



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y
COMPUTACIÓN

**IMPLEMENTACIÓN DE UN SISTEMA DE VIDEOVIGILANCIA
CON UN MÉTODO DE AUTENTICACIÓN BASADO EN
CADENA DE BLOQUES**

PROYECTO DE TITULACIÓN

Previo a la obtención del Título de:

MAGÍSTER EN TELECOMUNICACIONES

AUTORES:

Karen Stefany Narváez Narváez

Cesar David Alava Romero

GUAYAQUIL – ECUADOR

2023

AGRADECIMIENTOS

Expreso mi agradecimiento a Dios por servir como mi guía, así como a mis educadores, conocidos y compañeros que han impartido conocimientos y ejemplificado buenas prácticas profesionales. Estoy especialmente agradecido con mi esposo y padres, quienes son mi principal apoyo y motivación, permitiéndome alcanzar mis metas y aspiraciones. También extendo mi agradecimiento a mi tutor, quien ha sido fundamental al guiarme desde el inicio de mis estudios de posgrado hasta la concepción de esta investigación. Por último, quisiera agradecer a todos aquellos que me ayudaron durante la realización de este proyecto, sin quienes su apoyo no hubiera sido posible.

Karen Stefany Narváez Narváez

A mis queridos compañeros de estudio, quienes confiaron en mí y compartieron conmigo este camino lleno de desafíos y triunfos. Su amistad y colaboración han enriquecido esta experiencia de manera inigualable. A mis respetados maestros y profesores, cuya pasión por el conocimiento y dedicación a la enseñanza han guiado mi aprendizaje y han ampliado mis horizontes académicos. Su influencia perdurará en mi vida. A la prestigiosa institución, Escuela Superior Politécnica del Litoral, que me brindó la oportunidad de pertenecer a su comunidad académica. Gracias por ofrecer un ambiente enriquecedor y recursos excepcionales que han facilitado mi crecimiento intelectual. En este viaje de autodescubrimiento y aprendizaje, estas personas y entidades han sido los pilares que me han sostenido y motivado a alcanzar mis metas. A todos ellos, mi eterna gratitud.

Cesar David Alava Romero

DEDICATORIA

A Dios, fuente inagotable de sabiduría y fortaleza, por guiar mis pasos y brindarme la inspiración divina en este camino de conocimiento y crecimiento. A mi amado esposo Miguel, tu amor y apoyo incondicional han sido mi motor en cada desafío y logro de este trayecto académico. A mi querido hijo Vladimir, luz de mi vida, tu sonrisa y alegría constante han llenado de energía cada jornada de estudio y dedicación. A mis padres, pilares de amor y ejemplo, gracias por sembrar en mí el valor del esfuerzo y la perseverancia, y por ser mi sustento emocional en todo momento. Esta tesis es el fruto de sus bendiciones, amor y guía, dedicada con profundo cariño y gratitud, como un testimonio de lo que somos capaces de alcanzar juntos

Karen Stefany Narváez Narváez

A Dios, fuente de luz y sabiduría, que ha iluminado mi camino en cada paso de esta travesía académica. A mis padres, cuyo amor inquebrantable y apoyo constante han sido el cimiento de mis logros. Su dedicación y sacrificio han sido mi mayor inspiración. A mi amada esposa, compañera incansable en este viaje, por su paciencia, comprensión y amor incondicional que han sido mi refugio en las horas más desafiantes. A mis hijos, por su apoyo y paciencia en esos momentos donde no pude compartir con ellos por mis estudios.

Cesar David Alava Romero

TRIBUNAL DE EVALUACIÓN

.....
Ing. Ronald Criollo, MSc.

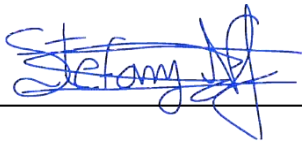
Profesor tutor


.....
PhD. María Antonieta Álvarez Villanueva

Profesor evaluador

DECLARATORIA EXPRESA

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, nos corresponde exclusivamente; y damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"



Ing. Karen Stefany Narváez Narváez



Ing. Cesar David Alava Romero

RESUMEN

El presente trabajo de titulación tiene como objetivo diseñar e implementar un sistema de videovigilancia y un módulo de autenticación para un inicio de sesión seguro mediante la tecnología de cadena de bloques.

La tecnología de Cadena de Bloques se caracteriza por su naturaleza confiable y su sistema descentralizado, que asegura la precisión de la información en los nodos Ethereum. Para facilitar la interacción entre la cadena de bloques y la plataforma de videovigilancia, se desarrolló un contrato inteligente con solidity. La creación de este contrato requirió la utilización del framework truffle para compilación, migración y escritura. Además, para el servidor web se utilizó Node.js y Web3.js como interfaz para interactuar con la cadena de bloques Ethereum en Ganache.

En el desarrollo de la plataforma de Sistema de videovigilancia de código abierto se utilizó la librería reactjs para la creación de las aplicaciones y nodejs para el servidor web. Se levantó un pequeño servidor RTSPtoWeb para convertir video proveniente de la cámara IP mediante el protocolo RTSP a un formato que puede usarse en la aplicación hecha por reactjs, En cuanto al desarrollo del módulo de autenticación se requirió utilizar Ganache como una cadena de bloques Ethereum y metamask como billetera digital en donde se importa las cuenta y la clave privada que nos proporciona Ganache.

Se evaluó la funcionalidad del método de autenticación y la seguridad de la cadena de bloques mediante un escaneo de vulnerabilidad del sistema en general, en donde arrojó solo dos vulnerabilidades de bajo riesgo. Se realizó pruebas de ataque y denegación de servicio en donde se concluyó que el sistema no presenta latencias, caída de servicio ni divulgación de información sensible como las credenciales de los usuarios.

Palabras clave: Inalterable, seguridad, contratos inteligentes, huella.

ABSTRACT

The objective of this degree work is to design and implement a video surveillance system and an authentication module for a secure login using block chain technology.

Blockchain technology is characterized by its reliable nature and its decentralized system, which ensures the accuracy of the information in the Ethereum nodes. To facilitate the interaction between the blockchain and the video surveillance platform, a smart contract was developed with Solidity. Creating this contract required the use of the Truffle framework for compilation, migration, and writing. In addition, for the web server, Node.js and Web3.js were used as the interface to interact with the Ethereum blockchain in Ganache.

In the development of the open-source video surveillance system platform, the reactjs library was used to create the applications and Nodejs for the web server. A small RTSPToWeb server was built to convert video from the IP camera through the RTSP protocol to a format that can be used in the application made by reactjs. Regarding the development of the authentication module, it was required to use Ganache as an Ethereum block chain and metamask as a digital wallet where the account and the private key provided by Ganache are imported.

The functionality of the authentication method and the security of the blockchain were evaluated through a vulnerability scan of the overall system, which returned only two low-risk vulnerabilities. Attack and denial of service tests were carried out where it was concluded that the system does not present latencies, service failure or disclosure of sensitive information such as user credentials.

Keywords: Unalterable, security, smart contracts, hash.

ÍNDICE GENERAL

RESUMEN.....	I
ABSTRACT.....	II
ÍNDICE DE FIGURAS.....	VI
ÍNDICE DE TABLAS.....	IX
CAPÍTULO 1.....	1
1. INTRODUCCIÓN.....	1
1.1. Descripción del problema.....	2
1.2. Justificación.....	4
1.3. Objetivos.....	5
1.3.1 Objetivo general.....	5
1.3.2. Objetivos específicos.....	5
1.4. Metodología.....	6
CAPÍTULO 2.....	8
2. MARCO TEÓRICO.....	8
2.1. Sistema de videovigilancia.....	8
2.1.1 Tipos.....	8
2.1.2 Características.....	10
2.2. Cadena de Bloques.....	11
2.2.1 Características.....	13
2.2.2 Aplicaciones.....	14
2.2.3 Tecnologías.....	16

2.2.4.Ventajas y desventajas	18
2.2.5.Contratos inteligentes.....	18
CAPÍTULO 3.....	20
3.DISEÑO DEL SISTEMA DE VIDEOVIGILANCIA CON UN MÓDULO DE AUTENTICACIÓN	20
3.1. Descripción del sistema de videovigilancia seguro.....	20
3.1.1 Hardware.....	20
3.1.2 Software	20
3.2. Arquitectura del sistema de videovigilancia seguro	21
3.3. Diseño del sistema de videovigilancia con un inicio de sesión seguro basado en cadena de bloques.....	23
3.3.1.Creación de la cadena de bloques	23
3.3.2 Despliegue del contrato inteligente	24
3.3.3.Interfaz de Usuario.....	25
3.3.4.Registro de usuario	26
3.3.5.Inicio de sesión	28
3.3.6 Adquisición de video de la cámara.....	30
CAPÍTULO 4.....	31
4.IMPLEMENTACIÓN Y PRUEBAS DE LA RED DEL SISTEMA DE VIDEOVIGILANCIA	31
4.1. Integración del sistema de videovigilancia con el método de autenticación.....	31
4.1.1 Hardware.....	31
4.1.2 Software	32

4.2. Pruebas y análisis de resultados del sistema de videovigilancia...	44
4.2.1.Pruebas de funcionalidad del método de autenticación.	44
4.2.2.Pruebas de seguridad en la cadena de bloques y análisis de resultados.....	45
CAPÍTULO 5.....	56
5. CONCLUSIONES Y RECOMENDACIONES	56
5.1. CONCLUSIONES.....	56
5.2. RECOMENDACIONES	57
BIBLIOGRAFÍA.....	58
ANEXOS.....	63

ÍNDICE DE FIGURAS

Figura 1.1: Esquema de un sistema analógico.....	3
Figura 1.2: Diagrama de la Solución	6
Figura 2.1: Esquema tradicional de un sistema analógico[12].....	9
Figura 2.2: Esquema de un sistema de vigilancia por IP [12].	10
Figura 2.3: (a) un sistema centralizado con intermediarios vs. (b) un sistema de cadena de bloques descentralizado[15].	12
Figura 2.4: Principales características de una cadena de bloques.....	14
Figura 2.5: Esquema de un contrato inteligente.	19
Figura 3.1: Arquitectura del sistema de videovigilancia seguro	21
Figura 3.2: Esquema sistemático del sistema de videovigilancia seguro	22
Figura 3.3: Emulador ganache con metamask[30]	24
Figura 3.4: Estructura del contrato inteligente	25
Figura 3.5: Página para inicio de sesión.....	25
Figura 3.6: Esquema de registro de usuarios.	26
Figura 3.7: Flujograma de registro de usuarios.	27
Figura 3.8: Esquema de inicio de sesión.....	28
Figura 3.9: Flujograma de inicio de sesión.	29
Figura 3.10: Código de servidor RTSPtoWeb.....	30
Figura 4.1: Conexión física de la cámara IP Hikvision al puerto LAN del router	31
Figura 4.2: Portal oficial de Ganache[31].....	32
Figura 4.3: Página inicial de Ganache luego de instalación	32

Figura 4.4: Llaves disponibles con 100 ETH de saldo para pruebas.....	33
Figura 4.5: Información de la cuenta	33
Figura 4.6: Portal oficial de metamask[32].....	34
Figura 4.7: Inicio de sesión en metamask	34
Figura 4.8: Importación de cuenta Ganache en metamask.	35
Figura 4.9: Cuenta importada a la red de Ganache.....	35
Figura 4.10: Plataforma oficial de Web3js[33].	36
Figura 4.11: Plataforma oficial Node.js [34].	36
Figura 4.12: Instalación de WLS.....	37
Figura 4.13: Características de Windows.	37
Figura 4.14: Instalación de curl en Ubuntu.	38
Figura 4.15: Instalación de Solidity.....	38
Figura 4.16: Instalación de truffle.	39
Figura 4.17: Verificación de librerías instaladas.	39
Figura 4.18: Aplicación creada en Visual Studio.	39
Figura 4.19: Se ejecuta nmp para iniciar la interfaz.....	40
Figura 4.20: Log de creación de usuario en Ganache.	40
Figura 4.21: Log de inicio de sesión en Ganache.....	41
Figura 4.22: Plataforma de videovigilancia para inicio de sesión.	41
Figura 4.23: Creación de usuario.	42
Figura 4.24: Portal de monitoreo del sistema de videovigilancia	42
Figura 4.25: Librería RTSPtoWeb.	43
Figura 4.26: Código del servidor RTSP.	43

Figura 4.27: Billetera digital no iniciada.	44
Figura 4.28: Billetera digital sin crédito.	45
Figura 4.29: Ingreso de URL en Owasp Zap.	46
Figura 4.30: Resultado de vulnerabilidades en Owasp Zap.	46
Figura 4.31: Puerta de enlace de Sistema de videovigilancia y la de cadena de bloques	48
Figura 4.32: Dirección IP de cámara	48
Figura 4.33: Inicio de Ettercap.....	49
Figura 4.34: Agregando las IPs de los objetivos 1 y 2.....	49
Figura 4.35: ARP Poisoning.	50
Figura 4.36: La identificación de conexiones remotas.....	50
Figura 4.37: Resultados del Ataque MIM.....	51
Figura 4.38: Captura de tráfico con Wireshark.	51
Figura 4.39: Huella de la transacción en los registros de Ganache.....	52
Figura 4.40: Instalación de framework Metasploit	53
Figura 4.41: Ejecución msf6.	53
Figura 4.42: Parámetros de red de Ganache	54
Figura 4.43: Inicio del ataque con exploit	54
Figura 4.44: Paquetes recibos en el equipo destino por el exploit.....	54
Figura 4.45: Registro de transacciones de Ganache en la ejecución de exploit	55

ÍNDICE DE TABLAS

Tabla 2.1. Comparación de las plataformas de código abierto.....	18
Tabla 4.1. Resumen de resultados obtenidos en el pentesting	47
Tabla 4.2. Direccionamiento de dispositivos del Sistema de videovigilancia y la red de cadena de bloques.	48

CAPÍTULO 1

1. INTRODUCCIÓN

Sin lugar a duda, la videovigilancia se ha convertido en un aspecto crucial para salvaguardar numerosos entornos, desde residencias privadas y empresas hasta áreas públicas y lugares de alto riesgo, en la actual era digital. Sin embargo, con el avance tecnológico surgen los correspondientes desafíos de seguridad. Uno de los problemas más urgentes es el robo de información, donde los atacantes pueden interceptar y modificar los datos registrados por las cámaras de videovigilancia.

En respuesta a esta preocupación, surge la tecnología cadena de bloques, una innovación que ha transformado la forma en que garantizamos la seguridad, integridad y confianza en la era digital. La Cadena de bloques, Inicialmente reconocida por su conexión con las criptomonedas[1], ha ampliado su alcance a varios campos. Uno de sus usos más auspiciosos es el ámbito de la seguridad por videovigilancia.

El presente proyecto de titulación tiene un enfoque innovador de un sistema de videovigilancia con autenticación basada en cadena de bloques. Esta solución no solo aborda los desafíos de autenticación y control de acceso, sino que también resuelve de manera efectiva el problema crítico de robo de información. A través de la inmutable naturaleza de la cadena de bloques y la autenticación descentralizada, este sistema establece nuevos estándares de seguridad garantizando la confianza y la pureza de los datos en toda la información obtenida.

1.1. Descripción del problema

Actualmente, se siguen implementando sistemas de videovigilancia sin considerar que su arquitectura puede ser propensa a vulnerabilidades, dado que usan sistemas centralizados donde cada red tiene un nodo central que es el responsable de transportar la información[2]. De hecho, los sistemas de videovigilancia que son usados para cualquier ambiente ya sea domésticos, empresariales, gubernamentales, etc., carecen de seguridad de autenticación robusta. Dicho de otra manera, utilizan métodos tradicionales de inicio de sesión, lo cual ha sido aprovechado por los ciberdelincuentes para ejecutar intrusiones a los sistemas de video seguridad como accesos no autorizados, espionaje y alteraciones de archivos de video almacenados.

Según la red autónoma bricolaje (DIY) estima que aumente un ochenta por ciento para el 2024 el mercado a nivel mundial en cámaras[3]. La utilización de estos sistemas de videovigilancia crece de forma considerable debido a la necesidad de seguridad física, por lo tanto, debe ser motivo de robustecer sus sistemas de autenticación para el control y gestión de los circuitos de videovigilancia.

Los métodos de autenticación son utilizados como herramienta para la validación de la identidad de los usuarios que requieren acceder a los sistemas de videovigilancia. Esta información de identidad es almacenada por los usuarios en servidores centralizados y que sirven para compararlos en el momento que alguien desee ingresar mediante una aplicación a la red de videovigilancia garantizando el acceso a personal autorizado. Hoy en día, se utilizan plataformas de autenticación con un servidor central que se valida en Internet desde una red de acceso, computadoras o teléfonos inteligentes, para tener el control o gestión de la red de seguridad por cámaras como se observa en la figura 1.1.

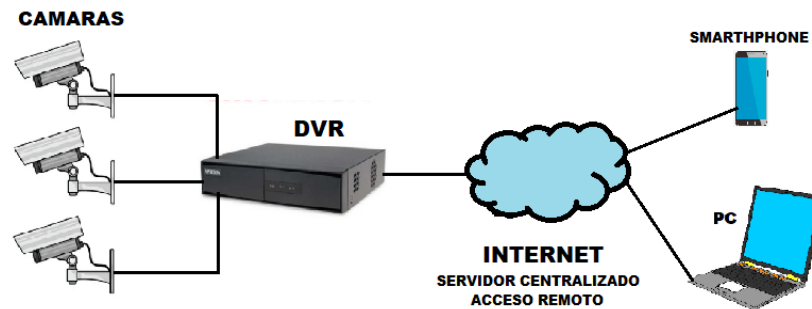


Figura 1.1: Esquema de un sistema analógico

Los métodos de autenticación, tales como: Autenticación QR y validación de credenciales por Doble Factor de Autenticación, son medios que requiere de dos niveles (factores) para poder autenticar su identidad. Estos métodos, si bien es cierto proporcionan un nivel adicional de seguridad, no facilita un rápido acceso y en ciertas ocasiones puede ponerse en contra del usuario cuando por cambios de dispositivos o pérdidas les impide autenticarse [4]. Además, la información del segundo nivel de autenticación es proporcionada desde un repositorio central y su objetivo es almacenar toda la información necesaria para autenticar al usuario. Aunque el doble factor aumenta el nivel de seguridad con la segunda capa de autenticación, aún se presenta la novedad de que la base de datos sigue siendo centralizada y almacena una lista de información secreta del usuario.

Debido a la poca importancia o desconocimiento en implementar los sistemas de videovigilancia con un inicio de sesión seguro, han surgido ciertos acontecimientos que se mencionan a continuación: En el 2016, se produjo un ataque masivo a cámaras de seguridad IP. Los agresores se apoderaron de algunas cámaras de seguridad conectadas a Internet con la finalidad de realizar ataques de denegación de servicio. Esto fue posible porque las credenciales de autenticación de algunos usuarios fueron débiles. Los ciberdelincuentes utilizaron programa maligno para infectar los sistemas de videovigilancia. La mayor cantidad de las cámaras infectadas ocurrió en Taiwán con un 24%, EE. UU. con un 12%, Indonesia un 9%, México 8%, Malasia 6%, Israel 5%, Vietnam 2%, Francia 2% y España 2%. En total, se localizaron en 105 países los sistemas infectados[5]. En octubre del 2018 en Madrid España, el partido

político “Podemos” denunció una intrusión a cámaras de videovigilancia que se habían instalado en los exteriores de la vivienda de personajes reconocidos localmente. Dichas cámaras tenían como propósito vigilar los exteriores de la urbanización. Estas imágenes fueron transmitidas en un sitio web con acceso público [6].

En marzo del 2020 en el Reino Unido, se produjo una intrusión considerable a cámaras de vigilancia y monitores de bebés. En muchos de estos dispositivos que se conectan al Internet no se realizaron el cambio de contraseñas por defecto de los fabricantes y por tal motivo fue un blanco para los ciberdelincuentes[7]. Asimismo, en marzo del 2021 en Estados Unidos, ciberatacantes tomaron el control de aproximadamente 150.000 cámaras de seguridad de centros de salud, empresas públicas y privadas, departamentos de policías, cárceles, escuelas inclusive una empresa de tecnología reconocida a nivel mundial como Tesla, demostrando su intrusión mediante transmisiones en vivo[8]. En Ecuador aún no hay registros oficiales que evidencien intrusiones masivas a los sistemas de videovigilancia, esto no excluye o exime que pueda llevarse a cabo, puesto que la videovigilancia con cámaras IP están implementadas en muchas empresas públicas y privadas

1.2. Justificación

En el artículo [9] proponen un análisis de las ventajas principales de cadena de bloques para Internet de la cosas y retos en la integración a las posibles topologías. En [10] plantean el diseño de un sistema de autenticación de seguridad basado en Hyperledger Fabric, como prueba de concepto usando únicamente 2 computadoras como dispositivos finales. En los casos anteriores no se ha tomado en consideración como caso de estudio en sistemas de videovigilancia y en usar más dispositivos físicos para probar o evaluar la confiabilidad y el rendimiento del sistema.

En Ecuador siendo un país en desarrollo tecnológico se ha enfocado en el uso de la tecnología de cadena de bloques para ámbito financiero y en el sistema jurídico, mas no para otras soluciones que tendrían un impacto crucial en la sociedad. Con la implementación de este sistema de videovigilancia utilizando

un módulo de autenticación descentralizado con cadena de bloques se desea mitigar las intrusiones no deseadas a los sistemas de videovigilancia y promover su uso para la seguridad de información sensible, dado que en la actualidad se sigue usando métodos convencionales de validación de credenciales teniendo acceso a los sistemas de video seguridad y sus archivos almacenados.

Además, este proyecto tendrá un impacto de índole académico, tecnológico y social, siendo un aporte crucial en la comunidad de las telecomunicaciones para que se siga mejorando el desarrollo de estos sistemas de seguridad. De esta manera se aspira brindar una contribución importante en el estudio de esta tecnología de cadena de bloques, nodos descentralizados y registros inalterables. Por otra parte, como un aporte social se desea que no solo los dispositivos de infraestructuras empresariales o industriales tengan este beneficio de protección de datos y autenticación, sino de manera general a usuarios puntuales que tengan sistemas de videovigilancia en sus hogares, oficinas o negocios, puesto que también pueden ser víctimas de estos delitos. Por tal motivo, se espera contribuir en el ámbito de futuros negocios o emprendimientos debido a los beneficios que esta tecnología nos brinda.

1.3. Objetivos

1.3.1. Objetivo general

Implementar un sistema de videovigilancia con un método de autenticación basado en cadena de bloques para mitigación de accesos no autorizados.

1.3.2. Objetivos específicos

- Analizar las tecnologías requeridas para el desarrollo de la interfaz del sistema de videovigilancia y el módulo de autenticación.
- Implementar la red de videovigilancia utilizando cámaras de uso doméstico y software de código abierto.

- Desarrollar el módulo de autenticación para el acceso utilizando tecnología de cadena de bloques.
- Integrar el sistema de videovigilancia con el módulo de autenticación descentralizado utilizando software de código abierto
- Evaluar el desempeño de la red de videovigilancia mediante un plan de pruebas.

1.4. Metodología

La implementación del proyecto de titulación está compuesta de 3 fases (ver figura 1.2). En la fase número 1, se implementa una red de videovigilancia local, utilizando cámaras IP de uso doméstico y un software de administración de código abierto. En la fase número 2, se desarrolla una red de cadena de bloques utilizando software de código abierto y máquinas virtuales para realizar el despliegue de manera local. Finalmente, en la fase número 3, se procede con el desarrollo de un módulo de autenticación, para luego ser integrado con la interfaz web de monitoreo, para que por medio de la cadena de bloques se realice la validación de acceso.

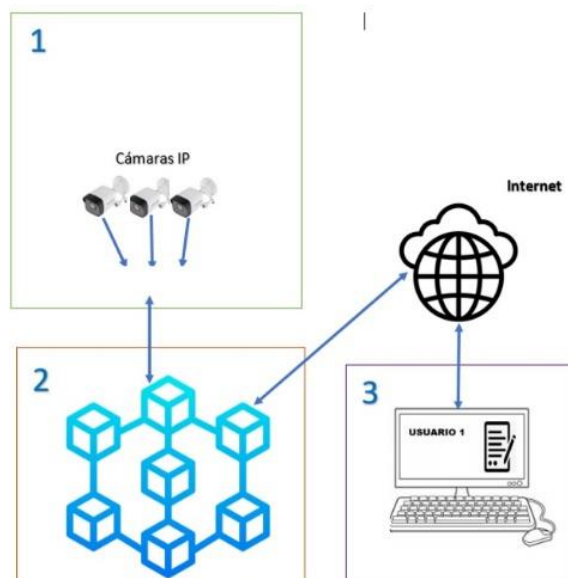


Figura 1.2: Diagrama de la Solución

Este proyecto de titulación está estructurado de la siguiente manera: En el capítulo 2 se hace una breve introducción de los conceptos de cadena de bloques, contratos inteligentes y comparativas de plataformas de código abierto. En el Capítulo 3 Se presenta el diseño del sistema de videovigilancia con el método de autenticación usando cadena de bloques, en el capítulo 4, continuamos con la implementación y se desarrolla pruebas de seguridad en donde se evaluará la cadena de bloques y, por último, capítulo 5, describe las conclusiones y recomendaciones del trabajo de titulación.

CAPÍTULO 2

2. MARCO TEÓRICO

El presente capítulo tiene objetivo explorar las bases teóricas y conceptuales subyacentes a la integración de la cadena de bloques como un método de autenticación en sistemas de videovigilancia. Se examina los elementos clave de la cadena de bloques, su aplicación en la autenticación de datos, y cómo esta tecnología puede abordar los desafíos de seguridad y autenticación en sistemas de videovigilancia.

2.1. Sistema de videovigilancia

A lo largo de la década, ha habido un aumento en la cantidad de sistemas de videovigilancia empleados tanto por hogares como por empresas que requieren dichos servicios. La versatilidad de estos sistemas radica en su capacidad de instalarse interna o externamente, brindando a los usuarios opciones de monitoreo en vivo o remoto a través de Internet. Estos sistemas poseen una influencia disuasiva, debido a que son percibidos por los individuos y de esta manera puede evitar que se lleve a cabo algún acto antisocial. Las demandas de la sociedad con relación a una videovigilancia de calidad, segura, escalable, que permita incorporarlo con otros sistemas y sea de menor costo, han generado que esta área experimentara transformaciones en la tecnología.

2.1.1. Tipos

Hay distintos tipos de sistemas como son los tradicionales que requieren de un equipamiento que usa cable coaxial, este sistema está pensado para que la transmisión de video ocurra en el mismo lugar. Con la llegada del video digital, ha habido una evolución de otros medios de transmisión, como la fibra óptica o los cables de par trenzado. Los avances logrados por estas tecnologías brindan a los fabricantes de cámaras o productos de almacenamiento una amplia diversidad en la

creación de sus respectivos sistemas.[11]. A continuación, se lista los diferentes tipos de sistemas de videovigilancia:

- **Sistemas Analógicos:** utilizan videocámaras que tienen conexiones de vídeo VGA y se visualizan en monitores estáticos que únicamente tienen la función de reproducir las imágenes capturadas por las cámaras conectadas. Para mejorar la gestión y orientación de las cámaras en el marco de un sistema convencional, se pueden emplear matrices de Vídeo o multiplexores. Estos sistemas utilizan microprocesadores para dirigir las entradas (cámaras) hacia las salidas (televisores)[12]. En tiempos modernos, es esencial que cualquier sistema de videovigilancia analógico incorpore un medio de recopilación de pruebas que pueda monitorearse para obtener información crucial y reducir la carga del monitoreo manual. Normalmente, las imágenes se graban y guardan en un VCR, un grabador de video, ver figura 2.1. En esencia, un sistema analógico, sin tener en cuenta el control del movimiento de la cámara, las capacidades de audio o las funciones complementarias.

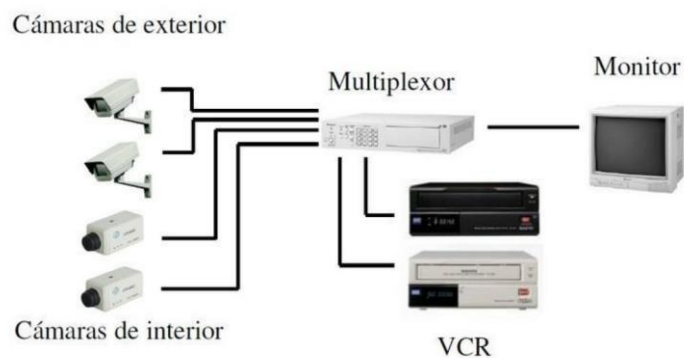


Figura 2.1: Esquema tradicional de un sistema analógico[12].

- **Sistema digital:** se basa en tecnología IP, solo incluye cableado en el punto de instalación. La comunicación se basa en el protocolo TCP/IP y se encarga de establecer una conexión directa entre las cámaras y la red en el lugar donde se

encuentran. El servidor y la cámara transmiten transmisiones de video que se pueden ver tanto en la red local como en Internet, ver figura 2.2.

El acceso a imágenes desde Internet se puede realizar mediante diversos métodos. Es posible utilizar un servidor web o ftp, ver vídeos en streaming en una página web alojada o acceder a imágenes a través de una intranet si el modelo de cámara lo permite. Estos sistemas se están expandiendo constantemente debido a su notable funcionalidad, adaptabilidad, escalabilidad y capacidad para integrarse sin problemas con las tecnologías actuales.

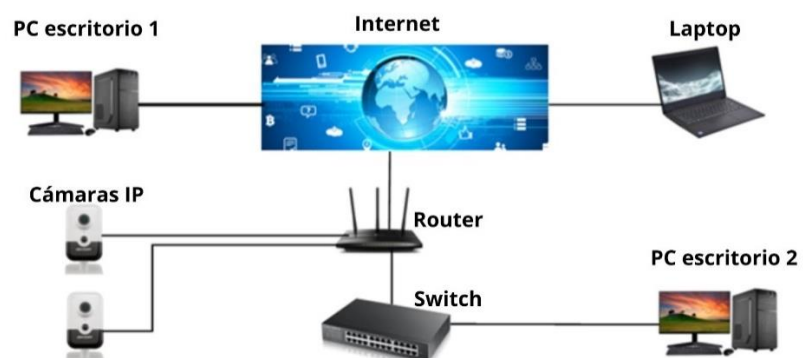


Figura 2.2: Esquema de un sistema de vigilancia por IP [12].

2.1.2. Características

Un sistema de videovigilancia cuenta con las siguientes características:

- Las cámaras de vigilancia son herramientas capaces de capturar y transmitir imágenes en tiempo real.
- Una de las características fundamentales es la capacidad de acumular y salvaguardar las imágenes capturadas.
- Es posible observar imágenes en vivo mediante la utilización de monitoreo en tiempo real.
- La capacidad de acceder remotamente a imágenes desde lugares más allá de la región monitoreada.

- La detección de movimiento es función que está diseñada para activar la grabación o señalar una alerta cada vez que se detecta movimiento dentro del área monitoreada.
- La acción de alertar a los usuarios cuando se detectan ciertos eventos se denomina comúnmente notificaciones.
- La calidad de una imagen puede determinarse por su resolución y fidelidad visual.
- La capacidad de integración es una característica de los sistemas o dispositivos de seguridad, que les permite interactuar con otros sistemas o dispositivos de forma fluida.
- El control de acceso ayuda regular la visualización de imágenes grabadas. Esto implica imponer restricciones sobre quién puede acceder a estas imágenes. [13].

2.2. Cadena de Bloques

Esta tecnología en particular es tanto pública como descentralizada, y funciona como un medio para producir una base de datos común que todos los participantes pueden utilizar para rastrear sus transacciones. Se puede comparar con un libro mayor de contabilidad, inmutable[14] y colectivo que una gran cantidad de computadoras escriben al mismo tiempo. Cada vez que un miembro de la red participa en una transacción digital, produce datos relevantes que se guardarán en un bloque. Una vez que este bloque haya alcanzado su capacidad, se vinculará a la cadena de bloques existente[15] [16].

El contenido guardado dentro de una red en particular depende de la intención original de su creación. Esto significa que la red podría utilizarse para diversos fines, como almacenar información de pago (como criptomonedas), métodos de autenticación, registros médicos, logística o datos de trazabilidad de alimentos, o incluso recuentos de votos electorales.

En comparación con una red centralizada que almacena datos en un solo servidor, la red de cadena de bloques opera en múltiples computadoras en todo el mundo. Esta característica única otorga a la red de cadena de bloques varios beneficios, incluida una mayor privacidad y descentralización, lo que le permite

funcionar sin depender de una autoridad centralizada y una mayor seguridad[16]. La distinción entre una ejecución centralizada de transacciones y un sistema de cadena de bloques descentralizado se ejemplifica en la figura 2.3. Donde (a) es un sistema centralizado con intermediarios y todos los participantes deben acudir a él para obtener registros y en (b) es un sistema descentralizado en la que cualquiera puede acceder, el registro es compartido, transparentado y ejecutado por todos los participantes.

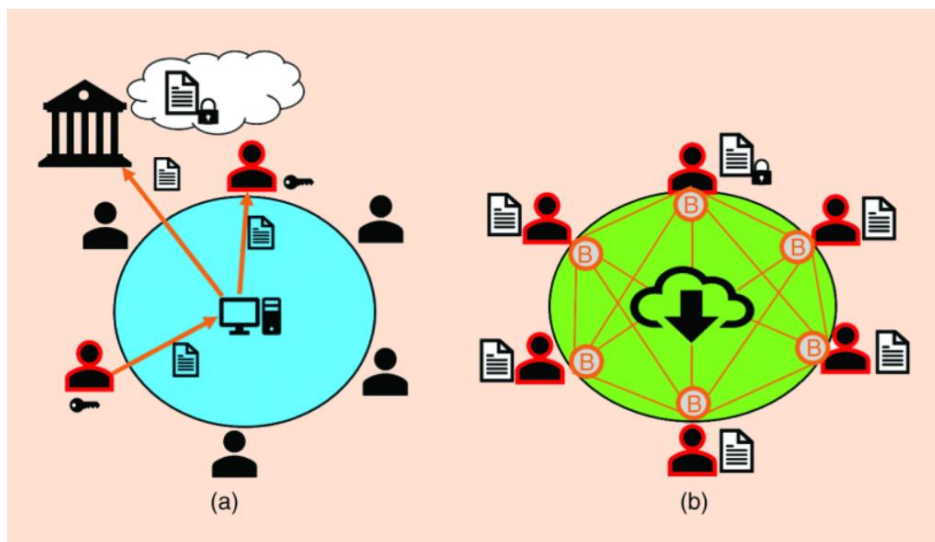


Figura 2.3: (a) un sistema centralizado con intermediarios vs. (b) un sistema de cadena de bloques descentralizado[15].

Las cadenas de bloques se pueden categorizar según sus propiedades y reglas predefinidas. Estas categorías tienen sus propios beneficios y limitaciones, lo que las hace perfectas para diversas aplicaciones. En general, existen dos categorías: la primera se basa en la apertura de la red, mientras que la segunda se basa en el nivel de permisos necesarios para agregar datos. Las cadenas de bloques públicas ofrecen transparencia a cualquiera que desee acceder a los datos presentes en la cadena. Por el contrario, solo unos pocos seleccionados con acceso autorizado pueden acceder a la información almacenada en cadenas de bloques privadas.

2.2.1. Características

Las características inherentes de la cadena de bloques [17] juegan un papel crucial en su funcionamiento[18] las más importantes son:

- **Descentralización:** es un atributo clave de una red de cadena de bloques es debido a que la red trabaja de manera distribuida, sin ningún ente centralizado que tenga control sobre los datos que se almacena en la red.
- **Seguridad:** garantizan que el contenido de cada bloque sea genuino y no esté corrupto.
- **Inmutabilidad:** garantiza que los datos que se ingresan en un bloque sean altamente confiables, ya que es muy difícil alterar la información registrada una vez que se ha agregado al bloque.
- **Transparencia:** característica clave de los datos almacenados en la cadena de bloques, dado que sólo los participantes de la red pueden ver los datos, se fomenta una cultura de apertura y claridad.
- **Consenso:** evita cualquier caso de fraude o replicación, todos los usuarios conectados en una red deben estar de acuerdo sobre la validez de cualquier nuevo bloque que se agregue. Este acuerdo es necesario antes de poder realizar cualquier adición.
- **Irreversibilidad:** Una vez que se agrega un bloque a la cadena, su eliminación se convierte en una tarea imposible, que sirve para reforzar el historial de transacciones.
- **Automatización:** los contratos inteligentes ayudan a la automatización porque se ejecutan acciones predeterminadas automáticamente después de que se cumplan condiciones específicas.
- **Eficiencia:** la capacidad de agilizar procesos y reducir costos se debe porque elimina intermediarios y minimiza los requisitos de conciliación de datos.

- Integridad: La cadena de bloques es a prueba de manipulación, garantiza la pureza de los datos.
- Historial: capacidad para almacenar registros de cada transacción realizada. Esta característica es particularmente valiosa para fines de seguimiento y auditoría.

La tecnología es autónoma y mantiene el anonimato de los participantes de la transacción mediante la utilización de credenciales públicas y privadas. Las características principales de la cadena de bloques se muestran en la figura 2.4.



Figura 2.4: Principales características de una cadena de bloques.

2.2.2. Aplicaciones

Entre los usos de la tecnología de cadena de bloque tenemos:

- Criptomonedas: Las monedas digitales son representaciones virtuales de dinero que utilizan la red de cadena de bloques para permitir transacciones seguras entre las partes sin la participación de intermediarios. Un buen ejemplo de esto es Bitcoin, que documenta cada transacción realizada utilizando su moneda en la red de cadena de bloques, funcionando como un libro de contabilidad.

- **Contratos inteligentes:** son una herramienta esencial en el mundo de las criptomonedas; sin embargo, sus aplicaciones potenciales se extienden más allá de este campo. Proporcionan a las partes la capacidad de generar acuerdos sin la participación de intermediarios, gracias a la cadena de bloques, que exige el cumplimiento del contrato. Una posible aplicación de los contratos inteligentes es la automatización de los pagos de nómina, con la condición de que los empleados solo reciban su pago si el proyecto se completa antes de fin de mes. Esto garantiza que todas las partes cumplan el acuerdo sin falta.[19].
- **Almacenamiento y seguimiento de datos:** La red ha permitido la creación de bases de datos a las que se permite ingresar, compartir y modificar en tiempo real, lo que permite un almacenamiento y monitoreo de datos eficientes. Esta característica es especialmente útil en las cadenas de distribución, como en el seguimiento de productos alimenticios. La red cadena de bloques se utiliza para registrar cada paso del proceso, desde la producción hasta la venta, lo que garantiza la transparencia y la trazabilidad.
- **Método de autenticación:** La sugerencia de utilizar la tecnología de cadena de bloques para mejorar las medidas de seguridad en comparación con los métodos convencionales es una opción creíble. Esto se atribuye en gran medida a la estructura descentralizada de la red de cadena de bloques y a la utilización de algoritmos criptográficos, que proporcionan un mayor nivel de protección contra las amenazas cibernéticas, al tiempo que eliminan la necesidad de intermediarios centralizados. Además, la integración de contratos inteligentes pone en primer plano una capa adicional de medidas de seguridad personalizadas.[20].
- **Votación electrónica:** Garantiza la seguridad y la lucidez de los procedimientos de votación electrónica al crear un libro de votos

inmutable y ayudar eficazmente contra el fraude electoral y brindar confianza en los resultados.

- Propiedad intelectual: es solución prometedora para el seguimiento de obras protegidas por derechos de autor desde su creación hasta su distribución y uso. Al hacerlo, tiene el potencial de agilizar la gestión de licencias y regalías, y ofrecer un registro claro y transparente del historial de propiedad intelectual.
- Propiedad de activos: Facilita el seguimiento de los activos como por ejemplo los bienes inmuebles, el proceso de transferencia de propiedad se simplifica y verifica su precisión.
- Educación: Puede proporcionar una plataforma segura para la verificación de credenciales tanto educativas como profesionales. Esto permitiría a estudiantes y profesionales acceder a registros permanentes de sus logros académicos y certificaciones a través de un sistema inalterable.

2.2.3. Tecnologías

En el ámbito del desarrollo de aplicaciones, varias plataformas de cadena de bloques están a nuestra disposición. Los principios básicos de esta tecnología se modifican para adaptarse a industrias particulares o casos de uso para cada protocolo de cadena. Esta subsección proporciona algunos ejemplos de plataformas de código abierto[21]:

- Hyperledger Fabric; es un proyecto de código abierto y está diseñado para ayudar a las empresas en la construcción rápida y eficiente de aplicaciones privadas de cadena de bloques. Lo que distingue a este proyecto es su arquitectura modular, así como sus niveles excepcionales de confidencialidad, resiliencia, flexibilidad y escalabilidad. Además, tiene un libro mayor compartido, transacciones y contratos inteligentes, denominados “chaincode” en Hyperledger Fabric. Estas cualidades lo hacen apropiado para una numerosa variedad de aplicaciones, que incluyen, entre otras, el control de las cadenas de suministro, las

finanzas en el comercio, las recompensas y la lealtad, y la compensación y liquidación de activos financieros[21], [22].

- Ethereum: es denominado una cadena de bloques pública y posee una característica única llamada contratos inteligentes. Estos contratos inteligentes o también conocidos como su terminología en inglés smart contracts, son esencialmente códigos ejecutables que realizan transacciones dentro de Ethereum. Al utilizar contratos inteligentes, Ethereum se adapta a la programación y se puede utilizar para desarrollar aplicaciones móviles y web descentralizadas. Estos sistemas a veces se denominan "DApps" o aplicaciones descentralizadas. Ethereum es esencialmente una red de computadoras de código abierto que se utiliza para la transferencia de fondos entre varias partes y el almacenamiento de datos[23]. El término "Ethereum" no se refiere únicamente a una criptomoneda. Más bien, es una plataforma digital. Las unidades de moneda utilizadas como pago dentro de la red Ethereum se denominan "Ethers" o "gas". En esencia, son criptomonedas que es utilizada para comprar servicios de la red Ethereum[24].
- Ganache: Proporciona un medio para el desarrollo eficiente y eficaz de las aplicaciones distribuidas en las cadenas de bloques Ethereum y Filecoin. Su cadena de bloques personal es una herramienta ideal para utilizar durante todo el proceso de desarrollo; permitiendo el desarrollo, la implementación y las pruebas seguras de sus dApps en un entorno seguro y predecible.
Ganache está disponible en un aplicativo gráfico de usuario y una interfaz de línea de comandos. La interfaz de usuario de Ganache es una aplicación que se puede usar en un escritorio y ofrece soporte para las tecnologías Ethereum y Filecoin. Mientras tanto, la CLI de ganache es una herramienta muy robusta que se utiliza para fines de desarrollo de Ethereum[25]. Ganache viene equipado con un cliente Ethereum que está integrado con los últimos estándares Ethereum. Permite el uso de muchas cuentas de Ethereum con saldos predeterminados, que pueden utilizarse para experimentar con diferentes escenarios. Ganache se puede integrar fácilmente con las

reconocidas herramientas de desarrollo de Ethereum, como Remix, Truffle y Web3.js[26].

2.2.4. Ventajas y desventajas

En la tabla 2.1 se presenta un resumen comparativo de tres plataformas de cadena de bloques con código abierto.

Tabla 2.1. Comparación de las plataformas de código abierto.

PLATAFORMAS	VENTAJAS	DESVENTAJAS
Hyperledger Fabric	<ul style="list-style-type: none"> *El sistema proporciona un libro mayor inalterable y descentralizado. *Tiene la capacidad de ofrecer niveles significativos de escalabilidad, rendimiento y confiabilidad. *Requiere un permiso de membresía.[27]. 	<ul style="list-style-type: none"> *El equipo de programación no está compuesto por programadores muy expertos. *Se ha demostrado la escasez de aplicaciones prácticas. *La estructura de este objeto es intrincada y multifacética. *La API y el SDK proporcionados tienen un alcance mínimo. *No es una red tolerante a fallas[28].
Ethereum	<ul style="list-style-type: none"> *Fiabilidad. *El respaldo de las grandes corporaciones. *Descentralización. *La probabilidad de inflación es mínima. *No hay límite en los ETH[29]. 	<ul style="list-style-type: none"> *La transparencia. *Siempre ha sido la segunda de bitcoin. *Problemas de escalabilidad. *Creciente precio del gas. *Ethereum 2.0 crea conflictos en la comunidad[29].
Ganache	<ul style="list-style-type: none"> *Simplicidad cuando se trata de crear, probar y lanzar contratos inteligentes y proyectos dApp. *Su naturaleza determinista garantiza resultados confiables, lo que ayuda a minimizar la probabilidad de errores durante el proceso de desarrollo. *Ofrece dos variaciones distintas, cada una con diferentes capacidades para satisfacer diferentes necesidades[26]. 	<ul style="list-style-type: none"> *Como red privada, no puede replicar con precisión las acciones de minería que ocurren en la red principal. Como resultado, esto puede causar problemas cuando los desarrolladores requieren la prueba de contratos inteligentes que dependen de las acciones de los mineros. *El límite de gas en la red primaria es dinámico, lo que puede generar complicaciones al emplear números precisos dentro de Ganache[26].

2.2.5. Contratos inteligentes

Es un programa que opera en la cadena de bloques y tiene la capacidad de auto verificarse, autoejecutarse y resistir la manipulación. Su

implementación brinda una amplia flexibilidad para crear, diseñar y resolver problemas del mundo real con menores costos e inversión de tiempo, sin la necesidad de sistemas tradicionales de terceros. Los contratos inteligentes son tanto irrevocables como rastreables. Todos los detalles de la transacción se incluyen en un contrato inteligente y luego se ejecutan automáticamente.

Para garantizar la precisión y velocidad del proceso de utilización de recursos, es imperativo cumplir con cada condición. Además, cuando se establece en la red de cadena de bloques, se fortalece la seguridad de todas las partes involucradas. Esto se logra mediante la inmutabilidad de la información dentro de la red, debido tanto al sistema de consenso como a las propiedades de la red distribuida. Como resultado, todas las transacciones y datos registrados dentro de la cadena permanecen permanentes a menos que sean necesarias modificaciones (ver figura 2.5).

Los contratos inteligentes tienen la ventaja de ser fácilmente programables de forma serializada. Esto se logra almacenando sus códigos dentro de una cadena de bloques o protocolos compartidos entre redes informáticas. Los contratos están diseñados para ejecutarse cuando se produce un evento desencadenante específico, tal como se define en el contrato. La facilidad de serialización y ejecución basada en eventos hacen de los contratos inteligentes un activo valioso.



Figura 2.5: Esquema de un contrato inteligente.

CAPÍTULO 3

3. DISEÑO DEL SISTEMA DE VIDEOVIGILANCIA CON UN MÓDULO DE AUTENTICACIÓN

En este capítulo se explica el procedimiento seguido para desarrollar un sistema de videovigilancia de código abierto y con un método de autenticación utilizando cadena de bloques, Además se describe el desarrollo del hardware y software del proyecto.

3.1. Descripción del sistema de videovigilancia seguro.

El sistema de videovigilancia seguro de cadena de bloques es una solución que está diseñada específicamente para asegurar la autenticidad, confidencialidad e integridad de todos los datos de videovigilancia. Esta innovadora combinación de tecnologías aborda los problemas que surgen del acceso no autorizado y la manipulación de imágenes capturados por cámaras de vigilancia. Es importante señalar que, durante esta etapa, se seleccionaron y definieron cuidadosamente el hardware y software de código abierto que se utiliza para el desarrollo de la interfaz web de monitoreo del sistema de videovigilancia, el método de autenticación y la integración de los sistemas antes mencionados.

3.1.1. Hardware

A continuación, se describen los componentes de hardware específicos del plan propuesto:

- Cámara IP
- Cable UTP
- Laptop

3.1.2. Software

A continuación, se mencionan los componentes de software a intervenir en la propuesta planteada:

- Ganache: Cadena de bloques Ethereum local.
- Web3js: Librería javascript para trabajar con redes Ethereum.
- Truffle: Librería para el desarrollo de contratos inteligentes

- Solidity: Compilador de contratos inteligentes.
- Reactjs: Librería para el desarrollo de aplicaciones front-end.
- Nodejs: Servidor web en donde visualizaremos nuestra interfaz de videovigilancia.
- Metamask: Extensión de Chrome para manejo de una billetera digital.

3.2. Arquitectura del sistema de videovigilancia seguro

La arquitectura general se centra en el método de autenticación usando cadena de bloques como eje principal, donde está dividido en 5 bloques para comprender mejor el sistema: Servidor web, interfaz web, plugin billetera digital, contrato inteligente, servidor de cadena de bloques, tal como se muestra en la figura 3.1.

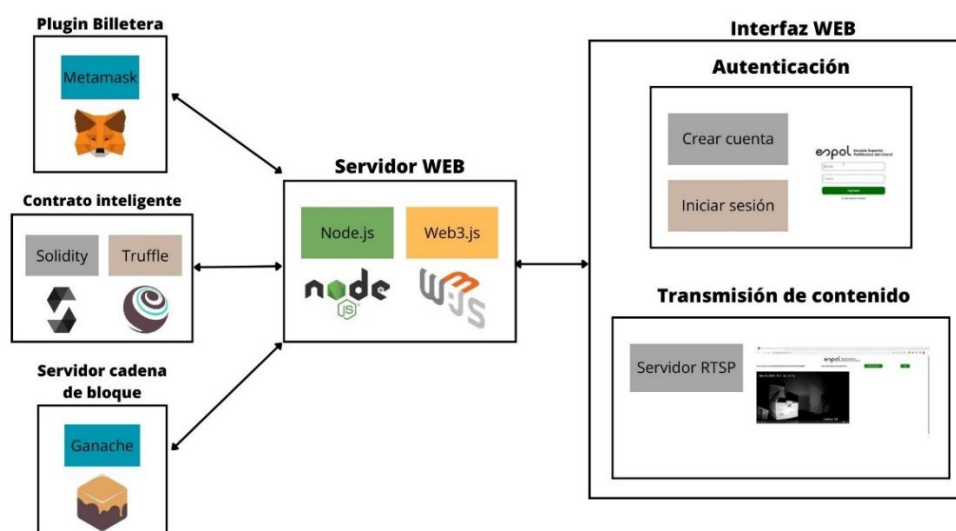


Figura 3.1: Arquitectura del sistema de videovigilancia seguro

- Servidor web: está conformado por nodejs el cual, es el entorno de ejecución de código JavaScript para construir aplicaciones web y web3js, que es la biblioteca de JavaScript que se utiliza para interactuar con la infraestructura de cadena de bloques. En definitiva, son los intermediarios en la comunicación bidireccional entre la interfaz web, la billetera digital y el servidor de la cadena de bloques.

- **Plugin billetera:** está conformado por un navegador y la extensión metamask, con la billetera digital se aprueba o se rechaza transacciones.
- **Contrato inteligente:** está integrado por solidity el cual, es el lenguaje de programación en el que se escriben los contratos inteligentes y truffle, ayuda al despliegue de los contratos inteligentes en la red Ganache.
- **Servidor de cadena de bloques:** lo integra Ganache, el mismo que se comunica de forma bidireccional con la interfaz de web por medio de los contratos inteligentes.
- **Interfaz web:** Es la pieza principal del front-end, está conformado por Autenticación y Transmisión de contenido. En autenticación, permite iniciar sesión o crear una nueva cuenta, su comunicación con el servidor de cadena de bloques o Ganache es mediante web3js. En transmisión de contenido incluye el servidor del protocolo de transmisión en tiempo real (RTSP), su función es transmitir el video capturado por la cámara en la interfaz web. A continuación, en la figura 3.2, se detalla sistemáticamente los pasos a ejecutarse en el funcionamiento del sistema de videovigilancia con un método de autenticación seguro.

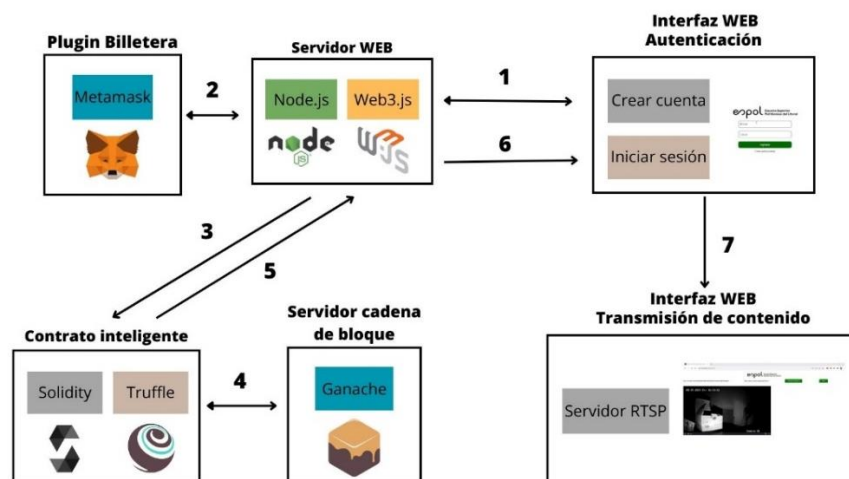


Figura 3.2: Esquema sistemático del sistema de videovigilancia seguro

- Paso 1, al dar clic en “crear cuenta” se establece comunicación bidireccional con el servidor web.
- Paso 2, el servidor web se comunica bidireccionalmente con plugin billetera, donde solicita aprobar o rechazar transacción. El usuario final aprueba la transacción.
- Paso 3, el servidor web envía los datos de la nueva cuenta de usuario hacia un contrato inteligente.
- Paso 4, el contrato inteligente con los datos de la nueva cuenta de usuario es desplegado y distribuido en múltiples bloques en el servidor de cadena de bloques Ganache.
- Paso 5, se elabora un contrato inteligente con los datos de la nueva cuenta y se envía al servidor web para contrastar con la información que ingresa el usuario al iniciar sesión.
- Paso 6, el servidor web valida que la información ingresada por el usuario en la interfaz web para iniciar sesión coincida con los datos del contrato inteligente.
- Paso 7, al dar clic en iniciar sesión se accede al portal de transmisión de contenido o video.

3.3. Diseño del sistema de videovigilancia con un inicio de sesión seguro basado en cadena de bloques.

La base del diseño del proyecto propuesto se basa en una interfaz web de código abierto. Esta interfaz utiliza tecnología de cadena de bloques para autenticar a los usuarios mediante un contrato inteligente. El contrato inteligente actúa como mediador entre la cadena de bloques y la gestión de identidad, y es responsable de supervisar el protocolo del sistema. La estructura general del diseño se compone de seis fases distintas, que se explican en detalle a continuación.

3.3.1. Creación de la cadena de bloques

Se inicia con la creación de una cadena de bloques local de Ganache para desarrolladores, que se instala en una computadora para imitar el nodo Ethereum y simular transacciones con criptomonedas ficticias que

proporcionar Ganache en modo de direcciones disponibles que contienen un número de cuenta y clave privada en hexadecimal. Adicionalmente, se instala la extensión de metamask en cualquier navegador para experimentar trabajar con una billetera digital e interactuar con la cadena de bloques local, ver figura 3.3. Es decir, metamask envía transacciones por medio de contratos inteligentes y Ganache las recibe. Para el uso de la billetera es necesario tener una cuenta e iniciar sesión y verificar que tenga saldo disponible para comenzar a transaccionar.



Figura 3.3:Emulador ganache con metamask[30]

3.3.2. Despliegue del contrato inteligente

Se comienza diseñando el contrato inteligente en solidity, que es un lenguaje de programación y compilador específico para Ethereum y otras plataformas de la cadena de bloques. Se define las variables, funciones y eventos que serán necesarios para la aplicación descentralizada. Para probar el contrato inteligente que va a ser desplegado en la red, utilizamos truffle que facilita la migración en Ganache. En la figura 3.4 se observa la codificación del contrato de autenticación elaborado en solidity en donde tiene definidos los eventos de creación de usuario y una estructura con las variables "username, email y password" para la búsqueda del usuario cuando se inicia sesión.

```
Auth.sol X
src > contracts > Auth.sol
1 // SPDX-License-Identifier: MIT
2 pragma solidity >=0.4.22 <0.9.0;
3
4 contract Auth
5 {
6     uint public userCount = 0;
7
8     mapping(string => user) public userList;
9
10    struct user
11    {
12        string username;
13        string email;
14        string password;
15    }
16
17    // events
18
19    event userCreated(
20        string username,
21        string email,
22        string password
23    );
24
25    function createUser(string memory _username,
26                        string memory _email,
27                        string memory _password) public
28    {
29        userCount++;
30        userList[_email] = user(_username,
31                                _email,
32                                _password);
33        emit userCreated(_username,
34                          _email,
35                          _password);
36    }
37 }
```

Figura 3.4: Estructura del contrato inteligente

3.3.3. Interfaz de Usuario.

Se desarrolla el front-end de la plataforma de videovigilancia con reactjs para construir la interfaz de usuario interactiva y reactiva, nodejs como servidor web y web3js para comunicación con la red Ganache. Además, la interfaz de usuario interactúa con metamask (billetera digital) al momento de crear nueva cuenta o iniciar sesión. La figura 3.5, muestra la página de inicio de sesión de la interfaz web.

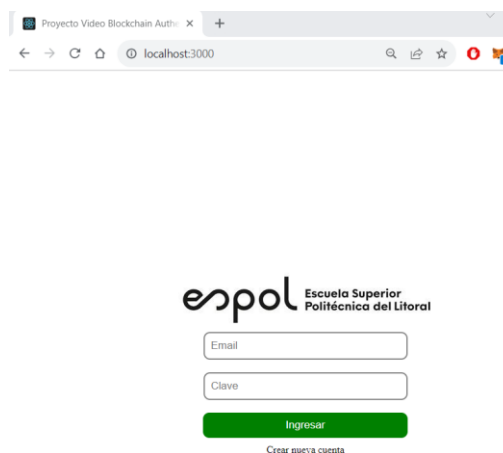


Figura 3.5: Página para inicio de sesión.

3.3.4. Registro de usuario

Para una explicación del proceso de creación de cuenta se realizó un esquema sistematizado y flujograma, tal como se observa en la figura 3.6 y figura 3.7.

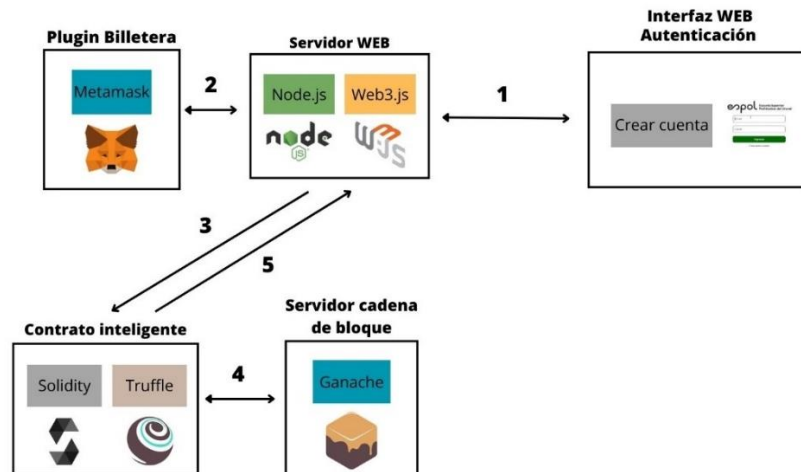


Figura 3.6: Esquema de registro de usuarios.

- Paso 1, el servidor web con la ayuda de nodejs levanta la interfaz web y al dar clic en crear cuenta, se muestra un formulario para el ingreso de usuario, correo y contraseña.
- Paso 2, con la ayuda de web3js, que es la interfaz que se conecta a la red Ethereum, se establece la comunicación con el plugin de metamask (billetera digital), donde se valida si cuenta con saldo Ethereum, luego solicita aprobar o denegar la transacción por un pequeño consumo de gas o ether. El usuario final aprueba la transacción.
- Paso 3, el servidor web envía los datos del usuario y se cargan en un contrato inteligente, elaborado y compilado por solidity,
- Paso 4, el contrato inteligente con los datos de la nueva cuenta de usuario es desplegado y distribuido en múltiples bloques mediante truffle.
- Paso 5, se elabora un contrato inteligente con los datos de la nueva cuenta y se envía al servidor web, al finalizar se mostrará nuevamente la página de inicio para proceder a iniciar sesión.

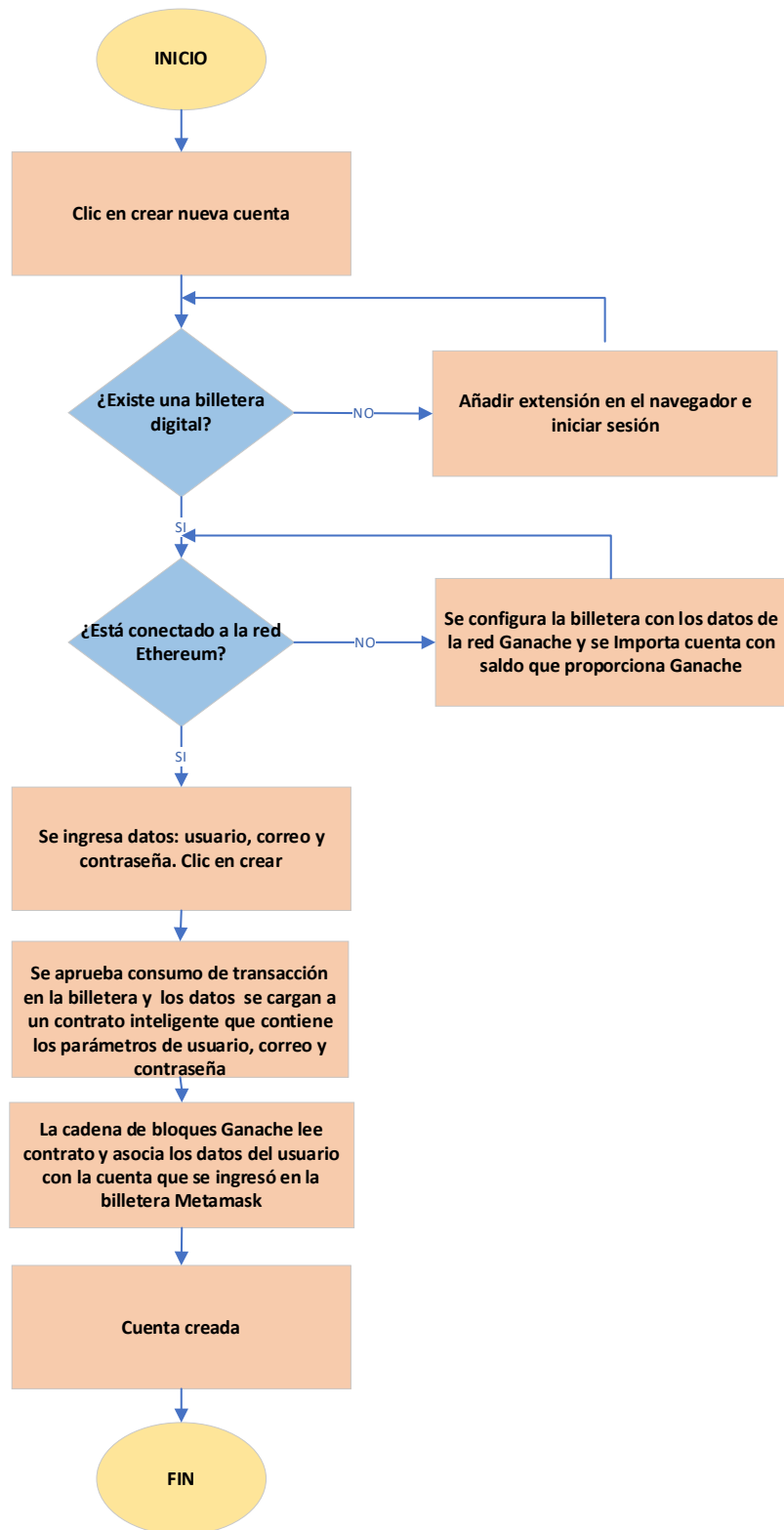


Figura 3.7: Flujograma de registro de usuarios.

3.3.5. Inicio de sesión

Para una explicación detallada del proceso de inicio de sesión se realizó un esquema gráfico sistematizado y flujograma, tal como se muestra en la figura 3.8 y figura 3.9.

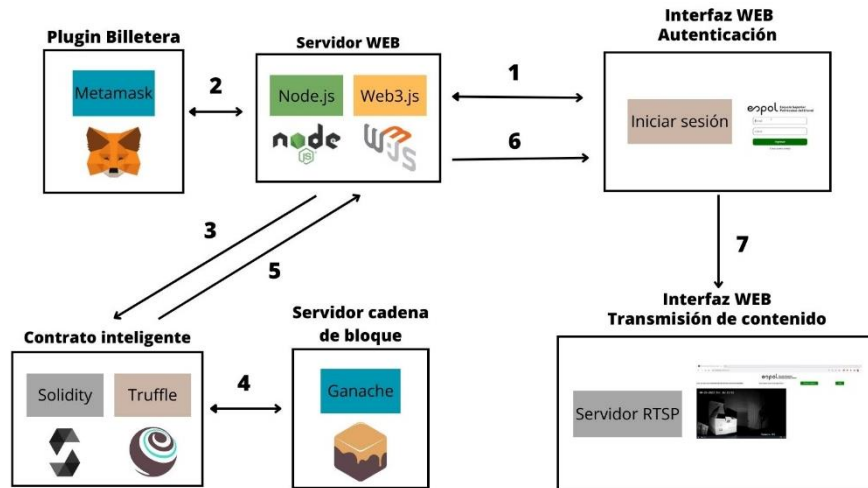


Figura 3.8: Esquema de inicio de sesión.

- Paso 1, el servidor web con la ayuda de nodejs levanta la interfaz web y al dar clic en iniciar sesión, se muestra un formulario para el ingreso de correo y contraseña.
- Paso 2, web3js establece la comunicación con el plugin de metamask (billetera digital), donde hace una validación previa, si cuenta con saldo Ethereum.
- Paso 3, los datos ingresados se cargan en un contrato inteligente elaborado en solidity que contiene los parámetros de usuario y correo, el mismo que es enviado a Ganache mediante truffle.
- Paso 4, el servidor de cadena de bloques Ganache lee el contrato inteligente y valida que coincida en sus registros.
- Paso 5, se elabora un contrato inteligente con los datos legítimos del usuario al servidor web para contrastar con la información que se ingresó el usuario en la interfaz de inicio de sesión.
- Paso 6, el servidor web valida que la información ingresada por el usuario en la interfaz web para iniciar sesión coincida con los datos del contrato inteligente.

- Paso 7, el usuario accede a la plataforma de transmisión de contenido en donde va a poder visualizar el vídeo de las cámaras gracias al servidor del protocolo de transmisión de tiempo real.

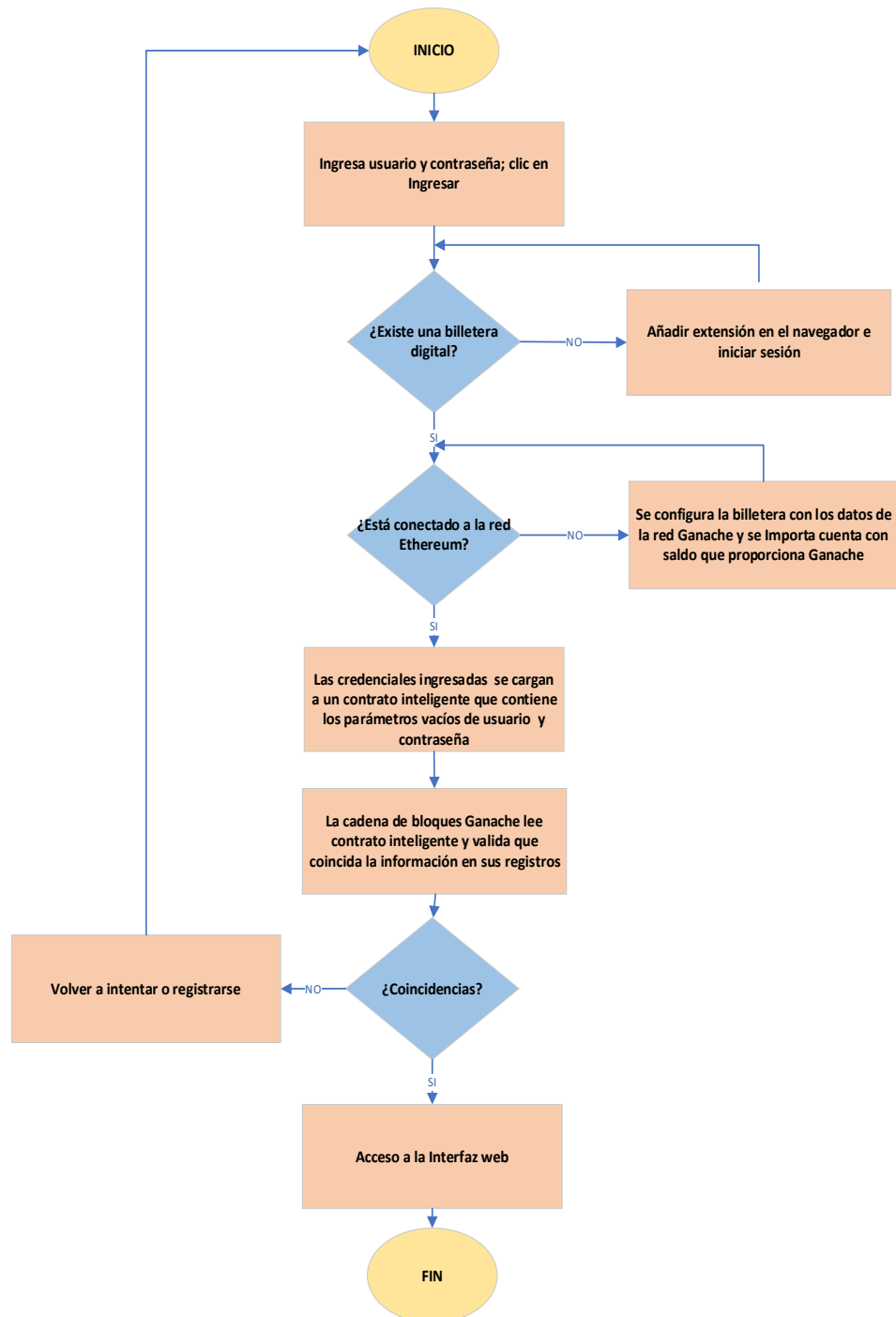


Figura 3.9: Flujograma de inicio de sesión.

3.3.6. Adquisición de video de la cámara.

Inicia con la codificación en GO para levantar un pequeño servidor con el protocolo de transmisión en tiempo real o R.T.S.P, se usó la librería pública RTSPtoWeb con el objetivo de convertir el flujo de video que captura la cámara y permita ser recibido por nuestra interfaz web. Es de indicar que, se utilizó videojs para el manejo del reproductor de video vía HTML5. En la figura 3.10, se observa el archivo “config.json” el cual, es la configuración del servidor, donde se define el puerto predeterminado para la conexión HTTPS, el puerto R.T.S.P y la url donde se visualizará el video.

```
terminal Help config.json - RTSPtoWeb - Visual Studio Code
{} config.json M X
{} config.json > {} server > {} defaults
1  {
2    "server": {
3      "debug": true,
4      "http_debug": false,
5      "http_demo": true,
6      "http_dir": "web",
7      "http_login": "demo",
8      "http_password": "demo",
9      "http_port": ":8083",
10     "https": false,
11     "https_auto_tls": false,
12     "https_auto_tls_name": "",
13     "https_cert": "server.crt",
14     "https_key": "server.key",
15     "https_port": ":443",
16     "ice_servers": ["stun:stun.l.google.com:19302"],
17     "log_level": "debug",
18     "rtsp_port": ":5541",
19     "token": {
20       "backend": "http://127.0.0.1/test.php"
21     },
22     "defaults": {
23       "audio": true
24     }
25   },
26   "streams": {
27     "pattern": {
28       "channels": {
29         "0": {
30           "url": "rtsp://admin:Tesis2023@192.168.100.133:554/Streaming/Channels/01",
31           "debug": false,
32           "audio": true
33         }
34       },
35       "name": "pattern"
36     }
37   }
}
```

Figura 3.10: Código de servidor RTSPtoWeb.

CAPÍTULO 4

4. IMPLEMENTACIÓN Y PRUEBAS DE LA RED DEL SISTEMA DE VIDEOVIGILANCIA

Este capítulo se enfoca en el desarrollo y pruebas del sistema de videovigilancia con un módulo de autenticación seguro usando cadena de bloques. Se describen las especificaciones de los componentes y las tecnologías utilizadas en la solución del prototipo desplegado como prueba de concepto, así como los códigos que se aplican para interactuar con la red de cadena de bloques.

4.1. Integración del sistema de videovigilancia con el método de autenticación

La integración de un sistema de videovigilancia con un método de autenticación proporciona una capa adicional de seguridad al garantizar que solo usuarios autorizados tengan acceso a la aplicación de monitoreo. A continuación, se describe sistemáticamente el desarrollo e integración del Sistema tanto en hardware y software.

4.1.1. Hardware

Se realiza la conexión física de la cámara ip con el cable ethernet y se conecta al puerto LAN del router, con el objetivo que se le asigne una IP a la cámara. Ver figura 4.1.

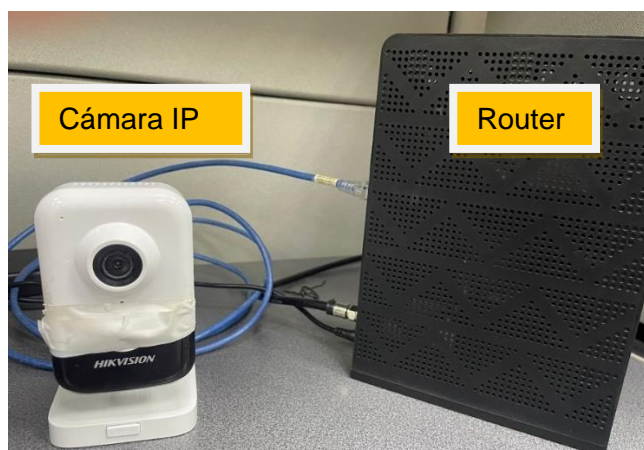


Figura 4.1: Conexión física de la cámara IP Hikvision al puerto LAN del router

4.1.2. Software

A continuación, se explica los pasos a seguir del desarrollo de la aplicación web y el método de autenticación con la utilización de los softwares que intervienen en los mismos.

- Se procede a descargar el instalador de Ganache en una portátil desde la siguiente dirección (<https://trufflesuite.com/ganache/>). Ver figura 4.2.



Figura 4.2: Portal oficial de Ganache[31].

- Luego de esto, se ejecuta el asistente de instalación y se procede a iniciar el servidor. Ver figura 4.3, página inicial del servidor Ganache.

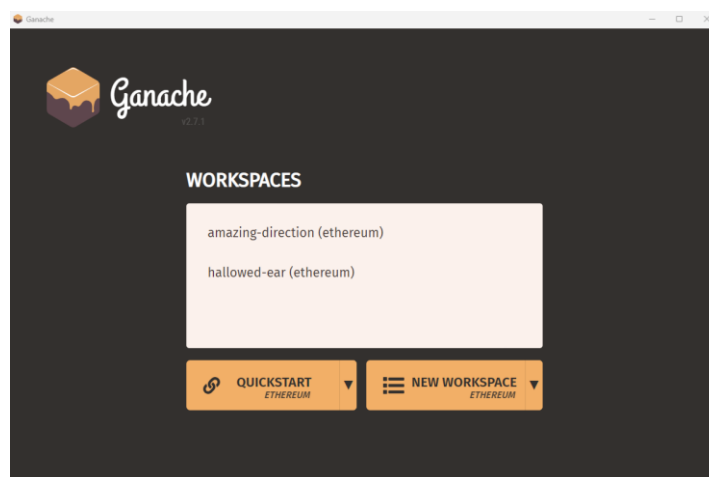


Figura 4.3: Página inicial de Ganache luego de instalación

- Se elige la opción Quickstart (Ethereum) lo cual nos lleva a la siguiente pantalla en la que estarán disponible saldos de Ethereum para pruebas. Ver figura 4.4 de las cuentas disponibles cada una con 100 ETH disponibles.

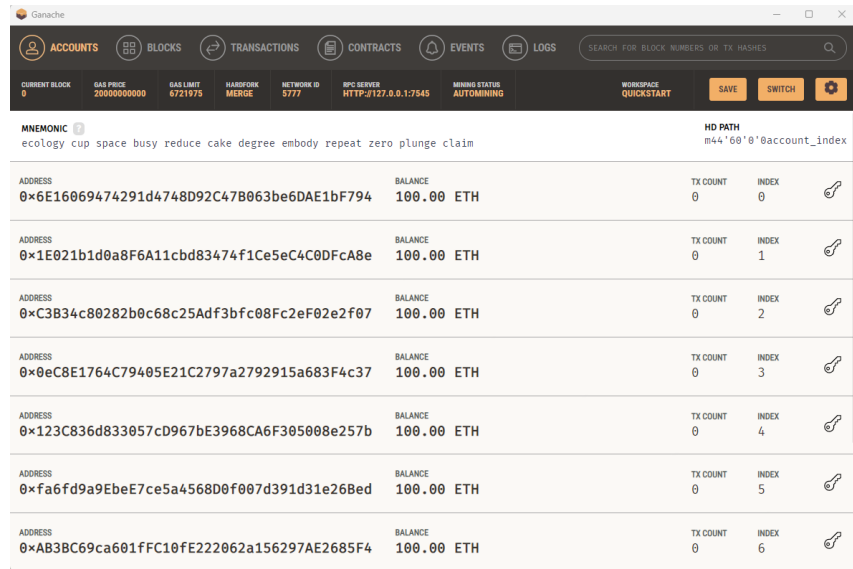


Figura 4.4: Llaves disponibles con 100 ETH de saldo para pruebas.

- De esta pantalla podremos usar alguna de las “address” provistas para agregar saldo para nuestra billetera digital usando los datos de la llave de cada dirección. Ver figura 4.5 donde se muestra la clave privada.

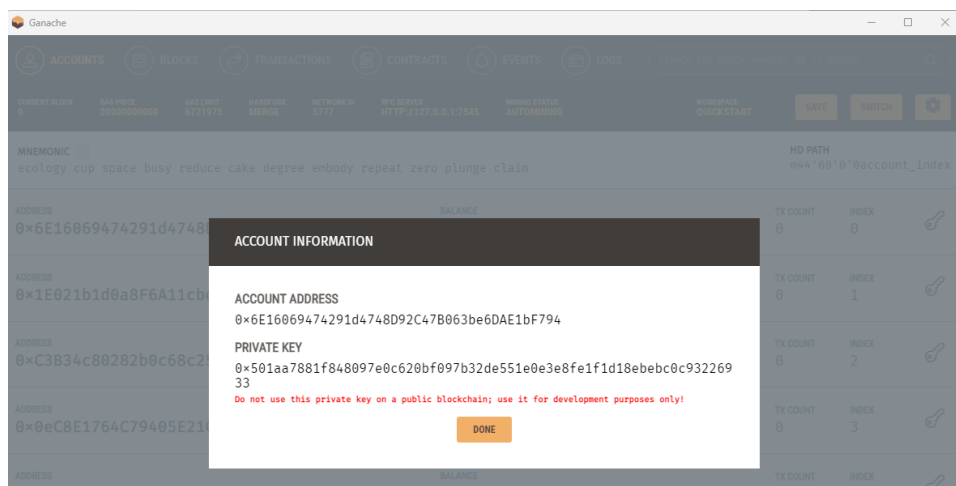


Figura 4.5: Información de la cuenta

- Se procede a instalar la billetera digital de metamask en el navegador Google Chrome, se descarga el plugin o extensión desde la siguiente dirección (<https://metamask.io/>), ver figura 4.6.

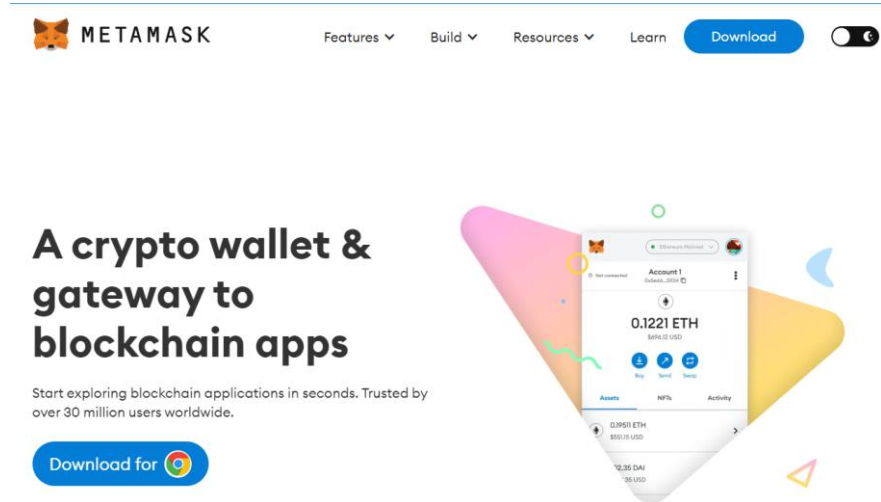


Figura 4.6: Portal oficial de metamask[32].

- Con la extensión instalada y el servidor Ganache iniciado, se procede a iniciar sesión en el plugin. Previamente se crea una cuenta y luego se inicia sesión como aparece en la figura 4.7

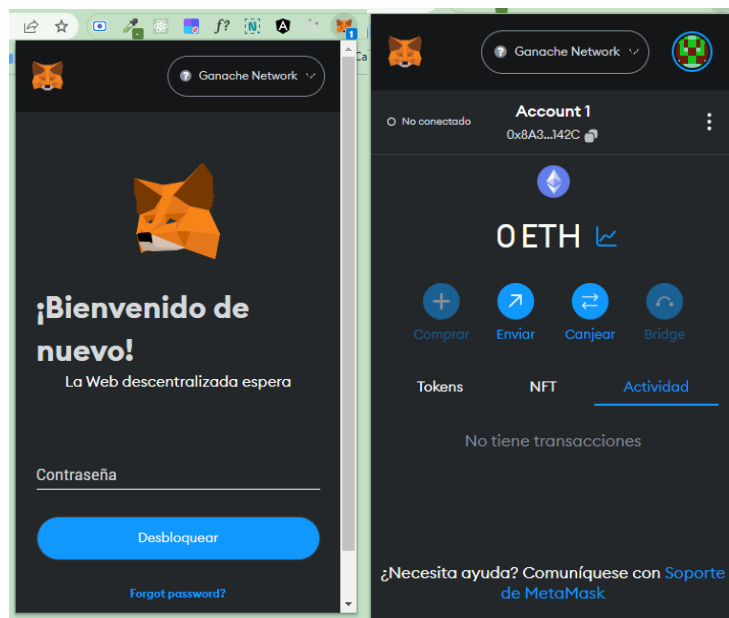


Figura 4.7: Inicio de sesión en metamask

- Se procede a agregar el saldo de Ethereum a nuestra billetera digital usando una llave privada adquirida en el servidor Ganache en los pasos anteriores en el menú perfil opción importar cuenta. Ver figura 4.8.



Figura 4.8: Importación de cuenta Ganache en metamask.

- Posteriormente, ya tendremos cargado saldo de ethereum para la billetera digital como se observa en la figura 4.9 para ejecutar las operaciones via cadena de bloques.

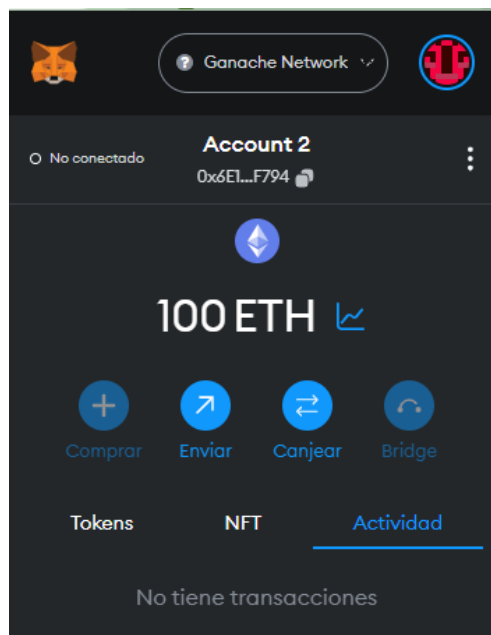


Figura 4.9: Cuenta importada a la red de Ganache

- Se usa la librería web3js en React.js mediante su instalación via N.P.M. La página oficial de la librería para su descarga es (<https://web3js.org/>), ver figura 4.10. Esta librería se usará en nuestra aplicación web para poder comunicarnos con la cadena de bloques.

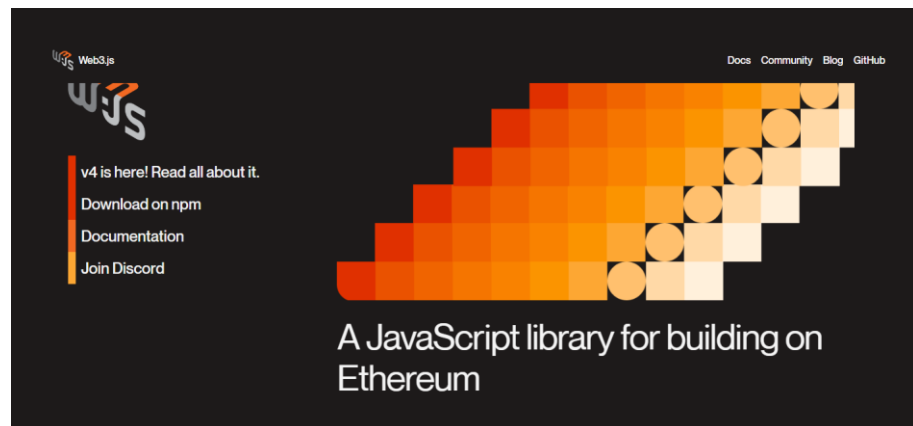


Figura 4.10: Plataforma oficial de Web3js[33].

- Se instala Nodejs (<https://nodejs.org/en>), ver figura 4.11, que es un prerequisite para utilizar React. Se recomienda utilizar el asistente de instalación.



Figura 4.11: Plataforma oficial Node.js [34].

- Se instala WSL (Windows Subsystem for Linux), ver figura 4.12 y Ubuntu en la laptop desde la tienda de Windows, como prerequisite es necesario contar con el S.O Windows 10 o superior.

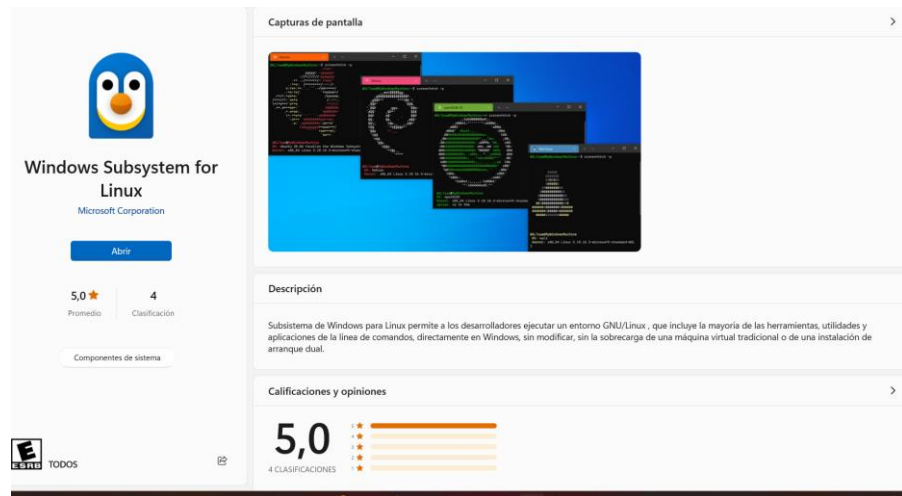


Figura 4.12: Instalación de WLS.

- Activar las características para Subsistema de Linux y Plataforma de máquina virtual, ver figura 4.13. Reiniciar equipo.

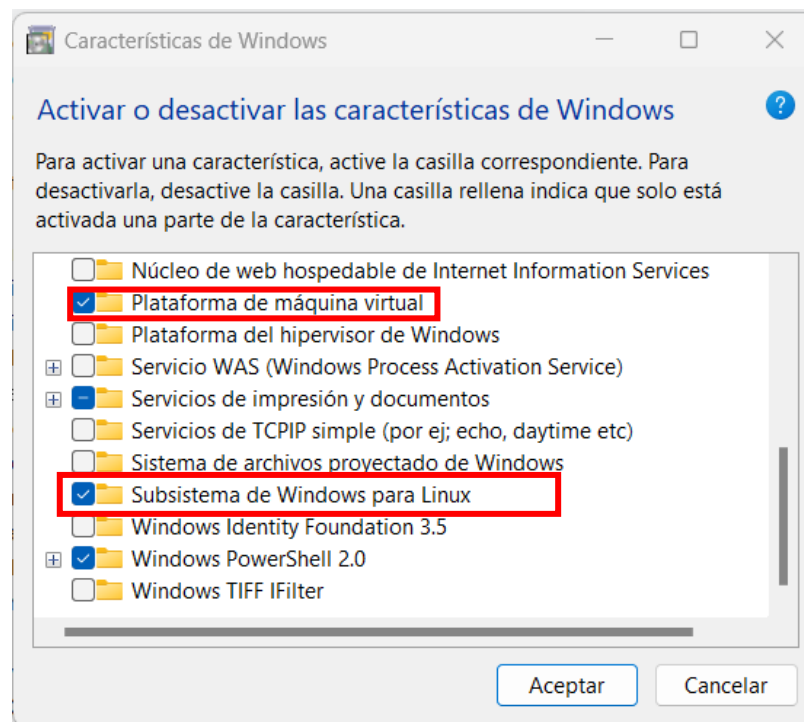


Figura 4.13: Características de Windows.

- Se instala curl en el bash del subsistema Linux (Core Ubuntu) con el siguiente comando: `sudo apt-get install curl`. Ver figura 4.14.

```

Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.90.1-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

This message is shown once a day. To disable it please create the
/home/lnarvaez/.hushlogin file.
lnarvaez@LAPTOP-Q8LEMESL:~$ sudo apt-get install curl

```

Figura 4.14: Instalación de curl en Ubuntu.

- Se instala nvm con los comandos: `curl -o- más el repositorio: https://raw.githubusercontent.com/creationix/nvm/v0.33.0/install.sh|bash` y `nvm install node`. Adicional, se procede a instalar solidity con el comando `npm install -g solc`, ver figura 4.15.

```

Subsistema de Windows para Linux ya está disponible en Microsoft Store.
Puede actualizar ejecutando "wsl.exe --update" o visitando https://aka.ms/wslstorepage
Instalar WSL desde Microsoft Store le proporcionará las últimas actualizaciones de WSL, más rápido.
Para obtener más información, visite https://aka.ms/wslstoreinfo

Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.10.16.3-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

This message is shown once a day. To disable it please create the
/home/npalma/.hushlogin file.
npalma@NPCHP:~$ solcjs --version
0.8.19+commit.7dd6d404.Emscripten.clang
npalma@NPCHP:~$

```

Figura 4.15: Instalación de Solidity

- Se instala el framework truffle en la maquina local mediante el comando: `npm install -g truffle`. En la figura 4.16 y en la figura 4.17, se muestra la verificación de la versión de la librería truffle que tiene instalado.

```
npm install truffle
Microsoft Windows [Versión 10.0.22621.2134]
(c) Microsoft Corporation. Todos los derechos reservados.
C:\Windows\System32>npm install -g truffle
] \ idealTree:npm: 5.11.1 idealTree buildDeps
```

Figura 4.16: Instalación de truffle.

```
Microsoft Windows [Versión 10.0.22621.1848]
(c) Microsoft Corporation. Todos los derechos reservados.
C:\Users\NPC>truffle -v
Truffle v5.8.4 (core: 5.8.4)
Ganache v7.8.0
Solidity v0.5.16 (solc-js)
Node v18.16.0
Web3.js v1.8.2
```

Figura 4.17: Verificación de librerías instaladas.

- Se procede a crear la aplicación de frontend mediante el siguiente comando: `npx create-react-app auth-blockchain`. Una vez terminada la creación de la aplicación se procede a abrir la misma en el visual studio code, ver figura 4.18, se muestra la carpeta Screens que contiene los componentes creados de la aplicación web: `capture.js`, `home.js`, `signin.js` y `signup.js`. Luego de esto se procede a instalar las librerías iniciales necesarias mediante el comando siguiente: `npm install react-router-dom web3`. Para iniciar la interfaz web se lo hace con el comando `npm start`, ver figura 4.19.

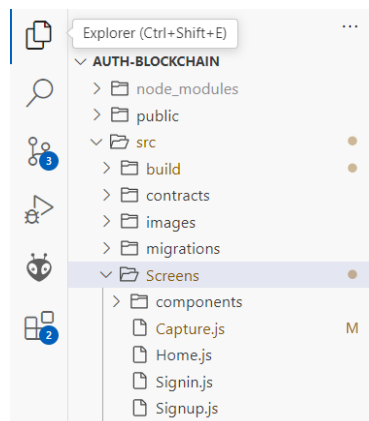


Figura 4.18: Aplicación creada en Visual Studio.

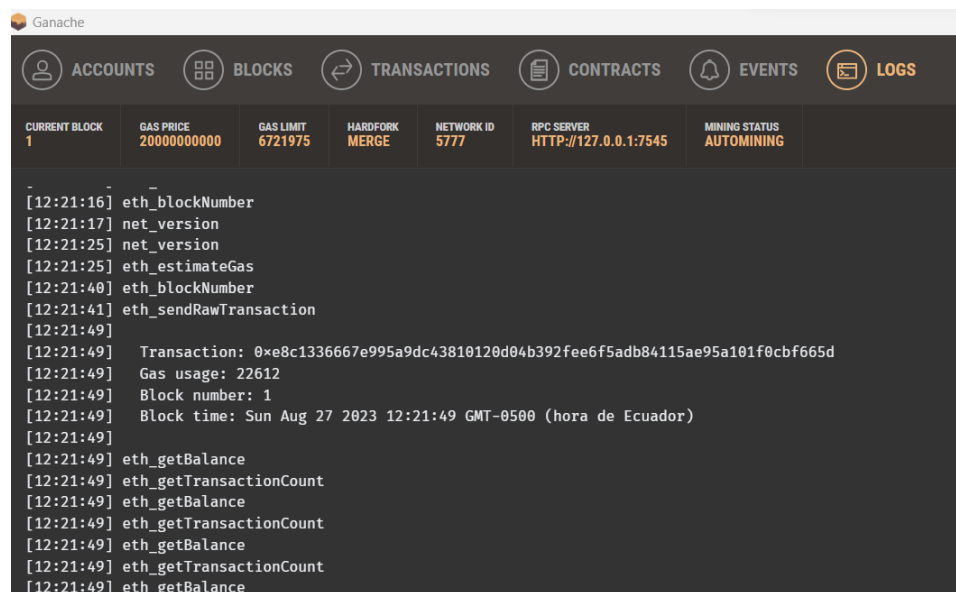
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Microsoft Windows [Versión 10.0.22621.2215]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\TESIS\auth-blockchain>npm start
Compiled with Webpack...
```

Figura 4.19: Se ejecuta npm para iniciar la interfaz.

- Cuando se inicie la interfaz, se procede a dar clic en crear el usuario y se levanta el Metamask (billetera digital) y evidencia que se está intentando realizar una transacción contra la red de cadena de bloques el cual solicita aceptar bajo un pequeño consumo de ether (modena Ethereum). Para validar que existió una comunicación o transacción con la cadena de bloques se puede acceder a los logs del servidor Ganache, ver figura 4.20 y para el caso de un inicio de sesión no es necesario el consumo de gas o eth pero si se refleja esa actividad en los logs de Ganache, ver figura 4.21.



```
Ganache
ACCOUNTS BLOCKS TRANSACTIONS CONTRACTS EVENTS LOGS

CURRENT BLOCK 1
GAS PRICE 20000000000
GAS LIMIT 6721975
HARDFORK MERGE
NETWORK ID 5777
RPC SERVER HTTP://127.0.0.1:7545
MINING STATUS AUTOMINING

[12:21:16] eth_blockNumber
[12:21:17] net_version
[12:21:25] net_version
[12:21:25] eth_estimateGas
[12:21:40] eth_blockNumber
[12:21:41] eth_sendRawTransaction
[12:21:49]
[12:21:49] Transaction: 0xe8c1336667e995a9dc43810120d04b392fee6f5adb84115ae95a101f0cbf665d
[12:21:49] Gas usage: 22612
[12:21:49] Block number: 1
[12:21:49] Block time: Sun Aug 27 2023 12:21:49 GMT-0500 (hora de Ecuador)
[12:21:49]
[12:21:49] eth_getBalance
[12:21:49] eth_getTransactionCount
[12:21:49] eth_getBalance
[12:21:49] eth_getTransactionCount
[12:21:49] eth_getBalance
[12:21:49] eth_getTransactionCount
[12:21:49] eth_getBalance
```

Figura 4.20: Log de creación de usuario en Ganache.

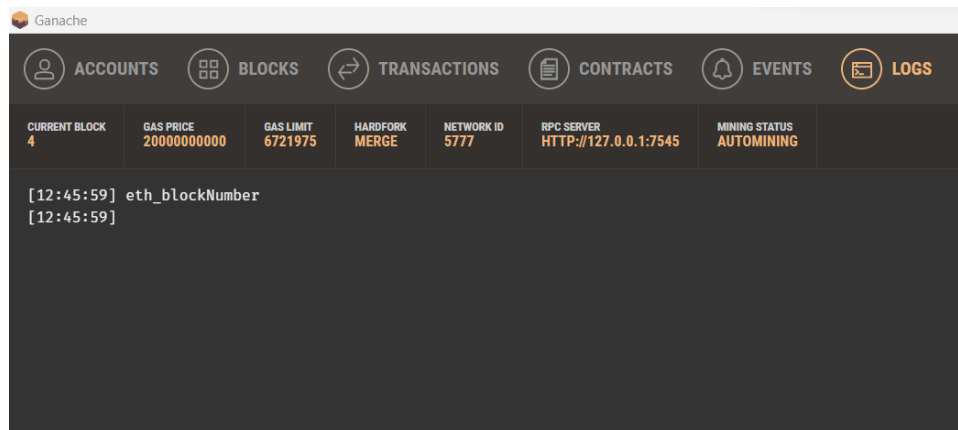


Figura 4.21: Log de inicio de sesión en Ganache.

- A continuación, en la figura 4.22, se muestra el despliegue de la aplicación web con la cual, se interactúa con la red de cadena de bloques local privada. En la figura 4.23 se observa la interacción con la billetera digital metamask, donde solicita rechazar o confirmar transacción y en la figura 4.24, se muestra la plataforma de monitoreo del sistema de videovigilancia.

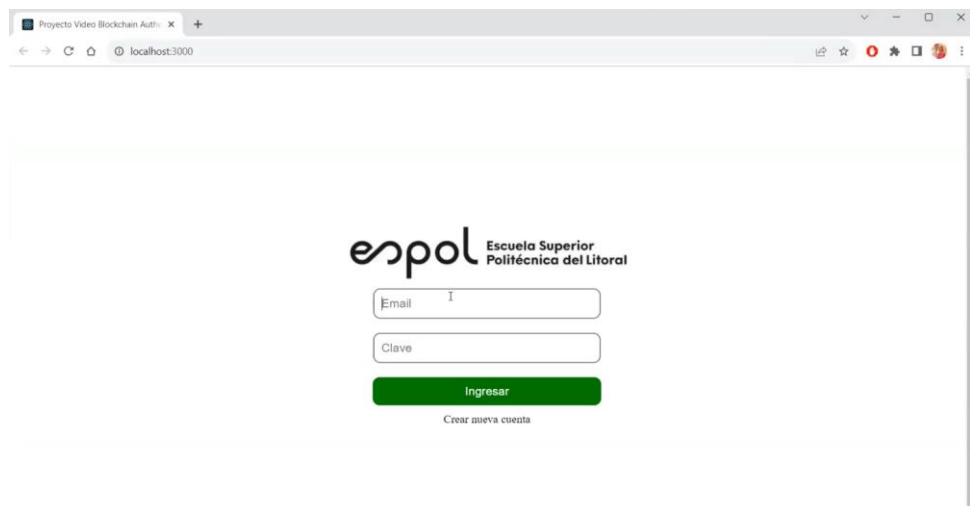


Figura 4.22: Plataforma de videovigilancia para inicio de sesión.

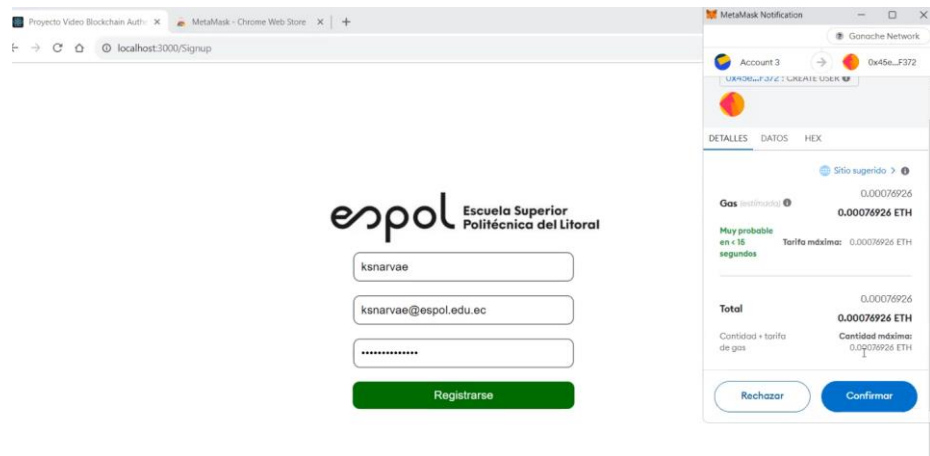


Figura 4.23: Creación de usuario.

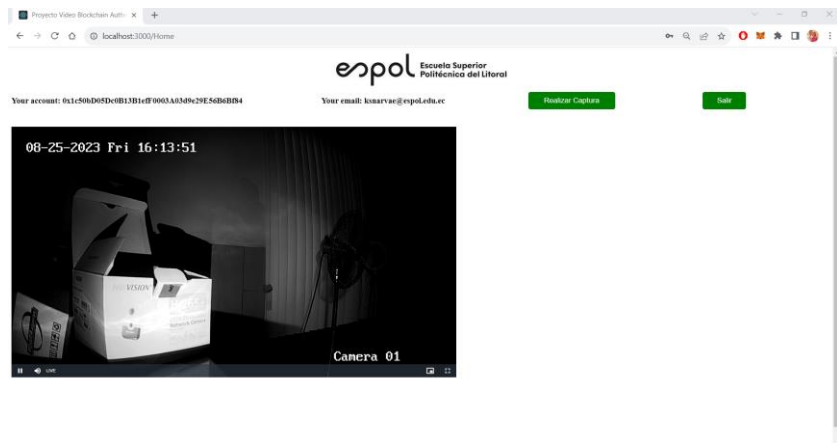


Figura 4.24: Portal de monitoreo del sistema de videovigilancia

- Para la captura de video de la cámara IP se usará el proyecto RSTPToWeb, ver figura 4.25: servidor para convertir video proveniente de la cámara IP mediante el protocolo de transmisión de tiempo real (RTSP) a un formato que puede usarse en la aplicación hecha en React. El proyecto se clonará desde la siguiente url git clone <https://github.com/deepch/RTSPtoWeb>. Se debe modificar la url en el archivo config.json en donde se incluye la dirección ip, usuario y contraseña de la cámara, ver figura 4.26. Se inicia el servidor con el comando “go run.”.

RTSPtoWeb share you ip camera to world!

RTSPtoWeb converts your RTSP streams to formats consumable in a web browser like MSE (Media Source Extensions), WebRTC, or HLS. It's fully native Golang without the use of FFmpeg or GStreamer!

Table of Contents

- [Installation](#)
- [Configuration](#)
- [Command-line](#)
- [API documentation](#)
- [Limitations](#)
- [Performance](#)
- [Authors](#)
- [License](#)

Installation

Figura 4.25: Librería RTSPtoWeb.

```
config.json M x
config.json > {} server > https_port
1  {
2      "server": {
3          "debug": true,
4          "http_debug": false,
5          "http_demo": true,
6          "http_dir": "web",
7          "http_login": "demo",
8          "http_password": "demo",
9          "http_port": ":8083",
10         "https": false,
11         "https_auto_tls": false,
12         "https_auto_tls_name": "",
13         "https_cert": "server.crt",
14         "https_key": "server.key",
15         "https_port": ":443",
16         "ice_servers": ["stun:stun.l.google.com:19302"],
17         "log_level": "debug",
18         "rtsp_port": ":5541",
19         "token": {
20             "backend": "http://127.0.0.1/test.php"
21         },
22         "defaults": {
23             "audio": true
24         }
25     },
26     "streams": {
27         "pattern": {
28             "channels": {
29                 "0": {
30                     "url": "rtsp://admin:Tesis2023@192.168.100.133:554/Streaming/Channels/01",
31                     "debug": false,
32                     "audio": true
33                 },
34             },
35             "name": "pattern"
36         }
37     }
}
```

Figura 4.26: Código del servidor RTSP.

- Se instala el Framework VideoJS en el proyecto RTSPtoWeb para manejo de reproductor de video via html5.

4.2. Pruebas y análisis de resultados del sistema de videovigilancia.

Se idea un plan de pruebas tanto para validar la funcionalidad del método de autenticación y pruebas de seguridad para evaluar la cadena de bloques. A su vez, en esta sección se expone el análisis de resultados de las pruebas de seguridad. Por otra parte, como requisito de pruebas de seguridad del sistema, fue necesaria la instalación del software Owasp zap y Kali Linux, revisar Anexo 2.

4.2.1. Pruebas de funcionalidad del método de autenticación.

En esta sección se realiza dos pruebas de funcionalidad del método de autenticación con el fin del observar el comportamiento de la billetera digital (metamask) ante las siguientes posibles situaciones:

- **Sesión no iniciada en la billetera digital (metamask)**

Cuando se accede a la página principal de la plataforma web y la sesión no está iniciada en el plugin de la billetera digital, aparecerá un error que indica que no se detecta a metamask, ver figura 4.27, puesto que no puede comunicarse con la cadena de bloques y por tal motivo siempre debe tener iniciada sesión con la cuenta del usuario y a su vez tener asociado la dirección de la cadena de bloques para que se le otorgue crédito Ethereum, puesto que sin aquello, no le permitirá crear nuevas cuentas o iniciar sesión.

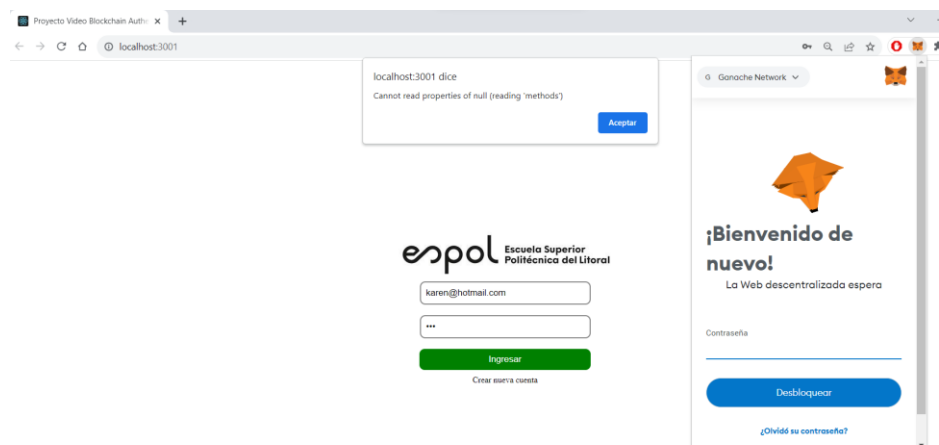


Figura 4.27: Billetera digital no iniciada.

- **Billetera digital sin crédito**

Si la billetera digital metamask no tiene una cuenta con saldo “eth” como se muestra en la figura 4.28, no va a permitir la creación de nuevas cuentas o inicios de sesión. Aparecerá un mensaje que no tiene suficiente saldo para pagar el costo de las transacciones en la red.

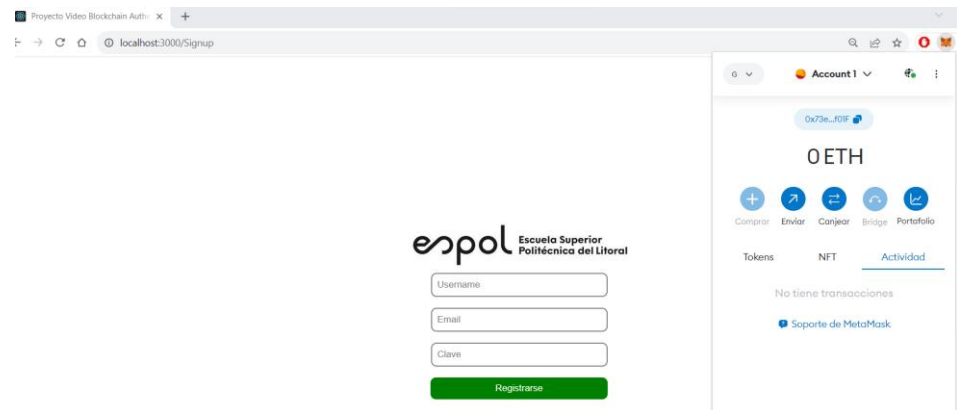


Figura 4.28: Billetera digital sin crédito.

4.2.2. Pruebas de seguridad en la cadena de bloques y análisis de resultados.

Se realizaron tres pruebas de seguridad para evaluar a la cadena de bloques y la misma debe cumplir con tres requisitos de seguridad vitales: confidencialidad, integridad y disponibilidad. La confidencialidad garantiza que sólo los usuarios permitidos puedan participar en el sistema, mientras que la integridad asegura que los mensajes se entreguen a su destino previsto sin ninguna manipulación. La disponibilidad prioriza que los usuarios puedan acceder a los datos que necesitan en cualquier momento.

- **Prueba 1: Escaneo de vulnerabilidades**

Para la identificación del número de vulnerabilidades del Sistema con un módulo de autenticación a nivel de URL se utilizó la herramienta Owasp

Zap con la finalidad de realizar un pentesting y así comprobar la seguridad que garantiza la cadena de bloques.

1. Se selecciona el modo escaneo rápido y se ingresa la URL de nuestra interfaz web, ver figura 4.29, y clic en Atacar. En la figura 4.30 se muestra el resultado del escaneo de vulnerabilidades en la parte inferior derecha en la sección de alertas con una breve descripción de estas.

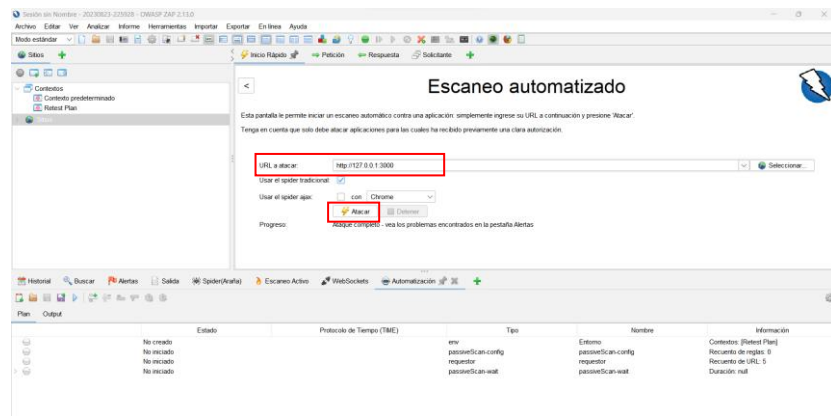


Figura 4.29: Ingreso de URL en Owasp Zap.

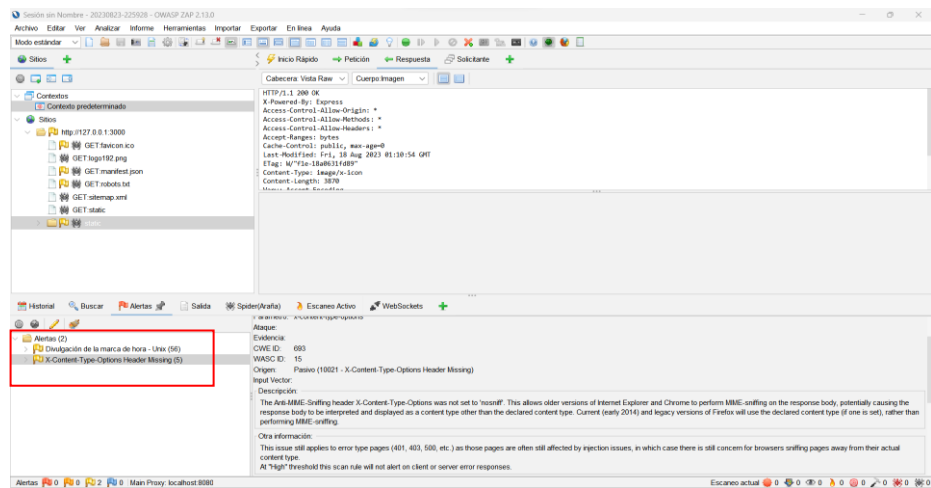


Figura 4.30: Resultado de vulnerabilidades en Owasp Zap.

Resultados: Aplicado el pentesting sobre el Sistema y una vez obtenido las alertas de las vulnerabilidades, se resume en la Tabla 4.1. Esta tabla muestra el número de alertas obtenidas y discriminadas en cada tipo de alerta, junto con el nivel de riesgo del tipo de alerta.

Tabla 4.1. Resumen de resultados obtenidos en el pentesting

Tipo de alerta	Riesgo	Cantidad
Divulgación de la marca de hora - Unix	Bajo	56 (2.800,0 %)
X-Content-Type-Options Header Missing	Bajo	5 (250,0 %)

Análisis e interpretación de resultados: Al integrar la tecnología de cadena de bloques en el método de autenticación podemos concluir que no se muestran vulnerabilidades de Nivel Alto ni Medio. A su vez, no se evidencian alertas de infiltración de información como las credenciales de los usuarios y las 2 únicas alertas con riesgo bajo no afectan a la integridad del sistema ni alteraciones en la cadena de bloques (ver informe en Anexo 1).

- **Prueba 2: Ataque Man-in-the-middle (MIM)**

Un ciberataque conocido como ataque "man-in-the-middle" es un tipo de ataque que tiene lugar dentro de una red de área local. Este acto malicioso implica que un ciberdelincuente se posicione estratégicamente entre dos partes, lo que les permite interceptar y alterar la comunicación entre ellas. Para ejecutar un ataque de intermediario en un sistema de videovigilancia, se intercepta y manipula el enlace de comunicación que conecta la cámara IP y la cadena de bloques Ethereum. Los pasos son los siguiente:

1. Accedemos al símbolo del sistema del equipo donde tenemos instalado el servidor Ganache y obtenemos la IP puerta de enlace mediante el comando "ipconfig" como se muestra en la figura 4.31. En la figura 4.32 se observa la obtención de la IP de la cámara mediante la aplicación celular IP Scanner. En la tabla 4.2 se muestra un resumen del direccionamiento de los dispositivos.

```

Adaptador de Ethernet Ethernet:

Sufijo DNS específico para la conexión. . . :
Vínculo: dirección IPv6 local. . . . . : fe80::6b4d:5efa:967e:c346%22
Dirección IPv4. . . . . : 192.168.100.39
Máscara de subred. . . . . : 255.255.255.0
Puerta de enlace predeterminada. . . . . : 192.168.100.1

```

Figura 4.31: Puerta de enlace de Sistema de videovigilancia y la de cadena de bloques



Figura 4.32: Dirección IP de cámara

Tabla 4.2. Direccionamiento de dispositivos del Sistema de videovigilancia y la red de cadena de bloques.

Dispositivos	Direccionamiento
IP del gateway	192.168.100.1
IP de cámara	192.168.100.8

2. Se inicia la máquina virtual con Kali Linux conectado a la red en la que se encuentra la cámara y la cadena de bloques. Se debe instalar en el sistema operativo Kali Linux el paquete Ettercap con el comando “sudo apt install ettercap-graphical” y para ejecutarlo

escribir el comando **"sudo Ettercap -G"**. En la figura 4.33 se muestra la interfaz principal de Ettercap.

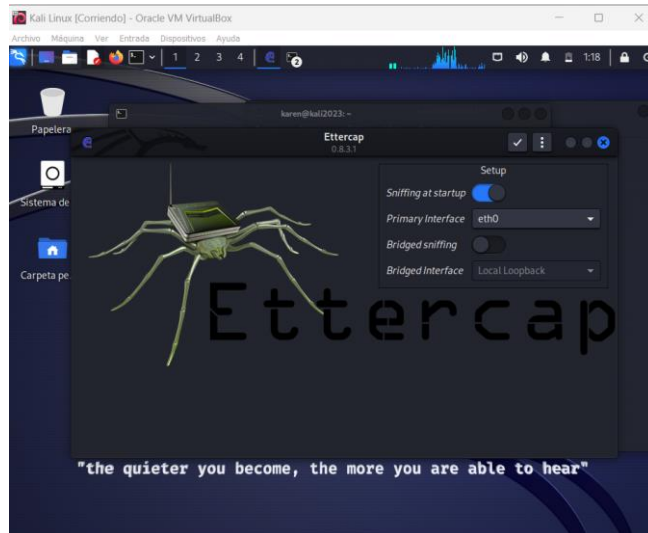


Figura 4.33: Inicio de Ettercap

3. Se selecciona la dirección IP del gateway 192.168.0.1 y se agrega al target 1. Seguido a esto, se selecciona la dirección IP de la cámara IP de destino 192.168.100.8 y se agrega al target 2 como se muestra en la figura 4.34

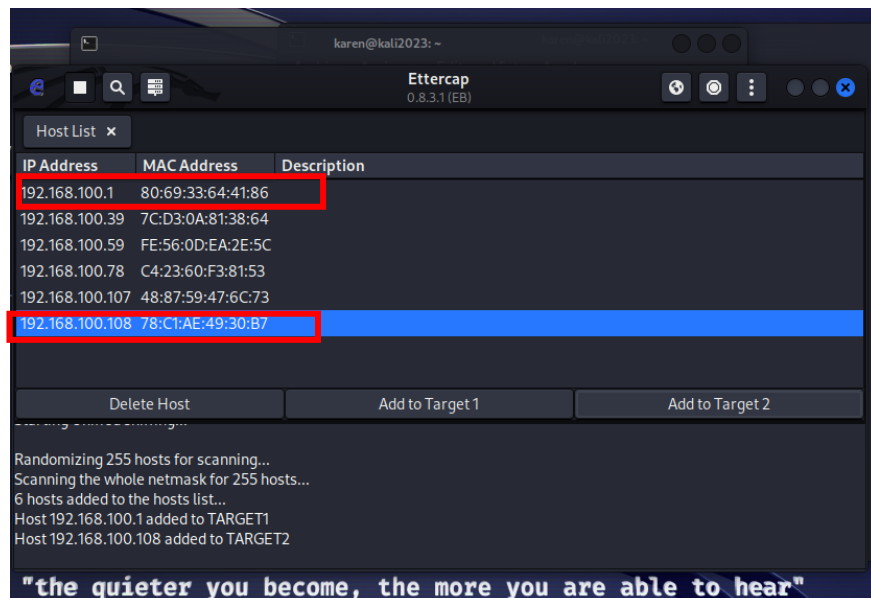


Figura 4.34: Agregando las IPs de los objetivos 1 y 2.

4. Seleccione el menú superior derecho y se hace clic en la pestaña MITM y en el menú desplegable se selecciona "ARP poisoning", tal como se muestra en la figura 4.35.

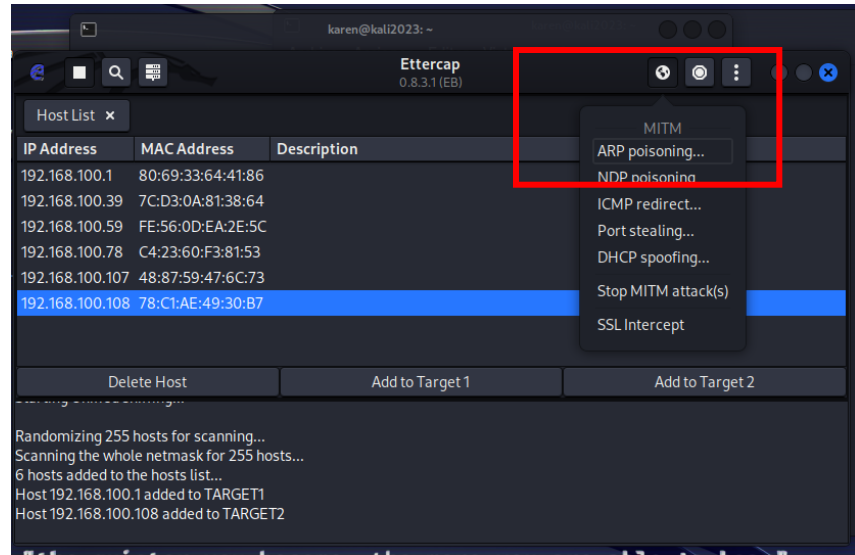


Figura 4.35: ARP Poisoning.

5. Luego de seleccionar ARP Poisoning, se debe seleccionar "Detectar conexiones remotas". Posterior a esto Ettercap comenzará el envenenamiento de ARP, ver figura 4.36.

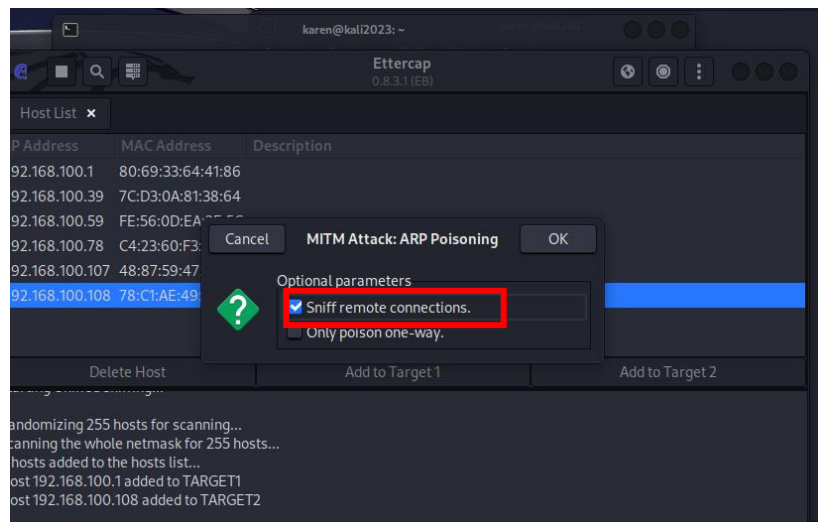


Figura 4.36: La identificación de conexiones remotas.

Resultados: Al usar el aplicativo Ettercap no se recaudó información sensible del sistema de videovigilancia ni de la cadena de bloques como se muestra en la figura 4.37. Adicional a esto, se usó Wireshark para la captura de tráfico de paquetes que se envía desde la cámara hacia la cadena de bloques lo cual, solo se visualizó la huella de la transacción, ver figura 4.38. En los logs de Ganache se comprueba la información de la huella que fue lo único visible por Wireshark, ver imagen 4.39.

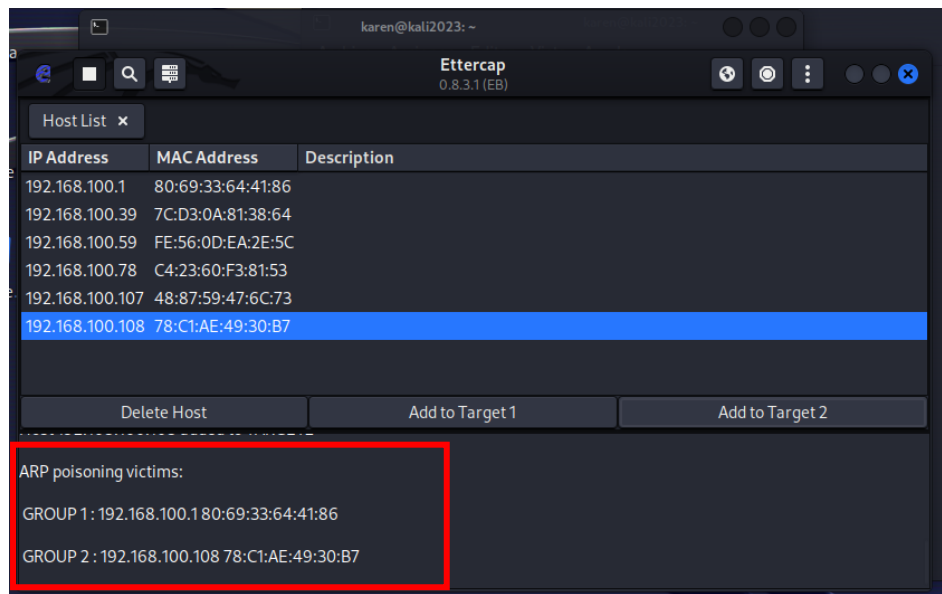


Figura 4.37: Resultados del Ataque MIM.

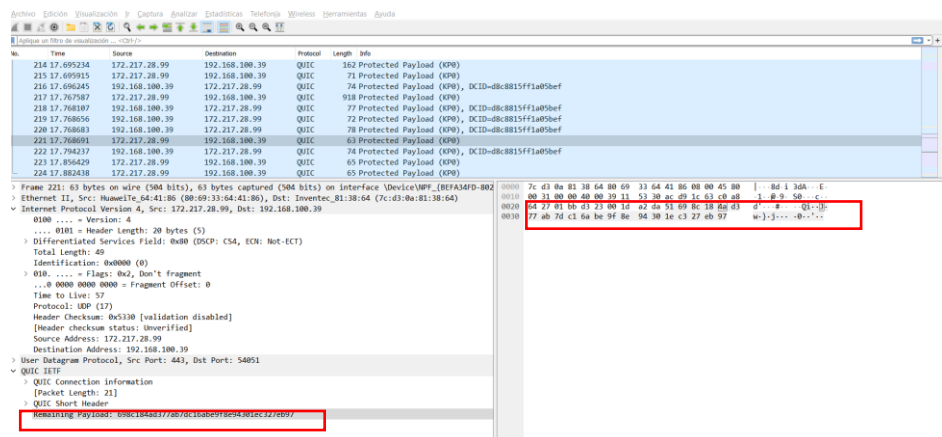


Figura 4.38: Captura de tráfico con Wireshark.

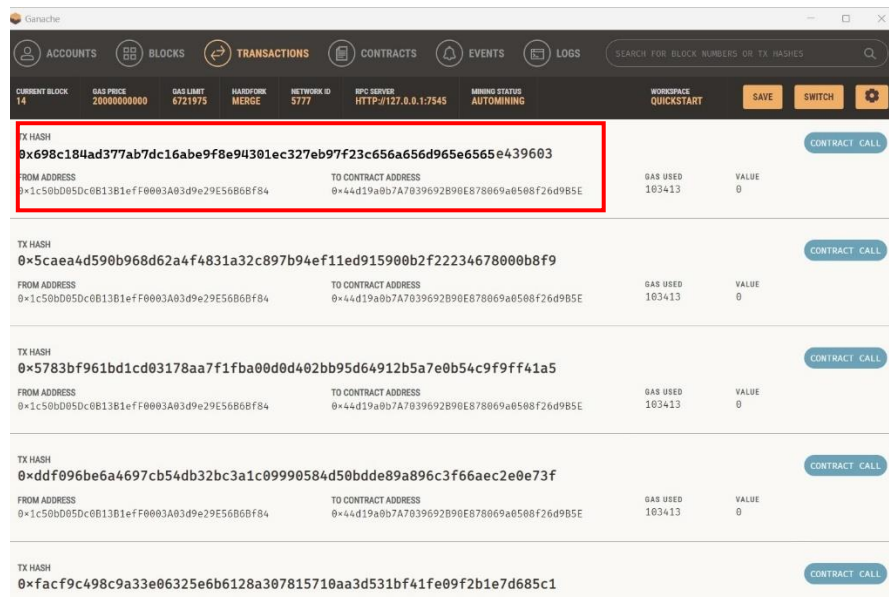


Figura 4.39: Huella de la transacción en los registros de Ganache

Análisis e Interpretación de Resultados: Durante el ataque Man in the Middle (MIM), solo se ve comprometido el hash o huella de la transacción, no los datos de autenticación que el contrato inteligente envía desde la cadena de bloques Ganache. Esto indica que la característica de seguridad de la tecnología de cadena de bloques, la confidencialidad, permanece intacta, si se cumple uno de sus requisitos primordiales

- **Prueba 3: Ataque de denegación de servicio**

Los atacantes a menudo pueden realizar acciones que causan una avalancha de información inútil, lo que lleva a la congestión de la red, en un esfuerzo por impedir que los usuarios autorizados accedan a los servicios de la red. Sin embargo, la estructura descentralizada de cadena de bloques tiene la capacidad de resistir ataques DoS/DDoS. Incluso si ciertos nodos están obstruidos, los atacantes no podrán detener todos los nodos de la red, ya que las transacciones en la cadena de bloques requieren pagos, A continuación, los pasos para realizar este ataque en mención:

1. Abrir la consola en modo administrador y escribir msfconsole, y comenzará con la instalación como se muestra figura 4.40.

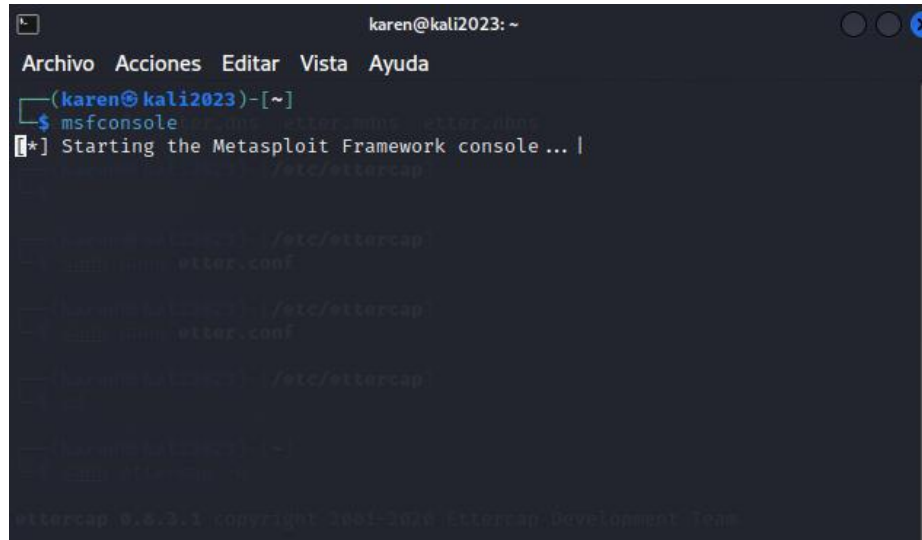


Figura 4.40: Instalación de framework Metasploit

2. Luego de la instalación se ingresa el comando "auxiliary/dos/tcp/synflood", ver imagen 4.41.

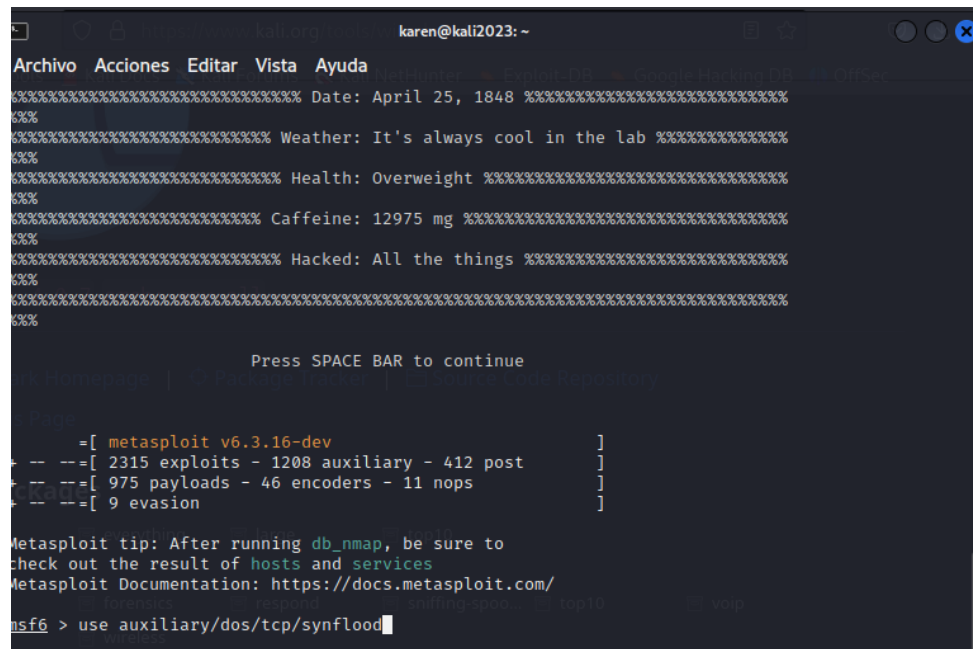


Figura 4.41: Ejecución msf6.

- Se procede a establecer rhost y rport, que es la dirección ip de destino y el número de puerto respectivamente como se muestra en la figura 4.42.

```
msf6 auxiliary(dos/tcp/synflood) > set rport 7545
rport => 7545
msf6 auxiliary(dos/tcp/synflood) > set rhost 192.168.100.39
rhost => 192.168.100.39
msf6 auxiliary(dos/tcp/synflood) >
```

Figura 4.42: Parámetros de red de Ganache

- Para empezar el ataque de denegación de servicio, ingresamos el comando exploit, ver imagen 4.43.

```
msf6 > use auxiliary/dos/tcp/synflood
msf6 auxiliary(dos/tcp/synflood) > set rport 7545
rport => 7545
msf6 auxiliary(dos/tcp/synflood) > set rhost 192.168.100.39
rhost => 192.168.100.39
msf6 auxiliary(dos/tcp/synflood) > exploit
[*] Running module against 192.168.100.39
```

Figura 4.43: Inicio del ataque con exploit

- Se inicia Wireshark en la computadora destino para verificar cuántos paquetes son capturados como llegada, ver imagen 4.44.

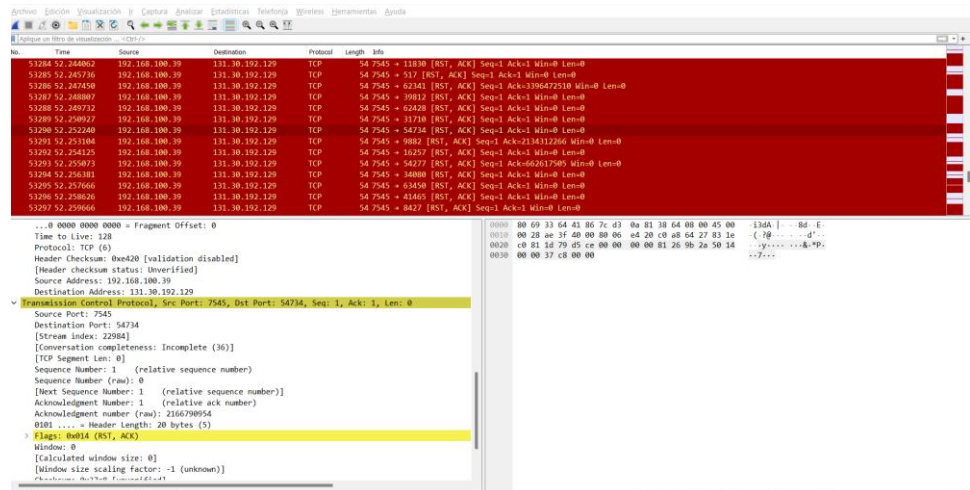


Figura 4.44: Paquetes recibos en el equipo destino por el exploit.

Resultados: En la captura de tráfico que realiza Wireshark, se visualiza un total de 53297 paquetes que fueron generados luego del ataque de denegación de servicio, ver imagen 4.45.

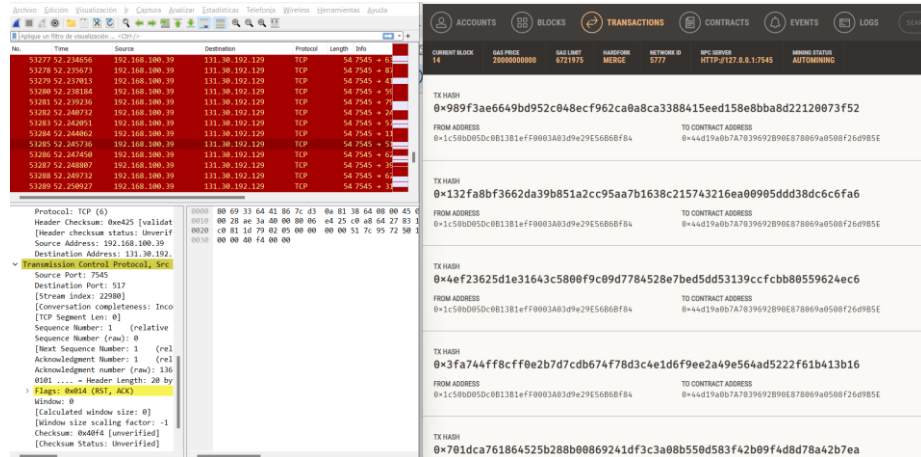


Figura 4.45: Registro de transacciones de Ganache en la ejecución de exploit

Análisis e Interpretación de Resultados: En el momento que ocurría el ataque de denegación de servicio, la cadena de bloques Ganache continuaba realizando transacciones ininterrumpidamente sin presentar latencias al momento de iniciar sesión o registrar nuevos usuarios, gracias a su concepto de que es descentralizada distribuida de múltiples nodos. Aún si el hacker utiliza herramientas robustas para denegación de servicio y logre caer un nodo Ethereum, solo denegará el servicio en ese nodo mas no se infiltrará para obtener información puesto que para eso necesitará la dirección y la clave privada de Ganache, las credenciales de la billetera digital y del acceso a la plataforma. En este escenario comprobamos 2 requisitos más del concepto de cadena de bloques, disponibilidad e inmutabilidad

CAPÍTULO 5

5. CONCLUSIONES Y RECOMENDACIONES

Finalizado las pruebas y validaciones del sistema de videovigilancia con un método de autenticación basado en cadena de bloques, el mismo que fue realizado como prueba de concepto, se concluye y se recomienda lo siguiente.

5.1. CONCLUSIONES

- A través del estudio de la cadena de bloques realizado para la creación de este proyecto titulación, no solo descubrimos una solución que capitaliza los beneficios únicos que ofrece esta tecnología, sino que también hemos aportado evidencia sólida de que esta tecnología puede ser utilizada de manera efectiva para abordar problemas de seguridad como es el caso de robo de información sensible.
- Al utilizar el contrato inteligente integrado en la cadena de bloques Ethereum, es posible regular las políticas de control de acceso, así como monitorear el almacenamiento de datos y la gestión del flujo. Esta innovadora herramienta permite establecer una normativa específica que regule el acceso al Sistema de Videovigilancia.
- En el corazón del sistema de verificación se encuentra una cadena de bloques que funciona como un mecanismo de back-end, garantizando el flujo fluido de las transacciones entre los usuarios y la interfaz web. Esto se logra mediante la validación y verificación de los medios, que culmina con la creación de un bloque de transacciones que se agrega a la cadena de bloques después de lograr un consenso en toda la red. El bloque en sí se convierte en una alteración imborrable de la red y se difunde a otros nodos. Este cambio es definitivo y no se puede replicar, garantizando así que solo se conceda acceso a los usuarios permitidos.
- Nuestro sistema de seguridad con el método de autenticación en cadena de bloques superó con éxito el plan de pruebas de seguridad realizadas en el capítulo 4, donde se concluye que la funcionalidad de la huella y la criptografía fortalece la seguridad del sistema.

5.2. RECOMENDACIONES

- Para la instalación de solidity, que es el compilador de contrato inteligentes, se sugiere contar con un sistema operativo Windows 11 para evitar inconvenientes de características de Windows deshabilitadas.
- Dado que la cadena de bloques es una tecnología nueva, actualmente no existen marcos de referencia ni estándares establecidos que regulen el desarrollo de aplicaciones descentralizadas. Por lo tanto, se recomienda buscar a profesionales expertos para incorporarlo como solución empresarial.
- Cada vez que se apaga o se reinicie el dispositivo donde se tiene instalado Ganache es necesario reiniciar la librería truffle para los contratos inteligentes con los siguientes comandos: `truffle build` y `truffle migrate --reset`.
- Para mejorar la protección y control del acceso a otras aplicaciones, un enfoque es utilizar tecnología de cadena de bloques y contratos inteligentes para la autenticación, puesto que se ha evidenciado que con el uso de cadena de bloques le ofrece un nivel más alto de seguridad y ayuda a mitigar filtración de información sensible.

BIBLIOGRAFÍA

- [1] Y. Pinto and L. Peña, "Revisión de la evolución de criptomonedas y su incursión en algunos países de América Latina," 2023. Accessed: Jul. 28, 2023. [Online]. Available: https://repositorio.unbosque.edu.co/bitstream/handle/20.500.12495/10906/PENA_Stephanie_PINTO_Carolina%20Trabajo%20de%20investigaci%c3%b3n%20de%20final.pdf?sequence=1&isAllowed=y
- [2] D. O. Sánchez, "Deployment of Blockchain technology in Serverless infrastructure," UNIVERSIDAD COMPLUTENSE DE MADRID, Madrid, 2019. Accessed: Aug. 18, 2023. [Online]. Available: <https://docta.ucm.es/rest/api/core/bitstreams/72873c26-b6fc-4249-a291-b1048dbc19c1/content>
- [3] WISENET, "Crecimiento acelerado en cámaras de seguridad tipo DIY | Ventas de Seguridad," 2020. <https://www.ventasdeseguridad.com/2020031011925/noticias/empresas/crecimiento-acelerado-en-camaras-de-seguridad-tipo-diy.html> (accessed Mar. 19, 2023).
- [4] Matt Zand and Rajneesh Gupta, "How two-factor authentication works with blockchain | 2021-02-01 | Security Magazine," 2021. <https://www.securitymagazine.com/articles/94479-how-two-factor-authentication-works-with-blockchain> (accessed Mar. 19, 2023).
- [5] Net and El Hacker, "Blog elhacker.NET: Utilizan 25.000 cámaras de videovigilancia para lanzar ataques DDoS," 2019. <https://blog.elhacker.net/2016/06/utilizan-25.mil-camaras-de-cctv-videovigilancia-para-lanzar-ataques-DDoS-botnet.html> (accessed Mar. 24, 2023).
- [6] JOSÉ MANUEL ROMERO and El País, "La cámara de Interior que vigilaba el chalet de Iglesias y Montero fue pirateada | Política | EL PAÍS," Apr. 09, 2019.

https://elpais.com/politica/2019/04/07/actualidad/1554663804_480848.html
(accessed Mar. 24, 2023).

- [7] BBC News Mundo, “La advertencia de que se están hackeando las cámaras de vigilancia de los bebés (y cómo puedes protegerte) - BBC News Mundo,” Mar. 03, 2020. <https://www.bbc.com/mundo/noticias-51681204> (accessed Aug. 18, 2023).
- [8] EL TIEMPO, “Hackeo de cámaras de seguridad expuso a Tesla y a otras empresas - Novedades Tecnología - Tecnología - ELTIEMPO.COM,” Mar. 10, 2021. <https://www.eltiempo.com/tecnosfera/novedades-tecnologia/hackeo-de-cameras-de-seguridad-expuso-a-tesla-y-a-otras-empresas-572336> (accessed Feb. 22, 2023).
- [9] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, “On blockchain and its integration with IoT. Challenges and opportunities,” *Future Generation Computer Systems*, vol. 88, pp. 173–190, 2018, doi: 10.1016/j.future.2018.05.046.
- [10] Z. Gong-Guo and Z. Wan, “Blockchain-based IoT security authentication system,” in *Proceedings - 2021 International Conference on Computer, Blockchain and Financial Development, CBFDF 2021*, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 415–418. doi: 10.1109/CBFDF52659.2021.00090.
- [11] E. Maribeth and A. Mena, “Implementación De Un Sistema De Video Vigilancia Para Los Exteriores De La Ups, Mediante Mini Computadores Y Cámaras Raspberry Pi,” Universidad Politécnica Salesiana, Guayaquil, 2018. Accessed: Aug. 17, 2023. [Online]. Available: <https://dspace.ups.edu.ec/bitstream/123456789/10379/1/UPS-GT001404.pdf>
- [12] José Miguel Castillo Castillo, “Sistemas de videovigilancia y CCTV,” pp. 1–50, 2019, Accessed: Aug. 17, 2023. [Online]. Available: <http://myelectronic.byethost10.com/Mis%20proyectos/Sistemas%20de%20vid%20evigilancia%20y%20CCTV.PDF?i=1>

- [13] D. Sanchez, “¿Cámara IP vs Cámara Análoga? Conoce las Diferencias,” Sep. 20, 2020. <https://info.ita.tech/blog/camara-ip-vs-analogica-diferencias> (accessed Aug. 17, 2023).
- [14] E. A. Shammar, A. T. Zahary, and A. A. Al-Shargabi, “A Survey of IoT and Blockchain Integration: Security Perspective,” *IEEE Access*, vol. 9, pp. 156114–156150, Nov. 2021, doi: 10.1109/ACCESS.2021.3129697.
- [15] D. Puthal, N. Malik, S. P. Mohanty, E. Kougianos, and C. Yang, “The Blockchain as a Decentralized Security Framework,” *IEEE Consumer Electronics Magazine*, vol. 7, no. 2, pp. 18–21, Mar. 2018, doi: 10.1109/MCE.2017.2776459.
- [16] K. Wust and A. Gervais, “Do you need a blockchain?,” in *Proceedings - 2018 Crypto Valley Conference on Blockchain Technology, CVCBT 2018*, Institute of Electrical and Electronics Engineers Inc., Nov. 2018, pp. 45–54. doi: 10.1109/CVCBT.2018.00011.
- [17] Fedor Iván Cambiazo Alvear, “Oportunidades de aplicación de tecnología Blockchain en micro redes,” UNIVERSIDAD DE CHILE, Santiago de Chile, 2021. Accessed: Aug. 18, 2023. [Online]. Available: <https://repositorio.uchile.cl/bitstream/handle/2250/182764/Oportunidades-de-aplicacion-de-tecnologia-Blockchain-en-micro-redes.pdf?sequence=1>
- [18] H. Si, C. Sun, Y. Li, H. Qiao, and L. Shi, “IoT information sharing security mechanism based on blockchain technology,” *Future Generation Computer Systems*, vol. 101, pp. 1028–1040, Dec. 2019, doi: 10.1016/J.FUTURE.2019.07.036.
- [19] O. Jean Pierre and D. Auri, *MANUAL DE BLOCKCHAIN*, Cedice Futuro. Caracas, 2022. Accessed: Aug. 18, 2023. [Online]. Available: <https://libreriacedice.org.ve/wp-content/uploads/2022/08/Manual-de-Blockchain-CEDICE.pdf>
- [20] Josué López, CEO/CSO Auditech, and Block-Auth, “Autenticación Blockchain vs Contraseñas tradicionales - Auditech.” <https://auditech.es/blog/blockchain->

- vs-contrasenas-una-nueva-era-en-la-seguridad-de-autenticacion/ (accessed Aug. 18, 2023).
- [21] AWS, “¿Cómo funciona la tecnología de cadena de bloques?,” 2020. <https://aws.amazon.com/es/what-is/blockchain/?aws-products-all.sort-by=item.additionalFields.productNameLowercase&aws-products-all.sort-order=asc> (accessed Aug. 18, 2023).
- [22] J. Sousa, A. Bessani, and M. Vukolic, “A byzantine Fault-Tolerant ordering service for the hyperledger fabric blockchain platform,” in *Proceedings - 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2018*, Institute of Electrical and Electronics Engineers Inc., Jul. 2018, pp. 51–58. doi: 10.1109/DSN.2018.00018.
- [23] R. Singh and N. Bansal, “Ethereum Decentralized peer-to-peer Contracts Mechanism,” in *2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, IEEE, May 2023, pp. 1909–1912. doi: 10.1109/ICACITE57410.2023.10183181.
- [24] Ethereum, “¿Qué es Ethereum? | ethereum.org,” Aug. 17, 2023. <https://ethereum.org/es/what-is-ethereum/> (accessed Aug. 18, 2023).
- [25] “Ganache | Overview - Truffle Suite.” <https://trufflesuite.com/docs/ganache/> (accessed Aug. 18, 2023).
- [26] Blockchain Council, “Ganache Blockchain: All you need to Know - Blockchain Council,” May 18, 2023. <https://www.blockchain-council.org/blockchain/ganache-blockchain-all-you-need-to-know/> (accessed Aug. 18, 2023).
- [27] J. Maldonado, “Qué es Hyperledger Fabric, la blockchain de las empresas,” 2023. <https://observatorioblockchain.com/blockchain/que-es-hyperledger-fabric-la-blockchain-de-las-empresas/> (accessed Aug. 18, 2023).
- [28] SoluLab, “Hyperledger Fabric On Blockchain technology: What are the Advantages and Disadvantages? | by SoluLab | Medium,” Jan. 29, 2020. <https://solulab.medium.com/hyperledger-fabric-on-blockchain-technology->

- what-are-the-advantages-and-disadvantages-43fffc6a27fe (accessed Aug. 18, 2023).
- [29] Criptotario, “Ventajas Y Desventajas De Ethereum,” 2020. <https://criptotario.com/ventajas-desventajas-ethereum> (accessed Aug. 18, 2023).
- [30] Farhan Khan, “Logo Ganache y Metamask,” Jan. 18, 2019. https://media.licdn.com/dms/image/C4D12AQHMatPDpLjwKA/article-cover_image-shrink_423_752/0/1547586410680?e=1698278400&v=beta&t=6nXzK10Oi2GnqSo5he9yVxcoh5mFVUfR4_gcK9zXcgY (accessed Aug. 19, 2023).
- [31] “Ganache - Truffle Suite.” <https://trufflesuite.com/ganache/> (accessed Aug. 28, 2023).
- [32] “MetaMask.” <https://metamask.io/> (accessed Aug. 28, 2023).
- [33] “Web3.js — Javascript Ethereum API.” <https://web3js.org/> (accessed Aug. 28, 2023).
- [34] “Node.js.” <https://nodejs.org/en> (accessed Aug. 28, 2023).

ANEXOS

Anexo 1: Informe del Escaneo de vulnerabilidades con Owap zap

24/8/23, 10:59

ZAP Informes de Escaneo

Contextos

No se seleccionó ningún contexto, por lo que todos los contextos se incluyeron de forma predeterminada.

Sitios

Se incluyeron los siguientes sitios:

- <http://127.0.0.1:3000>

(Si no se seleccionó ningún sitio, todos los sitios se incluyeron de manera predeterminada).

Un sitio incluido también debe estar dentro de uno de los contextos incluidos para que sus datos se incluyan en el informe.

Niveles de riesgo

Incluido : [Alto](#) , [Medio](#) , [Bajo](#) , [Informativo](#)

Excluido : [Ninguno](#)

Niveles de confianza

Incluido : [Confirmado por Usuario](#) , [Alta](#) , [Media](#) , [Baja](#)

Excluidos : [Confirmado por Usuario](#) , [Alta](#) , [Media](#) , [Baja](#) , [Falso positivo](#)

Resúmenes

Recuentos de alertas por riesgo y confianza

Esta tabla muestra el número de alertas para cada nivel de riesgo y confianza incluido en el informe.

(Los porcentajes entre paréntesis representan el recuento como porcentaje del número total de alertas incluidas en el informe, redondeado a un decimal).

Confianza

	Confirmado por Usuario	Medios de alta comunicación		Baja	Total
Alto	0 (0,0 %)	0 (0,0 %)	0 (0,0 %)	0 (0,0 %)	0 (0,0 %)
Medio	0 (0,0 %)	0 (0,0 %)	0 (0,0 %)	0 (0,0 %)	0 (0,0 %)
bajos	0 (0,0 %)	0 (0,0 %)	1 (50,0 %)	1 (50,0 %)	2 (100,0 %)
Informativo	0 (0,0 %)	0 (0,0 %)	0 (0,0 %)	0 (0,0 %)	0 (0,0 %)
Total	0 (0,0 %)	0 (0,0 %)	1 (50,0 %)	1 (50,0 %)	2 (100%)

Recuentos de alertas por sitio y riesgo

Esta tabla muestra, para cada sitio para el que se generaron una o más alertas, la cantidad de alertas generadas en cada nivel de riesgo.

Las alertas con un nivel de confianza de "Falso positivo" se han excluido de estos recuentos.

(Los números entre paréntesis son el número de alertas emitidas para el sitio o por encima de ese nivel de riesgo).

	alto (= alto)	Medio (>= Medio)	Bajo (>= Informativo)	Informativo
http://127.0.0.1:3000	0	0	2	0
Sitio	(0)	(0)	(2)	(2)

Recuentos de alertas por tipo de alerta

Esta tabla muestra el número de alertas de cada tipo de alerta, junto con el nivel de riesgo del tipo de alerta.

(Los porcentajes entre paréntesis representan cada recuento como un porcentaje, redondeado a un decimal, del número total de alertas incluidas en este informe).

Tipo de alerta	Riesgo	Contar
Divulgación de la marca de hora - Unix	bajos	56 (2.800,0 %)
Falta el encabezado X-Content-Type-Options	bajos	5 (250,0 %)
Total		2

Alertas

Riesgo = Bajo , Confianza = Media (1)

<http://127.0.0.1:3000> (1)

[Falta el encabezado X-Content-Type-Options \(1\)](#)

▼ OBTENER <http://127.0.0.1:3000/favicon.ico>

Etiquetas de alerta	<ul style="list-style-type: none"> ▪ OWASP 2021 A05 ▪ OWASP 2017 A06
Descripción de la alerta	El encabezado Anti-MIME-Sniffing X-Content-Type-Options no estaba configurado en 'nosniff'. Esto permite que las versiones anteriores de Internet Explorer y Chrome realicen un rastreo MIME en el cuerpo de la

que el cuerpo de la respuesta se interprete y se muestre como un tipo de contenido distinto del tipo de contenido declarado. Las versiones actuales (principios de 2014) y heredadas de Firefox utilizarán el tipo de contenido declarado (si está configurado), en lugar de realizar un rastreo MIME.

Otra información

Este problema todavía se aplica a las páginas de tipo de error (401, 403, 500, etc.), ya que esas páginas a menudo todavía se ven afectadas por problemas de inyección, en cuyo caso todavía existe la preocupación de que los navegadores detecten páginas que no corresponden a su tipo de contenido real.

En el umbral "Alto", esta regla de análisis no alertará sobre respuestas de error del cliente o del servidor.

En el umbral "Alto", esta regla de análisis no alertará sobre respuestas de error del cliente o del servidor.

Pedido

▼ Línea de solicitud y sección de encabezado (272 bytes)

```
GET
http://127.0.0.1:3000/favicon.ico
HTTP/1.1
host: 127.0.0.1:3000
user-agent: Mozilla/5.0 (Windows
NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/114.0.0.0
Safari/537.36
pragma: no-cache
cache-control: no-cache
referer: http://127.0.0.1:3000
```

▼ Cuerpo de solicitud (0 bytes)

Respuesta

▼ Línea de estado y sección de encabezado (426 bytes)

```
HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: *
Access-Control-Allow-Headers: *
Accept-Ranges: bytes
Cache-Control: public, max-age=0
Last-Modified: Fri, 18 Aug 2023
01:10:54 GMT
ETag: W/"f1e-18a0631fd89"
Content-Type: image/x-icon
Content-Length: 3870
Vary: Accept-Encoding
Date: Thu, 24 Aug 2023 04:02:29 GMT
Connection: keep-alive
Keep-Alive: timeout=5
```

► Cuerpo de respuesta (3870 bytes)

Parámetro

x-content-type-options

Solución

Asegúrese de que la aplicación/servidor web establezca el encabezado de tipo de contenido correctamente y que establezca el encabezado X-Content-Type-Options en 'nosniff' para todas las páginas web.

Si es posible, asegúrese de que el usuario final utilice un navegador web moderno y compatible con los estándares que no realice ningún tipo de detección de MIME, o que la aplicación web o el servidor web puedan indicarle que no realice la detección de MIME.

Riesgo = Bajo , Confianza = Baja (1)

<http://127.0.0.1:3000> (1)

Divulgación de la marca de hora - Unix (1)

▼ OBTENER <http://127.0.0.1:3000/static/js/bundle.js>

Etiquetas de alerta

- [OWASP 2021 A01](#)
- [OWASP 2017 A03](#)

Descripción de la alerta

Una marca de tiempo ha sido divulgada por el servidor de la aplicación/el navegador - Unix

Otra información

1950641247, que evalúa: 2031-10-24 15:47:27

Pedido

▼ Línea de solicitud y sección de encabezado (280 bytes)

```
GET
http://127.0.0.1:3000/static/js/bundle.js HTTP/1.1
host: 127.0.0.1:3000
user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.0.0 Safari/537.36
pragma: no-cache
cache-control: no-cache
referer: http://127.0.0.1:3000
```

▼ Cuerpo de solicitud (0 bytes)

Respuesta

▼ Línea de estado y sección de

Anexo 2: Preparación de escenarios para pruebas de seguridad con KALI LINUX

1. Se carga la imagen ISO de Kali Linux en la máquina virtual e iniciamos el equipo. Seleccionamos la opción "Graphical install", como se observa en la figura A2.1

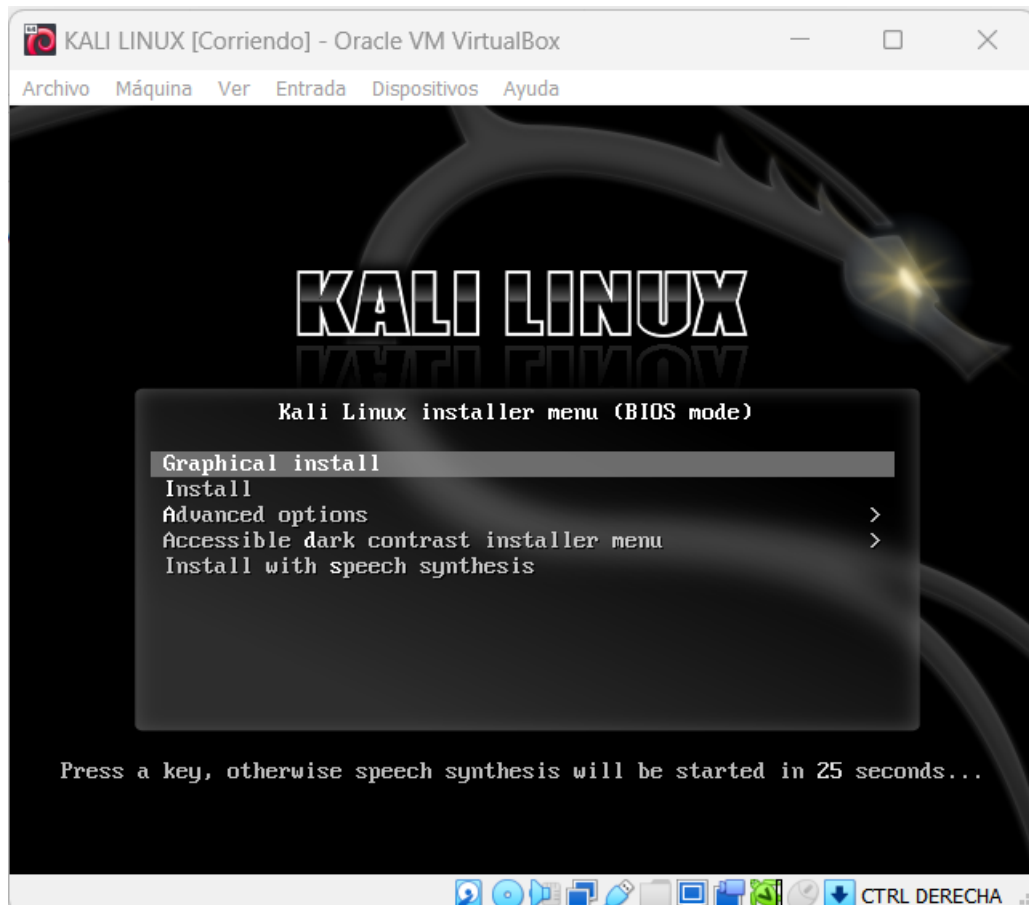


Figura A2.1. Iniciamos el equipo con la imagen ISO de Kali Linux

2. Se selecciona el idioma de la instalación, ver figura A2.2:

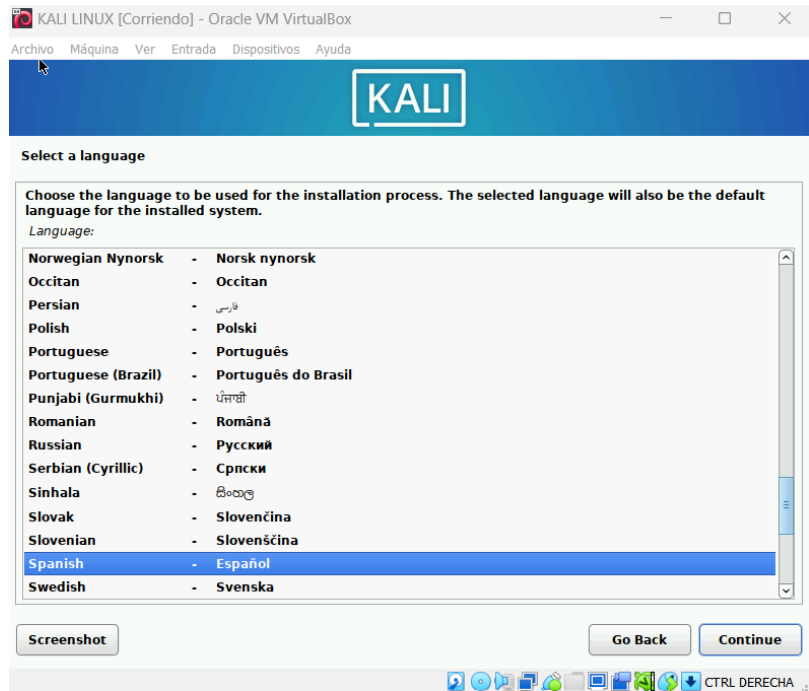


Figura A2.2 Seleccionamos idioma español

3. Se define la ubicación, ver figura A2.3:



Figura A2.3 Seleccionamos ubicación.

4. Clic en Continuar para que se carguen componentes como se muestra en la figura A2.4 y en la figura A2.5:

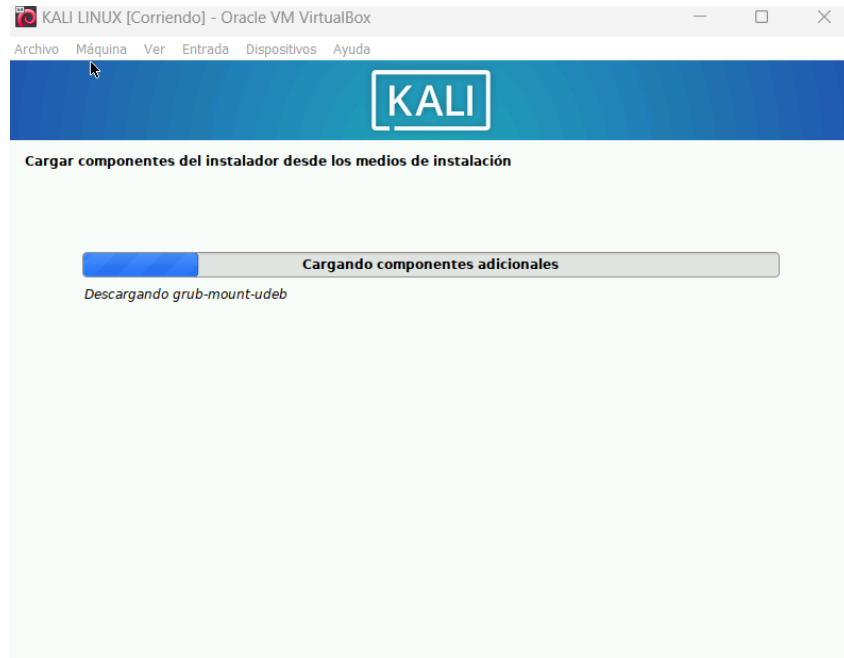


Figura A2.4. Cargando componentes adicionales

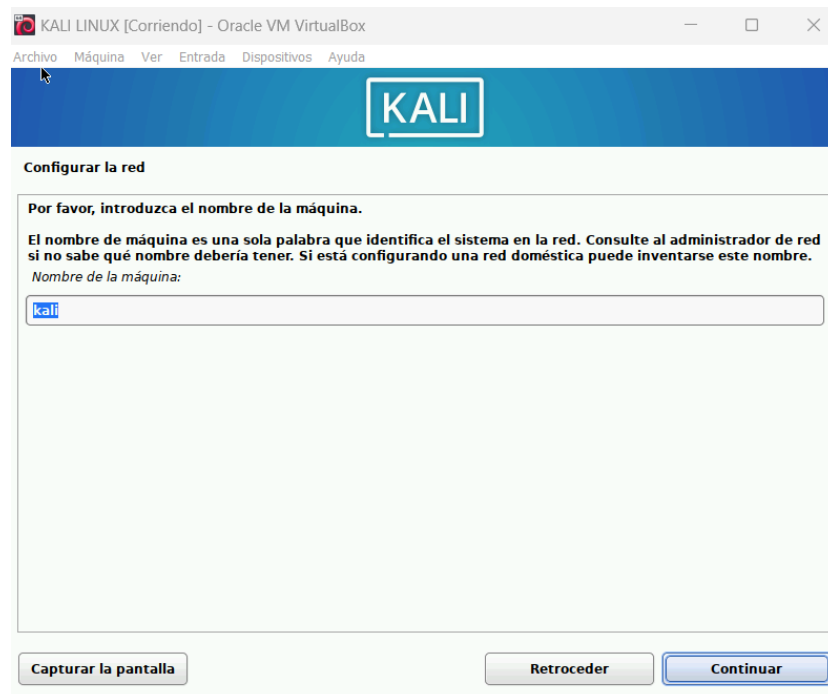


Figura A2.5. Ingreso de nombre de la red.

5. Se define el dominio, ver figura A2.6:

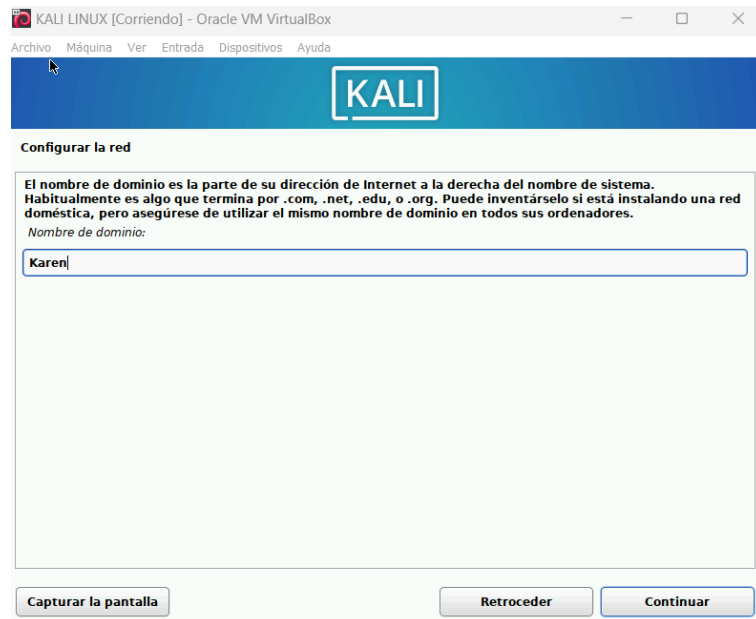
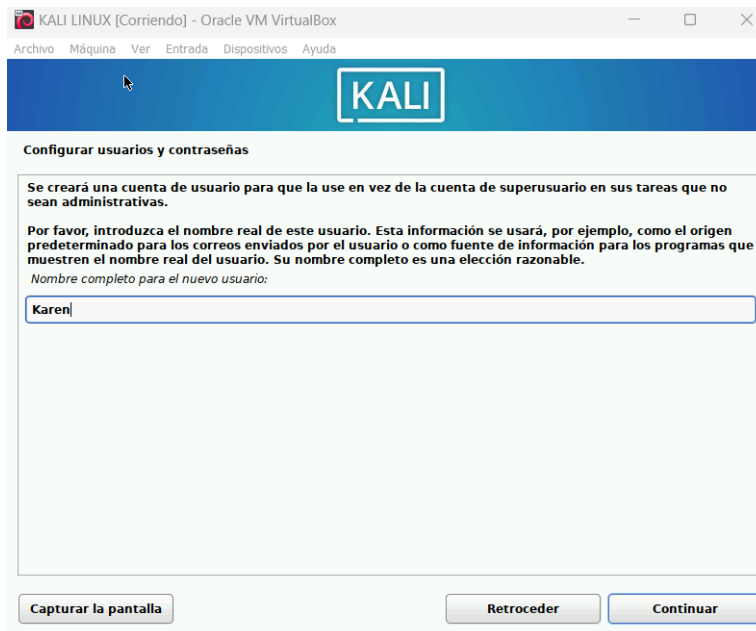


Figura A2.6. Se configura nombre de dominio

6. Clic en Continuar para ingresar el nombre de usuario, ver figura A2.7. Se ingresa la contraseña, tal como se observa en la figura A2.8 y se selecciona la zona horaria, ver figura A2.9



7.

Figura A2.7. Configuración de usuario

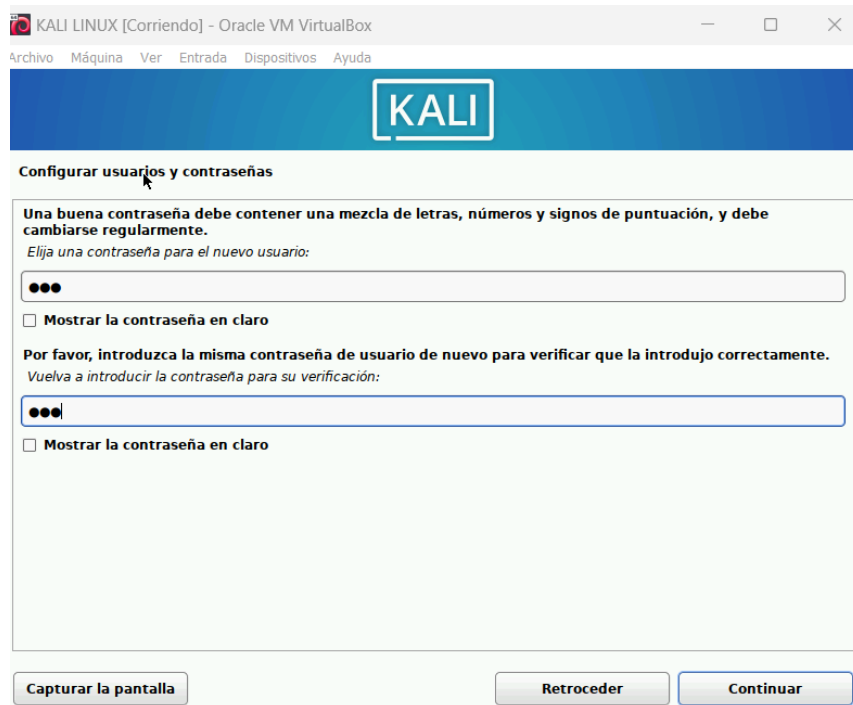


Figura A2.8. Configuración de contraseña

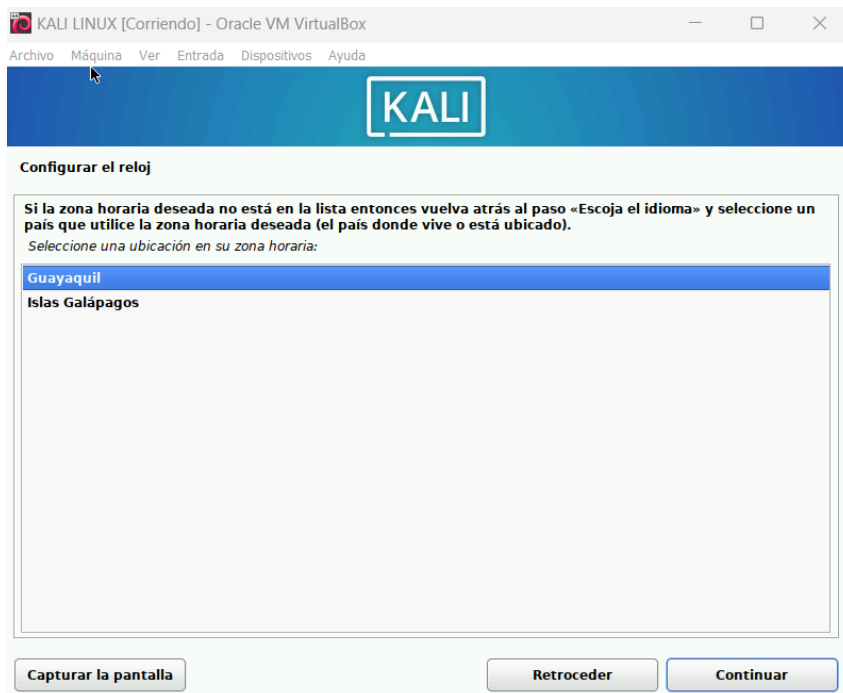


Figura A2.9. Configuración de Zona Horaria

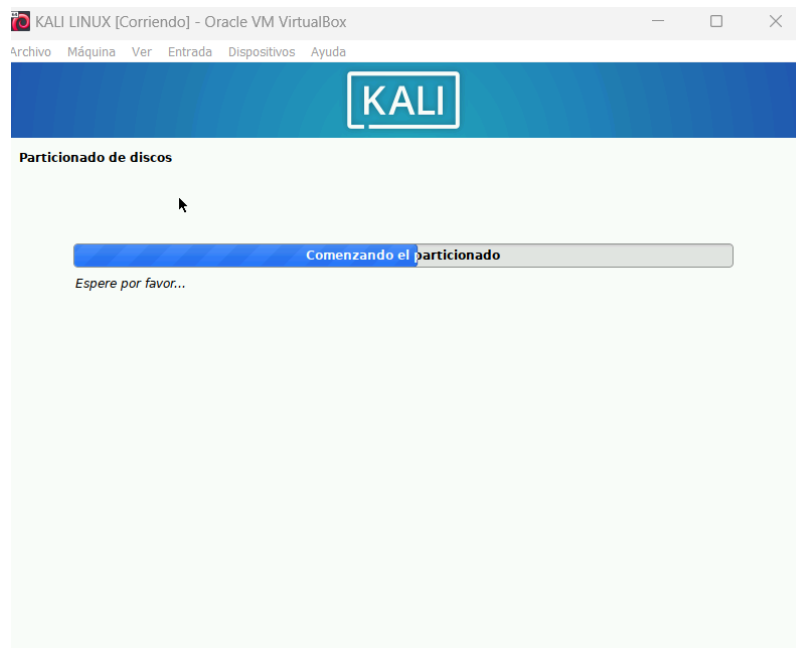


Figura A2.10. Partición de disco

8. Se define como se desea el particionado del disco y puede ser automático o asignando particiones, ver figura A2.11.

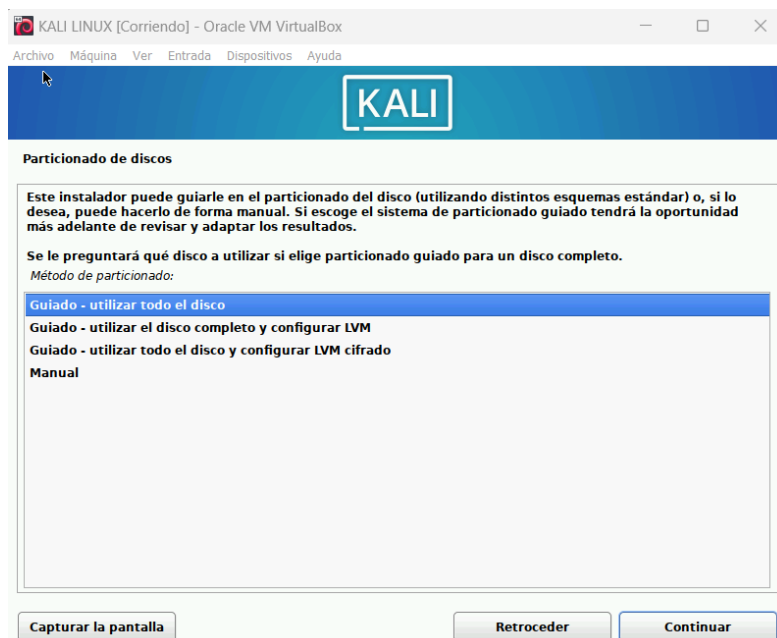


Figura A2.11. Partición de disco guiado

9. Se asigna una partición para los temporales y para la carpeta de Kali Linux.
Ver figura A2.12 y A2.13.

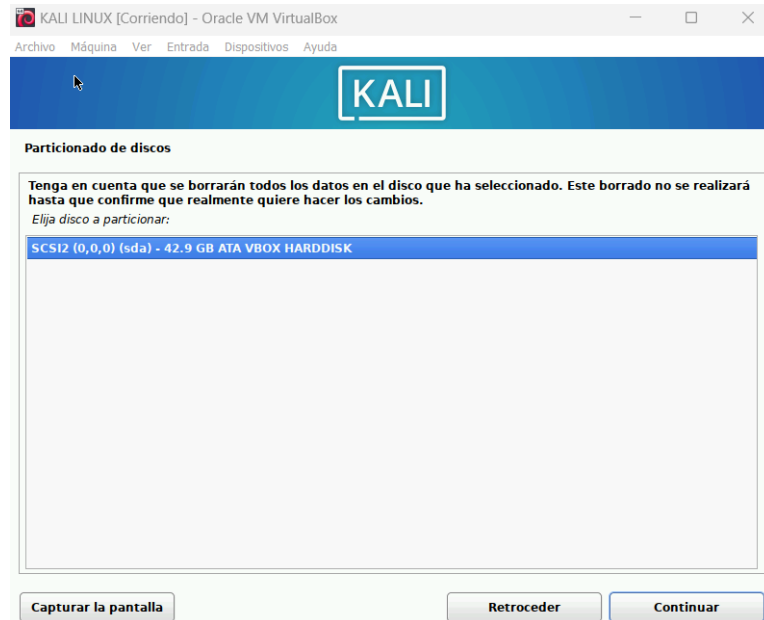


Figura A2.12. Partición de disco guiado para temporales

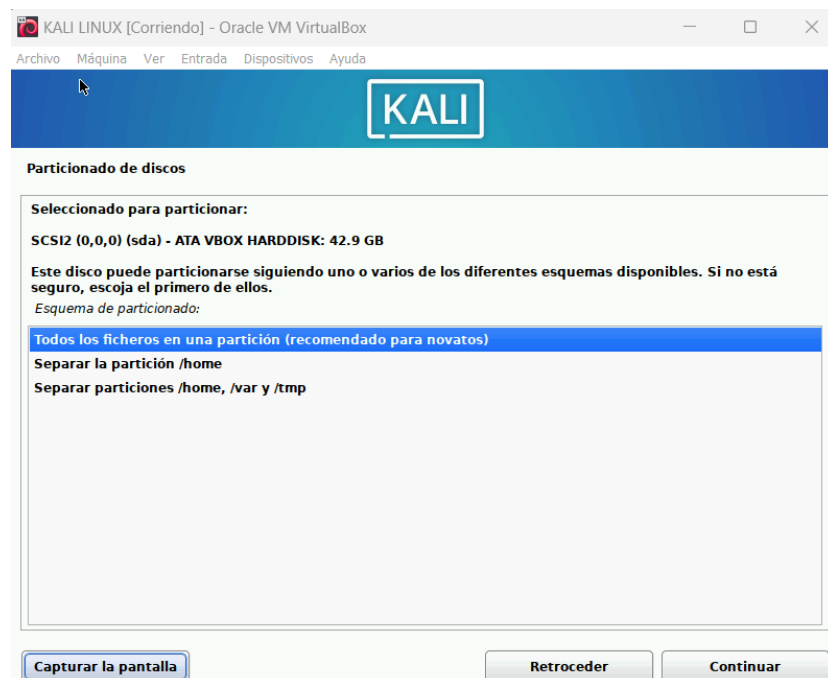


Figura A2.13. Esquemas de partición de disco para novatos

10. Se da clic en Continuar y se confirma el proceso, ver figura A2.14

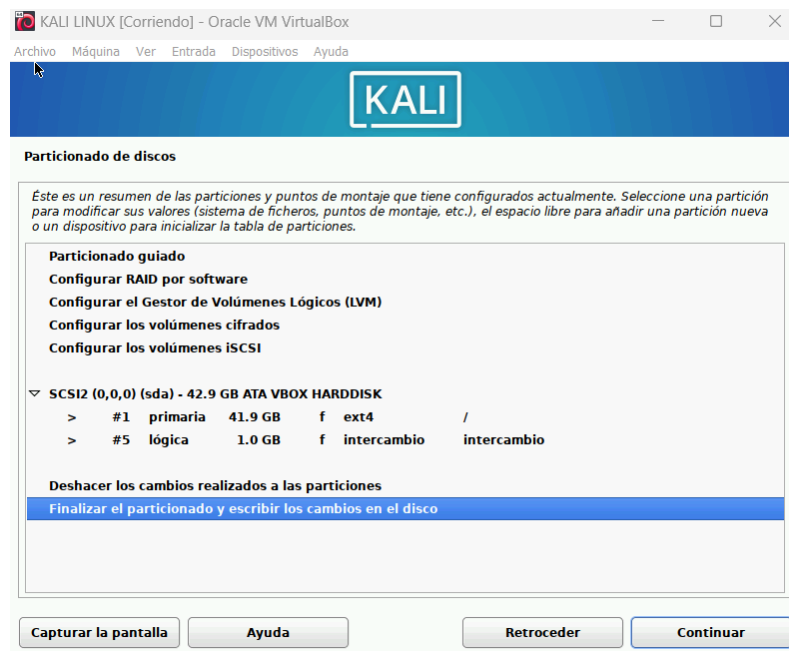


Figura A2.14. Confirmación de finalizar particionado.

11. Inicia instalación de Kali Linux, ver figura A2.15.

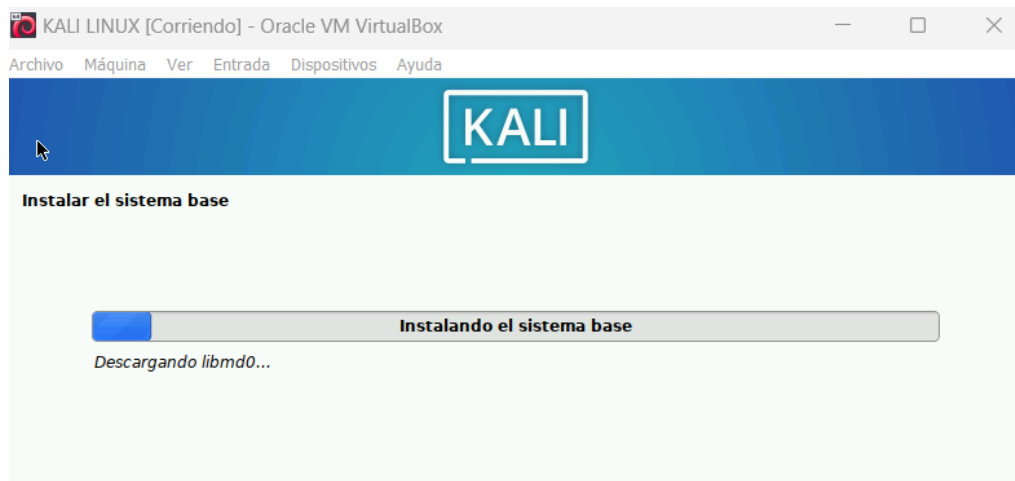


Figura A2.15. Proceso de instalación de Kali Linux

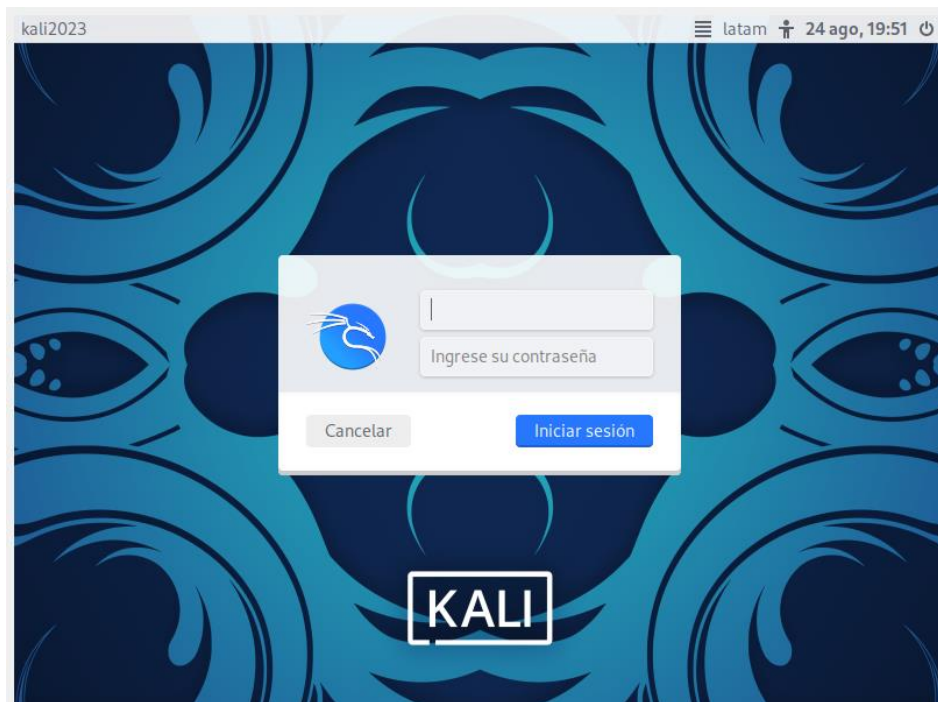
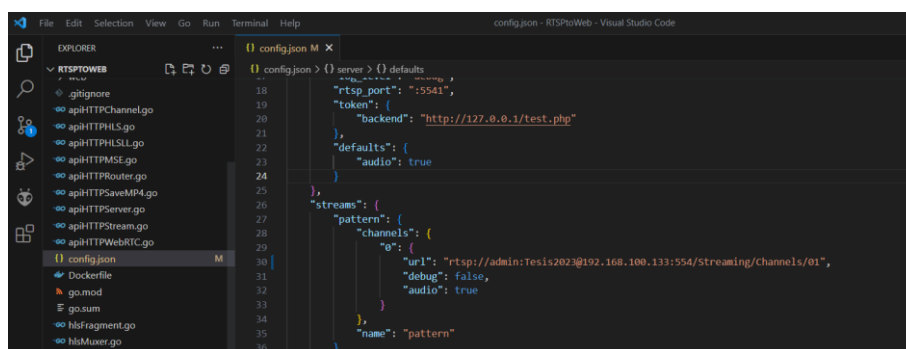


Figura A2.16. Kali Linux instalado en máquina virtual

Anexo 3: Manual de usuario del sistema de videovigilancia

Nota: Conectar la o las cámaras por cable ethernet a la red.

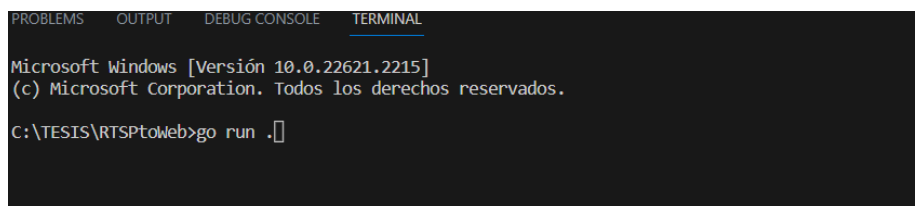
- **Paso 1:** Abrir el proyecto RTSPtoWeb en visual studio code y modificar en el archivo config.json en la línea 30 únicamente la ip de la cámara, dependerá según la red en la que se encuentre. Guiarse con la figura A3.2.



```
config.json > {} defaults
18   "rtsp_port": ":5544",
19   "token": {
20     "backend": "http://127.0.0.1/test.php"
21   },
22   "defaults": {
23     "audio": true
24   }
25 },
26 "streams": {
27   "pattern": {
28     "channels": {
29       "g": {
30         "url": "rtsp://admin:Tesis2023@192.168.100.133:554/Streaming/Channels/01",
31         "debug": false,
32         "audio": true
33       }
34     },
35     "name": "pattern"
36   }
}
```

• **Figura A3.1 Cambio de ip de la cámara.**

- **Paso 2:** Iniciar el servidor RTSP para el flujo de video con el comando go run ., ver figura A3.2.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Microsoft Windows [Versión 10.0.22621.2215]
(c) Microsoft Corporation. Todos los derechos reservados.
C:\TESIS\RTSPtoWeb>go run .
```

Figura A3.2 Ejecución del servidor RTSP.

- **Paso 3:** Iniciar el aplicativo del servidor de cadena de Bloques Ganache, y dar clic en Quick Start, ver figura A3.3.

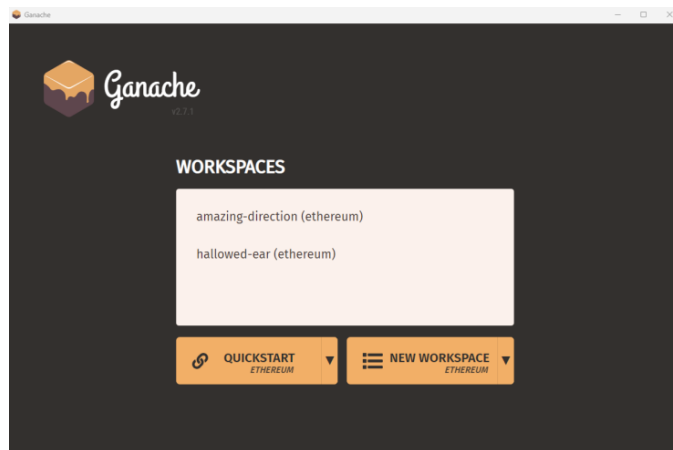


Figura A3.3 Portal inicial de Ganache

- **Paso 4:** Abrir el código del proyecto auth-blockchain en visual studio code y modificar el archivo .env en la línea 1, la ip de la cámara, ver figura A3.4.

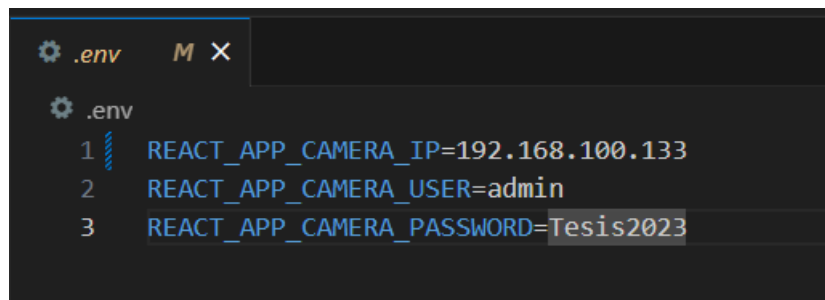


Figura A3.4 Datos de la cámara

- **Paso 5:** Ejecutar el proyecto con el terminal command prompt usando el comando npm start, ver figura A3.5.



Figura A3.5 Ejecución del proyecto de autenticación.

- **Paso 6:** Se inicia el interfaz web en donde a la par se inicia el metamask donde solicita iniciar sesión. Se ingresa la cuenta en la billetera digital, ver figura A3.6.

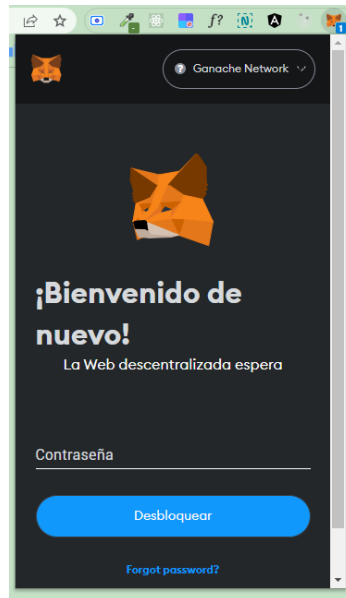


Figura A3.6 Inicio de sesión de metamask.

- **Paso 7:** Se agrega la cuenta que proporciona el Ganache en la billetera digital para tener saldo Ethereum, ver figura A3.7.

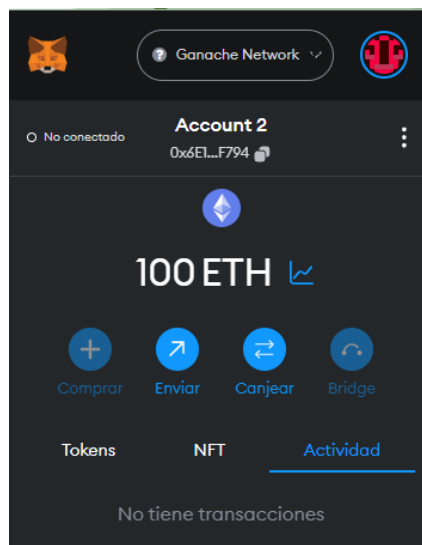


Figura A3.7 Saldo Ethereum en billetera digital.

- **Paso 8:** Se procede a crear una cuenta nueva en la interfaz, se llena el formulario con los datos de usuario, correo y contraseña. Se le da clic en crear cuenta, ver figura A3.8. Seguido a eso, se procede a procesar la transacción por medio de la billetera digital donde solicita rechazar o aceptar la misma por un pequeño costo de saldo.

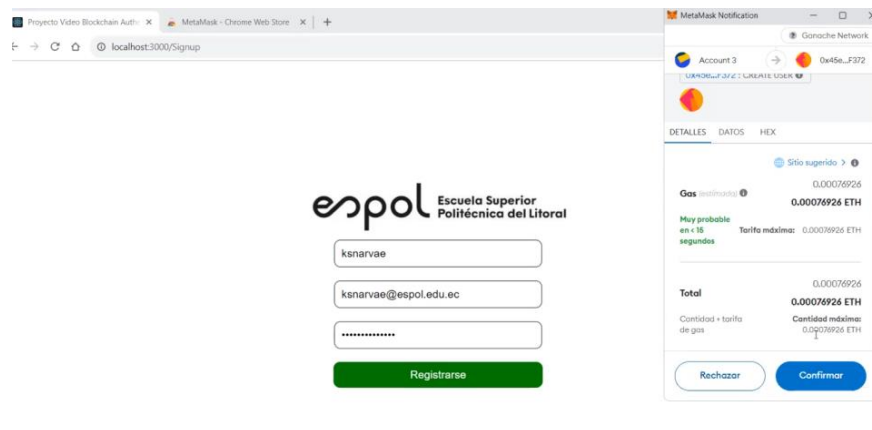


Figura A3.8 Transacción en proceso.

- **Paso 9:** Luego del registro del usuario se procede a iniciar sesión. Se ingresa usuario y contraseña, ver figura A3.9.

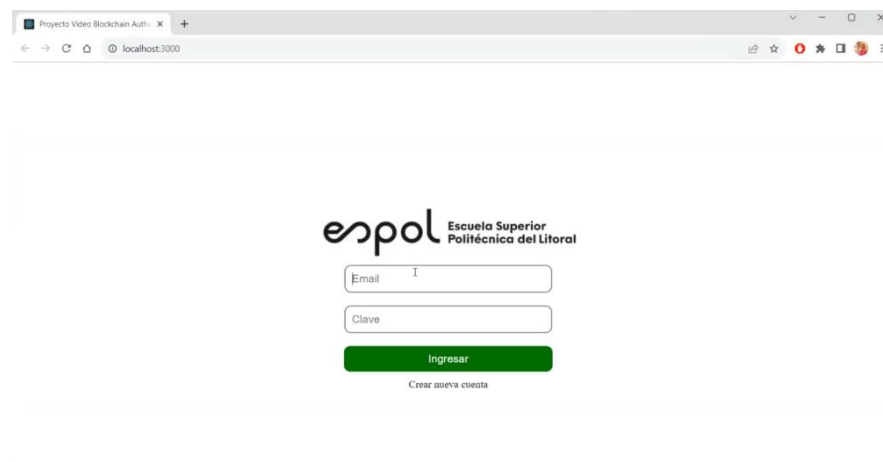


Figura A3.9 Inicio de sesión.

- **Paso 10:** Luego de iniciar sesión se accede al portal de monitoreo en donde visualizamos el video capturado por la cámara, ver figura A3.10.

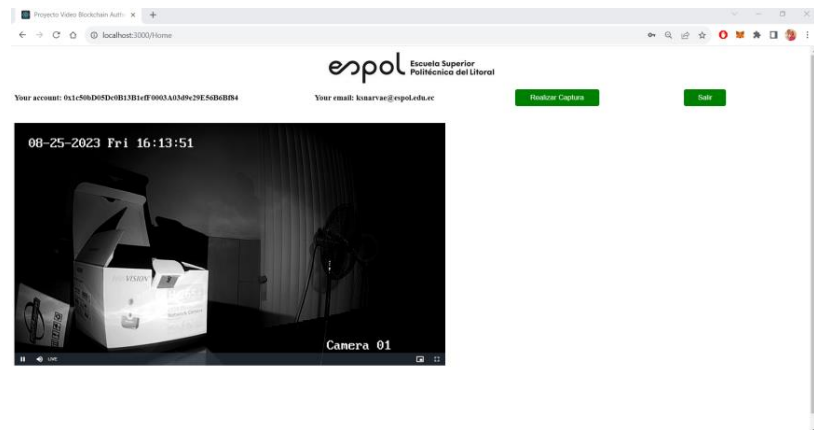


Figura A3.10 Portal de monitoreo.