

GESTIÓN DE SLA (SERVICE LEVEL AGREEMENT) APLICANDO BLOCKCHAIN Y SMART CONTRACT PARA DOMINIOS DE REDES 5G/B5G: UNA PROPUESTA

Ariel Irigoyen Castro¹, Jorge Enrique Preciado Velasco², Miguel Alonso Arévalo³, José Lozano Risk⁴

^{1,2,3,4}CICESE, México

¹e-mail: irigoyen@cicese.edu.mx

²e-mail: jprecia@cicese.edu.mx

³e-mail: aalonso@cicese.edu.mx

⁴e-mail: jlozano@cicese.edu.mx

RESUMEN

La arquitectura de servicio de las redes 5G/B5G se basa en la tecnología de Network Slices para crear redes virtuales optimizadas para servicios específicos. Este trabajo propone un esquema de automatización para gestionar acuerdos de nivel de servicio (SLA) de segmentos de red (*Network Slices*) en redes 5G/B5G mediante contratos inteligentes (*Smart Contracts*) basados en *Blockchain*. En el esquema planteado, los proveedores presentan propuestas de SLA transformadas en contratos inteligentes mediante negociación para luego aplicar recompensas o penalizaciones según las métricas de Calidad de Servicio (QoS) proporcionadas por las API de los proveedores. Este enfoque busca reducir la carga operativa de los proveedores de servicios y garantizar la transparencia y la confianza al cliente proveyendo un entorno descentralizado, inmutable y auditable. Se espera resaltar en trabajos futuros la necesidad de analizar el impacto del uso de *Blockchain* en el desempeño de la implementación y la automatización de la gestión de SLA para *Network Slice*.

PALABRAS CLAVES: Contratos inteligentes, SLA, Blockchain, Network Slice

SLA (SERVICE LEVEL AGREEMENT) MANAGEMENT APPLYING BLOCKCHAIN AND SMART CONTRACT FOR 5G/B5G NETWORK DOMAINS: A PROPOSAL

ABSTRACT

The service architecture of 5G/B5G networks relies on network segment technology to create virtual networks optimized for specific services. This work proposes an automation scheme to manage Network Slice Service Level Agreements (SLAs) in 5G/B5G networks using Blockchain-based Smart Contracts. In the proposed scheme, providers present SLA proposals transformed into smart contracts through negotiation, then apply rewards or penalties according to the Quality of Service (QoS) metrics provided by the providers' APIs. This approach seeks to reduce the operational burden on service providers and guarantee transparency and trust to the client by providing a decentralized, immutable, and auditable environment. In future work, we expect to highlight the need to analyze the impact of using Blockchain on the implementation performance and automation of SLA management for Network Slice.

INDEX TERMS: Smart Contracts, SLA, Blockchain, Network Slice.

1. INTRODUCCIÓN

En las redes móviles de nueva generación, la creciente heterogeneidad de casos de uso y servicios, así como el aumento del tráfico, plantean desafíos significativos. En comparación con las redes 4G, donde diversos servicios comparten la misma infraestructura, la evolución hacia una arquitectura basada en servicios (SBA, por sus siglas en inglés *Service Based Architecture*) se vuelve esencial para abordar las necesidades específicas de aplicaciones como cirugías a distancia, realidad extendida y vehículos autónomos en las redes 5G.

El concepto de Segmentos de Red (NS, por sus siglas en inglés *Network Slices*) surge como una solución para satisfacer los requisitos de servicios diversificados [1], dividiendo la red física en redes virtuales optimizadas para aplicaciones o servicios específicos [2]. Los operadores encuentran beneficios en la eficiencia de recursos y en la capacidad de negociar estos recursos en forma de NS, creando un mercado en crecimiento para diversos interesados verticales, que se espera continúe en redes posteriores a 5G (B5G, por sus siglas en inglés *Beyond 5G*) [1], [3].

Los proveedores de servicios exploran modelos innovadores, como plataformas de virtualización de funciones de red (NFV, por sus siglas en inglés *Network Function Virtualization*), para ofrecer recursos y servicios mediante Acuerdos de Nivel de Servicio (SLA por las siglas en inglés *Service Level Agreement*). Sin embargo, la gestión del cumplimiento del SLA por parte de los proveedores y la necesidad de transparencia y agilidad por parte de los clientes plantean desafíos operativos y de confianza. Para el caso de discrepancias entre medidas y cálculos o ambigüedad del contrato, se requiere que la ejecución dependa de un mediador para resolver la disputa. Este mediador introduce costos adicionales, burocracia y retrasos en el proceso [4]. En estos casos, el proceso de cumplimiento de SLA involucra la intervención humana, lo cual puede conllevar a errores. Una alternativa para solventar los problemas mencionados es el uso de la tecnología Blockchain (BC, por las siglas en inglés *Blockchain*) [4]-[9].

Los Contratos Inteligentes (SC por sus siglas en inglés *Smart Contracts*) funcionan sobre la tecnología BC. Los SC son códigos que se ejecutan sobre la BC para facilitar la ejecución y cumplimiento de un acuerdo entre partes no fiables sin la participación de un tercero de confianza. Se pueden aprovechar las características de los SC como su inmutabilidad, auditabilidad, transparencia y su funcionamiento en entornos distribuidos para automatizar el ciclo de vida de los SLA. Sin embargo, en el contexto de redes B5G, el uso de BC y SC en la gestión de SLA para NS supone un reto, dado los requisitos exigentes de latencia, uso eficiente de energía y altas velocidades de transferencia de información, ya que el funcionamiento de BC se basa en algoritmos criptográficos y mecanismos de consenso que requieren del uso de recursos de cómputo, lo cual puede impactar en el desempeño de un esquema de gestión que incorpore BC.

En el entorno actual, donde los servicios de telecomunicaciones son críticos, los SC ofrecen una solución atractiva para facilitar que los proveedores cumplan con los niveles de servicio acordados. Estos contratos pueden rastrear métricas como el tiempo de actividad, el ancho de banda, la latencia y otros KPI (KPI, por sus siglas en inglés *Key Performance Indicators*), y desencadenar automáticamente, en caso de incumplimiento, penalizaciones o compensaciones al cliente. Además, al estar respaldados por la tecnología BC, los SC proporcionan un registro inmutable y descentralizado de los términos del SLA y su cumplimiento, lo que aumenta la transparencia y la confianza entre las partes involucradas. Si bien aún se encuentran en etapas tempranas de adopción, los SC representan una solución prometedora para mejorar la eficiencia, la transparencia y la responsabilidad en la prestación de servicios.

La idea de utilizar SC para la gestión de SLA en escenarios de telecomunicaciones inició hace relativamente poco tiempo. Por ejemplo, en [5] se presenta una solución para verificar de manera segura y dinámica el cumplimiento de los SLA utilizando BC y SC. La evaluación de esta solución muestra que los costos y tiempos de respuesta son aceptables, sin embargo, está limitada a una sola métrica que es la disponibilidad, y no aborda la compensación monetaria automática en caso de violaciones de SLA. Los autores proponen utilizar en trabajos futuros algún mecanismo de oráculos distribuidos, que son sistemas que permiten acceder a la información externa a la BC, para aumentar la confiabilidad en la obtención de las métricas.

En [6] se establece una metodología para evaluar el rendimiento y escalabilidad de BC en escenarios de redes 6G y se hace una evaluación experimental del desempeño de la BC de consorcio Quorum, donde se comprueba que con la configuración adecuada se pueden satisfacer los requisitos de rendimiento y escalabilidad para escenarios de compartición de recursos de red, de facturación y pago de servicios. Este estudio analiza el desempeño de la BC, la cual usa el mecanismo de consenso de tolerancia a Fallas Bizantinas de Estambul (IBFT siglas del inglés *Istanbul Byzantine Fault Tolerance*), pero no considera el impacto causado al utilizar otros mecanismos de consenso ni extiende sus resultados a comparaciones al utilizar otro tipo de BC.

En [7] se plantea un sistema que automatiza los procesos de acuerdo y monitoreo de SLA entre un centro de dirección de TI, que actúa como proveedor de servicios, y las diferentes secciones de la universidad, que son los usuarios de

estos. Utiliza contratos inteligentes en una red Blockchain privada lo que permite un seguimiento de los acuerdos SLA y un monitoreo continuo del cumplimiento de los objetivos de servicio. Se señala que a pesar de que la implementación de una BC privada conlleva costos adicionales y desafíos técnicos, el sistema demuestra ser rápido y eficiente. Sin embargo, no se debe dejar de considerar los costos y complejidades en el mantenimiento y operación de un sistema que se introducen al utilizar este tipo de BC.

Las investigaciones comentadas anteriormente no tratan específicamente la gestión de SLA de *Network Slices*, pero se puede tratar de forma similar. Por ejemplo, en [8] se presenta *NSBchain* como una solución de intermediación basada en BC que interconecta al proveedor de infraestructura 5G y a los inquilinos de la red. *NSBchain* utiliza contratos inteligentes para negociar y definir acuerdos de asociación entre múltiples dominios administrativos, lo que resulta en implementaciones de servicios de extremo a extremo complejas, sin embargo, no considera la monitorización. Esta función es posible si se agrega un oráculo en el esquema de *NSBchain* que se plantea. Dicho oráculo tiene la característica de poder acceder a información fuera de la BC y entregarla a un SC y así el SC puede verificar el cumplimiento o no del acuerdo.

Para la realización de pruebas antes de hacer despliegues de contratos inteligentes en una BC se hacen pruebas de forma local, como ocurre en el artículo [9]. En este estudio se realizó un experimento para evaluar el desempeño de una arquitectura de confianza basada en BC y SC para la gestión de SLA en redes 5G. La arquitectura utiliza contratos inteligentes para almacenar los SLA, monitorear las violaciones y calcular las compensaciones y penalizaciones según corresponda, sin embargo, solo utiliza dos métricas de monitoreo para los experimentos cuando el conjunto de métricas relacionadas pudiera impactar en el desempeño de dicha arquitectura, pero en este trabajo no se realiza la validación de la arquitectura propuesta en una BC real.

La implementación de BC y SC en un esquema descentralizado para el comercio de recursos de red sin depender de un corredor central es un estudio realizado por [4] donde se indica que el uso de BC y SC permite aumentar la confianza entre los operadores al eliminar un punto único de falla. Sin embargo, no se proporciona un análisis detallado de los costos asociados con las diferentes opciones de implementación de BC para este caso de uso específico. Se destacan las ventajas del enfoque descentralizado basado en BC para el arrendamiento de segmentos de red, pero también reconoce los desafíos de escalabilidad, rendimiento y costos que deben abordarse para implementaciones a gran escala.

Este trabajo propone un esquema para automatizar la gestión de SLA para NS mediante Contratos Inteligentes (SC) basados en la tecnología *Blockchain*. Los SC facilitarán la negociación, el establecimiento de SLA y el monitoreo de parámetros de Calidad de Servicio (QoS, por sus siglas en inglés *Quality of Service*), integrando funciones de intermediación y construyendo un sistema de reputación. Se utiliza la red Sistema de Archivos Interplanetarios (IPFS, por sus siglas en inglés *Inter-Planetary File System*) para el almacenamiento distribuido de documentos SLA. Con el enfoque descentralizado del esquema se busca reducir la carga operativa de los operadores y proporcionar transparencia a los clientes. Se identifica la necesidad de abordar los desafíos específicos de latencia, eficiencia energética y velocidades de transferencia en redes B5G al implementar Blockchain y SC. Se revisa la literatura existente, señalando la falta de integración completa de funciones clave, como la negociación, la gestión de pagos y recompensas, y la monitorización, así como la ausencia de un sistema de reputación basado en el cumplimiento de SLA en las propuestas analizadas.

Este trabajo propone una solución integral para la gestión automatizada de SLA para NS, se abordan algunas de las limitaciones identificadas en la literatura existente. Finalmente, se proporcionan recomendaciones para la implementación, así como actividades futuras relacionadas con este trabajo

2. FUNDAMENTOS

Para la comprensión de nuestra propuesta de automatización del ciclo de vida de SLA de *Network Slices* utilizando SC, es necesario hacer una exposición previa, en los apartados siguientes, de algunos conceptos e ideas que se relacionan en dicha propuesta.

SLA en NS

Un Acuerdo de Nivel de Servicio (SLA) es un contrato o acuerdo entre un proveedor de servicios y un cliente. Es un documento legal y detallado que suele incluir las funciones y responsabilidades a cumplir de todas las partes implicadas, como el periodo de duración, la calidad del servicio, y las penalizaciones [10]. Los SLA identifican y definen las necesidades del cliente y a la vez controlan sus expectativas del servicio con relación a la capacidad del proveedor. Los SLA deben simplificar asuntos complicados y proporcionar un marco de entendimiento, reducir las áreas de conflicto y favorecer el diálogo ante la disputa. También constituyen un punto de referencia para el proceso de mejora continua, ya que el poder medir adecuadamente los niveles de servicio es el primer paso para mejorarlos y de esa forma cumplir con los índices de desempeño (KPI), e índices de calidad (KQI, por sus siglas en inglés *Key Quality Indicators*), etc. [11].



Figura 1: Ciclo de vida de un SLA [14].

En el ámbito de redes de telecomunicaciones los SLA definen objetivos (SLO por sus siglas en inglés *Service Level Objectives*) que se basan en parámetros de QoS. En las redes 5G basadas en NS, cada unidad de métrica está relacionada con la calidad del servicio, como latencia, retardo, velocidad de datos, capacidad, rendimiento, movilidad, seguridad, consumo de energía, densidad de conexión, tiempo de respuesta, nivel de servicio, etc., definidas y estandarizadas por organizaciones, como la UIT (Unión Internacional de Telecomunicaciones), el ETSI (Por sus siglas en inglés *European Telecommunications Standards Institute*), entre otras [12]. Una vez establecido el SLA estas métricas deben ser monitoreadas bajo los diferentes umbrales que serán específicos según el servicio para el cual este destinado el NS. En [13] se plantea el ciclo de vida de un SLA, que se muestra en la Figura 1, como guía para automatizar la ejecución de SLA, en la cual se apoya nuestra propuesta.

Blockchain y Smart Contract

Una BC es un base de datos distribuida gestionada por una red *peer-to-peer* en la que los registros se almacenan como transacciones en un bloque. Para cada transacción, los datos de la transacción son firmados y verificados por otros participantes en la red mediante algoritmos criptográficos. Si la mayoría de los participantes está de acuerdo en que la transacción es válida, se añade un nuevo bloque a la cadena de bloques y se comparte con todos los demás nodos [13]. Aunque se le llamen transacciones estas pueden ser datos de cualquier tipo que estarán disponibles para los nodos de la *Blockchain* de manera inmutable.

Aparte de la criptografía, que incluye el *hash* criptográfico (función que genera claves de manera casi unívoca) y la firma digital, el éxito clave de *Blockchain* se basa en los mecanismos de consenso que determinan el rendimiento general y la escalabilidad de un sistema. El consenso se describe como el acuerdo entre un grupo de nodos sobre la veracidad de sus datos [14]. La correcta selección de un mecanismo de consenso es de gran importancia ya que impacta directamente en el desempeño de la *Blockchain* en cuanto a poder de cómputo, velocidad de las transacciones y latencia [6].

Los SC se almacenan y funcionan sobre la BC. En la actualidad se utilizan en aplicaciones de finanzas distribuidas, cadenas de suministro, seguros, voto electrónico, propiedad electrónica, entre otras [15]. Son pedazos de código que pueden escribirse en diferentes lenguajes de programación, y constituyen una herramienta de automatización que provee confianza para la ejecución de acuerdos. El lenguaje seleccionado depende de la plataforma de BC donde vaya a desplegarse el contrato. La ejecución de sus funciones se lleva a cabo por una máquina virtual que para el caso de la BC *Ethereum* (EVM, por las siglas en inglés *Ethereum Virtual Machine*) [16]. Un SC permite la respuesta a eventos que ocurran en la BC, respondiendo a condiciones preprogramadas y siendo capaz de efectuar pagos y transacciones de forma que queda registrada toda su actividad en la BC.

Oráculos

Los SC no tienen acceso a información externa a la BC (información del “mundo real”), solo a los datos de la BC a que pertenecen [17], por lo que se requiere de un sistema o servicio conocido como oráculos y atender eventos externos, es decir, que les permita tomar decisiones basadas en eventos fuera de la BC. Estos oráculos, según su tipo, pueden obtener la información requerida por un SC a partir de una API, o de dispositivos de Internet de las cosas (IoT, por las siglas del inglés *Internet of Things*), de la Web, entre otras fuentes. Se encargan de verificar dicha información y firmarla antes de enviarla a un SC luego de que este la solicite. Los oráculos pueden ser centralizados o descentralizados. El uso de un oráculo centralizado supone debe ser intolerante a fallas ya que, de presentar problemas, el oráculo ya no podría transferir datos al SC. En cambio, un oráculo descentralizado se compone de varios nodos y ante caída en alguno de ellos puede continuar operando, por lo que es preferible su uso para garantizar disponibilidad de un servicio que utilice oráculos.

Aplicaciones descentralizadas

En una BC el almacenamiento del código fuente, los datos de las transacciones y los protocolos de las aplicaciones se basan en una cadena de bloques que utiliza una red *peer-to-peer* descentralizada. Una aplicación descentralizada (dApp por sus siglas en inglés *Distributed Application*), es un tipo de aplicación informática que funciona en una red descentralizada de nodos, en lugar de depender de servidores centralizados y se caracterizan por su autonomía y no depender de empresas o administradores que decidan los cambios en la aplicación. Es un concepto incluso anterior a BC, sin embargo, en la actualidad existe un auge de las dApps basadas en BC y simplemente se les llama dApp [18]. Ante un posible ataque cibernético, las dApps ofrecen un mayor nivel de seguridad, dado que los ataques a las redes distribuidas de BC tendrían que dirigirse contra cada ordenador individual y resolver el cifrado criptográfico. Esto es difícil de realizar con una dApp, ya que su código fuente se almacena en una cadena de bloques y, por tanto, en muchos ordenadores diferentes e interconectados. Así, el código fuente de las dApps se almacena de forma descentralizada en la BC, se distribuye en la red y es ejecutable [19].

Según su diseño, en algunos casos se hace uso de una interfaz web que facilite la interacción de los usuarios con el SC, para lo que necesitan lo conocido como “billeteras virtuales” ya que requieren del intercambio de activos en la ejecución de determinadas funciones del SC.

IPFS

Para el almacenamiento y/o recuperación de documentos SLA, los autores de [20] señalan que la red IPFS está formada por ordenadores que ejecutan el software cliente IPFS. Cualquiera puede unirse a la red IPFS, ya sea como nodo IPFS que ejecuta el cliente IPFS, o como usuario de la red que almacena y recupera archivos. Se puede almacenar cualquier tipo de archivo, incluidos texto, música, vídeo e imágenes. A diferencia de HTTP, en IPFS los datos se

identifican por su contenido y no por su ubicación. Cuando se sube un archivo a IPFS, se genera un *hash* del contenido. Este *hash* identifica el contenido de forma única y puede utilizarse para recuperar el archivo. Si se sube un archivo diferente, el *hash* será completamente distinto, pero siempre es necesario volver a calcular el *hash* del archivo localmente para asegurar de que coincide con el *hash* IPFS original.

Interfaz de monitoreo de *Network Slice* para clientes verticales

La arquitectura del núcleo de 5G se compone de varias funciones de red (NF por las siglas en inglés *Network Function*) de forma que separa al plano de usuario del plano de control. Estas NF exponen su funcionalidad a través de Interfaces Basadas en Servicios (SBI, por las siglas del término en inglés, *Service Based Interfaces*), las cuales están implementadas como API RESTful sobre HTTP/2 [21] de forma que permiten el intercambio de información entre las NF.

Es muy importante exponer información sobre las capacidades de la red a aplicaciones internas y externas cuando los operadores desean integrar 5G con procesos industriales verticales. La estandarización de estas interfaces facilita la integración para los clientes verticales, especialmente aquellos con operaciones internacionales y relaciones de múltiples operadores [22].

Los operadores o proveedores de servicios pueden utilizar el bus SBI para extraer información sobre el desempeño del NS y así servir a sus clientes, a partir de una API, con los KPI que describen el comportamiento del NS arrendado. La API tendría *endpoints* a los que el cliente puede acceder para extraer la información necesaria para garantizar el cumplimiento del SLA sobre determinado NS. Anteriormente cada operador tuvo su propio diseño para esta API, pero 3GPP en su actualización *Release 16* definió un marco de trabajo común de API (CAPIF, por las siglas en inglés *Common API Framework*) para facilitar la exposición del núcleo de la red hacia nuevos habilitadores de aplicaciones de diversas industrias [23]. CAPIF es un marco de trabajo estandarizado que permite exponer las capacidades de la red 5G a través de API abiertas.

3. PROPUESTA DE GESTIÓN DE SLA UTILIZANDO BLOCKCHAIN Y SMART CONTRACT

En la Figura. 2 se muestra el esquema de la propuesta de gestión de SLA para NS usando SC. El esquema se compone de una dApp, que permite la interacción de clientes y proveedores con la BC.

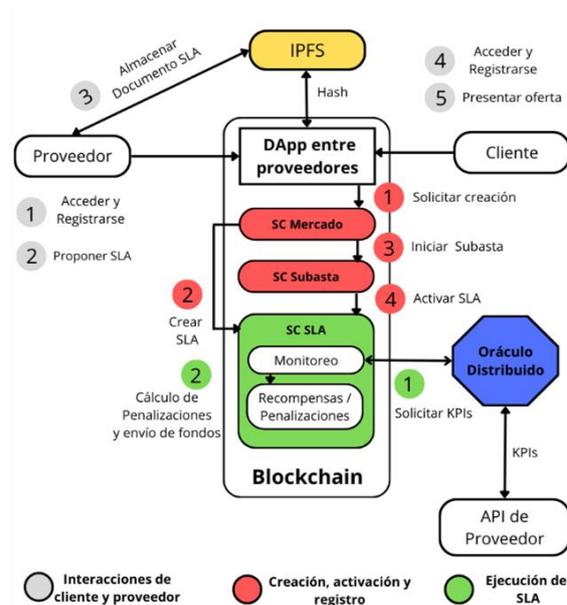


Figura 2: Esquema de gestión de SLA usando SC [Elaboración propia]

Los proveedores pueden enviar propuestas de SLA que queden registradas en IPFS en forma de documento, mientras que la dApp se encarga de validarla y registrarla en el SC Mercado teniendo en cuenta el *hash* (representación única de datos mediante una cadena de caracteres fija) del documento que se almacena en IPFS. SC Mercado transforma esta propuesta en un SC SLA que no tiene cliente asignado. Cada cliente puede hacer un descubrimiento por medio de la dApp de los SC SLA creados sin cliente asignado y ofrecer cierta cantidad de dinero por uno en específico. Esta oferta llega a SC Subasta a través de la dApp, el cual asignará un cliente a un SC SLA creado al determinar la oferta ganadora entre las realizadas por los clientes. Una vez establecido el cliente para un SC SLA, este último monitorea el cumplimiento del acuerdo y efectúa las recompensas y/o penalizaciones según las métricas solicitadas al oráculo, las cuales corresponden con el NS del proveedor asociado al contrato. El oráculo obtiene las métricas de la API expuesta para ese proveedor en específico. Toda la información de la actividad de los SC quedaría registrada en la BC, la que sería aprovechada por SC Sistema de reputación para exponer a los clientes la reputación de los diferentes proveedores teniendo en cuenta las violaciones en SLA anteriores.

Interacción con la BC

En la Figura 2 se muestra como los proveedores y clientes (que pueden ser otros proveedores) deben acceder y registrarse en la dApp para poder interactuar con la BC. Para ello deben poseer una billetera digital para almacenar activos, realizar transacciones y pagar de tarifas de gas (unidad que mide el costo de realizar operaciones o ejecutar contratos inteligentes en la red). Los proveedores podrán presentar sus propuestas de SLA en la dApp en forma de transacción almacenando un *hash*, el cual apunta al documento de SLA que será guardado en IPFS.

Este documento se guarda de manera distribuida por lo que estará disponible siempre y cuando se conozca su *hash*. La presentación de los acuerdos de SLA, que se hará en forma de transacción, debe contener los umbrales máximo y mínimo de los parámetros de QoS según se estipule en el SLA del NS en cuestión a arrendar. También debe contener las llaves de acceso de las API y *endpoints* correspondientes que se utilizarán para monitorear el cumplimiento del SLA, y la ventana de tiempo para la realización de la subasta del acuerdo.

Funcionalidades de los bloques de *Smart Contract*

SC Mercado será el programa encargado de crear los nuevos contratos y restringir acceso a la creación de contratos por entidades no registradas en la dApp. Así mismo, validará las propuestas de SLA para su despliegue.

SC Subasta llevará a cabo los procesos de negociación de los SLA el cual debe ocurrir en una ventana de tiempo y dará la posibilidad a los proveedores de establecer el acuerdo con el mejor postor. Definirá el proveedor y el cliente para el SC SLA a ser creado. En caso de empate, incorporará lógicas de resolución automática para seleccionar la oferta ganadora.

SC SLA será la plantilla para cada SLA a crear, contendrá los parámetros a monitorear y se alimentará de la información provisionada por un oráculo descentralizado para la ejecución del contrato. Será encargado de efectuar las recompensas o penalizaciones por ocurrencia de incidencias o indisponibilidad. También debe definir condiciones y lógicas para el cierre adecuado de un SLA, y la resolución de disputas.

SC Sistema de reputación se encargará de dar una puntuación a cada proveedor por el desempeño de los servicios previamente ofrecidos, datos que son extraídos de las recompensas y penalizaciones calculadas en los SC SLA correspondientes a cada proveedor.

Obtención de métricas

SC SLA se crea con la información de los datos necesarios para monitorear el cumplimiento del SLA. Estos datos provienen de una API diseñadas por los operadores de red para posibilitar la exposición de las funciones de red de su arquitectura 5G, añadiendo una capa de seguridad. Aunque existe CAPIF como estándar para la implementación de ellas cada operador se reserva el derecho de usar su propio estilo.

Diagramas de secuencia

Con el objetivo de representar de forma detallada las interacciones entre los componentes del esquema, durante el desarrollo del trabajo de investigación [24] se diseñaron diagramas de secuencia (casos de uso) que definen acciones a realizar tanto cliente como proveedor y expresan el flujo de ejecución para funcionalidades específicas basadas e implementadas para automatizar el ciclo de vida del SLA. Las funcionalidades definidas para el descubrimiento y establecimiento del SLA, se muestran en la Figuras 3a y Figura 3b respectivamente.

En los esquemas todas las acciones que realicen los usuarios son de forma directa con la dApp y esta hace las llamadas correspondientes a las funciones de los SC según sea el caso.

4. RECOMENDACIONES DE IMPLEMENTACIÓN

Para la implementación de la propuesta deben tenerse en cuenta cuáles son las tecnologías específicas que se utilizarán como billetera digital, plataforma de BC, lenguaje de SC, *framework* de desarrollo, oráculo a utilizar, API para consultar los KPI, así como las librerías para el desarrollo de la dApp.

Billetera Digital

En el esquema propuesto los clientes y proveedores deben utilizar una billetera digital para realizar transacciones relativas al establecimiento de un SLA en forma de SC. Una opción popular es *Metamask*, que es una extensión de navegador web y una aplicación móvil que le permite administrar claves privadas de Ethereum. Al hacerlo, sirve como billetera digital, y puede interactuar con aplicaciones descentralizadas, o dApps [25].

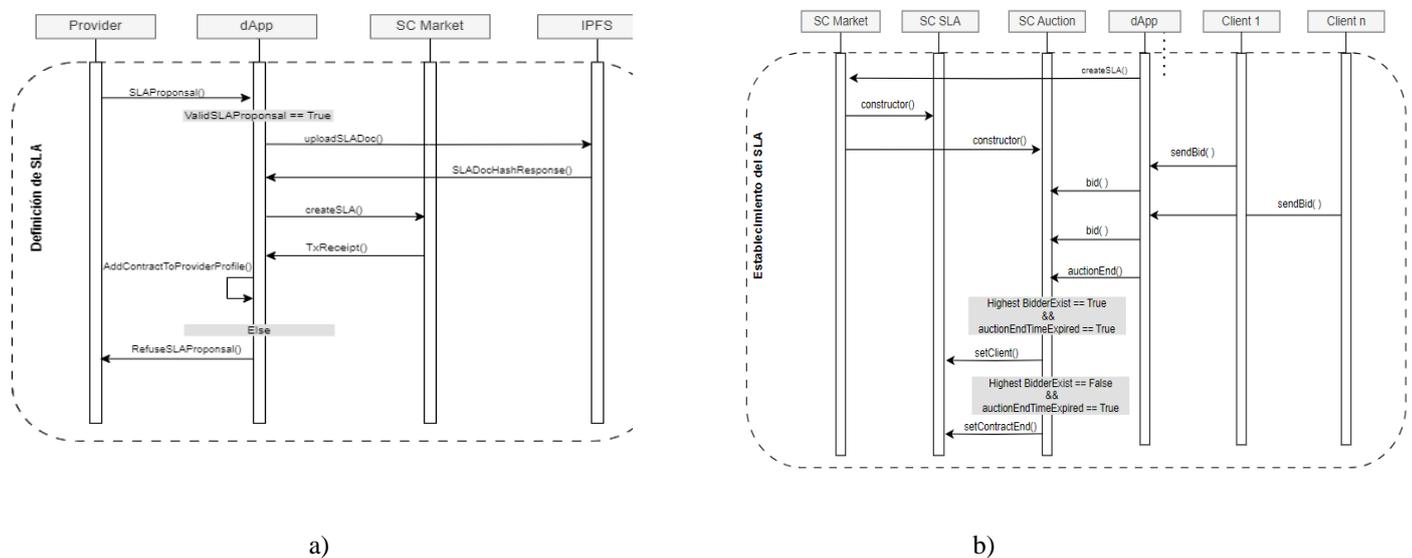


Figura 3: a) Diagrama de secuencia de definición del SLA, b) Diagrama de secuencia de establecimiento del SLA.

Selección de Blockchain

Otro elemento importante es la elección de la tecnología de BC a utilizar. Existen dos tipos de BC: las permissionadas y las no permissionadas. La implementación de una aplicación descentralizada (dApp) en una u otra presenta diferencias fundamentales que afectan su estructura y funcionamiento. En una BC permissionada, el acceso a la red

está restringido a un conjunto selecto de participantes conocidos, mientras que, en una no permissionada, la red está abierta a cualquier persona que desee unirse como se plantea en [26].

En un entorno de BC permissionado, la identidad de los participantes se conoce y se valida antes de que se les permita unirse a la red. Esto proporciona un grado mayor de confianza y control sobre la red, ya que los nodos se operan por entidades preaprobadas. Sin embargo, esta restricción puede limitar la descentralización y la resistencia a la censura, ya que la toma de decisiones y el consenso están más centralizados.

En contraste, en un BC no permissionado, la red es más abierta y accesible, lo que favorece la descentralización y la resistencia a la censura. Cualquier persona puede unirse y participar en la red sin necesidad de aprobación previa. Esto conlleva a una mayor diversidad de nodos y una menor posibilidad de manipulación por parte de un grupo selecto. Sin embargo, la apertura también puede dar lugar a desafíos en términos de seguridad y confianza, ya que los participantes no están necesariamente verificados.

Respecto a la eficiencia y velocidad de la red. En una BC permissionada, al tener un conjunto predefinido de nodos validados, las transacciones pueden procesarse más rápidamente ya que se requiere menos esfuerzo de consenso. En cambio, en una BC no permissionada, donde la red es más abierta, el consenso puede llevar más tiempo y ser más complejo, lo que puede afectar la velocidad de las transacciones. Sin embargo, requiere menos gastos. En las BC permissionadas se requiere inversión en desarrollo inicial y configuración de nodos, con un énfasis en la seguridad y el mantenimiento continuo. La infraestructura, licencias de software y servicios de soporte técnico también contribuyen a los costos. Es importante valorar estas características en la selección de la BC a utilizar.

Lenguaje de Programación de *Smart Contracts*

En las tablas I y II se aprecian los lenguajes de programación de SC que utilizan algunas BC permissionadas y no permissionadas respectivamente. Se puede observar que el lenguaje más utilizado es Solidity. Su popularidad radica en su diseño específico para el desarrollo de contratos inteligentes.

Tabla 1: Lenguajes de Smart Contract según Blockchain permissionadas

Plataforma BC	Lenguaje de SC
Hyperledger Fabric	Go, JavaScript
Corda	Kotlin, Java
Quorum	Solidity

Solidity facilita la creación de lógica de negocio descentralizada y autónoma al proporcionar una sintaxis similar a la de otros lenguajes de programación, como *JavaScript* y *C++* [27]. Se recomienda la utilización de este lenguaje por su posibilidad de reutilización en varias plataformas. Para la programación de SC en *Solidity* comúnmente se utiliza *Remix* que como se explica en [28] es un entorno de desarrollo en línea para contratos inteligentes en *Ethereum* y otras BC. Permite escribir, probar y desplegar fácilmente contratos inteligentes directamente desde el navegador, con funciones de edición, depuración y pruebas automatizadas. Su interfaz intuitiva y compatibilidad con varias BC hacen que sea una herramienta valiosa para los desarrolladores de contratos inteligentes.

Tabla 2: Características de Blockchain no permissionadas.

Plataforma	Lenguaje de Smart Contract	Mecanismo de consenso	de Testnet
Ethereum	Solidity	PoS	Ropsten, Rinkeby, Goerli, Sepolia
Cardano	Plutus	Ouroboros	Cardano Testnet
EOS	C++	dPoS	Jungle Testnet, Kylin Testnet
Tron	Solidity	DPoS	Shasta Testnet
Solana	Rust	PoS + PoH	Devnet
VeChain	Solidity	PoA	Testnet

Redes de prueba (*Testnet*)

Una idea importante previo a la implementación, es la posibilidad de simular el comportamiento del esquema a implementar, en la cuarta columna de la Tabla I, se muestran algunas redes de prueba (*Testnet*) que funcionan con fichas (*token*) falsos para la realización de transacciones, estos *tokens* se adquieren de plataformas que reciben el nombre de *faucets* para el caso de las BC no permissionadas (públicas). En el caso de las BC permissionadas las pruebas se realizan en *testnet* propias.

Las transacciones en una BC se realizan por parte de los nodos que pertenecen a ella. En el caso de una BC pública solo es necesario instalar un software que funcione como cliente para ser otro nodo, sin embargo, se requiere sincronización con la BC y esto significa que debe descargar la cadena completa lo cual puede significar gran demora. Una solución podría ser utilizar los servicios de proveedores de nodos.

Mecanismos de consenso

El mecanismo de consenso Prueba de Participación (PoS, por sus siglas en inglés *Proof of Stake*) hace que los nodos pongan una apuesta para que uno sea elegido aleatoriamente como creador del siguiente bloque. Cuando se elige un bloque, el creador recibirá las comisiones de transacción asociadas a ese bloque. Si el ganador de un bloque intenta añadir un bloque no válido, perderá su apuesta [29]. El uso de este protocolo va alineado con el uso eficiente de la energía por tanto una BC que use este protocolo como Ethereum es candidata para la implementación de la propuesta.

Otro mecanismo que también considera el uso eficiente de energía es Prueba de Autoridad (PoA por las siglas en inglés *Proof of Authority*). Para este los validadores deben pasar un riguroso procedimiento de investigación antes de que se les permita crear nuevos bloques [30]. La esencia este mecanismo es la confianza en la identidad del validador. Las personas con PoA tienen la motivación de mantener la posición que han alcanzado, ya que deben ganarse el privilegio de ser validadores. Al dar reputación a las identidades, se anima a los validadores a defender el proceso de transacción, ya que no quieren que sus identidades se asocien a una mala reputación [30]. *VeChain* como BC pública utiliza este mecanismo por tanto también es candidata para la implementación de la propuesta. Tanto *VeChain* como

Ethereum utilizan *Solidity* como lenguaje de programación para *Smart Contracts* lo que refuerza su selección como candidatos. Estas BC son públicas así que los costos asociados a implementación y pruebas serían menores.

Oráculo *Chainlink*

Un oráculo descentralizado actúa como un puente entre los contratos inteligentes y datos del mundo real, garantizando la integridad y la confiabilidad de las métricas utilizadas para los SLA. La red de nodos del oráculo *Chainlink* ofrece seguridad, resistencia a la censura y una conexión fluida con fuentes externas, asegurando la precisión de los datos críticos para la ejecución automática y confiable de SLA en entornos *Blockchain* [19]. Por lo que utilizar *Chainlink* para la obtención de métricas en la gestión automatizada de SLA mediante *Smart Contracts* ofrece una solución confiable y descentralizada.

Creación de API

Flask es recomendable para la creación de API sobre otros marcos de referencia (*frameworks*) debido a su simplicidad y flexibilidad. *Flask* adopta un enfoque minimalista, lo que facilita la creación rápida de API sin imponer una estructura rígida. Su diseño modular permite a los desarrolladores seleccionar y usar las herramientas que necesitan, lo que resulta en un desarrollo más ágil y personalizado. Además, *Flask* es conocido por su curva de aprendizaje suave, lo que lo hace accesible para aquellos que están comenzando en el desarrollo de API. Su naturaleza extensible y la amplia comunidad de usuarios y recursos disponibles también contribuyen a su popularidad en comparación con otros *frameworks* más complejos.

Diseño de la dApp

La librería *Web3* de *Python* facilita la interacción con SC y la BC, proporcionando una interfaz consistente y fácil de usar. Su sintaxis intuitiva y funciones integradas simplifican tareas complejas, como la firma de transacciones y la gestión de eventos. Además, la comunidad activa respalda constantemente su desarrollo, asegurando actualizaciones y soporte continuo. La librería *Web3* de *Python* ofrece una herramienta confiable y eficiente para desarrolladores que buscan crear dApps en la red *Ethereum* de manera efectiva y con código *Python* accesible. Es una elección sólida para el diseño de la dApp debido a su versatilidad y compatibilidad con la red *Ethereum*.

Análisis de desempeño

La metodología planteada en [6] puede utilizarse para el análisis evaluación del desempeño del esquema propuesto. Este enfoque implica analizar detalladamente el escenario para entender la necesidad de BC, definir qué debe registrarse en la cadena e identificar los momentos en que se generan transacciones de lectura y escritura. En este contexto, se elige un modelo de tasa de llegada de transacciones, como una distribución de Poisson, y se calculan las tasas de llegada teóricas de transacciones de lectura ($\lambda\alpha$) y escritura ($\lambda\beta$) basándonos en el escenario. Luego, se procede a implementar la BC y medir su rendimiento máximo en términos de operaciones por segundo de lectura y escritura. La comparación de estos valores con $\lambda\alpha$ y $\lambda\beta$ es esencial. Si la BC puede manejar las tasas de llegada del escenario, se considera adecuada. De lo contrario, se requieren ajustes como la consolidación de transacciones o la escalabilidad del sistema.

Desafíos y limitaciones

La validación del esquema propuesto enfrenta desafíos técnicos y presupuestarios debido a su complejidad y requisitos de infraestructura. Aunque el esquema permite la automatización de SLA, monitoreo de KPI y gestión de pagos para NS, no aborda la asignación directa de recursos en redes 5G ni la creación de NS reales. Esto se debe a que se requeriría una API real de un proveedor de NS de la cual no se dispone; en su lugar, se propone utilizar una abstracción de la API. Por otro lado, se plantea realizar la validación en BC locales o en *testnets*, más no en una BC real, ya que esto conllevaría un gasto excesivo de recursos, innecesarios para la investigación. Sin embargo, utilizar una BC real constituiría una mejor demostración de la capacidad del esquema propuesto para la gestión de SLA de NS. La opción elegida para la implementación del esquema en cuanto al tipo de BC utilizada es un factor determinante en el costo total de implementación. Se tiene opción de utilizar una BC pública o un BC privada. La pública requiere menor gasto

financiero en cuanto a infraestructura, pero se sacrifica la privacidad de los datos de monitoreo para los proveedores. En cambio, utilizar un BC privada garantiza la privacidad de los datos, pero los costos en hardware y poder de cómputo son más elevados lo cual es una limitante.

5. TRABAJO FUTURO

De los análisis realizados, se plantea implementar la propuesta utilizando las plataformas *Ethereum* y *VeChain*, así como también una BC permissionada como *Hyperledger Fabric*. El objetivo será comparar el desempeño del esquema en cada BC para evaluar su *throughput*, latencia y costos. Para el caso de *Hyperledger Fabric* se pretende utilizar varios mecanismos de consenso para esta evaluación. Así mismo, hacer el diseño e implantación de la dApp utilizando el *framework Flask* y la librería *web3* de *Python*. Se construirá la lógica y código de los SC en *Solidity* para reutilizarse en cada implementación independientemente de la BC utilizada.

6. CONCLUSIONES

Este trabajo presentó una propuesta para la automatización de la gestión de SLA para NS en entornos de redes B5G utilizando SC ejecutándose sobre *Blockchain*. El objetivo es dotar de mayor agilidad, transparencia y confiabilidad este proceso que en la actualidad resulta costoso y burocrático tanto para los proveedores como para los clientes de los servicios de NS. La propuesta de esquema contempla la utilización de diferentes SC para funciones específicas dentro del ciclo de vida del SLA. Siendo diseñados de forma modular, interactúan de forma que integran funciones de negociación, establecimiento de SLA, monitorización, pagos por recompensas / penalizaciones y gestión de reputación. Para la implementación de esta arquitectura se realizaron algunas recomendaciones en cuanto a la selección de plataforma BC, lenguaje de programación de los contratos, *framework* para API, librerías para el *frontend* de la aplicación descentralizada y mecanismos para evaluar el impacto en desempeño que la BC pueda introducir.

Este sistema distribuido de gestión SLA para NS ofrece un entorno transparente y confiable para la relación entre proveedores y clientes de estos servicios sin necesidad de terceros intermediarios. El potencial de esta propuesta radica en la posibilidad de reducir costos operacionales y aumentar la eficiencia en la gestión de la calidad de servicio de las redes del futuro, impactando positivamente modelos de negocios que habiliten nuevos casos de uso con requerimientos estrictos, como los previstos para redes B5G.

Reconocimientos

Los autores desean agradecer al Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California (CICESE) y a CONAHCYT por la beca otorgada al estudiante para cursar el posgrado.

7. REFERENCIAS BIBLIOGRÁFICAS

- [1] W. Wu *et al.*, “AI-Native Network Slicing for 6G Networks,” May 2021, [Online]. Available: <http://arxiv.org/abs/2105.08576>
- [2] GSM Association, “An Introduction to Network Slicing,” 2017. [Online]. Available: www.gsma.com.
- [3] Z. Ren, X. Li, Q. Jiang, Y. Wang, J. Ma, and C. Miao, “Network Slicing in 6G: An Authentication Framework for Unattended Terminals,” *IEEE Netw.*, vol. 37, no. 1, pp. 78–86, 2023, doi: 10.1109/MNET.112.2100738.
- [4] N. Afraz, F. Wilhelmi, H. Ahmadi, and M. Ruffini, “Blockchain and Smart Contracts for Telecommunications: Requirements vs. Cost Analysis,” *IEEE Access*, vol. 11, pp. 95653–95666, 2023, doi: 10.1109/ACCESS.2023.3309423.
- [5] J. P. De Brito Goncalves, R. Lima Gomes, R. Da Silva Villaca, E. Municio, and J. Marquez-Barja, “A Quality of Service Compliance System Empowered by Smart Contracts and Oracles,” in *Proceedings - 2020 IEEE International Conference on Blockchain, Blockchain 2020*, Institute of Electrical and Electronics Engineers Inc., Nov. 2020, pp. 532–538. doi: 10.1109/Blockchain50366.2020.00077.

- [6] B. Li, S. Deng, X. Yan, and S. Dustdar, "The Confluence of Blockchain and 6G Network: Scenarios Analysis and Performance Assessment," Jul. 2022, [Online]. Available: <http://arxiv.org/abs/2207.04744>
- [7] Z. F. Azzahra and I. G. B. B. Nugraha, "Service-Level Agreement Management with Blockchain-based Smart Contract to Improve the Quality of IT Service Management.," Association for Computing Machinery (ACM), Feb. 2023, pp. 260–266. doi: 10.1145/3587828.3587867.
- [8] X. Costa-Pérez, V. Sciancalepore, L. Zanzi, and A. Albanese, "Blockchain for Mobile Networks," in *Blockchains*, Wiley, 2023, pp. 185–213. doi: 10.1002/9781119781042.ch7.
- [9] S. Ben Saad, A. Ksentini, and B. Brik, "A Trust architecture for the SLA management in 5G networks," in *IEEE International Conference on Communications*, Institute of Electrical and Electronics Engineers Inc., Jun. 2021. doi: 10.1109/ICC42927.2021.9500990.
- [10] L.-J. Jin, V. Machiraju, and A. Sahai, "Analysis on Service Level Agreement of Web Services," 2002. Accessed: Jul. 15, 2024. [Online]. Available: <https://www.csd.uoc.gr/~hy565/docs/pdfs/papers/HPL-2002-180.pdf>
- [11] E. Kapassa, M. Touloupou, and D. Kyriazis, "SLAs in 5G: A Complete Framework Facilitating VNF- and NS-Tailored SLAs Management," in *2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, IEEE, May 2018, pp. 469–474. doi: 10.1109/WAINA.2018.00130.
- [12] M. A. Habibi, B. Han, M. Nasimi, and H. D. Schotten, "The Structure of Service Level Agreement of Slice-based 5G Network," Jun. 2018, [Online]. Available: <http://arxiv.org/abs/1806.10426>
- [13] C. Schweizer Zürich, "SLAMer: a blockchain-based SLA Management System," Tesis de licenciatura, Universidad de Zurich, Zurich, Switzerland, 2019. [Online]. Available: <http://www.csg.uzh.ch/>
- [14] W. Viriyasitavat and D. Hoonsopon, "Blockchain characteristics and consensus in modern business processes," *J Ind Inf Integr*, vol. 13, pp. 32–39, Mar. 2019, doi: 10.1016/j.jii.2018.07.004.
- [15] T. Kerikmäe and A. Rull, *The future of law and etechnologies*. Springer International Publishing, 2016. doi: 10.1007/978-3-319-26896-5.
- [16] "Ethereum Virtual Machine (EVM)." Accessed: Nov. 05, 2023. [Online]. Available: <https://ethereum.org/en/developers/docs/evm/>
- [17] A. Beniiche, "A Study of Blockchain Oracles," Mar. 2020, [Online]. Available: <http://arxiv.org/abs/2004.07140>
- [18] P. Zheng, Z. Jiang, J. Wu, and Z. Zheng, "Blockchain-based Decentralized Application: A Survey," 2023, doi: 10.1109/OJIM.2022.1234567.
- [19] Anónimo, "¿Que es una Dapp? Definición y funcionalidad," IONOS. Accessed: Dec. 27, 2023. [Online]. Available: <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/dapp/>
- [20] N. Sangeeta and S. Y. Nam, "Blockchain and Interplanetary File System (IPFS)-Based Data Storage System for Vehicular Networks with Keyword Search Capability," *Electronics (Switzerland)*, vol. 12, no. 7, Apr. 2023, doi: 10.3390/electronics12071545.
- [21] 3GPP, "3GPP 23501 System Architecture for the 5G System," vol. Vol. 16.0.2, no. Número Realease 16, p. 308, 2019, Accessed: Nov. 24, 2023. [Online]. Available: . <http://www.3gpp.org>
- [22] Telecommunication Engineering Centre Khurshid Lal Bhawan, "5G Core Network," 2021. Accessed: Nov. 16, 2023. [Online]. Available: https://www.tec.gov.in/public/pdf/Studypaper/5G%20Core%20Network_Study%20Paper_v8.pdf
- [23] A. S. Charismiadis, J. M. Salcines, D. Tsolkas, D. A. Guillen, and J. G. Rodrigo, "The 3GPP Common API framework: Open-source release and application use cases," in *2023 Joint European Conference on Networks and Communications and 6G Summit, EuCNC/6G Summit 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 472–477. doi: 10.1109/EuCNC/6GSummit58263.2023.10188337.
- [24] A. Irigoyen-Castro and J. E. Preciado-Velasco, "Gestión de SLA (Service Level Agreement) Aplicando Blockchain y Smart Contract para dominios de redes 6G," Tesis de Maestría, CICESE, Ensenada, Baja California, 2024, En proceso de publicación.
- [25] "Comenzar con MetaMask," Metamask.io. Accessed: Nov. 23, 2023. [Online]. Available: <https://support.metamask.io/hc/es/articles/360015489531-Comenzar-con-MetaMask>
- [26] D. Yaga, P. Mell, N. Roby, and K. Scarfone, "Blockchain technology overview," Gaithersburg, MD, Oct. 2018. doi: 10.6028/NIST.IR.8202.
- [27] "Solidity." Accessed: Nov. 23, 2023. [Online]. Available: <https://docs.soliditylang.org/en/v0.8.23>
- [28] V. Tiwari Bosco, S. Kumar Singh, V. Rao Vadi Bosco, V. Tiwari, and V. Rao Vadi, "Smart Contract Using Solidity (Remix-Ethereum IDE)," *Don Bosco Technical School, Delhi International Journal of Advanced Research in Computer and Communication Engineering ISO*, vol. 3297, no. 2, 2007, doi: 10.17148/IJARCC.2023.12253.
- [29] S. Y. Lin, L. Zhang, J. Li, L. li Ji, and Y. Sun, "A survey of application research based on blockchain smart contract," *Wireless Networks*, vol. 28, no. 2, pp. 635–690, Feb. 2022, doi: 10.1007/s11276-021-02874-x.

- [30] S. Fahim, S. Katibur Rahman, and S. Mahmood, “Blockchain: A Comparative Study of Consensus Algorithms PoW, PoS, PoA, PoV,” *International Journal of Mathematical Sciences and Computing*, vol. 9, no. 3, pp. 46–57, Aug. 2023, doi: 10.5815/ijmsc.2023.03.04.

SOBRE LOS AUTORES

Ariel Irigoyen Castro: nació en La Habana, Cuba, en el año 1995. Recibió su grado de Ingeniero en Telecomunicaciones y Electrónica de la Universidad Tecnológica de La Habana (CUJAE), en 2020. Fue profesor de Líneas de Transmisión y Procesamiento Digital de Señales en la CUJAE (2020-2022). En 2022 fue admitido en el posgrado de Telecomunicaciones y Electrónica de la facultad de física aplicada de CICESE. ORCID: <https://orcid.org/0009-0005-8466-9418>

Jorge Enrique Preciado Velasco: (miembro senior de la IEEE y miembro de la ACM) nació en Ensenada, México. Recibió su grado de Ingeniero en Comunicaciones y Electrónica de la Universidad de Guadalajara en 1977, su grado de Maestro en Ciencias en Electrónica y Telecomunicaciones del CICESE en 1983, y su grado de Doctor en Ciencias en Telecomunicaciones de la UABC en 2021. Ha sido en dos ocasiones presidente de la Junta Directiva de CUDI, CIO de la Universidad de Colima (2008–2012), México, y director de la División de Telemática en CICESE (1997–2005). Desde 1988, ha sido investigador en el Departamento de Electrónica y Telecomunicaciones del CICESE. Sus intereses de investigación incluyen la gestión de redes y servicios TIC, la nueva generación comunicaciones inalámbricas y QoS en redes. ORCID: <https://orcid.org/0000-0003-4543-2301>

Miguel Ángel Alonso Arévalo nació en Guadalajara, México. Cursó la licenciatura en Ing. Electrónica en el Instituto Tecnológico y de Estudios Superiores de Occidente (ITE-SO) con especialidad en Telecomunicaciones. De 1998 al 2000 cursó la maestría en Telecomunicaciones conjuntamente en el CINVESTAV Unidad Guadalajara y en la Escuela Nacional Superior de Telecomunicaciones de Breñaña, en Francia. Después trabajó como ingeniero de verificación en Intel Guadalajara. Posteriormente, en 2002 regresó a Francia a estudiar el doctorado en procesamiento digital de señales e imágenes en Télécom-Paritech en París, Francia. De 2007 a 2010 realizó un postdoctorado en la Universidad Tecnológica de Eindhoven. Es investigador en el Departamento de Electrónica y Telecomunicaciones del CICESE desde diciembre de 2010. Es miembro del SNI nivel I. Sus áreas de interés son las aplicaciones del procesamiento digital de señales e imágenes y del aprendizaje de máquinas. ORCID: <https://orcid.org/0000-0001-5453-3142>

José Eleno Lozano Rizk recibió su título de Ingeniero en Computación de la Facultad de Ingeniería, Arquitectura y Diseño (FIAD) de la Universidad Autónoma de Baja California (UABC, Ensenada, México) en 2003. En 2007 recibió una Maestría en Ingeniería en Computación y, en 2019, recibió el título de Doctor en Ciencias de la UABC. Actualmente se desempeña como Jefe de Departamento de Cómputo del Centro de Investigación CICESE en la Dirección de Telemática. Sus intereses de investigación son el cómputo de alto rendimiento (HPC), el análisis de *big data*, los sistemas distribuidos, el Internet de las cosas (IoT) y las redes definidas por software (SDN). ORCID: <https://orcid.org/0000-0002-6154-5712>

CONFLICTO DE INTERESES

No existe conflicto de intereses entre los autores, ni con ninguna institución a la que cada uno está afiliado, ni con ninguna otra institución. Las opiniones expresadas aquí son únicamente responsabilidad de los autores

CONTRIBUCIONES DE LOS AUTORES

- **Autor 1:** 40 % conceptualización, creación y desarrollo del artículo.
- **Autor 2:** 30 % revisión crítica de cada una de las versiones del borrador del artículo y aprobación de la versión final a publicar, conceptualización.
- **Autor 3:** 15 % contribución a la idea y organización del artículo

- **Autor 4:** 15 % contribución a la idea y organización del artículo

Esta revista provee acceso libre inmediato a su contenido bajo el principio de hacer disponible gratuitamente investigación al público. Los contenidos de la revista se distribuyen bajo una licencia *Creative Commons Attribution-NonCommercial 4.0 Unported License*. Se permite la copia y distribución de sus manuscritos por cualquier medio, siempre que mantenga el reconocimiento de sus autores y no se haga uso comercial de las obras.

