.gob.pe/nosotros/organigrama/

Microsoft. (30 de Noviembre de 2020). *Arquitectura de Windows Communication Foundation*. Obtenido de https://docs.microsoft.com/es-es/dotnet/framework/wcf/architecture

Municipalidad Provincial del Cusco. (05 de Octubre de 2018). *Reglamento de Organizciòn y Funciones (ROF).* Obtenido de https://www.cusco.gob.pe/wp-content/uploads/2015/10/rof-2013.pdf

Pinzon, I. (s.f.). *Gestion del Riesgo en Seguridad Informatica*. Obtenido de Universidad Piloto de Colombia: http://repository.unipiloto.edu.co/bitstream/handle/20.500.12277/2840/Gestion%20del%20riesgo%20en%20seguridad%20informatica.pdf?sequence=1&%3BisAllowed=y#:~:text

=Se%20entiende%20por%20riesgo%20de,tres%20caracter%C3%ADsticas%20principales%20de%20la

Rouse, M., & TechTarget. (22 de Junio de 2021). *Blockchain*. Obtenido de https://searchdatacenter.techtarget.com/es/definicion/Blockchain

Sanchez Herrera, S. A. (2021). *Sistema de voto electrónico basado en blockchain.* Lima: Pontifica Universidad Catolica del Perú.

Sanchez, J. C. (2011). *Metodología de la Investigación Científica y Tecnológica.* Madrid-España.

SUTRAN. (2014). *TEXTO ÚNICO ORDENADO DEL REGLAMENTO* . Lima - Perú.

TechClub. (30 de Noviembre de 2020). *WCF – Windows Communication Foundation*. Obtenido de https://techclub.tajamar.es/wcf-windows-communication-foundation/

ANEXOS

Anexo 1: Código Fuente - Método Recuperar Persona

```
665     #region Recupera Persona
        1 referencia
666     private WebService.ongei.Persona RecuperarPorWebService(string numeroDocumento)
667     {
668         // Se concatena la URL mas el DNI a buscar
669         string url = URLPERSONA + "?dni=" + numeroDocumento;
670         // Declaramos el documento de tipo XML
671         XmlDocument xmlDocument = new XmlDocument();
672         // Carga del documento XML desde la dirección URL
673         xmlDocument.Load(url);
674         // Declaramos el administrador de espacios y nombres mediante la tabla de nombres del lector XML
675         XmlNamespaceManager ns = new XmlNamespaceManager(xmlDocument.NameTable);
676         try
677         {
678             // Agregamos el espacio de nombre con el prefijo y la URL correspondiente
679             ns.AddNamespace("S", "http://schemas.xmlsoap.org/soap/envelope/");
680             ns.AddNamespace("w", "http://ws.reniec.gob.pe/");
681             // Se valida que el DNI ingresado exista
682             string errorCode = xmlDocument.SelectSingleNode(
683                         "//S:Envelope/S:Body/w:consultarResponse/return/coResultado", ns).InnerText;
684             if (errorCode != "0000") return null;
685             // Recuperamos los datos en la clase Persona
686             XmlNode nodoPersona = xmlDocument.SelectSingleNode(
687                         "//S:Envelope/S:Body/w:consultarResponse/return/datosPersona", ns);
688             WebService.ongei.Persona retval = new WebService.ongei.Persona
689             {
690                 DNI = numeroDocumento,
691                 ApellidoPaterno = nodoPersona.SelectSingleNode("apPrimer").InnerText,
692                 ApellidoMaterno = nodoPersona.SelectSingleNode("apSegundo").InnerText,
693                 Nombres = nodoPersona.SelectSingleNode("prenombres").InnerText,
694                 Direccion = nodoPersona.SelectSingleNode("direccion").InnerText,
695                 NroUbigeo = nodoPersona.SelectSingleNode("ubigeo").InnerText,
696                 Restriccion = nodoPersona.SelectSingleNode("restriccion").InnerText
697             };
698             retval.errorStatus = new ErrorStatus { ErrorCode = 0 };
699             return retval;
700         }
701         catch (Exception ex)
702         {
703             return new WebService.ongei.Persona
704             {
705                 errorStatus = new ErrorStatus { ErrorCode = 0xee, Message = ex.Message }
706             };
707             //#if DEBUG
708             //ErrorStatus errorCode = new ErrorStatus { ErrorCode = 0xee, Message = ex.Message };
709             //#else
710             //ErrorStatus errorCode = new ErrorStatus { ErrorCode = 0xee, Message = "No se pudo recuperar los datos de persona" };
711             //#endif
712             //throw new WebFaultException<ErrorStatus>(errorCode, System.Net.HttpStatusCode.InternalServerError);
713         }
714     }
```

Anexo 2: Código Fuente - Método Recuperar Vehículo

```
804     #region Recupera Vehiculo
        1 referencia
805     public WebService.ongei.Vehiculo RecuperarVehiculo(string placa)
806     {
807         // Se convierte a mayusculas la placa ingresada
808         placa = placa.ToUpper();
809         // declaramos la entidad
810         Entities dbContext = new Entities();
811         dbContext.Configuration.ProxyCreationEnabled = false;
812
813         // Se crea una lista de todas las zonas registrales según SUNARP
814         List<ZonaOficinaRegistral> zonas = dbContext.ZonaOficinaRegistral.OrderByDescending(x => x.orden).ToList();
815         try
816         {
817             // Se realiza la búsqueda en todas las zonas registrales.
818             foreach (ZonaOficinaRegistral zona in zonas)
819             {
820                 // Se concatena la URL vehículo mas la placa, la zona y la oficina registral.
821                 string url = URLVEHICULO + "?placa=" + placa + "&zona=" + zona.zona + "&oficina=" + zona.oficina;
822                 // se declara el documento de tipo XML
823                 XmlDocument xmlDocument = new XmlDocument();
824                 // Se realiza la carga del documento XML desde la direccion URL.
825                 xmlDocument.Load(url);
826                 // Realiza la busqueda del Vehículo, mientras sea null o vacio, la búsqueda continua
827                 XmlNode nodoPlaca = xmlDocument.SelectSingleNode("//out/placa");
828                 if (nodoPlaca == null || nodoPlaca.InnerText.Trim() == "")
829                 {
830                     Thread.CurrentThread.Join(500);
831                     continue;
832                 }
833                 // Recuperamos los datos del vehículo en la clase Vehículo
834                 XmlElement data = xmlDocument.DocumentElement;
835                 var retval = new WebService.ongei.Vehiculo
836                 {
837                     Placa = nodoPlaca.InnerText,
838                     Color = data.SelectSingleNode("color").InnerText,
839                     Serie = data.SelectSingleNode("serie").InnerText,
840                     Vin = data.SelectSingleNode("vin").InnerText,
841                     NumeroMotor = data.SelectSingleNode("nro_motor").InnerText,
842                     Marca = data.SelectSingleNode("marca").InnerText,
843                     Estado = data.SelectSingleNode("estado").InnerText,
844                     Sede = data.SelectSingleNode("sede").InnerText,
845                 };
846                 // Recuperamos la lista de propietarios
847                 foreach (XmlNode propietario in data.SelectNodes("propietarios/nombre"))
848                 {
849                     if (retval.Propietarios == null) retval.Propietarios = new List<string>();
850                     // Recuperamos los propietarios en la clase Propietarios
851                     retval.Propietarios.Add(propietario.InnerText);
852                 }
853                 retval.errorStatus = new ErrorStatus { ErrorCode = 0 };
854                 return retval;
855             }
856         }
857         catch (Exception ex)
858         {
859             return new WebService.ongei.Vehiculo
860             {
861                 errorStatus = new ErrorStatus { ErrorCode = 0xee, Message = ex.Message }
862             };
863         }
864         return null;
865     }
```

Anexo 3: Código Fuente - Método Recuperar Propietario

```
1 referencia
public WebService.ongei.Propietario RecuperarPropietario(string ApPaterno, string ApMaterno, string Nombres,
    string R_Social, string NroPlaca)
{
    // Se convierten los datos  a mayusculas
    ApPaterno = ApPaterno.ToUpper();
    ApMaterno = ApMaterno.ToUpper();
    Nombres = Nombres.ToUpper();
    R_Social = R_Social.ToUpper();
    // Declaramos una nueva entidad
    Entities dbContext = new Entities();
    dbContext.Configuration.ProxyCreationEnabled = false;
    try
    {
        int nro = 0;
        string url;
        // Si los datos de persona son nulos o vacios, declaramos la URL con tipo 'Persona Juridica'
        if ((ApPaterno == null || ApPaterno == "") && (ApMaterno == null || ApMaterno == ""))
        {
            url = URLPROPIETARIO + "?tipo=J" + "&apellido_paterno=" + ApPaterno + "&apellido_materno="
                                 + ApMaterno + "&nombres=" + Nombres + "&razon_social=" + R_Social;
        }
        // Si los datos de persona NO son nulos o vacios, declaramos la URL con tipo 'Persona Natural'
        else
        {
            url = URLPROPIETARIO + "?tipo=N" + "&apellido_paterno=" + ApPaterno + "&apellido_materno="
                                 + ApMaterno + "&nombres=" + Nombres + "&razon_social" + R_Social;
        }
        // Se declara el documento de tipo XML
        XmlDocument xmlDocument = new XmlDocument();
        // Se realiza la carga del documento XML desde la direccion URL.
        xmlDocument.Load(url);
        // Declaramos una lista de tipo xmlnodelist
        XmlNodeList xnodo = xmlDocument.SelectNodes("//xml/content/item");

        // Recorremos la lista buscando el numero de placa ingresado (el propietario puede tener varios vehiculos)
        for (int i = 0; i < xnodo.Count; i++)
        {
            string placa = xnodo[i].SelectSingleNode("numeroPlaca").InnerText;
            if (NroPlaca.Trim() == placa.Trim())
            {
                nro = i + 1;
            }
        }
        // recuperamos los datos del propietario según el item del vehiculo ingresado
        XmlNode nodoPropietario = xmlDocument.SelectSingleNode("//xml/content/item[" + nro + "]");
        WebService.ongei.Propietario retval = new WebService.ongei.Propietario
        {
            TipoDoc = nodoPropietario.SelectSingleNode("tipoDocumento").InnerText,
            NroDoc = nodoPropietario.SelectSingleNode("numeroDocumento").InnerText,
            Placa = nodoPropietario.SelectSingleNode("numeroPlaca").InnerText,
        };
        retval.errorStatus = new ErrorStatus { ErrorCode = 0 };
        return retval;
    }

    catch (Exception ex)
    {
        return new WebService.ongei.Propietario {
            errorStatus = new ErrorStatus { ErrorCode=0xee, Message = ex.Message }
        };
        //throw new WebFaultException<ErrorStatus>(errorCode, System.Net.HttpStatusCode.InternalServerError);
        //throw new System.Web.HttpException(ex.InnerException.Message);
        //throw new WebFaultException<ErrorStatus>(new ErrorStatus { ErrorCode = 0xee, Message = ex.Message }, System.Net.HttpStatusCode.OK);
    }
}
```

Anexo 4: Código Fuente - Método Recuperar Conductor

```
747         #region Recupera Conductor
            1 referencia
748         public WebService.ongei.Conductor RecuperarPorWebServiceConductor(string numeroDocumento)
749         {
750             // Se concatena la URL mas el numero de documento a buscar
751             string url = URLCONDUCTOR + "?accion=2&tipo_documento=2&documento=" + numeroDocumento;
752             // se declara el documento de tipo XML
753             XmlDocument xmlDocument = new XmlDocument();
754             // Se realiza la carga del documento XML desde la direccion URL.
755             xmlDocument.Load(url);
756
757             // Declaramos el administrador de espacios y nombres mediante la tabla de nombres del lector XML
758             XmlNamespaceManager ns = new XmlNamespaceManager(xmlDocument.NameTable);
759             try
760             {
761                 ns.AddNamespace("soap", "http://schemas.xmlsoap.org/soap/envelope/");
762                 ns.AddNamespace("xsi", "http://www.w3.org/2001/XMLSchema-instance");
763                 ns.AddNamespace("xsd", "http://www.w3.org/2001/XMLSchema");
764                 ns.AddNamespace("ns", "http://tempuri.org/");
765
766                 // Validamos que el documento ingresado exista
767                 string errorCode = xmlDocument.GetElementsByTagName("CodigoRespuesta")[0].InnerText;
768                 if (errorCode != "MSJ00") return null;
769
770                 // Recuperamos los datos en la clase Conductor
771                 XmlNode nodoConductor = xmlDocument.GetElementsByTagName("Licencia")[0];
772                 WebService.ongei.Conductor retval = new WebService.ongei.Conductor
773                 {
774                     DNI = numeroDocumento,
775                     Licencia = xmlDocument.GetElementsByTagName("NroLicencia")[0].InnerText,
776                     CategoriaLic = xmlDocument.GetElementsByTagName("Categoria")[0].InnerText,
777                     ApellidoPaterno = xmlDocument.GetElementsByTagName("ApellidoPaterno")[0].InnerText,
778                     ApellidoMaterno = xmlDocument.GetElementsByTagName("ApellidoMaterno")[0].InnerText,
779                     Nombres = xmlDocument.GetElementsByTagName("Nombre")[0].InnerText,
780                     Departamento = xmlDocument.GetElementsByTagName("Departamento")[0].InnerText,
781                     Provincia = xmlDocument.GetElementsByTagName("Provincia")[0].InnerText,
782                     Distrito = xmlDocument.GetElementsByTagName("Distrito")[0].InnerText,
783                     FechaExpedicion = DateTime.Parse(xmlDocument.GetElementsByTagName("Fechaexpedicion")[0].InnerText),
784                     FechaRevalidacion = DateTime.Parse(xmlDocument.GetElementsByTagName("Fecharevalidacion")[0].InnerText),
785                     Estado = xmlDocument.GetElementsByTagName("Estadolicencia")[0].InnerText,
786                     Direccion = xmlDocument.GetElementsByTagName("Direccion")[0].InnerText,
787                     Correlato = xmlDocument.GetElementsByTagName("Correlato")[0].InnerText
788                 };
789                 return retval;
790             }
791             catch (Exception ex)
792             {
793 #if DEBUG
794                 ErrorStatus errorCode = new ErrorStatus { ErrorCode = 0xee, Message = ex.Message };
795 #else
796                 ErrorStatus errorCode = new ErrorStatus { ErrorCode = 0xee, Message = "No se pudo recuperar los datos de conductor" };
797 #endif
798                 throw new WebFaultException<ErrorStatus>(errorCode, System.Net.HttpStatusCode.InternalServerError);
799             }
800         }
```