UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN

TRABAJO DE INTEGRACIÓN CURRICULAR

TEMA:

"MECANISMO DE AUTENTICACIÓN UTILIZANDO CADENA DE BLOQUES EN UN ENTORNO INALÁMBRICO DE PRUEBAS EN LA FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS"

Trabajo de titulación previo a la obtención del título de Ingeniero en Electrónica y Redes de Comunicación

Línea de investigación: Innovación tecnológica y de productos

AUTOR:
Galo Mauricio Beltrán Manosalvas

DIRECTOR:
Msc. Fabián Geovanny Cuzme Rodríguez.

Ibarra, Ecuador 2026

UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	0401678867		
APELLIDOS Y NOMBRES:	Beltrán Manosalvas Galo Mauricio		
DIRECCIÓN:	Panamericana Norte, Barrio El Pailón, Parroquia Los Andes, Cantón Bolívar, Carchi		
E-MAIL:	gmbeltran@utn.edu.ec		
TELÉFONO FIJO:		**TELÉFONO MÓVIL:**	0989779363

DATOS DE LA OBRA	
TÍTULO:	MECANISMO DE AUTENTICACIÓN UTILIZANDO CADENA DE BLOQUES EN UN ENTORNO INALÁMBRICO DE PRUEBAS EN LA FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
AUTOR:	BELTRÁN MANOSALVAS GALO MAURICIO
FECHA:	27/02/2026
PROGRAMA:	■ **PREGRADO** ☐ **POSGRADO**
TÍTULO POR EL OPTA:	INGENIERO EN ELECTRÓNICA Y REDES DE COMUNICACIÓN
DIRECTOR:	MSC. FABIÁN GEOVANNY CUZME RODRÍGUEZ
ASESOR:	MSC. HERNÁN MAURICIO DOMÍNGUEZ LIMAICO

2. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 27 días del mes de febrero de 2026

EL AUTOR:

Galo Mauricio Beltrán Manosalvas

CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Ibarra, 27 de febrero de 2026

MSC. FABIÁN GEOVANNY CUZME RODRÍGUEZ

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:

Haber revisado el presente informe final del trabajo de Integración Curricular, el mismo que se ajusta a las normas vigentes de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.

MSc. Fabián Geovanny Cuzme Rodríguez

C.C.: 1311527012

DEDICATORIA

De manera especial para mis abuelitos Zoilo, Laura, Olmedo y Marina; que, con sus enseñanzas, experiencias, don de gente y tesón contribuyeron en mi formación personal y de carácter desde mis primeros años de vida.

A mis padres Galo Gilberto y Ligia Marlene, por su entrega para conmigo y mis hermanos; depositado su confianza y apoyándonos paso a paso en cada etapa de nuestras vidas.

A mis hermanos, Virginia, Nathaly, Geovanny, Deyvid y Ruby; con quienes crecí y sigo compartiendo vivencias y aprendiendo de ellos; a mis sobrinos, Brandon y la pequeña Montse que me motivan a esforzarme y ser un buen ejemplo.

A todos mis tíos, tías y primos; que en algún punto me extendieron su ayuda y también son un ejemplo de superación, en especial a mi tía Vicky y su familia que siempre apoya y ayuda a sus sobrinos.

Finalmente, a todos esos compañeros y amigos que el alma mater me permitió encontrar y conocer, con los que compartí alegrías, desaciertos, preocupaciones y desvelos dentro y fuera de un aula de clase, fortaleciendo una amistad a lo largo de esta etapa académica; es por y para ustedes.

Con sinceridad:

Galo Mauricio

AGRADECIMIENTO

Agradezco a mis padres, a su ejemplo de vida, sus enseñanzas, valores y al inmenso esfuerzo que han realizado apoyándome a lo largo de mi vida y en las diferentes etapas del ámbito académico; inculcando en mí la perseverancia, el valor del trabajo honesto y el amor a la familia, siendo mis referentes morales e inspiración para supérame a diario personal y profesionalmente.

A mis hermanos, los cuales son mi modelo a seguir quienes, con su motivación y afecto, han influido en mi crecimiento personal y formación profesional; mi gratitud también a toda mi familia extendida quienes de una u otra manera me han apoyado y me han brindado su aliento para alcanzar y cumplir mis metas.

A la Universidad Técnica del Norte, a la Facultad de Ingeniería en Ciencias Aplicadas, al cuerpo docente y personal administrativo de la Carrera de Ingeniería en Electrónica y Redes de Comunicación quienes aportaron con su conocimiento y valiosa ayuda a lo largo de mi vida universitaria; así como también el Departamento de Desarrollo Tecnológico e Informático de la universidad por la apertura para brindar y facilitar la información requerida para el contexto y desarrollo de este trabajo de titulación.

A mi director de tesis, MSc. Fabián Cuzme; por la paciencia, tiempo y experticia dedicados; al MSc. Mauricio Domínguez, por aportar su guía, comentarios y recomendaciones en calidad de asesor a fin de desarrollar este trabajo de titulación; finalmente también mencionar a todos los profesores a quienes he tenido el agrado de conocer y que han aportado con sus valiosos conocimientos en mi formación académica y profesional.

¡Gracias totales!

Galo Mauricio Beltrán Manosalvas

RESUMEN

La tecnología de Cadena de Bloques ha surgido como una solución innovadora para mejorar mecanismos de seguridad y autenticación brindando; confidencialidad e integridad ya que las transacciones y procesos se registran en un libro distribuido y se validan mediante algoritmos de consenso, pruebas de trabajo y contratos inteligentes; de manera descentralizada y sin la participación de terceras partes.

Este proyecto de tesis propone el desarrollo de un mecanismo de autenticación inalámbrico basado en la tecnología de cadena de bloques; desplegado en una red de pruebas en la Facultad de Ingeniería en Ciencias Aplicadas de la Universidad Técnica del Norte, tomando como referencia y base teórica a los mecanismos de autenticación, seguridad y conexión inalámbrica existentes y usados en redes WiFi con estándar IEEE 802.11; para el planteamiento de este caso de estudio.

En el desarrollo de este proyecto se aplica la metodología en cascada en donde se cumplieron etapas procedimentales para el mecanismo, y donde se analizaron diferentes formas en la que se puede aprovechar la tecnología de cadena de bloques como parte de la solución al mecanismo de autenticación propuesto; aprovechando las capacidades de validación de los contratos inteligentes, algoritmos de consenso y/o pruebas de trabajo, y que mediante investigación y búsqueda se encontró plataformas que permitan levantar y experimentar en entornos de cadenas de bloques de pruebas.

El mecanismo de autenticación propuesto, basado en tecnología de cadena de bloques, fue implementado en un entorno de pruebas mediante una interfaz de acceso y conexión inalámbrica. Aprovechando las características de seguridad descentralizada de las cadenas de bloques, se implementó un sistema de autenticación de usuarios que ofrece una alternativa al mecanismo convencional utilizado por el protocolo WiFi, obteniendo así un enfoque diferente de convergencia

en seguridad para el proceso de autenticación y el uso de la tecnología de cadena de bloques. Las pruebas de concepto realizadas a esta solución permitieron a dispositivos y usuarios ingresar una dirección de cadena de bloques válida como credencial para acceder a los recursos de red, con un intervalo de confianza entre 557.10 ms y 688.15 ms.

Palabras Clave: Cadena de Bloques, Seguridad, Autenticación, WiFi, Mecanismo, Validación, IEEE 802.11, Convergencia.

ABSTRACT

Blockchain technology has emerged as an innovative solution to improve security and authentication mechanisms, providing confidentiality and integrity, as transactions and processes are recorded in a distributed ledger and validated through consensus algorithms, proofs of work, and smart contracts; in a decentralized and third-party-free manner.

This thesis project propose the development of a wireless authentication mechanism based on blockchain technology, implementing a test network in the Facultad de Ingeniería en Ciencias Aplicadas of the Universidad Técnica del Norte, it references and is theoretically based on existing and widely used authentication, security, and wireless connection mechanisms in WiFi networks with IEEE 802.11 standard, for the development of this case study.

In the development of this project, the waterfall methodology was applied, where procedural stages for the mechanism were completed, and where different forms in which blockchain technology can be used as part of the solution to the proposed authentication mechanism were analyzed; taking advantage of the validation capabilities of smart contracts, consensus algorithms and/or proof of work, and through research and search, platforms were found that allow setting up and experimenting in test blockchain environments.

The proposed authentication mechanism, based on blockchain technology, was implemented in a test environment using a wireless access and connection interface. Leveraging the decentralized security features of blockchains, a user authentication system was implemented that offers an alternative to the conventional mechanism used by the Wi-Fi protocol, thus providing a different approach to security convergence for the authentication process and the use of blockchain technology. Proof-of-concept tests of this solution allowed devices and users to enter

a valid blockchain address as a credential to access network resources, with a confidence interval between 557.10 ms and 688.15 ms.

Keywords: Blockchain, Security, Authentication, WiFi, Mechanism, Validation, IEEE 802.11, Convergence.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1. Capítulo I: Antecedentes

Este capítulo detalla los lineamientos iniciales a tomar en cuenta para el desarrollo del siguiente trabajo de titulación, donde se describen; el tema, el planteamiento del problema, los objetivos, el alcance y su justificación, a fin de establecer la importancia y limitaciones para el desarrollo de esta investigación.

1.1. Tema

"MECANISMO DE AUTENTICACIÓN UTILIZANDO CADENA DE BLOQUES EN UN ENTORNO INALÁMBRICO DE PRUEBAS EN LA FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS"

1.2. Problema

Tomado como referencia a la tecnología de acceso a la infraestructura de red inalámbrica usada en la Facultad de Ingeniería en Ciencias Aplicadas (FICA), que utiliza un servidor FreeRadius/LDAP con EAP 802.1X centralizado, ubicado en la DMZ de la universidad y el cual no cuenta con su respectivo equipo de respaldo y que además este servidor debe ser compatible y se debe sincronizar con la Wireless LAN Controller (WLC) de la universidad (GARRIDO, 2018), para integrar el método de acceso y validación en los Puntos de Acceso o APs (Lederkremer, 2019) de la facultad; los cuales propagan los Identificadores de Red, Service Set Identifier o SSID (Coleman et al., 2016) utilizados para acceder a la infraestructura de red inalámbrica y que además garantizan la conexión a Internet del personal administrativo, docentes y estudiantes; siendo el único sistema de validación y control de usuarios.

Actualmente, en la Universidad Técnica del Norte como en la Facultad de Ingeniería en Ciencias Aplicadas, la administración y acceso a la infraestructura inalámbrica de red, depende de equipos centralizados; donde, la WLC, junto con un servidor FreeRadius enlazado a una base de datos LDAP y bajo estándar IEEE 802.1X; permiten el acceso y validación de usuarios. Este tipo de autenticación, se basa en un servidor de autenticación con soporte RADIUS y protocolos EAP-802.1X, que depende de que el usuario y el AP acepten la políticas de seguridad y autenticación mediante él envió y uso de mensajes según el método EAP seleccionado lo que deriva en un mayor intercambio de tramas (Brincat et al., 2019) y tiempos de retardo o desempeño (Jiang et al., 2019), sin tener en cuenta parámetros como; autenticación anónima, acceso seguro, ataques de tipo hombre en el medio o accesos maliciosos (Niu et al., 2018; Sanda & Inaba, 2016); es así, que este enfoque centralizado presenta inconvenientes como; la existencia de un único punto de fallo o que la validación de credenciales de usuario se realice por una entidad única.

Después del análisis y la comprensión de los parámetros necesarios usados para el acceso a la infraestructura de red inalámbrica, las características y debilidades del método de autenticación utilizado; se propone investigar un mecanismo opcional de acceso inalámbrico basado en la tecnología de Cadena de Bloques, la cual es la base para las criptomonedas (Nakamoto, 2008), considerando una arquitectura donde los usuarios utilicen una aplicación Ethereum Wallet o bien un interfaz de usuario, además del uso y elección de los protocolos ERC20, ERC223, ERC721 o JSON-RPC (Balmaseda Aranda, 2018; K. Carrión, 2018; IETF, 2006); la implementación de una Cadena de Bloques Privada que usará plataformas Ethereum, Go-Ethereum, Geth, QuorumChain, CASPER o Ganache con Hardhat o Truffle Framework (Balmaseda Aranda, 2018; FRUTOS, 2019) y el desarrollo del contrato inteligente bajo el lenguaje de programación Solidity y su respectivo Command Line Interface (CLI); así los usuarios deberán acceder en su dispositivo la

aplicación o interfaz de acceso, ingresar sus credenciales, el entorno de cadena de bloques validará el registro de los usuarios y el cumplimiento del contrato inteligente; como parte de la investigación para el mecanismo de autenticación y acceso inalámbrico propuesto.

De esta manera se plantea la investigación de un método alternativo de autenticación, tomando en cuenta los factores negativos presentes en el método de autenticación usado, como; la existencia de un solo ente de validación, limitantes dependientes de la configuración de un segundo equipo, dirección IP, dominio, puerto de red o implicaciones de carácter técnico como seguridad, intercambio de tramas, tiempos de retardo y sus efectos directos en el desempeño del mecanismo de acceso inalámbrico en la Facultad de Ingeniería en Ciencias Aplicadas. Los cuales se pretenden solucionar, adaptando un conjunto de técnicas que usa la tecnología de fondo de las criptomonedas; específicamente los métodos empleados para compartir y validar transacciones mediante un contrato inteligente, encadenando bloques de información; donde cada bloque depende del bloque anterior, y cada nodo conoce dicha cadena de bloques, para así demostrar la existencia de una alternativa adaptable para la autenticación inalámbrica a más de que esta puede ser una plataforma anónima, segura, confiable y distribuida (Niu et al., 2018; Sanda & Inaba, 2016) que no depende de una tercera parte de confianza, y que supone un mejor desempeño.

1.3. Objetivos

1.3.1. Objetivo General

Adaptar un mecanismo de autenticación inalámbrica utilizando Cadena de Bloques en un entorno inalámbrico de pruebas en la Facultad de Ingeniería en Ciencias Aplicadas

1.3.2. Objetivos Específicos

Analizar la situación actual del método de autenticación inalámbrico usado actualmente en la Facultad de Ingeniería en Ciencias Aplicadas.

Definir el funcionamiento de la tecnología de Cadena de Bloques como método de autenticación.

Establecer las herramientas, algoritmos y protocolos para la adaptación del nuevo mecanismo de autenticación para redes inalámbricas.

Establecer un entorno de pruebas que permita simular la red de conectividad inalámbrica de la Facultad de Ingeniería en Ciencias Aplicadas, para el mecanismo de autenticación propuesto.

1.4. Alcance

Este proyecto plantea, investigar y adaptar la tecnología de la Cadena de Bloques en un método alternativo de autenticación y acceso a redes inalámbricas, donde el proceso de desarrollo técnico se establece en cuatro etapas que se detallan a continuación.

En la primera etapa se investigará y se buscará referencias bibliográficas para comprender las características generales de la tecnología de Cadena de Bloques como; su Definición, Arquitectura, Elementos, Tipología y Tratamiento de Datos; para luego abordar las particularidades más específica para su desarrollo como; la creación de Nodos, Mineros, generación de bloques, Bloque Genesis, funcionamiento del Algoritmo de Consenso, Algoritmos Criptográficos, Tokens, Contratos Inteligentes, Transacciones y Minería; temas en los que se

deberá profundizar y explorar para su aplicabilidad y uso en la autenticación inalámbrica, como punto de partida.

La segunda etapa, implica levantar información sobre la situación actual concerniente al método de autenticación inalámbrico implementado por la Dirección de Desarrollo Tecnológico e Informático de la Universidad Técnica del Norte, así como la administración de red de la Facultad de Ingeniería en Ciencias Aplicadas, de manera más precisa lo concerniente a la Infraestructura, Configuración, Protocolos, Equipos, Servidores, Redundancia, SSID propagado y mecanismos de Autenticación y Acceso inalámbrico que se usan.

En la siguiente etapa, se seleccionará el software, los protocolos, algoritmos, herramientas y los complementos, además se tomará en cuenta las especificaciones de hardware que se requieran; se integrará la tecnología de Cadena de Bloques al mecanismo de acceso inalámbrico, con base en la gestión de acceso de usuarios y validación de credenciales; se consideraran el número de nodos, se elijarán librerías, se determinara el desarrollo y/o uso de interfaz de cliente externo, el uso de procedimientos remotos y llamada de procesos para él envió de datos hacia la Cadena de Bloques; dentro de la adaptación y desarrollo del mecanismo de acceso.

La última etapa consiste en establecer un entorno de pruebas o ambiente de simulación de la red inalámbrica de la Facultad de Ingeniería en Ciencias Aplicadas, para realizar pruebas concepto o de funcionamiento del mecanismo de autenticación inalámbrico usando Cadena de Bloques, que permita acceder al recurso de red.

1.5. Justificación

Actualmente la Facultad de Ingeniería en Ciencia Aplicadas cuenta con diferentes SSID y configuraciones de autenticación de acceso inalámbrico, por un lado está la encriptación WPA2 PSK, donde solo se configura y se solicita una clave para acceder a la red y por otra parte está el caso en donde, el servidor FreeRadius es el único que autentica y autoriza el acceso inalámbrico de los usuarios, de modo que el acceso a las redes inalámbricas, en la Faculta de Ingeniería en Ciencias Aplicadas puede verse vulnerado, ya que el acceso a ciertas redes depende únicamente de la clave WPA2 PSK configurada, por lo que no se puede controlar el acceso e identidad, así también, en un entorno donde se depende de un servidor RADIUS centralizado que no cuenta un servidor de respaldo; el servicio de acceso a la redes inalámbricas se vería afectado; otro aspecto a señalar es que se depende de la configuración y la sincronización con la controladora MikroTik donde se conectan los APs de la facultad, ya que se debe asignar una dirección IP, dominio de red o número de puerto del servidor RADIUS; donde para que el servicio funcione correctamente se, deben direccionar las peticiones de validación hacia el servidor RADIUS, que accede a los registros de la LDAP para validar la conexión; por otra parte cabe mencionar parámetros relacionados con el desempeño, rendimiento, intercambio de tramas en el acceso o el hecho de que no exista otro método alternativo de autenticación.

En el panorama actual, la tecnología detrás del Bitcoin y otras criptomonedas ha llamado la atención de muchos investigadores y desarrolladores, quienes después de comprender la tecnología de Cadena de Bloques, han buscado llevar y explotar sus capacidades en diferentes sectores de la industria como; la salud, el transporte, las finanzas, la ciberseguridad, el Internet de las Cosas y las Telecomunicaciones; y es en estos últimos sectores mencionados donde, se han elaborado una cantidad considerable de artículos científicos y publicaciones tecnológicas, para

aprovechar y usar la tecnología; en el acceso a redes de sensores (Balmaseda Aranda, 2018), acceso seguro, confiable y anónimo (Niu et al., 2018; Wei et al., 2019) o autenticación y acceso inalámbrico (Wei et al., 2019); donde aparte de poner aprueba el alcance de la capacidades de la tecnología de Cadena de Bloques (Jiang et al., 2019), también se analiza el rendimiento y desempeño que involucra la implementación de esta tecnología (Sanda & Inaba, 2016).

De esta manera en el presente proyecto se propone el uso de la tecnología emergente de Cadena de Bloques, como solución a necesidades de seguridad y ciberseguridad, mediante el uso de algoritmos de consenso, sistemas descentralizados e infraestructuras de clave pública, en el sector de las telecomunicaciones; específicamente eliminando la necesaria presencia de una entidad intermediaria y destacando la capacidad de validación y verificación de información publicada en la cadena, que se basa en el consenso de los nodos involucrados; usando estas características y siguiendo la línea de investigación propuesta por la Facultad de Ingeniería en Ciencias Aplicadas "Innovación y Transferencia Tecnológica"; este proyecto generara información que permitirá analizar la Tecnología de Cadena de Bloques en virtud de sus ventajas y vulnerabilidades, en el diseño y desarrollo de un mecanismo de autenticación inalámbrico.

Finalmente, hay que mencionar, que, al ser uno de los primeros estudios sobre esta temática en la carrera, este trabajo pretende ser un acercamiento teórico – práctico al uso de la tecnología de Cadena de Bloques por lo que, en su defecto, se hará uso de la plataforma Blockchain privada de pruebas de Ethereum o se procurará generar una plataforma Blockchain privada para crear entornos de estudio; asimismo la implementación del mecanismo de autenticación esta fuera del alcance de este proyecto, quedando está parte a criterio de los administradores de red.

2. Capítulo II: Marco Teórico

El presente capítulo detalla la fundamentación teórica base para el desarrollo de esta investigación, donde se describen conceptos, características y términos relacionados con la Tecnología de la Cadena de Bloques, su arquitectura, funcionamiento, vulnerabilidades, usos y aplicaciones; también sobre temas relacionados con Criptografía y Tecnologías de Acceso a Redes Inalámbricas.

2.1. Términos Relacionados

Este apartado describe los términos básicos; usados de manera recurrente durante el desarrollo de esta investigación, los cuales están presentes en el estudio de los campos de ciberseguridad, redes inalámbricas y cadena de bloques; y que se conceptualizan brevemente a continuación.

- **Amenaza:** Se define como una entidad o circunstancia desfavorable que atente contra el buen funcionamiento de un sistema informático o red; y tiene consecuencias negativas provocando indisponibilidad o pérdidas; puede ser de causa fortuita o intencional aprovechando vulnerabilidades o debilidades existentes (Cuzme, 2017; Instituto Nacional de Ciberseguridad, 2017).

- **Vulnerabilidad:** Son las fallas y defectos de seguridad que se traducen en debilidades y que pueden ser explotadas para repercutir en el correcto funcionamiento de un sistema informático o de red, estos agujeros de seguridad son aprovechados por atacantes para acceder a los sistemas y redes con fines maliciosos (Cuzme, 2017; Instituto Nacional de Ciberseguridad, 2017; Panda Security, n.d.).

- **Riesgo:** Es la probabilidad de que ocurra un incidente o se materialice una amenaza, se plantea respecto a la existencia de vulnerabilidades frente a amenazas que producen daño o un impacto desfavorable (INCIBE, 2017).

- **Autenticación:** Una característica primordial en una comunicación segura, y es el proceso de comprobación; donde alguien es, quien dice ser; al acceder a un equipo o servicio (Instituto Nacional de Ciberseguridad, 2017).

- **Triangulo CIA:** También conocido como triangulo de seguridad informática o CIA por sus siglas en inglés, abarca los tres pilares fundamentales que se encargan de brindar protección a la información sensible. Estos son Confidencialidad, Integridad y Disponibilidad (Flores, 2021).

- **Confidencialidad:** Garantiza que el acceso a la información sea solo para aquellos autorizados a su acceso, siendo uno de los fundamentos de la seguridad de la información (Cuzme, 2017; Interdominio, 2020).

- **Integridad:** Es el fundamento que avala y verifica la exactitud de la información, así como también que esta no haya sido alterada, perdida o destruida accidental o intencionalmente por terceros (Cuzme, 2017; Instituto Nacional de Ciberseguridad, 2017).

- **Disponibilidad:** Es la capacidad de accesibilidad y uso de información cuando los usuarios autorizados lo requieran, junto con la integridad y la confidencialidad conforman los fundamentos de la seguridad de la información (Cuzme, 2017; Instituto Nacional de Ciberseguridad, 2017).

- **Encriptación:** Es una técnica usada para ocultar la información y en donde solo los individuos autorizados que disponen de una clave o código puedan descifrar y visualizar la información; permitiendo que los procesos de intercambio de datos sean más seguros (BBVA, 2018; Munro, 2020).

- **Criptografía:** Se denomina el conjunto de técnicas de cifrado o codificación de información, que se crean a partir de algoritmos matemáticos complejos; a fin de que dicha información sea ininteligible a terceros no autorizados (CRIPTONOTICIAS, 2019; Lizarraga et al., 2018).

- **Cifrado Simétrico:** Es un mecanismo de cifrado en donde se usa la misma clave para cifrar y descifrar y las dos partes que interviene en el proceso deben acordar de antemano dicha clave (Torres Cardona, 2021).

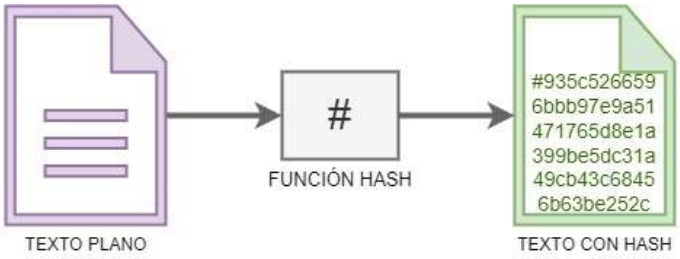
- **Cifrado Asimétrico:** Es un método de cifrado que usa dos claves diferentes una pública y una privada, las cuales están vinculadas matemáticamente entre sí; la clave pública se comparte, pero la privada debe mantenerse oculta; ambas pueden usarse para cifrar información y la clave opuesta a la que se use, se empleará luego para decodificarla (Torres Cardona, 2021).

- **Algoritmo:** Se define como algoritmo a, un conjunto ordenado de pasos para solucionar un problema; en criptografía y en las criptomonedas los algoritmos de cifrado son operaciones o funciones matemáticas usadas en combinación con claves para garantizar la confidencialidad e integridad de la información o de las transacciones (CRIPTONOTICIAS, 2019; Instituto Nacional de Ciberseguridad, 2017; Panda Security, n.d.).

- **Funciones Hash:** Una función hash criptográfica, es un algoritmo matemático, que con una entrada X, obtiene una salida Y como se observa en la Figura 1 (Martínez Bohórquez, 2017). Otros autores también definen que es una clave o huella digital única, irrepetible y no que se puede modificar; además identifica un conjunto de datos cifrados a los que se les aplica una función matemática aleatoria compleja, que da como resultado una cadena alfanumérica única de longitud fija denominada hash, que será diferente si se modifica un solo bit del paquete de datos original, dando como resultado un hash distinto (BBVA, 2018; CRIPTONOTICIAS, 2019; Lizarraga et al., 2018).

FIGURA 1

FUNCIÓN HASH



Tomado y adaptado de Simulación de una moneda virtual con Blockchain (p.11) por (Serna, 2017).

2.2. Redes Inalámbricas

Para contextualizar mejor el campo de investigación en relación al tema planteado, es necesario mencionar en esta sección los aspectos básicos pertinentes a una red inalámbrica; la cual permite la conexión de nodos sin necesidad de un medio físico, usando ondas electromagnéticas para el envío y recepción de información entre dispositivos (Gamba & Valencia, 2021); y que usa el aire como medio de transmisión y la radiación electromagnética, la cual es susceptible a interferencias por lo que su desarrollo específico de redes cubren una longitud de área establecida, ya sea para el uso empresarial o doméstico. Las redes inalámbricas pueden clasificarse según dos

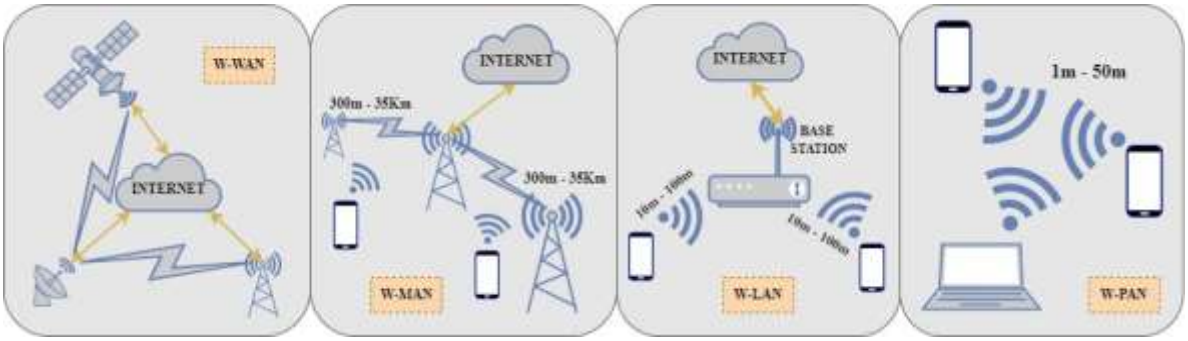
criterios; por su zona o área de cobertura y por su infraestructura, arquitectura o modelo adaptado (Yépez Lapo, 2021).

2.2.1. Redes Inalámbricas por Área de Cobertura

Dentro de la clasificación de las redes inalámbricas se hace referencia al dimensionamiento, zona o área que estas pueden cubrir y el alcance que tienen. Los grupos específicos según el área de cobertura son WPAN, WLAN, WMAN y WWAN, los cuales se ilustran en la Figura 2, donde se representa sus respectivos diagramas y se explican brevemente a continuación.

FIGURA 2

DIAGRAMAS DE REDES INALÁMBRICAS POR ÁREA DE COBERTURA.



Tomado y adaptado de (Salazar, 2016; Yépez Lapo, 2021).

Las **redes inalámbricas de área personal** o **WPAN** *(Wireless Personal Area Network)*, son redes con un corto alcance que cubren una distancia de diez metros; usadas para conectar y comunicar dispositivos personales o muy cercanos, generalmente no involucran el uso de una infraestructura para la conexión; siendo una solución eficiente y de bajo costo (Salazar, 2016). Las tecnologías que permiten desplegar este tipo de redes son; Bluetooth (IEEE 802.15.1), UWB (IEEE 802.15.3), ZigBee (IEEE 802.15.4), HomeRF y enlaces infrarrojos con tecnología IrDA (Yépez Lapo, 2021).

Por otra parte las **redes inalámbricas de acceso local** o **WLAN** *(Wireless Local Area Network)*, se diseñan para proporcionar acceso de dispositivos inalámbricos en redes domésticas, educativas o empresariales con áreas de cobertura de hasta cien metros, su implementación y uso frecuente se debe a que permiten la movilidad de los usuarios dentro del área de cobertura, sin desconectarse de la red (Collaguazo, 2017; Salazar, 2016) estas redes se basan en tecnologías IEEE 802.11 más conocido como Wi-Fi y otras no tan comerciales como HIPERLAN 1 y 2 (Yépez Lapo, 2021).

Así mismo, las redes conocidas como bucles locales de radio (BLR), **redes inalámbricas de área metropolitana** o **WMAN** *(Wireless Metropolitan Area Network)*, se basan en la tecnología IEEE 802.16 (Yépez Lapo, 2021) que hace referencia al estándar WiMAX (*Worldwide Interoperability for Microwave Access*) y tienen un alcance en el orden de las decenas de kilómetros; esta tecnología maneja una arquitectura de comunicación punto a multipunto y su objetivo es proporcionar alta velocidad en la transmisión de datos en redes inalámbricas de área metropolitana e interconectar redes locales inalámbricas o WLANs por WiMAX en una gran WMAN, lo que permite conectar ciudades sin la necesidad de recurrir a un medio físico o uso de cableado (Salazar, 2016).

Finalmente las **redes inalámbricas de área ampliada** o **WWAN** *(Wireless Wide Area Network)*, tienen un alcance que supera los cincuenta kilómetros, hacen uso de frecuencias licenciadas o reguladas y generalmente su cobertura abarca extensas áreas como ciudades o países; se conectan usando sistemas de comunicación satelital y antenas en radio bases y estaciones base de proveedores de servicios de Internet y telefonía celular (Salazar, 2016) y para interconectarse y transmitir información con otros puntos a lo largo del planeta; las principales tecnologías que se

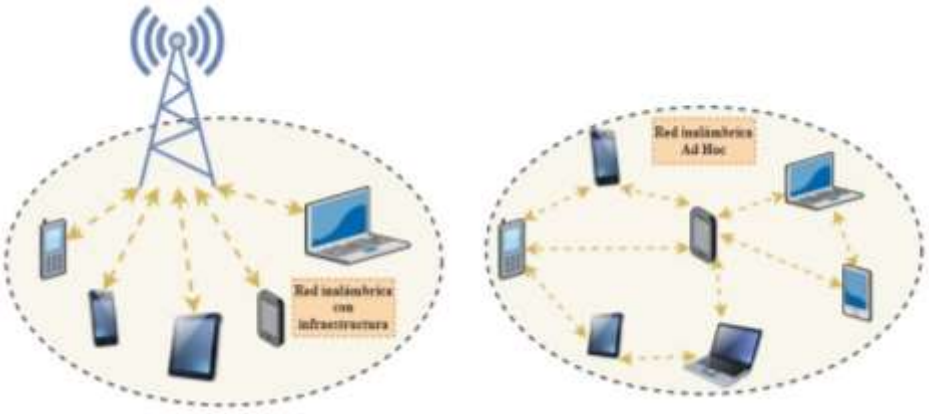
usan son; 2G-GSM, 2.5G-GPRS, 3G-UMTS, LTE, 4G, 5G, (Yépez Lapo, 2021) próximamente 6G y por su puesto las de comunicación satelital.

2.2.2. Redes Inalámbricas por Arquitectura

Las redes inalámbricas también catalogadas como redes móviles deben permitir que sus estaciones o usuarios accedan a la información, sin que sea relevante su ubicación geográfica dentro del área de cobertura (Yépez Lapo, 2021); por tal razón existen dos modos de configuración respecto a la arquitectura que se muestran en la Figura 3 y se usa en el despliegue de una red inalámbrica, siendo ad hoc e infraestructura (Salazar, 2016).

FIGURA 3

DIAGRAMAS DE REDES INALÁMBRICAS POR ARQUITECTURA.



Tomado y adaptado de (Yépez Lapo, 2021).

- **Redes sin Infraestructura o Ad Hoc**

En este tipo de redes, fundamentalmente no se hace uso de una infraestructura de red preestablecida, así como también no se requiere de una administración centralizada de red; ya que los propios hosts y nodos móviles conforman una infraestructura de red sin limitaciones de tamaño y puede conformarse por cientos o miles de hosts móviles, estableciendo una topología de

conexión entre nodos cambiante debido a la movilidad de los mismos (Yépez Lapo, 2021), esto se representa en la Figura 3 en la sección referente a Redes Ad Hoc.

Por ende conceptualmente, las redes Ad Hoc deben ser auto configurables y de fácil despliegue, ofreciendo conectividad emergente entre dos o más hosts/nodos inalámbricos; y que para acceder al enlace y compartir información deben conectarse; automáticamente, periódicamente, bajo demanda o solicitud; así mismo los hosts pueden salir de la red dinámicamente, sin notificar a otros nodos y en lo posible sin interrumpir la comunicación de la red; su uso y aplicación es frecuente en los ámbitos: militar, industrial, académico, el sector de la salud y en emergencias de búsqueda y rescate (Collaguazo, 2017; Salazar, 2016; Yépez Lapo, 2021).

Así pues, es denominado como **Conjunto Independiente de Servicios Básicos** o **IBSS** (*Independient Basic Service Set)*, y hace referencia a redes sin infraestructura que conectan grupos pequeños de dispositivos inalámbricos entre sí; los nodos se comunican de igual a igual y con una comunicación punto a punto, lo que limita que la arquitectura no admita una pasarela de conexión hacia redes cableadas o a Internet (Romo, 2022; Salazar, 2016; Yépez Lapo, 2021).

- **Redes con Infraestructura**

En el modo infraestructura se usan dispositivos de control de tráfico denominados; estaciones base (*Base Station*, BS) o puntos de acceso (*Access Point*, AP); estos dispositivos actúan como pasarela (*Gateway*) permitiendo el tráfico entre: la red cableada, la red inalámbrica y los dispositivos móviles; la conexión de los nodos inalámbricos con el punto de acceso se realiza uno a la vez, mediante solicitudes de transmisión y envió de información para conectarse con dispositivos en la infraestructura cableada o con otros nodos inalámbricos como se muestra en la

Figura 4; por otra parte el despliegue de una red modo infraestructura representa un gasto adicional por la instalación de puntos de acceso, pero su implementación brinda seguridad, gestión, control, escalabilidad y estabilidad (Collaguazo, 2017; Salazar, 2016; Yépez Lapo, 2021).

FIGURA 4

MODELO DE ARQUITECTURA DE RED INALÁMBRICA MODO INFRAESTRUCTURA

Nota: Representación de red inalámbrica modo infraestructura; donde se aprecia también diagramas para BSS y ESS. Tomado y adaptado de (Collaguazo, 2017; Romo, 2022; Yépez Lapo, 2021).

Dentro de la configuración de redes inalámbricas con infraestructura podemos diferenciar dos modos; el primero denominado **Conjunto de Servicios Básicos** o **BSS** *(Basic Service Set)*, que está compuesto por un punto de acceso y los dispositivos móviles, el AP coordina la comunicación de los dispositivos de red y también realiza funciones de almacenamiento temporal y de pasarela hacia otras redes, ver Figura 4 (Romo, 2022; Salazar, 2016).

Por otra parte el segundo modo de configuración es el **Conjunto de Servicios Extendidos** o **ESS** *(Extended Service Set)*, el cual conecta varios puntos de acceso o BSS permitiendo a los

dispositivos inalámbricos moverse libremente, en un rango de cobertura mayor y de un punto de acceso a otro sin desconectarse de la red ESS, como se aprecia en la Figura 4; además toda la comunicación pasa por el punto de acceso, lo que significa que los dispositivos móviles no pueden comunicarse entre ellos, a menos que sea mediante el punto de acceso al que estén asociados (Romo, 2022; Salazar, 2016; Yépez Lapo, 2021).

2.1.1. Cadena de Bloques

Como lo conceptualiza Preukschat: "*Una Cadena de Bloques no es otra cosa que una base de datos distribuida entre diferentes participantes, protegida criptográficamente y organizada en bloques de transacciones relacionados entre sí matemáticamente"* (Preukschat et al., 2017); de esta manera se entiende que la base para una Cadena de Bloques es el conceso de la información entre todos los participantes.

Otra definición similar expresa que: "*Una Cadena de Bloques, o Blockchain, también conocida como libro de contabilidad distribuido (distributed ledger), es una base de datos distribuida que registra bloques de información y los entrelaza para facilitar la recuperación de la información y la verificación de que esta no ha sido cambiada"* (Acuña, 2017).

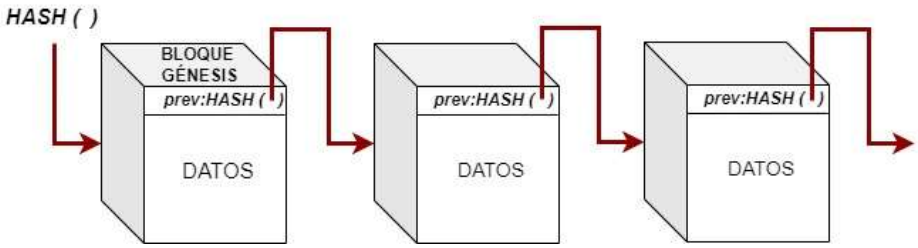
Una tercera opinión formada de las muchas existentes, elaboradas y publicadas por investigadores en libros y artículos científicos manifiesta que: "*Blockchain puede definirse como un libro digital compartido que abarca una lista de bloques conectados y almacenados en una red distribuida, descentralizada y protegida mediante criptografía, sirviendo como un depósito de información irreversible e incorruptible*" (Beck & Müller-bloch, 2017) (Pacheco Jiménez, 2019).

De estos conceptos se resume que una Cadena de Bloques o Blockchain es un concepto que surge y se relaciona a las criptomonedas, más estrechamente al Bitcoin; como lo describe en

su propuesta Satoshi Nakamoto, donde detalla la funcionalidad independiente de esta tecnología, para realizar transacciones entre pares y se ubica como pilar de las criptomonedas, definiéndose básicamente como un gran registro consensuado, seguro, descentralizado e infalsificable (Casas et al., 2019); donde todos los nodos validan la legitimidad de la información en las transacciones; y los bloques de información se enlazan mediante un "hash" que conecta el bloque actual con el anterior y así sucesivamente hasta llegar al bloque génesis (Acuña, 2017). En la Figura 5 se representa como se enlazan los bloques en una cadena de bloques.

FIGURA 5

ENLACE DE BLOQUES DE INFORMACIÓN MEDIANTE HASH EN UNA CADENA DE BLOQUES



Nota: La imagen muestra como los bloques se conectan y dependen del hash del bloque anterior. Tomado y adaptado de Estudio sobre Bitcoin y Tecnología Blockchain. *Cuadernos Cef*, *1*(November), 1–44, por (Acuña, 2017).

2.2. WIFI

Acrónimo para la tecnología *Wireless Fidelity*, la cual define el conjunto de normas para redes inalámbricas de área local o WLAN, es una tecnología de conexión inalámbrica de bajo coste, usando como medio de transmisión ondas electromagnéticas bajo el estándar IEEE 802.11 (Andaluz, 2021; Monzon & Angulo, 2020). Las principales ventajas de esta tecnología son: la movilidad de los dispositivos, los cuales pueden ubicarse en cualquier lugar dentro del área de cobertura sin depender de un medio físico de conexión; la flexibilidad que permite, mantener la configuración de red, aun después de instalar nuevos dispositivos y la escalabilidad que facilita

extender la red, después de haber sido implementada. Como desventaja las redes WiFi son susceptibles a interferencias generadas por atenuación, lluvia, niebla, dispersión y entre otras (Andaluz, 2021).

2.2.1. IEEE 802.11

Ratificado en 1997 por el IEEE o Instituto de Ingenieros Eléctricos y Electrónicos (*Institute of Electrical and Electronic Engineers)*, especifica un ancho de banda de 1 a 2 Mbps, trabajando a 2,4 GHz. Este estándar define el conjunto de reglas y protocolos que garantizan la interoperabilidad, el control de acceso al medio (MAC) y a la capa física (PHY); para el despliegue de WLANs y sus requerimientos de funcionamiento a nivel de capa física y de datos para soluciones con tecnología WiFi (Andaluz, 2021; Romo, 2022; Salazar, 2016). A continuación, se detalla brevemente la evolución y características principales de la familia del estándar IEEE 802.11 que también se resume en la Tabla 1; avances y rectificaciones realizadas a partir del primer estándar presentado en 1997 (Romo, 2022).

Después de la primera versión se denominaría WiFi al conjunto IEEE 802.11; siendo **IEEE 802.11b** ratificado en el año de 1999 a partir del IEEE 802.11 original, este estándar es usado actualmente y trabaja a 2,4 GHz con una tasa de transferencia de 11 Mbps; la siguiente versión **IEEE 802.11a** se definiría como WiFi5 e incorpora una transferencia de datos más rápida, pero no es compatible con otras versiones; trabaja a 5GHz y con una tasa de transferencia de 54 Mbps; mientras que **IEEE 802.11c** es una versión modificada del estándar 802.11d, para permitir la compatibilidad con dispositivos 802.11 a nivel capa de datos y a su vez **IEEE 802.11d** permite el uso internacional de las redes 802.11 locales; es decir que distintos dispositivos puedan intercambiar datos en rangos de frecuencia permitidos por el país de origen del dispositivo.

Por otra parte, la versión **IEEE 802.11e** incluye mejoras para audio y video, además define los requisitos de ancho de banda y retardo de transmisión; dedicado a mejorar la calidad de servicio sobre la capa enlace de datos. Así mismo **IEEE 802.11f** fue creado para garantizar la interoperabilidad de puntos de acceso en una red inalámbrica de área local multi proveedor y define el registro de los APs de acceso y el intercambio de información cuando un usuario se traslada de un AP a otro. En **IEEE 802.11g** las tasas de transferencia de datos iguales a las de IEEE 802.11a, es compatible con IEEE 802.11b; con un ancho de banda de 54Mbps y en el rango de frecuencia de 2,4 GHz. Para **IEEE 802.11h** se decidió resolver problemas de coexistencias de las redes 802.11 con sistemas de radares o satélites, con base en reglamentos europeos para WLANs a 5GHz; especificando el control de potencia de transmisión y frecuencia dinámica.

Además, con el fin de mejorar la seguridad en la transferencia de datos aparece **IEEE 802.11i**, determinando vulnerabilidades actuales en la seguridad para protocolos de autenticación y de codificación, basándose en el protocolo AES. Con **IEEE 802.11n** surge la nueva generación de WiFi, con tasas superiores a 100 Mbps, se basa en la tecnología MIMO y trabaja en las frecuencias de 2,4 y 5 GHz. Posteriormente **IEEE 802.11ac** brinda una velocidad de 1,3 Gbps, operando en la banda de 5GHz; presenta menos interferencias, usa modulación 256QAM y posee la tecnóloga MU-MIMO. Finalmente, **IEEE 802.11ax** opera en 2,4 y 5 GHz, introduce OFDMA mejorando la eficiencia espectral y soporte de modulación 1024-QAM.

TABLA 1

CARACTERÍSTICAS DE ESTÁNDARES IEEE DE LA FAMILIA 802.11

Estándar IEEE	Característica	Banda	Ancho de Canal	Velocidad	Publicación
802.11	Legacy	2.4 GHz	20 MHz	1 – 2 Mb/s	1997

802.11b	Más comercializado (Wi-Fi 1 o Wi-Fi B)	2.4 GHz	20 MHz	11 Mb/s	1999
802.11a	Banda de 5 GHz (Wi-Fi 2 o Wi-Fi A)	5 GHz	20 MHz	54 Mb/s	1999
802.11g	Revisión de 802.11b (Wi-Fi 3 o Wi-Fi G)	2.4 GHz	20 MHz	54 Mb/s	2003
802.11n	Introduce SU-MIMO	2.4-5 GHz	20, 40 MHz	> 600 Mb/s	2009
802.11ac wave 1	Revisión 802.11n. Solo SU-MIMO	5 GHz	20, 40, 80 MHz	>800 Mb/s	2014
802.11ac wave 2	Introduce MU-MIMO (Wi-Fi 5 o Wi-Fi AC)	5 GHz	20, 40, 80, 160 MHz	> 1.3 Gb/s	2016
802.11ax	Usa MU-MIMO e introduce OFDMA (Wi-Fi 6 o Wi-Fi AX)	2.4-5 GHz	20, 40, 80, 160 MHz	>2.4 Gb/s	2019

Nota: La tabla muestra un resumen de las características de los estándares IEEE 802.11 más utilizadas. Tomado y adaptado de (INTEL, 2021; Romo, 2022; Tito, 2022).

2.2.2. Mecanismos de Seguridad

Los mecanismos protegen y evitan intrusiones, haciendo segura a la tecnología WiFi, al momento de enviar y recibir datos sin que un atacante pueda capturar y modificar la información transmitida, haciendo necesario la protección de la comunicación entre los equipos de enrutamiento y los adaptadores inalámbricos (Sánchez, 2021).

Entre los principales mecanismos se tiene WEP, WAP, Filtrado MAC y otros; los mismo que se describen a continuación:

WEP (Wireless Equivalen Privacy), es el primer tipo de cifrado implementado, usa un algoritmo de cifrado RC4, en teoría cifra claves de 64 a 128 bits, pero en realidad son 40 a 104 valores y los bits restantes se usan en el vector de inicialización. Este tipo de seguridad se basa en una clave publica compartida que se usa para cifrar los datos, reduciendo el nivel de seguridad (Sánchez, 2021).

Por otra parte, **WAP** (WiFi Protected Access), surge para cubrir las vulnerabilidades del protocolo WEP, posee diferentes formas de autenticación que se desarrollaron de acuerdo con el uso que se le daría, siendo estas la versión para empresas y la de uso personal. La versión **WAP Enterprise** se basa en una clave de distribución y se diseñó para compañías ya que se requiere un servidor RADIUS que interactúa bajo protocolo EAP con el cliente y APs. En cuanto al mecanismo **WAP/WPA2 Personal** se basa en una clave pre-compartida PSK, que se diseñó específicamente para viviendas y redes pequeñas; al ser doméstico, el estándar no requiere de una autenticación de servidor. En el proceso cada dispositivo cifra el tráfico de la red y usa una PSK de 120 a 256 bits, sin embargo, la mejora WPA2 introduce un concepto de apretón de manos con "4-way handshake" que permite al AP y al cliente demostrarse mutuamente que ambos conocen el PSK (Sánchez, 2021).

Otro mecanismo es el **Filtrado MAC**, este mecanismo que permite el acceso a la red solo a aquellos dispositivos que hayan registrado su dirección física MAC en una tabla de acceso; este mecanismo se puede configurar a la par de otros mecanismos de seguridad; ya que mediante el clonando la dirección física, se puede suplantar el acceso; siendo esta la vulnerabilidad de este mecanismo (Romo, 2021).

Otra alternativa como mecanismo de precaución o protección adicional dentro de las redes inalámbricas WiFi, es la **Detención de Difusión de SSID**, involucra los puntos de acceso los cuales difunden periódicamente su SSID (*Service Set Identifier* o Identificador de Conjunto de Servicios); donde se puede configurar el AP para que no difunda las tramas *Beacon* que contienen información del SSID, el inconveniente de este mecanismo se presenta cuando los usuarios no puedan conectarse debido a que el SSID no está visible (Romo, 2021).

Por otro lado, **IEEE 802.1x**, es el estándar de control de acceso, permite usar diferentes tipos y mecanismo de autenticación y autorización basado en la arquitectura cliente - servidor que restringe la conexión de equipos no autorizador a una red, el protocolo involucra tres participantes; el suplicante, el autenticador y el servidor de autenticación y autorización (Interdominio, 2020; Monzon & Angulo, 2020; Villanueva, 2020).

Finalmente**, EAP** (*Extensible Authentication Protocol*), se basa en el uso de un controlador de acceso denominado autenticador, el usuario o solicitante y un servidor de autenticación (Lopez, 2021) existen variantes del protocolo EAP según la modalidad de autenticación que empleen (Villanueva, 2020) las cuales se explican a continuación.

Dentro de las variantes EAP que emplean certificados de seguridad encontramos; **EAP-TLS** donde se requiere la instalación y configuración de certificados de seguridad en los clientes y en el servidor; proporciona una autenticación mutua sólida, ya que el servidor autentica al cliente y viceversa, soporta el uso de claves dinámicas WEP. La sesión de autenticación entre el cliente y el autenticador se cifra empleando el protocolo TLS (*Transparent Layer Substrate*). La solución **EAP-TTLS** desarrollada por Funk Software y Certicom, es similar a EAP-TLS, con la diferencia de que requiere la instalación de un certificado en el servidor; garantizando una autenticación fuerte del servidor; mientras que, por parte del cliente la autenticación se efectúa una vez que se establece la sesión TLS, usando otro método como PAP, CHAP, MS-CHAP o MS-CHAP v2. Otra modalidad desarrollada por Microsoft, CISCO y RSA Security es **PEAP**, parecida a EAP-TTLS, requiere de certificado de seguridad en el servidor y provee protección a métodos más antiguos de EAP, establece un túnel seguro TLS entre el cliente y el autenticador (Villanueva, 2020).

Por otra parte, las variantes EAP que usan contraseñas son; **EAP-MD5** la cual utiliza un nombre de usuario y contraseña cifrada con el algoritmo MD5 para la autenticación. Hay que destacar que es susceptible a ataques de diccionario, además el cliente no tiene manera de autenticar al servidor y comprobar que se está conectando a la red adecuada, adicional el esquema no es capaz de generar claves WEP dinámicas, debido a esto EAP-MD5 no se usa muy comúnmente. El Protocolo de Autenticación Extensible Ligero o **LEAP**, es una variante propiedad de Cisco, con un esquema de nombre de usuario y contraseña, soporta calves dinámicas WEP y al ser una tecnología propietaria exige que todos los puntos de acceso sean Cisco y que el servidor sea compatible con LEAP (Villanueva, 2020).

2.3. Tecnología de Cadena de Bloques

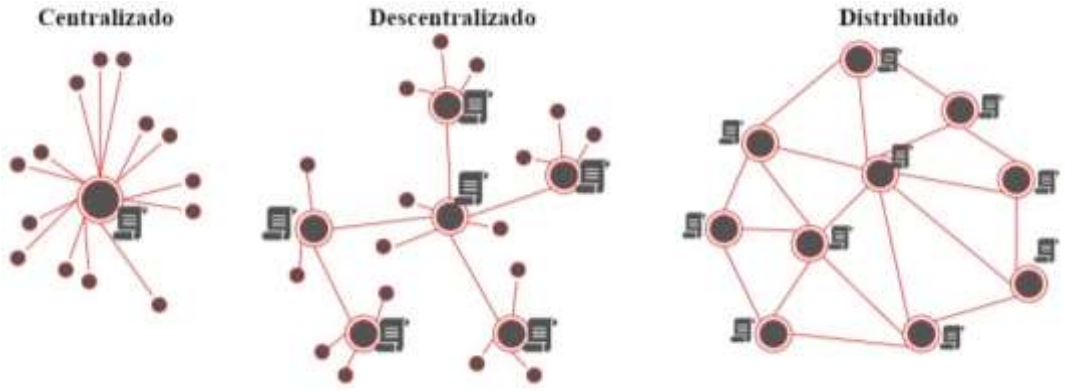
En el año 2008 Satoshi Nakamoto publicó una propuesta innovadora que permitía realizar transacciones electrónicas, este concepto sentarían las bases para las criptomonedas actuales; el documento titulado “Bitcoin: A Peer-to-Peer Electronic Cash System”, describe como se realiza el pago directo entre pares, sin pasar por un tercera parte de confianza, ya sea esta una entidad bancaria o gubernamental; done las transacciones se validan en consenso por un conjunto de nodos y se registran en una base de datos pública, transparente y descentralizada formando una cadena de paquetes o bloques que dependen enteramente del bloque generado anteriormente (Nakamoto, 2008).

A continuación, se describen brevemente conceptos y definiciones previas que ayudaran a comprender, familiarizar y relacionar los fundamentos referentes a la Tecnología de Cadena de Bloques.

- **DLT :** Tecnología de Libro Mayor Distribuido o *Distrubuted Ledger Technology*, también conocida como tecnología de registro distribuido, libro mayor distribuido o base de datos distribuida; almacena datos en una red de nodos descentralizados, los diferentes nodos pueden acceder y verificar cualquier entrada; así mismo cada nodo participante mantiene la base de datos como se observa en la Figura 6; tomando como estructura redes entre pares (peer-to-peer o P2P) y algoritmos de consenso, que también son fundamentales en las tecnologías detrás de cadena de bloques y las criptomonedas (Bitcobie, 2018; Lizarraga et al., 2018; Munro, 2020).

FIGURA 6

DIAGRAMAS DE REDES CON BASES DE DATOS



Nota: La imagen esquematiza los modelos de bases de datos respecto al acceso y distribución de información entre los nodos. Tomado y adaptado de Aplicación de Contratos Inteligentes en Ethereum (p. 5) por (Romero Solís, 2019).

- **Nodo:** Es un equipo que conforma la red de una cadena de bloques, se encarga de almacenar y distribuir copias actualizadas y en tiempo real de las operaciones y transacciones que se realizan, cuando se genera un nuevo bloque un nodo realiza la validación y retrasmisión de transacciones mientras recibe una copia de la cadena de bloques completa (BBVA, 2018; Tovar, 2018).

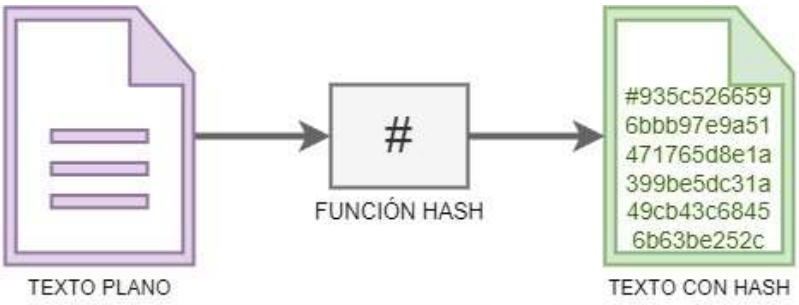
- **Red P2P:** Una red de pares o *peer-to-peer*, es una red que se crea entre distintos nodos conectados entre sí a través de una misma red mediante el mismo software o protocolo, que actúa como sistema de comunicación entre ellos, semejante al esquema que se aprecia en la Figura 6 y que referente a una red distribuida (Martín, 2021a).
- **Sistema Descentralizado:** La tecnología de cadena de bloques se basa en un sistema descentralizado, es decir, que toda la información es controlada por todos y cada uno de los usuarios, ordenadores o nodos de la red, sin que exista un ente centralizado que controle la información ver Figura 6 esquema de red descentralizada; sino que son los propios usuarios, los que validan y almacenan la información registrada en la red P2P (Martín, 2021a).

- **Token:** Es una unidad de valor o representación monetaria de una criptomoneda, es el código digital que define cada fracción; que puede ser poseído, comprado y vendido (Munro, 2020). Es la representación abstracta y digital de un valor, que debe corresponderse con un activo real, cuando se tokeniza un activo, este se representa digitalmente en la red mediante como un token, de esta manera es necesario que coexistan simultáneamente el activo real y el digital (Martín, 2021a).
- **Hash:** Conjunto concatenado de caracteres alfanuméricos, que son el resultado de aplicar un algoritmo matemático sobre un archivo u objeto digital, siendo el hash un identificador único para cada archivo u objeto al que se le aplica, por lo que tienen cumple esta función para cada uno de los bloques de datos. El hash es inmutable y unidireccional, lo que permite calcular el hash de un archivo mediante la aplicación de un algoritmo, pero no permite puede obtenerse el archivo digital a partir del hash. En caso de que se modifique el contenido del archivo, el hash asociado a este variará también como se representa y

ejemplifica en la Figura 7; además el hash tiene una triple función fundamental: la de identificación de los bloques, la de eliminar la posibilidad de alteración de la información contenida en la cadena de bloques y permitir el seguimiento de los datos (Martín, 2021a).

FIGURA 7

GENERACIÓN DE CODIFICACIÓN BAJO FUNCIÓN HASH.



ENTRADA	VALOR HASH
cadena	a27ea2eb3552252c5ce5d31aa2cf342ea1ffa0844365e0e97c32e86771eb8581
cadenas	f8bb34d8f837a36044feda47d8a2a562454f915ebcc8449b8733e347b2af4463

Nota: En la imagen se aprecia como se genera una secuencia alfanumérica al aplicar el algoritmo Hash SHA256 a las palabras "*cadena - cadenas*" de ejemplo las cuales solo varían en una letra, pero el algoritmo genera una cadena de caracteres totalmente al aplicar la función hash. Tomado y adaptado de Simulación de una moneda virtual con Blockchain (p.11) por (Serna, 2017).

- **Mineros:** Los nodos mineros son equipos de alta capacidad computacional, conectados permanentemente a la red, vigilando que las transacciones se realicen correctamente; además validando, creando y enlazando bloques al resolver desafíos o retos matemáticos, acción por la que reciben una recompensa o retribución (BBVA, 2018; Lizarraga et al., 2018).

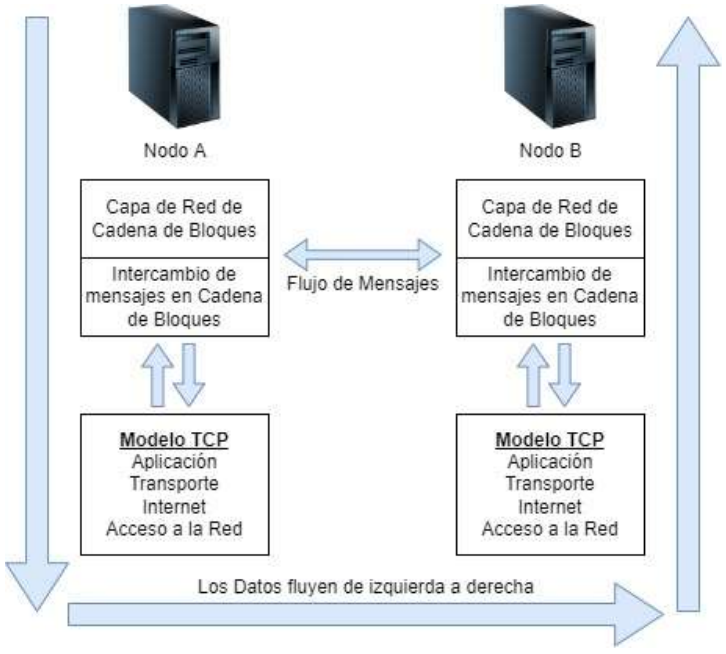
2.3.1. Arquitectura

La arquitectura de red de una cadena de bloques, es básicamente una red distribuida P2P (*peer to peer)* basado en la parte superior de la capa de red (ver Figura 8); las redes P2P hacen

referencia a un grupo de equipos actuando como un nodos para compartir información entre ellos; por lo tanto una cadena de bloques se ejecuta en una red distribuida de servidores, conocidos también como nodos, los cuales tienen como propósito, proveer y establecer el consenso de la cadena de bloques en cualquier momento, además de almacenar una copia de la cadena de bloques. La aplicación fundamental de la cadena de bloques es la de un registro o libro de transacciones, semejante a un registro público, donde se almacenan todas las transacciones que se realizan en la red, esto hace que sea un sistema muy seguro, transparente y descentralizado (Rupsha, 2017).

FIGURA 8

ARQUITECTURA DE RED DE UNA CADENA DE BLOQUES



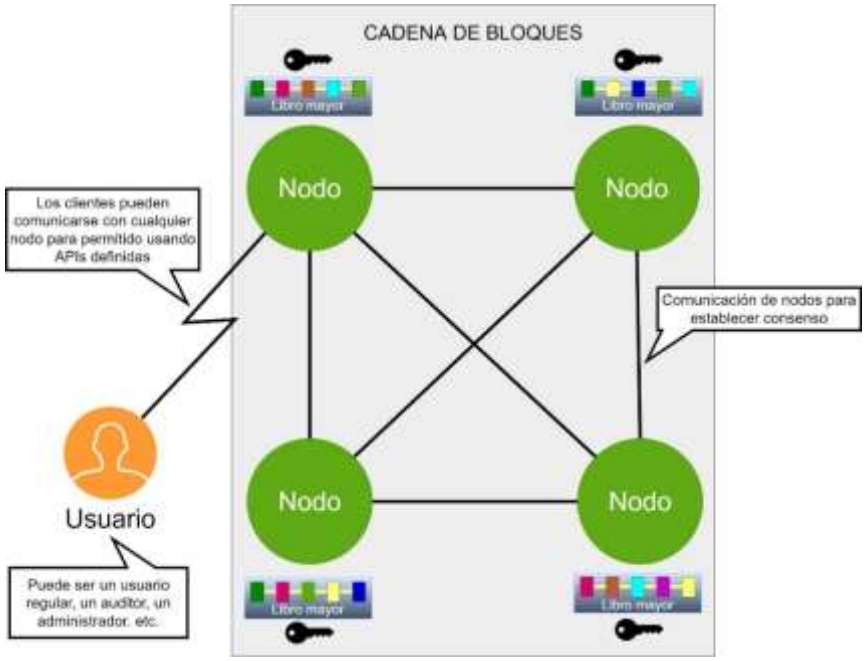
Nota: La imagen muestra un diagrama de flujo de datos entre los nodos de una cadena de bloques. Tomado y adaptado de *Using Blockchain Technology and Smart Contracts for Access Management in IoT devices* (p.14). por (Rupsha, 2017).

El artículo "Cloud Customer Architecture for Blockchain" se representa en alto nivel la estructura de una red de cadena de bloques, la cual puede observarse en la Figura 9; en la imagen es posible identificar cinco componentes principales de la arquitectura: los usuarios, los nodos, la

cadena de bloques (Libro mayor) y los mecanismos de seguridad (llave) y comunicación utilizados (enlaces) (Cloud Standards Customer Council, 2017).

FIGURA 9

VISTA DE ALTO NIVEL DE UNA RED DE CADENA DE BLOQUES

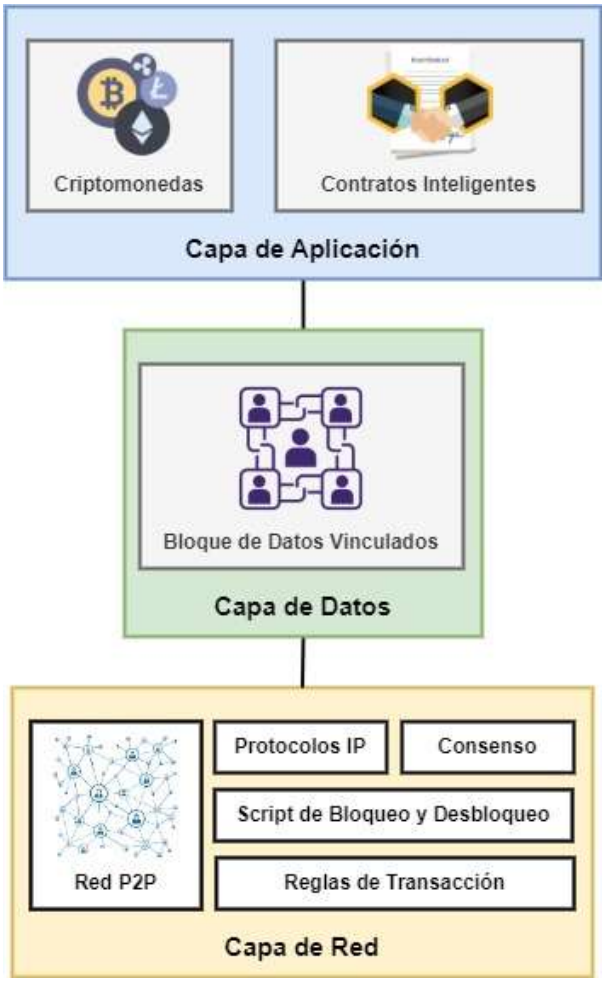


Nota: La imagen muestra un diagrama generalizado de los componentes de una cadena de bloques. Tomado y adaptado de *Cloud customer architecture for enterprise social collaboration* (p.2). por (Cloud Standards Customer Council, 2017).

Por otra parte en la investigación "Tecnologías Blockchain y sus aplicaciones" se menciona que en una Cadena de Bloques se pueden determinar tres capas, las cuales se observan en la Figura 10 y son: la capa de red que permite a la cadena de bloques conectarse e interactuar con los usuarios y el entorno además descentraliza el sistema mediante la red *peer to peer* y protocolos IP; la capa de datos define su estructura y la de los algoritmos, haciendo que la cadena de bloques se transparente y descentralizada; finalmente la capa de aplicación es la representación de la cadena de bloques visualizándola como las conocidas criptomonedas o los contratos inteligentes (Mela & Cedeño, 2019).

FIGURA 10

CAPAS EN UNA CADENA DE BLOQUES



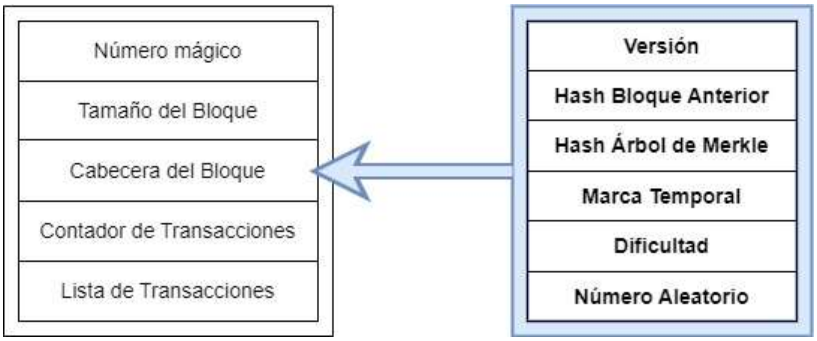
Nota: La imagen muestra la división de capas en una cadena de bloques. Tomado y adaptado de *Tecnologías Blockchain y sus aplicaciones* (p.7). por (Mela & Cedeño, 2019).

- **Bloque**

Es un paquete de datos con registros históricos de las transacciones, que son permanentes e incorruptibles a los que se puede acceder en cualquier momento y va estrictamente enlazado con su bloque predecesor, contiene además algunos campos adicionales que se observa en la Figura 11, necesarios para el funcionamiento y validación en una red de cadena de bloques (Bitcobie, 2018; Montoya, 2021; Munro, 2020).

FIGURA 11

ESTRUCTURA DE UN BLOQUE



Tomado y adaptado de *Sistema de autenticación basado en blockchain para la gestión de billetes en un entorno de transporte inteligente* (p.33). por (Montoya, 2021).

A continuación, se detallan los campos que componen un bloque, así como a su cabecera, los cuales se aprecia en la Figura 11:

Bloque:

- **Número mágico:** Cadena de caracteres alfanuméricos que identifican un formato.
- **Tamaño de Bloque:** Dimensión en bytes del bloque.
- **Contador de Transacciones:** Número de transacciones incluidas en el bloque.
- **Lista de Transacciones:** Inventario de las transacciones incluidas en el bloque.

Cabecera de Bloque:

- **Versión:** Versión del bloque.
- **Hash Bloque Anterior:** Referencia al bloque anterior para poder generar y organizar un registro ordenado.

- **Hash Árbol de Merkle:** Referencia a la raíz de la estructura del árbol que se produce del enlace de las transacciones del bloque.
- **Marca Temporal:** Registra el momento exacto en el que se crea el bloque.
- **Dificultad:** Requisito de la prueba de esfuerzo para que el bloque sea validado.
- **Número aleatorio:** Se emplea en la prueba de esfuerzo.

- **Descentralización y Consenso**

La **descentralización** es una característica de los sistemas que no dependen de un punto único centralizado para funcionar, favorece la independencia, limita la censura y el control; dentro del mundo de las criptomonedas implica que todos los nodos realizan la toma de decisiones en conjunto, independientes de una entidad centralizada, banco o gobierno (CRIPTONOTICIAS, 2019; Lizarraga et al., 2018; Tovar, 2018).

Como complemento al concepto descentralizado; el **consenso** es el proceso clave en la tecnología de cadena de bloques, ya que involucra a todos los nodos participantes de la red, los cuales acuerdan la validez de las transacciones, actualizan el registro y aseguran que las copias de este sean exactas entre sí (Bitcobie, 2018; Tovar, 2018).

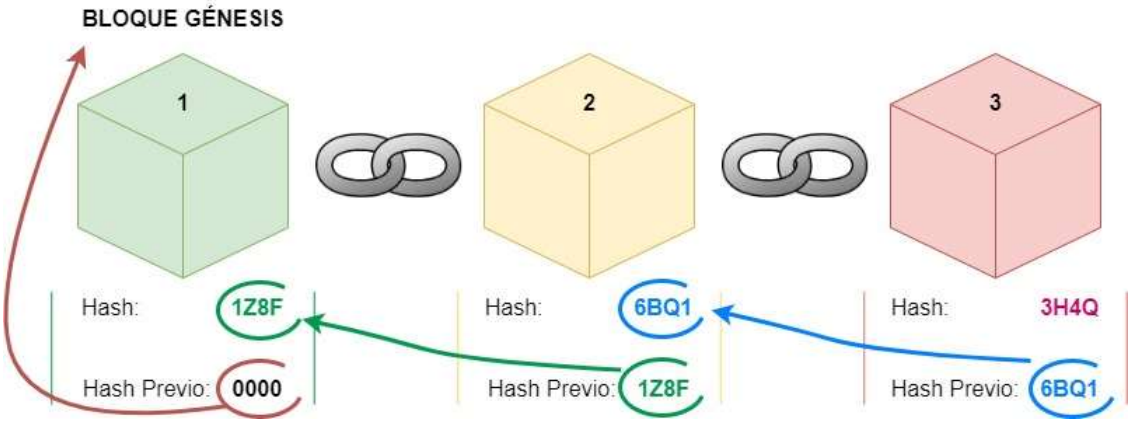
2.3.2. Funcionamiento de una Cadena de Bloques

Una cadena de bloques inicia con un bloque, conocido como bloque génesis, el cual no tiene predecesor, todos los nodos participantes dentro de la red contienen el bloque génesis; al generar un bloque nuevo este se verifica y se agrega. Los bloques se agregan a la cadena de bloques en una manera lineal enlazándose con el bloque anterior mediante el hash como se puede apreciar

en la Figura 12; el cual se genera con la información del bloque y el hash del bloque anterior (Montoya, 2021).

FIGURA 12

CADENA DE BLOQUES A PARTIR DEL BLOQUE GÉNESIS



Nota: La imagen muestra un diagrama de ejemplo para la generación de Hash a partir del Hash del bloque anterior.

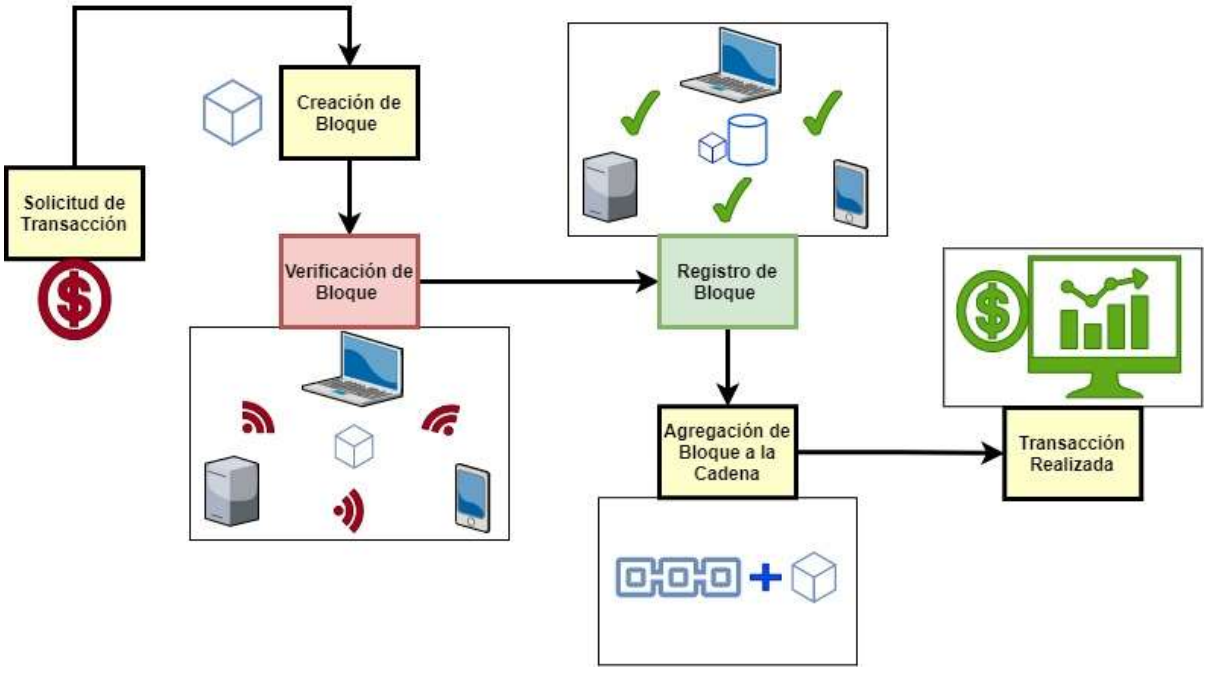
Un bloque contiene registros de los cambios de estado en una transacción en la cadena de bloques. Las transacciones se validan y almacenan en la cadena de bloques una vez que se verifican por todos los nodos de la red como se explica en la Figura 13. Cada bloque en la cadena contiene una lista de transacciones y un valor de hash. El valor de hash asignado está referido al hash del bloque previo. El valor de hash es usado para prevenir que los datos sean modificados. Además, los nodos pueden ejecutar transacciones usando una clave pública emparejada a una clave privada. Sin embargo, la clave pública es usada como una dirección única para identificar al propietario de la cuenta. La clave privada permite al propietario firmar digitalmente sus propias transacciones.

La cadena de bloques selecciona un nodo para crear el siguiente bloque en la cadena, al dar ese privilegio al nodo que soluciona un problema matemático complejo que requieren potencia computacional. Si un nodo soluciona el problema, este consolida al siguiente bloque y lo transmite,

el cual es agregado y verificado por los nodos participantes de la red que también resolvieron el problema matemático validando así la integridad del bloque.

FIGURA 13

DIAGRAMA DEL FUNCIONAMIENTO BÁSICO DE UNA CADENA DE BLOQUES



Nota: La imagen muestra un diagrama de funcionamiento de una cadena de bloques. Adaptado de *Using Blockchain To Support Provenance in the Internet of Things* por (Kaku, 2017).

El nodo que resuelve el problema y publica el bloque es recompensado y este proceso se denomina minería; los nodos restantes agregaran el bloque validado a su registro distribuido de transacciones, así si de alguna manera se llegase a modificar o alterar la información contenida en el bloque de manera maliciosa no coincidiría con el bloque registrado por los demás nodos miembros de la red en el registro distribuido, garantizando así la integridad de las transacciones realizadas. A parte de su capacidad de mantener la seguridad de las transacciones, la tecnología de cadena de bloques permite un consenso distribuido sobre el estado de la base de datos, que garantiza que las transacciones se realicen una sola vez (Kaku, 2017).

- **Contrato Inteligente**

Básicamente un contrato inteligente es una aplicación computarizada que ejecuta automáticamente transacciones y acuerdos comerciales, esta aplicación también hace cumplir las obligaciones de todas las partes en un contrato sin gasto añadido de un intermediario (Tapscott & Tapscott, 2016). Un contrato inteligente es un código trazable e irreversible en una cadena de bloques, que permite la verificación e implementación de un acuerdo o transacción en la cadena de bloques(Pereira et al., 2019).

- **Algoritmo de Consenso**

La clave para la tecnología de cadena de bloques es el consenso ya que fundamentalmente permite a los nodos participantes de la misma, sustentar en un protocolo común de verificación y confirmación las transacciones realizadas, así como su irreversibilidad. Igualmente, este consenso debe proporcionar a los usuarios una copia inalterable y actualizada de las operaciones realizadas en la cadena de bloques (Preukschat et al., 2017). El mecanismo de conceso también se encarga de verificar que los nodos participantes sean honestos, es decir que sean reales y no participantes falsos con fines maliciosos o atacantes, a fin de tener más participación en las decisiones; una vez verificada una transacción esta no podrá ser eliminada o modificada (Plaza, 2018). Nunca un solo nodo o autoridad puede decidir si una transacción debe añadirse al registro, más bien como ya se mencionó, la mayoría de los nodos honestos en la red participan buscando un consenso. El protocolo permite que los dispositivos conectados trabajen juntos como un grupo y en caso de que un nodo falle, este se mantiene apto para llegar al consenso. En casos donde dos mineros logren resolver el problema matemático casi al mismo tiempo, se crea una bifurcación en la cadena de bloques, en este caso se seleccionará la ramificación más larga y las otras se descartan; de esta

manera el protocolo asegura la tolerancia a fallos con redundancia integrada y administración descentralizada de transacciones (Thakur, 2017).

Los algoritmos de consenso usados en plataformas de cadenas de bloques inciden en atributos como; la velocidad de ejecución de transacciones, eficiencia, escalabilidad o la capacidad de manipulaciones maliciosas en el registro distribuido (Pereira et al., 2019); a continuación, se presenta una breve descripción de estos algoritmos.

Prueba de Trabajo o Proof of Work (PoW): El algoritmo de consenso de Prueba de Trabajo es extensamente usado en cadenas de bloques, introducido y aplicado a la criptomoneda Bitcoin (Nakamoto, 2008); básicamente la prueba de trabajo hace referencia a la solución de un problema matemático y su verificación a través de varios puntos en la red, por lo general implica la resolución de un desafío por prueba y error hasta llegar a una solución aceptable. El proceso de solucionar este problema matemático es conocido como minería y es realizado aleatoriamente por varios puntos de la red, idealmente la primera solución aceptable obtiene la recompensa por el desafío, en cualquier caso, la cantidad de tiempo y los recursos de procesamiento requeridos para solucionar el problema son sumamente altos, y es por esto que los mineros reciben una recompensa. El minado garantiza la validez de las transacciones y dificulta falsificaciones de datos (Hanif & Song, 2019). La desventaja de este algoritmo es que necesita gran cantidad de recursos computacionales y por consiguiente conlleva un gasto económico y ambiental en los procesos de minería; y como ventaja brinda seguridad al sistema ya que es muy complicado falsear los proceso y procedimientos (Balmaseda Aranda, 2018).

Prueba de Participación o Proof of Stake (PoS): Es una alternativa al algoritmo de prueba de trabajo que busca el consenso entre los nodos para validar bloques, pero realizando cálculos más sencillos y en base a que los nodos demuestren que tienen participación predomínate

en la red, al haber conseguido con anterioridad cierta cantidad de monedas antes de ser aceptado por la red. En este algoritmo los nodos se seleccionan aleatoriamente para validar bloques, y la probabilidad de esta selección aleatoria depende de la participación que tenga (Tunala & Moncayo, 2018). Su principal beneficio es que no requiere muchos recursos computacionales, además de que genera un menor impacto ambiental.

Prueba de Posesión o Proof of Possession (PoP): Este algoritmo posibilita que una entidad pueda demostrar que estaba en posesión de algún dato concreto en un momento de tiempo, como los datos son almacenados en un servidor confiable, junto con una prueba de que los datos no han sido manipulados, proporciona garantías de seguridad probabilística (Balmaseda Aranda, 2018).

2.3.3. Principios Caracterizadores de la Cadena de Bloques

Como ya se mencionó, una cadena de bloques puede definirse como un libro digital compartido, conformado por una serie de bloques conectados y almacenados en una red distribuida, descentralizada y protegida mediante criptografía, siendo un depósito de información que se almacena de forma incorruptible e irreversible (Jiménez, 2019). Así se desprenden los siguientes principios de esta tecnología:

- **Inmutabilidad**

Como su nombre lo indica, la tecnología de cadena de bloques es básicamente "encadena" bloques sucesivamente mediante la criptografía usando hashes (Martín, 2021a). El encadenamiento debe ser inmutable ya que, si un nodo decidiera modificar el contenido de la cadena de bloques alterando una transacción que ya se ha realizado y se ha incluido en un bloque, esto sería detectado inmediatamente, debido a que el contenido de su versión del libro registro

distribuidos variará. De esta forma, el resto de los nodos denegarán el registro de cualquier otra nueva transacción que pretenda incluir este nodo en su versión, debido a que esta no coincidirá con el contenido del libro registro que tienen el resto de los nodos (Porxas & Conejero, 2018a).

- **Irrevocabilidad**

Este término hace referencia a la incorporación, generación o compartición de bloques e información dentro de una red de cadena de bloques, ya que está no se podrá eliminar; debido a que la información es poseída por todos los usuarios y se distribuye de manera automática a todos y cada uno de los nodos que intervienen en la red (Martín, 2021a; Porxas & Conejero, 2018a).

- **Transparencia**

Determina que todos los usuarios tengan acceso al libro registro o libro digital compartido y a la información sobre todas las transacciones efectuadas. En algunos tipos de redes, existe la posibilidad que usuarios que no forman parte de la red puedan también consultar el contenido de la cadena de bloques, como es el caso de las redes públicas de Bitcoin o Ethereum. Sin embargo, esta transparencia no supone la identificación del autor de las transacciones en todos y cada uno de los casos, debido a que, en algunas de las redes, los usuarios no se identifican para acceder y operar en la red de cadena de bloques. En estos casos, las transacciones son visibles a todos los que acceden a la red, pero se vinculan a un código identificatorio que, en muchas ocasiones, no revela la identidad del sujeto que realiza la transacción (Martín, 2021a).

2.3.4. Tipos de Redes de Cadena de Bloques

Es posible diferenciar algunos tipos de Cadenas de Bloques, ya sea por el acceso limitado a la red o por el acceso específico a los datos de la Cadena de Bloques en la red; como lo manifiesta en su investigación Schuurmans: *"Configurando el límite de acceso a la red se puede seleccionar*

entre una Cadena de Bloques Pública o Privada y configurando el límite de acceso a datos se puede seleccionar entre una Cadena de Bloques sin permiso o una autorizada" (Schuurmans, 2019); es decir que unas Cadenas de Bloques pueden estar abiertas a la participación de cualquiera que lo desee, y otras que se limitan solo a algunos participantes; y de manera preestablecida siempre independientes de una entidad que supervise o valide sus procesos (Preukschat et al., 2017). A continuación, se describen los tipos de Cadenas de Bloques.

- **Cadenas de Bloques Públicas.**

Son cadenas de bloques abiertas que permiten unirse, contribuir y ver los contenidos a cualquiera; estas representan la verdadera descentralización y transparencia; sin embargo, son generalmente lentas, más costosas de mantener y operar, además sus mecanismos de conceso son más complicados para prevenir ataques Sybil (Branislav, 2018). En otras palabras, cualquier nodo conectado a Internet puede participar ya sea accediendo y consultando las transacciones de la cadena de bloques o validando bloques, siendo así una cadena de bloques totalmente descentralizada (Kaku, 2017); algunos ejemplos de este tipo de cadena de bloques más conocidas son Bitcoin y Ethereum (Balmaseda Aranda, 2018).

- **Cadena de Bloques Privadas.**

En una cadena de bloques completamente privada, los permisos de escritura se mantienen centralizados en una organización, los permisos de lectura pueden ser públicos o restringidos en un grado arbitrariamente; las aplicaciones probables incluyen administración de la base de datos o auditoría interna (Kikitamara, 2017). Es decir que, este tipo de cadena de bloques permite un solo nodo superior autorizado para el acceso y cambio de datos, lo que significa que los permisos de escritura están centralizados en una organización; por otra parte, el resto de nodos tiene acceso

limitado sobre la cadena de bloques privada, y hay solo algunos nodos designados quienes están permitidos validar transacciones; además las transacciones son menos costosas en comparación con un cadena de bloques publica, este tipo es aplicable para grupos de área cerrados como una intranet (Komal, 2019); ejemplos de cadenas de bloques privadas incluyen Multichain, Chain, Blockstack y otras más (Kaku, 2017).

- **Cadena de Bloques sin Permiso (Permisssionless Blockchain).**

Es una cadena de bloques que no tiene restricciones sobre el acceso a la información para los participantes de la red; sin embargo, hay muchos casos en donde se prefiere la confidencialidad, y no sería factible que todos los participantes de la red tengan los datos disponibles. La confidencialidad en una cadena de bloques es la capacidad de los nodos de ocultar el contenido de las transacciones, o incluso la identidad al haber participado en una transacción, frente a otros nodos (Schuurmans, 2019). Este modelo al igual que en de la cadena de bloques pública permite que cualquiera en el mundo, pueda acceder y consultar las transacciones de la cadena de bloques, de igual manera, cualquiera pude convertirse en participante, con igualdad frente a los demás participantes, así como con los nodos involucrados; el único filtro vendería de cierta anonimidad disponible para los participantes en cuanto a datos personales se refiere y que está presente en Bitcoin o Ethereum. (Balmaseda Aranda, 2018).

- **Cadena de Bloques Autorizadas (Permissioned Blockchain).**

El principio de las cadenas de bloques autorizadas es que hay una regulación para quien es permitido unirse y participar en la red. Esto pude hacerse por un consorcio de compañías, agencias gubernamentales u otras organizaciones, ya sea invitando nuevos miembros uno por uno, o por un conjunto de criterios preestablecidos. Los beneficios, además de incrementar la privacidad,

incluyen el potencial para más flexibilidad en adaptarse a la red, mejor escalabilidad y transacciones más rápidas (Bergquist, 2017). En este tipo de cadena de bloques se combina las características de una cadena de bloque pública y privada, en términos de centralización y accesibilidad; esta es diseñada para redes semi cerradas, compuestas por muchas partes o empresas, donde el proceso de consenso es controlado por nodos los cuales son preespecificados por los participantes en un consorcio (Komal, 2019). Un ejemplo de cadena de bloques autorizada es RippleNet donde Microsoft, MIT y CGI trabajan como validadores de transacciones (Hanif & Song, 2019).

- **Cadenas de Bloques Híbridas.**

Es la combinación de las anteriores, donde se invitan a nodos participantes para el mantenimiento y las restricciones de consenso, se dejan las transacciones públicas y visibles, pero no su contenido; un ejemplo es BigchainDB y Evernym (Balmaseda Aranda, 2018).

Una descripción general de los tipos de cadenas de bloques, en función de las diferentes acciones posibles que se pueden realizar con transacciones y bloques, se muestran en la Tabla 2.

TABLA 2

TIPOS DE REDE DE CADENAS DE BLOQUES

Tipo de Cadena de Bloques		Lectura de Transacciones	Escritura de Transacciones	Validación de Transacciones/Bloques
Pública	Sin Permiso	Cualquiera	Cualquiera	Cualquiera
	Autorizadas	Cualquiera	Participantes Autorizados	Participantes Autorizados (todos o un subconjunto, dependiendo del protocolo)

Privada	Sin Permiso (Consorcios)	Participantes Autorizados	Participantes Autorizados	Participantes Autorizados (todos o un subconjunto, dependiendo del protocolo)
	Autorizadas (Empresas)	Conjunto limitado de nodos autorizados	Operador de Red	Operador de Red

Nota: Esta tabla muestra un resumen de los tipos de redes de cadenas de bloques. Tomado y adaptado de: *Blockchain technology potential in the chemical industry: an exploratory research on the value of blockchain technology for supply chain management of organizations in the chemical industry*. (Abril), 1–26, por (Schuurmans, 2019).

2.3.5. Ventajas e Inconvenientes de la Tecnología de Cadena de Bloques

La tecnología de cadena de bloques, como se ha explicado a lo largo de los puntos anteriores; es una tecnología novedosa cuyas aplicaciones prácticas pueden ayudar a innovar muchos ámbitos relacionados a la seguridad de redes e informática; sin embargo, a continuación, se mencionan de manera general los beneficios y limitantes que esta tecnología presenta actualmente.

En cuanto a las ventajas que plantea la cadena de bloques podemos destacar, fundamentalmente, su inmutabilidad como registro de datos; esto le otorga, la capacidad de crear un registro, sin la necesidad de que exista una autoridad central que autorice, verifique y realice la transacción. Asimismo, proporciona gran seguridad en la transmisión de datos gracias a la criptografía, reduciendo riesgos de robo o filtración de la información en gran medida gracias al anonimato y codificación de las transacciones, se disminuyen los costes de transacción y se realiza estas en tiempo real; además, se elimina el error humano y existe un aumento de la transparencia y fiabilidad de las operaciones. Proporciona gran seguridad, ya que, al estar compuesta por varios nodos, requiere que se ataque a cada uno de los nodos de manera coordinada y simultanea para que la red caiga o pueda manipularse de alguna forma la información, acción prácticamente imposible debido a los registros compartidos de las transacciones (Martín, 2021b).

Por otra parte, uno de los principales inconvenientes es la falta de regulación; ya que la popularidad adquirida por la tecnología de cadenas de bloques en el uso y desarrollo como base de las criptomonedas, ha ocasionado que muchos sectores y entidades involucrados directamente en el sector económico no reconozcan y no garanticen el uso de la criptomonedas como monedas de curso legal, por ende la tecnología de cadena de bloques que están usan para realizar las transacciones no se ha estandarizado debidamente.

Otro inconveniente a mencionar se relaciona con la protección de datos; ya que en las redes basadas en la tecnología cadena de bloques se manejan millones de datos de carácter personal y no personal, lo que plantea problemas de falta de control de la información, así como vulnerabilidad de esta; así como la seguridad ya que está claro que al tratarse de una tecnología novedosa y con poca regulación puede ser susceptible de ciberataques generalizados debido a su vulnerabilidad en materia de seguridad en las contraseñas y debido a problemas en el cifrado de datos o en los permisos de acceso (Martín, 2021b).

2.3.6. Aplicaciones de la Cadena de Bloques

Las cadenas de bloques no fueron muy populares en sus inicios, en 1991 cuando el sistema de cadena de bloques fue creado y no fue usado hasta el 2009 cuando Satoshi Nakamoto pseudónimo de la persona, personas u organización detrás del desarrollo de esta tecnología lo uso de base para su criptomoneda más conocida, el Bitcoin; pero las aplicaciones de este sistema de cadena de bloques son muy diversas y van más allá de la generación de criptomonedas; y puede usarse para firmar contratos, para votar en elecciones, guardar registros médicos, bancarios y muchas otras aplicaciones que aún están por desarrollarse (Baldeón & Zambrano, 2018).

En el año 2012 el Banco Central Europeo definió a las criptomonedas o monedas virtuales como un tipo de dinero digital no regulado, normalmente emitido y controlado por sus

desarrolladores, usado y aceptado entre los miembros de una concreta comunidad virtual. Posteriormente, en la Directiva del Parlamento Europeo y del Consejo por la que se modifica la Directiva (UE) 2015/849, la cual hace referencia relativa a la prevención de la utilización del sistema financiero para el blanqueo de capitales o la financiación del terrorismo, se definen las monedas virtuales como una representación digital de valor no emitida ni garantizada por un banco central ni por una autoridad pública, no necesariamente asociada a una moneda establecida legalmente, que no posee el estatuto jurídico de moneda o dinero, pero aceptada por personas físicas o jurídicas como medio de cambio y que puede transferirse, almacenarse y negociarse por medios electrónicos (Porxas & Conejero, 2018b).

Hoy en día existen miles de criptomonedas, cientos de ellas son creadas cada semana, aunque las más extendidas son el bitcoin (BTC) y el ether (ETH). Las redes Bitcoin y Ethereum tienen elementos comunes, pero se diferencian en algunos aspectos; mientras que el Bitcoin como red, fue creada en 2009 y se programó para finalizar la emisión de bitcoins cuando alcance la cifra de 21 millones de BTC emitidos. La red está diseñada para funcionar como medio de pago entre aquellos que deciden voluntariamente aceptarlo como tal (Porxas & Conejero, 2018b). En cambio, en 2015 es creada la Red Ethereum por Vitalik Buterin y la emisión de ethers (ETH) es en principio ilimitada, siendo la emisión anual 18 millones de unidades. La principal diferencia con la red Bitcoin es que la red Ethereum permite realizar transacciones más sofisticadas que solo el pago, al admitir que operen sobre su estructura ciertos smart contracts, a los que nos referiremos más adelante (Porxas & Conejero, 2018b).

Otra forma de medio de pago son las llamadas ICO (*Initial Coin Offering*) que ofrecen, a cambio de moneda de curso legal o moneda virtual, un token, una especie de vale virtual, instrumentado como apunte digital del derecho a la obtención de distintos beneficios posibles,

como el acceso o posibilidad de adquisición de un producto o servicio todavía no lanzado al mercado (*utility tokens*), o, incluso, un interés participativo en los futuros ingresos o el posible aumento del valor de la entidad emisora o del negocio (*equity tokens*) (Porxas & Conejero, 2018b).

Al igual que las ICO, los recién surgidos ILP (*Initial Loan Procurements*) están destinando a captar fondos para nuevos proyectos. En este caso, los usuarios que deciden acudir a la oferta reciben tokens de acceso a derechos de crédito transmisibles a terceros o FLATS (*Future Loan Access Tokens*). La aportación se articula a través de un contrato de préstamo con el receptor de los fondos en formato *smart contract*, código autoejecutable, en cuya virtud el prestador recibe los pagos de forma automática y sin la intervención de operador alguno. Sus valores defienden que los ILP permiten que los beneficios de los inversores no estén condicionados por la volatilidad de sus tokens, como sucedería a su decir en el caso de las ICO, puesto que el retorno solamente depende de los beneficios que el negocio llegue a obtener cada año lo que, de hecho, supuestamente también sucede bajo algunas formas de ICO. Aunque mucho menos numerosos que los de ICO, los ejemplos de ILP son una realidad en algunos países especialmente en Estonia (Porxas & Conejero, 2018b).

Así mismo los *smart contracts* o contratos inteligentes, se describen como contratos autoejecutables; si son o no contratos dependerá en cada caso de si concurren los requisitos de consentimiento, objeto y causa para ello. En cualquier caso, en rigor, la aptitud para ser jurídicamente contrato no corresponde a lo que comúnmente se conoce como *smart contract*, y que no es más que programa autoejecutable, sino a lo que se ha denominado contrato legal inteligente, del que el *smart contract* es solo parte, y que se ha definido como el contrato celebrado a través de una página web accesible para las partes cuya forma está constituida por la interfaz de usuario de la aplicación externa y uno o varios programas autoejecutables (*smart contracts*)

residentes en la cadena de bloques con capacidad para actuar recíprocamente con dicha interfaz (Porxas & Conejero, 2018b).

Finalmente, y con el reciente auge de criptomonedas, monedas digitales de bancos centrales y otros cripto-activos, se unen ahora los tokens no fungibles o NFT, por sus siglas en inglés (*Non-Fungible Token*), estos nuevos activos digitales comienzan a captar el interés de los inversores. En un sentido amplio, los tokens digitales pueden considerarse certificados de propiedad de activos virtuales o físicos. Dentro de estos, los NFT surgen como un tipo especial de token criptográfico que representa algo único. La diferencia con otros activos digitales, como pueden ser las criptomonedas, reside en que dichos NFTs no son fungibles. Estos tokens, por tanto, se caracterizan porque tienen propiedades únicas, por lo que no se pueden intercambiar. En la práctica, son activos individuales, indivisibles e insustituibles, que se generan digitalmente e identifican inequívocamente su propiedad. Los NFTs comienzan a usarse también como activos de garantía o colateral en la concesión de préstamos. Actualmente hay mercados de cambio y empeño como NFTfi que permiten a sus usuarios depositar sus NFTs como garantía para obtener un préstamo denominado en alguna criptomoneda (Funcas, 2021).

2.4. Trabajos Relacionados

Como referencia de investigación bibliográfica previa y afín al desarrollo de esta investigación; el artículo científico "*Trustroam: A Novel Blockchain-Based Cross-Domain Authentication Scheme for Wi-Fi Access*" propone un novedoso esquema de autenticación cruzada, en redes Wi-Fi basado en Cadena de Bloques, diferente a las soluciones jerárquicas tradicionales, autenticado usuarios y servidores de una manera anónima y distribuida, evitando varios problemas serios como, un único punto de falla o fuga de privacidad. Mediante los mecanismos de consenso

distribuidos y autenticación mutua, el esquema pretende ser altamente tolerante a fallas y comprometido a manejar ataques a servidores (Wei et al., 2019).

Mientras que en trabajo de tesis "*Authentication, Authorization and Accounting with Ethereum Blockchain*" se propone, un modelo alternativo para autenticación, autorización y contabilidad usando Cadena de Bloques basado en Ethereum, que se desarrolló con el fin de permitir a los usuarios acceder a un servicio en la nube y poder autenticar, autorizar y contabilizar con una sola identidad, sin la necesidad de compartir ninguna información privada de los usuarios (Thakur, 2017).

Otra propuesta descrita en "*Decentralized Access Control Using Blockchain*" se basa en la implementación de un prototipo de para un sistema de control de acceso descentralizado que soporte y de transparencia, sea auditable, inmutable e igualitario en un entorno colaborativo, usando la plataforma de Cadena de Bloques *Multichain*, servicios web RESTful y lenguaje de programación Java. El prototipo desarrollado evalúa dos métricas: el promedio de tiempo de respuesta y el rendimiento (Jamsrandorj, 2017).

De igual manera en "*BLOCKCHAIN: Aplicación en el Registro de la Propiedad e implicaciones en materia probatoria*" se aplica la tecnología de cadena de bloques en el Registro de la Propiedad abriendo una puerta a la modernización, agilización y simplificación de los trámites de registro. En muchos países ya han puesto en marcha proyectos piloto con el objetivo de desarrollar una red de cadena de bloques aplicable a los registros de bienes inmuebles, en aras de examinar cómo podría aplicarse esta tecnología, qué ventajas tendría y qué inconvenientes podría plantear. Pese a que muchos consideran que puede traer consigo una auténtica revolución de esta institución, pudiendo incluso llegar a sustituirla, parece conveniente estudiar si realmente esta nueva tecnología es compatible con los principios y garantías inherentes al registro de la

propiedad y, en caso de que lo fuera, cuáles son sus límites y en qué procedimientos podría utilizarse. Asimismo, se analizará el impacto que la implementación de la tecnología de la cadena de bloques puede ocasionar en el proceso civil, concretamente en materia probatoria (Martín, 2021b).

En "*Cómo integrar Blockchain en una arquitectura de software: resultados de una Revisión Multivocal de la Literatura*" se plantea el uso de la tecnología de cadena de bloques como un registro distribuido e inmutable que facilita el proceso de almacenar transacciones y el seguimiento de activos en una red descentralizada. Es una tecnología con el potencial de revolucionar industrias, desde las finanzas hasta el IoT. El objetivo de este trabajo es identificar las redes de Blockchain disponibles y sus principales características (algoritmo de consenso, descentralización, *smart contracts*, origen de la red), así como las posibles implicaciones de estas particularidades. Para cumplir el objetivo y analizar las principales características desde el punto de vista del arquitecto de software, se llevó a cabo una Revisión Multivocal de la Literatura. El resultado es la identificación y caracterización de 112 redes de Blockchain divididas en tres grandes familias: de uso general, de uso específico (financiero, videojuegos, identidad, pagos) y derivadas de criptomonedas. Se presenta una lista detallada de las redes disponibles. Este mapeo provee una guía a los arquitectos de software para que puedan tomar decisiones justificadas a la hora de incorporar la tecnología Blockchain (Sobral, 2021).

Finalmente, en lo referente al internet de las cosas (IoT) ha tenido un crecimiento exponencial durante los últimos años, sin embargo, la adopción del IoT se ve amenazada por diferentes problemáticas propias de esta tecnología. Escalabilidad, seguridad y privacidad, son problemas claves que se tienen que resolver a un corto plazo. En el documento "*Aplicación de Blockchain para la seguridad de los datos del Internet of Things*" la cadena de bloques se muestra como una

tecnología altamente segura, escalable y descentralizada, por lo cual se propone combinar estas dos tecnologías para sobrellevar inconvenientes de seguridad y privacidad de los datos en el IoT. Implementando una Cadena de bloques Privada con *Hyperledger Fabric* en tres máquinas virtuales junto a una API REST y una web *AngularJS*. Junto con demostrar que Blockchain y IoT se complementan, se verificó que *Hyperledger Fabric* es un *framework* altamente seguro, escalable y eficiente en cuanto a performance tanto en la red de cadena de bloques como el consumo de hardware. Esta implementación no solamente sirve para combinar la cadena de bloques con IoT, sino que abre las puertas para otros casos de usos en donde se utilicen distintas tecnologías(Reyes, 2018).

Como se puede apreciar en las descripciones citadas anteriormente, el uso de la tecnología de cadena de bloques, por su estrecha relación con mecanismos criptográficos de seguridad y algoritmos de consenso puede usarcé para garantizar y validar el flujo de información generando registros, de manera autónoma sin depender de un ente centralizado; esta tecnología que se ha ido popularizando con el auge de las criptomonedas y los NFTs; de tal manera, que es una alternativa a mecanismos transaccionales tradicionales por la robustez al momento de comprobar la identidad de las partes, mientras se demuestra la validez de las acciones que estas realizan entre sí. Además, los trabajos relacionados mencionados y mucha más bibliografía muestran los parámetros fundamentales y necesarios para hacer uso de la tecnología de cadena de bloques, aplicada en diferentes ámbitos con el único fin de brindar protección mediante los fundamentos de la seguridad informática; confidencialidad, integridad y disponibilidad; para que partiendo de esta base se pueda desarrollar el mecanismo de autenticación inalámbrico propuesto.

3. Capítulo III: Diseño

En este capítulo se muestra el proceso y metodología seleccionado para determinar requerimientos que solucionen y solventen el uso de la tecnología de cadena de bloques como mecanismo de autenticación alterno, respecto al método usado actualmente en la Facultad de Ingeniería en Ciencias Aplicadas, para el diseño y realización de pruebas posteriores en un entorno controlado; y en donde se realizará un estudio bibliográfico e investigativo de trabajos relacionados, artículos científicos y publicaciones concernientes al uso de la tecnología de cadena de bloques; para determinar de mejor manera las herramientas a utilizar en el desarrollo e implementación de la propuesta del presente trabajo de grado.

3.1. Metodología de Investigación

Con el propósito de recopilar y organizar los antecedentes e información relacionada para la respectiva ejecución de los objetivos propuestos en el presente proyecto se hace necesaria una Investigación Descriptiva, la cual considera distintos tipos de datos para la generación de información entre los cuales se encuentran conversaciones, entrevistas, documentos y evidencias bibliográficas del levantamiento de información (Lozada & Yangali, 2022).

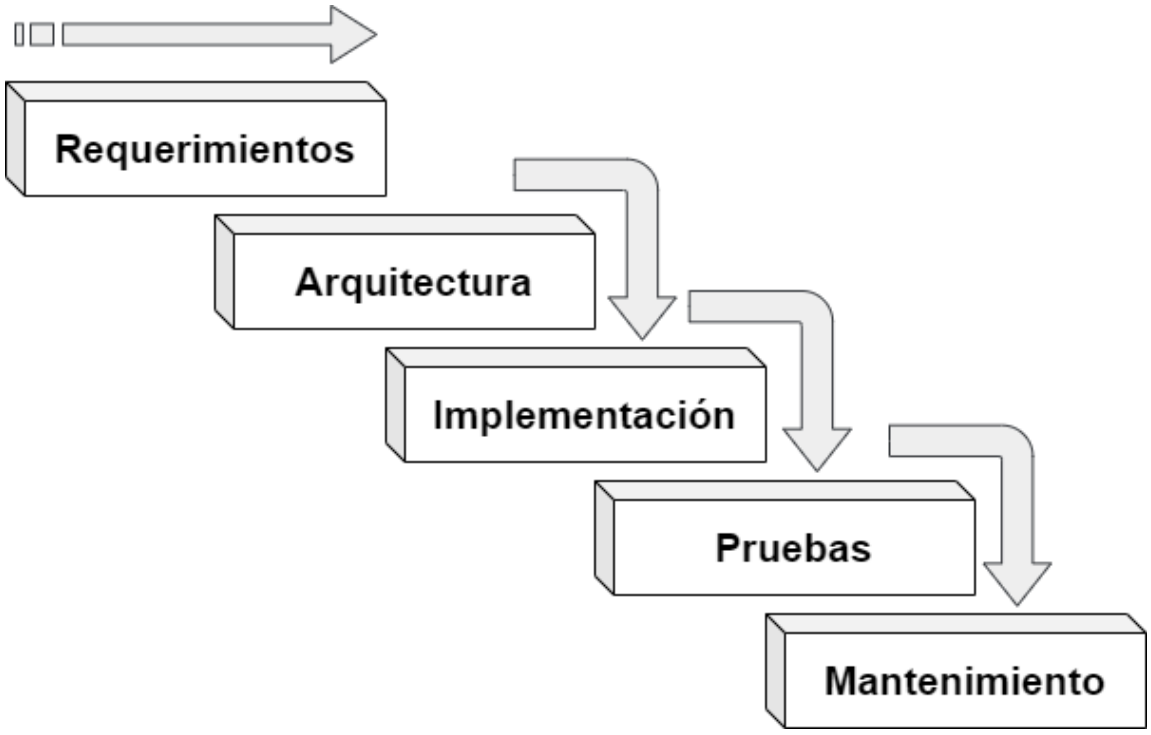
3.2. Metodología de Diseño y Desarrollo

Para el proceso de diseño y posterior desarrollo del presente trabajo de titulación se ha seleccionado la Metodología en Cascada ya que permite una fácil implementación y secuencia lógica; donde básicamente se orienta por la planeación inicial en el desarrollo con el levantamiento de requerimientos del proyecto con una estructura definida y que sigue una secuencia unidireccional sin permitir retroceso sobre la marcha en el desarrollo del proyecto (Aguirre & Aguirre, 2020) además establece fases y etapas para administrar el desarrollo y seguimiento del

proyecto (Solano & Porras, 2020) como se muestra en la Figura 14, las cuales se adaptarán para la realización de este trabajo y que se resumen a continuación:

FIGURA 14

METODOLOGÍA EN CASCADA



Tomado y adaptado de *Metodologías para el desarrollo de Proyectos* por (Aguirre & Aguirre, 2020).

- **Requerimientos:** Análisis y definición de las especificaciones y requisitos funcionales del proyecto, documentación e información relevante para el desarrollo de la siguiente etapa, así como el establecimiento de los actores directamente relacionados al proyecto.

- **Arquitectura:** En base a las especificaciones y requerimientos se establece una arquitectura, que describe componentes fundamentales mediante el uso de diagramas y esquemas que establecen las herramientas necesarias para el desarrollo del proyecto.

- **Implementación:** En esta etapa se da cumplimiento a lineamientos establecidos por la arquitectura, además se realizan evaluaciones tempranas en base al seguimiento de los diagramas planteados.

- **Pruebas:** Supervisión y ejecución, donde se verifica el cumplimento de requerimientos, se evalúa el funcionamiento del diseño en un entorno controlado; posteriormente se depuran y corrigen inconsistencias.

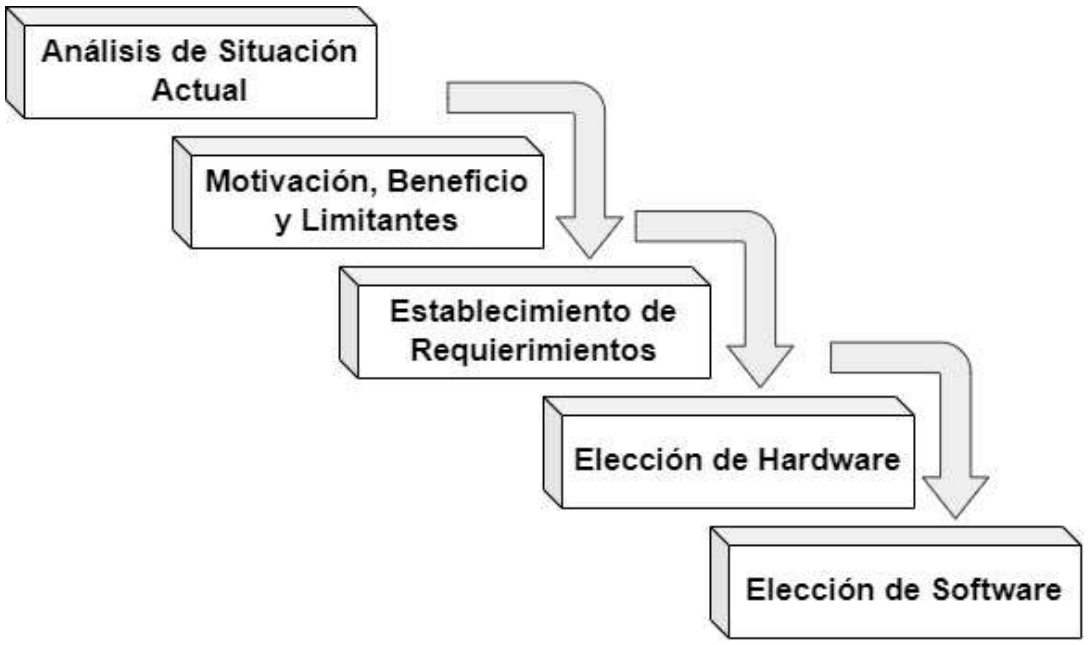
- **Mantenimiento:** Etapa final del proyecto la cual puede requerir mejoras, modificaciones o corrección de errores en función de nuevas especificaciones, así como ser simplemente una etapa de refinamiento del proyecto.

3.3. Requerimientos

En el análisis de requerimientos se usará las metodologías de investigación descritas en la Sección 3.1 y siguiendo el proceso detallado en la Figura 15; donde se obtendrá la información mediante el Departamento de Desarrollo Tecnológico e Informático (DDTI) de la Universidad Técnica del Norte (Anexo 1) además se sustentara con la respectiva recopilación bibliográfica para conocer la situación actual y el funcionamiento en general de la infraestructura inalámbrica de red correspondiente a la Facultad de Ingeniería en Ciencias Aplicadas; sus equipos, mecanismos de acceso, protocolos, redes inalámbricas propagadas, su distribución y configuración; información que permitirá definir los requisitos y establecer las especificaciones para el diseño del mecanismo de autenticación basado en la tecnología de cadena de bloques.

FIGURA 15

PROCESO PARA LA OBTENCIÓN Y ESTABLECIMIENTO DE REQUERIMIENTOS

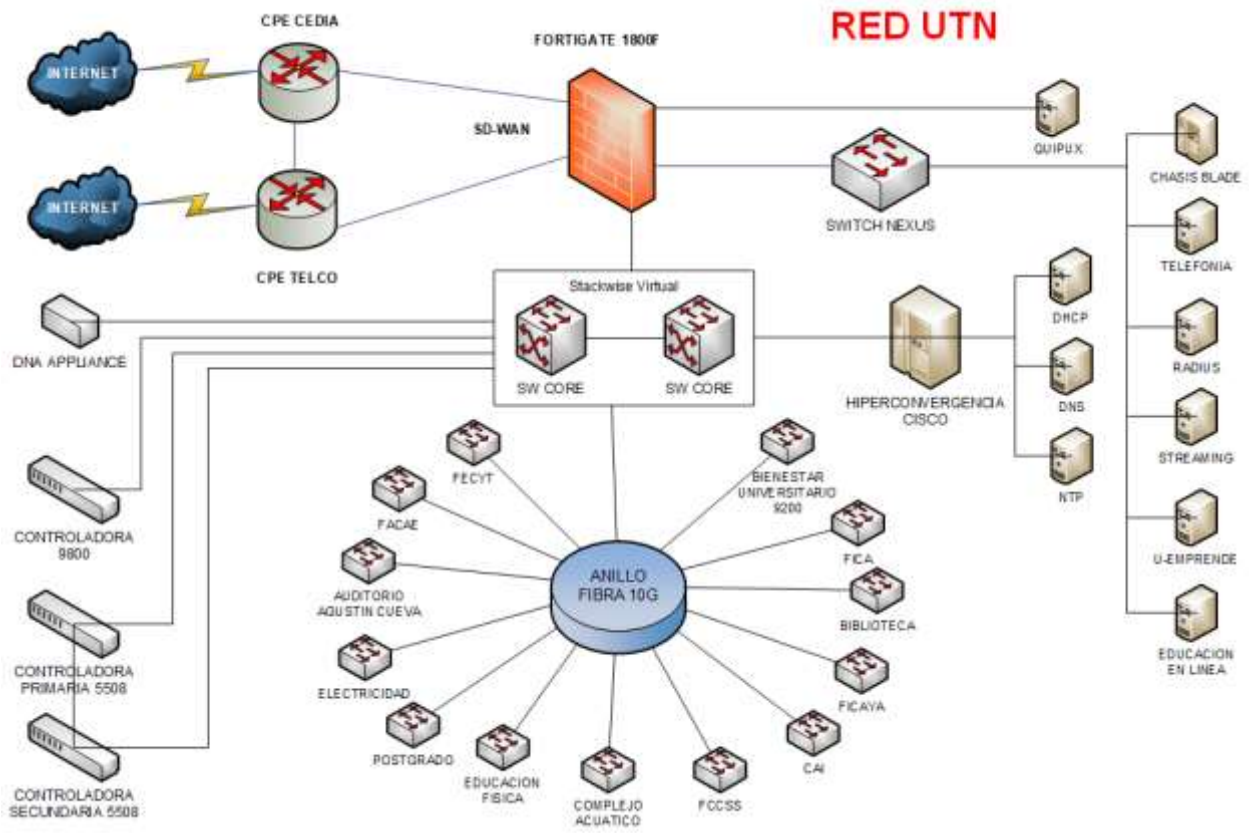


3.3.1. Situación Actual Red Inalámbrica FICA

La información facilitada por el DDTI de la UTN, permitió conocer cómo se configura y distribuye la red inalámbrica global, para el campus universitario ubicado en el sector El Olivo; que se muestra en la Figura 16, destacando muestra la distribución de los equipos de acceso, autenticación, autorización, control y conexión inalámbrica, los cuales también intervienen directamente en el funcionamiento e infraestructura inalámbrica de red de la Facultad de Ingeniería en Ciencias Aplicadas, haciendo énfasis además en el servidor Radius que cumple con funciones de autenticación, autorización, control y conexión inalámbrica.

FIGURA 16

DISTRIBUCIÓN GENERAL DE CONEXIONES RED ENTRE DDTI CAMPUS EL OLIVO UTN



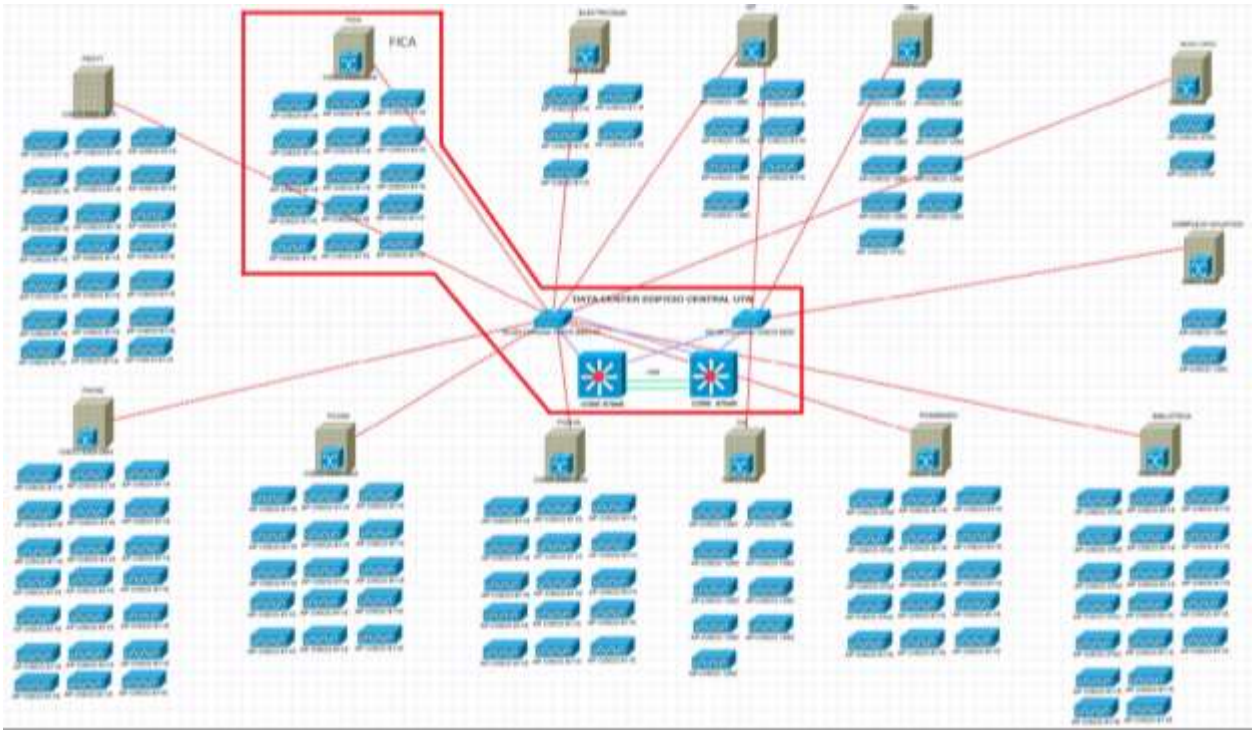
Nota: La imagen muestra un diagrama generalizado de los componentes de backbone en la infraestructura tecnológica del campus UTN El Olivo. Fuente Responsable Infraestructura Tecnológica DDTI UTN.

3.3.1.1. Topología

Respecto a la Facultad de Ingeniería en Ciencias Aplicadas, en esta se distribuyen múltiples puntos de acceso, los cuales se conectan a un switch Cisco C9300-48UXM-E ubicado en el Data Center de la facultad y que a su vez se conecta con data center de la UTN ubicado en el Edificio Central mediante el anillo de fibra óptica desplegado y que interconecta los principales edificios y facultades, permitiendo que los puntos de acceso propaguen la señal y el SSID respectivo de la red Eduroam como se muestra en la Figura 17.

FIGURA 17

DISTRIBUCIÓN DE CONEXIONES RED INALÁMBRICA DDTI CAMPUS EL OLIVO UTN



Nota: La imagen muestra un diagrama de los equipos y puntos de acceso inalámbricos de la infraestructura del campus UTN El Olivo y de la FICA. Fuente: Responsable Infraestructura Tecnológica DDTI UTN.

3.3.1.2. Equipos

A continuación, se describen los equipos que forman parte e intervienen en la conexión y acceso a la infraestructura de red inalámbrica presente en el edificio de la Facultad de Ingeniería en Ciencias Aplicadas, cuya distribución se muestra en la Tabla 3.

- **Cisco Catalyst 9800-40 Wireless Controller**

Controladora de red inalámbrica, cuenta con un sistema altamente escalable y una plataforma flexible que permite desplegar servicios de acceso inalámbrico para empresas en un área amplia, mejorando el rendimiento de la tecnología 802.11n y monitoreando simultáneamente

varios puntos de acceso para garantizar su rendimiento; permite agregar, monitorear y habilitar los puntos de acceso desplegados en las diferentes dependencias del campus universitario; realiza un control de equipos y ejecuta actualización de software y mantenimiento (CISCO, 2021b).

- **Cisco 5520 Wireless Controller**

La controladora inalámbrica Cisco 5520 proporciona control, administración y solución de problemas centralizados para implementaciones a gran escala en implementaciones de proveedores de servicios y grandes campus. Ofrece flexibilidad para admitir múltiples modos de implementación en el mismo controlador. Como componente de la red inalámbrica unificada de Cisco, este controlador proporciona comunicaciones en tiempo real entre los puntos de acceso Cisco Aironet y los puntos de acceso Cisco Catalyst, Cisco Prime Infraestructura y el motor de servicios de movilidad de Cisco, y es interoperable con otros controladores de Cisco (CISCO, 2021a).

- **Switch Cisco C9300-48UXM-E**

El equipo switch Cisco Catalyst 9300, proporciona hasta 1 Tbps de capacidad y UPOE+ de 90 vatios en una plataforma de conmutación apilable. Los switchs Catalyst 9000 forman la base de Cisco Software-Defined Access, para una arquitectura empresarial líder para brindar seguridad, IoT y comunicación en la nube (Cisco, 2020). Usado como equipo de distribución el cual se conecta al switch de Core mediante el anillo de fibra óptica y se ubica en el data center de la FICA.

- **Swicth Cisco WS-C4510R+E**

Los switches Cisco Catalyst 4500 Series permiten redes sin fronteras, proporcionando experiencias de usuario seguras, móviles y de alto rendimiento; permiten seguridad, movilidad, alto rendimiento de aplicaciones, video y ahorro de energía en una infraestructura que admite,

virtualización y automatización. Los switches Cisco Catalyst de la serie 4500 brindando rendimiento y, escalabilidad y servicios sin límites (CISCO, 2017). Este equipo también está ubicado en el data center de la FICA.

- **Swicth Cisco WS-C2960-48TC-L**

Son una familia de switches Cisco de configuración fija, Cisco independientes que proporcionan Fast Ethernet y Gigabit Ethernet, lo que permite mejores servicios LAN para empresas. El Catalyst 2960 ofrece seguridad integrada, incluyendo el control de admisión de red, la calidad de servicio avanzada, y la resistencia para entregar servicios inteligentes para el borde de la red (DS3Comunications, 2022). Se despliegan para brindar acceso cableado en los diferentes pisos y laboratorios del edificio de la FICA.

TABLA 3

SWITCHS DESPLEGADOS EN LA INFRAESTRUCTURA DE RED DE LA FICA

Ubicación	Identificador	Marca	Modelo
Datacenter	SW01.FICA.DC.DIS.PB.R01	Cisco	C9300-48UXM-E
Datacenter	SW02.FICA.DC.DIS.PB.R01	Cisco	WS-C4510R+E
FicaLab101	SW-Arquímedes	Cisco	WS-C2960-48TC-L
FicaLab201	SW-Bernoulli	Cisco	WS-C2960-48TC-L
FicaLab301	SW-Copérnico	Cisco	WS-C2960-48TC-L
FicaLab302	SW-Coulomb	Cisco	WS-C2960-24TC-L
FicaLab401	SW-Descartes	Cisco	WS-C2960-48TC-L
FicaLabCisco01	SW-Euclides	Cisco	WS-C2960-48TC-L
FicaLabCisco02	SW-Euler	Cisco	WS-C2960-48TC-L
SalaInvestigacion01	SW-Fourier	Cisco	WS-C2960-48TC-L
AsoProfesores01	SW-Galileo	Cisco	WS-C2960-24TC-L
FicaLab05	SW2-LAB5	Cisco	WS-C2950-24
Datacenter	SW_01_FICA	3COM	SuperStack 3226
FicaLab05	SW1-LAB5	Cisco	WS-C2950-24
Fica Lab 07	SW-LAB7	Cisco	WS-C2950-24

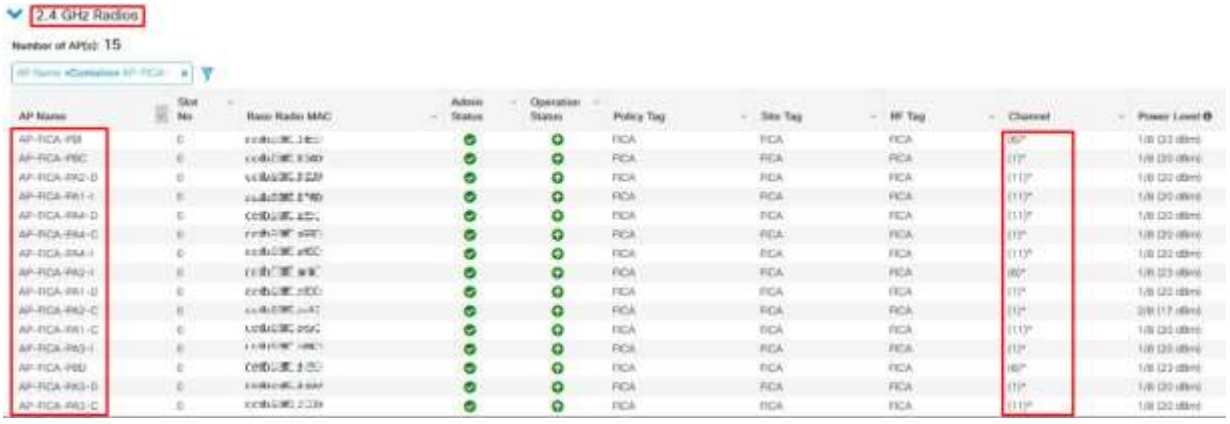
Nota: La tabla muestra los switchs usados en la infraestructura de red de la Facultad de Ingeniería en Ciencias Aplicadas. Fuente: Responsable Infraestructura Tecnológica DDTI UTN.

- **Access Point Cisco C9115AXI-A**

La serie Cisco Catalyst 9115 con Wi-Fi 6 es una generación de puntos de acceso empresarial, los cuales se encuentran desplegados en el edificio de la facultad que y se detallan en la Tabla 4 y se observan la propagación de señal WiFi en frecuencia de 2.4 GHz y 5 GHz respectivamente en las capturas de las Figuras 18 y 19. Los equipos son resistentes, seguros e inteligentes; con rendimiento constante en entornos exigentes con crecimiento exponencial de dispositivos de Internet de las cosas (IoT) y aplicaciones de próxima siguiente generación. Proporcionan resiliencia resistencia y conectividad superior, seguridad integrada con clasificación y contención avanzadas e innovaciones de hardware y software para automatizar, proteger y simplificar las redes. La generación de puntos de acceso Cisco Catalyst 9100, con capacidades de Wi-Fi 6 (802.11ax) de alto rendimiento e innovaciones en seguridad y análisis de RF, permite la digitalización de un extremo a otro y ayuda a acelerar la implementación de servicios (CISCO, 2023)comerciales más allá de Wi-Fi.

FIGURA 18

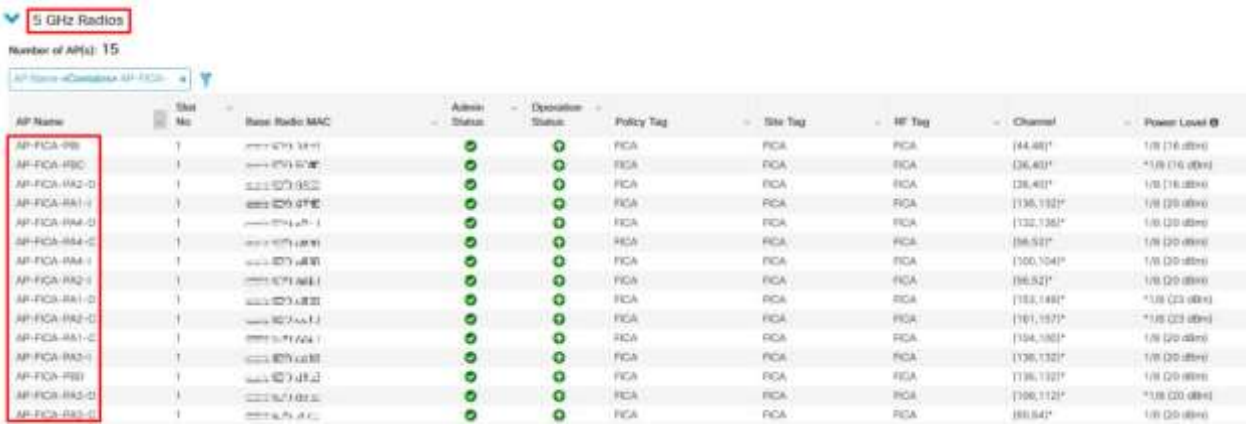
CAPTURA DE EQUIPOS DESPLEGADOS EN EL EDIFICIO FICA PARA LA BANDA DE 2.4 GHZ



Fuente: Responsable Infraestructura Tecnológica DDTI UTN.

FIGURA 19

CAPTURA DE EQUIPOS DESPLEGADOS EN EL EDIFICIO FICA PARA LA BANDA DE 5 GHZ



Fuente: Responsable Infraestructura Tecnológica DDTI UTN.

TABLA 4

PUNTOS DE ACCESO DESPLEGADOS EN LA INFRAESTRUCTURA INALÁMBRICA DE RED DE LA FICA

Nro.	MODELO	NOMBRE
1	CISCO C9115AXI-A	AP-FICA-PBI
2	CISCO C9115AXI-A	AP-FICA-PBC
3	CISCO C9115AXI-A	AP-FICA-PA2-D
4	CISCO C9115AXI-A	AP-FICA-PA1-I
5	CISCO C9115AXI-A	AP-FICA-PA4-D
6	CISCO C9115AXI-A	AP-FICA-PA4-C
7	CISCO C9115AXI-A	AP-FICA-PA4-I
8	CISCO C9115AXI-A	AP-FICA-PA2-I
9	CISCO C9115AXI-A	AP-FICA-PA1-D
10	CISCO C9115AXI-A	AP-FICA-PA2-C
11	CISCO C9115AXI-A	AP-FICA-PA1-C
12	CISCO C9115AXI-A	AP-FICA-PA3-I
13	CISCO C9115AXI-A	AP-FICA-PBD
14	CISCO C9115AXI-A	AP-FICA-PA3-D
15	CISCO C9115AXI-A	AP-FICA-PA3-C
16	CISCO AIR-AP1562E-A-K25	AP-EXTERIOR-FICA

Fuente: Responsable Infraestructura Tecnológica DDTI UTN.

3.3.1.3. Acceso Inalámbrico

La Facultad de Ingeniería en Ciencias Aplicadas (FICA), así como las demás facultades, edificios y dependencias del Campus Universitario El Olivo, usan los equipos de conexión y acceso inalámbricos descritos y detallados en las secciones anteriores; para permitir a los usuarios mediante sus dispositivos móviles o equipos portátiles, acceder a los recursos de red y navegar en la Internet, no sin antes validar y otorgar los permisos de acceso mediante las credenciales de usuarios respectivas que asigna la Dirección de Desarrollo Tecnológico e Informático (DDTI) en su Servidor Radius/OpenLDAP el cual se conecta a la Corporación Ecuatoriana para el Desarrollo de la Investigación y la Academia (CEDIA) y permite el acceso a la Red Inalámbrica Avanzada EDUROAM.

- **CEDIA**

Corporación Ecuatoriana para el Desarrollo de la Investigación y la Academia o también conocida como la Red Nacional de Investigación y Educación Ecuatoriana (RNIE), promueve la exploración e investigación de proyectos innovadores que vinculan a instituciones ecuatorianas; relacionando investigadores, docentes y estudiantes con proyectos y concursos de desarrollo científico, para generar un crecimiento académico constante de las instituciones que la conforman las cuales son universidades, escuelas politécnicas, institutos de investigación y colegios los que se muestran en la Figura 20 (CEDIA, 2022).

FIGURA 20

INSTITUCIONES ACADÉMICAS QUE FORMAN PARTE DE LA RED CEDIA

Nota: Tomado de la página web de *CEDIA* (CEDIA, 2022).

- **EDUROAM**

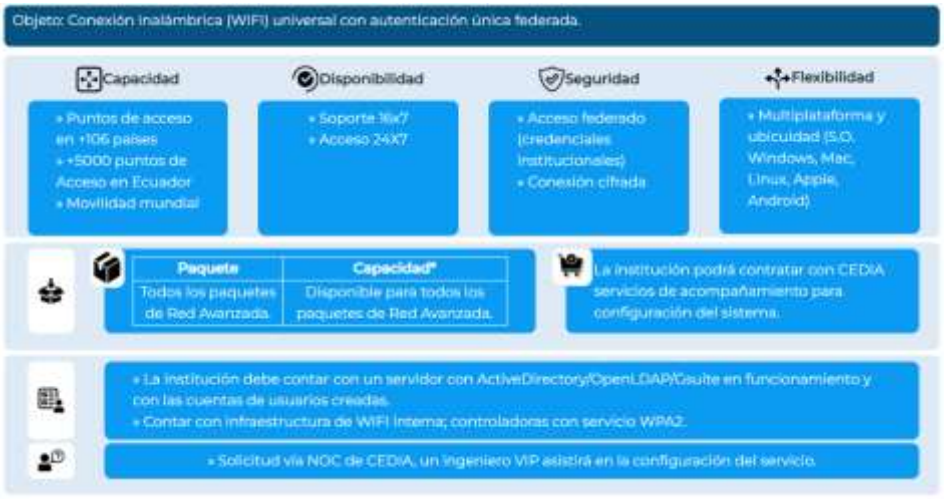
EDUcation-ROAMing o EDUROAM se denomina al servicio WiFi académico mundial de movilidad, destinado al uso de la comunidad académica y de investigación; permitiendo la conectividad a Internet y a la Red Avanzada dentro de los diferentes campus de las instituciones participantes tanto a nivel nacional como al rededor del mundo. La red federada está disponible en 89 países y 5331 instituciones al rededor del mundo, en Ecuador está disponible en 121 campus con más de 3000 puntos de acceso (CEDIA, 2022).

Para acceder a este servicio se necesitan las credenciales de validación, en este caso un correo electrónico y una clave institucional, los servidores y puntos de acceso se configuran de tal manera que cuando el usuario se encuentra en otras instituciones que cuentan con una red Eduroam, el dispositivo móvil o equipo portátil crea un túnel hacia la institución donde se crearon

las credenciales para verificarlas antes de la conexión; posteriormente la navegación se realiza por medio de la institución a la que se ha conectado el usuario como se muestra en la Figura 21.

FIGURA 21

FUNDAMENTOS RED EDUROAM



Nota: La imagen muestra los elementos y características de funcionamiento de acceso a la red EDUROAM que se explican mejor en el siguiente párrafo. Tomado de la página web de *CEDIA* (CEDIA, 2022).

- **Servidor Radius/LDAP**

En la Universidad Técnica del Norte el servicio federado EDUROAM, permite la conectividad de los usuarios internos como son personal administrativo, personal docente y estudiantes a Internet. El servidor Radius-UTN como se observa en la Figura 22, posee un enlace hacia un directorio LDAP que permite clasificar y administrar en forma ordenada a los usuarios dentro de la red (Administrativos, Facultades, Docentes y Estudiantes por Facultad y Carrera). El servicio federado posee una alta seguridad gracias a que éste utiliza certificados digitales al momento de establecer una conexión. El servidor Radius es un servidor AAA; permite la autenticación por medio del correo institucional y contraseña, la autorización permitiendo acceder a una dirección IP del servicio, y la contabilidad, cuando el servicio responde a la petición realizada

por el usuario. El servicio EDUROAM se encuentra dentro de la DMZ de la Universidad Técnica del Norte, permitiendo aislar a éste del resto de la red evitando ataques (GARRIDO, 2018).

FIGURA 22

CAPTURA DE LA INTERFAZ DE SERVIDOR RADIUS-UTN

Nota: En la imagen se observa la configuración de puerto y dirección IP del Servidor AAA Radius/LDAP. Fuente: Responsable Infraestructura Tecnológica DDTI UTN.

- **Proceso de conexión**

Están autorizados para acceder a los servicios inalámbricos de red propagados en el campus o en cualquier otro lugar donde se encuentre la red Eduroam a nivel nacional o internacional los estudiantes, docentes, funcionarios administrativos, empleados y trabajadores de la Universidad Técnica del Norte; para el acceso se requiere el uso de las credenciales respectivas de cada usuarios las cuales son: usuario (dirección de correo institucional) y contraseña (se encuentra en la plataforma de portafolios UTN); además es necesario que para la conexión de dispositivos se haga uso de la herramienta “eduroamCAT (Configuration Assitant Tool)”, el cual es un software de

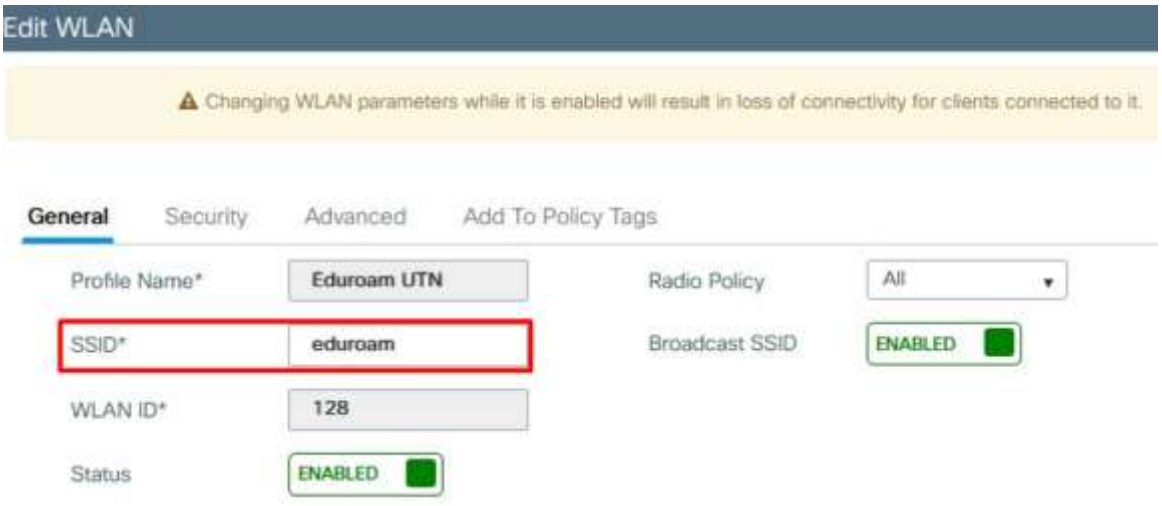
seguridad usado para evitar problemas de desconexión e intermitencias al realizar una conexión errónea de dispositivos (UTN, 2022), estos procedimientos se detallan en el Manual de instalación EDUROAM que consta en el Anexo 1, para los diferentes sistemas operativos y dispositivos.

- **SSID Propagado**

Los Puntos de Acceso inalámbricos que se distribuyen en el campus universitario, se encargan de propagar la señal de acceso "eduroam" como se muestra en la interfaz de configuración en la Figura 23, la cual permite la validación de credenciales y posterior conexión y acceso a la red inalámbrica, siguiendo los pasos descritos en el Anexo 1; en la Figura 24 se muestra una captura de las redes inalámbricas realizada en las instalaciones de la Facultad de Ingeniería en Ciencias Aplicadas, donde se observa que el SSID "eduroam" es propagado por varios puntos de acceso en diferentes canales de transmisión.

FIGURA 23

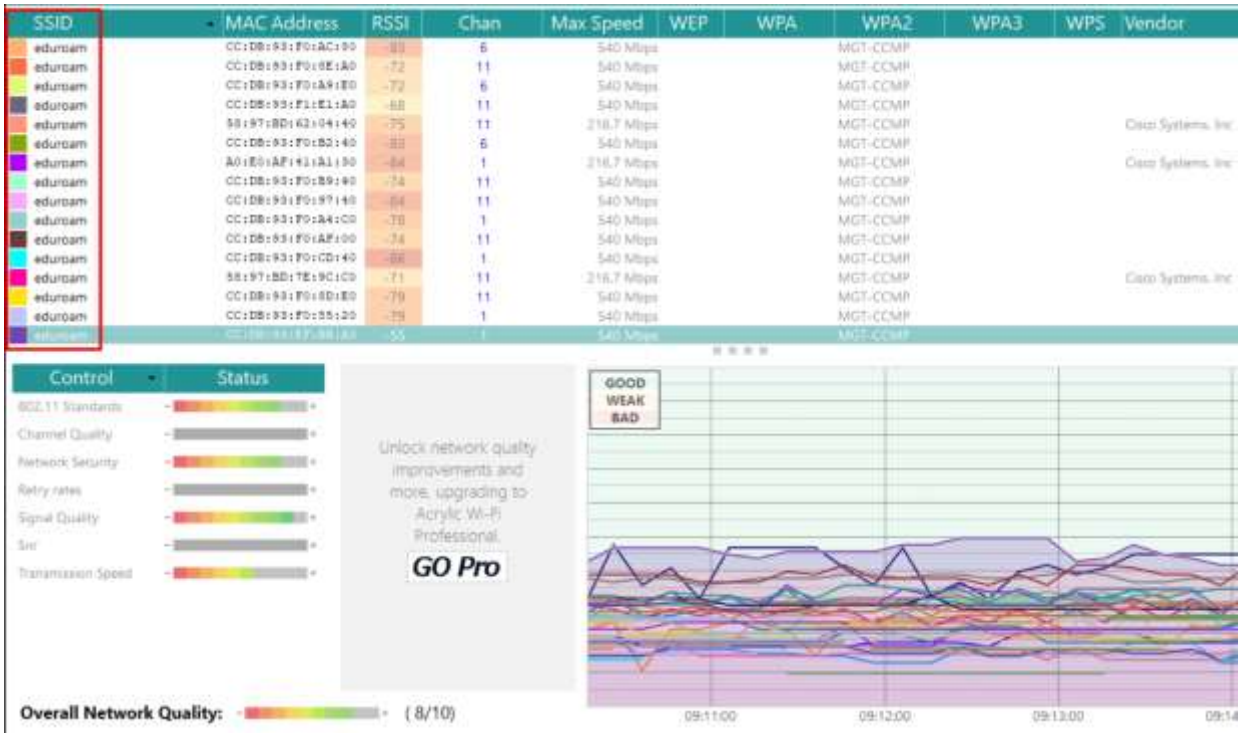
CONFIGURACIÓN DE SSID PROPAGADO PARA LE RED EDUROAM



Fuente: Responsable Infraestructura Tecnológica DDTI UTN.

FIGURA 24

SSID PROPAGADO PARA LE RED EDUROAM EN EL EDIFICIO DE LA FICA



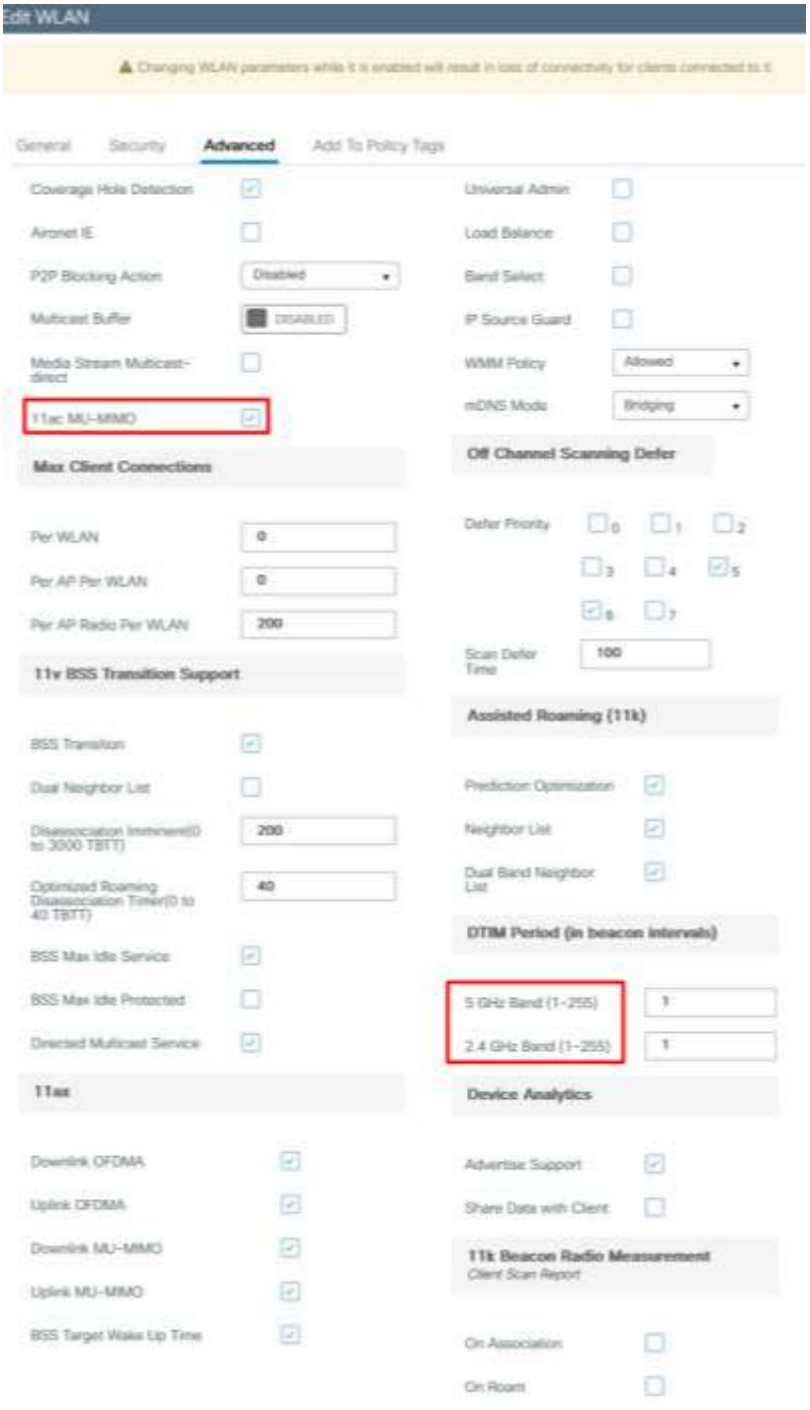
Nota: La imagen muestra la red propagada "eduroam", la dirección MAC de los diferentes puntos de acceso, los canales y velocidad de transmisión.

3.3.2. Análisis de Situación Actual

De acuerdo con la información solicitada y obtenida del DDTI mediante el Anexo 2.1 y la elaboración y aprobación del cuestionario del Anexo 2.2, se obtuvo los datos que se observan en las imágenes, figuras y capturas expuestas a continuación, así como también de las respuestas del Anexo 2.3; de manera que con ayuda de todos estos datos y detalles adicionales que se observa en las Figuras 25 y 26 se logró definir, las características en base a las especificaciones técnicas de los equipos involucrados en el acceso inalámbrico de red de la FICA y que se aplica también para el entorno del Campus UTN El Olivo el que se describe y resume en la Tabla 5; estableciendo y sustentando el contexto actual para el desarrollo del presente proyecto.

FIGURA 25

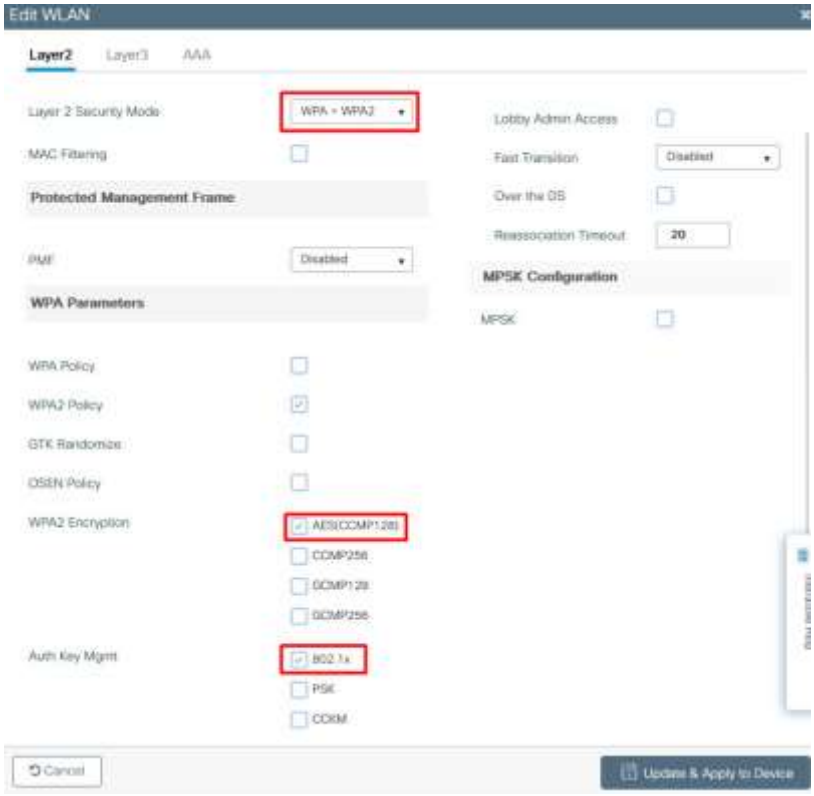
CONFIGURACIÓN DE VERSIÓN DE PROTOCOLO Y CANALES INALÁMBRICOS



Nota: La imagen muestra la versión de estándar IEEE 802.11 que soportan los equipos inalámbricos y las bandas en los que propagan la señal. Fuente: Responsable Infraestructura Tecnológica DDTI UTN.

FIGURA 26

CAPTURA DE INTERFAZ DE CONFIGURACIÓN DE SEGURIDAD Y PROTOCOLO INALÁMBRICO



Nota: La imagen muestra el modo de seguridad WPA/WPA2, la encriptación AES(CCMP128) y autenticación bajo protocolo IEEE 802.1x. Fuente Responsable Infraestructura Tecnológica DDTI UTN.

TABLA 5

ANÁLISIS DE SITUACIÓN ACTUAL EN LA INFRAESTRUCTURA INALÁMBRICA DE RED DE LA FICA

EQUIPO DE ACCESO	ESTÁNDAR IEEE	PROTOCOLO	SEGURIDAD	SSID	CIFRADO
Cisco C9115AXI-A	802.11ac 802.11ax	Radius/LDAP Federado CEDIA	WPA2/802.1x	"eduroam"	AES-CCMP

Nota: La tabla muestra la tecnología usada en la infraestructura de red inalámbrica de la Facultad de Ingeniería en Ciencias Aplicadas.

3.3.3. Motivación, Beneficio y Limitantes

Como se explica ampliamente en la Sección 1.5, la presente investigación busca desarrollar un mecanismo de autenticación inalámbrico alternativo al que se usa actualmente en la FICA; el cual como ya se ha descrito y que, emplea un servidor centralizado (Radius/LDAP) que se encarga de validar las credenciales de usuario. De tal manera se ha planteado diseñar un mecanismo de autenticación que aproveche los beneficios descentralizados que brinda la tecnología de Cadena de Bloques.

La investigación y desarrollo de soluciones a partir de la explotación de las características y ventajas que aporta la tecnología detrás del entendimiento y combinación de los modelos descentralizados, algoritmos de consenso, pruebas de trabajo, generación de hash, criptografía y validación de claves, forman las bases robustas para una emergente tecnología enfocada en la seguridad distribuida, donde la carga de garantizar la Confidencialidad, Integridad y Disponibilidad recae en todos los nodos que forman parte de la red, los cuales se encargan de validar mutuamente las transacciones realizadas y llevar un registro actualizado e inmutable; elementos y peculiaridades que se espera integrar y demostrar en la realización presente proyecto.

En cuanto a los inconvenientes que se pueden presentar al adaptar un mecanismo de autenticación inalámbrica con las bases tecnológicas de la Cadena de Bloques, surgen restricciones relacionadas al hardware, ya que la configuración de nodos, minería y generación de bloques requiere gran cantidad de recursos de procesamiento y memoria, condicionando el desarrollo y diseño en un entorno real, por lo que se hace necesario crear un entorno controlado generando una cadena de bloques privada y limitada a pruebas muy específicas.

Además, también se debe considerar y mencionar los impedimentos de uso de versiones de software específicas, ya sean sistemas operativos, compiladores e interfaces de lenguajes de

programación; sin mencionar la dificultad intrínseca de adaptar en un todo este conjunto de tecnologías y protocolos para desarrollar el mecanismo de autenticación.

3.3.4. Establecimiento de Requerimientos

Considerando los lineamientos establecidos por el estándar ISO/IEC/IEEE 29148, el cual especifica los procesos y productos que se debe tomar en cuenta al momento de desarrollar un proyecto en ingeniería y siendo los Satakeholders herramientas descriptivas de los elementos o los individuos críticos en el proceso y cumplimiento de los objetivos propuestos para la realización de este proyecto, estos son representados en la Tabla 6.

TABLA 6

ACTORES FUNDAMENTALES EN EL DESARROLLO DEL PROYECTO

N°	Actor	Cargo
1	Galo Mauricio Beltrán Manosalvas	Tesista
2	Msc. Fabián Cuzme Rodríguez	Director
3	Msc. Mauricio Domínguez Limaico	Asesor
4	DDTI UTN – Red Inalámbrica FICA	Beneficiario

3.3.4.1. Nomenclatura de Requerimientos

El estándar define que se deben establecer requerimientos que influyen directamente en el ciclo de vida al diseñar un sistema o software, los cuales describan los elementos, necesidades y condiciones a tomar en cuenta a la hora de seleccionar definir y elegir equipos y herramientas de hardware y software; cuya nomenclatura se muestra en Tabla 7.

TABLA 7

NOMENCLATURA DE REQUERIMIENTOS

Requerimiento	Abreviatura
Stakeholders	STSR
Sistema	SYSR
Arquitectura	SRSH

Adicional, para el correcto seguimiento y cumplimento del estándar se deben asignar y establecer prioridades a cada uno de los requerimientos, la cuales determinan su importancia en base a criterios de usuario, rendimiento, diseño e implementación; las cuales se describen como:

- **Prioridad Alta:** Define el cumplimiento crítico de un requerimiento en el desarrollo de un proyecto, afectando la funcionalidad del proyecto y cumplimiento de los objetivos propuestos.

- **Prioridad Media:** El cumplimiento o implementación de estos requerimientos no es necesariamente indispensable, ya que pueden omitirse o no; siempre y cuando no afecten la funcionalidad del proyecto y el cumplimento de sus objetivos.

- **Prioridad Baja:** Los requerimientos de prioridad baja no afectan la funcionalidad del proyecto o influyen en el desarrollo, pueden complementar a requerimientos de mayor prioridad, pero su implementación no es necesaria.

3.3.4.2. Requerimientos de Stakeholders

Como se detalla en la Tabla 8, los requerimientos de Stakeholders (STSR) se establecen para definir los requisitos del sistema en relación con los interesados y actores involucrados en el desarrollo del proyecto, mediante la limitación y evaluación de requerimientos operacionales y de

usuario; estos se determinan en base a la información de la Sección 3.3.1 obtenida con los Anexo 1 y Anexo 2; así como del análisis bibliográfico de información realizada en el Anexo 3.

TABLA 8

REQUERIMIENTOS DE STAKEHOLDERS

STSR				
REQUERIMIENTOS OPERACIONALES				
#	**REQUERIMIENTO**	**PRIORIDAD**		
		Alta	**Media**	**Baja**
STSR1	Se implementará un mecanismo de autenticación basado en la tecnología de cadena de bloques en la red inalámbrica de la FICA.			X
STSR2	El mecanismo de autenticación propuesto, basado en la tecnología de cadena de bloques debe usar los equipos e infraestructura inalámbrica desplegado en la FICA.		X	
STSR3	El mecanismo de autenticación propuesto, basado en la tecnología de cadena de bloques debe permitir que los usuarios se conecten a la infraestructura inalámbrica de red de la FICA.		X	
STSR4	El mecanismo de autenticación propuesto, basado en la tecnología de cadena de bloques debe interactuar con el SSID "eduroam" y/o el servidor Radius/LDAP del DDTI.			X
STSR5	Se creará un entorno inalámbrico de pruebas para el mecanismo de autenticación propuesto, basado en la tecnología de cadena de bloques en la FICA.	X		
STSR6	El SSID de conexión y acceso al entorno inalámbrico del entorno de pruebas en la FICA, debe estar disponible para la detección de los dispositivos de los usuarios.	X		

STSR7	El mecanismo de autenticación propuesto, basado en la tecnología de cadena de bloques debe interactuar con el estándar 802.11 para permitir el acceso y conexión WiFi al entorno inalámbrico de pruebas en la FICA.	X		
STSR8	Los usuarios podrán acceder a Internet dentro del entorno de pruebas desplegado para mecanismo de autenticación propuesto, basado en la tecnología de cadena de bloques.			X
STSR9	Se empleará herramientas de simulación para desarrollar el mecanismo de autenticación propuesto, basado en la tecnología de cadena de bloques en el entorno inalámbrico de pruebas en la FICA.	X		
STSR10	La cadena de bloques validara la conexión y acceso de usuarios al entorno inalámbrico de pruebas en la FICA.	X		

REQUERIMIENTOS DE USUARIO

#	REQUERIMIENTO	PRIORIDAD		
		Alta	**Media**	**Baja**
STSR11	Los usuarios podrán conectare red inalámbrica de la FICA, usando el mecanismo de autenticación propuesto, basado en la tecnología de cadena de bloques.			X
STSR12	Los usuarios podrán conectarse al entorno inalámbrico de pruebas en la FICA, usando el mecanismo de autenticación propuesto, basado en la tecnología de cadena de bloques.	X		
STSR13	Los usuarios deberán disponer de credenciales de conexión para poder conectarse al entorno inalámbrico de pruebas en la FICA, mediante el mecanismo de autenticación propuesto, basado en la tecnología de cadena de bloques.		X	
STSR14	Los usuarios podrán conectarse al entorno inalámbrico de pruebas en la FICA, usando el mecanismo de autenticación propuesto, basado en la tecnología de cadena de bloques; mediante una interfaz gráfica amigable.		X	

STSR15	Los usuarios podrán conectarse al entorno inalámbrico de pruebas en la FICA, usando el mecanismo de autenticación propuesto, basado en la tecnología de cadena de bloques; desde diferentes dispositivos y/o sistemas operativos	X

3.3.4.3. Requerimientos de Sistema

Aquí se definirán los requerimientos de sistema (SYSR) en base a requerimientos de interfaz, uso, performance, físicos, modos y estados; en la Tabla 9 se resumen las limitantes de puesta en marcha y los ámbitos a cumplir, que se relacionan con los requisitos de Stakeholders.

TABLA 9

REQUERIMIENTOS DE SISTEMA

SYSR				
REQUERIMIENTOS DE INTERFAZ				
#	**REQUERIMIENTO**	**PRIORIDAD**		
		Alta	**Media**	**Baja**
SYRS1	El mecanismo de autenticación propuesto debe interactuar con al menos un dispositivo inalámbrico de red (Puntos de Acceso), para permitir la conexión al entorno inalámbrico de pruebas en la FICA.	X		
SYRS2	El equipo servidor o instancia donde se levantará el mecanismo de autenticación propuesto debe permitir la validación y acceso a la conexión mediante una interfaz.	X		
SYRS3	El mecanismo de autenticación propuesto debe interactuar juntamente con los equipos de usuario, los de red y el servidor o instancia para brindar el acceso a la red del entorno inalámbrico de pruebas en la FICA.	X		

SYRS4	El uso y configuración de complementos o componentes adicionales para el mecanismo autenticación propuesto debe ser fácil de realizarse.		X	

REQUERIMIENTOS DE USO				
#	**REQUERIMIENTO**	**PRIORIDAD**		
		Alta	**Media**	**Baja**
SYRS5	El usuario debe ingresar sus credenciales para conectarse a la red del entorno inalámbrico de pruebas en la FICA, para el mecanismo de autenticación propuesto.		X	
SYRS6	El mecanismo de autenticación propuesto debe permitir la conexión y acceso a Internet de los dispositivos de usuario.	X		
SYRS7	La asignación de credenciales de usuario se realizará de manera previa para el mecanismo de autenticación propuesto.		X	
SYRS8	El mecanismo de autenticación propuesto debe interactuar con el SSID configurado previamente para la conexión de usuarios.	X		

REQUERIMIENTOS DE PERFORMANCE				
#	**REQUERIMIENTO**	**PRIORIDAD**		
		Alta	**Media**	**Baja**
SYRS9	El mecanismo de autenticación propuesto debe permitir la conexión inalámbrica mediante un SSID propagado en el entorno inalámbrico de pruebas en la FICA.	X		
SYRS10	El tiempo empleado en el proceso de autenticación mediante el mecanismo de autenticación propuesto no debe ser prolongado.		X	
SYRS11	El mecanismo de autenticación propuesto debe levantarse en un equipo servidor o instancia virtual dedicada.		X	

REQUERIMIENTOS FÍSICOS				
#	**REQUERIMIENTO**	**PRIORIDAD**		
		Alta	**Media**	**Baja**

SYRS12	El punto de acceso usado para el mecanismo de autenticación propuesto debe ubicarse estratégicamente para que propague la señal inalámbrica.	X		
SYRS13	La ubicación del servidor o instancia donde se configurará el mecanismo debe ser estratégica y segura.		X	
REQUERIMIENTOS DE MODOS Y ESTADOS				
#	**REQUERIMIENTO**	**PRIORIDAD**		
		Alta	**Media**	**Baja**
SYRS14	El mecanismo de autenticación propuesto y el SSID configurado deben estar disponibles en un entorno controlado para la conexión de usuarios.	X		

3.3.4.4. Requerimientos de Arquitectura

Respecto a los requerimientos de arquitectura (SRSH), aquí se plantean y referencian las especificaciones de uso de hardware y software, así como también los requisitos lógicos y de diseño, y que están detallados en la Tabla 10.

TABLA 10

REQUERIMIENTOS DE ARQUITECTURA

SRSH				
REQUERIMIENTOS DE DISEÑO				
#	**REQUERIMIENTO**	**PRIORIDAD**		
		Alta	**Media**	**Baja**
SRSH1	El mecanismo de autenticación propuesto autorizará la conexión de usuarios y equipos a la red del entorno inalámbrico de pruebas en la FICA.	X		

SRSH2	El mecanismo de autenticación propuesto gestionara los equipos y usuarios conectados a la red del entorno inalámbrico de pruebas en la FICA.			X
SRSH3	El mecanismo de autenticación propuesto auditara los equipos y usuarios conectados a la red del entorno inalámbrico de pruebas en la FICA.			X
SRSH4	El mecanismo de autenticación propuesto permitirá administrar y configurar credenciales de usuarios.		X	

REQUERIMIENTOS LÓGICOS

#	REQUERIMIENTO	PRIORIDAD		
		Alta	**Media**	**Baja**
SRSH5	El entorno inalámbrico de pruebas configurado para el mecanismo de autenticación propuesto deberá ser propagada y estar disponible en la FICA.	X		
SRSH6	De ser necesario, se configurará extensiones y complementos en los dispositivos para permitir la conexión al entorno inalámbrico de pruebas en la FICA.		X	
SRSH7	Se requerirá de una aplicación o interfaz de para la conexión de dispositivos a la red del entorno inalámbrico de pruebas en la FICA.		X	

REQUERIMIENTOS DE HARDWARE

#	REQUERIMIENTO	PRIORIDAD		
		Alta	**Media**	**Baja**
SRSH8	El equipo para configurar el mecanismo de autenticación propuesto debe disponer de memoria de alto procesamiento.	X		
SRSH9	El equipo para configurar el mecanismo de autenticación propuesto debe disponer de un procesador de alto rendimiento.	X		
SRSH10	El equipo para configurar el mecanismo de autenticación propuesto debe disponer de una tarjeta gráfica.			X

SRSH11	El equipo para configurar el mecanismo de autenticación propuesto debe disponer de gran espacio de almacenamiento.			X
SRSH12	El equipo para configurar el mecanismo de autenticación propuesto debe disponer de interfaz de conexión de red de alta velocidad.	X		
SRSH13	El punto de acceso inalámbrico configurado con el mecanismo de autenticación propuesto deberá ser compatible con el estándar IEEE 802.11 y sus versiones b, n, ac y ax.	X		
SRSH14	El punto de acceso inalámbrico configurado trabajara en las bandas de 2.4 y/o 5 GHz.	X		
	REQUERIMIENTOS DE SOFTWARE			
#	**REQUERIMIENTO**	**PRIORIDAD**		
		Alta	**Media**	**Baja**
SRSH15	El mecanismo de autenticación propuesto debe ser compatible con sistema operativo Windows o Linux.		X	
SRSH16	Compatibilidad del sistema operativo con lenguajes de programación de código abierto.	X		
SRSH17	El software deberá permitir compilar y ejecutar librerías de la tecnología de cadena de bloques.	X		
SRSH18	Herramientas de software compatibles para el diseño de la interfaz gráfica.		X	

3.3.5. Elección de Hardware y Software

La elección de hardware y software para el desarrollo del mecanismo de autentificación propuesto basado en la tecnología de cadena de bloques, se realiza de acuerdo con el análisis de los requerimientos de Stakeholders, Sistema y Arquitectura establecidos; evaluados en base a lo

establecido en el Anexo 3 y valorando las opciones de selección de acuerdo con la obtención de mayor puntaje en base a la ponderación de cumplimiento “1” o incumplimiento “0”.

3.3.5.1. Hardware

La selección de hardware se hará principalmente tomando en cuenta los requerimientos establecidos en la Tabla 10, también se incluyen otros que se consideran en la infraestructura física para el desarrollo de este proyecto y sus necesidades. En la Tabla 11 se muestra la valoración de cada requerimiento para la elección del equipo de hardware a partir de tres opciones planteadas.

TABLA 11

ELECCIÓN DE HARDWARE

Equipo de Hardware	Requerimientos											Valor
	STSR2	STSR5	STSR9	SYRS1	SYRS11	SYRS13	SRSH8	SRSH9	SRSH10	SRSH11	SRSH12	
Laptop Dell G15 5510: Intel Core i7-10870H CPU, RAM 32GB, NVIDIA GeForce RTX 3050, 512GB SSD, Adaptador Realtek PCIe GbE, Windows 10 x64	0	1	0	0	0	0	1	1	1	1	1	**6**
Máquina Virtual: Intel Core i7-10870H Anfitrión, RAM 10GB, Almacenamiento 50GB, Adaptador Intel PRO/1000 MT, Windows Server 2016 x64	1	1	1	1	1	1	1	1	0	1	1	**10**
Instancia Virtual: CPU 2 Sockets 8 Core, RAM16 GB, Almacenamiento 512GB, Adaptador Intel E1000 MT, SO: Windows Server 2016 x64	1	1	1	1	1	1	1	1	0	1	1	**10**
1 = Cumple 0 = No Cumple												

La elección de hardware idónea, según la valoración de los requerimientos; es la opción de levantar una Instancia Virtual, para el despliegue del mecanismo de autenticación propuesto, con los requisitos mínimos y características técnicas que se muestran en la Tabla 12.

TABLA 12

REQUISITOS MÍNIMOS DE HARDWARE

Requisitos Mínimos de Instancia Virtual	
Procesador	De alto rendimiento x64, x86 o ARM de dos, cuatro núcleos o superior
Memoria RAM	Al menos 8 GB o 16 GB
Almacenamiento	Al menos 50 GB HDD o SDD
GPU	Tarjeta gráfica integrada de 2 GB
Conexión	Adaptador de red Gigabit Ethernet y Conexión estable a Internet

Para los equipos de acceso inalámbrico al entorno inalámbrico de pruebas en la FICA, que se usarán para el despliegue del mecanismo propuesto; se consideran los modelos de equipos disponibles en los laboratorios y/o los usados en la infraestructura de la facultad, los cuales se valoran en la Tabla 13.

TABLA 13

ELECCIÓN DE EQUIPOS DE ACCESO INALÁMBRICO

Equipos de Punto de Acceso	**Requerimientos**									**Valor**
	STSR2	**STSR5**	**STSR7**	**SYRS1**	**SYRS3**	**SYRS9**	**SYRS12**	**SRSH13**	**SRSH14**	
Mikrotik RB931Ui-2HnD	0	1	1	1	1	1	1	0	1	**7**
Mikrotik RBwAP2nD	0	1	1	1	1	1	1	0	1	**7**
Cisco C9115AXI-A	1	0	1	1	0	0	1	1	1	**6**
1 = Cumple 0 = No Cumple										

En la Tabla 14, se muestran las características principales de los equipos de punto de acceso inalámbrico, que cumplen con la mayor cantidad de requerimientos establecidos y considerados en el desarrollo de este proyecto.

TABLA 14

CARACTERÍSTICAS EQUIPOS DE PUNTO DE ACCESO INALÁMBRICOS

Equipos para Punto de Acceso Inalámbrico	
Mikrotik RB931Ui-2HnD	Frecuencia nominal de la CPU: 600 MHz Núcleos de CPU: 1 RAM: 128 MB Wi-Fi 802.11b/g/n de 2,4 GHz Velocidad Wifi: 300Mbps
Mikrotik RBwAP2nD	CPU frecuencia nominal: 650 MHz CPU número de núcleos: 1 RAM: 64 MB Estándares inalámbricos: 802.11b/g/n Velocidad Wifi: 300Mbps

3.3.5.2. Software

La selección de software se realizará principalmente en base a los requerimientos establecidos en la Tabla 10, considerando algunas alternativas de codificación y programación requeridas para el despliegue, desarrollo y uso de un entorno de cadena de bloques; así como las herramientas, complementos y bibliotecas usadas en los procesos propios que intervienen al momento de la validación e interacción con los nodos y usuarios que forman parte de una red de cadena de bloques; también se tomaran en cuenta otros requerimientos establecidos que incidan y se relacionen al uso de software en general como son; el sistema operativo que se usará para desplegar el mecanismo de autenticación, el cliente Ethereum que es el software que permite la interacción con la red Ethereum, creando los contratos inteligentes, ejecución de transacciones y

validación de bloques, también se considera el lenguaje de programación que se empleará para la configuración de los contratos inteligentes y algoritmos de consenso, junto con el entorno de desarrollo integrado y las librerías Ethereum y de programación, por otra parte también se analizaran las plataformas de desarrollo de interfaz gráfica y los complementos de validación de credenciales que usan las cadenas de bloques como son la cripto billeteras.

En la Tabla 15 se enlistan y ponderan las opciones de software a usar para el levantamiento y desarrollo del mecanismo de autenticación propuesto.

TABLA 15

ELECCIÓN DE SOFTWARE

Software	Requerimientos											Valor
	STSR5	**STSR9**	**STSR14**	**SYRS2**	**SYRS4**	**SYRS11**	**SRSH7**	**SRSH15**	**SRSH16**	**SRSH17**	**SRSH18**	
Sistema Operativo												
Windows Server	1	1	1	0	0	1	0	1	1	1	1	**8**
Linux	1	1	1	0	0	1	0	1	1	1	1	**8**
macOS	0	0	0	0	0	0	0	0	1	1	1	**3**
Cliente Ethereum												
Geth	1	1	1	1	1	1	0	1	1	1	1	**10**
Parity	0	0	1	0	0	0	0	0	1	1	1	**4**
Besu	1	0	0	0	0	0	0	0	1	1	1	**4**
Ganache	1	1	1	1	1	1	1	1	1	1	1	**11**
Lenguaje de Programación												
Visual Studio Code	1	1	1	1	1	1	1	1	1	1	1	**11**
Atom	0	0	1	0	1	0	0	0	1	1	1	**4**
Solidity	1	1	1	1	1	1	1	1	1	1	1	**11**
Python	1	1	1	1	1	1	1	1	1	1	1	**11**
Sublime Text	0	0	1	0	0	0	0	0	1	1	1	**4**

Entorno de Desarrollo Integrado (IDE)												
Truffle	1	1	1	1	1	1	1	1	1	1	1	**11**
Remix	1	1	1	1	0	0	1	1	1	1	1	**9**
Node.js	1	1	1	1	1	1	1	1	1	1	1	**11**
Embark	0	0	1	0	0	0	0	0	1	1	1	**4**
Librerías Ethereum												
Web3.js	1	1	1	1	1	1	1	1	1	1	1	**11**
Ethers.js	1	1	1	1	1	1	1	1	1	1	1	**11**
Librerías de Programación												
JSS	1	1	1	1	1	1	1	1	1	1	1	**11**
CSS	1	1	1	1	1	1	1	1	1	1	1	**11**
HTML	1	1	1	1	1	1	1	1	1	1	1	**11**
Entorno de Desarrollo Integrado Gráfico												
Boostrap/TypeScript	1	1	1	1	1	1	1	1	1	1	1	**11**
Vite React	1	1	1	1	1	1	1	1	1	1	1	**11**
Cripto billetera												
Bitcoin Wallet	0	0	1	1	0	0	1	1	0	1	0	**5**
METAMASK	1	1	1	1	1	1	1	1	1	1	1	**11**
1 = Cumple 0 = No Cumple												

Conforme a los puntajes obtenidos en la Tabla 15 y el cumplimiento de los parámetros de requerimientos de la Tabla 10; las herramientas de software escogidas para el desarrollo de este proyecto son; Distro Linux y/o Windows Server como sistema operativo ya que facilitan el uso e instalación de aplicaciones para el despliegue de tecnología de cadena de bloques. Entre las herramientas consideradas para el levantamiento de cadena de bloques se establece; Geth y Ganache como clientes Ethereum, Visual Studio Code, Solidity y Python como lenguajes de programación, Truffle y Node.js para el entorno de desarrollo integrado (IDE) y Web3.js o Ethers.js como librería Ethereum; todos estos componentes mencionados anteriormente se consideran por su compatibilidad y facilidad de interacción al momento de levantar y configurar

una cadena de boques. Finalmente, para el desarrollo de la interfaz gráfica se seleccionó las librerías de programación JSS, CSS, HTML y Boostrap en conjunto con Vite React y TypeScript ya que se emplean en el diseño de entornos e interfaces web sencillas y amigables con el usuario, y que de ser necesario en conjunto con la cripto billetera Metamask se puede usar esta como una extensión en varios navegadores web; la Tabla 16 muestra un resumen de las herramientas de software seleccionadas que cumplen con los requerimientos establecidos y donde se justifica su uso para el desarrollo del mecanismo de autenticación propuesto.

TABLA 16

HERRAMIENTAS DE SOFTWARE

Componentes de Software Seleccionados		Justificación
Sistema Operativo	Distro Linux y/o Windows Server	Compatibilidad con aplicaciones e instalación de herramientas de tecnología de Cadena de Bloques.
Cliente Ethereum	Geth y Ganache	Herramientas y complementos compatibles entre sí, necesarios para el levantamiento de un entorno de cadena de bloques.
Lenguaje de programación	Visual Studio Code, Solidity, Python	
Entorno de Desarrollo Integrado (IDE)	Truffle, Node.js	
Librerías Ethereum	Web3.js. Ethers.js	
Librerías de programación	JSS, CSS, HTML	Diseño de entornos e interfaces web, en conjunto con la extensión de la cripto billetera se puede usar varios navegadores web
Interfaz Grafica	Boostrap/ TypeScript, Vite React	
Cripto billetera	Metamask	

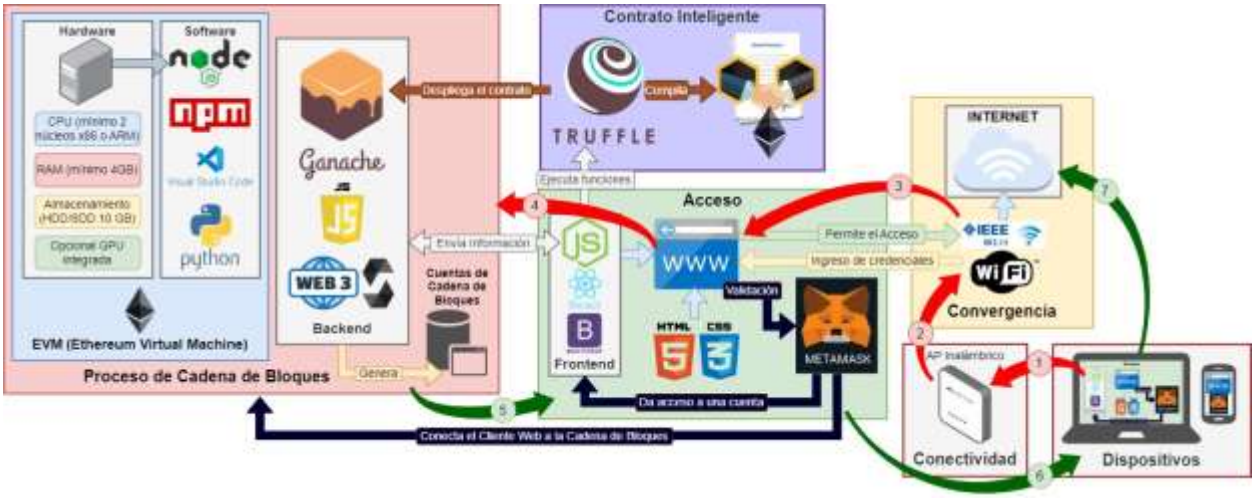
3.4. Arquitectura de la solución propuesta

Para el desarrollo del mecanismo de autenticación inalámbrico propuesto, se planteó usar la tecnología de cadena de bloques, diseñando un entorno virtual que brinde las características presentes en un entorno real y que incluye el levantamiento de un cadena de bloques como se

muestra en la Figura 27; donde se pueden apreciar los componentes de hardware y software, además de las herramientas y complementos necesarios para generar una cadena de bloques y facilitar tareas de compilación de código y despliegue de aplicaciones necesarias como; contratos inteligentes, validación de cuentas y transacciones así como para el levantamiento de una interfaz de conexión y acceso de dispositivos inalámbricos; que permita la interacción o convergencia tecnológica de una cadena de bloques con la tecnología de acceso inalámbrico WiFi y el estándar IEEE 802.11, para el acceso y conexión de usuarios a la infraestructura del entorno inalámbrico de pruebas en la facultad, haciendo uso de las claves y/o credenciales de validación establecidas y generadas al levantar la cadena de bloques. Así mismo, se puede visualizar la subdivisión por bloques para el proceso de desarrollo del mecanismo de autenticación propuesto a fin de adaptar la tecnología de cadena de bloques en el entorno inalámbrico de pruebas en la FICA.

FIGURA 27

ARQUITECTURA DE DISEÑO PARA EL MECANISMO DE AUTENTICACIÓN PROPUESTO



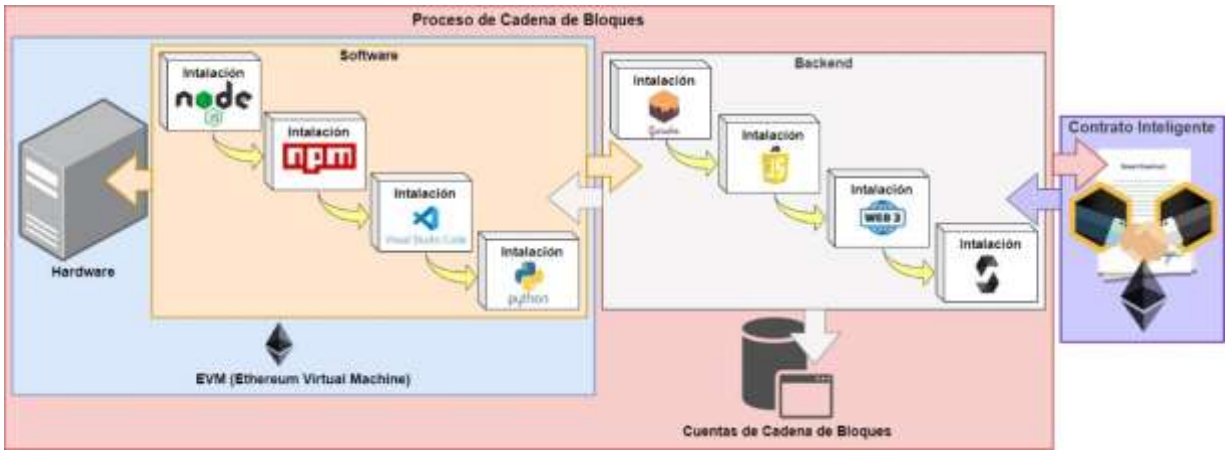
Nota: El diagrama muestra la distribución por bloques planteados para la arquitectura del mecanismo de autenticación propuesto y en donde se puede apreciar el proceso de conexión desde los dispositivos (1) de usuario hacia el AP inalámbrico (2), el acceso (3), la validación (4) con las flechas rojas y finalmente la autenticación (5) y el acceso (6) a Internet (7) con las flechas verdes.

3.4.1. Levantamiento de Cadena de Bloques

En esta primera etapa se levanta un entorno de pruebas con tecnología de cadena de bloques la cual se aprecia en la Figura 28, donde se muestran los componentes que permite desplegar una Máquina Virtual Ethereum (*Ethereum Virtual Machine o EVM)*; entorno para experimentar e inspeccionar la configuración, ejecución de comandos y transacciones que se ejecutan en los nodos que forman parte de una red de cadena de bloques, además proporciona cuentas Ether las cuales se pueden usar con una cripto billetera para realizar transacciones y observar su comportamiento (Ethereum, 2024); este software está disponible para distribuciones de sistemas operativos Linux y Windows.

FIGURA 28

DIAGRAMA DEL PROCESO DE LEVANTAMIENTO DE CADENA DE BLOQUES



Nota: El diagrama muestra los elementos de software necesarios para levantar un entorno EVM y a su vez el Backend involucrado en la creación de una Cadena de Bloques y las respectivas cuentas.

3.4.1.1. Máquina Virtual Ethereum o EVM

Una *Ethereum Virtual Machine* es un entorno de ejecución e interpretación de contratos inteligentes en la red Ethereum que permite la creación de aplicaciones descentralizadas; para levantar una EVM local para desarrollo y pruebas (Dwyer, 2024), hay que tomar en cuenta el

conjunto de requisitos que se describen en la Sección 3.3.5; donde se establecen las características de hardware junto con las herramientas y complementos de software para levantar una red de pruebas Ethereum.

Dentro de los componentes de software planteados en la Figura 28; se usa **Node.js** que permite desarrollar y ejecutar código en la cadena de bloques, empleado también en la creación de aplicaciones descentralizadas con JavaScript y contratos inteligentes (Cañar & Jara, 2022; Node.js, 2023), mientras que **NPM (Node Package Manager)** gestiona módulos, paquetes y bibliotecas para el desarrollo de aplicaciones en la cadena de bloques, proporcionando el uso de Web3.js o Ethers.js permitiendo crear contratos inteligentes, gestionar cuentas y transacciones (Cañar & Jara, 2022; npmjs, 2023); por otra parte **Visual Studio Code** es el entorno de desarrollo integrado (IDE) o editor de código, que soporta múltiples lenguajes y extensiones de programación, útiles en la depuración y manejo de contratos inteligentes; empleando **Python** como herramienta de desarrollo y prueba de scripts adicionales (Cañar & Jara, 2022; Jameson, 2023).

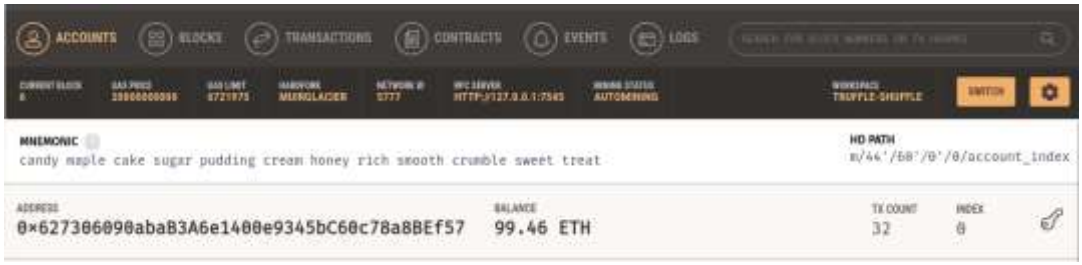
3.4.1.2. Cuentas de Cadena de Bloques

Las cuentas o direcciones en una cadena de bloques identifican a los participantes de la red permitiendo la interacción con los contratos inteligentes; cada dirección está asociada a una clave criptográfica única y estas pueden ser para cuentas de usuario que usan claves privadas para enviar, recibir tokens y criptomonedas; o para cuentas de contrato que se asocian a contratos inteligentes y se ejecutan cuando se realiza una transacción, usando claves públicas para el caso de las criptomonedas como Bitcoin o Ethereum. Las direcciones de una cadena de bloques suelen ser cadenas alfanuméricas generadas mediante algoritmos criptográficos; por ejemplo, la dirección de cadenas de bloques para Bitcoin (BTC) inicia con identificador con el número "1" o "3" mientras

que en Ethereum (ETH) la dirección comienza con "0x", como las que se facilitan en el entorno de pruebas Ganache y se muestra en la Figura 29 (OKX, 2023).

FIGURA 29

CUENTAS ETHEREUM DE PRUEBA PARA LA APLICACIÓN GANACHE



Nota: Tomado de la página web Truffle Suite (Kingsley, 2023).

3.4.1.3. Backend

El desarrollo de backend para este proyecto implica a la arquitectura de software y a las herramientas que se ocupan del procesamiento y almacenamiento de los datos subyacentes de la cadena de bloques; así como el despliegue e implementación de contratos inteligentes y aplicaciones que permitan interactuar con el frontend, asegurando la integridad de las operaciones y procesos que se realizan dentro del entorno de la cadena de bloques.

Dentro de los componentes de la arquitectura de software para el backend del mecanismo de autenticación propuesto se usó el entorno de pruebas **Ganache**, el cual permite desplegar una cadena de bloques Ethererum local, para realizar pruebas y depurar aplicaciones descentralizadas y contratos inteligentes, esta herramientas además proporciona un grupo de cuentas que permiten la interacción con la cadena bloques, dispone de una extensión para Visual Studio Code la cual permite manejar la cuentas, visualizar los logs de transacciones e inspeccionar el estado de la cadena de bloques (Kingsley, 2023; Lobanov, 2023); así como el uso de la tecnología **Web3** o **Ethers**, que hace referencia a las cadenas de bloques y donde se busca una web descentralizada

que permita la contribución directa de los usuarios al desarrollo técnico mediante mecanismos que regulan su interacción, por lo tanto, no necesitan una entidad centralizada que valide o autorice las interacciones (Cloudflare, 2023) y finalmente se emplea **Solidity** ya que a pesar de que Ethereum permite el desarrollo de contratos inteligentes en varios lenguajes, este es el más usado, siendo un lenguaje de alto nivel con una sintaxis similar a JavaScript (Soliditylang, 2023).

Por otra parte, respecto a las características de despliegue del backend, se considerarán; la **Validación y Consenso**, que permite la verificación de las transacciones y los bloques que se agregan a la cadena, ejecutando algoritmos de consenso como Prueba de Trabajo (Proof of Work) o Prueba de Participación (Proof of Stake); asegurando la integridad y la seguridad de la cadena de bloques (Qbit, 2024); también hay que proporcionar un mecanismo eficiente de **Almacenamiento de Datos** de la cadena; usando una base de datos distribuida o una estructura de almacenamiento específica diseñada para la cadena de bloques. Respecto a la recepción y procesando de las transacciones enviadas dentro de la cadena de bloques, la **Gestión de Transacciones** verifica su autenticidad, actualizando el estado de la cadena de bloques y propagando la información a los nodos de la red. De igual manera dentro la **Seguridad y Criptografía** se implementan los mecanismos criptográficos para garantizar la seguridad de la información y la privacidad de las transacciones; incluyendo la generación y verificación de firmas digitales, así como el manejo de claves criptográficas (Sáez, 2024).

3.4.2. Contrato Inteligente

En esta etapa se define las herramientas de uso y despliegue para el contrato inteligente, usando los componentes de software Web3 o Ethers en conjunto con un nodo Ethereum y mediante el compilador Solidity ejecuta el código dentro de un contrato inteligente, el cual no es más que un programa informático basado en cadena de bloques que ejecuta automáticamente acuerdos

cuando se cumplen condiciones predefinidas, controlando y permitiendo que se realicen las transacciones (Sáez, 2024). El entorno Truffle Framework ayuda a desarrollar, probar y desplegar contratos inteligentes en un entorno Ethereum, automatiza la compilación y pruebas para agilizar la creación de aplicaciones descentralizadas; al integrarse con editores como Visual Studio Code permite hacer uso de extensiones que potencian en cuanto a eficiencia en el desarrollo (Lobanov, 2023); adicionalmente la librería Truffle Contract permite interactuar y extraer información para este caso, desde el frontend de la aplicación con los datos que genera el contrato inteligente en formato Json; el rol y uso del contrato inteligente se muestra en diagrama de la Figura 30.

FIGURA 30

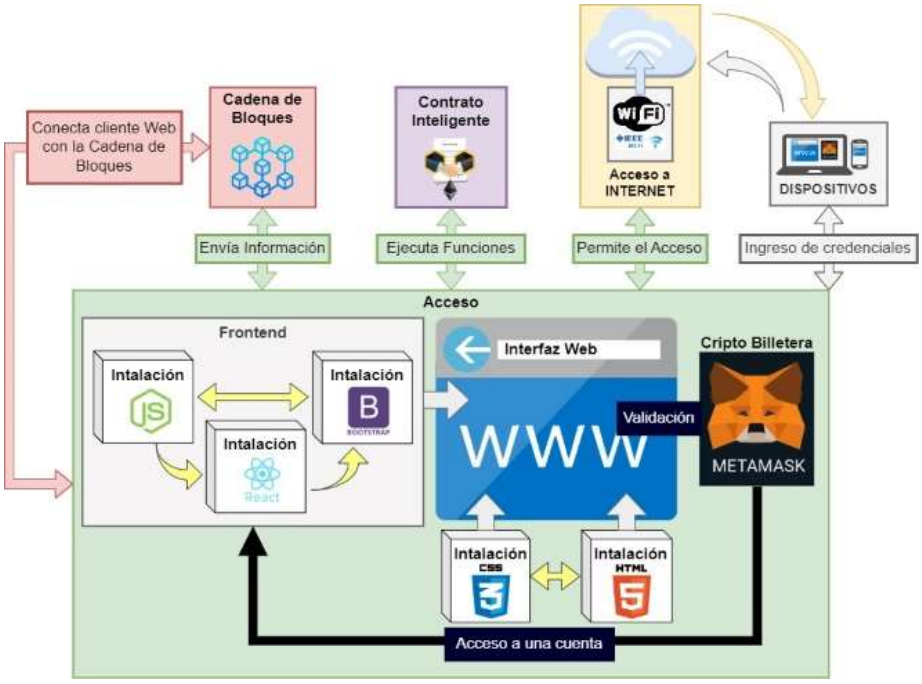
DIAGRAMA DE CONTRATO INTELIGENTE

3.4.3. Acceso

Para facilidad de los usuarios de la infraestructura inalámbrica de la red de pruebas, la gestión de conexión y acceso, se realizará mediante una interfaz web; que permita ingresar la credencial de usuario valida, donde la inmutabilidad de esta información permita la revisión de los registros del entorno de cadena de bloques; además admita el ingreso y el acceso de los dispositivos mediante el mecanismo de autenticación propuesto y que de ser necesario se complementará con una extensión o aplicación de billetera criptográfica como lo muestra la Figura 31.

FIGURA 31

DIAGRAMA DE ACCESO



3.4.3.1. Frontend

En el contexto de una cadena de bloques, el frontend se refiere a la interfaz de usuario (User Interface o UI) a través de la cual los usuarios interactúan con la cadena de bloques y realizan diversas operaciones. Aunque la cadena de bloques en sí misma es una tecnología subyacente y no tiene una interfaz de usuario directa, el frontend proporciona una capa de presentación y facilita la interacción con la cadena de bloques. El frontend de una cadena de bloques puede ser una aplicación web, una aplicación móvil u otro tipo de interfaz que permite a los usuarios ver y manipular los datos y las transacciones en la cadena de bloques.

Algunas de las funciones que puede proporcionar un frontend en una cadena de bloques incluyen; la **Visualización del Estado de la Cadena de Bloques** para mostrar información relevante sobre el estado actual, el número de bloques, las transacciones recientes, las direcciones

de cadena de bloques, saldos o permitir la **Creación y Envío de Transacciones** utilizando el frontend dentro de la cadena de bloques lo que implica el ingreso de direcciones de las partes involucradas y cualquier otra información requerida como parte del protocolo de la cadena de bloques; también se puede permitir la **Exploración de Bloques y Transacciones** en la cadena, ver bloques individuales y las transacciones contenidas en ellos.

Dentro de la **Gestión de Identidad y Claves**, el frontend proporciona funcionalidades para administrar claves criptográficas o credenciales de validación; esto puede incluir la conexión con la billetera, el respaldo de claves privadas y la importación y exportación de claves a otros dispositivos; así como la **Interacción con Contratos Inteligentes**, esto puede incluir llamar a métodos del contrato y el seguimiento de los eventos generados por el este.

Por otra parte, entre las herramientas consideradas para desplegar el entorno de frontend tenemos a la biblioteca JavaScritp **React** de código abierto, usada para el desarrollo de interfaces de usuario interactivas y de alto rendimiento para aplicaciones web, con un enfoque basado en componentes para gestionar el estado y el flujo de datos de la aplicación (Hunt, 2013); el uso del framework también de código abierto **Boostrap** para el diseño y desarrollo de sitios y aplicaciones web intuitivas para el usuario, que cuenta con herramientas y componentes predefinidos para ayudar a crear interfaces web atractivas y funcionales basadas en **HTML** (HyperText Markup Language), el lenguaje estándar usado para crear y diseñar documentos web, ya que estructura el contenido que se muestra y se representa en un navegador web y es fundamental para la inclusión de texto, imágenes, enlaces y demás recursos multimedia en un formato comprensible para los navegadores (ManzDev, 2023).

Junto al lenguaje de programación de diseño **CSS** (Cascading Style Sheets), que es utilizado para controlar la presentación y el aspecto visual de documentos web, permitiendo definir

como se muestran los elementos HTML como fuentes, colores, diseños y márgenes en una página (CEI, 2024), y con el lenguaje de programación de alto nivel **JavaScript** con su funcionalidad dinámica e interactiva para el desarrollo de entornos web con HTML (Arias, 2023); adicional a esto, el uso del framework **Vite** proporciona una experiencia de desarrollo ágil simplificando la creación de diseños web consistentes (Vite.dev, 2023) y adaptables a diferentes dispositivos y tamaños de pantalla; finalmente como complemento para que la interfaz web pueda interactuar con el entorno de la cadena de bloques la extensión para navegadores **MetaMask** actúa como billetera digital o cripto billetera, para gestionar y administrar las claves privadas de Ethereum, permitiendo a los usuarios gestionar los activos digitales e interactuar con aplicaciones de la cadena de bloques de forma segura y sencilla (MetaMask Learn, 2024).

3.4.3.2. Registro de credenciales de usuarios

El registro de usuarios se realizara mediante un contrato inteligente, el cual permite registrar y solicitar la aprobación de solicitud de conexión; además si se requiere, deberá existir un administrador de credenciales quien se encargará de asignar e ingresar la información de las credenciales de conexión de los usuarios, las cuales se almacenan en la cadena de bloques aprovechando su principio de inmutabilidad siendo el administrador el único que podrá ver la información generada y que se ha ido almacenado en la cadena de bloque en el transcurso del tiempo.

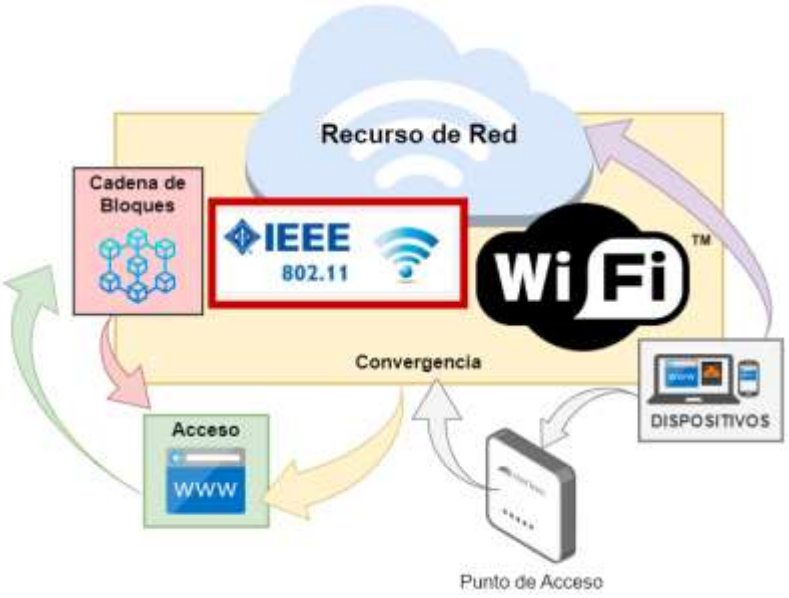
3.4.4. Convergencia

En esta etapa se buscará una configuración que correlacione la tecnología de cadena de bloques con el estándar IEEE 802.11, es decir que la red Ethereum permita al equipo de punto de acceso inalámbrico brindar la conexión y acceso a la red inalámbrica configurada y que Ganache mediante su red Ethereum valide la conexión y el acceso a Internet o al recurso de red, como se

representa en la Figura 32. La convergencia entre el estándar IEEE 802.11 y la cadena de bloques se refiere a la integración de la tecnología inalámbrica WiFi con la cadena de bloques para crear una solución y aplicación descentralizada.

FIGURA 32

DIAGRAMA DE CONVERGENCIA



La cadena de bloques es un registro digital descentralizado e inmutable que permite la verificación y el almacenamiento seguro de transacciones y datos. Proporciona transparencia, seguridad y trazabilidad a través de la distribución y el consenso de la información en una red de nodos. En relación con la **Identidad** y **Autenticación,** la cadena de bloques proporciona una infraestructura segura para gestionar la identidad digital y la autenticación en redes inalámbricas. Los usuarios pueden tener un control más seguro de su identidad y la capacidad de compartir selectivamente información personal verificada a través de la cadena de bloques, lo que puede mejorar la privacidad y la confianza en las conexiones inalámbricas.

3.4.5. Conectividad

La conectividad respecto al diseño planteado en la Figura 27, y en base a la propuesta del diseño de un mecanismo de autenticación utilizando cadena de bloques en una red inalámbrica de pruebas en la FICA, se representa por los dispositivos que permiten acceder al entorno de la red inalámbrico de pruebas; en este caso los equipos de punto de acceso inalámbrico o APs; los cuales deben permitir la conexión inalámbrica de dispositivos mediante el estándar IEEE 802.11, para su posterior validación y autorización mediante credenciales en la interfaz web, permitiendo así la conexión y acceso a los recurso de red o Internet.

3.4.6. Dispositivos

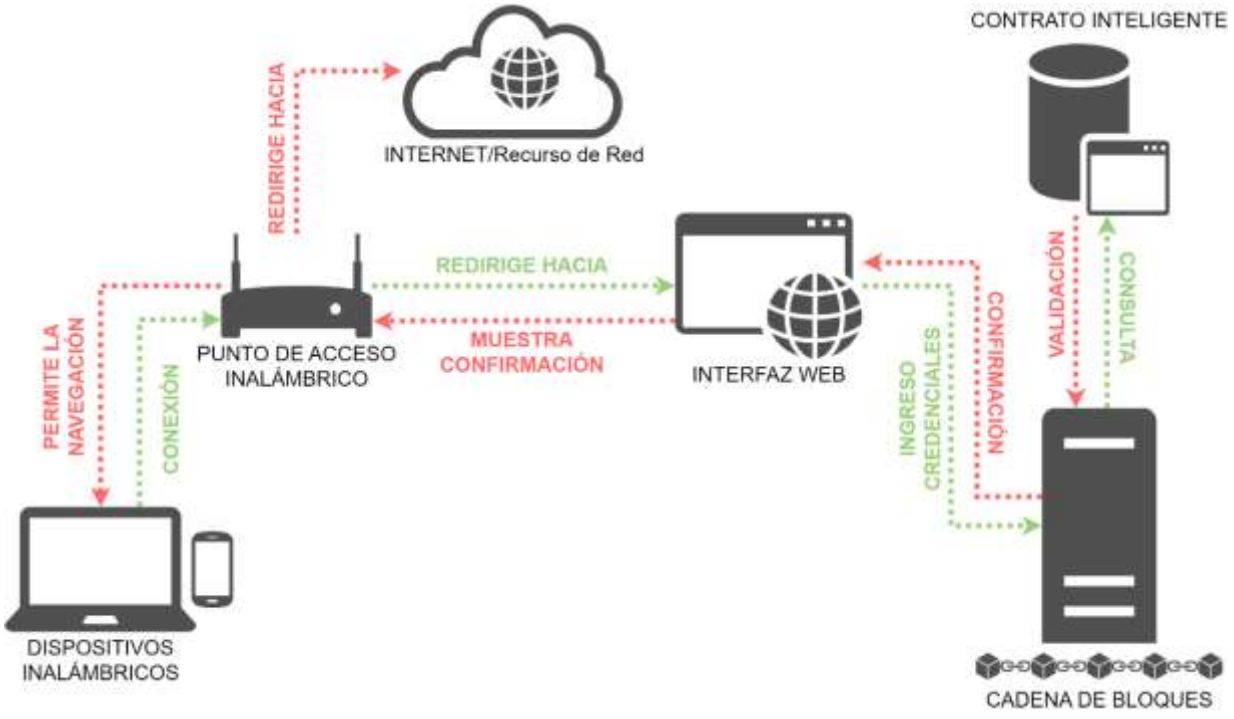
Respecto a los dispositivos que van a usarse dentro del proceso del proyecto planteado y que se observa em la Figura 27 estos serán; todos los equipos inalámbricos de los usuarios que acceden y hacen uso de los recursos de red e Internet de la FICA cotidianamente, los cuales deberán interactuar con la interfaz web de validación y acceso independientemente de si son equipos móviles inteligentes, laptops, tabletas electrónicas o el sistema operativo que usen, siempre y cuando puedan conectarse a la red inalámbrica de pruebas mediante la tecnología WiFi y realicen el ingreso de credenciales usando un navegador web y de ser el caso, juntamente con el complemento de la billetera digital o cripto billetera web.

3.5. Implementación

En base a los elementos y componentes usados para poder adaptar el mecanismo de autenticación propuesto, se elaboró el diagrama de topología que se muestra en la Figura 33, donde se grafica el proceso básico de petición de acceso, validación y autorización de conexión, aprovechando la tecnología de cadena de bloques para que actúe como entidad certificadora de credenciales de acceso a la red inalámbrica.

FIGURA 33

DIAGRAMA TOPOLÓGICO PARA EL MECANISMO DE AUTENTICACIÓN PROPUESTO

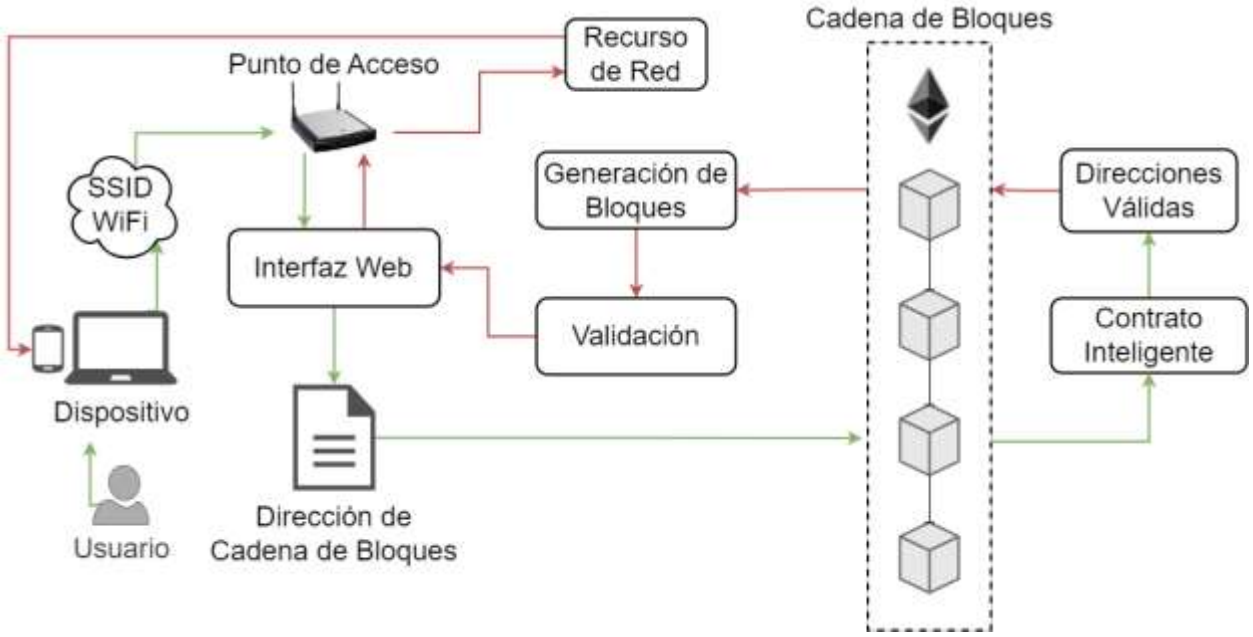


3.5.1. Diseño de Implementación

Para visualizar la implementación se elaboró un diagrama basado en la arquitectura establecida en la Sección 3.4 para el mecanismo de autenticación propuesto, el cual se representa en la Figura 34; donde se esquematiza el proceso de conexión al entorno de red inalámbrico de pruebas desde el punto de vista del usuario. Las flechas en color verde simbolizan el procedimiento a realizarse desde un dispositivo que quiere acceder mediante el identificador de red inalámbrica que propaga el punto de acceso, haciendo uso de la interfaz web para la respectiva identificación e ingreso de direcciones, posteriormente la consulta y verificación de los datos en el contrato inteligente desplegado en la cadena de bloques. El procedimiento descrito por las flechas rojas muestra la validación de direcciones, generación de bloques y registro transaccional respectivo, informado mediante la interfaz web para autorizar el acceso al recurso inalámbrico de red.

FIGURA 34

ARQUITECTURA DEL MECANISMO DE AUTENTICACIÓN BASADO EN CADENA DE BLOQUES



En el proceso para le mecanismo de autenticación propuesto, se contempla una etapa previa para establecer las condiciones del contrato inteligente, que servirá para validar el acceso y conexión del dispositivo de usuario; a continuación, se hará uso de la cadena de bloques para verificar los datos; la interfaz web ayudará a gestionar las conexiones de usuarios, además de proporcionar un entorno amigable para el ingreso de credenciales y solicitud de conexión.

El desarrollo del contrato inteligente se llevará a cabo juntamente con el de la interfaz web usando los compiladores, herramientas e IDEs que permiten realizar pruebas rápidas de funcionamiento, y que en conjunto con Solidity y su lenguaje base permite la integración con una interfaz basada en HTML y JavaScript donde el contrato se vinculan con la red de pruebas de Ethereum, usando los complementos y mecanismos para el uso de las librerías Ethers.js o Web3.js que permiten recuperar información de las direcciones de la cadena de bloques, semejante al uso de la cripto-billetera Metamask.

3.5.2. Desarrollo del Mecanismo de Autenticación utilizando Cadena de Bloques

Como solución para lograr la autenticación inalámbrica mediante el mecanismo de autenticación propuesto se usará una interfaz de ingreso de direcciones de cadena de bloques para los usuarios desarrollada en Vite React con TypeScript, el entorno de la cadena de bloques Ganache generará y validará las direcciones para conceder el acceso al entorno inalámbrico de red; mientras que las condiciones y funciones del contrato inteligente que interactúa con el entorno de la cadena de bloques se hará usando el framework Truffle, las configuraciones del equipo para el punto de acceso inalámbrico se realizan usando la información y documentación disponible en el sitio web del fabricante (Mikrotik, 2025), en este caso se usa un equipo Mikrotik AP RB951Ui 2HnD, el cual propagara el identificador de red y será parte de la infraestructura del entorno de pruebas inalámbrico.

3.5.2.1. Diagrama de Clases Mecanismo de Autenticación

El mecanismo de autenticación propuesto requiere el despliegue y configuración de diversos componentes, los cuales se han dividido y representado en un diagrama de clases para tener un enfoque claro de las secciones y su relación dentro del desarrollo del mecanismo de autenticación, estas se definen a continuación y se representan en la Figura 35. Los atributos que componen la clase **ContratoInteligente** para el mecanismo de autenticación contiene la dirección de propietario de contrato, variable de direcciones permitidas, tiempos de acceso y sus métodos son; **addAllowedAccount(address _account)** que agrega una dirección de cuenta permita del entorno de cadena de bloques, **removeAllowedAccount(address _account)** que elimina una dirección de cuenta permita del entorno de cadena de bloques, **isAccountAllowed(address _account)** que consulta si una dirección de cuenta del entorno de cadena de bloques es permitida, **grantAccess(address _account)** que brinda acceso mediante una dirección de cuenta válida,

batchAddAccounts(address[] calldata _accounts) que autoriza determinadas direcciones de cuenta del entorno de cadena de bloques.

La clase **InterfazWeb** representa la interacción de los usuarios con el contrato inteligente y el entorno de la cadena de bloques, contiene atributos como la identificación de contrato compilado que genera un archivo .abi, la dirección de contrato compilado, la extensión para exportar los valores de las direcciones de cuenta, las direcciones de cuenta disponibles generadas en el entorno de cadena de bloques y variables de validación, despliegue de las direcciones y los métodos para esta clase son; **validateAccount(accountAddress: string)** que verifica la dirección ingresada, **checkConnection()** que confirma la conexión hacia la cadena de bloques, **getAccounts()** que obtiene las direcciones de cuenta validas, **checkBlockchainConnection()** que verificar la conexión desde la cadena de bloques.

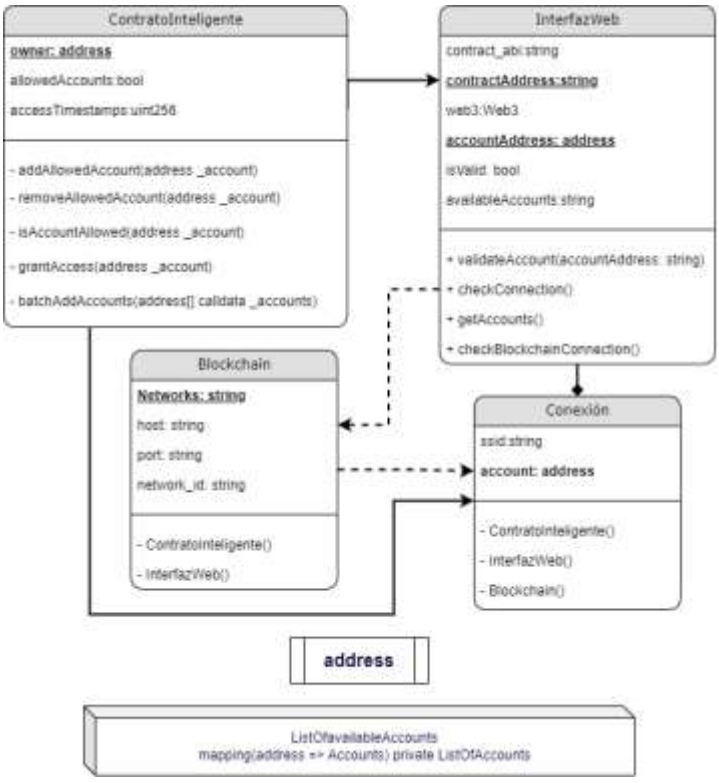
Para la clase **Conexión** está definida como la infraestructura inalámbrica del mecanismo de autenticación y la interacción con la interfaz web, el contrato inteligente y la cadena de bloques, sus atributos son identificador de la red inalámbrica del entorno de pruebas y las direcciones de cuenta validas usadas para la conexión y acceso, los métodos dentro de esta clase representan la interacción directa y codependencia con las clases; ContratoInteligente(), InterfazWeb() y Blockchain().

Dentro de la clase **Blockchain** se representa el entorno de cadena de bloques desplegado y como interactúa con el contrato inteligente, los usuarios, la interfaz web y el mecanismo de acceso, sus atributos son la dirección IP del host donde se despliega, el puerto del entorno y el identificador de red del entorno de cadena de bloques; de igual manera los métodos dentro de esta clase representan la interacción directa y codependencia con las clases; ContratoInteligente() e InterfazWeb().

Finalmente, la clase **ListOfavailableAccounts** que organiza y gestiona las direcciones de cuenta del entorno de cadena de bloques, el método de esta clase es; **mapping(address => Accounts) private ListOfAccounts** que almacena una lista de las direcciones de cuenta validas de la cadena de bloques.

FIGURA 35

DIAGRAMA DE CLASES MECANISMO DE AUTENTICACIÓN BASADO EN CADENA DE BLOQUES



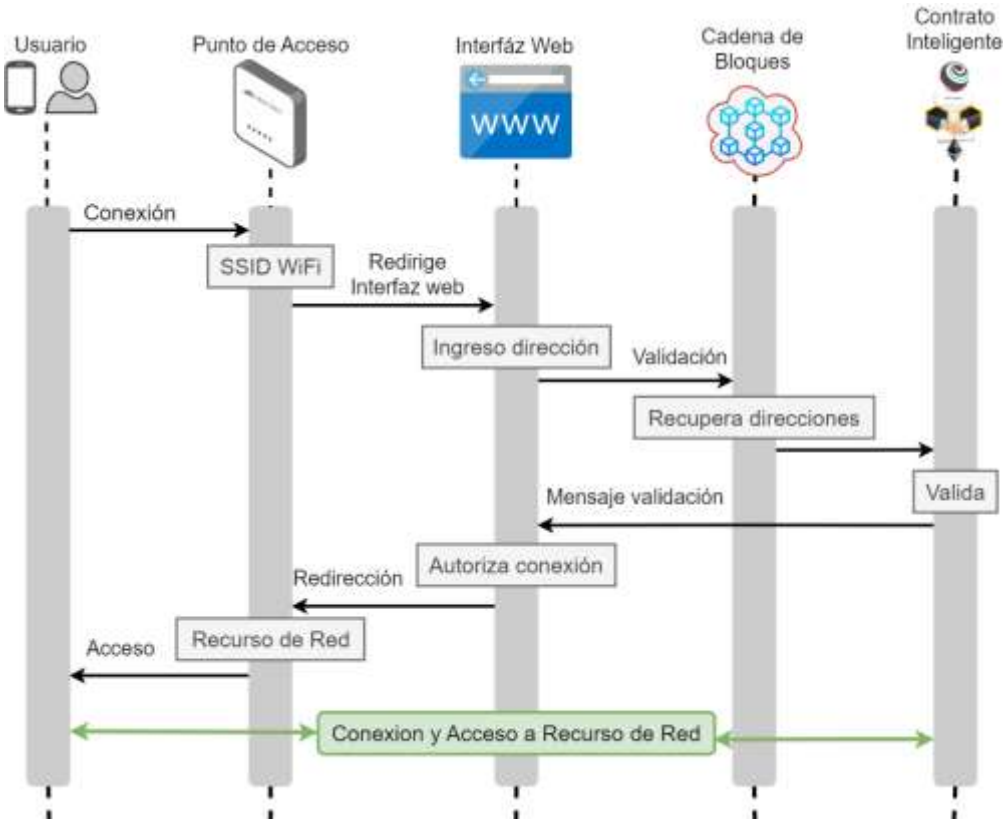
3.5.2.2. Diagrama de Secuencia Mecanismo de Autenticación

Con el fin de tener una apreciación grafica del proceso que se realizará usando el mecanismo de autenticación propuesto, se elabora el diagrama de secuencia que se muestra en la Figura 36; donde se representa el comportamiento de conexión desde el usuario, quien inicia el proceso de conexión a través del punto de acceso inalámbrico hasta la interfaz web del mecanismo de autenticación, donde se realiza el ingreso de una dirección de cuenta válida, para su posterior

validación mediante la cadena de bloques, finalmente se realiza la autenticación permitiendo la conexión del dispositivo el cual pude acceder al entorno inalámbrico de la red de pruebas.

FIGURA 36

DIAGRAMA DE SECUENCIA DE USUARIO PARA EL MECANISMO DE AUTENTICACIÓN



3.5.2.3. Mecanismo de Autenticación utilizando Cadena de Bloques

Para el desarrollo del mecanismo, se debe instalar y configurar las herramientas necesarias para interactuar con la red de pruebas Ethereum a su vez desplegar y probar los contratos inteligentes, así como el uso de librerías requeridas para el entorno de interfaz web y su interacción con los usuarios; como se muestra en el esquema de la Figura 37, la instalación y configuración de los complementos que se describen el esquema se detallan en el Anexo 4.

FIGURA 37

ESQUEMA DE HERRAMIENTAS REQUERIDAS PARA EL MECANISMO DE AUTENTICACIÓN

3.5.3. Desarrollo de Contrato Inteligente

El desarrollo de un contrato inteligente implica escribir y compilar código de programación para desplegarlo en la plataforma de cadena de bloques Ethereum de Ganache, este ejecuta automáticamente los términos y condiciones del contrato en el framework Truffle, cuando las variables definidas cumplan ciertas condiciones como se muestra en la Figura 38.

En el contrato inteligente se define la dirección **address public owner**, que identifica al dueño del contrato inteligente y define quien lo despliega; los atributos que constan en el contrato inteligente están conformados por un mapeo **allowedAccounts**, que es un registro de las direcciones válidas bajo una variable *bool,* otro mapeo **accessTimestamps**, para un registro de tiempo de acceso de cada dirección de cuenta. Entre los eventos estan; **AccountAllowed**, que autoriza una dirección de cuenta; **AccountRevoked**, que revoca el permiso de una dirección de cuenta y **AccessGranted**, que ingresa una dirección de cuenta en el mecanismo. Mientras que las funciones son;

addAllowedAccount(address _account) que añade una dirección de cuenta a la lista de permitidos, **removeAllowedAccount(address _account)** que revoca los permisos de una dirección de cuenta, **isAccountAllowed(address _account)** que consulta si la dirección de cuenta es válida, **grantAccess(address _account)** que registra el acceso de una dirección de cuenta permitida previamente guardando un block.timestamp al momento del acceso y **batchAddAccounts(address[] _accounts)** que añade varias direcciones de cuenta a la vez.

FIGURA 38

CARACTERÍSTICAS DE CONTRATO INTELIGENTE

contracts > AccessControl.sol > ...

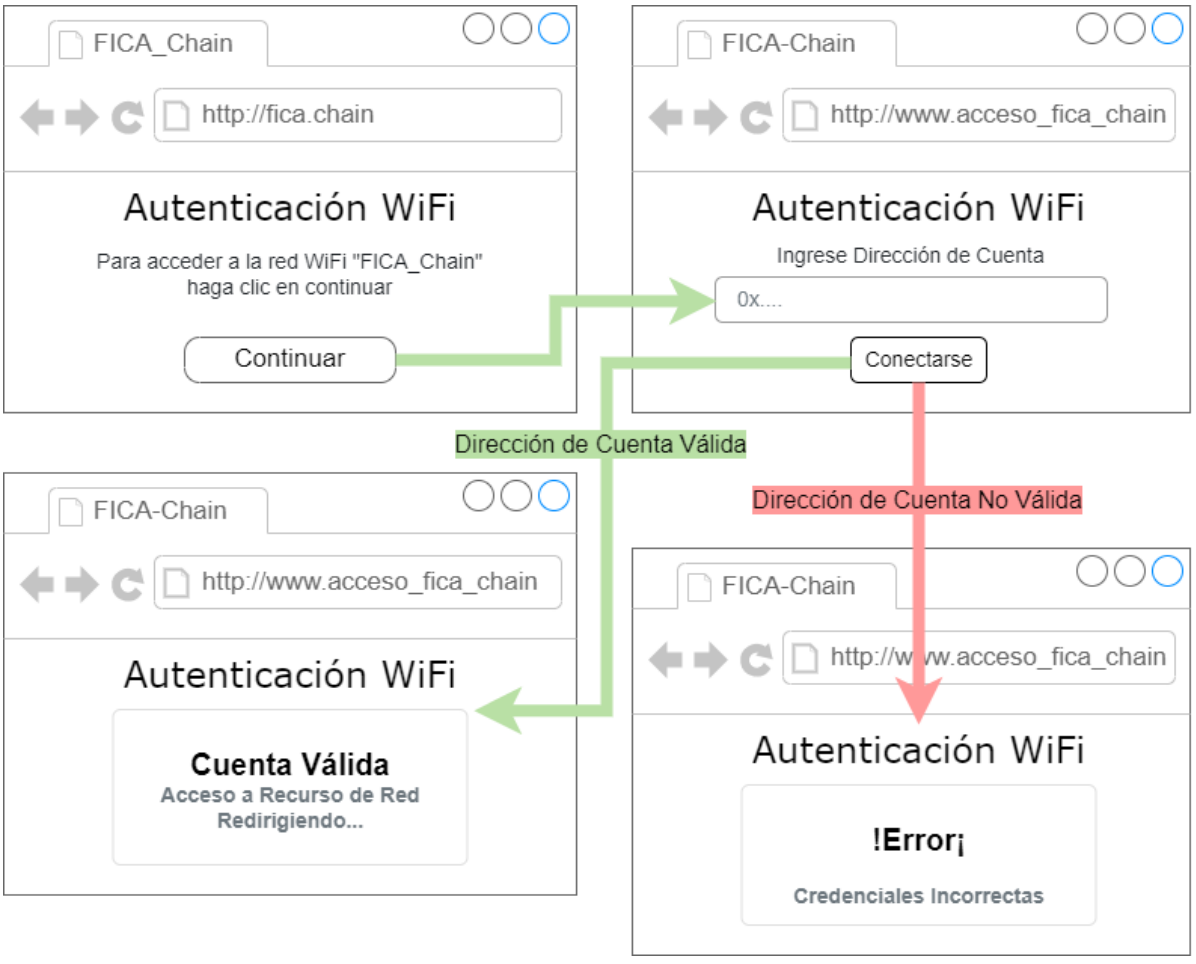
```solidity
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract AccessControl {
    address public owner;
    mapping(address => bool) private allowedAccounts;
    mapping(address => uint256) public accessTimestamps;

    event AccountAllowed(address indexed account);
    event AccountRevoked(address indexed account);
    event AccessGranted(address indexed account, uint256 timestamp);

    constructor() {
        owner = msg.sender;
    }

    modifier onlyOwner() {
        require(msg.sender == owner, "Not owner");
        _;
    }

    function addAllowedAccount(address _account) external onlyOwner {
        allowedAccounts[_account] = true;
        emit AccountAllowed(_account);
    }

    function removeAllowedAccount(address _account) external onlyOwner {
        allowedAccounts[_account] = false;
        emit AccountRevoked(_account);
    }

    function isAccountAllowed(address _account) external view returns (bool) {
        return allowedAccounts[_account];
    }

    function grantAccess(address _account) external {
        require(allowedAccounts[_account], "Account not allowed");
        accessTimestamps[_account] = block.timestamp;
        emit AccessGranted(_account, block.timestamp);
    }

    function batchAddAccounts(address[] calldata _accounts) external onlyOwner {
        for (uint i = 0; i < _accounts.length; i++) {
            allowedAccounts[_accounts[i]] = true;
            emit AccountAllowed(_accounts[i]);
        }
    }
}
```

3.5.4. Diseño de Interfaz Web

Para el diseño de la interfaz web, como ya se indicó en la Sección 3.3.5.2 se usará la biblioteca JavaScript juntamente con Vite React y TypeScript que permite desarrollar interfaces de usuario sencillas y amigables con los usuarios, en base a los lineamientos de diseño mostrado en la Figura 39.

FIGURA 39

DIAGRAMA DE DISEÑO DE INTERFAZ WEB

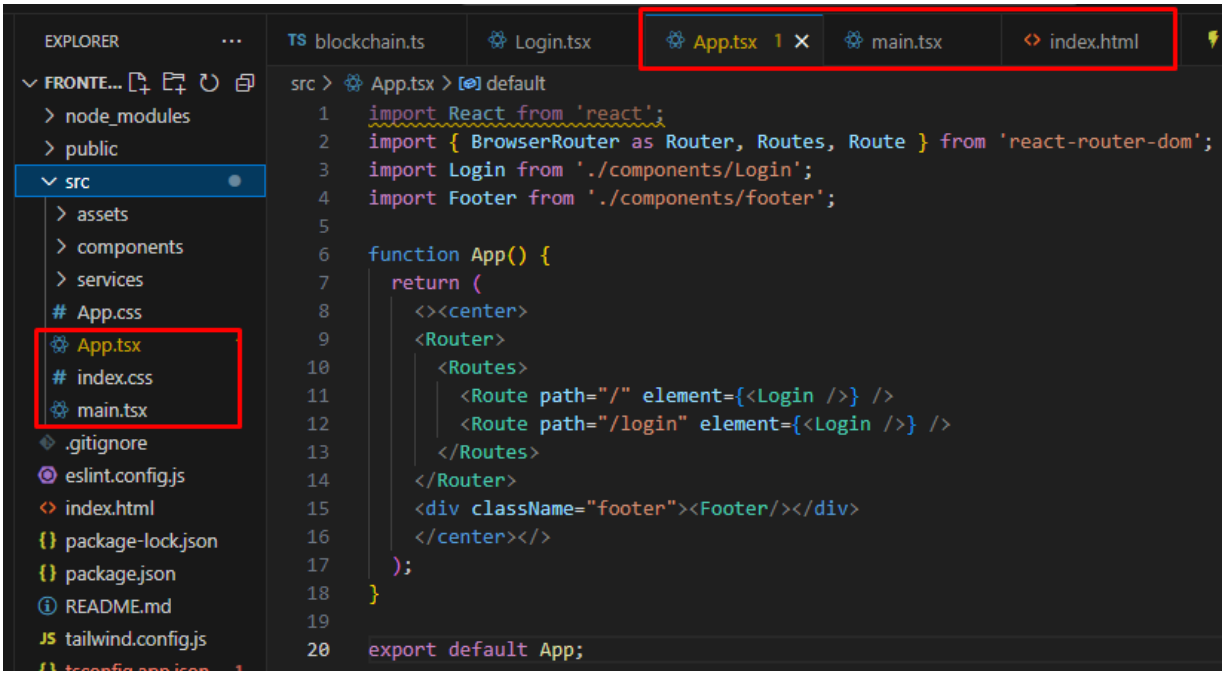


Nota: La imagen muestra el diseño de la interfaz web para el acceso y conexión de usuarios mediante el mecanismo de autenticación propuesto, con las respectivas ventanas de inicio, ingreso de dirección de cuenta y validación de conexión o error; que se despliega en los dispositivos de usuario.

La configuración de la interfaz de usuario inicia con la generación de la estructura, los archivos y directorios que se usan para el desarrollo de un proyecto con las herramientas Vite React con Typescript y sus complementos; realizando las modificaciones de cada archivo del proyecto, los contenidos dentro de la plantilla básica TypeScript, el archivo *index.html*, *main.tsx* y *App.tsx*; que usa ViteReact para renderizar la interfaz donde se personaliza la presentación y la disposición de los elementos como se muestra en la Figura 40.

FIGURA 40

ARCHIVOS Y DIRECTORIOS DE INTERFAZ WEB



En el archivo *Login.tsx* se establecen los campos para que el usuario realice el ingreso de la dirección de una cuenta valida generada en el entorno de cadena de bloques, la validación de la conexión se realiza mediante la configuración de un botón, que muestra en la interfaz un mensaje para la validación exitosa o de error representado en la Figura 41, además dentro de la parte lógica se vincula con el archivo *blockchain.ts*, el cual interactúa con el entorno de cadena de bloque

recuperando las direcciones de cuenta validas mediante el enlace asociado a la dirección de cuenta que se genera al desplegar el contrato inteligente como se observa en la Figura 42.

FIGURA 41

CONFIGURACIÓN DE ARCHIVO LOGIN.TSX

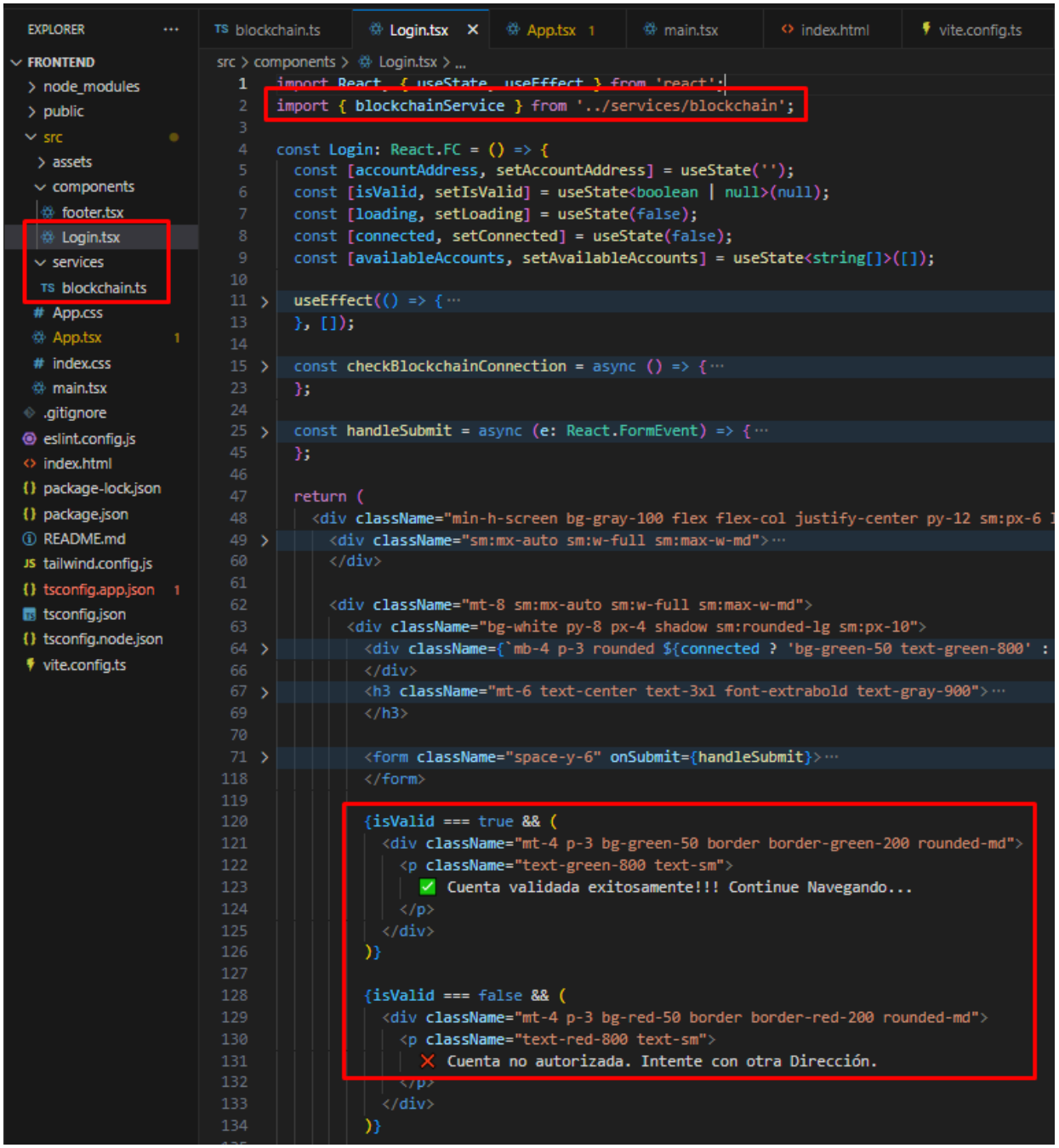
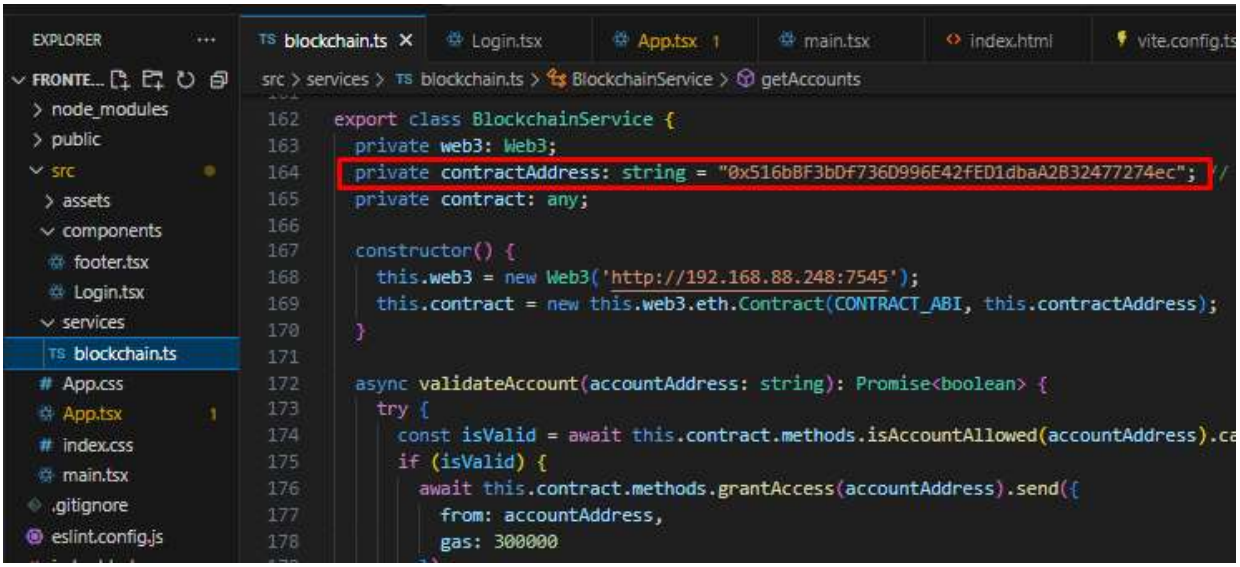


FIGURA 42

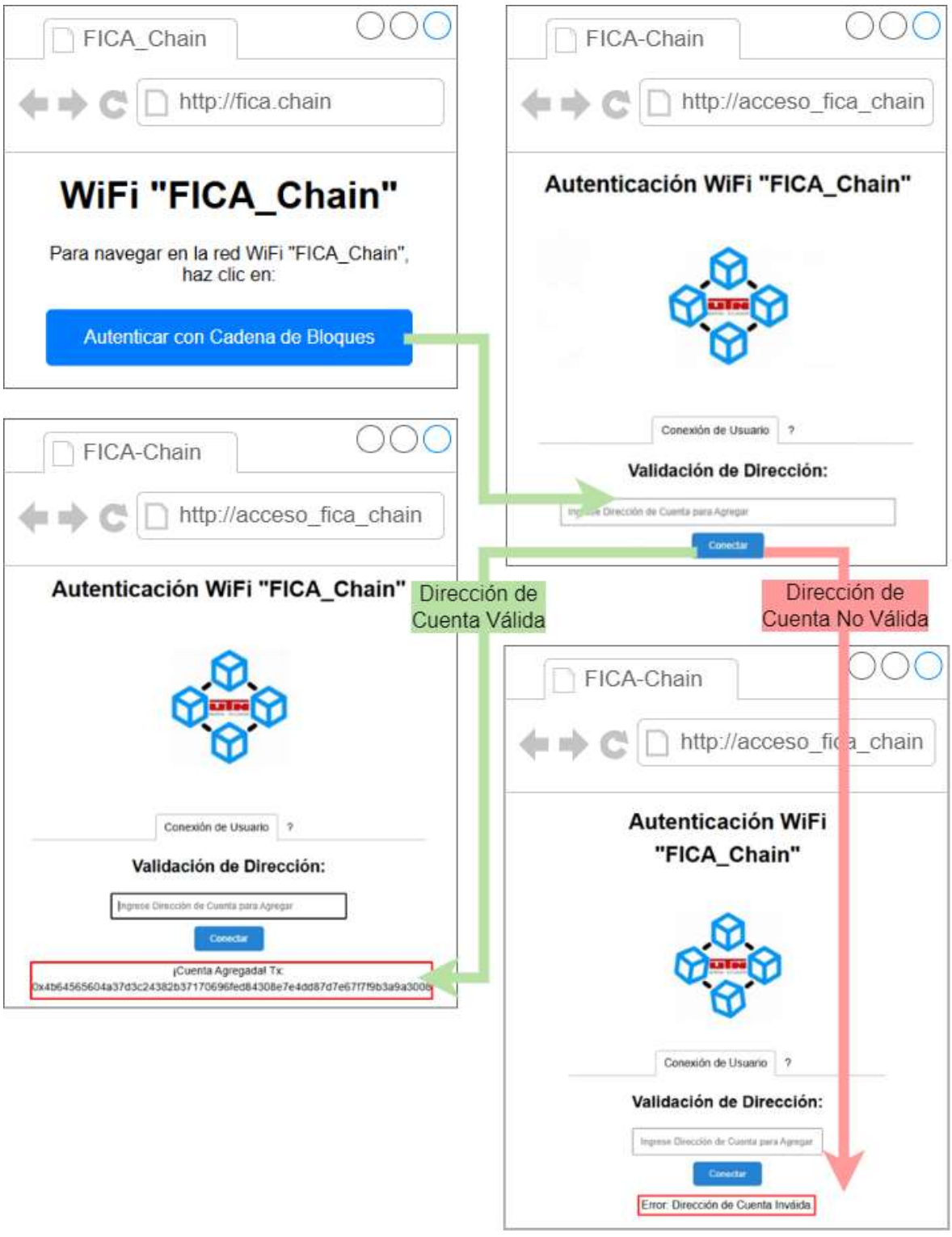
CONFIGURACIÓN DE ARCHIVO BLOCKCHAIN.TS



Los componentes de despliegue de la interfaz web interactúa de manera interna y lógica con otros elementos desarrollados para el mecanismo de autenticación propuesto como; el entorno de cadena de bloques Ganache, el Framework Truffle para el despliegue del contrato inteligente y la configuración del punto de acceso inalámbrico Mikrotik AP RB951Ui 2HnD. Finalmente, la interfaz web se mostrará cuando un dispositivo de usuario se conecte al identificador de red inalámbrico configurado para el entorno de pruebas; redirigiendo al usuario hacia el proceso que le permite ingresar una dirección de cuenta valida y disponible en la cadena de bloques, validar la conexión y acceder al recurso de red; como se aprecia en la Figura 43.

FIGURA 43

INTERFAZ WEB MECANISMO DE AUTENTICACIÓN PROPUESTO

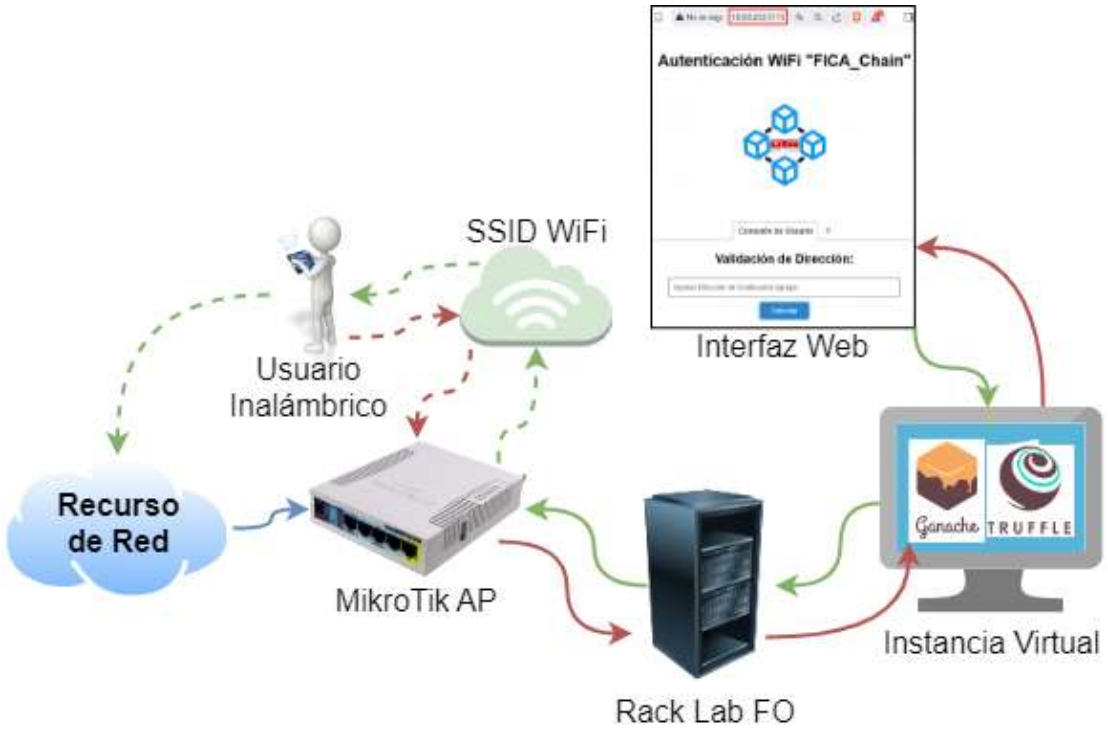


3.5.5. Despliegue de mecanismo de autenticación

Para el despliegue del mecanismo de autenticación se plantea diseñar un entorno inalámbrico de pruebas limitado, semejante a la red de la Facultad de Ingeniería en Ciencias Aplicadas, el cual se puede observar en la Figura 44, y que se levantará usando los equipos e infraestructura disponibles en el Laboratorio de Fibra Óptica de la Carrera de Telecomunicaciones de la facultad. Se usará un equipo Mikrotik RB931Ui-2HnD, una máquina o instancia virtual, donde configurará la cadena de bloques de pruebas, el mecanismo de autenticación y la interfaz web de validación y acceso.

FIGURA 44

DISEÑO DE ENTORNO DE PRUEBAS DESPLEGADO



Nota: La imagen muestra los elementos del entorno de la red de pruebas planteado para el mecanismo de autenticación propuesto; donde el usuario se conecta al SSID de la red inalámbrica, es redirigido a la interfaz web para el ingreso de la dirección de cuenta y luego de la validación se concede el acceso al recurso de red.

4. Capítulo IV: Análisis de Resultados

El capítulo presentará, una evaluación definida dentro del entorno de pruebas establecido para la implementación del mecanismo de autenticación propuesto, durante este proceso se obtendrán datos e información necesaria para el despliegue del mecanismo, su funcionalidad, rendimiento, seguridad, desempeño con de la red inalámbrica y accesibilidad de la interfaz; proporcionando una percepción de la operatividad del mecanismo de autenticación basado en cadena de bloques para una red inalámbrica de pruebas en la Facultad de Ingeniería en Ciencias Aplicadas.

4.1 Pruebas

La realización de pruebas permitirá verificar la funcionalidad, integración, seguridad y rendimiento del mecanismo de autenticación propuesto; las cuales se establecen y detallan en la Tabla 17, donde se define el entorno y alcance da las pruebas, los requisitos previos o parámetros a tomar en cuenta al momento de realizar la prueba y la metodología de validación. y/o resultados a obtener.

TABLA 17

PLAN DE PRUEBAS

PLAN DE PRUEBAS		
Pruebas de Funcionalidad		
Prueba	**Requerimiento**	**Resultado Esperado**
TEST 1: Levantar el entorno de red inalámbrico de pruebas en la facultad para el mecanismo de autenticación.	Se despliega el entorno topológico de pruebas propuesto con los componentes y equipos de red.	Se comprobará el funcionamiento de los equipos dentro de la infraestructura topológica de red.
TEST 2: Configurar los equipos y componentes del entorno de red inalámbrico de	Se establecen las configuraciones de IP, SSID y seguridad Wi-Fi en el punto de acceso inalámbrico; el	Se comprobará la conectividad y acceso de los equipos y dispositivos al recurso de red, sin usar el

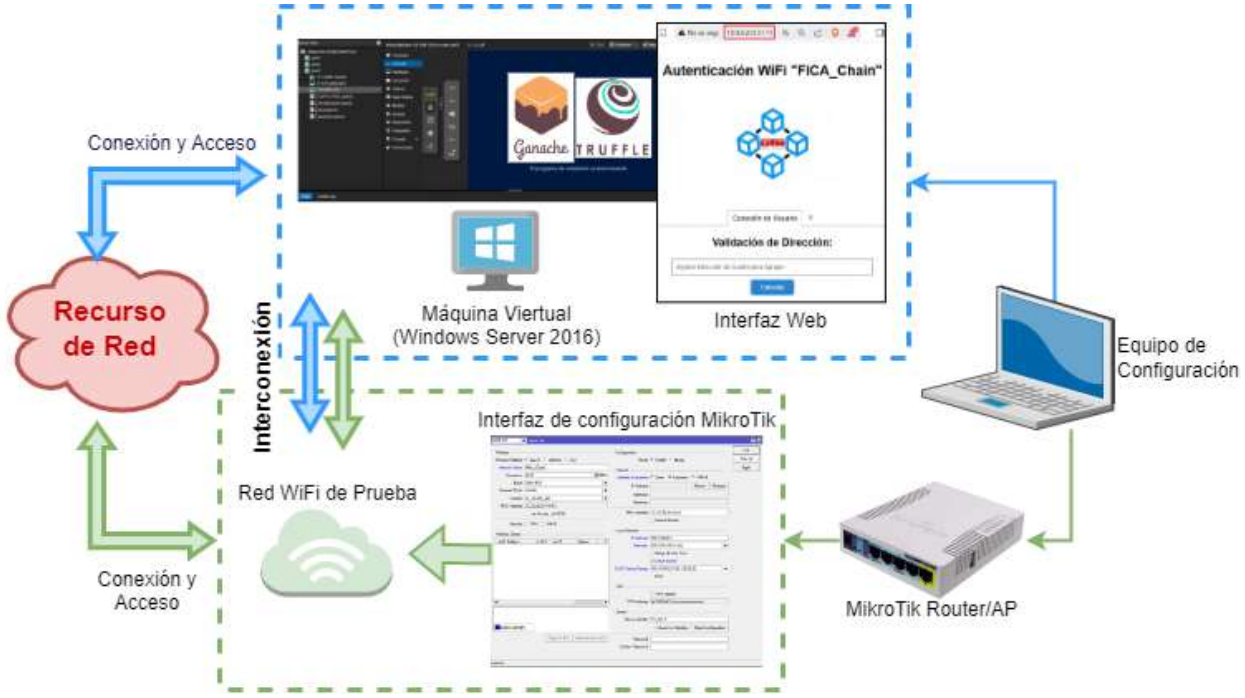
pruebas de la facultad para el mecanismo de autenticación.	despliegue del entorno de la tecnología de Cadena de Bloques en el servidor y los complementos de uso en los dispositivos de usuarios.	mecanismo de autenticación propuesto.
TEST 3: Evaluación funcional de interfaz gráfica de acceso y entorno de cadena de bloques.	Levantamiento y configuración de interfaz web, despliegue de componentes de tecnología de cadena de bloques.	Se evaluará acceso a la interfaz web, y la interacción con los componentes y el entorno de cadena de bloques.
Pruebas de Integración		
Prueba	**Requerimiento**	**Resultado Esperado**
TEST 4: Integración dirección de cuenta y usuario.	Acceso a interfaz web e ingreso de credenciales de usuario asignadas.	Se evaluará la adaptabilidad del dispositivo de usuario a la interfaz web y los complementos de conexión.
TEST 5: Autenticación de usuario.	Convergencia de tecnologías de cadena de bloques con protocolo Wi-Fi.	Se verificará el acceso al recurso de red después de la autenticación.
Pruebas de Seguridad		
Prueba	**Requerimiento**	**Resultado Esperado**
TEST 6: Autenticación de usuario sin credenciales.	Ingreso de credenciales de usuario no válidas.	Se verificará, que el dispositivo de usuario no pueda acceder al recurso de red y que la interfaz web muestre el mensaje de error respectivo.
TEST 7: Comportamiento de entorno de cadena de bloques.	Ingreso de credenciales de usuario válidas y no válidas.	Se verificará el funcionamiento del entorno de cadena de cadena de bloques, observando los logs generados.
Pruebas de Rendimiento		
Prueba	**Requerimiento**	**Resultado Esperado**
TEST 8: Conectividad y autenticación.	Asignación y uso de direcciones de cuentas disponibles a usuario con diferentes tipos de dispositivos de conexión.	Se verificará el comportamiento de entorno de red y de cadena de bloques al validar la conexión de diferentes equipos.
TEST 9: Gestión de conexión.	Uso de direcciones de cuenta y conexión de usuario.	Se evaluará el tiempo de respuesta durante el proceso de autenticación.

4.1.1. TEST 1: Levantar el entorno de red inalámbrico de pruebas en la facultad para el mecanismo de autenticación

En esta evaluación inicial, se debe comprobar el funcionamiento de los equipos y componentes de la infraestructura de la red de pruebas para el despliegue del mecanismo de autenticación propuesto, con la finalidad de comprender el funcionamiento de las interfaces de configuración de estos elementos, como se plantea en la Figura 45; donde un equipo de configuración para este caso una laptop se conecta a la infraestructura de la red de pruebas y a su vez conecta, crea y configura la instancia virtual que permite levantar el servidor con sistema operativo Windows Server 2016; de igual manera se conecta dentro de la infraestructura de pruebas un equipo MikroTik o punto de acceso inalámbrico y se accede a la interfaz de configuración, para realizar los cambios que sean requeridos para el entorno de pruebas.

FIGURA 45

ENTORNO DE PRUEBAS DE CONEXIÓN DE COMPONENTES Y EQUIPOS

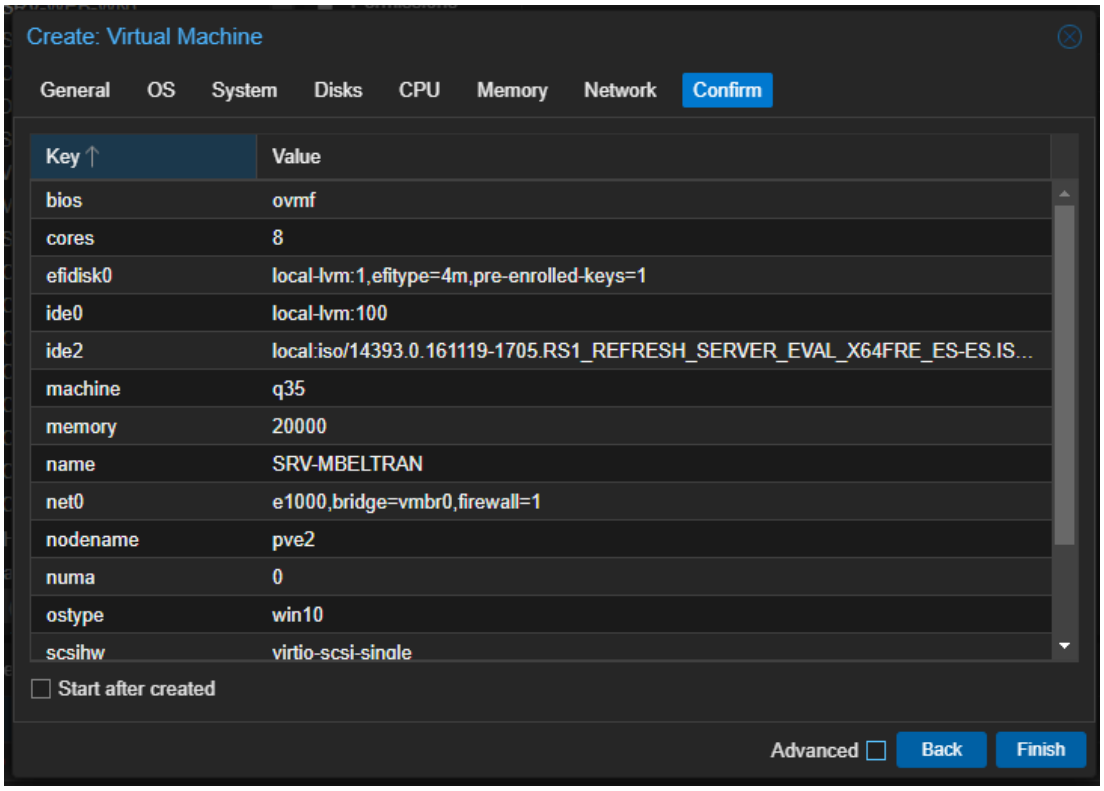


4.1.1.1. Resultado Test 1

La validación de esta etapa inicial de evaluación se demuestra, describe y detalla a continuación; iniciando con el proceso de configuración de la instancia virtual, donde se deben establecer los parámetros del procesamiento, RAM, almacenamiento, interfaces de conexión, imagen del sistema operativo que se usará y nombre identificador de la instancia; tal y como lo muestra la Figura 46.

FIGURA 46

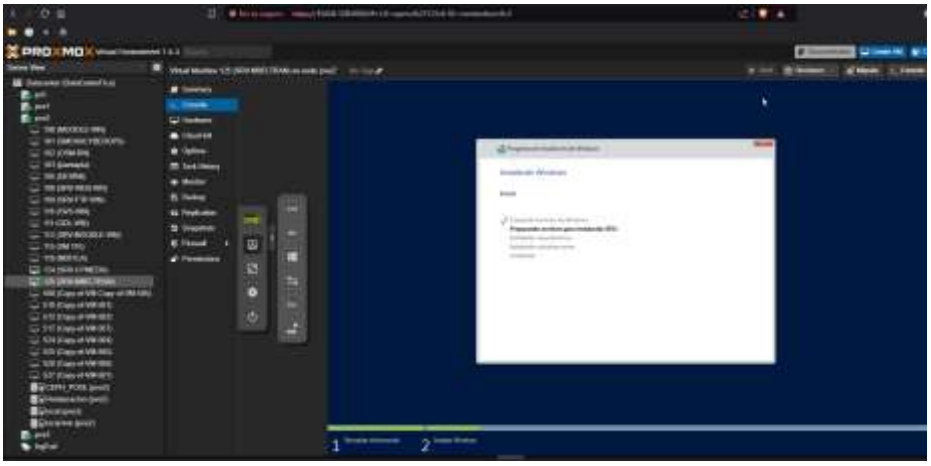
RESUMEN DE CONFIGURACIÓN PARA LA CREACIÓN DE UNA INSTANCIA VIRTUAL



Una vez creada la instancia virtual, esta se inicia para continuar con la instalación y configuración del sistema operativo Windows Server 2016 como lo muestra la Figura 47, proceso en el cual se selecciona idioma, distribución del teclado, partición de disco en la cual se desea instalar el sistema operativo, configuración de contraseña de administrador y entre otras configuraciones típicas de instalación del sistema operativo.

FIGURA 47

INSTALACIÓN DE WINDOWS SERVER 2016 EN INSTANCIA VIRTUAL



Para la conexión del equipo MikroTik a la infraestructura de red y al equipo de configuración, se muestra en la Figura 48; donde un puerto ethernet del equipo MikroTik se conecta al equipo de configuración, mientras que otro puerto se conecta a la LAN del recurso de red, para posteriormente formar la infraestructura de pruebas entre el Servidor Windows, el punto de acceso inalámbrico MikroTik, la respectiva red WiFi que se configure y el mecanismo de autenticación propuesto.

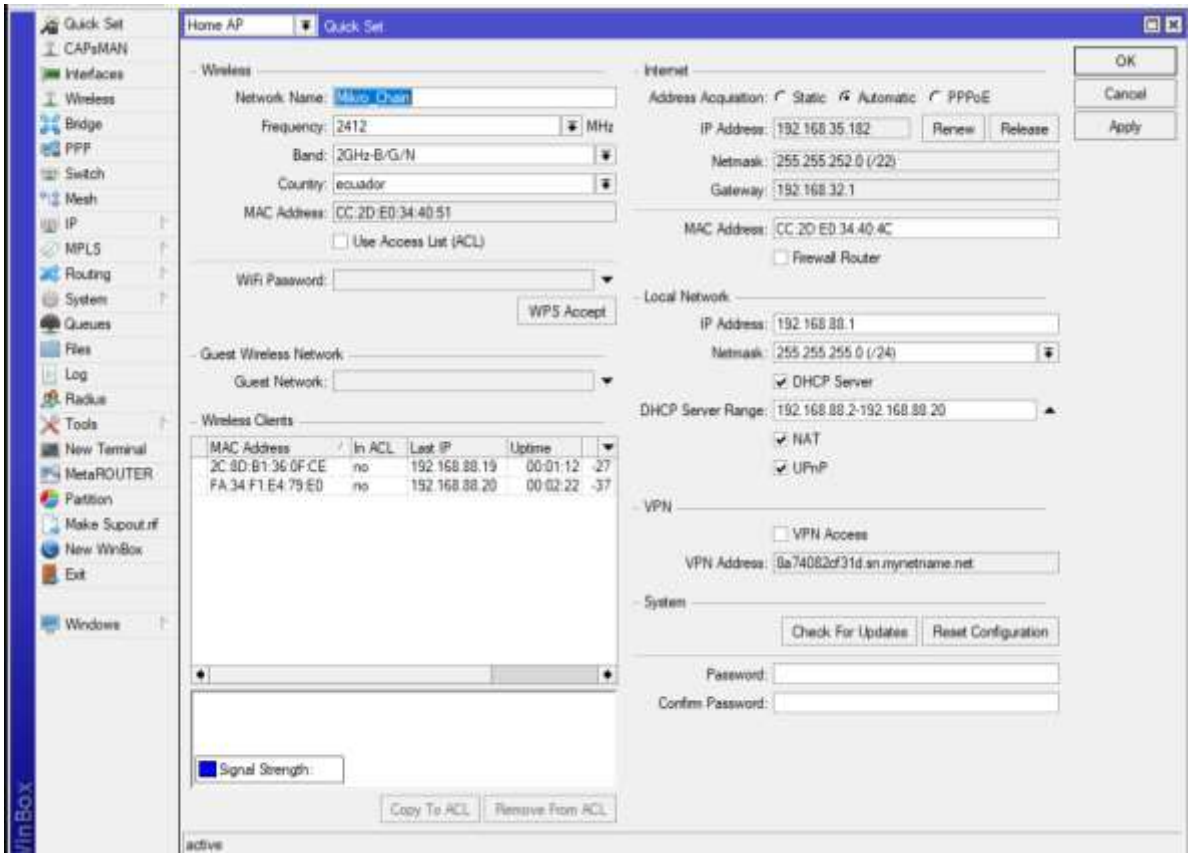
FIGURA 48

CONEXIÓN DE PUNTO DE ACCESO INALÁMBRICO MIKROTIK

Usando la dirección MAC de la interfaz ethernet conectada al equipo de configuración y con la ayuda de la herramienta WinBox, se accede a la interfaz de configuración del equipo MikroTik, para realizar los cambios requeridos e integrar el equipo al entorno de red de pruebas, como se observa en la Figura 49; la configuración mediante el menú "*Quick Set*" en modo "*Home AP*" facilita establecer los parámetros básicos Wireless como SSID, frecuencia, banda, país y seguridad; la conexión, configuración y asignación de dirección IP automática, estática o de punto a punto sobre protocolo ethernet (PPPoE) para el acceso a Internet y la configuración de domino de red propagada con su máscara de red, servidor para asignación y direccionamiento IP automático (DHCP), la traducción de direcciones IP (NAT) y la detección universal (UPnP).

FIGURA 49

INTERFAZ DE CONFIGURACIÓN PUNTO DE ACCESO INALÁMBRICO MIKROTIK

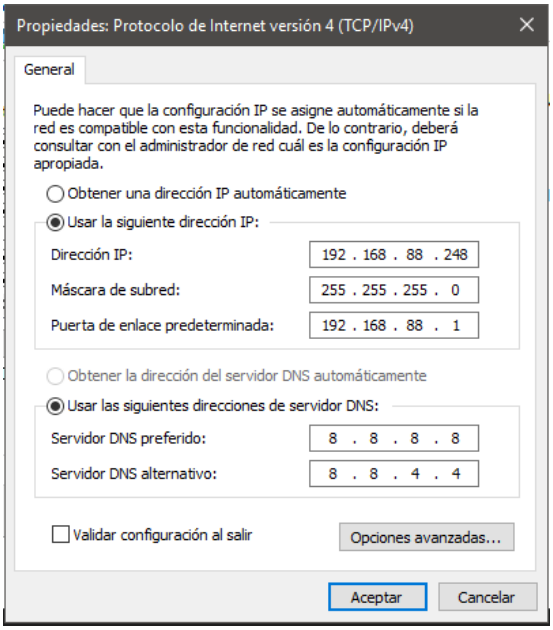


4.1.2. TEST 2: Configurar los equipos y componentes del entorno de red inalámbrico de pruebas de la facultad para el mecanismo de autenticación

La evaluación de la segunda prueba de funcionalidad se resume en garantizar la conectividad en el entorno de pruebas; configurando y asignando las respectivas direcciones IP en la máquina Windows Server, asignando una dirección IP dentro del dominio de red del área local, la máscara de red y la dirección de puerta de enlace respectiva; además de las direcciones del servicio de dominio de red (DNS) si se requiere, como se muestra en la Figura 50.

FIGURA 50

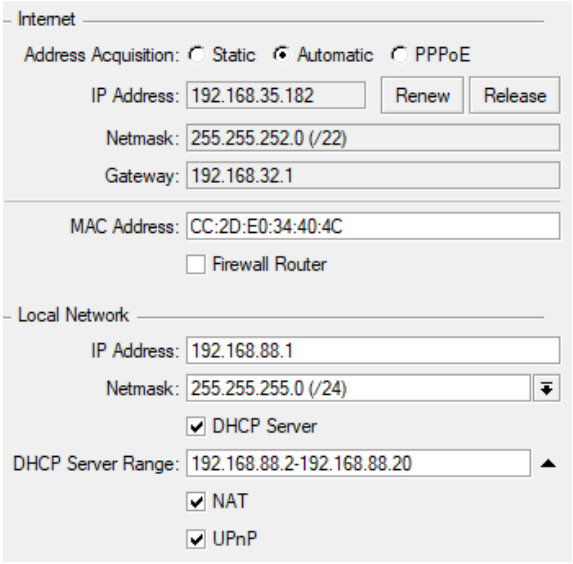
CONFIGURACIÓN DE IPS EN WINDOWS SERVER



De igual manera, se configura el acceso del equipo MikroTik con los parámetros de red del entorno de pruebas para la conexión y acceso a los recursos de red; obtenido los parámetros de manera automática por DHCP; así mismo, se configuran las características del dominio, mascara, servicio y rango para el protocolo dinámico de configuración de host (DHCP) de la red local que propaga el equipo MikroTik y se muestra en la Figura 51.

FIGURA 51

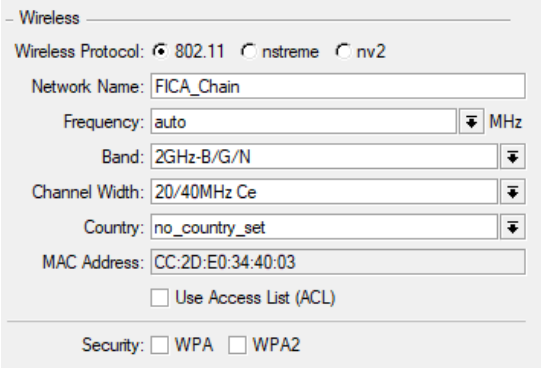
CONFIGURACIÓN DE IPS EN EQUIPO MIKROTIK



Para comprobar la conectividad entre el equipo Windows Server y el punto de acceso MikroTik se configuran los parámetros inalámbricos de red como se muestra en la Figura 52; el identificador de red, la frecuencia, banda, el país y la seguridad WiFi; esta red permite realizar pruebas de conexión entre los dispositivos conectados a la red inalámbrica propagada por el punto de acceso MikroTik y el equipo Windows Server.

FIGURA 52

CONFIGURACIÓN DE RED WIFI EN MIKROTIK

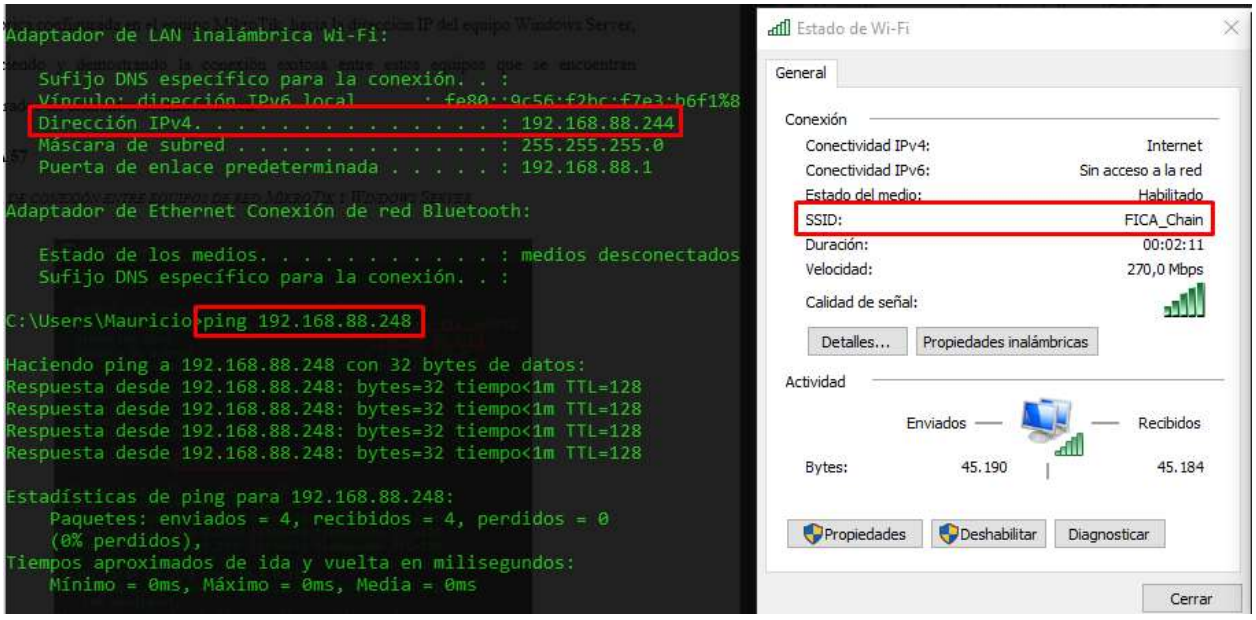


4.1.2.1. Resultado Test 2

La Figura 57, muestra la prueba de conectividad desde un dispositivo conectado a la red inalámbrica configurada en el equipo MikroTik, hacia la dirección IP del equipo Windows Server, estableciendo y demostrando la conexión exitosa entre, el equipo servidor que se encuentra conectado a la LAN y el equipo inalámbrico que se conecta mediante la WLAN del equipo MikroTik.

FIGURA 53

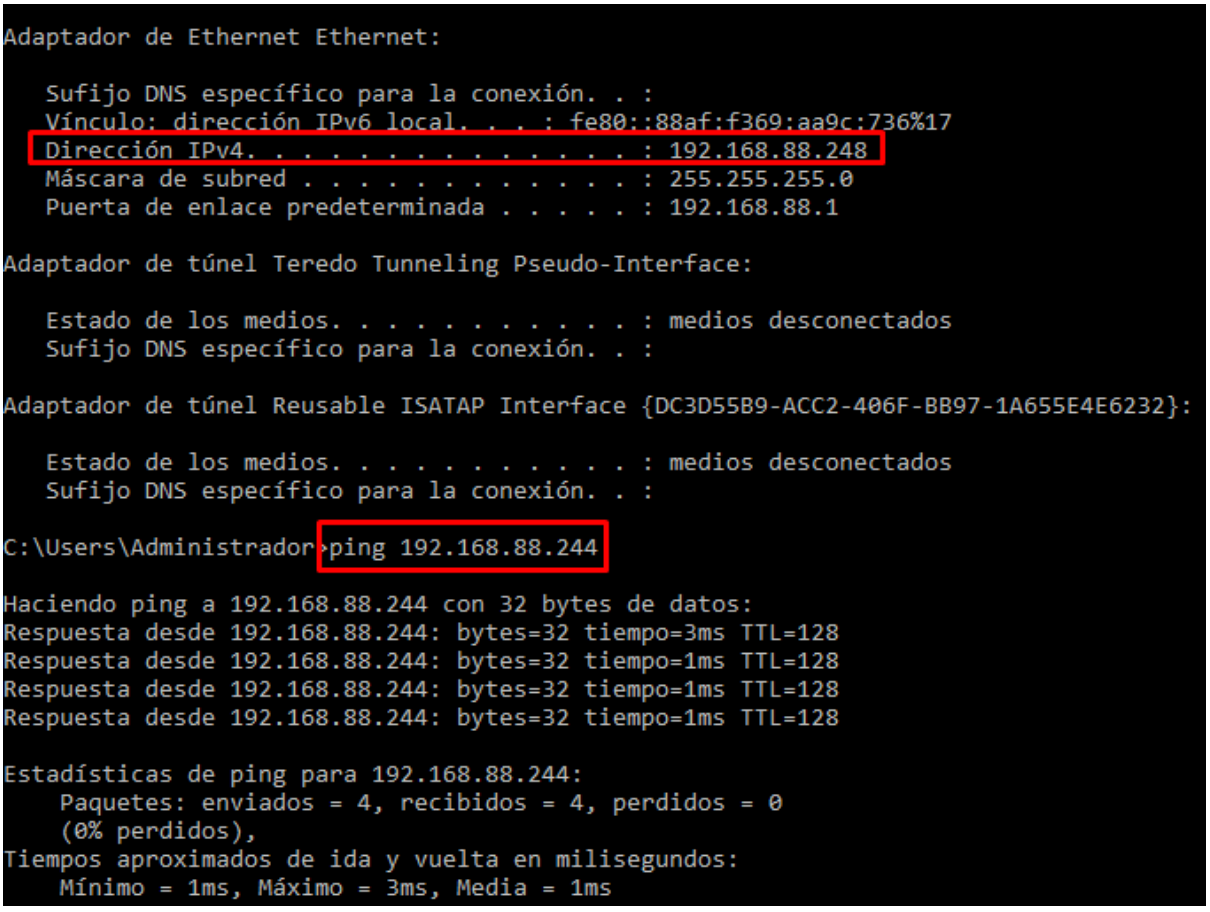
PRUEBA DE CONEXIÓN ENTRE EQUIPO DE WLAN Y WINDOWS SERVER



De igual manera, se prueba la conexión y acceso desde la dirección IP configurada en el equipo Windows Server, verificando y garantizando la posterior conexión y acceso de los equipos configurados en el entorno inalámbrico de pruebas del mecanismo de autenticación propuesto, como se aprecia en la Figura 54.

FIGURA 54

PRUEBA DE CONEXIÓN DESDE WINDOWS SERVER

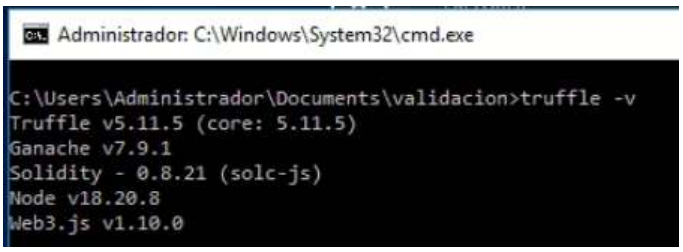


4.1.3. TEST 3: Evaluación funcional de interfaz gráfica de acceso y entorno de cadena de bloques

Como tercera parte de la evaluación de funcionalidad, se debe levantar y configurar el entorno de interfaz web, así como sus componentes; para lo cual se debe instalar las herramientas de software requeridas y necesarias; en la Figura 55 se muestra los complementos Truffle, Ganache, Solidity, Node.js y Web3.js instalados en el servidor Windows, el proceso de instalación y configuración de estos componentes se detalla en el Anexo 4.1.

FIGURA 55

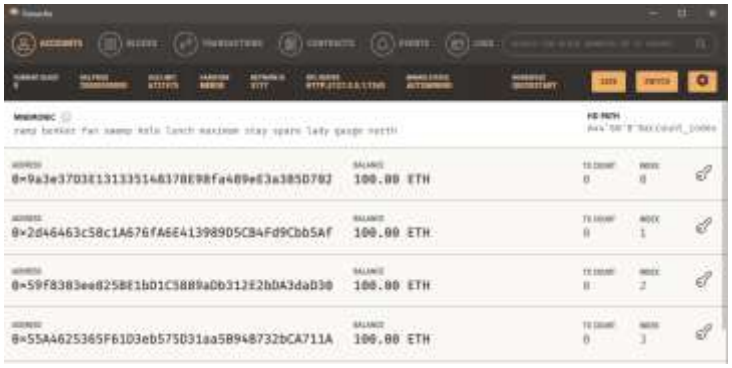
VERIFICACIÓN DE INSTALACIÓN DE COMPLEMENTOS



La instalación del entorno de pruebas Ethereum lo proporciona la plataforma Ganache, cuya ejecución muestra una interfaz gráfica como se observa en la Figura 56 o también puede hacerse mediante líneas de comandos usando el terminal de consola; las do maneras de ejecutar Ganache muestran los mismos parámetros; las cuentas, bloques, transacciones, el contrato, los eventos y los logs; además indica configuraciones para limitar el gasto, identificador de red, el puerto, la dirección del servidor y el tipo de entorno desplegado; también muestra las direcciones, el balance en "*ethers*" y las claves privadas de cada cuenta.

FIGURA 56

INTERFAZ DE PLATAFORMA GANACHE ETHEREUM

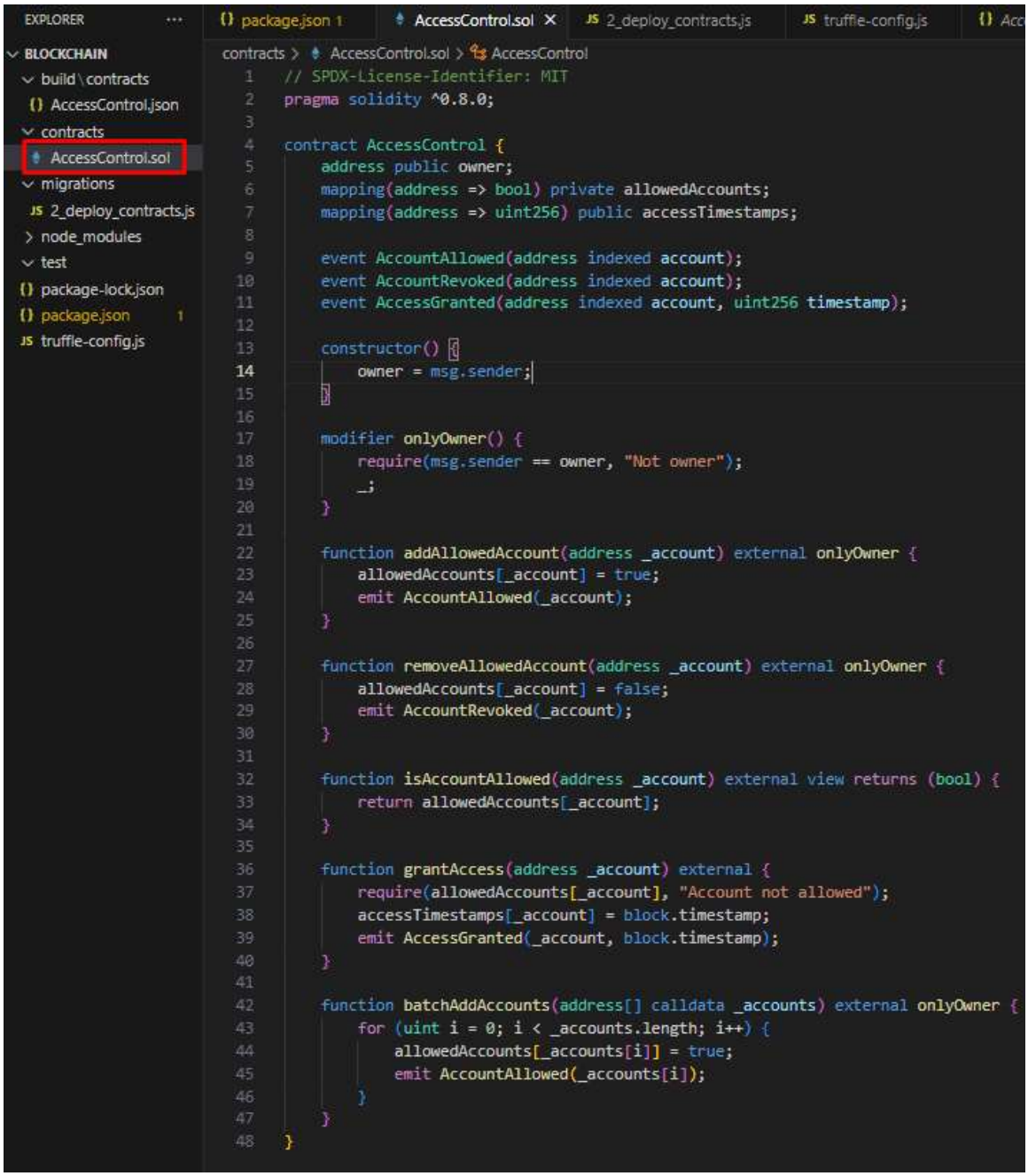


Dentro del entorno de pruebas de cadena de bloques, es necesario crear los componentes del contrato inteligente; mediante la consola se ejecuta el comando "*truffle init*", creando los

componentes del framework Truffle, al crear y modificar el archivo "*AccessControl.sol*", se establecen los parámetros y configuraciones del contrato inteligente de despliegue, para validar las direcciones de cuenta Ethereum como se muestra en la Figura 57.

FIGURA 57

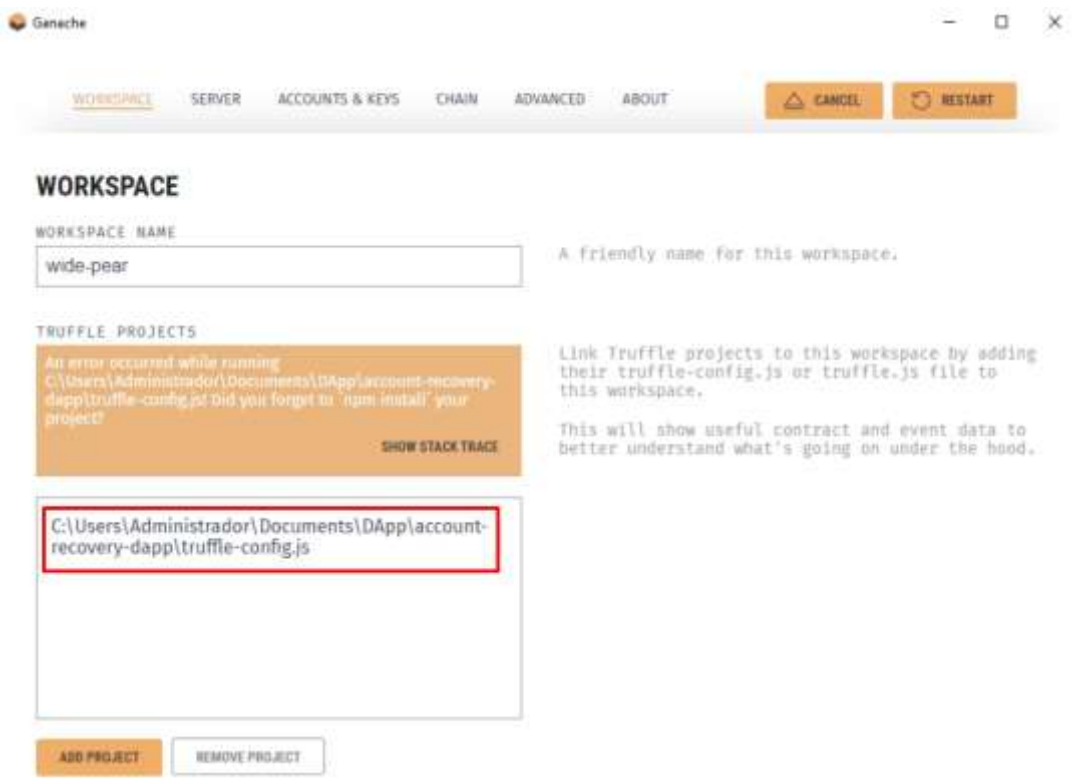
CONTENIDO DEL CONTRATO INTELIGENTE EN SOLIDITY



De igual manera, se configura del archivo *Truffle-config.js*, el cual permite desplegar el contrato dentro de la cadena de bloques de prueba y con la interacción de entornos y herramientas que permite Visual Studio Code, se modifican los archivos y scripts necesarios para la plataforma Truffle y Ganache. Para usar inicialmente el entorno de cadena de bloques, es necesario enlazar el archivo *Truffle-config.js* con la plataforma Ganache, agregando la ruta del directorio que está resaltado en rojo en la Figura 58; para que su despliegue permita generar transacciones iniciales en la cadena de bloques de prueba, de acuerdo con lo establecido en el contrato inteligente. El despliegue del contrato inteligente y configuración del entorno de cadena de bloques se amplía en el Anexo 4.8 donde se configura el archivo "*1_depoy_contracts.js*"; el cual llama, recupera y despliega los procesos establecidos en el archivo "*AccountRegistry.sol*".

FIGURA 58

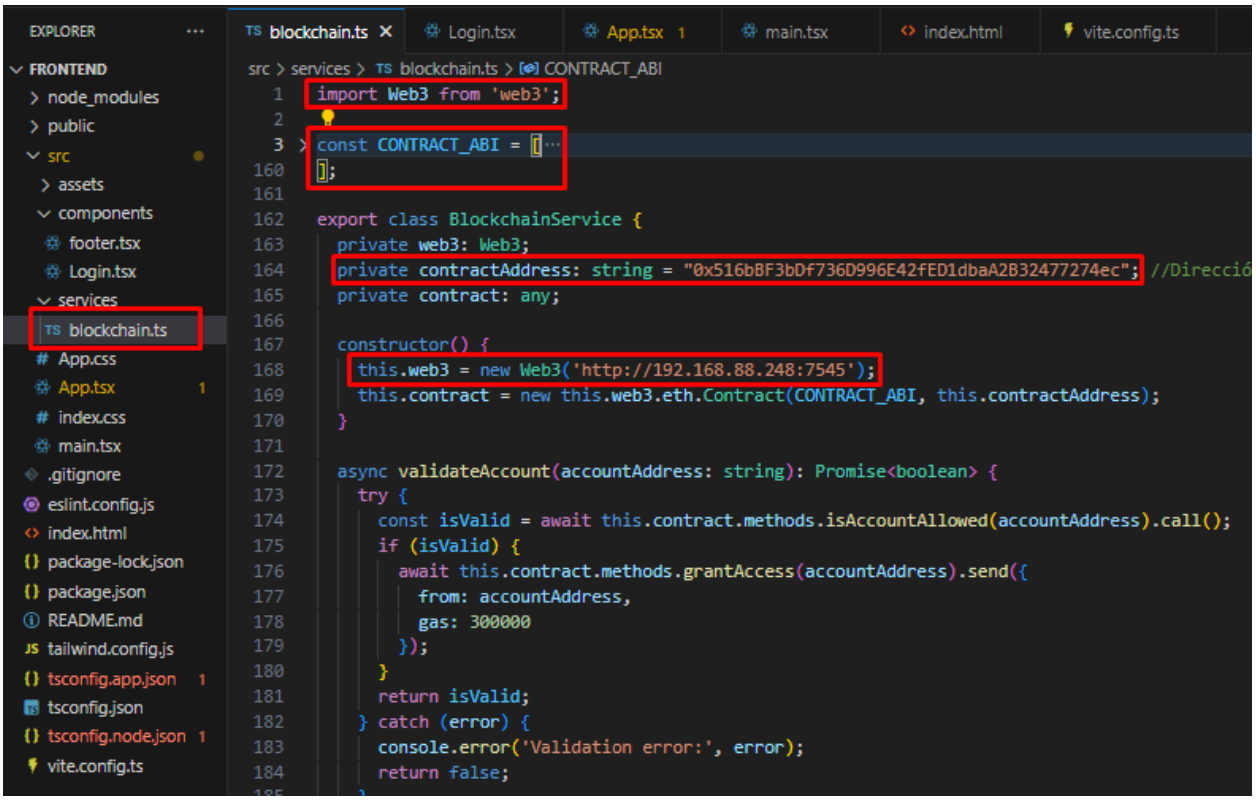
CONFIGURACIÓN DE ARCHIVO TRUFFLE-CONFIG.JS EN GANACHE



Para el despliegue de la interfaz gráfica de conexión y acceso se instalaron los paquetes de la plataforma Vite React con TypeScript, para posteriormente crear un proyecto de desarrollo, el cual genera una plantilla de archivos necesarios y requeridos que se modifican de acuerdo con el uso y necesidades de despliegue de la interfaz como se indica en el Anexo 4.9. Adicional a los archivos y directorios creados por Vite React, se crea el archivo "*blockchain.ts*", el cual de igual manera recupera información de la cadena de bloques generada y usa la librería "*web3*" para validar que las direcciones de cuentas que se ingresan en la interfaz web sean las mismas que se desplegaron en la plataforma Ganache, enlazándose mediante la dirección de contrato y el atributo ".abi", generadas al migrar el archivo "*AccessControl.sol*", además de la dirección y puerto del servidor Ganache, como se resaltan en la Figura 59.

FIGURA 59

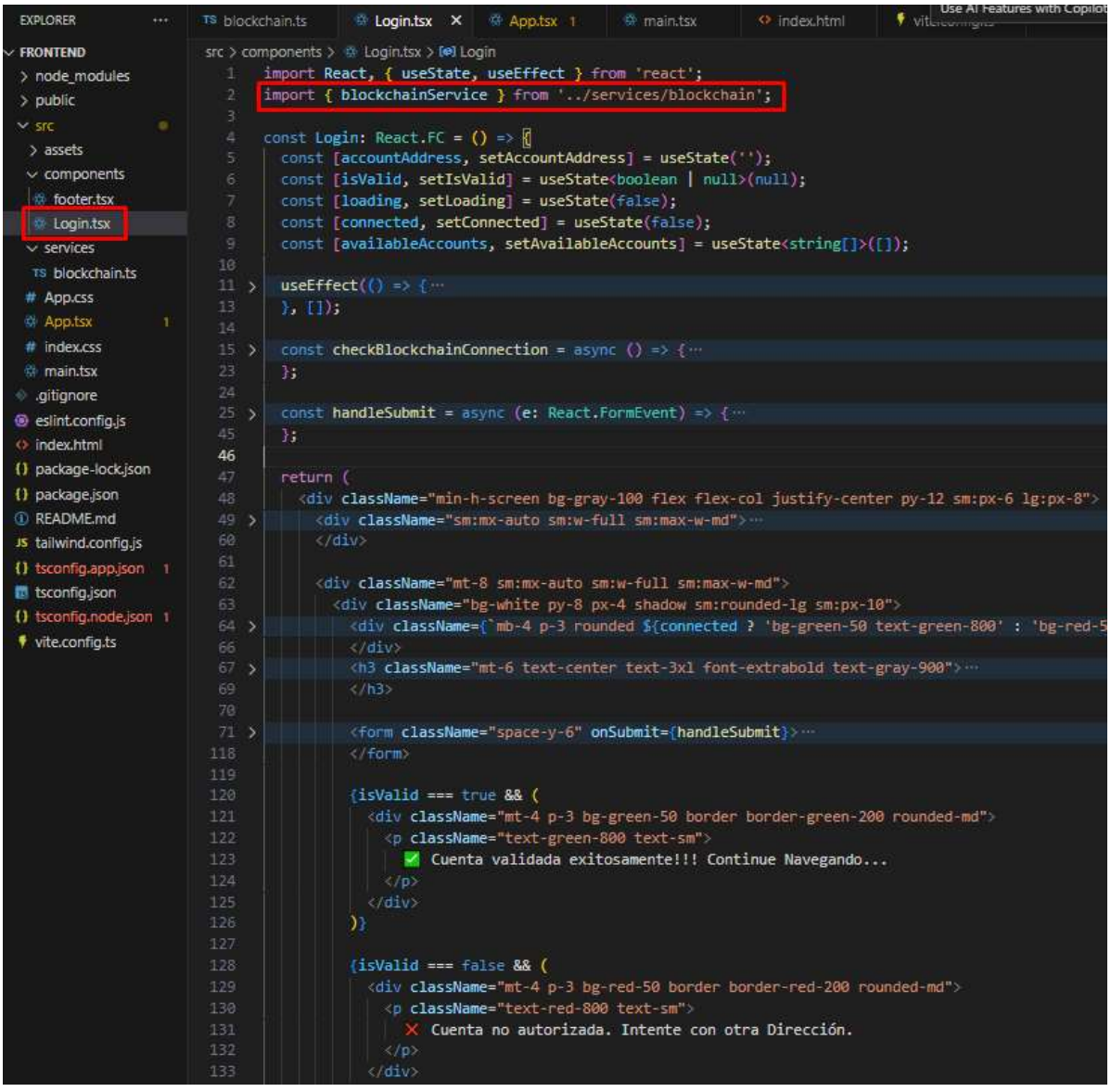
ARCHIVO BLOCKCHAIN.TS



Uno de los elementos de la pantalla principal de la interfaz web, se configura en el archivo "*Login.tsx*", el cual recupera información del archivo "*blockchain.ts*" para permitir el ingreso y validación de las direcciones de cuenta de la cadena de bloques Ganache y muestra en pantalla los mensajes de error, ingreso de datos incorrectos o de la validación realizada; su contenido se visualiza en la Figura 60.

FIGURA 60

ARCHIVO LOGIN.TSX

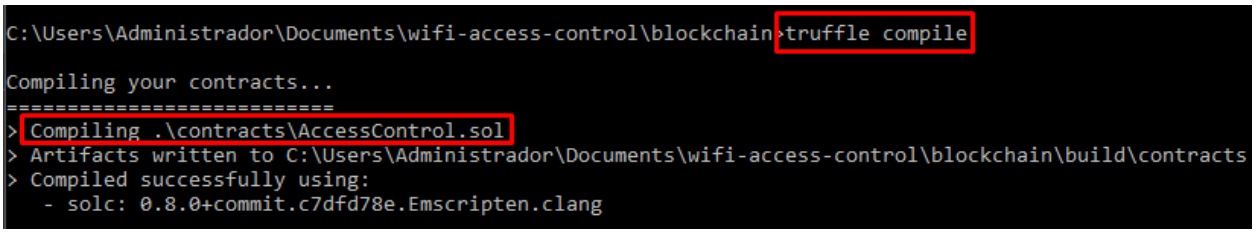


4.1.3.1. Resultado Test 3

Inicialmente se realiza una prueba de compilación del contrato inteligente con el comando "*truffle compile*" este verifica los componentes y construye la ruta "build" para desplegar el contrato, esto se observa en la Figura 61; donde Truffle hace una comprobación previa de componentes y errores y despliegan un mensaje de consola.

FIGURA 61

PRUEBA DE COMPILACIÓN DE CONTRATO INTELIGENTE



A continuación, como parte de la prueba de despliegue del contrato inteligente, se ejecuta el comando "*truffle migrate –network development*". Este proceso se aprecia en la Figura 62, y es el resultado de la compilación del contrato inteligente, muestra los parámetros de inicialización de la migración del contrato, las características de despliegue y los atributos de la transacción inicial realizada; el valor del componente resaltado "*contract address*", es utilizado para enlazar la interfaz web con el entorno Ganache.

FIGURA 62

MIGRACIÓN DE CONTRATO INTELIGENTE

```
C:\Users\Administrador\Documents\wifi-access-control\blockchain>truffle migrate --network development

Compiling your contracts...
===========================
> Compiling .\contracts\AccessControl.sol
> Artifacts written to C:\Users\Administrador\Documents\wifi-access-control\blockchain\build\contracts
> Compiled successfully using:
   - solc: 0.8.0+commit.c7dfd78e.Emscripten.clang


Starting migrations...
======================
> Network name:    'development'
> Network id:      5777
> Block gas limit: 6721975 (0x6691b7)


2_deploy_contracts.js
=====================

   Replacing 'AccessControl'
   -------------------------
   > transaction hash:    0x3f54978d423b07471763838dbe04ac5fa088c0a8821c2420017c23060960955d
   > Blocks: 0            Seconds: 0
   > contract address:    0x516bBF3bDf736D996E42fED1dbaA2B32477274ec
   > block number:        1
   > block timestamp:     1769517759
   > account:             0x0dfb18E371e1C73F7a36fee2d2780672a2fe2d15
   > balance:             99.998722167625
   > gas used:            378617 (0x5c6f9)
   > gas price:           3.375 gwei
   > value sent:          0 ETH
   > total cost:          0.001277832375 ETH
```

Dentro de la interfaz gráfica del entorno de cadena de bloques, podemos observar en la Figura 63; que después de migrar el contrato inteligente este se mostrara como desplegado y se enlazara con las direcciones de cuenta disponibles en el entorno Ganache.

FIGURA 63

CONTRATO DESPLEGADO EN GANACHE

Finalmente se prueba el acceso web a la interfaz gráfica desarrollada y configurada en el Servidor Windows; la cual permite la interacción, acceso y validación de cuentas del entorno de cadena de bloques de manera simplificada; mediante un campo de texto para el ingreso de una dirección válida de la cadena de bloques desplegada y un botón de conexión para validar la información ingresada; en la Figura 64 se visualiza el acceso usando un navegador web y la respectiva dirección IP del servidor.

FIGURA 64

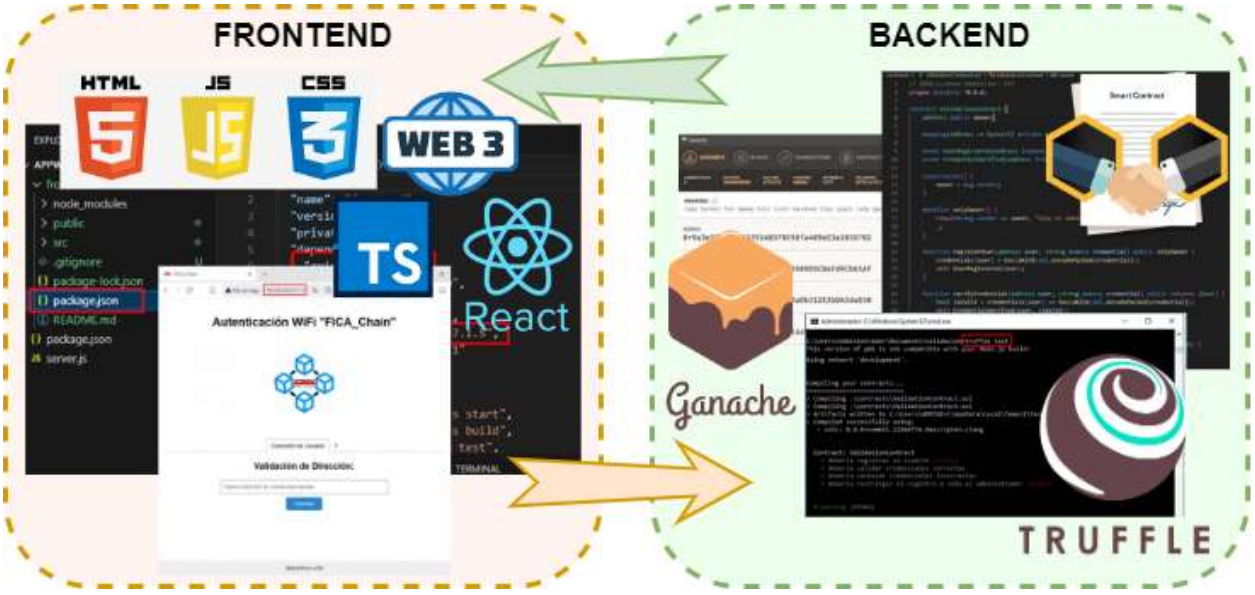
ACCESO A LA INTERFAZ WEB

4.1.4. TEST 4: Integración dirección de cuenta y usuario

En esta primera etapa correspondiente a la evaluación de integración; se interconecta el entorno de pruebas de cadena de bloques, con la interfaz de acceso web para asignar y configurar el acceso y validación de las direcciones de cuenta del entorno de cadena de bloques; mediante el contrato inteligente y la interfaz web, o la interacción entre el *frontend* y el *backend*; como se representa en el diagrama de la Figura 65.

FIGURA 65

INTEGRACIÓN DE FRONTEND Y BACKEND



4.1.4.1. Resultado Test 4

El enlace entre el entorno pruebas de la cadena de bloques y la interfaz web permite gestionar el ingreso de las credenciales de usuarios (direcciones de cuentas válidas), para autenticarse y acceder al recurso de red del entorno inalámbrico de pruebas de la facultad. En la Figura 66 se observa el ingreso y validación exitosa de una dirección de cuenta válida que forma

parte del entorno de cadena de bloques Ganache, mediante un mensaje que muestra que la dirección de cuenta ha sido validada y se permite el acceso al recurso de red.

FIGURA 66

INGRESO DE DIRECCIÓN DE CUENTA VÁLIDA

De igual manera se comprueba el acceso a la interfaz desde un dispositivo móvil, en este caso se observa en la Figura 67, la adaptabilidad de la interfaz y el correcto funcionamiento para la validación, mostrando el respectivo mensaje de validación.

FIGURA 67

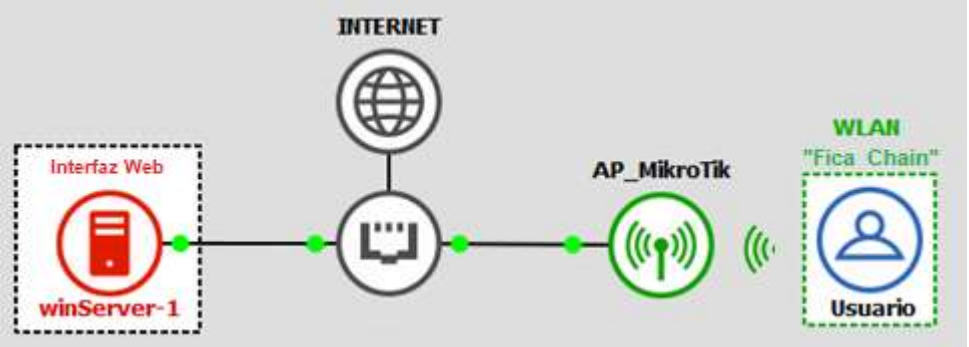
ACCESO A INTERFAZ WEB DESDE DISPOSITIVO

4.1.5. TEST 5: Autenticación de usuario

En esta etapa de integración, se evalúa básicamente la capacidad del mecanismo de autenticación propuesto, para validar la conexión e integrar la tecnología de cadena de bloques con el estándar IEEE 802.11, permitiendo que los usuarios puedan conectarse a los recursos de red, después de ingresar las credenciales correspondientes en la interfaz de acceso mediante sus dispositivos; este proceso es crítico ya que tanto el entorno de cadena de bloques, el protocolo WiFi, la interfaz gráfica de validación y conexión deben converger. En la Figura 68 se observa los elementos del entorno de pruebas planteado para el mecanismo de autenticación.

FIGURA 68

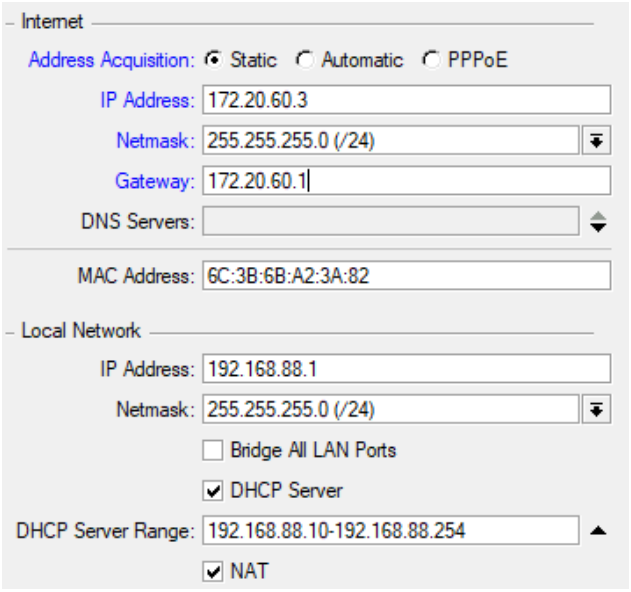
ELEMENTOS DEL ENTORNO DE PRUEBAS



La configuración del punto de acceso inalámbrico MikroTik, se realiza asignando las respectivas direcciones IP, para que exista la conectividad entre el servidor con el entorno de cadena de pruebas Ganache y la red inalámbrica propagada por el equipo MikroTik. En la Figura 69 se observan las configuraciones de red local, servicio DHCP y NAT para garantizar el acceso al servidor desde la red inalámbrica; en el Anexo 4.10 también se detallan estas configuraciones.

FIGURA 69

CONFIGURACIÓN DE RED AP MIKROTIK



Usando la terminal de comando del equipo MikroTik, se realiza la configuración de conexión, para limitar el acceso de la red inalámbrica como lo muestra la Figura 70, de manera que se redireccione la conexión al servidor que contiene la interfaz web, al momento de que un equipo se conecte al SSID propagado del entorno de pruebas.

FIGURA 70

CONFIGURACIÓN DE CONEXIÓN A SERVIDOR

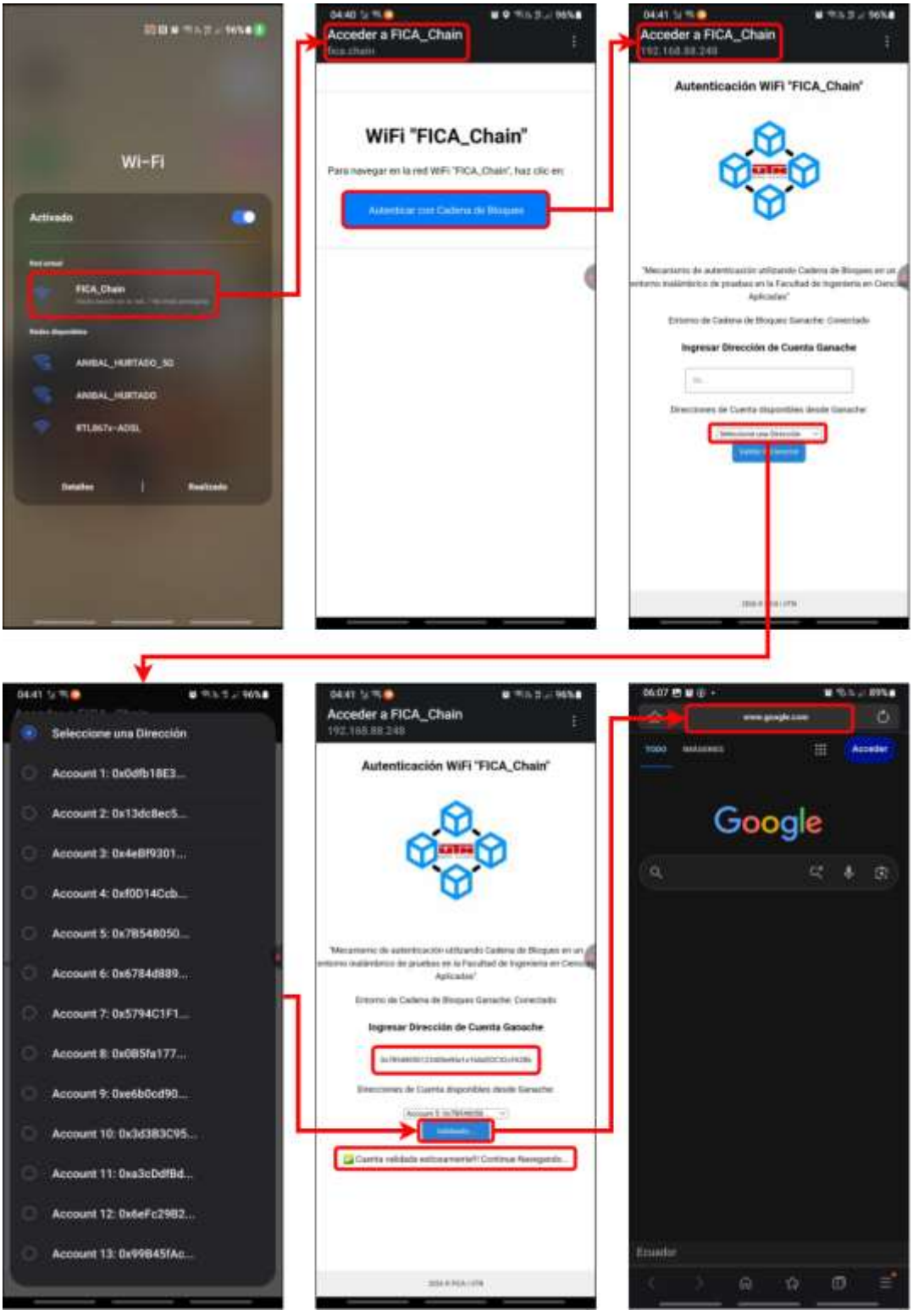
```
/ip hotspot add name=hotspot1 interface=bridge-local address-pool=static-only disabled=no
/ip hotspot network add address=192.168.88.0/24 gateway=192.168.88.1 dns-server=8.8.8.8
/ip hotspot profile set default hotspot-address=192.168.88.1 html-directory=hotspot login-by=http-ch
ap,mac-cookie
```

4.1.5.1. Resultado Test 5

El proceso de conexión y acceso al recurso de red mediante el mecanismo de autenticación propuesto se indica en la Figura 71; iniciando al momento de buscar entre los SSID disponibles el que se ha configurado para el entorno inalámbrico de pruebas, comprobando que en las redes inalámbricas disponibles se muestra el identificador “FICA_Chain”, que es una red abierta de acceso limitado; después de que el dispositivo se conecte a la red, se redirige automáticamente a una interfaz inicial que indica que se debe realizar un proceso de autenticación al presionar un botón; al dar clic se muestra la interfaz web principal del mecanismo de autenticación, mostrando que el entono de cadena de bloques se encuentra conectado y el campo donde se debe escribir o seleccionar una dirección de cuenta válida; posterior a ingresar o seleccionar una dirección de cuenta se presiona el botón para validar, el cual indicara si la dirección de cuenta es válida y se puede acceder al recurso de red; continuando así con el proceso redirigiendo el navegador web de la interfaz hacia el recurso de red o dirección de navegación web configurada; finalizando el proceso, el cual también se indica en el Anexo 5.

FIGURA 71

VERIFICACIÓN MECANISMO DE AUTENTICACIÓN



4.1.6. TEST 6: Autenticación de usuario sin credenciales

La configuración de la validación de dirección de cuentas del entorno de cadena de bloques se realiza cuando se ingresan direcciones de cuenta válidas y que se hayan desplegado en el entorno de cadena de bloques Ganache al que se enlaza la interfaz web, de tal manera que el ingreso de direcciones de cuneta que no consten o no hayan desplegado contaran como no validas y el mecanismo no realizará el proceso de validación, mostrando el respectivo mensaje de error.

4.1.6.1. Resultado Test 6

Para el desarrollo de esta prueba de seguridad, se plantea evaluar el funcionamiento del mecanismo, ingresando credenciales o datos no válidos en el campo de ingreso de dirección de cuenta en la interfaz; en la Figura 72 se observa la validación sin ingresar ningún tipo de datos, lo que muestra un mensaje de "*Error: Dirección de Cuenta Inválida*".

FIGURA 72

VALIDACIÓN SIN INGRESO DE DATOS

La verificación del mecanismo de validación ingresando datos de una dirección de cuenta válida, solo cambiando el último carácter por otro para alterar la dirección de cuenta, da como resultado el mismo mensaje de "*Error: Dirección de Cuenta Inválida*", como se observa en la Figura 73.

FIGURA 73

VALIDACIÓN DE DIRECCIÓN DE CUENTA ERRÓNEA

4.1.7. TEST 7: Comportamiento de entorno de cadena de bloques

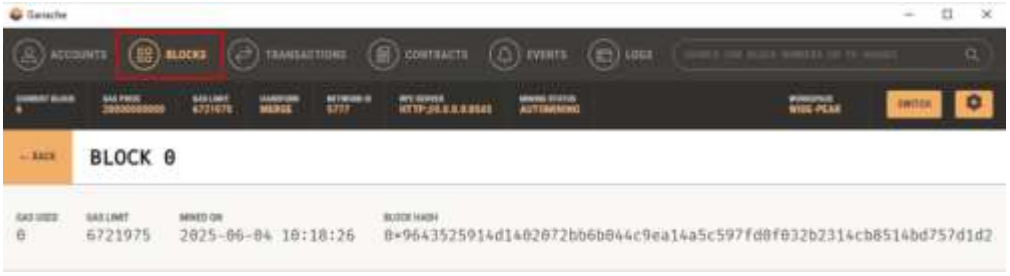
La segunda etapa de las pruebas de seguridad consiste en observar el comportamiento del entorno de cadena de bloques Ganache; al momento de realizar la validación de direcciones de cuentas ingresadas para la autenticación. Como ya se explicó en la Sección 4.1.6, cuando se ingresan direcciones de cuenta erróneas y se intentan validar, el mecanismo muestra el respectivo mensaje de error y no se realiza ninguna transacción, por lo que el análisis del comportamiento de la cadena de bloques se enfocará para los casos en los que se ingresen direcciones de cuenta válidas, que formen parte de las desplegadas en la cadena de bloques de prueba y que al momento de validar generen la respectiva transacción.

4.1.7.1. Resultado Test 7

Inicialmente, al generar el entorno de cadena de bloques en Ganache, se debe realizar la configuración del contrato inteligente, la compilación, la conexión con el archivo "*truffle-config.js*" y la posterior migración; al momento que finaliza la migración se observa en la interfaz gráfica del entorno Ganache en la sección de Bloques, la generación de un Bloque 0 correspondiente a la inicialización del entorno de pruebas, que se presenta en la Figura 74.

FIGURA 74

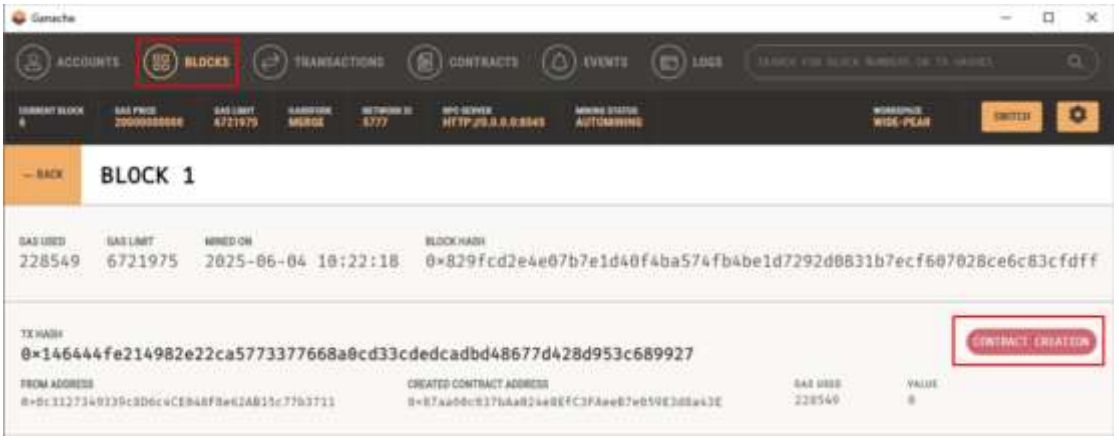
BLOQUE 0 DEL ENTORNO GANACHE DE PRUEBAS



De igual manera se genera un Bloque 1, el cual corresponde a la creación de la dirección del contrato como se resalta en la Figura 75, luego de que se ha compilado y se ha migrado.

FIGURA 75

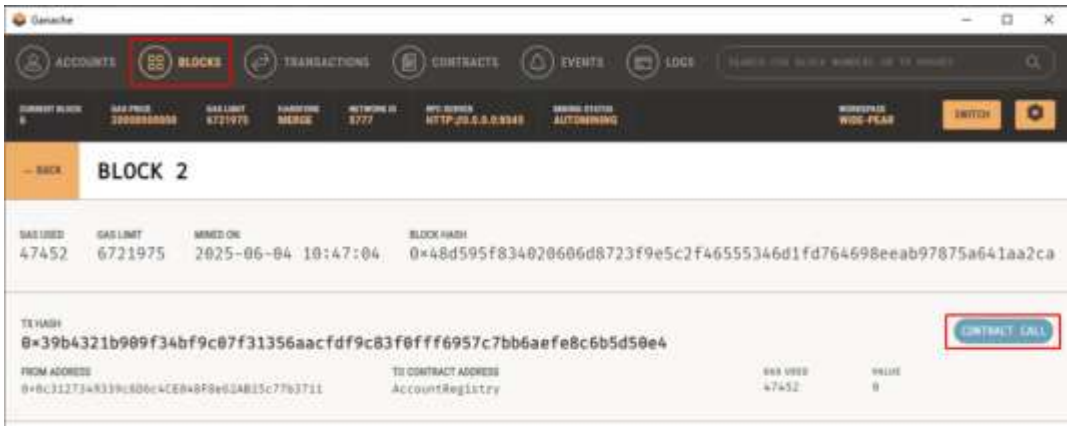
CREACIÓN DE DIRECCIÓN DE CONTRATO



Al momento de ingresar y validar direcciones de cuentas válidas en la interfaz web, se realizan transacciones y se generan bloques adicionales los cuales llaman y consultan al contrato inteligente configurado, como lo indica en la Figura 76.

FIGURA 76

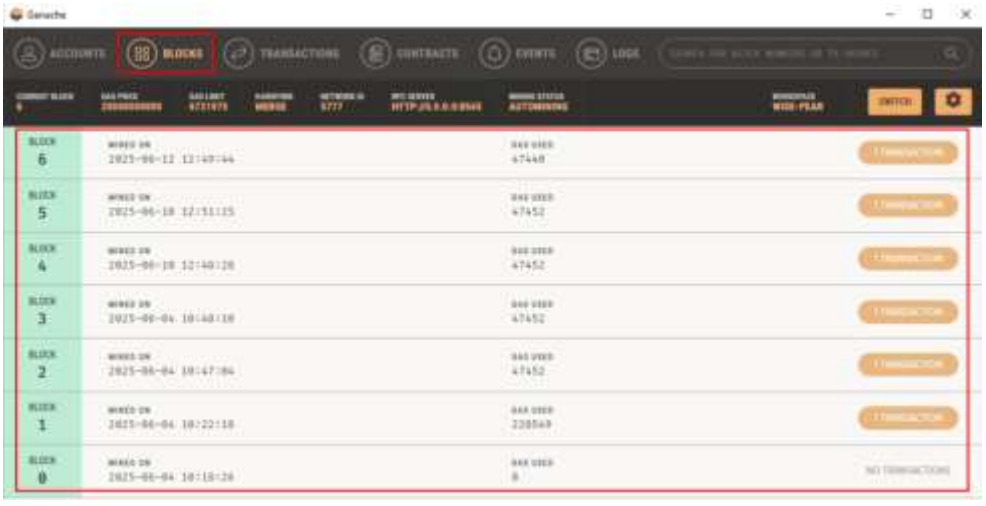
LLAMADA Y CONSULTA DE CONTRATO INTELIGENTE



Cada validación genera bloques, los cuales se asocian con su respectiva transacción, como se visualiza en la Figura 77.

FIGURA 77

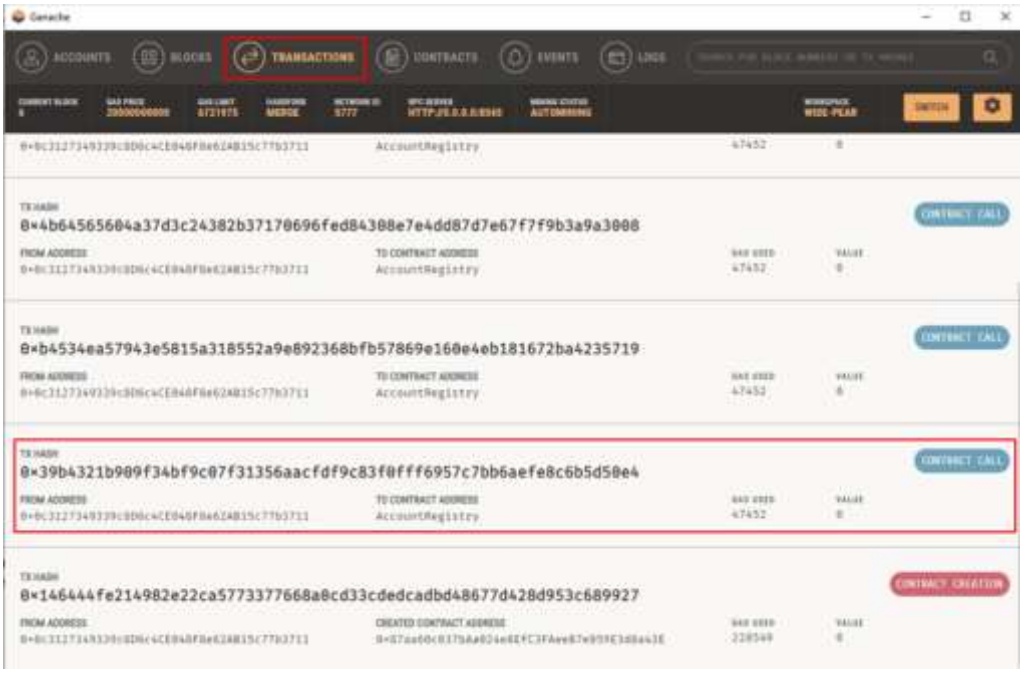
GENERACIÓN DE BLOQUES



Cada bloque se asocia a transacción, que a su vez se conecta a una dirección de cuenta, en la Figura 78 se puede observar estos detalles cuando se revisan las transacciones generadas.

FIGURA 78

TRANSACCIONES GENERADAS



Como inicialmente se asocia la ruta del directorio que contiene los componentes del contrato inteligente, en la Figura 79 se resalta esta ruta, además nos indica la dirección de contrato generada y su estado de despliegue.

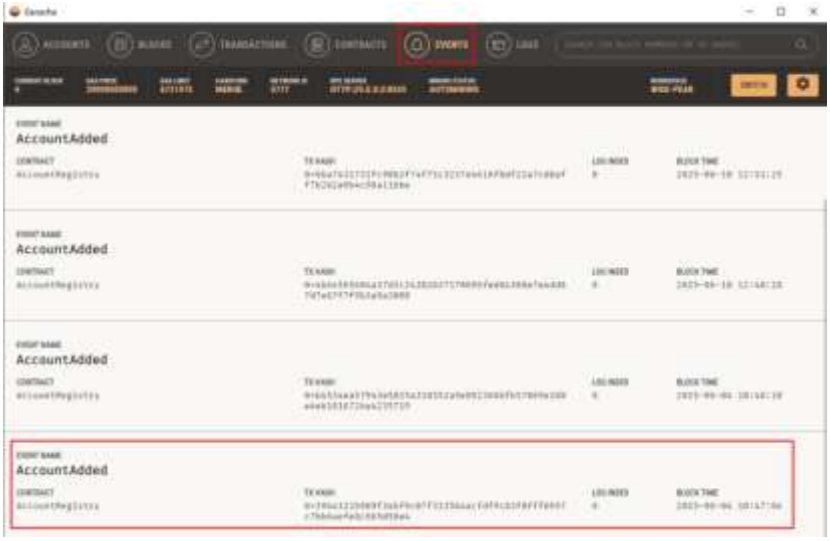
FIGURA 79

REVISIÓN DE CONTRATO EN GANACHE

Por otra parte, al revisar la sección de eventos también se identifican las transacciones generadas asociadas al tiempo de bloque, lo que interconecta la generación de bloques con las transacciones, representado en la Figura 80.

FIGURA 80

EVENTOS REGISTRADOS EN GANACHE



Finalmente, dentro de la sección de logs, encontramos las salidas por consola de la todas las transacciones y bloques, que se han generado desde el despliegue del entorno, mostrado en la Figura 81.

FIGURA 81

REGISTRO DE LOGS EN GANACHE

4.1.8. TEST 8: Conectividad y autenticación

En esta sección se evaluará el comportamiento del entorno de pruebas desplegado; usando el mecanismo de autenticación propuesto y la conexión, validación y acceso desde diferentes tipos de equipos de usuario.

4.1.8.1. Resultado Test 8

La conexión mediante el mecanismo de autenticación propuesto se realizó desde diferentes dispositivos los cuales se conectan al SSID "FICA_Chain", propagado por el punto de acceso inalámbrico, para lo cual se analiza el intercambio de tramas durante una conexión con una red inalámbrica de área local (WLAN), la Tabla 18 resume el tipo de tramas que usa el protocolo IEEE 802.11 en el proceso de conexión inalámbrico entre un dispositivo de usuario y el punto de acceso (Rowell, 2018).

TABLA 18

TRAMAS DE CONEXIÓN WLAN

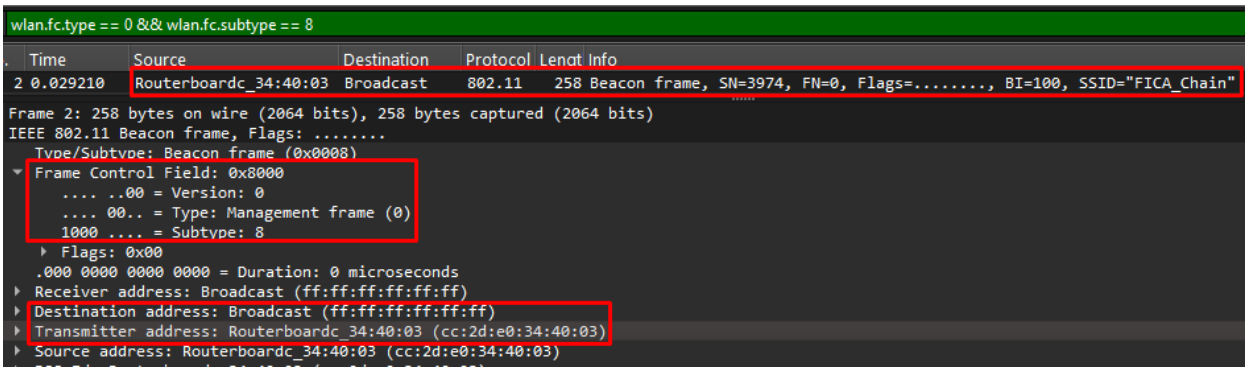
Trama IEEE 802.11	Tipo	Subtipo
Beacon	00	1000
Probe Request	00	0100
Probe Response	00	0101
Authentication Request	00	1011
Authentication Response	00	1011
Association Request	00	0000
Association Response	00	0001

Nota: La tabla muestra un orden resumido de las tramas que se intercambian durante el proceso de conexión WiFi junto con los valores de bits que identifican a cada trama en el campo "*Frame Control*". Tomado y adaptado de (Rowell, 2018).

De manera que, para la conexión del mecanismo de autenticación propuesto, se busca determinar la presencia de las diferentes tramas que se propagan e intercambian durante el proceso de conexión inalámbrica, en la Figura 82 se muestra la trama "*Beacon*" que anuncia la red "FICA_Chain" mediante broadcast e identifica el punto de acceso inalámbrico MikroTik mediante la dirección MAC "Routerboard_34:40:03 (cc:2d:e0:34:40:03)".

FIGURA 82

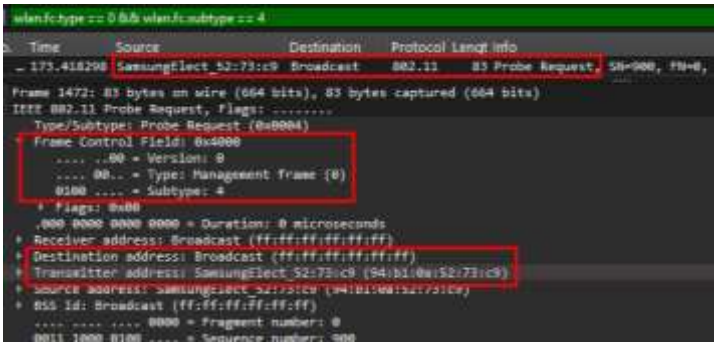
TRAMA BEACON PROPAGADA POR "FICA_CHAIN"



De forma similar los dispositivos inalámbricos escanean las redes enviando tramas "*Probe Request*" mediante broadcast y se identifican mediante su dirección MAC, para este caso se usa "SamsungElect_53:73:c9 (94:b1:0a:52:73:c9)", como se indica en la Figura 83.

FIGURA 83

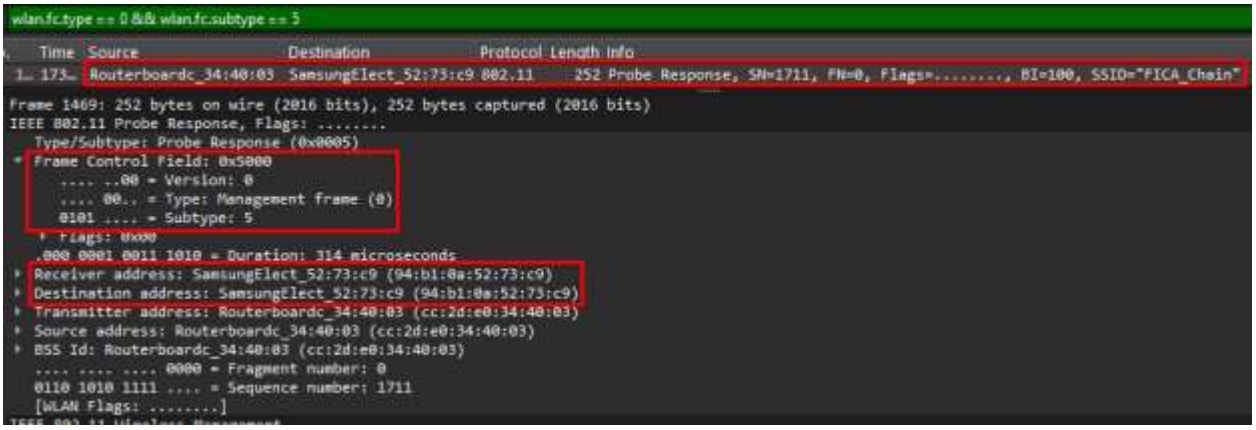
TRAMA PROBE REQUEST PROPAGADA POR DISPOSITIVO



La trama "*Probe Response*" identificada en la Figura 84, es la respuesta del punto de acceso Routerboard_34:40:03 al dispositivo SamsungElect_53:73:c9.

FIGURA 84

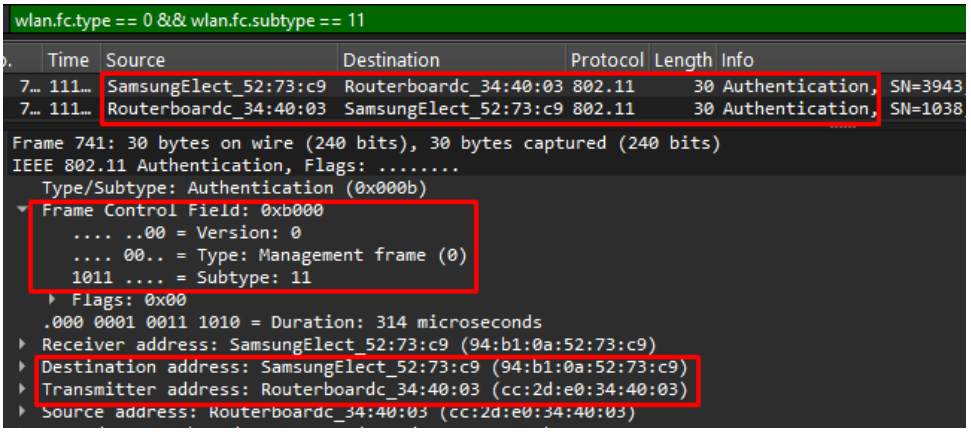
TRAMA PROBE RESPONSE PROPAGADA POR EL PUNTO DE ACCESO



El dispositivo SamsungElect_53:73:c9 solicita permiso para conectarse mediante la trama "*Authentication Request*" (Cliente - AP), mientras que el punto de acceso responde enviando una trama "*Authentication Response*" (AP -Cliente), esta trama tiene el mismo valor de subtipo, como se muestra en la Figura 85.

FIGURA 85

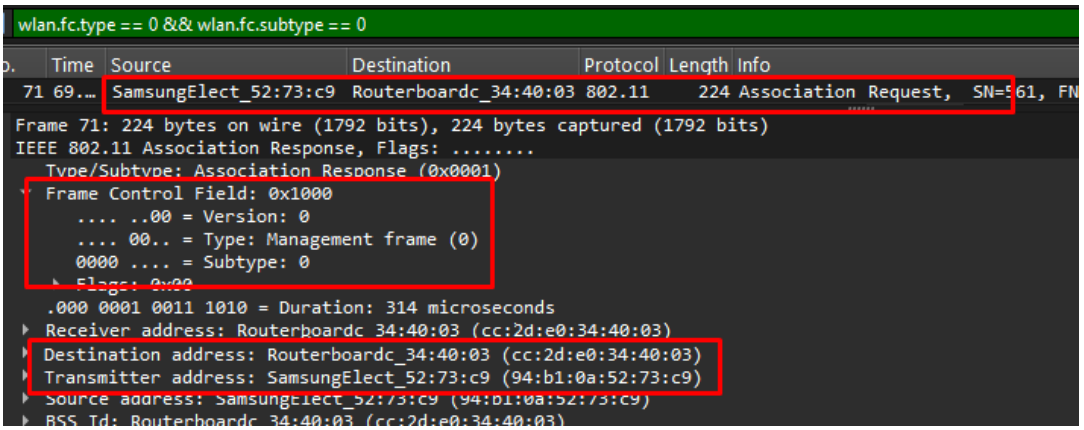
TRAMAS AUTHENTICATION REQUEST Y AUTHENTICATION RESPONSE



Finalmente, el establecimiento de la conexión desde el dispositivo SamsungElect_53:73:c9 hacia el punto de acceso Routerboard_34:40:03 mediante la trama "*Association Request*" se muestra en la Figura 86.

FIGURA 86

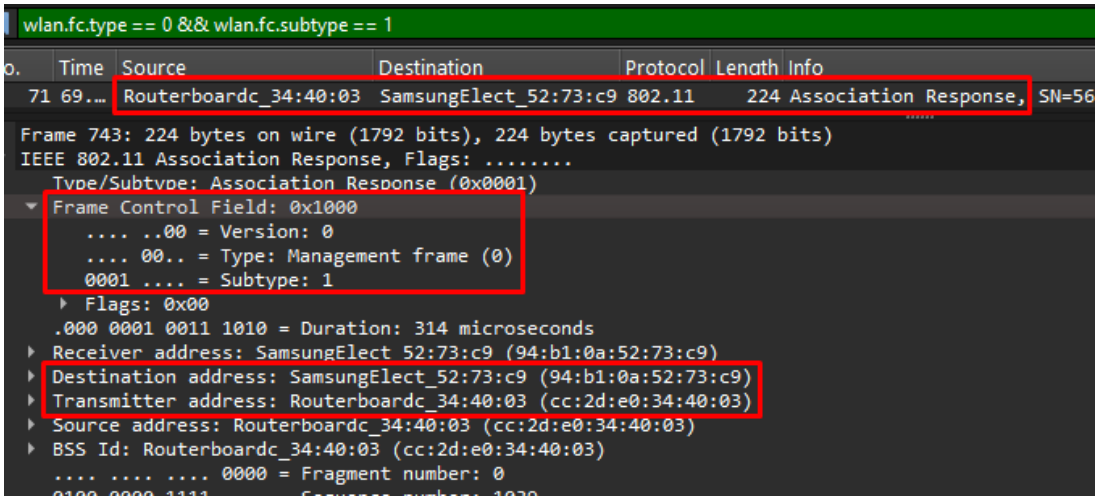
TRAMA ASSOCIATION REQUEST



Mientras que la trama de respuesta "*Association Response*" desde el punto de acceso Routerboard_34:40:03 hacia el dispositivo SamsungElect_53:73:c9, se detalla en la Figura 87.

FIGURA 87

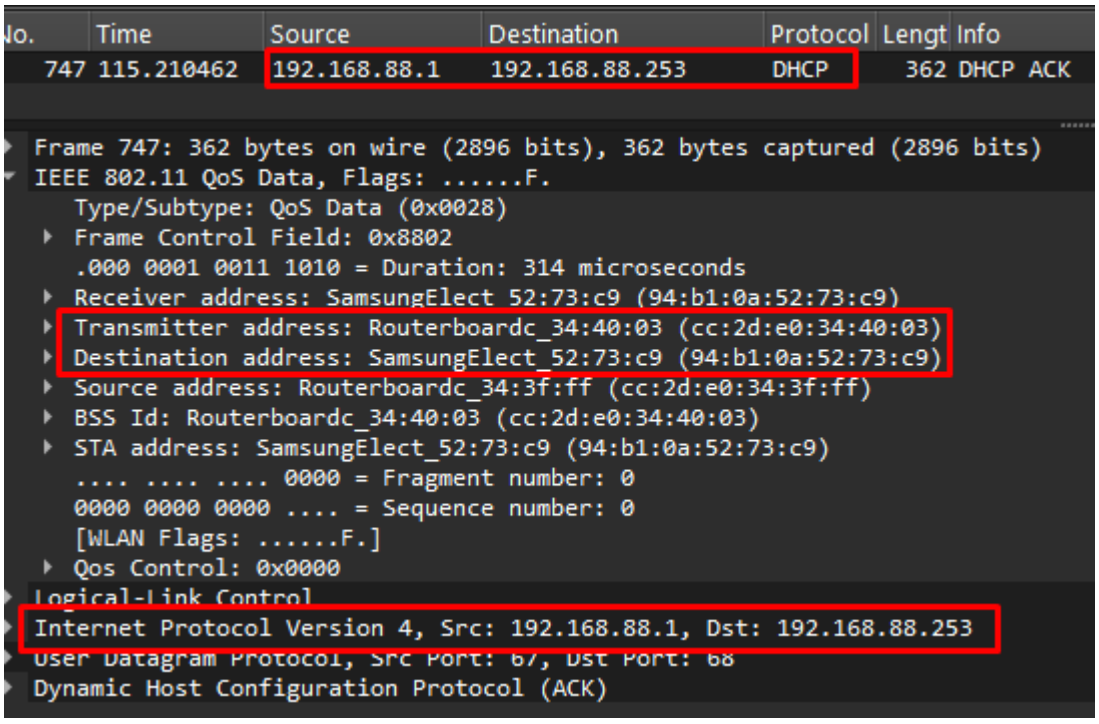
TRAMA ASSOCIATION RESPONSE



Complementando este proceso, también se recupera una trama DHCP para el direccionamiento IP y que se asocia con la dirección MAC del punto de acceso y la del dispositivo de usuario que se resalta en la Figura 88 y permite al dispositivo de usuario comunicarse con el dominio de red propagado por el punto de acceso mediante el SSID "FICA_Chain" mediante el Protocolo de Internet (IP).

FIGURA 88

TRAMA DE ASIGNACIÓN DHCP

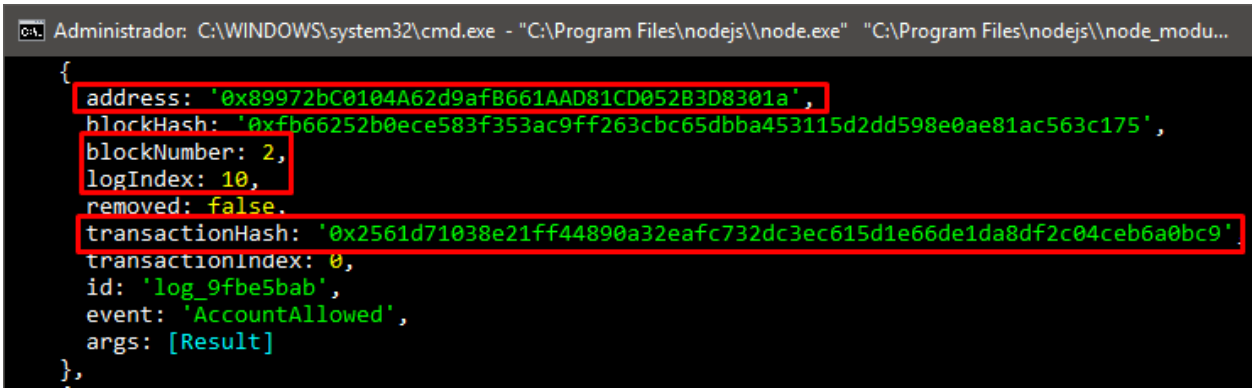


Dentro del proceso de la validación para el mecanismo de autenticación propuesto, se considera el enfoque del entorno de la cadena de bloques, donde se configuran las direcciones de cuenta válidas para ser ingresadas en la interfaz web de autenticación; en la Figura 89 se visualiza la activación de una de varias direcciones de cuenta generadas en el entorno Ganache; que es válida y permite al usuario validarse mediante la interfaz de conexión los atributos a tomar en cuenta son "*address,*

blockNumber, logIndex y *transactionHash*", que permiten buscar e identificar la cuenta a la que perteneces la transacción de activación.

FIGURA 89

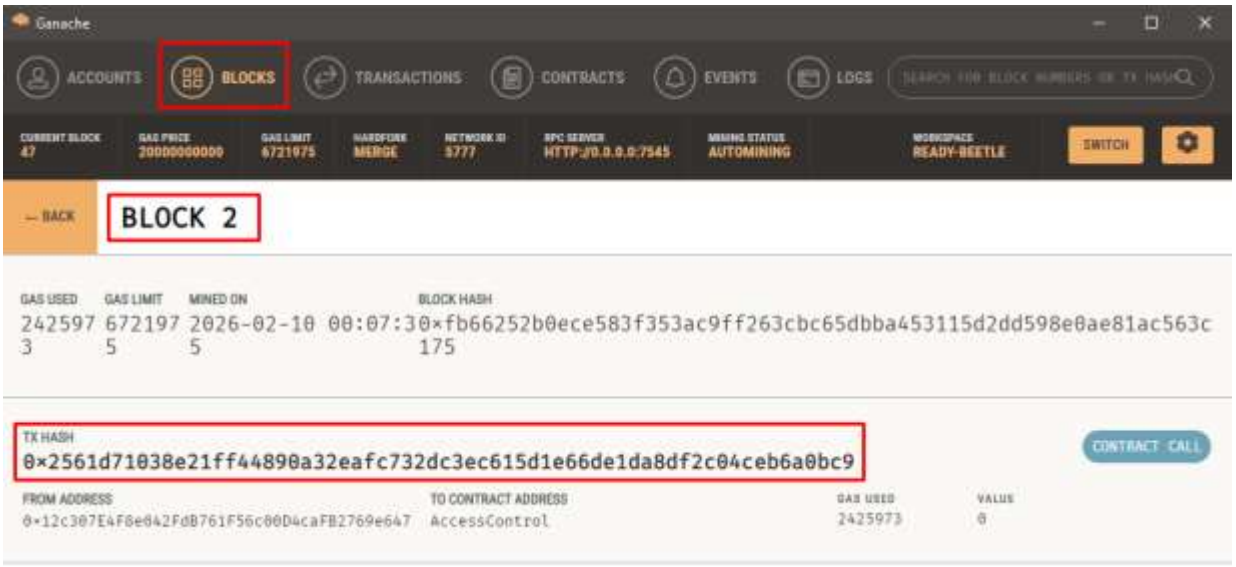
ATRIBUTOS DE DIRECCIÓN DE CUENTA VÁLIDA



Con los datos de la transacción, dentro de la interfaz del entorno Ganache, en la sección de bloques se busca el bloque y el hash de transacción respectivo, como se destaca en la Figura 90.

FIGURA 90

BLOQUE DE DIRECCIÓN DE CUENTA VÁLIDA



El hash de la transacción contine el log con la identificación de las direcciones de cuenta que se activaron y son válidas para usarse en el mecanismo de autenticación, en la Figura 91 se muestra el índice de log correspondiente a la transacción de la cuenta activa y válida.

FIGURA 91

LOG DE DIRECCIÓN DE CUENTA VÁLIDA

El contenido del log de transacción indicado en la Figura 92; muestra que la dirección de cuenta es permitida, el hash de transacción, el índice de log y el identificador de la dirección de cuenta.

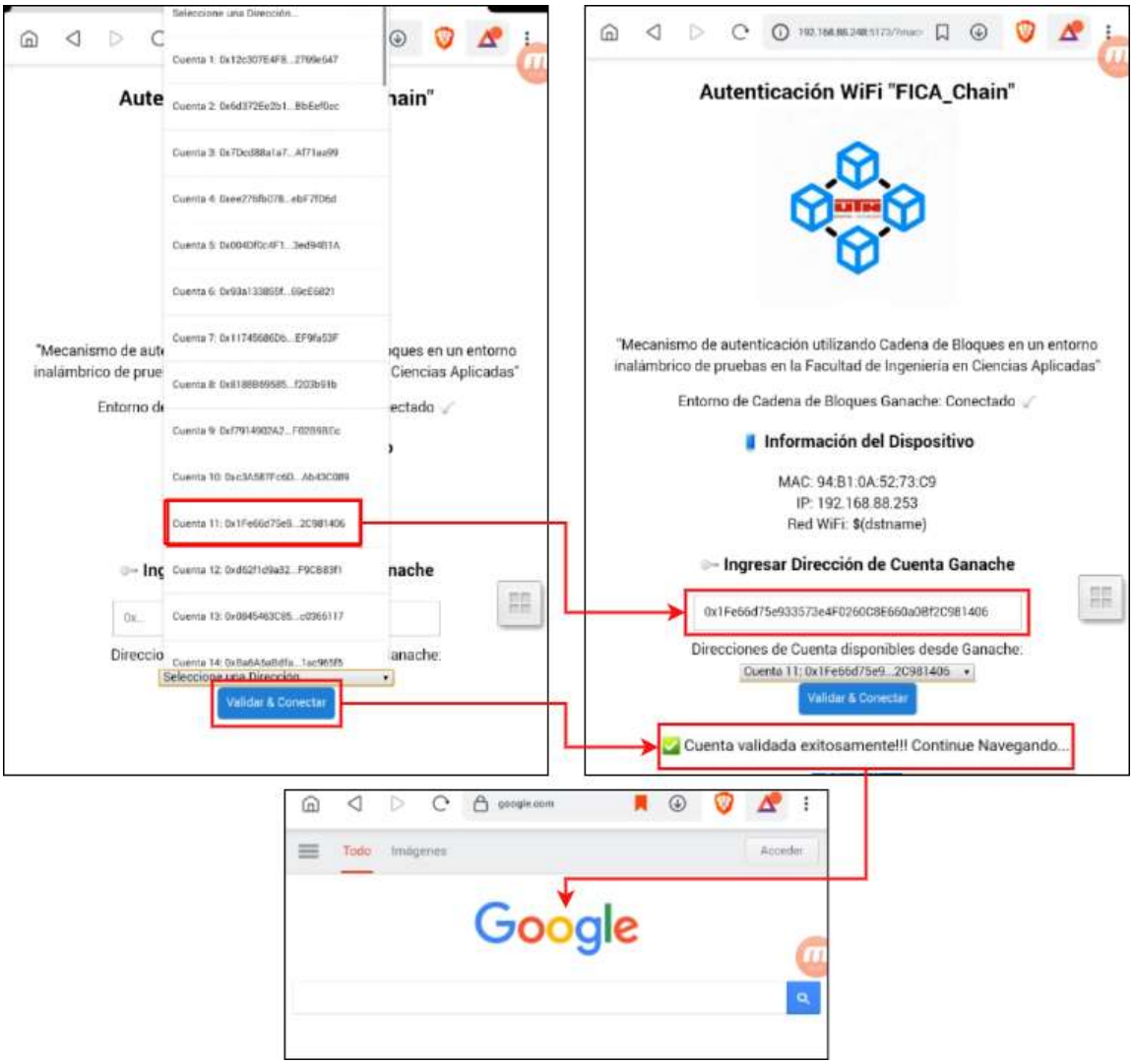
FIGURA 92

INFORMACIÓN DE DIRECCIÓN DE CUENTA VÁLIDA

De forma que usando la dirección de cuenta permitida "*0x1fe66d75e933573e4f0260c8e660a0bf2c981406*", se realiza el proceso de autenticación ingresando y/o seleccionando esta dirección en la interfaz de validación como se observa en la Figura 93.

FIGURA 93

INGRESO DE DIRECCIÓN DE CUENTA VÁLIDA

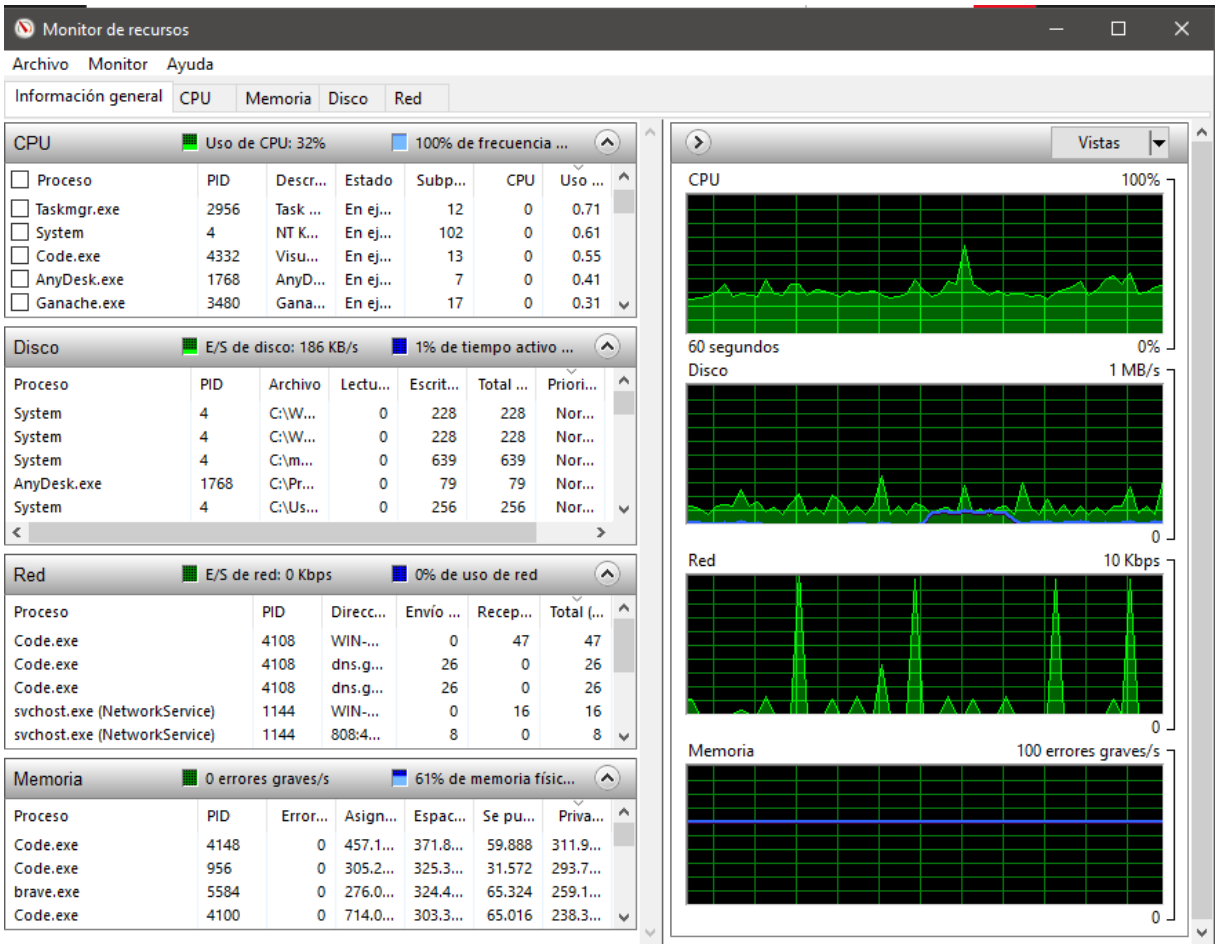


Adicionalmente, se verifica el comportamiento del servidor de cadena de bloques y la interfaz web del mecanismo de autenticación; la Figura 94 muestra el monitor de recursos del equipo al

momento de la conexión y validación de las direcciones de cuenta, demostrando el consumo de recursos de procesamiento y de la interfaz de red, mientras se realizan los procesos en el entorno Ganache y la interfaz de validación.

FIGURA 94

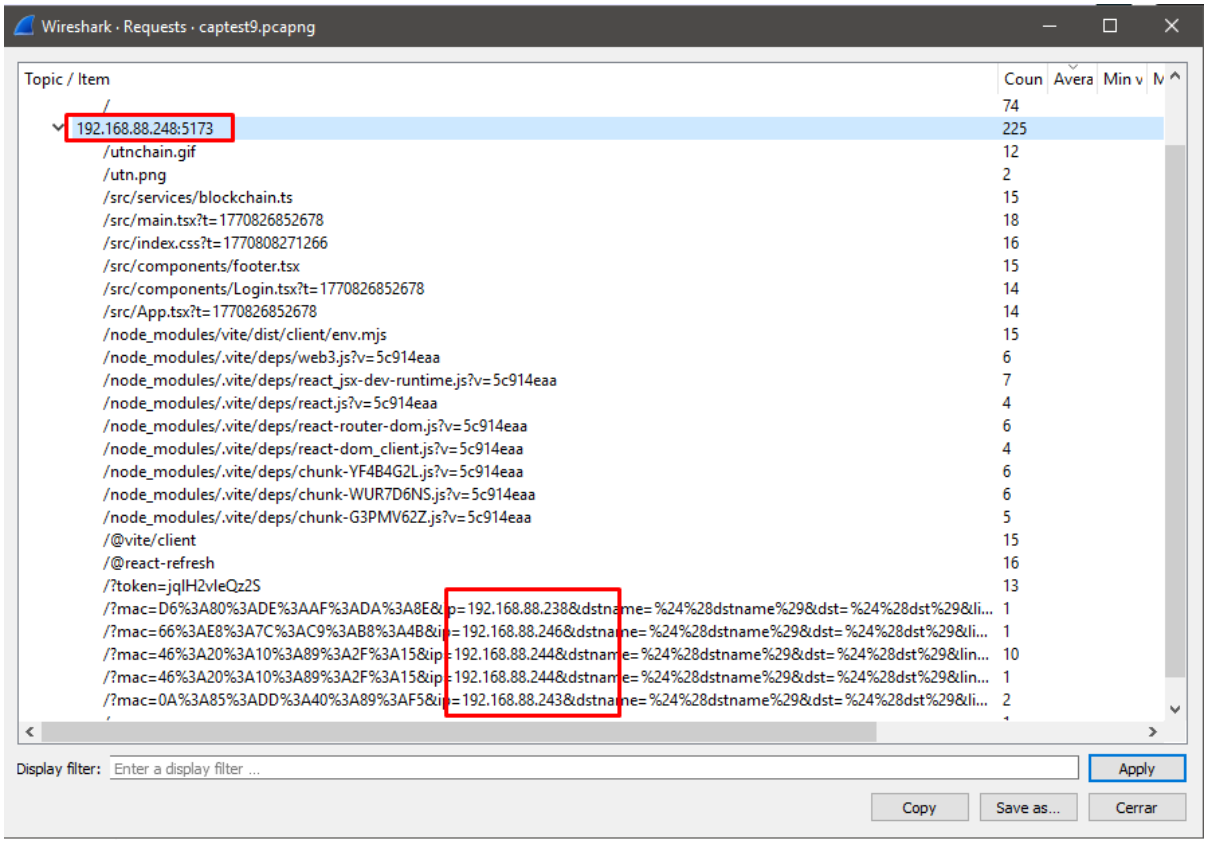
MONITOREO DE RECURSOS DEL SERVIDOR



El acceso de diferentes dispositivos, hacia el servidor con la dirección IP “192.168.88.248” y el puerto “5173” se observa en Figura 95; donde la captura de tráfico en la tarjeta de red del servidor muestra equipos de usuario con dirección IP asignada conectándose a la interfaz web ubicada en el servidor del mecanismo de autenticación propuesto.

FIGURA 95

CONEXIÓN DE USUARIOS CON SERVIDOR



4.1.9. TEST 9: Gestión de conexión

La evaluación de esta etapa comprende la conexión de diferentes usuarios, el tiempo de respuesta y el rendimiento del servidor; durante la validación de cuentas del mecanismo de autenticación. Para obtener estas métricas se plantea el desarrollo de una prueba de concepto, aplicada al entorno de cadena de bloques y el efecto de estos procesos en el uso de los recursos del equipo. Dentro del entorno de pruebas se define y despliega una versión modificada del contrato inteligente detallado en el Anexo 6.1; una sección de este contenido se observa en la Figura 96 y muestra las funciones donde se configuran los grupos o rondas de usuarios y los procesos de carga que se realizan en la prueba de autenticación para la obtención de resultados.

FIGURA 96

CONTRATO INTELIGENTE DE PRUEBA DE CONCEPTO

```
    // ====== Preparar grupo de usuarios ======
    function batchAllow(address[] calldata accounts) external onlyOwner {
        for (uint i = 0; i < accounts.length; i++) {
            allowedAccounts[accounts[i]] = true;
        }
    }

    // ====== Iniciar ronda de prueba ======
    function startRound(uint256 usersTarget) external onlyOwner {
        currentRound++;

        rounds[currentRound] = RoundStats({
            usersTarget: usersTarget,
            usersCompleted: 0,
            startTime: block.timestamp,
            endTime: 0
        });

        roundStartTime = block.timestamp;

        emit RoundStarted(currentRound, usersTarget);
    }

    // ====== Función usada por cada cliente ======
    function benchmarkAuthenticate() external {
        require(allowedAccounts[msg.sender], "Not allowed");

        lastAuthTime[msg.sender] = block.timestamp;

        rounds[currentRound].usersCompleted++;
        totalAuthentications++;

        emit UserAuthenticated(msg.sender, block.timestamp);

        // Cierre automático de ronda
        if (rounds[currentRound].usersCompleted == rounds[currentRound].usersTarget) {
            rounds[currentRound].endTime = block.timestamp;

            uint256 duration = rounds[currentRound].endTime - rounds[currentRound].startTime;

            emit RoundFinished(currentRound, duration);
        }
    }

    function getRoundDuration(uint256 roundId) external view returns (uint256) {
        return rounds[roundId].endTime - rounds[roundId].startTime;
    }
}
```

Usando la consola del Framework Truffle se despliega en el entorno de cadena de bloques el script de la prueba de concepto se especifica en el Anexo 6.2, donde se define el uso de direcciones de

cuenta y las conexiones simultaneas de usuarios respectivas, contemplando etapas de conexión de uno hasta cien usuarios conectados; recuperando los valores de los grupos de usuarios conectados, el tiempo total de duración por grupo en milisegundos y él envió de métricas; como se muestra en la Figura 97.

FIGURA 97

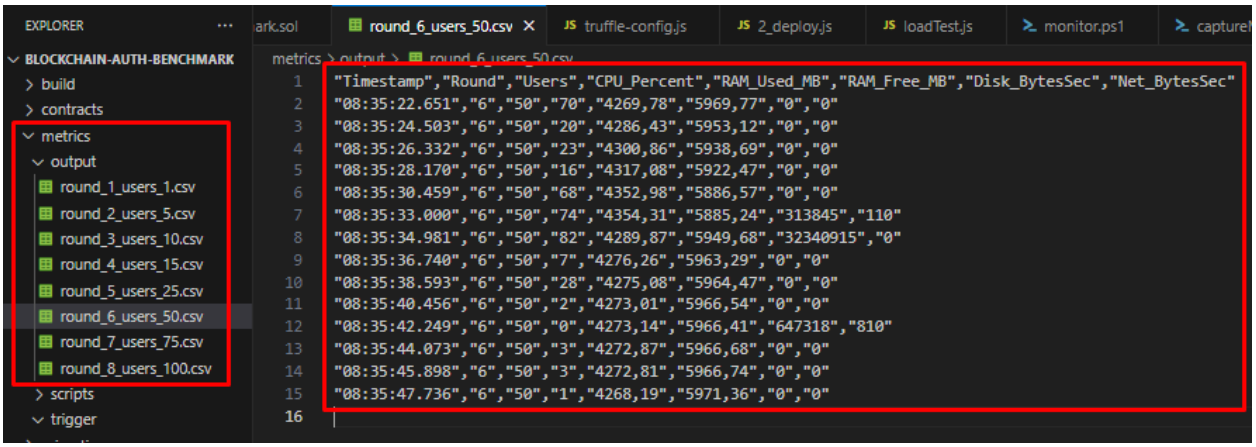
SCRIPT DE PRUEBA DE CONCEPTO

```
    const instance = await Access.deployed();
    const accounts = await web3.eth.getAccounts();

    const testBlocks = [1,5,10,15,25,50,75,100];

    let round = 0;

    for (let size of testBlocks) {

        round++;

        console.log(`\n============================`);
        console.log(`Grupo de: ${size} Usuario(s)`);
        console.log(`============================`);

        const users = accounts.slice(1, size+1);

        await instance.batchAllow(users);
        await instance.startRound(size);

        const start = Date.now();

        // Todos los usuarios ejecutan actividad real simultánea
        await Promise.all(
            users.map(u => simulateUser(instance, u))
        );

        const end = Date.now();
        const duration = end - start;

        console.log(`Duración total de grupo: ${duration} ms`);

        // Notificar al monitor CIM (Métricas)
        const payload = {
            round: round,
            users: size,
            durationMs: duration
        };

        fs.writeFileSync(triggerPath, JSON.stringify(payload));

        while (fs.existsSync(triggerPath)) {
            await new Promise(r => setTimeout(r,100));
        }
    }
```

Los datos e información correspondiente al consumo de recursos, durante el despliegue de la prueba de concepto, se recuperan y guardan en archivos "*.csv*" por cada grupo de usuarios; esto se configura mediante el script de monitoreo mostrado en el Anexo 6.3; mientras que en la Figura 98 se indican los archivos, que contienen los valores y porcentajes obtenidos, para el procesamiento, uso de RAM, uso de disco y tráfico de red.

FIGURA 98

MÉTRICAS DE PRUEBA DE CONCEPTO

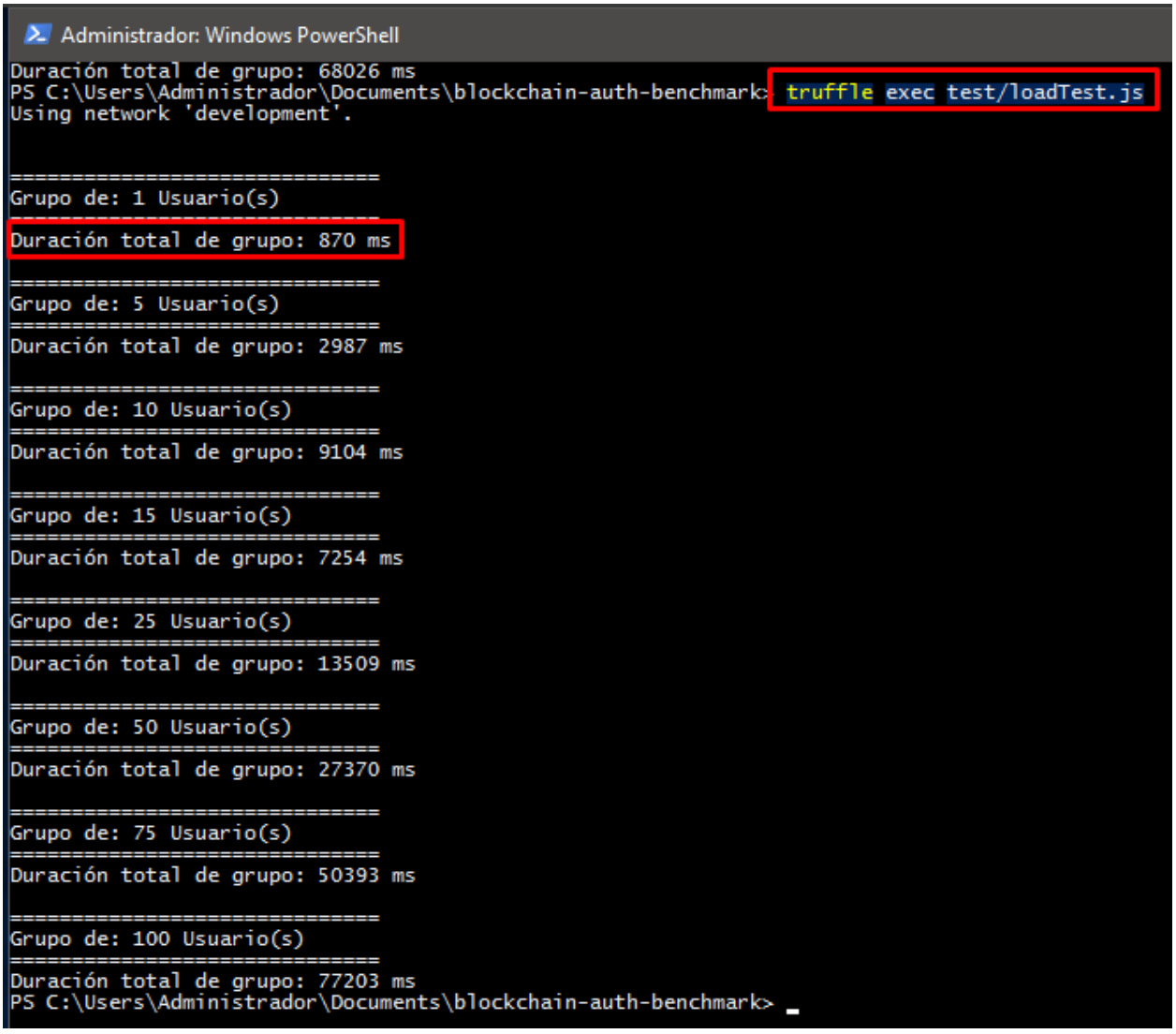


4.1.9.1. Resultado Test 9

El despliegue de la prueba de concepto se ejecuta en la consola del Framework Truffle, tomando en cuenta las configuraciones realizadas al contrato inteligente y al script de la prueba de concepto, para aplicarlas en el entorno Ganache mediante el comando "*truffle exec test/loadTest.js*" resaltado en la Figura 99; en la cual se aprecia el resultado de la prueba de concepto en la consola, mostrando los diferentes grupos o escenarios de conexión simultanea de usuarios concurrentes en el entorno de pruebas del mecanismo de autenticación propuesto; los datos obtenidos indican, el número de usuarios y el tiempo que tardo la conexión. Además, el script de monitoreo genera los archivos que contienen las métricas para cada grupo de usuarios.

FIGURA 99

RESULTADO PRUEBA DE CONCEPTO



Para tener una mejor interpretación de los resultados, se adapta un enfoque experimental cualitativo que evalúa el rendimiento del mecanismo de autenticación basado en cadena de bloques con las condiciones de la prueba de concepto, permitiendo el análisis del comportamiento del equipo en escenarios con grupos establecidos para; 1, 5, 10, 15, 25, 50, 75 y 100 usuarios, obteniendo diez muestras por cada grupo lo que permite contemplar la varianza temporal para la posterior ponderación y análisis estadístico (Hoefler & Belli, 2015).

Los resultados de las muestras para cada grupo de usuarios, generados tanto por la consola del framework Truffle como por el script de monitoreo en los archivos .csv, se agrupan y organizan en las tablas mostradas en el Anexo 6.4; el cálculo de la Media Muestral (H. Carrión, 2012) para cada grupo de usuario se realiza mediante la Ec. 1.

$$\bar{X} = \frac{1}{n}\sum_{i=1} x_i \qquad \text{(Ec. 1)}$$

Donde:

$x_i = valor\ variable\ de\ muestra$

$i = índice\ de\ variables\ de\ muestra$

$n = tamaño\ de\ la\ muestra$

Aplicando la media muestral a los valores obtenidos por cada grupo de usuarios se elabora la Tabla 19, que agrupa y detalla tiempos de cada muestra por para cada grupo de usuarios, obtenida al momento de aplicar y ejecutar la prueba de concepto.

TABLA 19

TIEMPOS DE GRUPO DE USUARIOS

Grupo Usuarios	Tiempo Grupo (ms)
1	723,60
5	3117,50
10	7089,10
15	7715,20
25	12973,50
50	29675,00
75	47364,20
100	66669,90

Con los datos de tiempo se realiza el cálculo para obtener los valores que tarda cada usuario en los diferentes escenarios, esto se realiza dividiendo el tiempo que tardo el grupo (Tiempo Grupo) para

el número de usuarios, donde se observa el valor máximo de tiempo es para 1 usuario con 723.60 ms y el valor mínimo de tiempo es para 15 usuarios con 514.35 ms, que indica una tendencia no lineal presente en arquitecturas dependientes de funciones de procesamiento (Hennessy & Patterson, 2019) ya que a partir de 50 usuarios se nota nuevamente un incremento; como se muestra en la Tabla 20.

TABLA 20

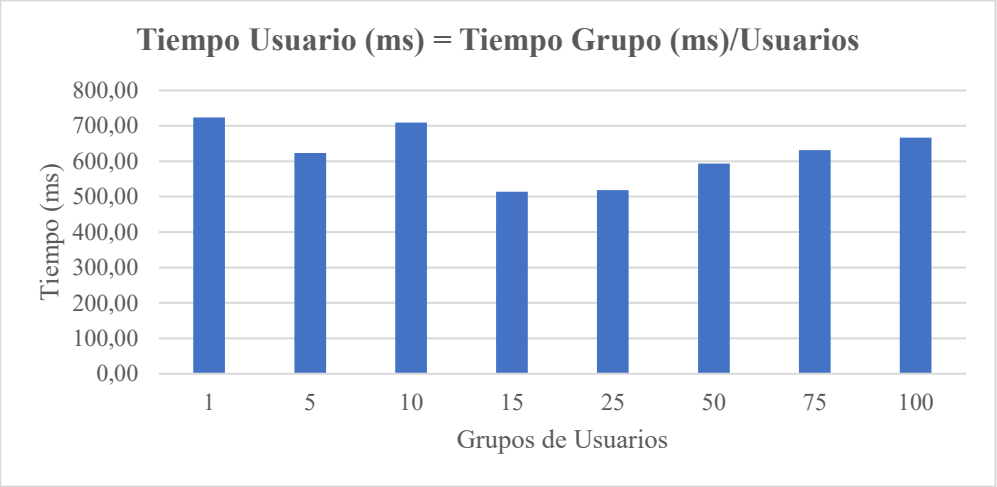
TIEMPOS DE USUARIO

# Usuarios	Tiempo Grupo (ms)	Tiempo Usuario (ms) = Tiempo Grupo (ms)/Usuarios
1	723,60	723,60
5	3117,50	623,50
10	7089,10	708,91
15	7715,20	514,35
25	12973,50	518,94
50	29675,00	593,50
75	47364,20	631,52
100	66669,90	666,70

La Figura 100 muestra la interpretación de los datos obtenidos respecto a la variación de respuesta de tiempo durante el proceso de autenticación, donde para 1 usuario se tiene un tiempo inicial alto; esto corresponde al efecto de calentamiento o *warm-up*, que se define como el periodo de tiempo que requiere un sistema para su ejecución estable, siendo un efecto en la medición de rendimiento y la simulación de sistemas (Hennessy & Patterson, 2019). Además, también se cuenta con el tiempo de ejecución o *runtime* de Node.js para estabilizar estructuras internas, optimizar el código y establecer conexiones (Singh, 2025), la inicialización del procedimiento de llamadas remoto (RPC) usando Ethers.js o Web3.js para la conexión puente con el nodo, la compilación JIT (*Just-In-Time*) junto a la asignación inicial de memoria (heap) que añade latencia (Bellsofaba, 2023; nodejs.org, 2024) y la apertura del canal HTTP con Ganache.

FIGURA 100

TIEMPOS DE CONEXIÓN DE USUARIOS



La Media muestral para los valores de la columna Tiempo de Usuario se obtiene usando la Ec 2; para este caso *n* representa el número de grupos de usuarios que son 8, para el calcular la media de tiempo de usuario.

$$\bar{X} = \frac{1}{n}\sum_{i=1} x_i \qquad \text{(Ec. 2)}$$

Donde para este caso:

$x_i = valor\ variable\ de\ tiempo$

$i = índice\ de\ variables\ de\ tiempo$

$n = número\ de\ grupos\ de\ usuarios = 8$

Dando como resultado:

$$Media\ de\ Tiempo\ de\ Usuario = \bar{\boldsymbol{X}} = \boldsymbol{622,63}\ (\boldsymbol{ms})$$

El cálculo de la Mediana se realiza con la Ec. 3, ya que el conjunto de datos es par se requiere ordenar los datos de Tiempo de Usuario para obtener los datos centrales como se resalta en la Tabla 21.

$$\tilde{X} = \frac{x_{(n/2)} + x_{(n/2+1)}}{2} \qquad \text{(Ec. 3)}$$

Donde para este caso:

$x = cada\ valor\ de\ menor\ a\ mayor$

$n = número\ total\ de\ datos$

$\frac{n}{2}\ y\ \frac{n}{2} + 1 = posiciones\ de\ los\ datos\ centrales$

Dando como resultado:

$$Mediana\ de\ Tiempo\ de\ Usuario = \widetilde{\boldsymbol{X}} = \mathbf{627,51}(\boldsymbol{ms})$$

TABLA 21

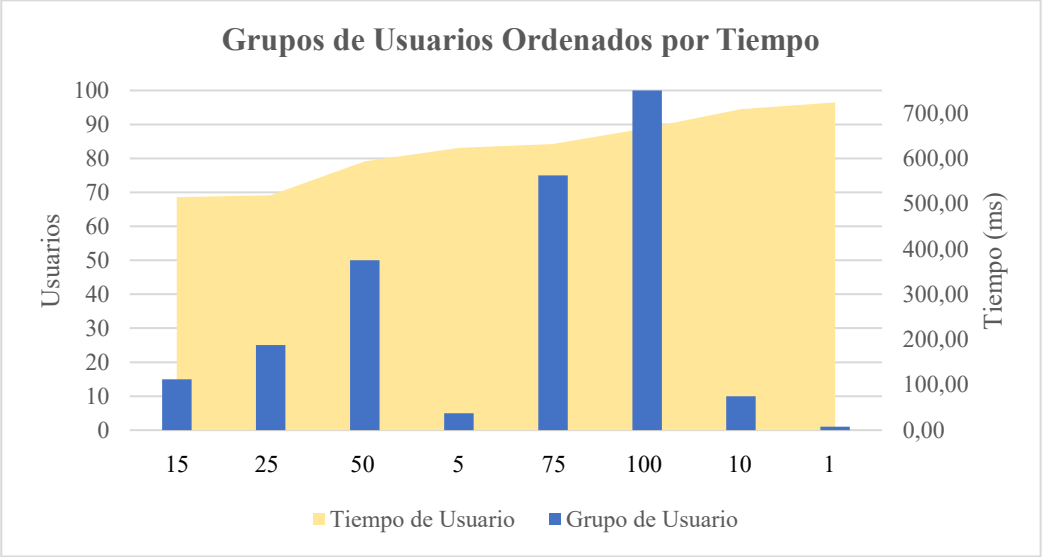
TIEMPOS DE USUARIO ORDENADOS

# Usuarios	Tiempo Grupo (ms)	Tiempo Usuario (ms)
15	7715,20	514,35
25	12973,50	518,94
50	29675,00	593,50
5	3117,50	623,50
75	47364,20	631,52
100	66669,90	666,70
10	7089,10	708,91
1	723,60	723,60

En la Figura 101 se demuestra mediante la gráfica de tiempos representada por el área amarilla, la cercanía de tiempos para los grupos de 5 y 75 usuarios representados en las barras azules, que son los valores considerados en el cálculo de la Mediana.

FIGURA 101

TIEMPOS DE CONEXIÓN DE GRUPO DE USUARIOS ORDENADOS



De la misma manera se calcula la Desviación Estándar usando la Ec. 4, este valor es necesario para el cálculo del Intervalo de Confianza.

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}} \qquad \text{(Ec. 4)}$$

Donde:

$x_i = valor\ variable\ de\ tiempo$

$n = número\ de\ grupos\ de\ usuarios = 8$

$(x_i - \bar{x})^2 = diferencia\ al\ cuadrado\ entre\ los\ valores\ y\ la\ media$

Dando como resultado:

$$Desviación\ Estándar = \boldsymbol{s = 78,36}$$

El Error Estándar de la media se calcula con la Ec 5:

$$EE = \frac{s}{\sqrt{n}} \qquad \text{(Ec. 5)}$$

Donde:

$s = Desviación\ estándar$

$n = número\ de\ grupos\ de\ usuarios = 8$

Dando como resultado:

$$\text{Error Estándar } = \boldsymbol{EE = 27,71}$$

Para el cálculo del Intervalo de Confianza expresado en la Ec 6, aplica un nivel de 95%, debido a que equilibra la precisión del intervalo, también se considera el riesgo de error igual a 0.05 (NIST, 2024); el valor crítico *t* cuando *n* < 30 usa la tabla t-Student (Anexo 6.5) y depende del grado de libertad igual a *n-1* (Berberyan et al., 2025).

$$IC = \bar{x} \pm t_{\alpha/2,n-1}\frac{s}{\sqrt{n}} \qquad \text{(Ec. 6)}$$

Donde:

$\bar{x} = Media\ muestral$ = 622.62 ms

$\alpha/2 = Riesgo\ de\ error = 0.05/2 \approx 0.025$

$n - 1 = Grados\ de\ libertad = 7$

$t_{\alpha/2,n-1} = Valor\ Crítico \approx 2.365$

$\frac{s}{\sqrt{n}} = Error\ estándar = 27.71$

$t_{\alpha/2,n-1}\frac{s}{\sqrt{n}} = Margen\ de\ error = \mathbf{65.53}$

Dando como resultado:

$$\text{Intervalo de Confianza al } 95\% \ = \{(622.62 - 65.53)\ ;\ (622.62 + 65.53)\}\ ms$$

$$\mathbf{IC}\ = \{\mathbf{557.10}\ ;\ \mathbf{688.15}\}\ \boldsymbol{ms}$$

Se calculan los valores para los niveles de confianza de 90% y 99%, con el número de grupos de grupos de usuarios (8), los grados de libertad (7) y la media muestral (622.62 ms). En la Tabla 22 se observa que; al aumentar el nivel de confianza, el valor critico aumenta y ensancha el margen de error; demostrando la relación inversamente proporcional de **confiabilidad** respecto a la **precisión** ayudando a determinar que el nivel de confiabilidad del 95% es equilibrado y los tiempos óptimos de conexión de usuario idealmente oscilan entre los valores del intervalo 557.10 ms – 688.15 ms.

TABLA 22

COMPARATIVA DE INTERVALOS DE CONFIANZA

Nivel	Valor crítico	Margen de Error	IC	
90%	1.895	52.502	570,13	675,13
95%	2.365	65.524	557,10	688,15
99%	3.499	96.942	525,69	719,57

De igual manera, aplicando el cálculo de la media muestral a los valores obtenidos del monitoreo de métricas, en la Tabla 23 se agrupa y detallan; tiempos, porcentajes de procesamiento, uso de RAM, disco y recursos de red requeridos por cada grupo de usuarios al momento de aplicar y ejecutar la prueba de concepto.

TABLA 23

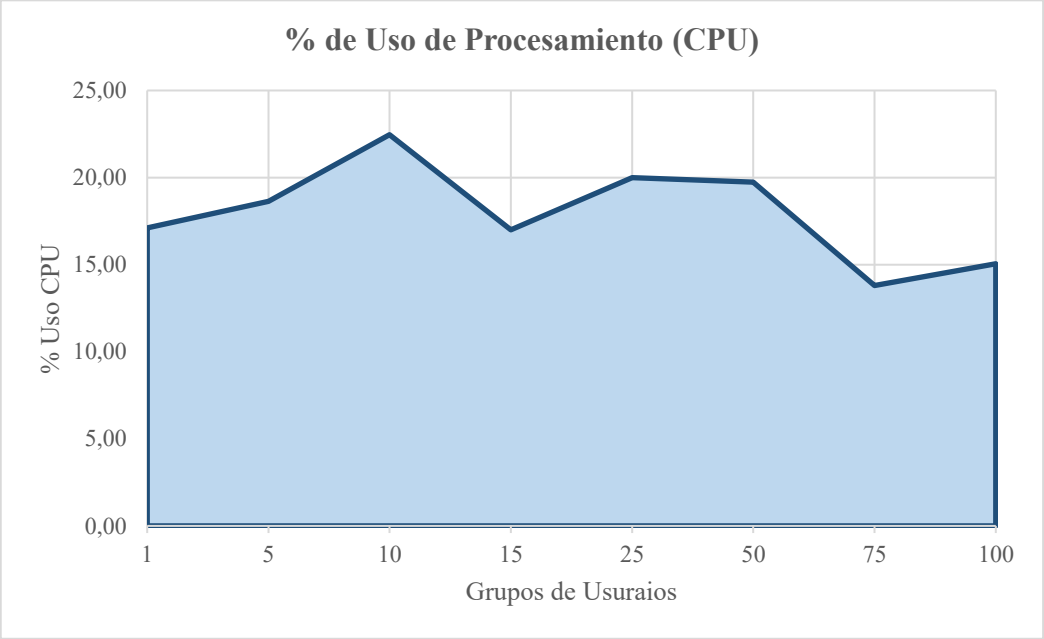
MÉTRICAS DE PRUEBA DE CONCEPTO

Grupo Usuarios	Tiempo Usuario (ms)	Uso CPU (%)	Uso RAM (MB)	Uso Disco (Bps)	Uso Red (Bps)
1	723,60	17,10	4131,12	88812,30	157,10
5	623,50	18,63	4193,94	10832,05	90,90
10	708,91	22,46	4268,15	51412,19	121,40
15	514,35	17,00	4379,59	245850,24	29,39
25	518,94	20,00	4315,44	59366,35	114,16
50	593,50	19,73	4303,42	599586,14	245,61
75	631,52	13,80	4237,09	98983,84	294,22
100	666,70	15,05	4228,07	536582,44	1074,22
Media	**622,63**	**17,97**	**4257,10**	**211428,19**	**265,88**

Con los datos de la tabla de métricas se elabora la Figura 102, donde se observa el comportamiento del porcentaje de uso del procesador, tomando en cuenta que nunca se acerca al 100% de uso y el equipo donde se encuentra el entorno de pruebas tiene un procesador Intel® Core™ i7-1080H CPU @ 2.20GHz; y de acuerdo con los escenarios de conexión establecidos se alcanzó el pico más alto para 10 usuarios con el 22.46%, su valor medio de 17.97% y su valor más bajo corresponde al grupo de 75 usuarios con un 13.80%; esta conducta se debe a que en la conexión con 10 usuarios se lleva a cabo la activación del colector de residuos generacional o *Garbage Collector*; la compactación de memoria (heap) y la optimización del JIT (Bellsofaba, 2023; nodejs.org, 2024), mientras que para las conexiones de 75 o 100 usuarios el porcentaje de uso de CPU se reduce debido a que para estos grupos de usuarios se prioriza el uso de disco y red, lo que ocasiona la reducción de procesos de CPU en espera de operaciones de lectura, escritura y conexión; reflejando la conducta de un sistema a *I/O bound* (limitado por entrada y salida), donde el uso del CPU se disminuye aunque se aumente la carga (Tanenbaum, 2015).

FIGURA 102

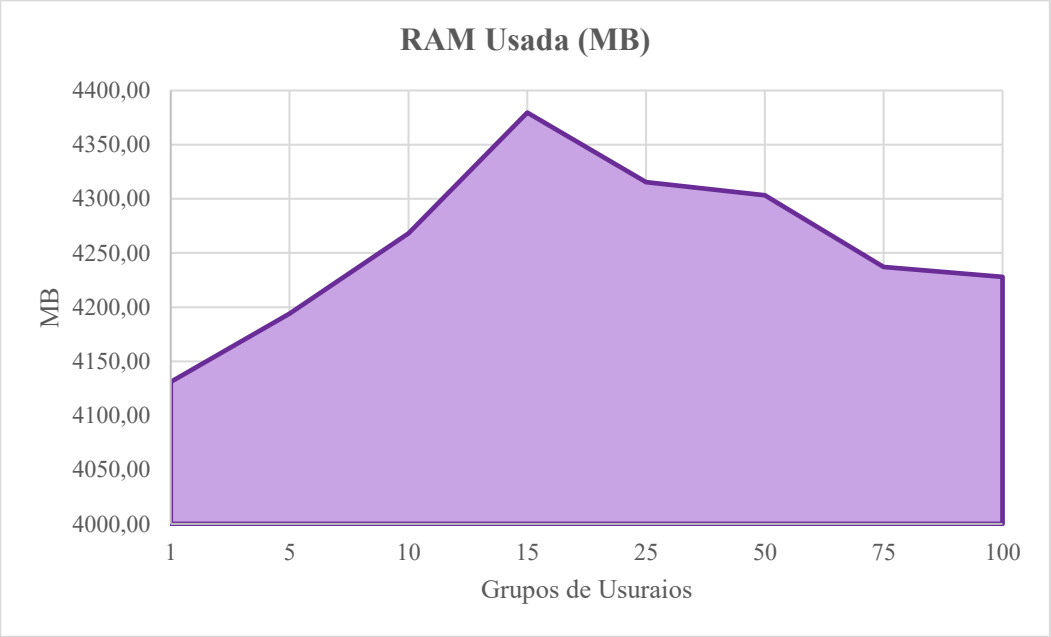
PORCENTAJE DE USO CPU



Un comportamiento similar se observa en la Figura 103, correspondiente al uso de RAM del equipo, mientras se realiza la prueba de concepto y aunque se haya asignado 10 GB (10000 MB) de RAM el consumo máximo llega a los 4379,59 MB con 15 usuarios, luego hay un descenso gradual hasta los 4228.07 con 100 usuarios; esta tendencia es típica en la expansión de memoria *heap* hasta un determinado umbral en un motor V8 y cuando se activa el *garbage collector* hay una liberación parcial de memoria (nodejs.org, 2024; Singh, 2025); esto se explica debido a que el *garbage collector* generacional produce picos temporales antes de estabilizar la memoria (Jones et al., 2023).

FIGURA 103

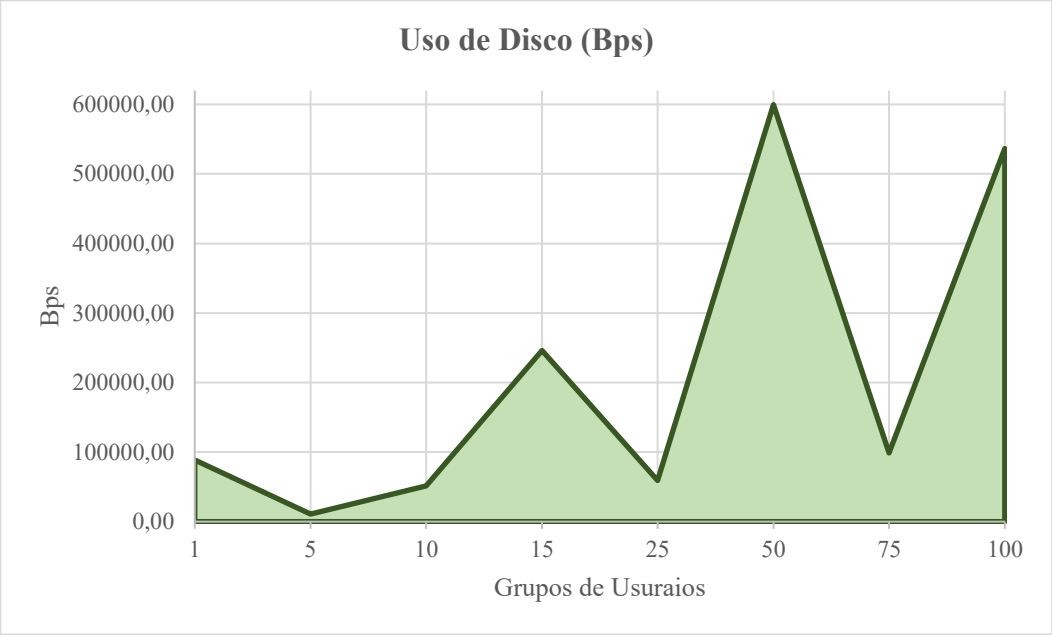
USO DE RAM



Respecto a la lectura y escritura de información, el comportamiento en el Disco de equipo es relativamente gradual a medida que aumenta el número de conexiones de usuarios; ya que el entorno Ganache debe consultar la información desplegada por el contrato inteligente, además de que se debe generar y almacenar los registros transaccionales para cada usuario que realiza la conexión. La Figura 104 permite visualizar que con el grupo de 50 usuarios se alcanza el valor máximo de 5999586.14 Bps, seguido del grupo de 10 usuarios con 536582.44 Bps y el del de 15 usuarios con 245850.24 Bps de escritura y acceso al disco; indicando que cuando el número de transacciones simultaneas aumenta se produce un *flush* (volcado) masivo de bloques, agrupando datos en memoria *cache*, hasta le ejecución de un *write-back* sincrónico; lo que básicamente es un procesamiento por lotes (batch persistence) y que coincide con los lineamientos de un modelo I/O buffering (Tanenbaum, 2015).

FIGURA 104

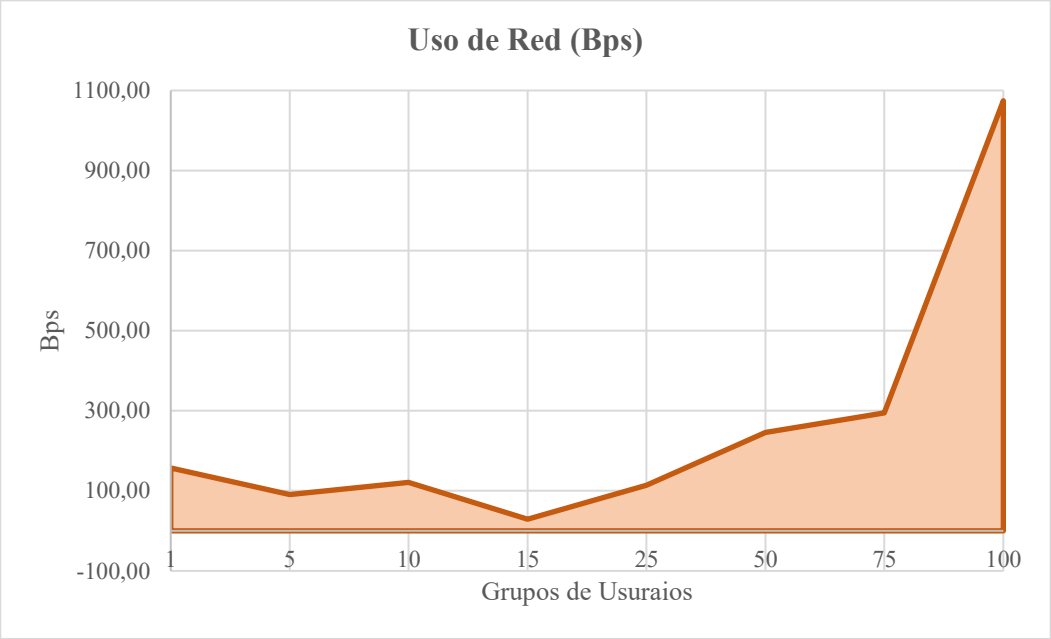
USO DE DISCO



Finalmente, la Figura 105 representa el comportamiento de la interfaz de red del equipo servidor, en la gráfica se muestra el intercambio de información que envía el entorno de cadena de bloques, que generalmente usa la interfaz de "*loopback*" identificada con la dirección IP 127.0.0.1 o identificada como "*localhost*" y además también se comunica mediante el puerto 8545 que usa Ganache, estas configuraciones se realizan al momento de desplegar el contrato inteligente y permiten que una vez asociado a la cadena de bloques se cumpla las condiciones establecidas; para este caso las de la prueba de concepto, es por esto que se encuentra en comunicación con el entorno de cadena de bloques y el entorno de pruebas de despliegue enviado y recibiendo datos, evidenciando que cuando se realizó la conexión del grupo de 100 usuarios se envió más información.

FIGURA 105

TRÁFICO DE RED



Conclusiones

Las conclusiones obtenidas durante el desarrollo del presente trabajo de titulación corresponden al análisis, en relación con el cumplimiento de los objetivos planteados; para lo cual se han llevado cabo diferentes actividades buscando la adaptabilidad de un mecanismo de autenticación usando la tecnología de cadena de bloques dentro de un entorno inalámbrico de pruebas en la facultad.

El análisis de métodos de autenticación tradicionales usados actualmente, que se establecen en mecanismos de autenticación basados que usan un servidor Radius/LDAP, un portal cautivo o una clave pre compartida para la validación y acceso a la red; sirvieron como puntos de referencia para orientar el desarrollo del mecanismo de autenticación y aprovechar las ventajas de la tecnología de cadena de bloques.

La revisión de diferentes trabajos de investigación, artículos científicos y documentos bibliográficos relacionada a proyectos que buscan adaptar y desarrollar aplicaciones para el uso de la tecnología descentralizada de cadena de bloques; aportaron académicamente en el planteamiento, solución y el desarrollo metodológico del mecanismo de autenticación propuesto; ya que se adoptó la metodología en Cascada que permite una fácil implementación y secuencia lógica; con el levantamiento de requerimientos en una estructura definida que establece fases y etapas para administrar el desarrollo y seguimiento del proyecto.

La plataforma Ganache Ethereum ofrece un entorno controlado de cadena de bloques que juntamente con el Framework Truffle permite realizar pruebas y experimentar su funcionamiento ya que es una de las herramienta que cuenta con una interfaz gráfica, donde se realizan las configuraciones de despliegue del entorno, el número de direcciones de cuenta, la configuración de interfaz de red y el puerto de conexión, además que permite enlazar contrato inteligente para

su despliegue, permitiendo la familiarización y comprensión de los procesos que se realizan y se generan al momento de interactuar como parte de una cadenas de bloques.

Dentro de la configuración y desarrollo de backend se contemplan componentes, elementos y paquetes de desarrollo como; NVM, Node.js, NPM, Web3.js y Solidity que son necesarios para el uso del entorno de cadena de bloques Ganache y permiten enlazar el entono de cadena de bloques, el contrato inteligente, el framework Truffle y la web de acceso y validación del mecanismo de autenticación propuesto.

La configuración del frontend bajo el componente Vite React con TypeScript permite crear los respectivos directorios, archivos y plantillas que contienen las rutas de visualización de los componentes de la interfaz web y de los de la cadena de bloques, útil para entornos en desarrollo ya que se pude editar y configurar los componentes mientras se ejecuta una vista previa, para observar su comportamiento y experimentar mientras se realizan configuraciones y cambios.

Respecto al levantamiento del entorno de pruebas para el mecanismo de autenticación propuesto, la arquitectura consta de un servidor Windows Server 2016 donde se configura el entorno de cadena de bloques y la interfaz web de acceso y validación, la parte de la infraestructura de red la proporciona el equipo Mikrotik AP RB951Ui 2HnD, que propaga el SSID para la red inalámbrica de pruebas, los usuarios se han definido como los dispositivos disponibles en el entorno de pruebas.

Finalmente se logró obtener los resultados esperados mediante el desarrollo del plan de pruebas de concepto planteado, aplicado al mecanismo de autenticación propuesto; comprobando funcionalidad, integración con el estándar IEEE 802.11, validación, autenticación y rendimiento; demostrando que un mecanismo de autenticación basado en cadena de bloques es teóricamente factible de acuerdo con la investigación realizada al desarrollar este proyecto de titulación.

En los resultados se evidencia un comportamiento no lineal, que es una característica de los sistemas concurrentes asincrónicos; mostrando una reducción en los tiempos de autenticación por usuario debido al efecto warm-up en el inicio del tiempo de ejecución y posteriormente en el rango de 15 a 25 usuarios concurrentes muestra una zona de operación optima respecto al tiempo de conexión inicial.

La ausencia de un aumento exponencial en el tiempo de respuesta indica que, el mecanismo teóricamente no se satura con 100 usuarios concurrentes, permitiendo plantear la escalabilidad del entorno de pruebas, respecto al hardware y software usado y desplegado para esta investigación.

El mecanismo muestra escalamiento no lineal; ya que los cuellos de botella o la congestión es dinámica, no dispara el uso de memoria y no hay una saturación crítica cambiando activamente entre un límite de usos de CPU a un límite de uso de entradas y salidas (I/O-bound), lo que se resume en un comportamiento estadísticamente estable.

Recomendaciones

Inicialmente, para comprender el funcionamiento de entorno de cadena de bloques, su interacción con el contrato inteligente y el framework de Truffle, se presentaron conflictos de compatibilidad de versiones para los componentes; NPM, Node.js, Web3.js, Ethers.js, Solidity, Ganache y Truffle, por lo que se aconseja visitar los sitios oficiales de los desarrolladores, para consultar que versiones son compatibles con cada uno de los elementos necesarios al desplegar un entorno de cadena de bloques y posteriormente para que funcionen correctamente cuando se implemente los componentes del mecanismo de autenticación propuesto.

Es necesario realizar pruebas iniciales de despliegue del contrato inteligente en la cadena de bloques de prueba, para verificar el cumplimento de las condiciones del contrato mediante la consola de Truffle, la cual permite testearlos y observar los resultados mediante mensajes de consola, además el uso de Ganache con interfaz gráfica ayuda a conocer cómo se despliega un contrato, como se genera los bloques, las direcciones de cuenta disponible y donde buscar e identificar las transacciones.

Considerando que el mecanismo de autenticación propuesto se desplego en un entorno controlado de pruebas, permitiendo la validación y acceso de determinados usuarios al recurso de red, mediante una conexión inalámbrica de acceso limitada, se recomienda el uso de este enfoque para escenarios educativos y/o experimentales, ya que la implementación en entornos de red reales o de producción puede presentar limitantes de desempeño y debido a las restricciones propias de Ganache al ser solo una plataforma de pruebas de cadena de bloques.

Tomando en cuenta las limitantes de despliegue del mecanismo de autenticación propuesto basado en cadena de bloques dentro de un entorno de pruebas, es necesario considerar su uso con cadenas de bloques reales, la cuales requieren uso de cripto-billeteras, direcciones de cuenta reales, manejo

de cripto activos, consumos de “gas” (rubro monetario que se paga por realizar una transacción) y la propiedad de inmutabilidad propia de estos entornos que cuenta con nodos de usuarios alrededor del mundo, lo que haría que el mecanismo de autenticación sea parte de un conglomerado descentralizado de autenticación y validación.

Considerando el comportamiento del mecanismo como un sistema no lineal concurrente asincrónico en la obtención e interpretación de datos de la prueba de concepto, algunas referencias bibliográficas sugieren descartar las valores de la muestras iniciales debido al efecto warm-up, definir grupos usuarios más cercanos entre sí o realizar el muestreo por grupos de usuario de manera aleatoria para verificar el comportamiento.

Por otra parte, explorando el desarrollo de un mecanismo de autenticación con cadena de bloques, se sugiere un escenario de investigación futura, donde ser parte de una cadena de bloques, permita conectarse al recurso de red generando micro transacciones, que se reflejen en una ganancia monetaria para la dirección de cuenta que ayude a negociar la autenticación.

Bibliografía

Acuña, H. (2017). Estudio sobre Bitcoin y Tecnología Blockchain. *Cuadernos Cef*, *1*(November), 1–44. http://www.ese.cl/wp-content/blogs.dir/1/files_mf/1510073019CUADERNOS_CEF_1_EstudiosobreBitcoinytecnologíaBlockchainv2003.pdf

Aguirre, J., & Aguirre, S. (2020). *Metodologías para el desarrollo de Proyectos*. https://repository.unicatolica.edu.co/handle/20.500.12237/2037

Andaluz, W. (2021). *AUTOMATIZACIÓN DE UN SISTEMA DE DESINFECCIÓN MEDIANTE TECNOLOGÍA INALÁMBRICA (WIFI) EN EL GOBIERNO AUTÓNOMO DESCENTRALIZADO PARROQUIAL RURAL DE PICAIHUA*. UNIVERSIDAD TÉCNICA DE AMBATO.

Arias, J. (2023, March 12). *¿Qué es JavaScript y cómo funciona?* Https://Es.Linkedin.Com/Pulse/Qu%C3%A9-Es-Javascript-y-C%C3%B3mo-Funciona-Jorge-Arias-Arg%C3%BCelles.

Baldeón, V., & Zambrano, J. (2018). *IMPLEMENTACIÒN DE UN PROTOTIPO DE UNA RED DESCENTRALIZADA BLOCKCHAIN PARA EL VOTO ELECTRÓNICO EN LA UNIVERSIDAD DE GUAYAQUIL* [Proyecto de Titulación]. Universidad de Guayaquil.

Balmaseda Aranda, F. J. (2018). Aseguramiento de Dispositivos IoT con Blockchain e Infraestructura de Clave Pública. In *Universidad Internacional de La Rioja* (Vol. 1).

BBVA. (2018). *Diccionario básico de "blockchain": diez términos que debes conocer*. Www.Bbva.Com. https://www.bbva.com/es/diccionario-basico-blockchain-diez-terminos-debes-conocer/

Beck, R., & Müller-bloch, C. (2017). Blockchain as Radical Innovation : A Framework for Engaging with Distributed Ledgers 2 . Literature Background : Blockchain as. *Proceedings of the 50th Hawaii International Conference on System Sciences*, 5390–5399.

Bellsofaba. (2023, August 19). *Understanding Remote Procedure Call (RPC) Nodes: Essential Web3 Components*. Https://Medium.Com/@bellsofaba/Understanding-Remote-Procedure-Call-Rpc-Nodes-Essential-Web3-Components-5a68a7c12865.

Berberyan, T., Nguyen, T., & Swan, A. (2025, August 20). *Confidence Interval for the Mean Using t-values*. Https://Stats.Libretexts.Org/Courses/Citrus_College/Statistics_C1000%3A_Introduction_to_Statistics/07%3A_Confidence_Interval_for_One_Sample/7.03%3A_Confidence_Interval_for_the_Mean_Using_t-Values.

Bergquist, J. (2017). *Blockchain Technology and Smart Contracts: Privacy-preserving Tools* (Number 17023).

Bitcobie. (2018). *Glosario Blockchain, términos y definiciones*. Www.Bitcobie.Com. https://www.bitcobie.com/glosario-blockchain/

Branislav, S. (2018). *Blockchain technologies adapted for data manipulation in IoT*.

Brincat, A. A., Lombardo, A., Morabito, G., & Quattropani, S. (2019). On the use of Blockchain technologies in WiFi networks. *Computer Networks*, *162*. https://doi.org/10.1016/j.comnet.2019.07.011

Cañar, J., & Jara, R. (2022). *Análisis y desarrollo de una aplicación de registro de permisos y ausentismos sobre una Blockchain mediante un smart contract desplegado en una tesnet de Ethereum* [UNIVERSIDAD POLITÉCNICA SALECIANA SEDE CUENCA]. https://dspace.ups.edu.ec/bitstream/123456789/22142/1/UPS-CT009634.pdf

Carrión, H. (2012). *Ingeniería de Tráfico de Telecomunicaciones*. Escuela Politécnica Nacional.

Carrión, K. (2018). *Análisis de la utilización de la tecnología Blockcahin para la gestión de la información en sisteas de alarmas residenciales.* Escuela Politécnica Nacional.

Casas, D. L., Alfonso, J., Torralbo, L., García, C., Casas, D. M., Rumayor, R., Manuel, J., & López, V. (2019). *Aproximación basada en Blockchain para crear un modelo de confianza en la enseñanza superior abierta y ubicua A trust model in open and ubiquitous higher education based on Blockchain technology*. *13*, 5–36.

CEDIA. (2022, May 28). *CEDIA*. Www.Cedia.Edu.Ec/.

CEI. (2024, October 23). *¿Qué es HTML y CSS?* Https://Cei.Es/Que-Es-Html-Css/.

CISCO. (2017, March 27). *Cisco Catalyst 4500 Series Switch Data Sheet*. https://www.cisco.com/c/en/us/products/collateral/switches/catalyst-4500-series-switches/product_data_sheet0900aecd801792b1.html

Cisco. (2020, April 19). *Cisco Catalyst 9300 Series Switches*. https://www.cisco.com/c/en/us/products/collateral/switches/catalyst-9000/nb-06-cat-9k-aag-cte-en.html

CISCO. (2021a, February 2). *Cisco 5520 Wireless Controller Data Sheet*. https://www.cisco.com/c/en/us/products/collateral/wireless/5520-wireless-controller/datasheet-c78-734257.html

CISCO. (2021b, November 22). *Cisco Catalyst 9800-L Wireless Controller Data Sheet*. Www.Cisco.Com. https://www.cisco.com/c/en/us/products/collateral/wireless/catalyst-9800-series-wireless-controllers/datasheet-c78-742434.html

CISCO. (2023, November 29). *Cisco Catalyst 9115 Series Wi-Fi 6 Access Points Data Sheet*. Https://Www.Cisco.Com/c/En/Us/Products/Collateral/Wireless/Catalyst-9100ax-Access-Points/Datasheet-C78-741988.Html.

Cloud Standards Customer Council. (2017). *Cloud Customer Architecture for Blockchain Executive Overview*. https://www.omg.org/cloud/deliverables/CSCC-Cloud-Customer-Architecture-for-Blockchain.pdf

Cloudflare. (2023, February 22). *La evolución del blockchain hacia Web3*. Https://Www.Cloudflare.Com/Es-Es/the-Net/How-Blockchain-Web3/.

Coleman, D. D., Westcott, D. A., & Harkins, B. E. (2016). *CWSP Certified Wireless Security Professional Study Guide: Exam CWSP-205* (J. Wiley & Sons, Ed.; 2nd ed.). https://books.google.com.ec/books?id=4K7LCgAAQBAJ&pg=PA52&dq=SSID&hl=es-419&sa=X&ved=0ahUKEwitk6Ooz6TnAhXS1VkKHUISDxMQ6AEIMjAB#v=onepage&q=SSID&f=false

Collaguazo, K. (2017). *PLAN DE MEJORA CONTINÚA BASADO EN EL ESTUDIO DE LA RED LOCAL INALÁMBRICA (WLAN) ACTUAL DE LA UNIVERSIDAD TÉCNICA DEL NORTE*. Universidad Técnica del Norte.

CRIPTONOTICIAS. (2019). *Glosario de Bitcoin y blockchains*. Www.Criptonoticias.Com. https://www.criptonoticias.com/criptopedia/glosario/#E

Cuzme, F. (2017). *SEGURIDAD EN REDES* (pp. 1–53).

DS3Comunications. (2022, May 25). *Switch Administrable capa L2 48 puertos 10/100, 02 puertos 10G fibra SFP LAN Base image Cisco Catalyst 2960S WS-C2960-48TC-L*. https://www.ds3comunicaciones.com/cisco/WS-C2960-48TC-L.html

Dwyer, K. (2024, February 8). *What is the Ethereum Virtual Machine (EVM): The Complete Guide*. Https://Www.Ankr.Com/Blog/What-Is-Evm-Ethereum-Virtual-Machine/.

Ethereum. (2024, June 21). *ETHEREUM VIRTUAL MACHINE (EVM)*. Https://Ethereum.Org/En/Developers/Docs/Evm/.

Flores, D. (2021). *Triángulo de Seguridad Informática: Qué es y sus objetivos*. 25 de Agosto. https://openwebinars.net/blog/triangulo-de-seguridad-informatica-que-es-y-sus-objetivos/#:~:text=El triángulo de la seguridad informática consta de%3A Confidencialidad%2C Integridad,los datos que se manejan.

FRUTOS, J. (2019). *Desarrollo de un servicio de seguimineto de mercancías basado en la Cadena de Bloques Ethereum*. 50.

Funcas. (2021). Llegan los tokens no fungibles (NFT). In *Notas Observatorio de la Digitañización Financiera*. https://edition.cnn.com/2021/03/23/t

Gamba, E., & Valencia, E. (2021). *Proyecto de investigación para el diseño e implementación de redes MESH como opción de conectividad a internet en entornos rurales*. 1–31.

GARRIDO, J. J. (2018). *TRANSICIÓN DE PROTOCOLO IPV4 A PROTOCOLO IPV6 PARA LA RED INALÁMBRICA EDUROAM DENTRO DE LA UNIVERSIDAD TÉCNICA DEL*

NORTE [UNIVERSIDAD TÉCNICA DEL NORTE]. http://repositorio.utn.edu.ec/handle/123456789/7870

Hanif, M., & Song, H. (2019). Blocks' Network: Redesign architecture based on blockchain technology. In *Proceedings - IEEE International Conference on Industrial Internet Cloud, ICII 2019*. https://doi.org/10.1109/ICII.2019.00017

Hennessy, J., & Patterson, D. (2019). *Computer Architecture A Quantitative Approach* (S. Merken, Ed.; Sexta). Katey Birtcher.

Hoefler, T., & Belli, R. (2015). Scientific benchmarking of parallel computing systems: twelve ways to tell the masses when reporting performance results. *Https://Dl.Acm.Org/Doi/10.1145/2807591.2807644*.

Hunt, P. (2013, June 5). *¿Por qué construimos React?* Https://Es.Legacy.Reactjs.Org/Blog/2013/06/05/Why-React.Html.

IETF. (2006). *RFC 4627*. The Application/Json Media Type for JavaScript Object Notation (JSON). https://www.ietf.org/rfc/rfc4627.txt

INCIBE. (2017). *Amenaza vs Vulnerabilidad, ¿sabes en qué se diferencian?* Www.Incibe.Es. https://www.incibe.es/protege-tu-empresa/blog/amenaza-vs-vulnerabilidad-sabes-se-diferencian

Instituto Nacional de Ciberseguridad. (2017). Glosario de Términos de Ciberseguridad. *Www.Incibe.Es*, 1–41. https://www.incibe.es/protege-tu-empresa/guias/glosario-terminos-ciberseguridad-guia-aproximacion-el-empresario

INTEL. (2021, August 20). *Diferentes protocolos de Wi-Fi y velocidades de datos*. Https://Www.Intel.La/Content/Www/Xl/Es/Support/Articles/000005725/Wireless/Legacy-Intel-Wireless-Products.Html.

Interdominio. (2020). *Glosario de términos*. Interdomino.Com. https://interdomino.com/glosario-de-terminos/

Jameson, C. (2023, August 18). *Python and Blockchain Basics (Ganache, Web3, Python Environment)*. Https://Medium.Com/@crjameson/Python-and-Blockchain-Basics-Ganache-Web3-Python-Environment-23b117441790.

Jamsrandorj, U. (2017). *Decentralized Access Control Using Blockchain* (Number August). University of Saskatchewan.

Jiang, X., Liu, M., Yang, C., Liu, Y., & Wang, R. (2019). A blockchain-based authentication protocol for WLAN mesh security access. *Computers, Materials and Continua*, *58*(1), 45–59. https://doi.org/10.32604/cmc.2019.03863

Jiménez, M. (2019). From the blockchain technology to the token economy | De la tecnología blockchain a la economía del token. *Derecho PUCP*, (83), 61–87. https://doi.org/http://dx.doi.org/10.18800/derechopucp.201902.003

Jones, Richard., Hosking, Antony., & Moss, Eliot. (2023). *The garbage collection handbook : the art of automatic memory management*. Chapman & Hall/CRC.

Kaku, E. (2017). *Using Blockchain To Support Provenance in the Internet of Things*.

Kikitamara, S. (2017). *Digital Identity Management on Blockchain for Open Model Energy System*. Radboun University.

Kingsley, A. (2023, January 25). *From Idea to Minimum Viable Dapp - How to use Ganache to enhance your auction dapp*. Https://Archive.Trufflesuite.Com/Blog/from-Idea-to-Minimum-Viable-Dapp-How-to-Use-Ganache-to-Enhance-Your-Auction-Dapp/.

Komal, K. (2019). *Blockchain based Peer-to-Peer Energy Trading using IoT devices*. Universidad de Oviedo.

Lederkremer, M. (2019). *Redes Informáticas* (RedUsers, Ed.). https://books.google.com.ec/books?id=7frADwAAQBAJ&pg=PA110&dq=Access+Point+AP&hl=es-419&sa=X&ved=0ahUKEwjpooy-1qTnAhVrzlkKHY6BCzwQ6AEIKzAA#v=onepage&q=Access Point AP&f=false

Lizarraga, B., Viñas, A., Molero, I., & Preukschat, A. (2018). *Glosario Blockchain - 80 palabras que necesitas conocer.* Blockchainespana.Com. https://blockchainespana.com/glosario/

Lobanov, L. (2023, July 9). *The Best VS Code Extensions for Blockchain Developers: Boosting Efficiency and Simplifying Development*. Https://Medium.Com/Coinmonks/the-Best-vs-Code-Extensions-for-Blockchain-Developers-Boosting-Efficiency-and-Simplifying-2f5ae6940afa.

Lopez, C. (2021). *¿Qué es el protocolo 802?1X / EAP y cómo funciona?* Es.Ccm.Net. https://es.ccm.net/contents/785-802-1x-eap

Lozada, O., & Yangali, J. (2022). Guía para la elaboración de la tesis. Enfoque cualitativo. *Universidad Norbert Wiener*. https://doi.org/10.37768/unw.vri.0005

ManzDev. (2023, May 19). *¿Qué es HTML?* Https://Lenguajehtml.Com/Html/Introduccion/Que-Es-Html/.

Martín, A. (2021a). *BLOCKCHAIN: aplicación en el Registro de la Propiedad e implicaciones en materia probatoria*.

Martín, A. (2021b). *BLOCKCHAIN: aplicación en el Registro de la Propiedad e implicaciones en materia probatoria* [Trabajo de Fin de Grado, Universidad de la Laguna]. https://riull.ull.es/xmlui/bitstream/handle/915/24220/Blockchain%20aplicacion%20en%20el%20Registro%20de%20la%20Propiedad%20e%20implicaciones%20en%20materia%20probatoria.%20.pdf?sequence=1

Martínez Bohórquez, L. E. (2017). *ALGORITMO PARA LA ENCRIPTACIÓN Y DESENCRIPTACIÓN ENTRE ARCHIVOS DIGITALES DE AUDIO E IMAGEN*. Universidad de San Buenaventura sede Bogotá.

Mela, J. L., & Cedeño, E. (2019). Tecnologías Blockchain y sus aplicaciones. *VIsión Antataura*, *3*. http://portal.amelica.org/ameli/jatsRepo/225/225971010/index.html

MetaMask Learn. (2024, March 15). *Comenzar con MetaMask*. Https://Support.Metamask.Io/Es/Start/Getting-Started-with-Metamask/.

Mikrotik. (2025, August 22). *RB951Ui-2HnD*. Https://Mikrotik.Com/Product/RB951Ui-2HnD.

Montoya, A. (2021). *Sistema de autenticación basado en blockchain para la gestión de billetes en un entorno de transporte inteligente* [Universidad de Alicante]. https://rua.ua.es/dspace/bitstream/10045/118148/1/Sistema_de_autenticacion_basado_en_blockchain_para_la_ges_Montoya_Ros_Adrian.pdf

Monzon, M., & Angulo, G. (2020). Guía metodológica para la implementación de parámetros de instalación o mantenimiento de redes WLAN. In *Manual de introducción Ionic* (Vol. 2, Number 26).

Munro, A. (2020). *De la A a la Z: El glosario fundamental de la criptomoneda*. Www.Finder.Com. https://www.finder.com/mx/glosario-de-criptomonedas

Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. 9. https://bitcoin.org/bitcoin.pdf

NIST. (2024). *Confidence Limits for the Mean*. Https://Www.Itl.Nist.Gov/Div898/Handbook/Eda/Section3/Eda352.Htm.

Niu, Y., Wei, L., Zhang, C., Liu, J., & Fang, Y. (2018). An anonymous and accountable authentication scheme for Wi-Fi hotspot access with the Bitcoin blockchain. *2017 IEEE/CIC International Conference on Communications in China, ICCC 2017*, *2018-Janua*(Iccc), 4–6. https://doi.org/10.1109/ICCChina.2017.8330337

Node.js. (2023, April 18). *Run JavaScript Everywhere*. Https://Nodejs.Org/En.

nodejs.org. (2024, May 20). *The V8 JavaScript Engine*. Https://Nodejs.Org/En/Learn/Getting-Started/the-v8-Javascript-Engine#the-v8-Javascript-Engine.

npmjs. (2023, September 7). *NPM*. Https://Www.Npmjs.Com/About.

OKX. (2023, April 27). *¿Qué es una dirección de blockchain?* Https://Www.Okx.Com/Es-La/Learn/What-Is-Blockchain-Address#:~:Text=Las%20direcciones%20blockchain%20son%20esenciales,Evitar%20el%20acceso%20no%20autorizado.

Pacheco Jiménez, M. N. (2019). De la tecnología blockchain a la economía del token. *Derecho PUCP*, (83), 61–87. https://doi.org/10.18800/derechopucp.201902.003

Panda Security. (n.d.). *Glosario de Panda Security Info*. Www.Pandasecurity.Com. Retrieved October 14, 2020, from https://www.pandasecurity.com/es/security-info/glossary/#LetraA

Pereira, M., Toscano, M., & Villar, P. (2019). *Plataformas blockchain y escenarios de uso*. Universidad de la Republica Uruguay.

Plaza, D. (2018). *Diseño y desarrollo de diplomas académicos digitales mediante la tecnología blockchain*. Pontificia Universidad Javierana.

Porxas, N., & Conejero, M. (2018a). Tecnología blockchain: Funcionamiento, aplicaciones y retos jurídicos relacionados. *Actualidad Jurídica Uría Menéndez*, *1*(13), 24–36.

Porxas, N., & Conejero, M. (2018b). Tecnología Blockchain: Funcionamiento, Aplicaciones y Retos Jurídicos Relacionados. *Actualidad Jurídica Uría Menéndez*, 24–36. https://blogs.imf.org/2018/03/13/addressing-the-dark-side-of-

Preukschat, A., Kuchkovsky, C., Gómez Lardies, G., Díez García, D., & Iñigo, M. (2017). *Blockchain: La Revolución Industrial de Internet* (Primera).

Qbit. (2024, June 7). *¿Qué son los protocolos de consenso?* Https://Qbitbs.Com/Que-Son-Protocolos-de-Consenso/.

Reyes, D. (2018). APLICACIÓN DE BLOCKCHAIN PARA LA SEGURIDAD DE LOS DATOS DEL INTERNET OF THINGS [UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA]. In *Cyber Resilience of Systems and Networks* (Number Noviembre 2018). https://repositorio.usm.cl/handle/11673/47827

Romero Solís, J. (2019). *APLICACIONES DE CONTRATOS INTELIGENTES EN ETHEREUM* [Universidad Carlos III de Madrid]. https://e-archivo.uc3m.es/bitstream/handle/10016/29653/TFG_Jose_Romero_Solis.pdf?sequence=1

Romo, N. (2022). *DISEÑO DE UN SISTEMA DE PUBLICIDAD MULTIMEDIA MANEJADO POR UNA APLICACIÓN ANDROID Y CONTROLADO EL ACCESO DE USUARIOS MEDIANTE UN PORTAL CAUTIVO PARA EL CENTRO COMERCIAL LAGUNA MALL*. UNIVERSIDAD TÉCNICA DEL NORTE.

Rowell, D. (2018, July 25). *WiFi Station Authentication and Association*. Https://Netbeez.Net/Blog/Station-Authentication-Association/.

Rupsha, B. (2017). *Using Blockchain Technology and Smart Contracts for Access Management in IoT devices*. UNIVERSITY OF HELSINKI.

Sáez, J. (2024, July 5). *Qué es Blockchain y cómo funciona la tecnología Blockchain*. Https://Www.Iebschool.Com/Blog/Blockchain-Cadena-Bloques-Revoluciona-Sector-Financiero-Finanzas/.

Salazar, J. (2016). Redes Inalámbricas. *Artículo*, *2*, 40.

Sánchez, J. E. (2021). *Seguridad actual en redes Wifi*. https://oa.upm.es/68021/1/TFG_JAVIER_ESTEBAN_SANCHEZ.pdf

Sanda, T., & Inaba, H. (2016). Proposal of new authentication method in Wi-Fi access using Bitcoin 2.0. *2016 IEEE 5th Global Conference on Consumer Electronics, GCCE 2016*, 0–4. https://doi.org/10.1109/GCCE.2016.7800479

Schuurmans, P. (2019). *Blockchain technology potential in the chemical industry: an exploratory research on the value of blockchain technology for supply chain management of organizations in the chemical industry* [Eindhoven University of Technology]. https://research.tue.nl/en/studentTheses/blockchain-technology-potential-in-the-chemical-industry

Serna, J. (2017). *Simulación de una moneda virtual con Blockchain* [Universidad de Alicante]. https://rua.ua.es/dspace/bitstream/10045/69467/1/Simulacion_de_una_moneda_virtual_con_Blockchain_SERNA_JAEN_JUAN.pdf

Singh, I. (2025, September 15). *Inside the V8 JavaScript Engine*. Https://Www.Thenodebook.Com/Node-Arch/v8-Engine-Intro#the-Config-Object-Disaster-We-Had.

Sobral, J. (2021). *Cómo integrar Blockchain en una arquitectura de software: resultados de una Revisión Multivocal de la Literatura* [Universidad ORT Uruguay]. https://dspace.ort.edu.uy/

Solano, E., & Porras, D. (2020). El modelo iterativo e incremental para el desarrollo de la aplicación de realidad aumentada Amón_RA. *Revista Tecnología En Marcha*. https://doi.org/10.18845/tm.v33i8.5518

Soliditylang. (2023, March). *Solidity*. Https://Soliditylang.Readthedocs.Io/Es/Latest/.

Tanenbaum, A. S. (2015). *Modern Operating Systems* (Cuarta Edición). Pearson.

Tapscott, D., & Tapscott, A. (2016). Blockchain revolution : how the technology behind bitcoin is changing money, business, and the world. In *Portafolio Penguin*. New York : Portfolio / Penguin, [2016].

Thakur, M. (2017). Authentication, Authorization and Accounting with Ethereum Blockchain (Master's Thesis) [University of Helsinki]. In *University of Helsinki*. https://helda.helsinki.fi/bitstream/handle/10138/228842/aaa-ethereum-blockchain.pdf?sequence=2

Tito, M. (2022). *EVALUACIÓN DEL IMPACTO DE LA INTERFERENCIA INTERPROTOCOLO ENTRE LAS TECNOLOGÍAS 802.11AH Y LORA EN UN ENTORNO DE SIMULACIÓN*. UNIVERSIDAD TÉCNICA DEL NORTE.

Torres Cardona, R. (2021). *CRIPTOGRAFÍA SIMÉTRICA Y ASIMÉTRICA Y SU APLICACIÓN EN MEDIOS DIGITALES COMO LAS IMÁGENES, VIDEO Y AUDIO*.

Tovar, A. (2018). *El glosario de las criptomonedas: Aquí «se habla Crypto»*. Www.Cambio16.Com. https://www.cambio16.com/glosario-de-las-criptomonedas/

Tunala, M., & Moncayo, R. (2018). *Sistema para mejorar la seguridad de la información en identidades digitales aplicando tecnología Blockchain*. Universidad de las Fuerzas Armadas.

UTN. (2022, August 20). *EDUROAM-UTN Red Inalámbrica (WiFi-UTN) Movilidad Académica*. Https://Eduroam.Utn.Edu.Ec/Index.Php/Como-Conectarse/. https://eduroam.utn.edu.ec/index.php/como-conectarse/

Villanueva, J. (2020). *Seguridad en Redes Inalámbricas 802.1x*. JaCkSecurity.com

Vite.dev. (2023, March 15). *About VIte*. Https://Es.Vite.Dev/Guide/.

Wei, Q., Li, S., Li, W., Li, H., & Wang, M. (2019). *Trustroam: A Novel Blockchain-Based Cross-Domain Authentication Scheme for Wi-Fi Access*. *2*(March 2019), 358–369. https://doi.org/10.1007/978-3-030-23597-0

Yépez Lapo, J. A. (2021). *Diseño de una red inalámbrica (Wi-Fi) para servicio de internet público en el barrio Las Gaviotas ubicado en el recinto Matilde Esther, del Cantón Bucay de la provincia del Guayas*. UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL.

Anexos

Anexo 1: Manual de Eduroam

1 Ingresar a http://cat.eduroam.org, seleccionar la universidad y descargar el perfil.

A

B

C

2 INSTALAR PERFIL

A

B

C

3 Selecciona el wi-fi con el nombre de "eudoram"

INSTALACIÓN

MacOs

1 Ingresar a sitio web
http://cat.eduroam.org

A

B

2 siga los pasos que indicados por el instalador

A

B

3 El instalador solicitará que ingrese los datos de usuario y contraseña.
Usuario: (nuestra dirección de e-mail institucional)

NOTA

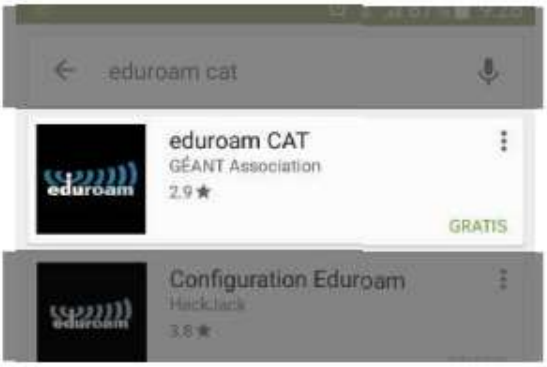
Es posible que el sistema operativo le pida que introduzca la clave de su usuario de sistema para completar el proceso de instalación.

INSTALACIÓN

Android

1 Descargar la aplicación eduroamCAT dentro de Google play

A



B

2 Ingresar a http://cat.eduroam.org, seleccionar la universidad y descargar el perfil.

A

B

C

3 Abrir el Perfil

4 Colocar Usuario y Contraseña.

INSTALACIÓN

iPhone

1 Ingresar a sitio web http://cat.eduroam.org
seleccion la universidad y descarga el perfil

A

B

2 Abrir el Perfil

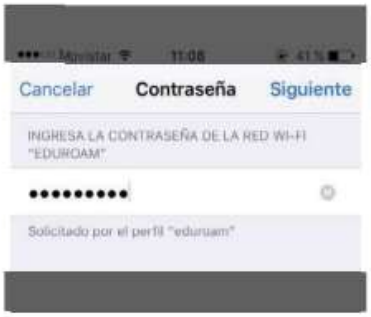
A

B

3 Colocar Usuario y Contraseña.

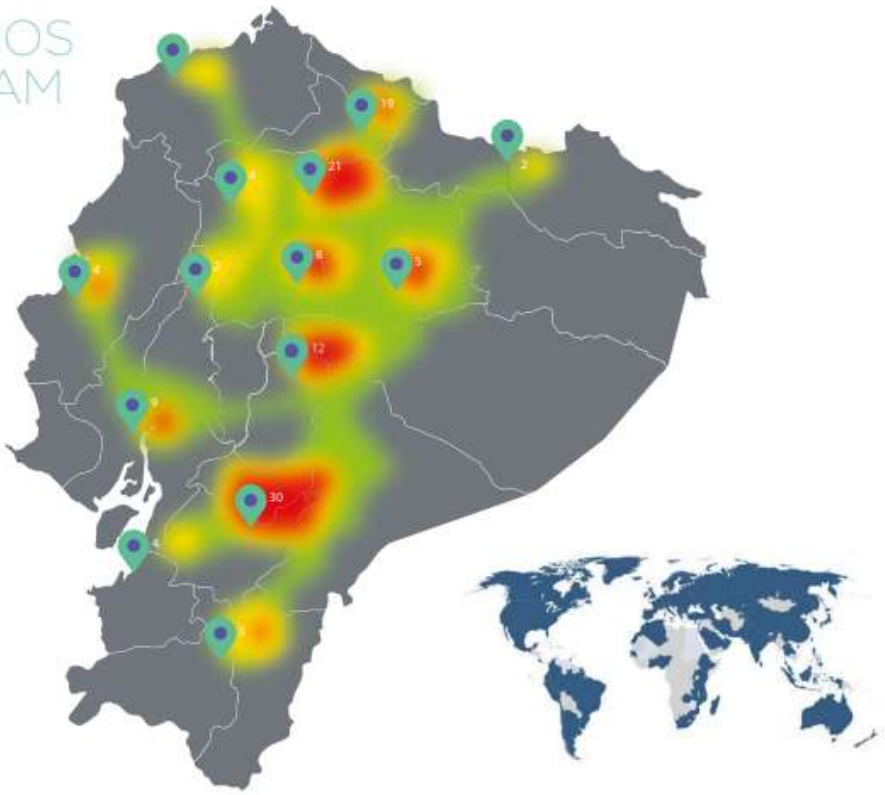
A

B



C

MIEMBROS EDUROAM



UNIVERSIDAD DE CUENCA
CEDIA
PUCE
UNL
ESPOL
UTPL
UTN
UDA
EUA
ESPOCH
EPN
UCE
ESPOL
INOCAR
UTA
UNAE
ITCA
UPS
UTMACH
USGP
UIDE
UCSG
UTE
UDLA
UNACH
ULEAM
ITSCO
ITSVN
USFQ
UPEC
YT
TECSU
IKIAM
UNEMI
UCACUE
UTI
FCAQ
ITSHCPP
UEPB
UARTES
UISRAEL
UP
PARQUES PÚBLICOS CUE.

RECUERDA

Eduroam es el servicio de movilidad segura desarrollado para la comunidad académica y de investigación. Abre tu portátil y estás conectado.

EDUROAM (EDUcation-ROAMing) permite la conectividad a Internet y Red Avanzada dentro de su propio campus y cuando visita a otras instituciones participantes a nivel nacional y alrededor del mundo.

Prerequisitos:

- La primera vez que va a configurar eduroam es necesario ya tenga activada una conexión a Internet para poder descargar los perfiles.
- Tener una cuenta institucional activa.
- Tener una clave válida para la cuenta institucional.

El nombre de la red WIFI en todo el planeta es eduroam, una vez conectado exitosamente en cualquier punto eduroam, automáticamente se reconectará apenas exista una antena con el servicio en cualquier parte del mundo, sin necesidad de pasos adicionales.

Información adicional, http://eduroam.ec.

ww.cedia.edu.ec
info@cedia.org.ec
(+593) 7 407 9300
CEDIAec

CUE
Oficinas _
Gonzalo Cordero 2-122
y J. Fajardo Esq.
Planta de producción _
Miguel Moreno y Av. 10
de Agosto.

UIO
Ladrón de Guevara
E11-253. EPN,
Casa Patrimonial.

Por un **Ecuador** que **Investiga** e **Innova** con niveles de clase mundial, conectando a los mejores.

Anexo 2: Situación Actual

Anexo 2.1: Solicitud de Información DDTI

Fecha:	Ibarra, 12 de mayo del 2022
Dirigido a:	MSc Daniel Jaramillo – COORDINADOR CITEL-CIERCOM
Estudiante:	Beltrán Manosalvas Galo Mauricio (gmbeltran@utn.edu.ec)
Facultad:	FICA
Carrera:	Ingeniería en Telecomunicaciones
Asunto:	Solicito muy comedidamente se me ayude, por su intermedio con la realización del trámite respectivo de solicitud de información a la Dirección de Desarrollo Tecnológico e Informático de la Universidad; con el fin de obtener los detalles de la situación actual referente a la infraestructura de red inalámbrica del Campus Universitario y específicamente de la Faculta de Ingeniería en Ciencias Aplicadas (topología, equipos, mecanismos de acceso, protocolos y redes inalámbricas propagadas); información que se usará para desarrollar y elaborar la documentación respectiva y los antecedentes del Proyecto de Titulación "Mecanismo de autenticación utilizando Cadena de Bloques en la red inalámbrica de la Facultad de Ingeniería en Ciencias Aplicadas"

Atentamente,

Mauricio Beltrán Manosalvas
ESTUDIANTE CITEL-CIERCOM

MSc. Fabián Cuzme Rodríguez
DIRECTOR PROYECTO DE TITULACIÓN

12 MAY/2022
15:26

UNIVERSIDAD TÉCNICA DEL NORTE

UNIVERSIDAD ACREDITADA RESOLUCIÓN 002-CONEA-2010-129-DC
Resolución N° 001-073-CEAACES-2013-13

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

RPC-SO-31-No 573-2016

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN
(No vigente, habilitado para registro de títulos)

Memorando UTN-FICA-CITEL-011

PARA: Juan Carlos García, Ingeniero, **Director Departamento Desarrollo Tecnológico e Informático**

DE: Daniel Jaramillo V., **Magíster**, **Coordinador CITEL**

ASUNTO: AUTORIZAR INGRESO A ESTUDIANTE PARA DESARROLLO DE TRABAJO DE TITULACIÓN

FECHA: 13 de mayo del 2022

Con la finalidad de atender el pedido realizado por el estudiante y el magíster Fabián Cuzme, autor y Director de tesis respectivamente, referente al pedido de que se proporcione información al señor Mauricio Beltrán Manosalvas, estudiante de la carrera de Ingeniería en Electrónica y Redes de Comunicación, respecto a la infraestructura de red inalámbrica del campus universitario, específicamente de la Facultad de Ciencias Aplicadas, la misma que le permitirá desarrollar y elaborar los documentos y antecedentes del proyecto de titulación: MECANISMO DE AUTENTICACIÓN UTILIZANDO CADENA DE BLOQUES EN LA RED INALÁMBRICA DE LA FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS.

Esta actividad estará bajo la supervisión del magíster Fabián Cuzme, director del trabajo de titulación; mucho agradeceré notificar la aceptación o no de este pedido, para de igual manera, informar a los interesados.

Por la gentil atención, le agradezco.

Atentamente,

CIENCIA Y TÉCNICA AL SERVICIO DEL PUEBLO

Daniel Jaramillo V. Mgs.
COORDINADOR CITEL

Anexo: solicitud
Silvia

Anexo 2.2: Formato de Entrevista Revisada y Aprobada

UNIVERSIDAD TÉCNICA DEL NORTE
Facultad de Ingeniería en Ciencias Aplicadas
Carrera de Ingeniería en Electrónica y Redes de Comunicación

ENTREVISTA DE PROYECTO DE TESIS	
TEMA:	MECANISMO DE AUTENTICACIÓN UTILIZANDO CADENA DE BLOQUES EN LA RED INALÁMBRICA DE LA FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS.
OBJETIVO:	Esta encuesta tiene la finalidad de conocer la información correspondiente a la situación actual y de infraestructura inalámbrica de red en la Facultad de Ingeniería en Ciencias Aplicadas.

ELABORADO POR:	**REVISADO POR:**
Galo Mauricio Beltrán Manosalvas TESISTA Fecha: 21 de noviembre del 2022	Ing. Fabián Geovanny Cuzme Rodríguez, MSc. DIRECTOR Fecha: 02 de diciembre del 2022

Datos del Entrevistado	
Nombre y Apellido:	**Firma:**
Cargo que desempeña:	
Fecha Entrevista:	

CUESTIONARIO:

1. ¿Qué tipo de red inalámbrica respecto al área de cobertura está disponible para el acceso y conexión de los estudiantes de la Facultad de Ingeniería en Ciencias Aplicadas?

   ☐ Red inalámbrica de área personal (WPAN)

   ☐ Red inalámbrica de área local (WLAN)

   ☐ Red inalámbrica de área metropolitana (WMAN)

   ☐ Red inalámbrica de área ampliada (WWAN)

2. ¿Qué tipo de red inalámbrica respecto a la arquitectura está disponible para el acceso y conexión de los estudiantes de la Facultad de Ingeniería en Ciencias Aplicadas?

   ☐ Red sin infraestructura o Ad Hoc (IBSS)

   ☐ Red con infraestructura (BSS/ESS)

3. ¿Qué tecnología de red inalámbrica se usa para el acceso y conexión de dispositivos de los estudiantes de la Facultad de Ingeniería en Ciencias Aplicadas?

   ☐ IEEE 802.11

   ☐ HIPERLAN

4. ¿Qué versiones de estándar IEEE 802.11 está disponible para el acceso y conexión de los estudiantes de la Facultad de Ingeniería en Ciencias Aplicadas?

   ☐ IEEE 802.11b

   ☐ IEEE 802.11n

   ☐ IEEE 802.11ac

   ☐ IEEE 802.11ax

5. ¿Qué tipo de seguridad inalámbrica se usa para el acceso y conexión inalámbrica en la Facultad de Ingeniería en Ciencias Aplicadas?

   ☐ WEP

   ☐ WAP

   ☐ Filtrado de MAC

   ☐ Tecnología IEEE 802.1x

6. ¿Cómo se autoriza el acceso y conexión de los estudiantes de la Facultad de Ingeniería en Ciencias Aplicadas a la red inalámbrica?

   ☐ Clave de conexión

   ☐ Portal Cautivo

   ☐ Registro de dirección MAC de dispositivos

   ☐ Servidor AAA

7. ¿A que identificador de red inalámbrica (SSID) los estudiantes de la Facultad de Ingeniería en Ciencias Aplicadas tienen permitido conectarse?

   ☐ WUTNDocentes

   ☐ eduroam

   ☐ BIBLIOTECA-UTN

   ☐ HEMEROTECA-UTN

8. ¿Existe un método alternativo para la conexión inalámbrica en la Facultad de Ingeniería en Ciencias Aplicadas o el campus universitario, al usado actualmente?

   ☐ Si

   ☐ No

9. ¿Considera necesario el desarrollo de un método alternativo para la conexión inalámbrica en la Facultad de Ingeniería en Ciencias Aplicadas?

   ☐ Si

   ☐ No

10. ¿Cuenta la facultad con recursos de hardware y software para levantar o implementar una solución de autenticación basada en la tecnología de cadena de bloques?

☐ Si

☐ No

11. ¿Conoce usted si los equipos de infraestructura inalámbrica (APs, switchs, servidor RADIUS/LDAP) tienen la capacidad de uso, convergencia y configuración con la tecnología de cadena de bloques?

☐ Si

☐ No

GRACIAS POR SU TIEMPO

Anexo 2.3: Entrevista Realizada

UNIVERSIDAD TÉCNICA DEL NORTE
Facultad de Ingeniería en Ciencias Aplicadas
Carrera de Ingeniería en Electrónica y Redes de Comunicación

ENTREVISTA DE PROYECTO DE TESIS	
TEMA:	MECANISMO DE AUTENTICACIÓN UTILIZANDO CADENA DE BLOQUES EN LA RED INALÁMBRICA DE LA FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS.
OBJETIVO:	Esta encuesta tiene la finalidad de conocer la información correspondiente a la situación actual y de infraestructura inalámbrica de red en la Facultad de Ingeniería en Ciencias Aplicadas.

Datos del Entrevistado	
Nombre y Apellido: Vinicio Guerra Morales	**Firma:**
Cargo que desempeña: Analista de Redes	
Fecha Entrevista: 2022/12/02	

CUESTIONARIO:

1. ¿Qué tipo de red inalámbrica respecto al área de cobertura está disponible para el acceso y conexión de los estudiantes de la Facultad de Ingeniería en Ciencias Aplicadas?

   ☐ Red inalámbrica de área personal (WPAN)

   ☒ Red inalámbrica de área local (WLAN)

   ☐ Red inalámbrica de área metropolitana (WMAN)

   ☐ Red inalámbrica de área ampliada (WWAN)

2. ¿Qué tipo de red inalámbrica respecto a la arquitectura está disponible para el acceso y conexión de los estudiantes de la Facultad de Ingeniería en Ciencias Aplicadas?

   ☐ Red sin infraestructura o Ad Hoc (IBSS)

☒ Red con infraestructura (BSS/ESS)

3. ¿Qué tecnología de red inalámbrica se usa para el acceso y conexión de dispositivos de los estudiantes de la Facultad de Ingeniería en Ciencias Aplicadas?

   ☒ IEEE 802.11

   ☐ HIPERLAN

4. ¿Qué versiones de estándar IEEE 802.11 está disponible para el acceso y conexión de los estudiantes de la Facultad de Ingeniería en Ciencias Aplicadas?

   ☒ IEEE 802.11b

   ☒ IEEE 802.11n

   ☒ IEEE 802.11ac

   ☒ IEEE 802.11ax

5. ¿Qué tipo de seguridad inalámbrica se usa para el acceso y conexión inalámbrica en la Facultad de Ingeniería en Ciencias Aplicadas?

   ☐ WEP

   ☐ WAP

   ☐ Filtrado de MAC

   ☒ Tecnología IEEE 802.1x

6. ¿Cómo se autoriza el acceso y conexión de los estudiantes de la Facultad de Ingeniería en Ciencias Aplicadas a la red inalámbrica?

   ☐ Clave de conexión

☐ Portal Cautivo

☐ Registro de dirección MAC de dispositivos

☒ Servidor AAA

7. ¿A que identificador de red inalámbrica (SSID) los estudiantes de la Facultad de Ingeniería en Ciencias Aplicadas tienen permitido conectarse?

☐ WUTNDocentes

☒ eduroam

☐ BIBLIOTECA-UTN

☐ HEMEROTECA-UTN

8. ¿Existe un método alternativo para la conexión inalámbrica en la Facultad de Ingeniería en Ciencias Aplicadas o el campus universitario, al usado actualmente?

☐ Si

☒ No

9. ¿Considera necesario el desarrollo de un método alternativo para la conexión inalámbrica en la Facultad de Ingeniería en Ciencias Aplicadas?

☒ Si

☐ No

10. ¿Cuenta la facultad con recursos de hardware y software para levantar o implementar una solución de autenticación basada en la tecnología de cadena de bloques?

☒ Si

☐ No

11. ¿Conoce usted si los equipos de infraestructura inalámbrica (APs, switchs, servidor RADIUS/LDAP) tienen la capacidad de uso, convergencia y configuración con la tecnología de cadena de bloques?

☐ Si

☒ No

GRACIAS POR SU TIEMPO

Anexo 3: Ficha de Requerimientos

UNIVERSIDAD TÉCNICA DEL NORTE

Facultad de Ingeniería en Ciencias Aplicadas

Carrera de Ingeniería en Electrónica y Redes de Comunicación

FICHA DE REQUERIMIENTOS	
TEMA:	MECANISMO DE AUTENTICACIÓN UTILIZANDO CADENA DE BLOQUES EN LA RED INALÁMBRICA DE LA FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS.
OBJETIVO:	Determinar correctamente los requerimientos para el desarrollo y cumplimiento de los objetivos planteados en el presente trabajo de titulación.

ELABORADO POR:	**REVISADO POR:**
Galo Mauricio Beltrán Manosalvas TESISTA Fecha: 27 de febrero del 2023	Ing. Fabián Geovanny Cuzme Rodríguez, MSc. DIRECTOR Fecha: 03 de marzo del 2023

RECURSOS BIBLIOGRÁFICOS
Cañar, J. Jara, R. (2022). Análisis y desarrollo de una aplicación de registro de permisos y ausentismos sobre una Blockchain mediante un smart contract desplegado en una tesnet de Ethereum. Universidad Politécnica Salesiana Sede Cuenca.
Carrión, K. (2018). Análisis de la utilización de la tecnología Blockchain para la gestión de la información en sistemas de alarmas residenciales. Escuela Politécnica Nacional.
Baldeón, V., & Zambrano, J. (2018). Implementación de un prototipo de una red descentralizada Blockchain para el voto electrónico en la Universidad de Guayaquil. Universidad de Guayaquil.
Bajaña, D. (2021). Modelo de Seguridad de Información basado en la tecnología Blockchain para los procesos del servicio de rentas internas del Ecuador. Universidad Politécnica Salesiana Sede Guayaquil.

Lita, E. Salazar, M. (2021). Propuesta de un modelo para mitigar noticias falsas en una red social basada en tecnología Blockchain. Universidad Central del Ecuador.
Balmaseda Aranda, F. J. (2018). Aseguramiento de Dispositivos IoT con Blockchain e Infraestructura de Clave Pública. Universidad Internacional de La Rioja (Vol. 1).
Cardozo Gladys, Perdomo Pablo. (2020). Comparación de plataformas para smart contracts basadas en blockchain. Universidad de la República de Uruguay.
Montoya, A. (2021). Sistema de autenticación basado en blockchain para la gestión de billetes en un entorno de transporte inteligente [Universidad de Alicante]. https://rua.ua.es/dspace/bitstream/10045/118148/1/Sistema_de_autenticacion_basado_en_blockchain_para_la_ges_Montoya_Ros_Adrian.pdf
Thakur, M. (2017). Authentication, Authorization and Accounting with Ethereum Blockchain (Master's Thesis) [University of Helsinki]. In University of Helsinki. https://helda.helsinki.fi/bitstream/handle/10138/228842/aaa-ethereum-blockchain.pdf?sequence=2
Kikitamara, S. (2017). Digital Identity Management on Blockchain for Open Model Energy System. Radboun University.
Komal, K. (2019). Blockchain based Peer-to-Peer Energy Trading using IoT devices. Universidad de Oviedo.
Rupsha, B. (2017). Using Blockchain Technology and Smart Contracts for Access Management in IoT devices. UNIVERSITY OF HELSINKI.
Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System.
Niu, Y., Wei, L., Zhang, C., Liu, J., & Fang, Y. (2018). An anonymous and accountable authentication scheme for Wi-Fi hotspot access with the Bitcoin blockchain. 2017 IEEE/CIC International Conference on Communications in China, ICCC 2017, 2018-Janua (Iccc), 4–6. https://doi.org/10.1109/ICCChina.2017.8330337
Sanda, T., & Inaba, H. (2016). Proposal of new authentication method in Wi-Fi access using Bitcoin 2.0. 2016 IEEE 5th Global Conference on Consumer Electronics, GCCE 2016, 0–4. https://doi.org/10.1109/GCCE.2016.7800479
Jiang, X., Liu, M., Yang, C., Liu, Y., & Wang, R. (2019). A blockchain-based authentication protocol for WLAN mesh security access. Computers, Materials and Continua, 58(1), 45–59. https://doi.org/10.32604/cmc.2019.03863
Wei, Q., Li, S., Li, W., Li, H., & Wang, M. (2019). Trustroam: A Novel Blockchain-Based Cross-Domain Authentication Scheme for Wi-Fi Access. 2(March 2019), 358–369. https://doi.org/10.1007/978-3-030-23597-0
Jamsrandorj, U. (2017). Decentralized Access Control Using Blockchain (Issue August). University of Saskatchewan.

Hanif, M., & Song, H. (2019). Blocks' Network: Redesign architecture based on blockchain technology. In Proceedings - IEEE International Conference on Industrial Internet Cloud, ICII 2019. https://doi.org/10.1109/ICII.2019.00017
Brincat, A. A., Lombardo, A., Morabito, G., & Quattropani, S. (2019). On the use of Blockchain technologies in WiFi networks. Computer Networks, 162. https://doi.org/10.1016/j.comnet.2019.07.011
Beck, R., & Müller-bloch, C. (2017). Blockchain as Radical Innovation : A Framework for Engaging with Distributed Ledgers 2 . Literature Background : Blockchain as. Proceedings of the 50th Hawaii International Conference on System Sciences,
Bergquist, J. (2017). Blockchain Technology and Smart Contracts: Privacy-preserving Tools
Casas, D. L., Alfonso, J., Torralbo, L., García, C., Casas, D. M., Rumayor, R., Manuel, J., & López, V. (2019). Aproximación basada en Blockchain para crear un modelo de confianza en la enseñanza superior abierta y ubicua A trust model in open and ubiquitous higher education based on Blockchain technology.
Frutos, J. (2019). Desarrollo de un servicio de seguimiento de mercancías basado en la Cadena de Bloques Ethereum.
Kaku, E. (2017). Using Blockchain To Support Provenance in the Internet of Things.
Romero Solís, J. (2019). Aplicaciones de Contratos Inteligentes en Ethereum Universidad Carlos III de Madrid.

NOMENCLATURA DE REQUERIMIENTOS

Requerimiento	Abreviatura
Stakeholders	STSR
Sistema	SYSR
Arquitectura	SRSH

REQUERIMIENTOS DE STAKEHOLDERS			
Requerimientos Operacionales			
Nomenclatura	**Requerimiento**	**Descripción**	**Prioridad**
STSR1	Se implementará un mecanismo de autenticación basado en la tecnología de cadena de bloques en la red inalámbrica de la FICA.	La aplicación de la Tecnología de Cadena de Bloques como alternativa al uso de los métodos de autenticación WiFi conocidos, esta propuesta innovadora, plantea una alternativa descentralizada de validación y verificación de credenciales según el documento "Trustroam: A Novel Blockchain-Based Cross-Domain Authentication Scheme for Wi-Fi Access" (Wei et al., 2019).	Baja

STSR2	El mecanismo de autenticación propuesto, basado en la tecnología de cadena de bloques debe usar los equipos e infraestructura inalámbrica desplegado en la FICA.	La integración de los equipos inalámbricos de red en un entorno de tecnología de Cadena de Bloques se menciona por (Niu et al., 2018); donde se presenta un enfoque de autenticación y acceso WiFi en base a una cadena de bloques de Bitcoin.	Media
STSR3	El mecanismo de autenticación propuesto, basado en la tecnología de cadena de bloques debe permitir que los usuarios se conecten a la infraestructura inalámbrica de red de la FICA.	El sistema descentralizado y transparente presentado por (Wei et al., 2019), también propone el acceso de usuarios a redes WiFi, de manera confiable sin tener que realizar procesos complicados y con la respectiva seguridad y protección.	Media
STSR4	El mecanismo de autenticación propuesto, basado en la tecnología de cadena de bloques debe interactuar con el SSID "eduroam" y/o el servidor Radius/LDAP del DDTI.	Según (Thakur, 2017), su propuesta explora el uso de la tecnología de Cadena de bloques, para mejorar los procesos de autenticación, autorización y contabilidad, procesos que lleva a cabo un servidor centralizado triple A, usado comúnmente como entidad de confianza.	Baja
STSR5	Se creará un entorno de pruebas para el mecanismo de autenticación propuesto, basado en la tecnología de cadena de bloques en la red inalámbrica de la FICA.	En la propuesta (Sanda & Inaba, 2016), se presenta un enfoque innovador para la autenticación de redes WiFi, utilizando la tecnología Bitcoin 2.0 y las transacciones como prueba de identidad de usuario.	Alta
STSR6	El SSID de conexión y acceso al entorno de pruebas en la red inalámbrica de la FICA, debe estar disponible para la detección de los dispositivos de los usuarios.	(Jiang et al., 2019), propone el uso de la tecnología de Cadena de Bloques para verificar y autenticar de manera confiable los dispositivos y usuarios que intentan acceder a la una red inalámbrica de área local en malla (WLAN mesh). La convergencia de las tecnologías de cadena de bloques y WiFi, pueden proporcionar un método confiable y seguro para verificar la identidad de los usuarios.	Alta

STSR7	El mecanismo de autenticación propuesto, basado en la tecnología de cadena de bloques debe interactuar con el estándar 802.11 para permitir el acceso y conexión WiFi al entorno de pruebas en la infraestructura inalámbrica de red de la FICA.	El protocolo presentado por (Jiang et al., 2019), basado en la tecnología de Cadena de Bloques, muestra como se usa para verificar y autenticar de manera confiable los dispositivos y usuarios que intentan acceder a la red en malla inalámbrica, que se tomara como referencia en la solución de los estos requerimientos.	Alta
STSR8	Los usuarios podrán acceder a Internet dentro del entorno de pruebas desplegado para mecanismo de autenticación propuesto, basado en la tecnología de cadena.		Alta
STSR9	Se empleará herramientas de simulación para desarrollar el mecanismo de autenticación propuesto, basado en la tecnología de cadena de bloques en la red inalámbrica de la FICA.	(Jamsrandorj, 2017), explora como la tecnología de Cadena de Bloques puede usarse para establecer un sistema de control de acceso descentralizado, presentando un enfoque donde, los registros de acceso y las políticas de control se almacenan en una Cadena de Bloques, garantizando la integridad y la transparencia en el proceso de autorización, que se adaptaran en la solución y desarrollo de este proyecto.	Alta
STSR10	La cadena de bloques validara la conexión y acceso de usuarios al entorno de pruebas en la red inalámbrica de la FICA.		Alta
Requerimientos de Usuario			
Nomenclatura	**Requerimiento**	**Descripción**	**Prioridad**
STSR11	Los usuarios podrán conectare red inalámbrica de la FICA, usando el mecanismo de autenticación propuesto.	El uso de la tecnología de Cadena de Bloques en redes WiFi, como la autenticación segura, el intercambio de credenciales y la administración de servicios. (Brincat et al., 2019), explora los desafíos y las consideraciones asociadas con la implementación de una Cadena de Bloques en este contexto.	Baja

STSR12	Los usuarios podrán conectarse al entorno de pruebas en la red inalámbrica de la FICA, usando el mecanismo de autenticación propuesto, basado en la tecnología de cadena de bloques.	(Sanda & Inaba, 2016), describe un método en el que los usuarios pueden autenticarse y acceder a redes Wi-Fi utilizando transacciones de Bitcoin como prueba de identidad. Este enfoque aprovecha las características de seguridad y descentralización de la tecnología Bitcoin 2.0 para crear un sistema confiable y resistente a ataques.	Alta
STSR13	Los usuarios deberán disponer de credenciales de conexión para poder conectarse al entorno de pruebas en la red inalámbrica de la FICA, mediante el mecanismo de autenticación propuesto, basado en la tecnología de cadena de bloques.	(Thakur, 2017), explora el uso de la tecnología de Cadena de Bloques de Ethereum para mejorar los procesos de autenticación, autorización y contabilidad en sistemas informáticos. En su artículo destaca cómo la implementación de contratos inteligentes en la plataforma Ethereum permite crear un sistema seguro y confiable para la gestión de identidad y acceso.	Media
STSR14	Los usuarios podrán conectarse al entorno de pruebas en la red inalámbrica de la FICA, usando el mecanismo de autenticación propuesto, basado en la tecnología de cadena de bloques; mediante una interfaz gráfica amigable.	Tomando como referencia el estudio de (Cañar & Jara, 2022), se describe como se analiza y desarrolla una aplicación basada en Cadena de Bloques para el registro de permisos y ausentismos. Centrándose en el uso de un contrato inteligente desplegado en una red de prueba de Ethereum para garantizar la transparencia y la inmutabilidad de los registros. Explorando la implementación de una aplicación, junto con los beneficios y desafíos asociados al uso de Cadena de Bloques.	Media
STSR15	Los usuarios podrán conectarse al entorno de pruebas en la red inalámbrica de la FICA, usando el mecanismo de autenticación propuesto, basado en la tecnología de cadena de bloques; desde diferentes dispositivos y/o sistemas operativos		Media

REQUERIMIENTOS DE SISTEMA			
Requerimientos de Interfaz			
Nomenclatura	**Requerimiento**	**Descripción**	**Prioridad**
SYRS1	El mecanismo de autenticación propuesto debe interactuar con al menos un dispositivo inalámbrico de red (Puntos de Acceso), para permitir la conexión al entorno de pruebas de la FICA.	Tener una convergencia entre la tecnología de Cadena de Bloques y la tecnología WiFi de conexión inalámbrica bajo el estándar IEEE 802.11, es fundamental para el desarrollo de un mecanismo de autenticación inalámbrica basada en cadena de bloques.	Alta
SYRS2	El equipo servidor o instancia donde se levantará el mecanismo de autenticación propuesto debe permitir la conexión y acceso remoto para la validación de conexión.	La conectividad remota y el acceso al equipo donde se levante el mecanismo de autenticación es importante debido a que desde ese equipo o instancia se deberá validar las credenciales de usuarios, además de que también deberá contar con conectividad desde y hacia cada uno de los equipos de red usados para implementar el mecanismo de autenticación.	Alta
SYRS3	El mecanismo de autenticación propuesto debe interactuar juntamente con los equipos de usuario, los de red y el servidor o instancia para brindar el acceso a la red inalámbrica de pruebas en la FICA.		Alta
SYRS4	La configuración de complementos o componentes adicionales para el mecanismo autenticación propuesto debe ser fácil de realizarse.	Como se pudo contemplar en diferentes trabajos de investigación, la adaptabilidad de la tecnología de Cadena de Bloques se usa para validar registros y en su mayoría de aplicaciones requiere un complemento o configuración adicional para enlazar una interfaz con una cripto billetera y poder usarla como medio de validación de cuentas y realización de transacciones, de tal manera se contempla integrar su uso en el mecanismo de autenticación propuesto.	Media

Requerimientos de Uso			
Nomenclatura	**Requerimiento**	**Descripción**	**Prioridad**
SYRS5	El usuario debe ingresar sus credenciales para conectarse a la red del mecanismo de autenticación propuesto.	El enfoque que se presenta para mejorar la seguridad en aplicaciones para dispositivos de Internet de las Cosas usando la tecnología de Cadena de Bloques y la infraestructura de clave pública. La combinación de estas tecnologías puede proporcionar autenticación segura, integridad de datos y confidencialidad en los dispositivos. La arquitectura propuesta, utiliza la Cadena de Bloques para el registro y la verificación de la identidad de los dispositivos y la infraestructura de clave pública para el cifrado y la autenticación de las comunicaciones. Analizando los beneficios de esta solución, como la resistencia a ataques y la garantía de la confidencialidad de los datos, así como los desafíos asociados con su implementación (Balmaseda Aranda, 2018); que se propone adecuar para el desarrollo de estos requerimientos.	Media
SYRS6	El mecanismo de autenticación propuesto debe permitir la conexión y acceso a Internet de los dispositivos de usuario.		Alta
SYRS7	La asignación de credenciales de usuario se realizará de manera previa para el mecanismo de autenticación propuesto.		Media
SYRS8	El mecanismo de autenticación propuesto debe interactuar con el SSID configurado previamente para la conexión de usuarios.		Alta
Requerimientos de Performance			
Nomenclatura	**Requerimiento**	**Descripción**	**Prioridad**
SYRS9	El mecanismo de autenticación propuesto debe permitir la conexión inalámbrica mediante un SSID propagado en el entorno de pruebas en la red inalámbrica de la FICA.	Como ya se explicó, la tecnología de Cadena de Bloques puede ser utilizada para establecer un sistema de control de acceso descentralizado. En muchos de los artículos revisados se presenta un enfoque en el que los registros de acceso y las políticas de control se almacenan en una cadena de bloques, lo que garantiza la integridad y la transparencia en el proceso de autorización (Jamsrandorj, 2017).	Alta
SYRS10	El tiempo empleado en el proceso de autenticación mediante el mecanismo de autenticación propuesto no debe ser prolongado.		Media

SYRS11	El mecanismo de autenticación propuesto debe levantarse en un equipo servidor o instancia virtual dedicada.	Para poder realizar las pruebas del mecanismo basado en la tecnología de Cadena de Bloques se plantea el despliegue de la tecnología dentro del entorno de red de pruebas para la FICA.	Media
Requerimientos de Físicos			
Nomenclatura	**Requerimiento**	**Descripción**	**Prioridad**
SYRS12	El punto de acceso usado para el mecanismo de autenticación propuesto debe ubicarse estratégicamente para que propague la señal inalámbrica.	Como se ha propuesto para el despliegue del mecanismo de autenticación se debe contemplar el correcto despliegue e instalación de equipos de red o el uso de los ya desplegados, incluyendo usar instancias o servidores de la infraestructura de red de la FICA.	Alta
SYRS13	La ubicación del servidor o instancia donde se configurará el mecanismo debe ser estratégica y segura.		Media
Requerimientos de Modos y Estado			
Nomenclatura	**Requerimiento**	**Descripción**	**Prioridad**
SYRS14	El mecanismo de autenticación propuesto y el SSID configurado deben estar disponibles para la conexión.	Para el correcto uso y acceso mediante el mecanismo de autenticación basado en la tecnología de cadena de bloques, se debe configurar y propagar un identificador de red para que los usuarios puedan conectarse mediante el mecanismo.	Alta

REQUERIMIENTOS DE ARQUITECTURA			
Requerimientos de Diseño			
Nomenclatura	**Requerimiento**	**Descripción**	**Prioridad**
SRSH1	El mecanismo de autenticación propuesto autorizará la conexión de usuarios y equipos a la red de pruebas inalámbrica de la FICA.	Como lo describe (Wei et al., 2019), su enfoque permite a los usuarios acceder de manera segura a redes confiables WiFi, sin tener que pasar por complicados procesos de autenticación. El modelo Trustroam mejora tanto la seguridad de la conexión WiFi, al tiempo que reduce la carga administrativa.	Alta

SRSH2	El mecanismo de autenticación propuesto gestionara los equipos y usuarios conectados a la red inalámbrica configurada en la FICA.	(Montoya, 2021), explora los beneficios de utilizar la tecnología de Cadena de Bloques en términos de transparencia, seguridad y resistencia a la falsificación. Además, plantea los desafíos y consideraciones relacionados con la implementación de la tecnología de Cadena de Bloques en el contexto del transporte inteligente; parámetros que se tomaran en cuenta al adaptarlo al mecanismo de autenticación basada en cadena de bloques.	Baja
SRSH3	El mecanismo de autenticación propuesto auditara los equipos y usuarios conectados a la red inalámbrica configurada en la FICA.		Baja
SRSH4	El mecanismo de autenticación propuesto permitirá administrar y configurar credenciales de usuarios.	Adoptar el uso de la cuentas y direcciones que intervienen en la tecnología de Cadena de Bloques para generar transacciones y validarlas, permitirá que el mecanismo propuesto, administre con su entrega el acceso de usuarios a la red de pruebas desplegada en la FICA.	Media
Requerimientos Lógicos			
Nomenclatura	**Requerimiento**	**Descripción**	**Prioridad**
SRSH5	La red inalámbrica configurada para el mecanismo de autenticación propuesto deberá ser propagada y estar disponible en la FICA.	Para cumplimiento de los objetivos propuestos, se debe levantar un entorno de pruebas en la FICA y desplegar el mecanismo de autenticación basado en cadena de bloques en esta red.	Alta
SRSH6	Se configurará extensiones y complementos en los dispositivos para permitir la conexión a la red inalámbrica configurada en la FICA.	Como ya se pudo apreciar en la bibliografía revisada, muchos autores mencionan en sus propuestas el uso de herramientas de completo, tanto para la configuración de una cadena de bloques, como para hacer pruebas en este entorno, por lo que se hace necesario; buscar y hacer uso de una extensión que permita la interacción de las cuentas de usuario con el entorno de cadena de bloques y la interfaz de acceso y registro de usuario.	Media
SRSH7	Se requerirá de una aplicación para la conexión de dispositivos a la red inalámbrica configurada en la FICA.	Así mismo algunos autores desarrollan y diseñan aplicaciones de usuario final o entornos web que ayudan a los usuarios al ingreso amigable al entorno de cadena de bloques, por lo que se contempla el despliegue de este tipo de interfaces.	Media

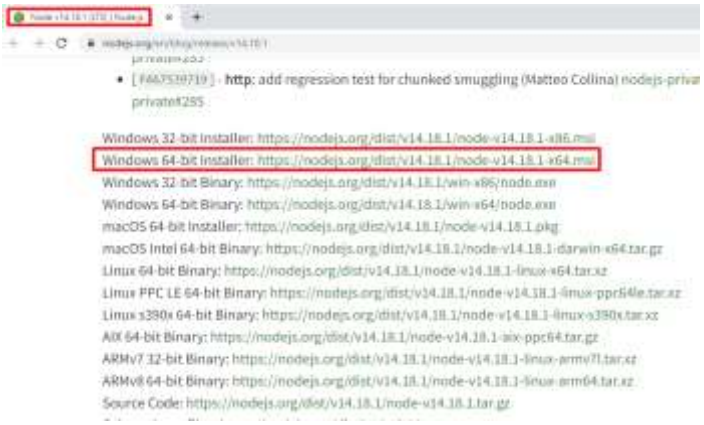
Requerimientos de Hardware			
Nomenclatura	**Requerimiento**	**Descripción**	**Prioridad**
SRSH8	El equipo para configurar el mecanismo de autenticación propuesto debe disponer de memoria de alto procesamiento.	En un entorno real de producción levantar un ambiente de Cadena de Bloques, requiere de gran cantidad de recursos, ya que se necesitan equipos unos dedicados a ser parte y formar la red de pares y que se encargan de generar transacciones y validarlas, lo que requiere capacidad de procesamiento, almacenamiento y alto rendimiento de sus componentes, sin contar con una conexión de red y acceso a Internet estable, ya que los diferentes nodos que forma parte de una Cadena de Bloques se distribuyen alrededor del mundo, por lo que según algunos autores, para lograr un entorno de pruebas de cadena de bloques en un ambiente controlado, se requiere un equipo que cumpla con los requerimientos mínimos mencionados.	Alta
SRSH9	El equipo para configurar el mecanismo de autenticación propuesto debe disponer de un procesador de alto rendimiento.		Alta
SRSH10	El equipo para configurar el mecanismo de autenticación propuesto debe disponer de una tarjeta gráfica.		Baja
SRSH11	El equipo para configurar el mecanismo de autenticación propuesto debe disponer de gran espacio de almacenamiento.		Media
SRSH12	El equipo para configurar el mecanismo de autenticación propuesto debe disponer de interfaz de conexión de red de alta velocidad.		Alta
SRSH13	El punto de acceso inalámbrico configurado deberá ser compatible con el estándar IEEE 802.11 y sus versiones b, n, ac y ax.	Tomando en cuenta el análisis de situación actual de red de la FICA, se determinó que los equipos desplegados en la infraestructura de red presentan las estas características de red respecto a la tecnología WiFi y su respectivo estándar, por lo que se debe tomar en cuenta estos parámetros al configurar los equipos del entorno de pruebas.	Alta
SRSH14	El punto de acceso inalámbrico configurado trabajara en las bandas de 2.4 y 5 GHz.		Alta

Requerimientos de Software			
Nomenclatura	**Requerimiento**	**Descripción**	**Prioridad**
SRSH15	El mecanismo de autenticación propuesto debe ser compatible con sistema operativo Windows o Linux.	(Cañar & Jara, 2022), en su planteamiento usan la plataforma Ethereum, que a su vez utiliza Ganache el framework Truffle, los cuales permiten la instalación y despliegue de un entorno de cadena de bloques en equipos con sistemas operativos de licencia libre como propietarios.	Media
SRSH16	Compatibilidad del sistema operativo con lenguajes de programación de código abierto.	La implementación de una cadena de bloques depende del lenguaje de programación que se use, (Cañar & Jara, 2022) para la creación del entorno de cadena de bloques propone Node.js y npm, con el compilador Solidity; los cuales son complemento para la plataforma Ethereum.	Alta
SRSH17	El software deberá permitir compilar y ejecutar librerías de la tecnología de cadena de bloques.	La librería Web3 JS y Truffle Contract, se usa para enlazar la cadena de bloques con la validación de transacciones.	Alta
SRSH18	Herramientas de software compatibles para el diseño de la interfaz gráfica.	Para el diseño de una aplicación o interfaz web de usuario (Cañar & Jara, 2022), usan Bootstrap y React, enlazándose con Metamask que es un complemento de una cripto billetera.	Media

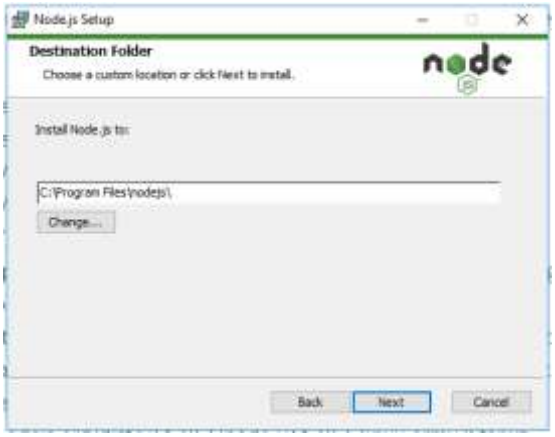
Anexo 4: Componentes

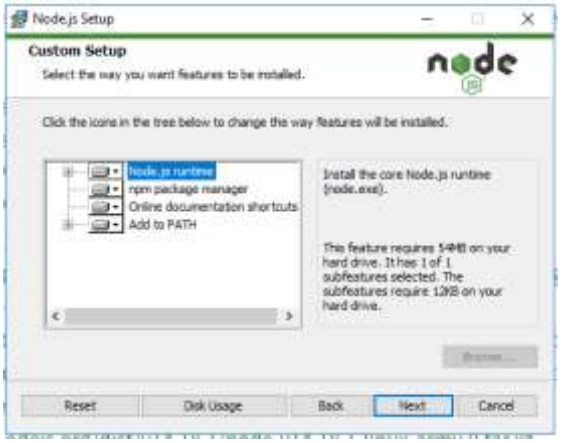
Anexo 4.1: Instalación de Node.js y npm

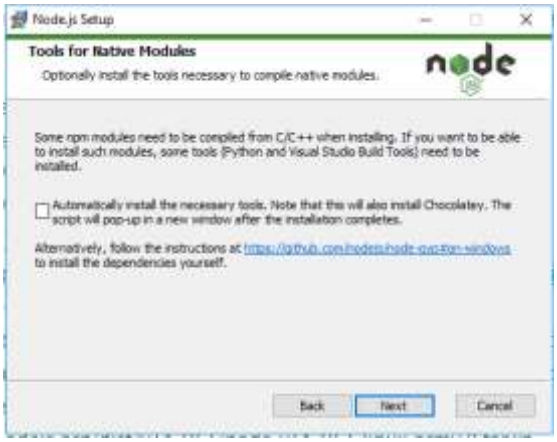
Descargar el instalador del sitio web “www.node.org” de Node.js y ejecutarlo.

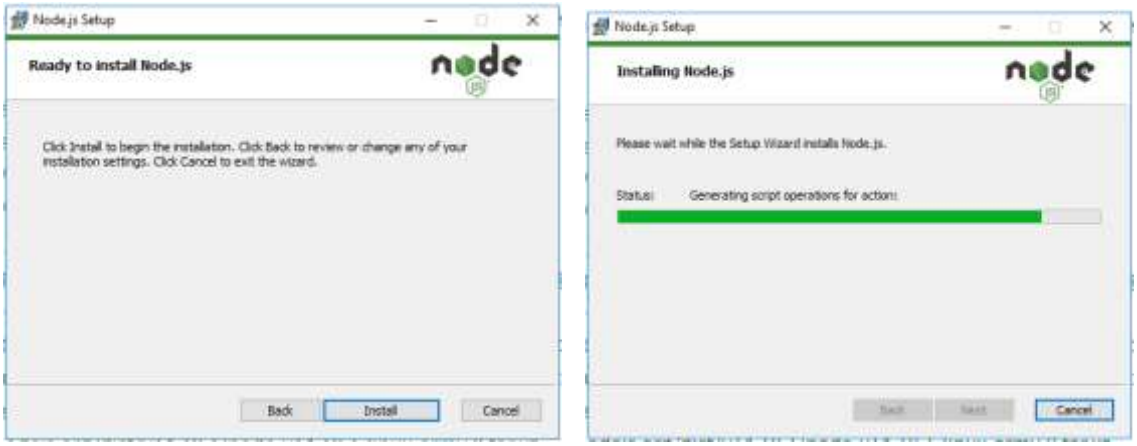


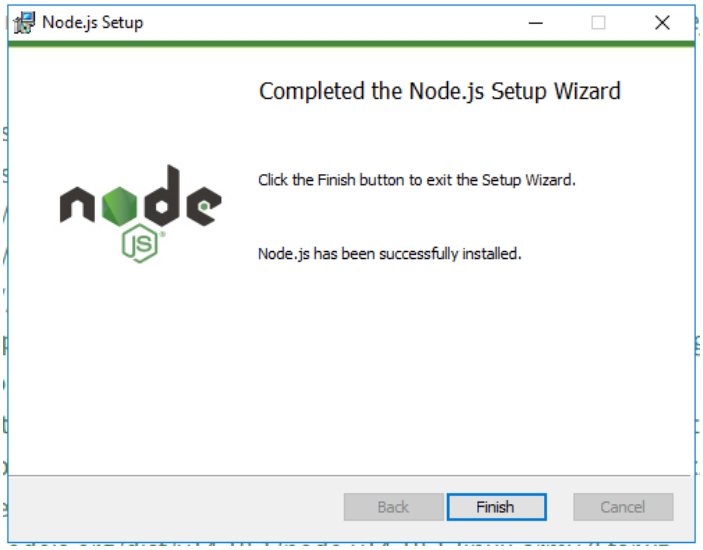
Realizar la instalación habitual en Windows.



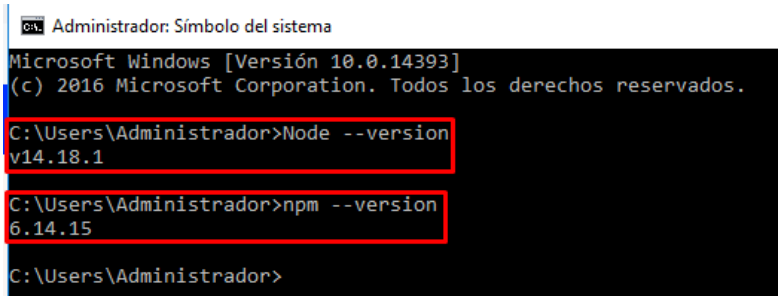






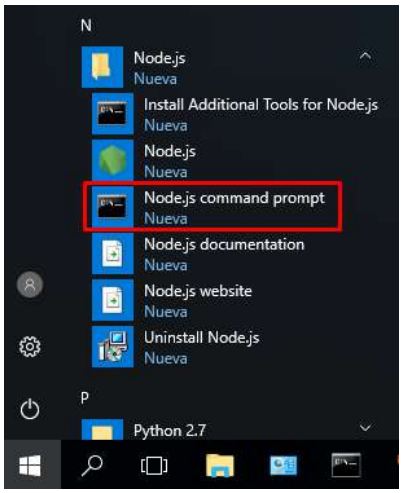


Verificación de instalación de Node.js y npm.

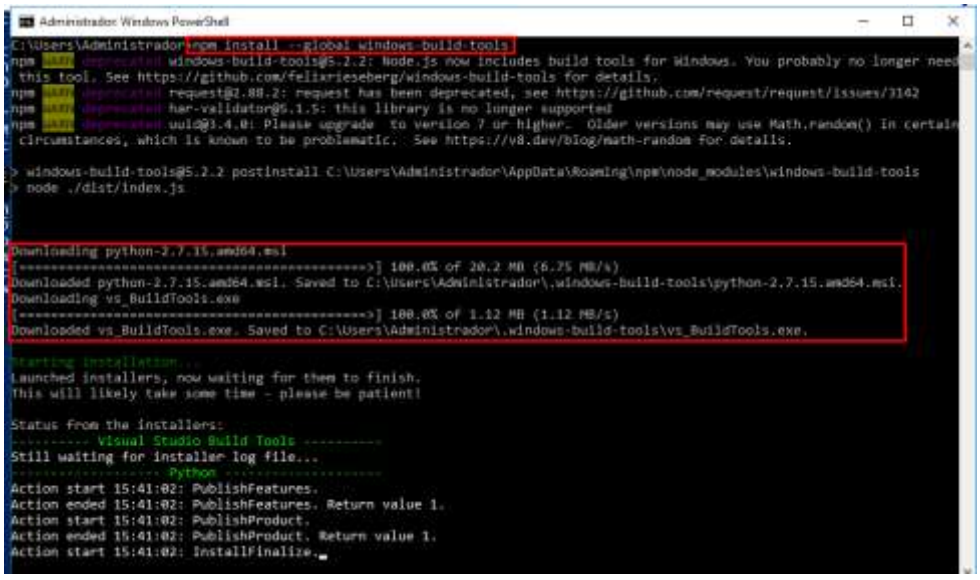


Anexo 4.2: Instalación de herramientas C/C++ en npm Node.js

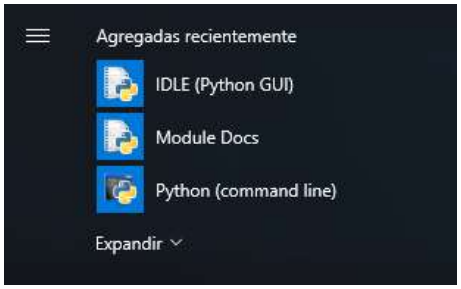
Ejecutar la interfaz de línea de comando de Node.js que se encuentra en la barra de inicio de Windows.



Digitar el comando: “npm install --global windows-build-tools”, el cual empezara a descargar las herramientas complementarias de Node.js y npm.



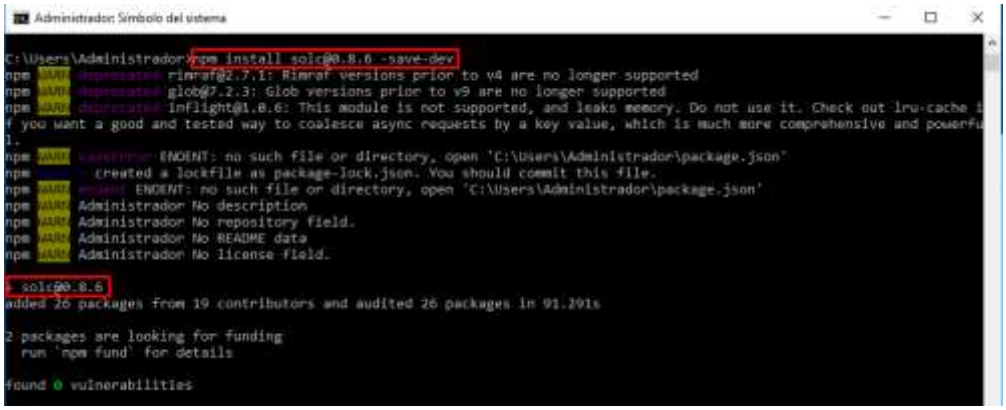
Verificar que se ha instalado el compilador de Python en la barra de inicio de Windows.



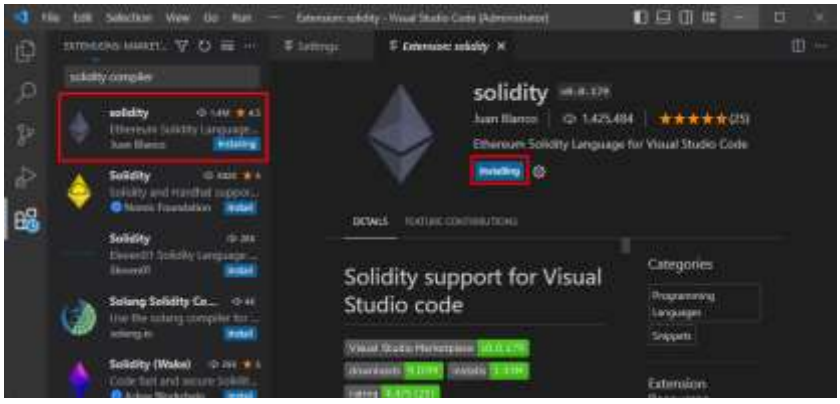
Anexo 4.3: Instalación y configuración del compilador Solidity

Verificar si se ha instalado la versión de Solidity 0.8.6 con el comando “npm show solc versions”.

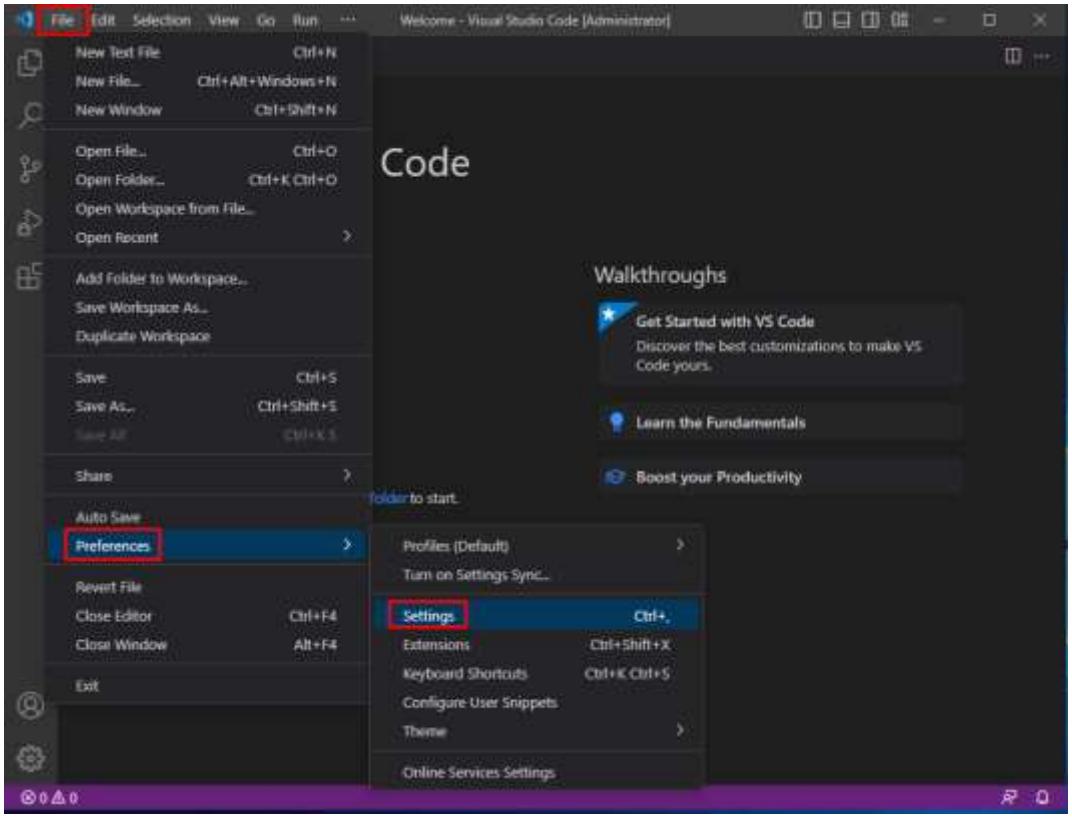
En caso de no existir la versión la instalamos con el comando “npm install solc@0.8.6 --sabe-dev” en la línea de comandos de Windows Power Shell o Node.js command prompt.



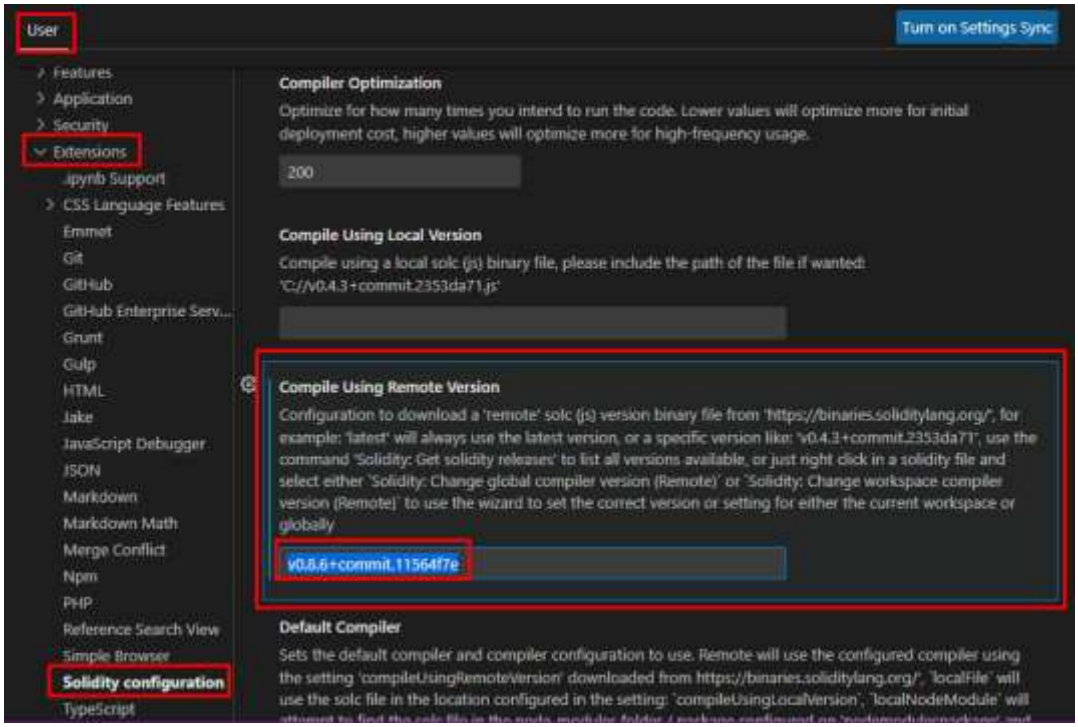
Para la configuración de la versión especifica de Solidity en Visual Studio Code, debemos instalar la extensión de configuración de Solidity desde el repositorio de extensiones.



Nos dirigimos al menú “File -> Preferences -> Settings”

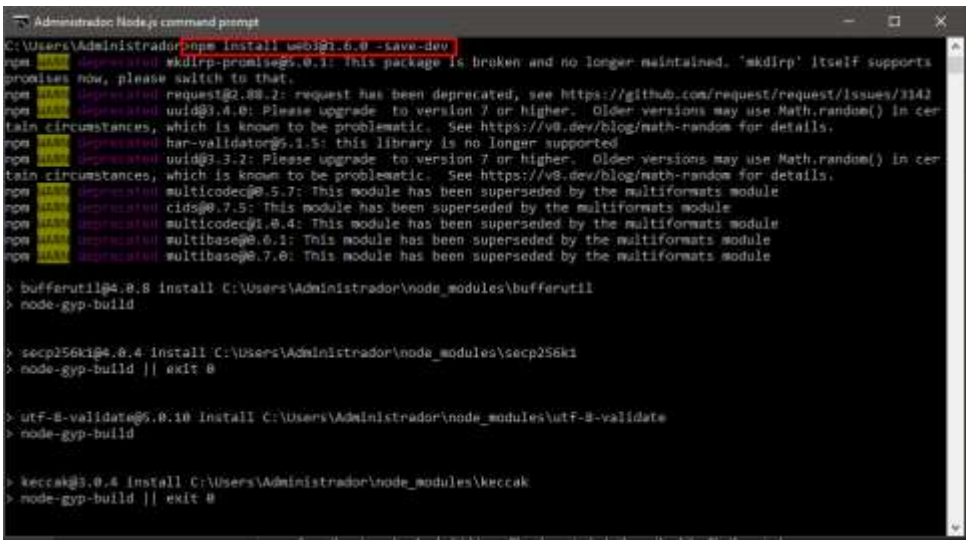


En la pestaña “User -> Extensions -> Solidity configuration -> Compile Using Remote Version” y digitar “v0.8.6+commit.11564f7e”.

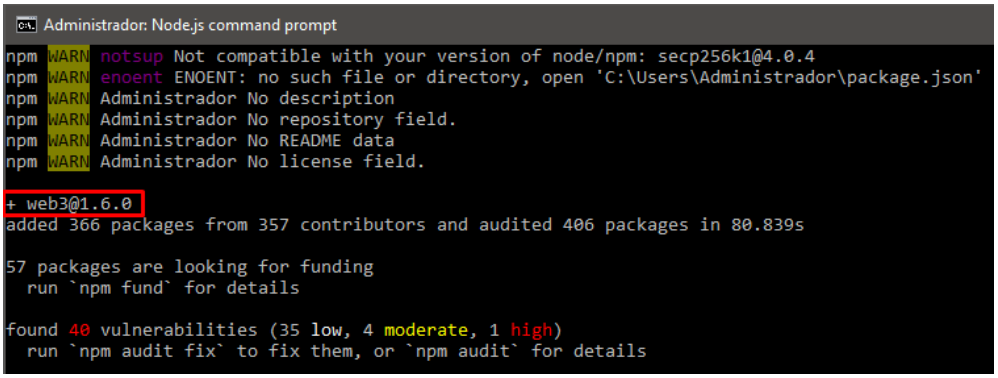


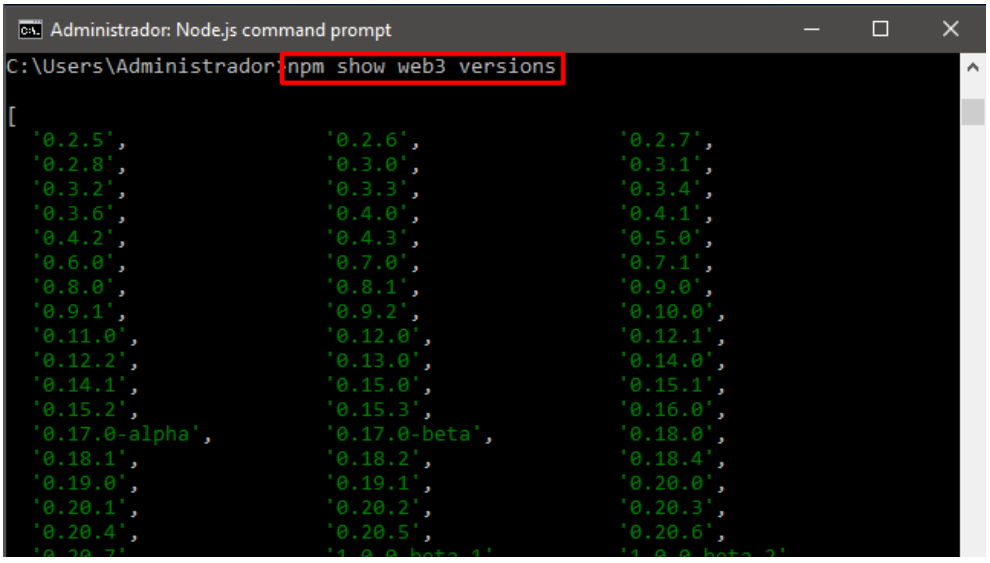
Anexo 4.4: Instalación de Web3 Js

Ejecutar el comando “npm install web3@1.6.0 –save-dev” en Windows Power Shell o o Node.js command prompt.



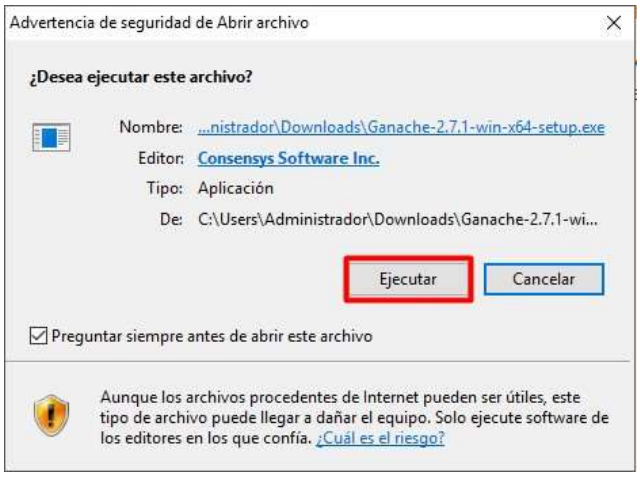
Verificar la instalación al finalizar el proceso o con el comando “npm show web3 versions”.



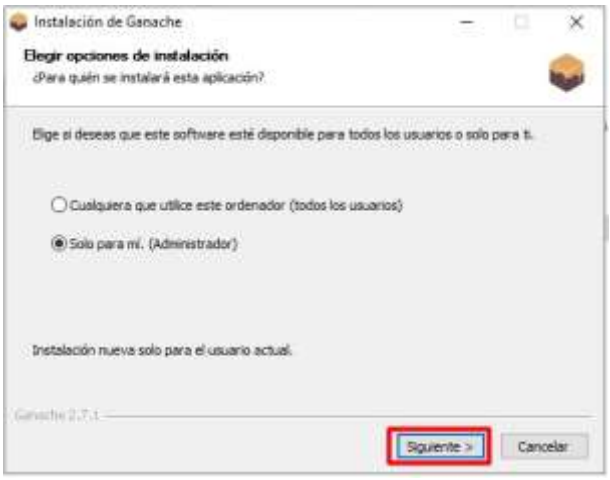


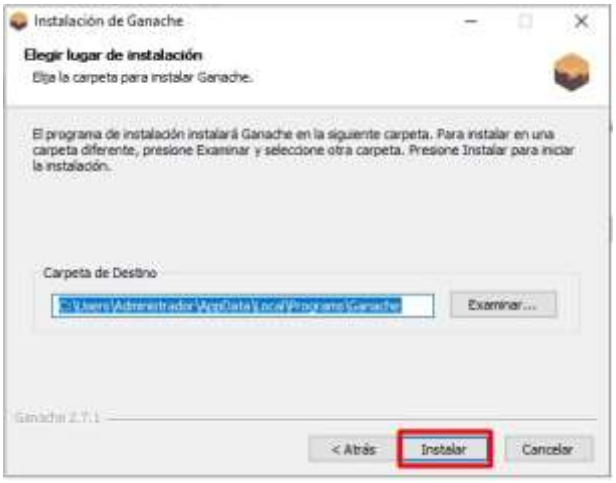
Anexo 4.5: Instalación de servidor Ganache

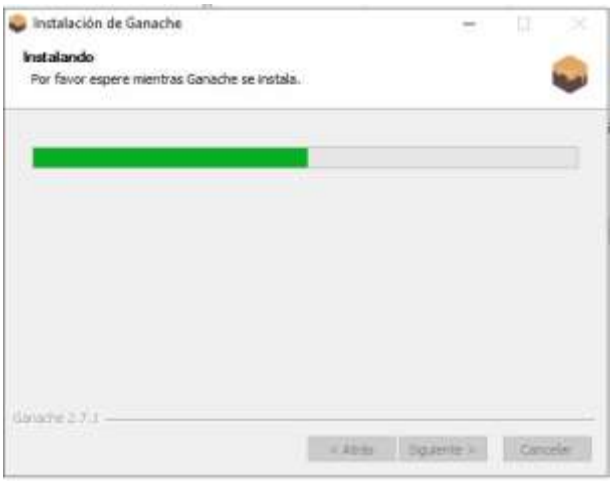
Descargar el instalador del sitio web “archive.trufflesuite.com/ganache” de Ganache y ejecutarlo.



Realizar la configuración e instalación habitual realizada den Windows

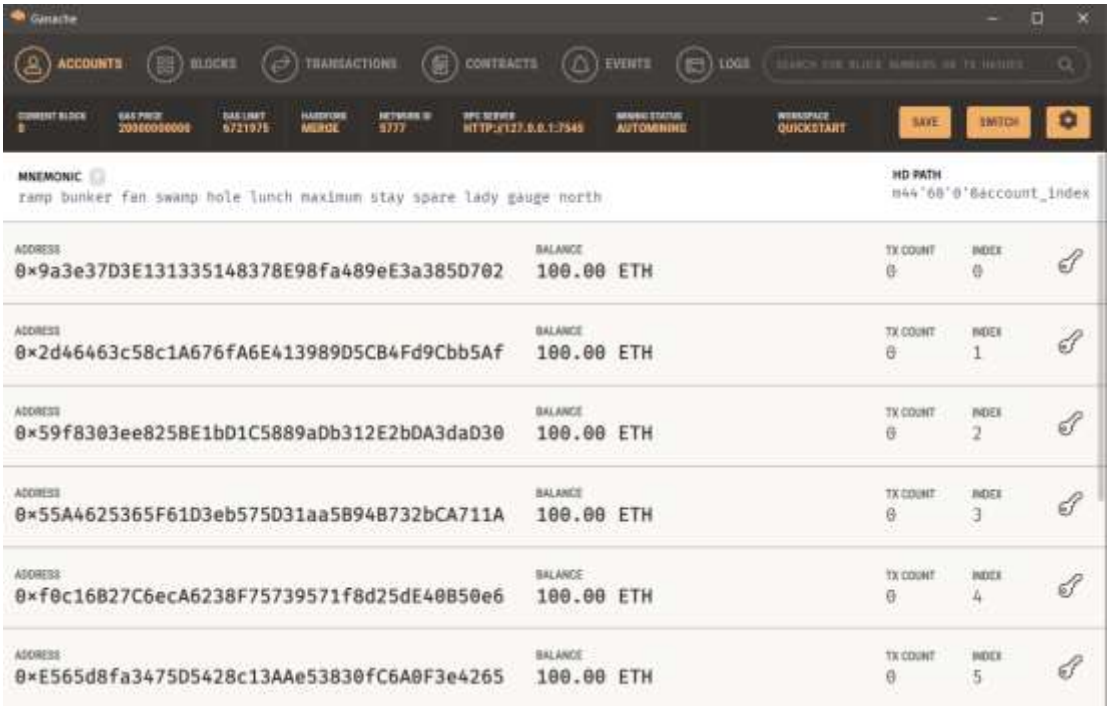






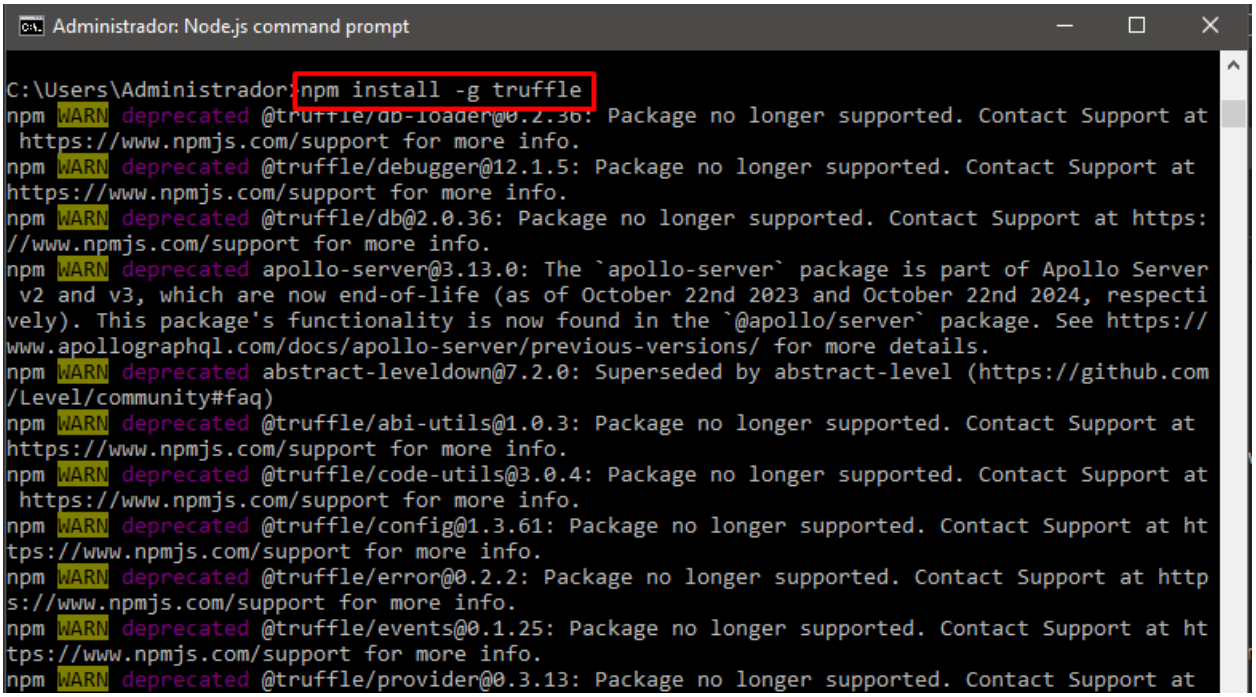
Al finalizar la instalación se ejecutará la aplicación de Ganache donde se mostrará las opciones de cadenas de bloques de prueba disponibles.

Al escoger un entorno de cadena de bloques Ethereum, nos muestra las cuentas de prueba disponibles en la interfaz de grafica de Ganache.

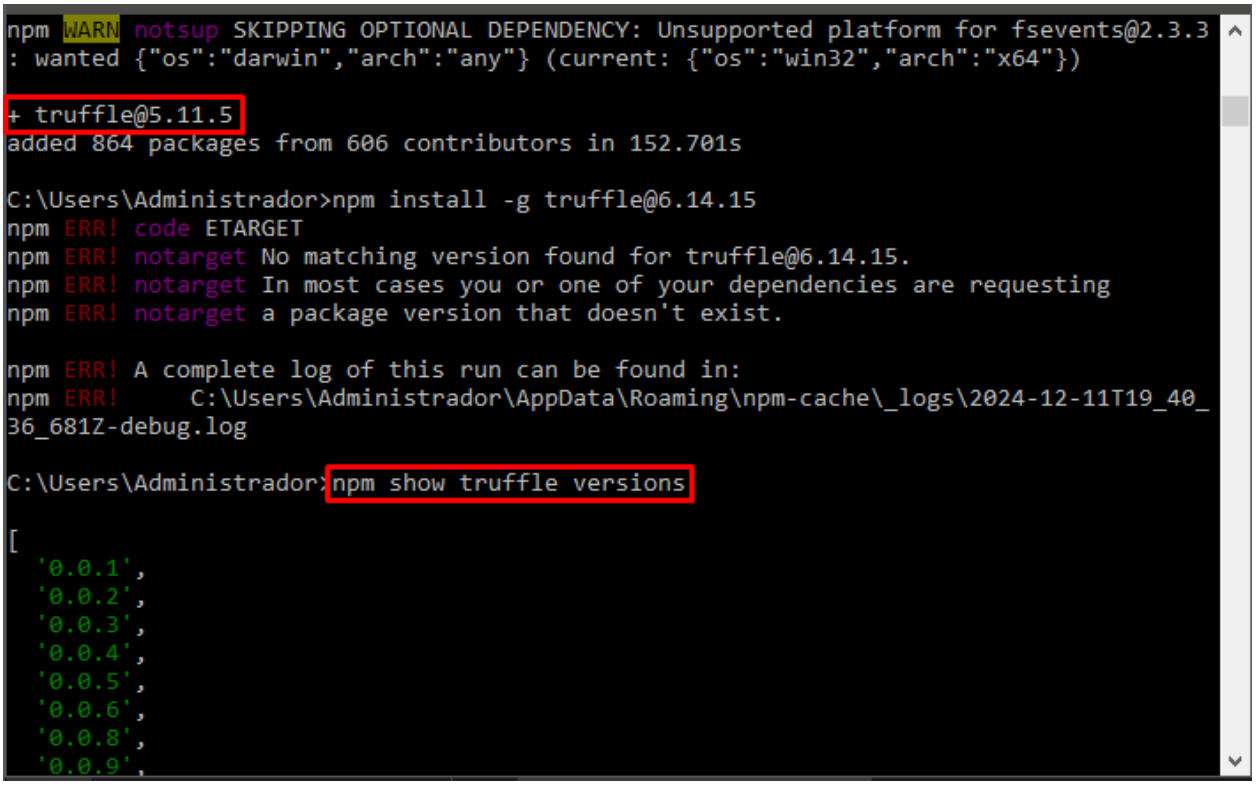


Anexo 4.6: Instalación de Truffle Framework

Para instalar el framework Truffle, se debe ejecutar la línea de comando "npm install -g truffle" en el Windows Power Shell o Node.js command prompt.

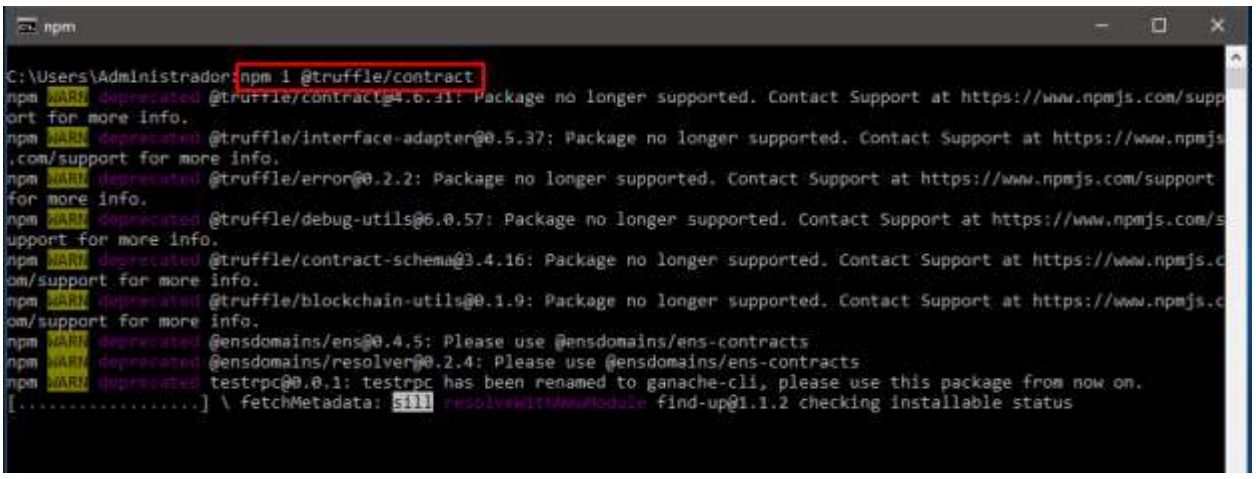


Verificar la instalación al finalizar el proceso o con el comando "npm show truffle versions".

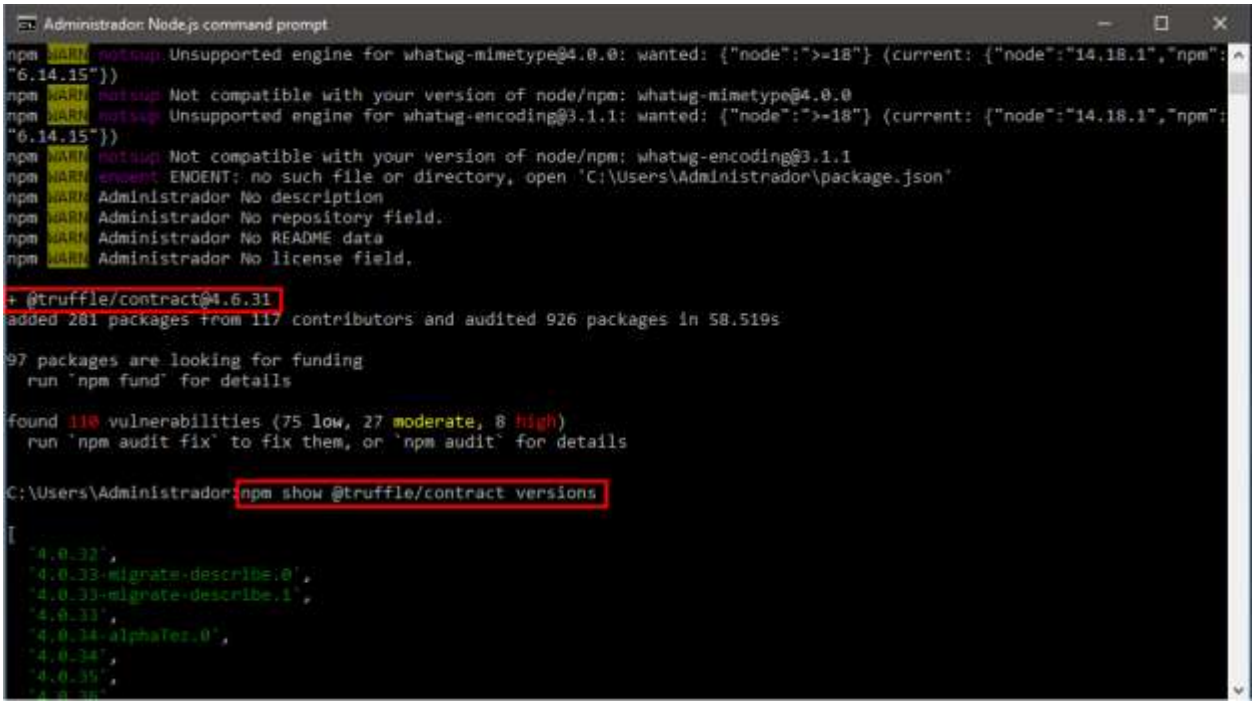


Anexo 4.7: Instalación de Truffle Contract

Para instalar el framework Truffle, se debe ejecutar la línea de comando “npm i @truffle/contract” en el Windows Power Shell o Node.js command prompt.



Verificar la instalación al finalizar el proceso o con el comando “npm show @truffle/contract versions”.

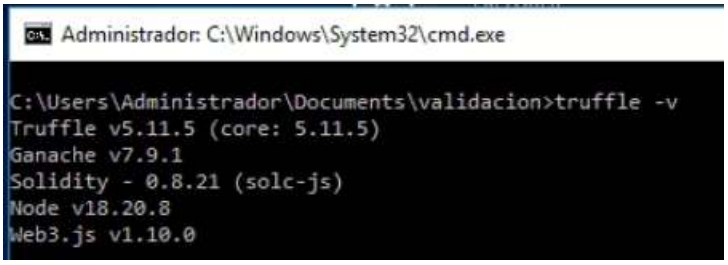


Anexo 4.8: Backend

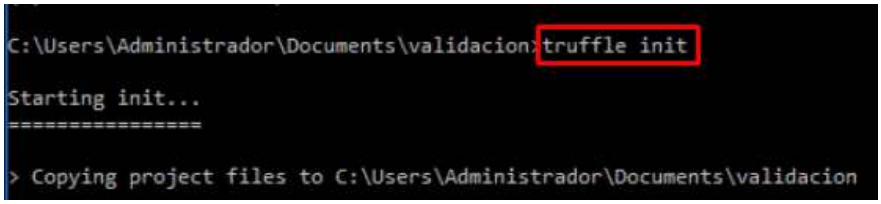
La estructura para el directorio que contiene el desarrollo del backend es defino como “blockchain” y se muestra a continuación.

```
├── blockchain/                    # Smart Contracts y config. Truffle
│   ├── contracts/                 # Contratos Solidity
│   │   ├── AccessControl.sol      # Contrato principal
│   │   └── Migrations.sol         # Contrato de migraciones (auto)
│   ├── migrations/                # Scripts de despliegue
│   │   ├── 1_initial_migration.js
│   ├── build/                     # Contratos compilados (se crea)
│   │   └── contracts/
│   │       └── AccessControl.json
│   ├── test/                      # Tests del contrato
│   ├── truffle-config.js          # Configuración Truffle
│   └── package.json               # Dependencias blockchain
```

Se debe instalar las herramientas de software requeridas y necesarias; los complementos Truffle, Ganache, Solidity, Node.js, Web3.js o Ether.js; instalarlos en el servidor Windows, la verificación de las versiones de los componentes instalados se realiza con el comando “truffle -v”.



La creación del entorno de pruebas de cadena de bloques inicia al crear los componentes del contrato inteligente; mediante la consola se ejecuta el comando “truffle init”, el cual crea los directorios y archivos necesarios para la compilación y despliegue del contrato inteligente.



Al crear y modificar el contenido del archivo “AccessControl.sol” dentro del directorio que contiene los componentes del contrato inteligente, se establecen los parámetros y configuraciones del contrato inteligente de despliegue, para validar las direcciones de cuentas Ethereum.

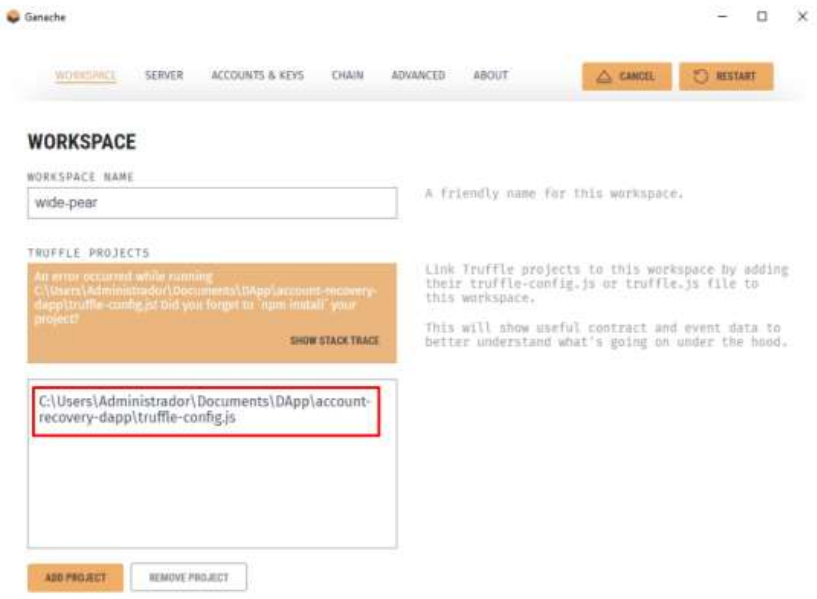
```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract AccessControl {
```

```
    address public owner;
    mapping(address => bool) private allowedAccounts;
    mapping(address => uint256) public accessTimestamps;

    event AccountAllowed(address indexed account);
    event AccountRevoked(address indexed account);
    event AccessGranted(address indexed account, uint256 timestamp);

    constructor() {
        owner = msg.sender;
    }

    modifier onlyOwner() {
        require(msg.sender == owner, "No propietario");
        _;
    }

    function addAllowedAccount(address _account) external onlyOwner {
        allowedAccounts[_account] = true;
        emit AccountAllowed(_account);
    }

    function removeAllowedAccount(address _account) external onlyOwner {
        allowedAccounts[_account] = false;
        emit AccountRevoked(_account);
    }

    function isAccountAllowed(address _account) external view returns (bool) {
        return allowedAccounts[_account];
    }

    function grantAccess(address _account) external {
        require(allowedAccounts[_account], "Cuenta no permitia");
        accessTimestamps[_account] = block.timestamp;
        emit AccessGranted(_account, block.timestamp);
    }

    function batchAddAccounts(address[] calldata _accounts) external onlyOwner {
        for (uint i = 0; i < _accounts.length; i++) {
            allowedAccounts[_accounts[i]] = true;
            emit AccountAllowed(_accounts[i]);
        }
    }
}
```

La configuración del archivo “Truffle-config.js”, el cual permite desplegar el contrato dentro de la cadena de bloques de prueba y con la interacción de entornos y herramientas que permite Visual Studio Code, se modifican los archivos y scripts necesarios para la plataforma Truffle y Ganache.

```
module.exports = {
    networks: {
      development: {
        host: "192.168.88.248", //Derección del servidor
        port: 7545,             //Puerto Ethereum
        network_id: "*",        //ID de red “*” para cualquier red
      },
    },
    compilers: {
      solc: {
        version: "0.8.0",
        settings: {
          optimizer: {
            enabled: true,
            runs: 200
          }
        }
      }
    }
  };
```

Para usar inicialmente el entorno de cadena de bloques, es necesario enlazar el archivo “Truffle-config.js” con la plataforma Ganache, agregando la ruta del directorio para que su despliegue permita generar transacciones iniciales en la cadena de bloques de prueba, de acuerdo con lo establecido en el contrato inteligente.

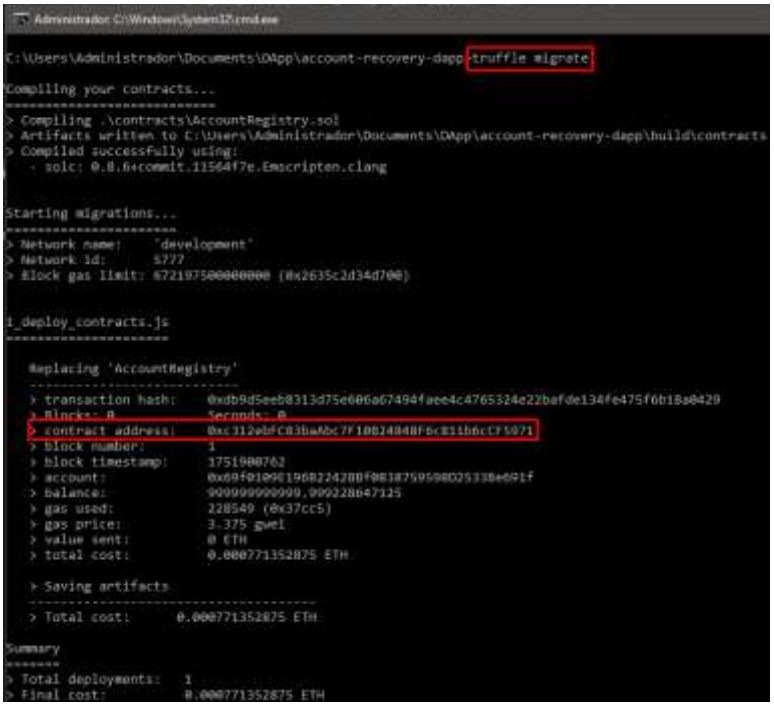


El despliegue del contrato inteligente se configura el archivo “2_depoy_contracts.js”; el cual llama, recupera y despliega los procesos establecidos en el archivo “AccessControl.sol”.

```
const AccessControl = artifacts.require("AccessControl");

module.exports = function(deployer) {
    deployer.deploy(AccessControl);
};
```

La compilación del contrato inteligente se realiza con el comando “truffle compile”, donde se realiza una comprobación previa de componentes y errores, que realizan el proceso y despliegan un mensaje en la consola.

El despliegue del contrato inteligente, se ejecuta el comando “truffle migrate” en una terminal de consola en la carpeta de inicialización. Este proceso es el resultado de la compilación del contrato inteligente y muestra los parámetros de inicialización de la migración del contrato, las características de despliegue y los atributos de la transacción inicial realizada; el valor del componente resaltado “contract address”, es utilizado para enlazar la interfaz web con el entorno Ganache.



Anexo 4.9: Frontend

La estructura para el directorio que contiene el desarrollo del frontend y se muestra a continuación.

```
├── frontend/                    # Aplicación React + Vite
│   ├── public/                  # Archivos estáticos
│   │   └── index.html
│   ├── src/                     # Código fuente
│   │   ├── components/          # Componentes React
│   │   │   ├── Login.tsx
│   │   │   └── Validation.tsx
│   │   ├── services/            # Servicios
│   │   │   └── blockchain.ts
│   │   ├── App.tsx              # Componente principal
│   │   ├── main.tsx             # Punto de entrada
│   │   └── vite-env.d.ts        # Tipos Vite
│   ├── package.json             # Dependencias frontend
│   ├── vite.config.ts           # Configuración Vite
│   ├── tsconfig.json            # Configuración TypeScript
│   └── index.html               # Plantilla HTML
```

Se instalan las herramientas de software requeridas y necesarias para el despliegue de la interfaz gráfica de conexión y acceso estos son los paquetes de la plataforma Vite React con TypeScript usando el comando “npm create vite@latest . -- --template react-ts”, lo que crea un proyecto de desarrollo, el cual genera una plantilla de archivos necesarios y requeridos que se modifican de acuerdo con el uso y necesidades de despliegue de la interfaz.

```
C:\Users\Administrador\Documents\truffle-project>npm create vite@latest my-dapp -- --template react-ts

> npx
> create-vite my-dapp --template react-ts

|
o  Scaffolding project in C:\Users\Administrador\Documents\truffle-project\my-dapp...
|
—  Done. Now run:
```

Se crea el archivo “blockchain.ts”, el cual de igual manera recupera información de la cadena de bloques generada y usa la librería “web3” para validar que las direcciones de cuentas que se ingresan en la interfaz web sean las mismas que se desplegaron en la plataforma Ganache, enlazándose mediante la dirección de contrato y la clave privada de administrador (generalmente es la clave privada de la primera cuenta del entorno Ganache), generadas al migrar el archivo “AccessControl.sol”, además de la dirección y puerto del servidor Ganache.

```
import Web3 from 'web3';

const CONTRACT_ABI = [
    {
      "inputs": [],
      "stateMutability": "nonpayable",
      "type": "constructor"
```

```
  },
  {
    "anonymous": false,
    "inputs": [
      {
        "indexed": true,
        "internalType": "address",
        "name": "account",
        "type": "address"
      },
      {
        "indexed": false,
        "internalType": "uint256",
        "name": "timestamp",
        "type": "uint256"
      }
    ],
    "name": "AccessGranted",
    "type": "event"
  },
  {
    "anonymous": false,
    "inputs": [
      {
        "indexed": true,
        "internalType": "address",
        "name": "account",
        "type": "address"
      }
    ],
    "name": "AccountAllowed",
    "type": "event"
  },
  {
    "anonymous": false,
    "inputs": [
      {
        "indexed": true,
        "internalType": "address",
        "name": "account",
        "type": "address"
      }
    ],
    "name": "AccountRevoked",
    "type": "event"
```

```
  },
  {
    "inputs": [
      {
        "internalType": "address",
        "name": "",
        "type": "address"
      }
    ],
    "name": "accessTimestamps",
    "outputs": [
      {
        "internalType": "uint256",
        "name": "",
        "type": "uint256"
      }
    ],
    "stateMutability": "view",
    "type": "function",
    "constant": true
  },
  {
    "inputs": [],
    "name": "owner",
    "outputs": [
      {
        "internalType": "address",
        "name": "",
        "type": "address"
      }
    ],
    "stateMutability": "view",
    "type": "function",
    "constant": true
  },
  {
    "inputs": [
      {
        "internalType": "address",
        "name": "_account",
        "type": "address"
      }
    ],
    "name": "addAllowedAccount",
    "outputs": [],
```

```
    "stateMutability": "nonpayable",
    "type": "function"
  },
  {
    "inputs": [
      {
        "internalType": "address",
        "name": "_account",
        "type": "address"
      }
    ],
    "name": "removeAllowedAccount",
    "outputs": [],
    "stateMutability": "nonpayable",
    "type": "function"
  },
  {
    "inputs": [
      {
        "internalType": "address",
        "name": "_account",
        "type": "address"
      }
    ],
    "name": "isAccountAllowed",
    "outputs": [
      {
        "internalType": "bool",
        "name": "",
        "type": "bool"
      }
    ],
    "stateMutability": "view",
    "type": "function",
    "constant": true
  },
  {
    "inputs": [
      {
        "internalType": "address",
        "name": "_account",
        "type": "address"
      }
    ],
    "name": "grantAccess",
```

```
      "outputs": [],
      "stateMutability": "nonpayable",
      "type": "function"
    },
    {
      "inputs": [
        {
          "internalType": "address[]",
          "name": "_accounts",
          "type": "address[]"
        }
      ],
      "name": "batchAddAccounts",
      "outputs": [],
      "stateMutability": "nonpayable",
      "type": "function"
    }
];

export class BlockchainService {
  private web3: Web3;
  private contractAddress: string = "0x516bBF3bDf736D996E42fED1dbaA2B32477274ec";
//Dirección de contrato generada al compilar en Truffle
  private contract: any;

  constructor() {
    this.web3 = new Web3('http://192.168.88.248:7545');
    this.contract = new this.web3.eth.Contract(CONTRACT_ABI,
this.contractAddress);
  }

  async validateAccount(accountAddress: string): Promise<boolean> {
    try {
      const isValid = await
this.contract.methods.isAccountAllowed(accountAddress).call();
      if (isValid) {
        await this.contract.methods.grantAccess(accountAddress).send({
          from: accountAddress,
          gas: 300000
        });
      }
      return isValid;
    } catch (error) {
      console.error('Validation error:', error);
      return false;
```

```
    }
  }

  async checkConnection(): Promise<boolean> {
    try {
      await this.web3.eth.getBlockNumber();
      return true;
    } catch {
      return false;
    }
  }

  async getAccounts(): Promise<string[]> {
    return await this.web3.eth.getAccounts();
  }
}

export const blockchainService = new BlockchainService();
```

Uno de los elementos de la pantalla principal de la interfaz web, se configura en el archivo "Login.tsx", el cual con el uso de la librería "ethers" permite el ingreso y validación de las direcciones de cuenta de la cadena de bloques Ganache y muestra en pantalla las notificaciones de errores, ingreso de datos incorrectos o de la validación realizada.

```
import React, { useState, useEffect } from 'react';
import { blockchainService } from '../services/blockchain';

const Login: React.FC = () => {
  const [accountAddress, setAccountAddress] = useState('');
  const [isValid, setIsValid] = useState<boolean | null>(null);
  const [loading, setLoading] = useState(false);
  const [connected, setConnected] = useState(false);
  const [availableAccounts, setAvailableAccounts] = useState<string[]>([]);

  useEffect(() => {
    checkBlockchainConnection();
  }, []);

  const checkBlockchainConnection = async () => {
    const isConnected = await blockchainService.checkConnection();
    setConnected(isConnected);

    if (isConnected) {
      const accounts = await blockchainService.getAccounts();
      setAvailableAccounts(accounts);
```

```
    }
  };

  const handleSubmit = async (e: React.FormEvent) => {
    e.preventDefault();
    setLoading(true);

    try {
      const valid = await blockchainService.validateAccount(accountAddress);
      setIsValid(valid);

      if (valid) {
        // Redirigir a Google después de 2 segundos
        setTimeout(() => {
          window.location.href = 'https://www.google.com';
        }, 2000);
      }
    } catch (error) {
      console.error('Error:', error);
      setIsValid(false);
    } finally {
      setLoading(false);
    }
  };

  return (
    <div className="min-h-screen bg-gray-100 flex flex-col justify-center py-12
sm:px-6 lg:px-8">
      <div className="sm:mx-auto sm:w-full sm:max-w-md">
        <h2 className="mt-6 text-center text-3xl font-extrabold text-gray-900">
          Autenticación WiFi "FICA_Chain"


          <center><img src="public/utnchain.gif" width={220}/></center>

        </h2>
        <p className="mt-2 text-center text-sm text-gray-600">
        "Mecanismo de autenticación utilizando Cadena de Bloques en un entorno
inalámbrico de pruebas en la Facultad de Ingeniería en Ciencias Aplicadas"
        </p>
      </div>

      <div className="mt-8 sm:mx-auto sm:w-full sm:max-w-md">
        <div className="bg-white py-8 px-4 shadow sm:rounded-lg sm:px-10">
```

```
          <div className={`mb-4 p-3 rounded ${connected ? 'bg-green-50 text-
green-800' : 'bg-red-50 text-red-800'}`}>
            Entorno de Cadena de Bloques Ganache: {connected ? 'Conectado' :
'Desconectado'}
          </div>
          <h3 className="mt-6 text-center text-3xl font-extrabold text-gray-900">
          Ingresar Dirección de Cuenta Ganache
          </h3>

          <form className="space-y-6" onSubmit={handleSubmit}>
            <div>

              <div className="mt-1">
                <input
                  id="account"
                  name="account"
                  type="text"
                  required
                  value={accountAddress}
                  onChange={(e) => setAccountAddress(e.target.value)}
                  placeholder="0x..."
                  className="appearance-none block w-full px-3 py-2 border
border-gray-300 rounded-md shadow-sm placeholder-gray-400 focus:outline-none
focus:ring-blue-500 focus:border-blue-500 sm:text-sm"
                />
              </div>

              {<div className="mt-2">
                <p className="text-xs text-gray-500">
                  Direcciones de Cuenta disponibles desde Ganache:
                </p>
                <select
                  className="mt-1 block w-full pl-3 pr-10 py-2 text-base border-
gray-300 focus:outline-none focus:ring-blue-500 focus:border-blue-500 sm:text-sm
rounded-md"
                  onChange={(e) => setAccountAddress(e.target.value)}
                  value={accountAddress}
                >
                  <option value="">Selecione una Dirección</option>
                  {availableAccounts.map((account, index) => (
                    <option key={index} value={account}>
                      Account {index + 1}: {account.slice(0, 100)}...
                    </option>

                  ))}
```

```
              </select>
            </div>}

          </div>

          <div>
          <p></p>
            <button
              type="submit"
              disabled={loading || !connected}
              className="w-full flex justify-center py-2 px-4 border border-
transparent rounded-md shadow-sm text-sm font-medium text-white bg-blue-600
hover:bg-blue-700 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-
blue-500 disabled:opacity-50 disabled:cursor-not-allowed"
            >
              {loading ? 'Validando...' : 'Validar & Conectar'}
            </button>
          </div>
        </form>

        {isValid === true && (
          <div className="mt-4 p-3 bg-green-50 border border-green-200 rounded-
md">
            <p className="text-green-800 text-sm">
              ✅ Cuenta validada exitosamente!!! Continue Navegando...
            </p>
          </div>
        )}

        {isValid === false && (
          <div className="mt-4 p-3 bg-red-50 border border-red-200 rounded-md">
            <p className="text-red-800 text-sm">
              ❌ Cuenta no autorizada. Intente con otra Dirección.
            </p>
          </div>
        )}

        <div className="mt-6">
          <div className="relative">
            <div className="absolute inset-0 flex items-center">
              <div className="w-full border-t border-gray-300" />
            </div>
            <div className="relative flex justify-center text-sm">
              <span className="px-2 bg-white text-gray-500">
```

```
                </span>
              </div>
            </div>

            <div className="mt-4 text-xs text-gray-600">
              <p></p>
              <p></p>
              <p></p>
            </div>
          </div>
        </div>
      </div>
    </div>
  );
};

export default Login;
```

Se modifica el contenido del archivo “main.tsx”.

```
import React from 'react'
import ReactDOM from 'react-dom/client'
import App from './App'
import './index.css'

ReactDOM.createRoot(document.getElementById('root')!).render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
)
```

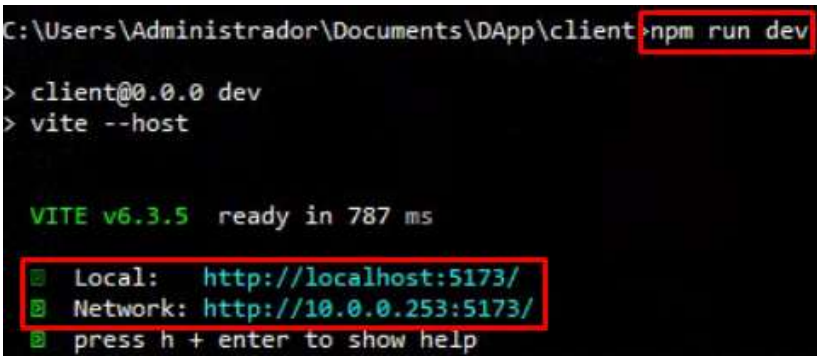
Se modifica el contenido del archivo “index.html”.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" type="image/svg+xml" href="/utn.png" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Autenticación WiFi FICA_Chain</title>
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/main.tsx"></script>
  </body>
</html>
```

Se modifica el contenido del archivo “vite.config.ts”.

```
import { defineConfig } from 'vite'
import react from '@vitejs/plugin-react'

export default defineConfig({
  plugins: [react()],
  server: {
    host: '192.168.88.248',
    port: 5173,
    proxy: {
      '/api': {
        target: 'http://192.168.88.248:3001',
        changeOrigin: true,
      }
    }
  }
})
```

La vista previa de como se muestra la interfaz en el navegador web, se ejecuta en la consola de la carpeta donde se creó el proyecto Vite React, el comando “npm run dev”, despliega la interfaz web y muestra las direcciones de acceso que se deben colocar en la barra de búsqueda del navegador web para acceder a la interfaz.



Anexo 4.10: Configuraciones de MikroTik AP

```
# ========================================
# CONFIGURACIÓN COMPLETA FICA_CHAIN - RouterOS 6.49.19
# ========================================

# 1. CONFIGURACIÓN INICIAL
/interface bridge add name=bridge-local comment="FICA_Chain Bridge"
/interface bridge port add bridge=bridge-local interface=wlan1
/interface bridge port add bridge=bridge-local interface=ether2
/interface bridge port add bridge=bridge-local interface=ether3
/interface bridge port add bridge=bridge-local interface=ether4
/interface bridge port add bridge=bridge-local interface=ether5

# 2. CONFIGURACIÓN IP
/ip address add address=192.168.88.1/24 interface=bridge-local
network=192.168.88.0

# 3. CONFIGURACIÓN WIFI
/interface wireless set wlan1 disabled=no ssid="FICA_Chain" mode=ap-
bridge band=2ghz-b/g/n security-profile=none

# 4. CONFIGURACIÓN HOTSPOT
/ip hotspot add name=hotspot1 interface=bridge-local address-
pool=static-only disabled=no
/ip hotspot network add address=192.168.88.0/24 gateway=192.168.88.1
dns-server=8.8.8.8
/ip hotspot profile set default hotspot-address=192.168.88.1 html-
directory=hotspot login-by=http-chap,mac-cookie

# 5. CREAR ARCHIVOS HTML
/file mkdir hotspot

# login.html
/file print file=login.html contents="<!DOCTYPE html>
<html>
<head>
<title>FICA_Chain</title>
<script>
function goToReact() {
    var url = 'http://192.168.88.248:5173?' +
        'mac=' + encodeURIComponent('${mac}') +
        '&ip=' + encodeURIComponent('${ip}') +
        '&dst=https://www.google.com';
    window.location.href = url;
```

```
}
</script>
<style>body{text-align:center;padding:50px;}</style>
</head>
<body>
<h1>FICA_Chain WiFi</h1>
<p>MAC: ${mac}</p>
<p>IP: ${ip}</p>
<button onclick='goToReact()'
style='padding:15px;background:green;color:white;'>
    🔗 Validar Blockchain
</button>
</body>
</html>
"

# alogin.html
/file print file=alogin.html contents="<!DOCTYPE html>
<html>
<head><meta http-equiv='refresh'
content='2;url=https://www.google.com'></head>
<body style='text-align:center;padding:50px;'>
<h1>✅ Acceso Concedido</h1>
<p>Redirigiendo a Google...</p>
</body>
</html>
"

# 6. SCRIPTS
/system script add name=autorizar-fica source={
    :local clientMAC \"\$1\"
    :local clientIP \"\$2\"
    :local username \"fica_\$clientMAC\"

    /ip hotspot user add name=\$username password=\$clientMAC
disabled=no
    /ip hotspot active add mac-address=\$clientMAC user=\$username
address=\$clientIP server=hotspot1

    :log info \"✅ Autorizado: \$clientIP\"
}

# 7. FIREWALL
/ip firewall address-list add list=autorizados-fica
address=192.168.88.1
```

```
/ip firewall address-list add list=autorizados-fica
address=192.168.88.248

/ip firewall filter add chain=forward dst-address=192.168.88.248
action=accept comment="Portal React"
/ip firewall filter add chain=forward src-address-list=autorizados-fica
action=accept comment="Autorizados FICA"
/ip firewall filter add chain=forward src-address=192.168.88.0/24
action=drop comment="Bloquear no auth"

/ip firewall nat add chain=srcnat out-interface=ether1
action=masquerade comment="NAT internet"

# 8. HABILITAR SERVICIOS
/ip service set www disabled=no

# 9. MENSAJE FINAL
:log info \"✅ Configuración FICA completada\"
:put \"====================================\"
:put \"FICA_CHAIN CONFIGURADO EXITOSAMENTE\"
:put \"WiFi: FICA_Chain (abierta)\"
:put \"Portal: 192.168.88.248:5173\"
:put \"Script: /system script run autorizar-fica\"
:put \"====================================\"
```

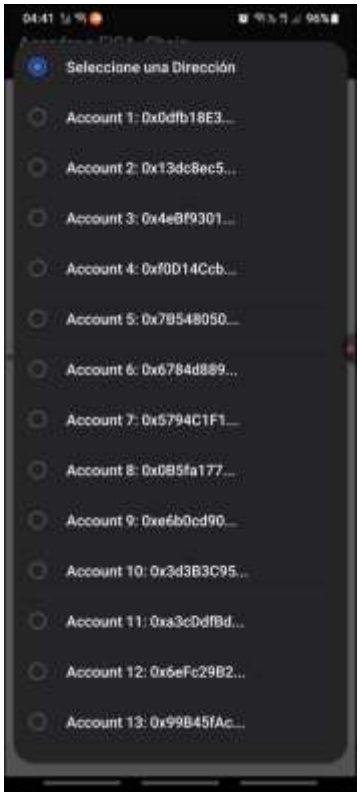
Anexo 5: Manual de Conexión al Mecanismo de Autenticación Propuesto

Conexión con dispositivo (Android, IOS o Windows), seleccionar la red identificada como "FICA_Chain"

Dependiendo del sistema operativo Android o IOS, la configuración del punto de acceso inalámbrico, redirigir al portal de conexión en el dominio "fica.chain" para Windows si no se redirige se ingres manualmente en una pestaña del navegador la IP 192.168.88.1.

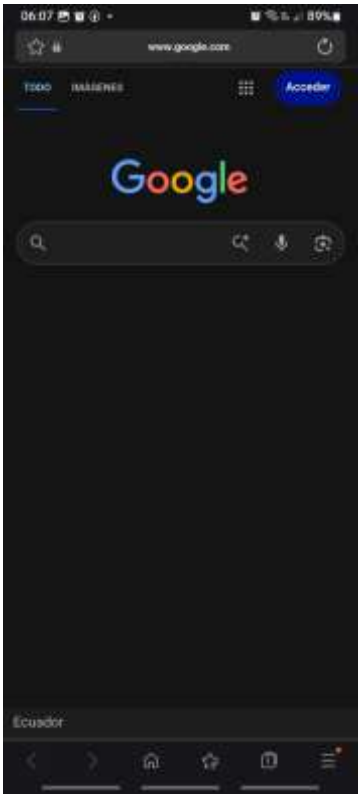
Al dar clic en “Autenticar con Cadena de Bloques” se muestra la interfaz de validación.

Seleccionar o ingresar una dirección de cadena de bloques.



Clic en “Validar y Conectar” y se muestra que la dirección de cuenta es válida.

El navegador se redirige para continuar con la navegación.



Anexo 6: Prueba de Concepto

Anexo 6.1: Contrato Inteligente de Prueba de Concepto

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract AuthBenchmark {

    address public owner;

    mapping(address => bool) public allowedAccounts;
    mapping(address => uint256) public lastAuthTime;

    // ====== Metricas ======
    uint256 public totalAuthentications;
    uint256 public currentRound;
    uint256 public roundStartTime;

    struct RoundStats {
        uint256 usersTarget;
        uint256 usersCompleted;
        uint256 startTime;
        uint256 endTime;
    }

    mapping(uint256 => RoundStats) public rounds;

    // ====== EVENTOS (para análisis externo) ======
    event RoundStarted(uint256 indexed roundId, uint256 usersTarget);
    event UserAuthenticated(address indexed user, uint256 timestamp);
    event RoundFinished(uint256 indexed roundId, uint256 duration);

    modifier onlyOwner() {
        require(msg.sender == owner, "Not owner");
        _;
    }

    constructor() {
        owner = msg.sender;
        allowedAccounts[msg.sender] = true;
    }

    // ====== Preparar grupo de usuarios ======
    function batchAllow(address[] calldata accounts) external onlyOwner {
```

```
        for (uint i = 0; i < accounts.length; i++) {
            allowedAccounts[accounts[i]] = true;
        }
    }

    // ====== Iniciar ronda de prueba ======
    function startRound(uint256 usersTarget) external onlyOwner {
        currentRound++;

        rounds[currentRound] = RoundStats({
            usersTarget: usersTarget,
            usersCompleted: 0,
            startTime: block.timestamp,
            endTime: 0
        });

        roundStartTime = block.timestamp;

        emit RoundStarted(currentRound, usersTarget);
    }

    // ====== Función usada por cada cliente ======
    function benchmarkAuthenticate() external {
        require(allowedAccounts[msg.sender], "Not allowed");

        lastAuthTime[msg.sender] = block.timestamp;

        rounds[currentRound].usersCompleted++;
        totalAuthentications++;

        emit UserAuthenticated(msg.sender, block.timestamp);

        // Cierre automático de ronda
        if (rounds[currentRound].usersCompleted ==
rounds[currentRound].usersTarget) {
            rounds[currentRound].endTime = block.timestamp;

            uint256 duration = rounds[currentRound].endTime -
rounds[currentRound].startTime;

            emit RoundFinished(currentRound, duration);
        }
    }

    function getRoundDuration(uint256 roundId) external view returns (uint256) {
```

```
        return rounds[roundId].endTime - rounds[roundId].startTime;
    }
}
```

Anexo 6.2: Script de Prueba de Concepto

```
const fs = require("fs");
const path = require("path");

const Access = artifacts.require("AuthBenchmark");

const triggerPath = path.join(__dirname,"../metrics/trigger/round.json");

// Intensidad configurable
const AUTH_REPEATS = 4;        // veces que cada usuario re-autentica
const STATE_READS  = 6;        // consultas por usuario
const BURST_CALLS  = 5;        // ráfagas paralelas
const THINK_TIME   = 40;       // ms entre acciones

async function simulateUser(instance, user) {

    for (let i = 0; i < AUTH_REPEATS; i++) {

        // Transacción real (consume CPU + EVM)
        await instance.benchmarkAuthenticate({from: user});

        // Lecturas RPC (simulan validaciones del portal)
        for (let j = 0; j < STATE_READS; j++) {
            await instance.allowedAccounts.call(user);
        }

        // Ráfaga paralela tipo tráfico web
        let burst = [];
        for (let k = 0; k < BURST_CALLS; k++) {
            burst.push(instance.lastAuthTime.call(user));
        }
        await Promise.all(burst);

        await new Promise(r => setTimeout(r, THINK_TIME));
    }
}

module.exports = async function(callback) {
```

```
const instance = await Access.deployed();
const accounts = await web3.eth.getAccounts();

const testBlocks = [1,5,10,15,25,50,75,100];

let round = 0;

for (let size of testBlocks) {

    round++;

    console.log(`\n============================`);
    console.log(`Grupo de: ${size} Usuario(s)`);
    console.log(`============================`);

    const users = accounts.slice(1, size+1);

    await instance.batchAllow(users);
    await instance.startRound(size);

    const start = Date.now();

    // Todos los usuarios ejecutan actividad real simultánea
    await Promise.all(
        users.map(u => simulateUser(instance, u))
    );

    const end = Date.now();
    const duration = end - start;

    console.log(`Duración total de grupo: ${duration} ms`);

    // Notificar al monitor CIM (Métricas)
    const payload = {
        round: round,
        users: size,
        durationMs: duration
    };

    fs.writeFileSync(triggerPath, JSON.stringify(payload));

    while (fs.existsSync(triggerPath)) {
        await new Promise(r => setTimeout(r,100));
    }
```

```
    }

    callback();
};
```

Anexo 6.3: Script de Captura de Métricas

```
$triggerPath = "..\trigger\round.json"
$outputDir   = "..\output"

Write-Host "=== Monitor de Rendimiento Iniciado ==="

while ($true) {

    if (Test-Path $triggerPath) {

        $data = Get-Content $triggerPath | ConvertFrom-Json

        $roundId   = $data.round
        $users     = $data.users
        $duration  = $data.durationMs

        Write-Host "Capturando métricas -> Ronda $roundId ($users usuarios)
durante $duration ms"

        $endTime = (Get-Date).AddMilliseconds($duration)

        $samples = @()

        while ((Get-Date) -lt $endTime) {

            $cpu = Get-CimInstance Win32_Processor | Measure-Object -Property
LoadPercentage -Average
            $os  = Get-CimInstance Win32_OperatingSystem
            $memFree = [math]::Round($os.FreePhysicalMemory / 1024, 2)
            $memTotal = [math]::Round($os.TotalVisibleMemorySize / 1024, 2)
            $memUsed = $memTotal - $memFree

            $disk = Get-CimInstance Win32_PerfFormattedData_PerfDisk_LogicalDisk
|
                    Where-Object {$_.Name -eq "_Total"}
```

```
            $net = Get-CimInstance Win32_PerfFormattedData_Tcpip_NetworkInterface
|
                   Measure-Object -Property BytesTotalPersec -Sum

            $sample = [PSCustomObject]@{
                Timestamp   = Get-Date -Format "HH:mm:ss.fff"
                Round       = $roundId
                Users       = $users
                CPU_Percent = $cpu.Average
                RAM_Used_MB = $memUsed
                RAM_Free_MB = $memFree
                Disk_BytesSec = $disk.DiskBytesPersec
                Net_BytesSec  = $net.Sum
            }

            $samples += $sample

            Start-Sleep -Milliseconds 200
        }

        $file = "$outputDir\round_${roundId}_users_${users}.csv"
        $samples | Export-Csv $file -NoTypeInformation

        Write-Host "Ronda $roundId guardada en $file"

        Remove-Item $triggerPath
    }

    Start-Sleep -Milliseconds 100
}
```

Anexo 6.4: Muestras por grupo de Usuarios

MUESTRAS DE CONEXIÓN DE 1 USUARIO					
# Muestra	**Tiempo (ms)**	**Uso CPU (%)**	**Uso RAM (MB)**	**Uso Disco (Bps)**	**Uso Red (Bps)**
1	742	9	4031,28	0	243
2	608	37	4067,28	0	0
3	806	11	4055,68	504423	0
4	712	19	4158,88	0	0
5	651	22	4162,53	19463	0
6	711	7	4140,23	0	0
7	639	1	4169,21	14312	1077
8	840	54	4160,07	349925	0
9	657	5	4185,87	0	0
10	870	6	4180,13	0	251
Media	**723,6**	**17,1**	**4131,116**	**88812,3**	**157,1**
Desviación Estándar	**89,83**	**16,73**	**57,02**	**182146,91**	**339,15**

MUESTRAS DE CONEXIÓN DE 5 USUARIOS					
# Muestra	**Tiempo (ms)**	**Uso CPU (%)**	**Uso RAM (MB)**	**Uso Disco (Bps)**	**Uso Red (Bps)**
1	2961,00	12,00	4116,19	0,00	0,00
2	3968,00	10,00	4146,50	0,00	0,00
3	3170,00	16,50	4124,83	0,00	684,00
4	2811,00	20,50	4218,90	0,00	0,00
5	3638,00	41,33	4231,42	43600,50	0,00
6	2667,00	28,50	4212,68	0,00	0,00
7	2977,00	12,50	4216,22	0,00	0,00
8	2923,00	13,00	4215,60	64720,00	225,00
9	3073,00	17,00	4240,30	0,00	0,00
10	2987,00	15,00	4216,79	0,00	0,00
Media	**3117,5**	**18,63**	**4193,941**	**10832,05**	**90,9**
Desviación Estándar	**393,70**	**9,57**	**46,05**	**23372,23**	**220,06**

MUESTRAS DE CONEXIÓN DE 10 USUARIOS					
# Muestra	**Tiempo (ms)**	**Uso CPU (%)**	**Uso RAM (MB)**	**Uso Disco (Bps)**	**Uso Red (Bps)**
1	5162	14,00	4167,41	0,00	0,00
2	5147	51,33	4212,51	23169,33	0,00
3	8090	16,80	4197,70	114162,80	549,20
4	5565	16,33	4270,06	157368,00	0,00
5	8174	12,40	4311,83	116564,60	0,00
6	8586	22,60	4298,97	0,00	80,40
7	5837	52,67	4296,30	0,00	83,33
8	9641	11,33	4304,97	102857,17	270,83
9	5585	15,33	4297,17	0,00	0,00
10	9104	11,80	4324,60	0,00	230,20
Media	**7089,10**	**22,46**	**4268,15**	**51412,19**	**121,40**
Desviación Estándar	**1783,57**	**15,90**	**55,02**	**63306,50**	**180,74**

MUESTRAS DE CONEXIÓN DE 15 USUARIOS					
# Muestra	**Tiempo (ms)**	**Uso CPU (%)**	**Uso RAM (MB)**	**Uso Disco (Bps)**	**Uso Red (Bps)**
1	7283	16,00	4285,23	0,00	0,00
2	8207	16,80	4338,06	0,00	49,00
3	6822	12,00	4299,65	0,00	0,00
4	10808	19,33	4416,89	2291552,67	187,17
5	7127	22,75	4405,13	85055,75	0,00
6	7084	11,00	4402,59	0,00	0,00
7	7887	11,40	4403,45	3058,00	0,00
8	6611	11,75	4397,62	6841,50	57,75
9	8069	37,75	4422,92	13637,50	0,00
10	7254	11,20	4424,37	58357,00	0,00
Media	**7715,20**	**17,00**	**4379,59**	**245850,24**	**29,39**
Desviación Estándar	**1207,52**	**8,32**	**52,04**	**719388,04**	**59,75**

MUESTRAS DE CONEXIÓN DE 25 USUARIOS					
# Muestra	**Tiempo (ms)**	**Uso CPU (%)**	**Uso RAM (MB)**	**Uso Disco (Bps)**	**Uso Red (Bps)**
1	12938	23,88	4201,51	0,00	55,25
2	12956	21,89	4401,53	0,00	24,44
3	13109	21,00	4178,17	131137,57	64,43
4	11684	26,14	4259,57	4532,00	35,00
5	13634	30,25	4262,94	44846,50	88,50
6	13399	12,75	4502,16	56381,25	319,75
7	12623	13,86	4497,07	218062,43	102,86
8	12748	12,71	4275,30	37089,71	0,00
9	13135	21,13	4298,38	101614,00	128,38
10	13509	16,38	4277,80	0,00	323,00
Media	**12973,50**	**20,00**	**4315,44**	**59366,35**	**114,16**
Desviación Estándar	**556,44**	**5,96**	**113,67**	**71814,81**	**115,55**

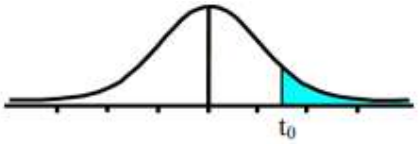
MUESTRAS DE CONEXIÓN DE 50 USUARIOS					
# Muestra	**Tiempo (ms)**	**Uso CPU (%)**	**Uso RAM (MB)**	**Uso Disco (Bps)**	**Uso Red (Bps)**
1	27225	22,00	4169,24	17652,67	16,27
2	26653	23,29	4515,36	55448,14	238,00
3	27645	24,47	4142,27	503361,80	30,27
4	27961	16,50	4267,53	10188,56	187,44
5	35008	20,00	4226,96	62947,32	50,63
6	30631	13,00	4359,19	133407,00	46,24
7	25511	12,00	4560,93	2756807,93	162,71
8	34950	23,21	4222,35	2673,89	163,47
9	33796	14,53	4278,70	74654,26	1495,37
10	27370	28,36	4291,62	2378719,86	65,71
Media	**29675,00**	**19,73**	**4303,42**	**599586,14**	**245,61**
Desviación Estándar	**3636,37**	**5,47**	**138,55**	**1051320,12**	**445,73**

MUESTRAS DE CONEXIÓN DE 75 USUARIOS					
# Muestra	**Tiempo (ms)**	**Uso CPU (%)**	**Uso RAM (MB)**	**Uso Disco (Bps)**	**Uso Red (Bps)**
1	44282	12,33	4160,19	651846,83	52,00
2	40367	11,70	4363,98	16589,39	256,04
3	59540	16,12	4114,83	71260,79	530,55
4	41083	13,27	4267,27	30367,09	20,18
5	51516	19,11	4299,76	103498,61	184,75
6	41613	10,74	4235,24	14082,22	100,52
7	45320	11,92	4229,83	26760,75	74,79
8	51009	14,43	4235,09	10620,96	850,29
9	48519	16,73	4230,04	36187,69	828,50
10	50393	11,64	4234,67	28624,04	44,61
Media	**47364,20**	**13,80**	**4237,09**	**98983,84**	**294,22**
Desviación Estándar	**6004,15**	**2,73**	**68,24**	**196375,45**	**324,16**

MUESTRAS DE CONEXIÓN DE 100 USUARIOS					
# Muestra	**Tiempo (ms)**	**Uso CPU (%)**	**Uso RAM (MB)**	**Uso Disco (Bps)**	**Uso Red (Bps)**
1	72301	10,90	4173,28	4813,51	91,79
2	55633	10,52	4156,46	2358,16	198,03
3	67685	15,86	4115,92	28258,38	89,22
4	52987	9,76	4386,49	533193,76	347,10
5	74159	16,02	4204,68	15743,02	1313,39
6	63246	22,69	4251,47	39857,97	96,71
7	68938	15,70	4225,55	8794,89	5882,59
8	66521	18,06	4284,37	1029613,75	152,89
9	68026	14,51	4242,16	11121,95	44,95
10	77203	16,51	4240,33	3692069,05	2525,51
Media	**66669,90**	**15,05**	**4228,07**	**536582,44**	**1074,22**
Desviación Estándar	**7656,05**	**3,91**	**74,85**	**1159474,57**	**1867,25**

Anexo 6.5: Tabla t-Student

Tabla t-Student



Grados de libertad	0.25	0.1	0.05	0.025	0.01	0.005
1	1.0000	3.0777	6.3137	12.7062	31.8210	63.6559
2	0.8165	1.8856	2.9200	4.3027	6.9645	9.9250
3	0.7649	1.6377	2.3534	3.1824	4.5407	5.8408
4	0.7407	1.5332	2.1318	2.7765	3.7469	4.6041
5	0.7267	1.4759	2.0150	2.5706	3.3649	4.0321
6	0.7176	1.4398	1.9432	2.4469	3.1427	3.7074
7	0.7111	1.4149	1.8946	2.3646	2.9979	3.4995
8	0.7064	1.3968	1.8595	2.3060	2.8965	3.3554
9	0.7027	1.3830	1.8331	2.2622	2.8214	3.2498
10	0.6998	1.3722	1.8125	2.2281	2.7638	3.1693
11	0.6974	1.3634	1.7959	2.2010	2.7181	3.1058
12	0.6955	1.3562	1.7823	2.1788	2.6810	3.0545
13	0.6938	1.3502	1.7709	2.1604	2.6503	3.0123
14	0.6924	1.3450	1.7613	2.1448	2.6245	2.9768
15	0.6912	1.3406	1.7531	2.1315	2.6025	2.9467
16	0.6901	1.3368	1.7459	2.1199	2.5835	2.9208
17	0.6892	1.3334	1.7396	2.1098	2.5669	2.8982
18	0.6884	1.3304	1.7341	2.1009	2.5524	2.8784
19	0.6876	1.3277	1.7291	2.0930	2.5395	2.8609
20	0.6870	1.3253	1.7247	2.0860	2.5280	2.8453
21	0.6864	1.3232	1.7207	2.0796	2.5176	2.8314
22	0.6858	1.3212	1.7171	2.0739	2.5083	2.8188
23	0.6853	1.3195	1.7139	2.0687	2.4999	2.8073
24	0.6848	1.3178	1.7109	2.0639	2.4922	2.7970
25	0.6844	1.3163	1.7081	2.0595	2.4851	2.7874
26	0.6840	1.3150	1.7056	2.0555	2.4786	2.7787
27	0.6837	1.3137	1.7033	2.0518	2.4727	2.7707
28	0.6834	1.3125	1.7011	2.0484	2.4671	2.7633
29	0.6830	1.3114	1.6991	2.0452	2.4620	2.7564
30	0.6828	1.3104	1.6973	2.0423	2.4573	2.7500
31	0.6825	1.3095	1.6955	2.0395	2.4528	2.7440
32	0.6822	1.3086	1.6939	2.0369	2.4487	2.7385
33	0.6820	1.3077	1.6924	2.0345	2.4448	2.7333
34	0.6818	1.3070	1.6909	2.0322	2.4411	2.7284
35	0.6816	1.3062	1.6896	2.0301	2.4377	2.7238
36	0.6814	1.3055	1.6883	2.0281	2.4345	2.7195
37	0.6812	1.3049	1.6871	2.0262	2.4314	2.7154
38	0.6810	1.3042	1.6860	2.0244	2.4286	2.7116
39	0.6808	1.3036	1.6849	2.0227	2.4258	2.7079
40	0.6807	1.3031	1.6839	2.0211	2.4233	2.7045
41	0.6805	1.3025	1.6829	2.0195	2.4208	2.7012
42	0.6804	1.3020	1.6820	2.0181	2.4185	2.6981
43	0.6802	1.3016	1.6811	2.0167	2.4163	2.6951
44	0.6801	1.3011	1.6802	2.0154	2.4141	2.6923
45	0.6800	1.3007	1.6794	2.0141	2.4121	2.6896
46	0.6799	1.3002	1.6787	2.0129	2.4102	2.6870
47	0.6797	1.2998	1.6779	2.0117	2.4083	2.6846
48	0.6796	1.2994	1.6772	2.0106	2.4066	2.6822
49	0.6795	1.2991	1.6766	2.0096	2.4049	2.6800

50	0.6794	1.2987	1.6759	2.0086	2.4033	2.6778
51	0.6793	1.2984	1.6753	2.0076	2.4017	2.6757
52	0.6792	1.2980	1.6747	2.0066	2.4002	2.6737
53	0.6791	1.2977	1.6741	2.0057	2.3988	2.6718
54	0.6791	1.2974	1.6736	2.0049	2.3974	2.6700
55	0.6790	1.2971	1.6730	2.0040	2.3961	2.6682
56	0.6789	1.2969	1.6725	2.0032	2.3948	2.6665
57	0.6788	1.2966	1.6720	2.0025	2.3936	2.6649
58	0.6787	1.2963	1.6716	2.0017	2.3924	2.6633
59	0.6787	1.2961	1.6711	2.0010	2.3912	2.6618
60	0.6786	1.2958	1.6706	2.0003	2.3901	2.6603
61	0.6785	1.2956	1.6702	1.9996	2.3890	2.6589
62	0.6785	1.2954	1.6698	1.9990	2.3880	2.6575
63	0.6784	1.2951	1.6694	1.9983	2.3870	2.6561
64	0.6783	1.2949	1.6690	1.9977	2.3860	2.6549
65	0.6783	1.2947	1.6686	1.9971	2.3851	2.6536
66	0.6782	1.2945	1.6683	1.9966	2.3842	2.6524
67	0.6782	1.2943	1.6679	1.9960	2.3833	2.6512
68	0.6781	1.2941	1.6676	1.9955	2.3824	2.6501
69	0.6781	1.2939	1.6672	1.9949	2.3816	2.6490
70	0.6780	1.2938	1.6669	1.9944	2.3808	2.6479
71	0.6780	1.2936	1.6666	1.9939	2.3800	2.6469
72	0.6779	1.2934	1.6663	1.9935	2.3793	2.6458
73	0.6779	1.2933	1.6660	1.9930	2.3785	2.6449
74	0.6778	1.2931	1.6657	1.9925	2.3778	2.6439
75	0.6778	1.2929	1.6654	1.9921	2.3771	2.6430
76	0.6777	1.2928	1.6652	1.9917	2.3764	2.6421
77	0.6777	1.2926	1.6649	1.9913	2.3758	2.6412
78	0.6776	1.2925	1.6646	1.9908	2.3751	2.6403
79	0.6776	1.2924	1.6644	1.9905	2.3745	2.6395
80	0.6776	1.2922	1.6641	1.9901	2.3739	2.6387
81	0.6775	1.2921	1.6639	1.9897	2.3733	2.6379
82	0.6775	1.2920	1.6636	1.9893	2.3727	2.6371
83	0.6775	1.2918	1.6634	1.9890	2.3721	2.6364
84	0.6774	1.2917	1.6632	1.9886	2.3716	2.6356
85	0.6774	1.2916	1.6630	1.9883	2.3710	2.6349
86	0.6774	1.2915	1.6628	1.9879	2.3705	2.6342
87	0.6773	1.2914	1.6626	1.9876	2.3700	2.6335
88	0.6773	1.2912	1.6624	1.9873	2.3695	2.6329
89	0.6773	1.2911	1.6622	1.9870	2.3690	2.6322
90	0.6772	1.2910	1.6620	1.9867	2.3685	2.6316
91	0.6772	1.2909	1.6618	1.9864	2.3680	2.6309
92	0.6772	1.2908	1.6616	1.9861	2.3676	2.6303
93	0.6771	1.2907	1.6614	1.9858	2.3671	2.6297
94	0.6771	1.2906	1.6612	1.9855	2.3667	2.6291
95	0.6771	1.2905	1.6611	1.9852	2.3662	2.6286
96	0.6771	1.2904	1.6609	1.9850	2.3658	2.6280
97	0.6770	1.2903	1.6607	1.9847	2.3654	2.6275
98	0.6770	1.2903	1.6606	1.9845	2.3650	2.6269
99	0.6770	1.2902	1.6604	1.9842	2.3646	2.6264
100	0.6770	1.2901	1.6602	1.9840	2.3642	2.6259
∞	0.6745	1.2816	1.6449	1.9600	2.3263	2.5758