Escuela
Politécnica
Superior

Sistema de seguimiento y monitorización de activos basado en blockchain

Grado en Ingeniería Informática

Trabajo Fin de Grado

Autor:
Daniel Asensi Roch
Tutor/es:
Higinio Mora Mora

Universitat d'Alacant
Universidad de Alicante

Julio 2023

Sistema de seguimiento y monitorización de activos basado en blockchain

Ejemplo práctico automóviles

Autor
Daniel Asensi Roch

Tutor/es
Higinio Mora Mora
Tecnología informática y computación

Grado en Ingeniería Informática

Escuela
Politécnica
Superior

Universitat d'Alacant
Universidad de Alicante

ALICANTE, Julio 2023

Quisiera expresar mi más profundo agradecimiento a todas aquellas personas que han formado parte de este emocionante y desafiante viaje.

Primero y ante todo, a mi familia, a quienes dedico este trabajo. Vuestra inquebrantable fe en mis capacidades y vuestro apoyo incondicional durante estos años de carrera han sido el faro que me ha guiado en cada paso que he dado. Habéis estado a mi lado tanto en los buenos momentos como en los malos, proporcionándome la fuerza necesaria para enfrentarme a los desafíos más duros.

A mis abuelos, vuestro apoyo ha sido una fuente de motivación y consuelo en momentos de adversidad. Vuestras sabias palabras y vuestra paciencia inagotable me han enseñado a ver cada derrota como una oportunidad de aprendizaje y a celebrar cada victoria con gratitud y humildad.

A mi querida novia y su maravillosa familia, vuestro interés y apoyo incesantes han marcado una gran diferencia en mi vida. Habéis extendido vuestra generosidad más allá de lo que nunca podría haber imaginado, y cada gesto de amabilidad y apoyo ha dejado una huella imborrable en mi corazón.

A mis amigos, vuestro espíritu inquebrantable y vuestra determinación incansable por mejorar cada día han sido una fuente constante de inspiración para mí. Vuestra compañía y amistad han hecho que este viaje sea aún más gratificante. Con vosotros a mi lado, sé que los cielos son el límite.

En cada uno de vosotros, encuentro una fuente de motivación y fuerza que me empuja a superarme a mí mismo. Este trabajo es el resultado de todas las lecciones que he aprendido de vosotros, y me siento verdaderamente agradecido de teneros en mi vida.

Gracias de todo corazón.

A mis padres y a mi novia los cuales me apoyaron incondicionalmente,
y a mis amigos,sin los cuales no me habría tomado
enserio el mejorar cada día

Cuando mientes, le robas al otro
el derecho a la verdad. Cuando engañas,
robas el derecho a la equidad

Khaled Hosseini.

Índice general

Índice de figuras

Índice de tablas

Índice de Códigos

1 Introducción

El estudio que se realiza tiene como objetivo solucionar la falta de confianza en aquellos mercados los cuales requieren una trazabilidad de los activos, donde la manipulación y la falsedad son comunes. La plataforma, que se desarrollará a lo largo de este escrito, busca crear un ecosistema en el que se pueda hacer un seguimiento del historial de un activo, desde su fabricación hasta su propietario actual, incluyendo reparaciones si las requiriera , mantenimientos o usuarios anteriores. Esto permitiría a los compradores tener información clara y confiable sobre el estado de un activo, reduciendo el riesgo de ser engañados.

1.1 Motivación

Las razones que me han llevado a realizar este estudio, han sido la general desconfianza que existe, a día de hoy, en los mercados automovilísticos, los cuales se encuentran gobernados por la pillería de tanto particulares como marcas de automoción, teniendo esta plataforma la finalidad de crear un ecosistema donde en concreto los vehículos puedan ser trazados desde cuando salió de la fábrica, hasta actual propietario, pudiendo leer de manera amigable, todos los accidentes, reparaciones y kilometrajes que ha tenido el vehículo, así como la cantidad de usuarios que han manejado el mismo. Dejando así al usuario que se encuentra disposición de informarse del automóvil en una situación ventajosa y de confianza, evitando las manipulaciones por parte de terceros sobre el estado del vehículo, penalizando la falsedad y dejando una huella por cada acción realizada sobre el mismo. Esta motivación me ha llevado a realizar el estudio de manera generalista, pero enfocándome en este mercado para tener un campo de implementación de estudio acotado.

1.2 Descripción del problema

El gran problema a los que nos enfrentamos durante este estudio es la trazabilidad, las modificaciones no autorizadas de los activos y el engaño producido por estas modificaciones, las cuales pueden dañar seriamente a los consumidores de dichos activos, así como entorpecer los sistemas gubernamentales o empresas los cuales requieren de información veraz, certificada y trazable.

1.2.1 Los mercados de segunda mano

Muchos de los problemas descritos en la sección anterior 1.2.3, se ven potenciados en los mercados de la segunda mano, donde el comprador se encuentra en una posición desventajosa a la hora de comprar cualquier activo, ya que algunos vendedores pueden modificar la información del activo con el objetivo de hacerlo más atractivo para el comprador, pero en realidad están engañándolo y causándole un perjuicio. Estas prácticas engañosas no solo son

ilegales, sino que también pueden causar problemas serios para los compradores potenciales, incluyendo problemas mecánicos, estructurales o de la índole del activo, problemas legales y pérdida de valor del activo. Por esta razón, es importante investigar y verificar la información antes de comprar o tasar un activo, y buscar fuentes confiables para obtener información sobre el historial del activo . [Almudí Coello et al. (2022)]

1.2.2 Burocracia y falta de sistemas

Otro de los grandes problemas que se encuentra el usuario, a la hora de conocer el estado del activo que desea comprar o vender, es la burocracia, la información sobre los activos se encuentra dispersa en diferentes agencias gubernamentales, lo que dificulta la obtención de información completa y actualizada de un activo. Los procesos para obtener información veraz y válida sobre un activo a menudo son complejos y requieren la presentación de una gran cantidad de documentos y la cumplimentación de diversos trámites. Esto puede llevar un tiempo prolongado y puede llegar a ser costoso tanto en tiempo como en dinero. La falta de estandarización puede dificultar la obtención de información precisa y confiable, además la protección de la privacidad puede dificultar el acceso a la información, ya que se requiere un proceso de verificación de la identidad y autorización del propietario para obtener cierta información.

1.2.3 Problema en el mercado automovilístico

La modificación de los cuadros de kilómetros (también conocida como "pillería de odómetro"[1]) es una práctica ilegal en la que se altera el contador de kilómetros de un vehículo para mostrar una cantidad menor de millas recorridas. Esto puede ser hecho mecánicamente, mediante la manipulación física del contador, o electrónicamente, mediante la manipulación del software del contador.[Romero (2022), Diaz (1985)]

Además de la pillería de odómetro, los usuarios de vehículos también pueden esconder defectos y choques mediante la reparación o el pintado para ocultar daños visibles en el exterior o en la estructura del vehículo. También pueden manipular los registros de reparaciones o mantenimientos, para ocultar la cantidad de revisiones necesarias o realizadas en el vehículo. Algunos vendedores pueden alterar los registros del vehículo para ocultar la cantidad de conductores o propietarios anteriores.

Estas prácticas son ilegales y pueden tener graves consecuencias legales para los responsables, además de causar problemas para los compradores potenciales del vehículo, ya que pueden ser engañados en cuanto al estado real del vehículo y pueden tener que asumir costos adicionales de reparación o de mantenimiento. [La Torre (2019)]

[1] La pillería de odómetro es una práctica ilegal que consiste en manipular el cuentakilómetros de un vehículo para disminuir la cantidad de kilómetros registrados y así aumentar su valor en el mercado de segunda mano. Esta práctica se lleva a cabo mediante la instalación de un dispositivo en el vehículo que puede detener, retroceder o acelerar el odómetro, lo que hace que el vehículo parezca que ha recorrido menos kilómetros de los que en realidad ha hecho.

1.3 Propuesta de solución

1.3.1 Mi solución

Como parte de mi trabajo de fin de grado, he desarrollado una solución innovadora para el registro y seguimiento de activos a través de la blockchain, en concreto me centraré en los vehículos, ya que fue lo que despertó mi interés en este campo de estudio. Esta solución se presenta como una aplicación sencilla y de bajo costo para el usuario, que permite registrar su vehículo y mantenerlo actualizado con información relevante.

Uno de los aspectos más importantes de esta solución es que permite el registro del vehículo en la blockchain, lo que garantiza su seguridad y protección contra posibles fraudes o alteraciones. Además, al estar registrados en la blockchain, los datos del vehículo estarán disponibles para su consulta en cualquier momento y desde cualquier lugar, lo que facilita el seguimiento y la gestión del vehículo.

La solución también ofrece la posibilidad de modificar la información del vehículo, añadiendo accidentes, datos de reparaciones o modificaciones, lo que permite mantener una historia completa y detallada del vehículo a lo largo del tiempo. Esta información será accesible tanto para el propietario del vehículo como para las entidades reguladoras del estado, como la ITV o la DGT, lo que garantiza la transparencia y la confianza en el proceso.

La información del vehículo podrá ser compartida con otros usuarios, lo que facilita la compra y venta de vehículos de segunda mano. De esta manera, cualquier persona interesada en comprar un vehículo podrá consultar su historial y obtener información detallada sobre su estado y su historial de mantenimiento, lo que aumenta la transparencia y la confianza en el proceso de compra.

En definitiva, esta solución innovadora para el registro y seguimiento de vehículos a través de la blockchain representa un avance importante en el sector de la automoción, al proporcionar una herramienta sencilla y eficaz para la gestión y seguimiento de los vehículos, garantizando la seguridad, transparencia y confianza en todo momento.

1.3.2 Aplicaciones descentralizadas

Una Organización Autónoma Descentralizada (en inglés Decentralized Autonomous Organization) o DAO, también llamada Empresa Autónoma Descentralizada (en inglés Decentralized Autonomous Corporation) o DAC, es una organización que está dirigida a través de reglas codificadas en programas de ordenador llamados contratos inteligentes. Un registro de transacción financiera de una DAO, así como dichas reglas, están gestionadas a través de un blockchain. [El Faqir et al. (2020)]

1.3.3 Objetivos

El objetivo de este trabajo son:

- Realizar un estudio de como realizar una trazabilidad en blockchain de cualquier activo.
- Realizar un análisis de la viabilidad de una solución descentralizada, para qué la compraventa de vehículos, tanto utilizados, como de concesionario, disponga de una herramienta segura y honesta para la obtención de información del vehículo.

- Realizar una prueba de concepto que permita a los usuarios registrar su vehículo personal en la DAO
- Crear una interfaz sencilla y segura, para que los usuarios se sientan cómodos interactuando con la blockchain.
- Desarrollar una serie de smart contracts utilizando el lenguaje de Solidity para interactuar con la blockchain de Ethereum.

Como se ve en el siguiente esquema, se propone una interfaz unificada para cada uno de los organismos que deseen participar en el uso de la aplicación, por un lado, tenemos a los usuarios, los cuales al igual que los organismos gubernamentales o direcciones tráfico se conectarán por interfaz web, los usuarios podrán ver la información de su vehículo y transferirlo, los organismos gubernamentales como el que realiza la ITV podrán modificar dicha información, al igual que la Dirección general de tráfico que podrá acceder a la información de los vehículos para mayor comodidad a la hora de realizar sanciones.

1.4 Metodología

La metodología de desarrollo ha requerido un enfoque investigativo centrado en soluciones descentralizadas, con el objetivo principal de diseñar una arquitectura que potencie la descentralización y garantice la trazabilidad de los activos en el tiempo.

Esta aproximación ha llevado a un stack de tecnologías que se diferencia notablemente de las configuraciones convencionales. Los contratos inteligentes, desarrollados en Solidity, actúan como el backend lógico de nuestra aplicación, eliminando la necesidad de un servidor centralizado y proporcionando una capa de confianza y seguridad inherente. Adicionalmente, la interacción con la blockchain, en nuestro caso Polygon, suplanta el uso de una base de datos tradicional, proporcionando un almacén de datos transparente, inmutable y resistente a la censura.

1.4.1 Herramientas

El desarrollo de los contratos inteligentes se ha llevado a cabo utilizando Hardhat, un entorno de desarrollo de Ethereum que facilita tareas como la compilación, el despliegue y las pruebas automatizadas. Esta herramienta permite simular una red blockchain local, lo que agiliza y optimiza el proceso de desarrollo y testing antes de la migración a la red de Polygon.

Para la construcción del frontend, hemos elegido Next.js, un marco de trabajo de JavaScript que permite un desarrollo rápido y efectivo, proporcionando características esenciales como el renderizado del lado del servidor y la generación estática de sitios. Para la estilización, utilizamos Bootstrap, gracias a su amplio conjunto de componentes predefinidos y su flexibilidad.

Por último, para el almacenamiento de la información, utilizamos IPFS (InterPlanetary File System), una red peer-to-peer descentralizada para el almacenamiento y la distribución de datos, que se alinea con nuestros objetivos de descentralización y resistencia a la censura. Todas estas herramientas anteriormente mencionadas serán explicadas a lo largo del capítulo 3

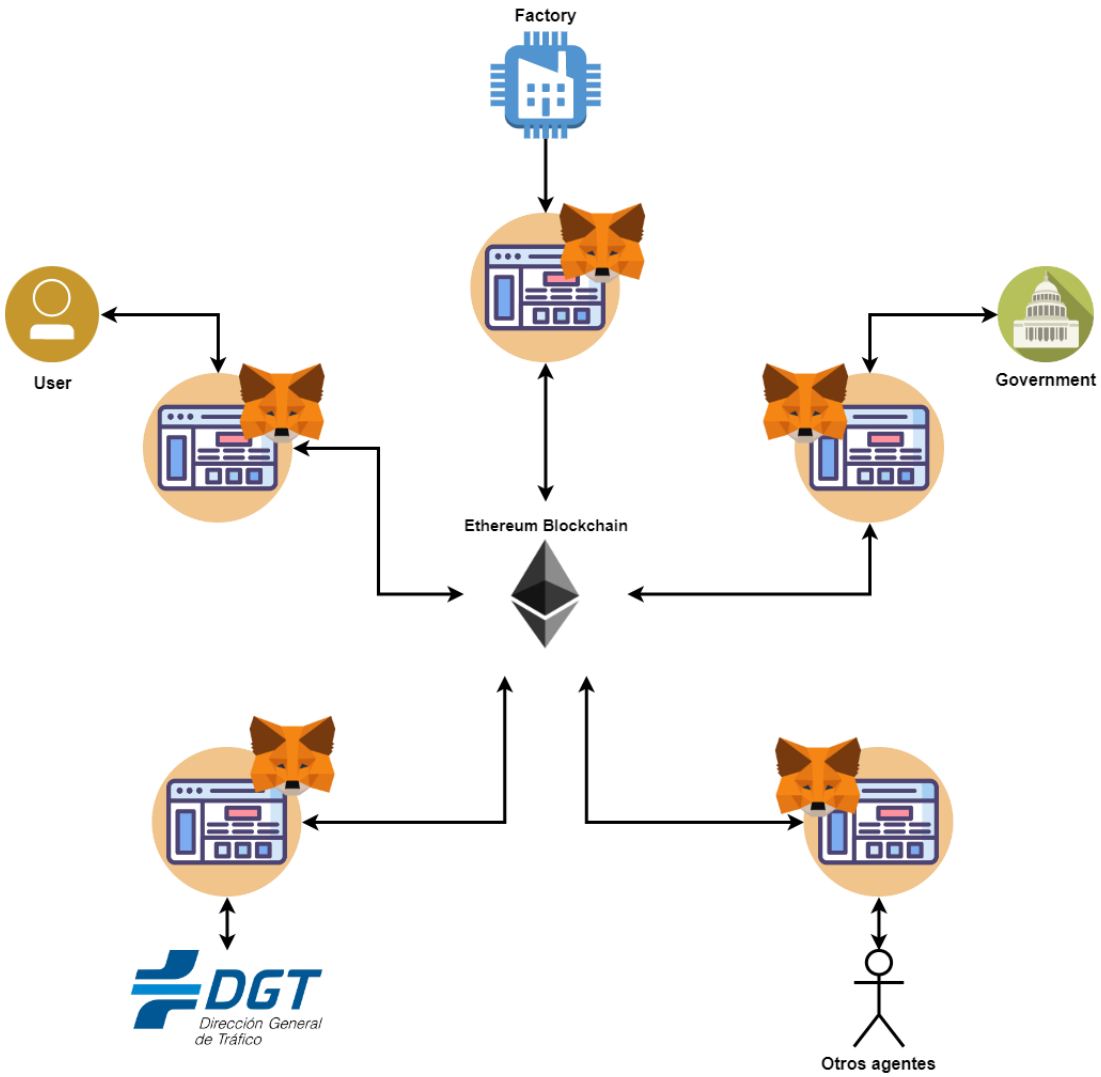

Figura 1.1: Esquema de uso

1.4.2 Diferencias entre trabajo en web2 y web3

El tránsito de la Web2 a la Web3 implica un cambio fundamental desde un modelo centralizado a uno descentralizado. En Web2, los datos son controlados por entidades centralizadas, con interacción a través de solicitudes HTTP/HTTPS. Sin embargo, en la Web3, los usuarios poseen y controlan sus datos, interactuando con contratos inteligentes en la blockchain. Este cambio introduce una mayor transparencia, seguridad y resistencia a la censura. Aunque requiere una nueva gama de habilidades técnicas, como el conocimiento de la blockchain y los contratos inteligentes, la transición a Web3 abre oportunidades para sistemas más justos y equitativos.

2 Estado del Arte

La trazabilidad de activos en blockchain ha ido ganando popularidad en los últimos años, debido al auge de la tecnología y de sus aplicaciones a diferentes ámbitos. El presente estado del arte tiene como objetivo abordar una visión general sobre los conceptos clave, las tecnologías existentes y las tendencias emergentes en el campo del seguimiento de activos en blockchain.

2.1 Conceptos clave

Los siguientes conceptos clave han sido obtenidos de la siguiente documentación. [Retamal et al. (2017)]

- **Blockchain:** Es una base de datos descentralizada y distribuida que permite almacenar y gestionar información de manera segura y transparente. La información se organiza en bloques que se enlazan entre sí a través de técnicas criptográficas.
- **Activos digitales:** Son representaciones digitales de bienes o derechos, como criptomonedas, tokens, contratos inteligentes o propiedades digitales, que se pueden rastrear y gestionar en una red blockchain.
- **Contratos inteligentes:** Son programas de software que se ejecutan automáticamente en una red blockchain y permiten la creación y gestión de activos digitales, así como la ejecución de acuerdos y transacciones entre partes.

2.2 Sistema de seguimiento de activos generales

En el contexto del seguimiento de activos en blockchain, una amplia variedad de herramientas y soluciones innovadoras han surgido para atender las necesidades de trazabilidad en el tiempo de los activos en múltiples industrias. Estas herramientas aprovechan las características clave de la tecnología blockchain, como la descentralización, la seguridad y la transparencia, al mismo tiempo que se integran con tecnologías complementarias, como Internet de las cosas (IoT), sensores avanzados y sistemas de identificación digital. Estas soluciones están diseñadas para abordar desafíos específicos en la gestión y rastreo de activos en diversos sectores, que van desde la cadena de suministro de alimentos, productos farmacéuticos y bienes de consumo, hasta la autenticación y verificación de activos de alto valor como diamantes, obras de arte y vinos exclusivos. A continuación, se presentará una selección de las principales herramientas y plataformas que están transformando la manera en que las empresas y organizaciones gestionan y rastrean sus activos, mejorando la transparencia, la eficiencia y la confianza en la cadena de suministro y en la gestión de activos en general, así como ofreciendo nuevas oportunidades para mejorar la sostenibilidad y la responsabilidad en diversos sectores económicos.

2.2.1 Herramientas actuales

2.2.1.1 Provenance

Provenance es una plataforma basada en blockchain que se centra en mejorar la transparencia y trazabilidad en las cadenas de suministro de productos y bienes. Utilizando la tecnología blockchain y contratos inteligentes, Provenance crea un registro inmutable y transparente de la cadena de suministro de un producto, incluyendo detalles sobre la procedencia de los materiales, prácticas laborales, procesos de producción sostenibles y certificaciones de calidad. La plataforma permite a las empresas demostrar su compromiso con la sostenibilidad y la ética, generando una ventaja competitiva en el mercado y mejorando la reputación de la marca.

Los consumidores pueden acceder a información detallada y verificable sobre los productos a través de etiquetas inteligentes, códigos QR y otras tecnologías de identificación. Esto fomenta una mayor conciencia y responsabilidad entre los consumidores, permitiéndoles tomar decisiones de compra informadas y apoyar a las empresas que comparten sus valores y prácticas sostenibles. Provenance ha sido adoptada por diversas industrias, como la moda, la alimentación, la electrónica y la cosmética, y ha sido utilizada por empresas de diferentes tamaños, desde pequeños productores hasta multinacionales.

Provenance se enfoca en emplear la tecnología blockchain y contratos inteligentes para registrar de manera segura y verificable información sobre el origen, producción e impacto de los productos.[Kim and Laskowski (2018)]

2.2.1.2 VeChain

VeChain es una plataforma blockchain enfocada en la gestión de la cadena de suministro y la trazabilidad de activos en diversos sectores. Utiliza identificadores únicos, como códigos QR, etiquetas NFC y chips RFID, junto con contratos inteligentes e Internet de las cosas (IoT) para monitorear y verificar el origen, la autenticidad y las condiciones de los productos en tiempo real. Los sensores integrados permiten rastrear factores críticos como temperatura y humedad, garantizando el transporte y almacenamiento adecuados de los productos.

La plataforma ofrece dos tokens, VeChain Token (VET) y VeChainThor Energy (VTHO), para facilitar las transacciones y ejecutar contratos inteligentes. VeChain se centra en la escalabilidad y adaptabilidad, permitiendo soluciones personalizadas según las necesidades específicas de las empresas en diferentes industrias, utiliza tecnologías avanzadas para mejorar la transparencia y eficiencia en la cadena de suministro, ofreciendo a las empresas una solución integral y adaptable. [She (2022)]

2.2.1.3 Ambrosus

Ambrosus es una plataforma blockchain de código abierto y ecosistema descentralizado que se enfoca en la trazabilidad y calidad de productos en las cadenas de suministro. Su infraestructura se basa en una arquitectura de múltiples capas que incluye la red principal AMB-NET, un protocolo de consenso personalizado denominado Proof of Authority (PoA) y el token nativo Amber (AMB). Los contratos inteligentes, la criptografía de clave pública y privada, y las técnicas de anonimato y ofuscación garantizan la privacidad y seguridad de los datos.

Ambrosus utiliza sensores y dispositivos IoT para recopilar datos en tiempo real sobre productos y activos en la cadena de suministro, como la temperatura, la humedad y la ubicación. Estos datos se almacenan en la cadena de bloques de Ambrosus mediante contratos inteligentes, creando un registro inmutable y verificable de la historia y condiciones del producto. Los contratos inteligentes también permiten la automatización de procesos y la ejecución de acuerdos basados en condiciones predefinidas, lo que mejora la eficiencia y reduce los errores humanos en la cadena de suministro.

Ambrosus también cuenta con un enfoque modular y un ecosistema abierto que fomenta la colaboración y la innovación. Los desarrolladores pueden crear aplicaciones y soluciones personalizadas (dApps) utilizando herramientas y API proporcionadas por la plataforma. [Garankina et al. (2018); Testaj (2022)]

2.2.1.4 Everledger

Everledger es una innovadora plataforma basada en blockchain centrada en proporcionar trazabilidad y transparencia en la cadena de suministro de productos de alto valor, como diamantes, piedras preciosas, arte, electrónica, vino y moda de lujo. Su objetivo es abordar problemas como la falsificación, el fraude y la falta de información en la procedencia de los productos.

La plataforma utiliza la tecnología blockchain para crear un registro digital seguro e inmutable de la historia y características de un producto, almacenando información crítica como el origen, la propiedad, los certificados de autenticidad y las condiciones de transporte y almacenamiento. Para lograr esto, Everledger asigna identificadores únicos a los productos, como números de serie, códigos QR y técnicas de microscopía, lo que permite rastrearlos a lo largo de su vida útil en la cadena de suministro. [Calvão (2019)]

Como dato interesante para futuros estudios sobre el trazado de activos, Everledger integra tecnologías complementarias como inteligencia artificial (IA), Internet de las cosas (IoT) y aprendizaje automático (ML) para mejorar la eficiencia y precisión en la recopilación, análisis y monitoreo de datos en tiempo real. Estas tecnologías permiten la detección de anomalías y patrones en la cadena de suministro, ayudando a identificar posibles problemas y riesgos, y a mejorar la toma de decisiones y la prevención de fraudes.

La plataforma también facilita el cumplimiento de las regulaciones y estándares de la industria en relación con la ética, sostenibilidad y responsabilidad social, lo que resulta en un mayor nivel de confianza y responsabilidad para las empresas y consumidores involucrados en la cadena de suministro. Al proporcionar una visibilidad mejorada y una autenticación confiable, Everledger permite a las partes interesadas tomar decisiones informadas y conscientes, reduciendo el riesgo de comprar productos falsificados o de origen desconocido. [Wüst and Gervais (2018)]

2.2.1.5 Modum

Modum es una innovadora empresa suiza que se centra en la optimización y monitoreo de la cadena de suministro, especialmente en la industria farmacéutica. Para lograr esto, Modum utiliza una combinación de tecnologías como blockchain, Internet de las cosas (IoT) y sensores inteligentes, creando soluciones que mejoran la calidad, seguridad y cumplimiento normativo en la cadena de suministro.

Modum emplea contratos inteligentes en su plataforma para establecer umbrales y condiciones específicas, como rangos de temperatura aceptables. Estos contratos inteligentes automatizan procesos y acciones en función del cumplimiento de las condiciones establecidas. Cuando se detecta alguna anomalía o desviación fuera de los parámetros establecidos, los contratos inteligentes pueden desencadenar alertas y notificaciones en tiempo real a las partes interesadas, permitiéndoles tomar medidas correctivas rápidas y eficientes.

La plataforma de Modum se centra en la trazabilidad y autenticidad de los productos, asegurando que las partes interesadas tengan acceso a información precisa y verificable sobre el origen, composición y calidad de los productos en la cadena de suministro. Esto es especialmente importante en la industria farmacéutica, donde el cumplimiento de las Buenas Prácticas de Distribución (GDP) y otras regulaciones es crucial.

Una de las ventajas de la solución de Modum es su escalabilidad y personalización, permitiendo a las empresas adaptar la plataforma a sus necesidades y requerimientos específicos. [Huang et al. (2019)]

2.2.2 Conclusión análisis generalista

A lo largo de nuestra discusión, hemos analizado varias soluciones basadas en blockchain, como Provenance, VeChain, Ambrosus, Everledger y Modum, que abordan diferentes aspectos de la trazabilidad y transparencia en la cadena de suministro. A pesar de sus diferencias y enfoques específicos, estas soluciones comparten varios puntos en común que destacan las ventajas y capacidades de la tecnología blockchain aplicada a la cadena de suministro.

Primero, todas estas soluciones utilizan la tecnología blockchain para crear registros digitales seguros e inmutables de la información y las transacciones relacionadas con los productos en la cadena de suministro. Esto permite a las partes interesadas, como fabricantes, distribuidores, minoristas y consumidores, acceder y verificar fácilmente la información sobre el origen, autenticidad, calidad y movimiento de los productos. La inmutabilidad de la información almacenada en la cadena de bloques garantiza su integridad y confiabilidad, lo que facilita la toma de decisiones informadas y responsables.

En segundo lugar, estas soluciones combinan la tecnología blockchain con otras tecnologías emergentes, como Internet de las cosas (IoT), inteligencia artificial (IA) y aprendizaje automático (ML). La integración de estas tecnologías complementarias permite una recopilación, análisis y monitoreo de datos más eficientes y precisos, lo que mejora la trazabilidad, seguridad y cumplimiento normativo en la cadena de suministro. Además, el uso de contratos inteligentes en estas plataformas permite automatizar procesos y acciones en función de condiciones específicas, lo que aumenta la eficiencia y la capacidad de respuesta ante anomalías o desviaciones.

En tercer lugar, todas las soluciones mencionadas tienen como objetivo mejorar el cumplimiento normativo y facilitar la adopción de prácticas éticas y sostenibles en la cadena de suministro. Al proporcionar trazabilidad y transparencia en tiempo real, estas plataformas permiten a las empresas demostrar su compromiso con la responsabilidad social y el cumplimiento de regulaciones y estándares de la industria. Esto puede resultar en una mayor confianza y reputación entre los consumidores y otras partes interesadas.

2.3 Sistemas de seguimiento de automóviles

Como he explicado en la sección 1.3.3, uno de los objetivos de este trabajo, es realizar una aplicación aproximada al análisis generalista realizado con anterioridad, haciendo énfasis en un activo que muchos de nosotros poseemos en nuestros hogares, es decir, los automóviles, por lo que realizaré un pequeño estado del arte de las soluciones parecidas a la mía que se encuentran en el mercado, realizando una enumeración de sus ventajas y desventajas, para así en la solución desarrollada en este proyecto intentar realizar una aplicación segura e intuitiva para el usuario.

2.3.1 Enfoque clásico

2.3.1.1 La cartilla del coche o libro de mantenimiento

Este documento posee toda la información importante sobre un vehículo, así como el número de chasis, el número de motor, la marca, el modelo, el año, etc. También puede contener información sobre el historial del vehículo, como el número de kilómetros, las revisiones realizadas, los cambios de propietario, o la cantidad de accidentes que ha tenido el vehículo.

Este es necesario en muchos lugares si requieres de hacer una reparación a tu vehículo, pero en muchos otros este es obviado y no actualizado. [RACE (2023)]

2.3.1.2 Libro digital de mantenimiento (Libro taller)

Aunque se ha intentado realizar un estándar, para proporcionar a los usuarios una herramienta donde queden registradas todas las revisiones del vehículo, así como los accidentes y propietarios del mismo, esta no ha llegado a ser formalizada ante los talleres, aseguradoras ni usuarios, debido a la gran cantidad de infraestructura necesaria para la misma y a la gran cantidad de regulaciones que necesitan los talleres para poder implementarla.[Costas (2019)]

2.3.1.3 Carfax

Carfax es un servicio que proporciona información sobre el historial de un vehículo. Este servicio recopila información de una variedad de fuentes, incluyendo registros de la industria automotriz, registros gubernamentales y servicios de reparación de vehículos. La información recopilada incluye detalles como el número de propietarios anteriores, accidentes reportados, kilómetros, servicios y revisiones realizadas, entre otros.[Rueda (2023)]

2.3.1.4 El precio, índole privada sin confianza

El gran inconveniente de Carfax, es que es una empresa, por lo que, los informes tienen un costo, lo que puede ser una desventaja para algunos compradores potenciales, la información proporcionada por una empresa privada puede ser incompleta o no estar completamente actualizada. Si una empresa privada es la mejor proveedora de información sobre automóviles, esto significa que los compradores potenciales de vehículos usados están altamente dependientes de esta empresa para obtener información precisa y completa sobre el historial de un vehículo, lo que podría generar dudas sobre la confiabilidad y precisión de la información proporcionada.

2.3.2 Soluciones que utilizan blockchain

A día de hoy se han probado multitud de soluciones para el problema que se intenta resolver en este trabajo. Pero cabe recalcar que todas sin mucho éxito, y muchas de ellas derivando sus proyectos a otras índoles. Los proyectos que explicaré a continuación han sido ordenados por similitud a mi solución:

2.3.2.1 Bitcar

Bitcar: BitCar es una plataforma basada en blockchain que permite a los usuarios poseer y comerciar fracciones de automóviles exóticos y coleccionables. La plataforma utiliza tokens no fungibles (NFT) y tokens de utilidad para representar la propiedad fraccionada de vehículos de lujo. La idea detrás de BitCar es permitir que las personas inviertan en automóviles exóticos y de alto rendimiento sin tener que comprar el vehículo completo, lo que hace que este tipo de inversión sea más accesible para un público más amplio.

BitCar utiliza la tecnología blockchain para garantizar la transparencia y la seguridad de la propiedad y las transacciones en la plataforma. Al utilizar NFT, cada fracción de un automóvil puede ser única y auténtica, lo que facilita la trazabilidad y la verificación de la propiedad. Además, la plataforma incluye funciones para permitir la compra, venta y comercialización de estas fracciones de automóviles, lo que brinda a los usuarios la oportunidad de beneficiarse de la apreciación del valor de los vehículos a lo largo del tiempo, además registra y verifica la información sobre los vehículos, incluyendo el historial de kilometraje y accidentes. Estos datos se guardan como NFTs en la blockchain de ethereum, lo que provoca que una vez minteados estos NFTs, se vuelvan inalterables, derivando a que solo se puedan intercambiar y destruir una vez creados. [Starčević and Puž]

2.3.2.2 Vinchain

VinChain es una innovadora plataforma basada en blockchain que busca revolucionar el mercado de automóviles usados al proporcionar un historial completo y transparente de vehículos, utilizando el número de identificación del vehículo (VIN) como clave única. El objetivo es abordar problemas comunes en la industria, como la manipulación del kilometraje y la falta de información precisa y verificable sobre el historial de los vehículos.

La plataforma VinChain recopila información de una amplia variedad de fuentes, incluidos concesionarios de automóviles, talleres de reparación, compañías de seguros, registros gubernamentales y sistemas de diagnóstico a bordo (OBD). Al almacenar todos estos datos en una cadena de bloques, VinChain garantiza la inmutabilidad y la transparencia de la información del vehículo, lo que dificulta la manipulación o falsificación de registros.

Además, la plataforma VinChain cuenta con su propio token, el VIN, que se utiliza para realizar transacciones y acceder a servicios dentro de la plataforma. Los usuarios pueden utilizar tokens VIN para adquirir informes del historial de vehículos, mientras que aquellos que contribuyen con datos e información pueden recibir tokens VIN como recompensa. Esto fomenta un ecosistema activo y autosostenible en el que los usuarios tienen incentivos para participar y mantener la calidad y precisión de los datos.

La plataforma emplea contratos inteligentes para gestionar el acceso a la información del vehículo, lo que permite un intercambio seguro de información entre las partes interesadas

sin intermediarios. Los usuarios pueden utilizar el token VIN nativo de la plataforma para adquirir informes del historial de vehículos o recibir recompensas por contribuir con datos e información.[Yoo and Ahn (2021)]

2.3.2.3 Mobility Open Blockchain Initiative (MOBI), otro enfoque

Por último contamos con una solución diferente, **Mobility Open Blockchain Initiative (MOBI)** es una colaboración global que reúne a fabricantes de automóviles, empresas tecnológicas, proveedores de servicios de movilidad y otros actores de la industria automotriz para acelerar la adopción de la tecnología blockchain y otras tecnologías distribuidas en la industria de la movilidad. MOBI fue fundada en 2018 por un consorcio de empresas líderes en la industria, incluidas BMW, Ford, General Motors, Renault y otras.

El objetivo principal de MOBI es crear un ecosistema descentralizado y estandarizado que facilite la transferencia segura y transparente de datos y servicios relacionados con la movilidad. Para lograr esto, MOBI se centra en áreas clave como la identidad digital del vehículo, el historial de uso y mantenimiento, la financiación y los seguros, la carga y el pago de energía, y la conducción autónoma, entre otros.

MOBI promueve el uso de estándares abiertos y protocolos comunes para garantizar la interoperabilidad entre las diferentes soluciones basadas en blockchain y tecnologías distribuidas. [Powell et al. (2021)]

2.3.2.4 Conclusión

En conclusión, el problema de la falta de transparencia y trazabilidad en el historial de los vehículos ha sido abordado por diversas soluciones, algunas de ellas basadas en la tecnología blockchain. Estas soluciones tienen en común el objetivo de proporcionar a los compradores potenciales de vehículos usados información precisa y completa sobre el historial del vehículo, incluyendo el número de propietarios anteriores, accidentes reportados, kilómetros, servicios y revisiones realizadas, entre otros.

Sin embargo, estas soluciones tienen diferentes enfoques y niveles de éxito. Algunas utilizan la tecnología blockchain de manera eficiente para garantizar la inmutabilidad y transparencia de los datos, mientras que otras enfrentan desafíos en cuanto a la actualización y seguridad de los datos. Además, algunas soluciones requieren una gran infraestructura y regulaciones para su implementación, lo que dificulta su adopción.

La solución ideal para abordar el problema de la falta de transparencia en el historial de los vehículos aún no ha sido encontrada. Es necesario seguir investigando y desarrollando soluciones innovadoras que utilicen la tecnología blockchain de manera efectiva para garantizar la seguridad y trazabilidad de la información, sin dejar de lado la accesibilidad y la facilidad de uso para los usuarios.

Solución	Transparencia	Inmutabilidad	Enfoque	Fuentes de información
Cartilla del coche	Baja	No	Historial del vehículo	Talleres, propietarios
Libro Taller	Baja	No	Historial del vehículo	Talleres, propietarios
Carfax	Media	No	Historial del vehículo	Registros gubernamentales, talleres
Bitcar	Alta	Sí	Inversión en vehículos exóticos	Registros gubernamentales, talleres
VinChain	Alta	Sí	Historial del vehículo	Registros gubernamentales, talleres, seguros, OBD

Tabla 2.1: Tabla comparativa soluciones

3 Marco teórico y tecnologías

3.1 La descentralización

Por descentralización nos referimos, al proceso de transferir el poder o autoridad desde una sola entidad reguladora centralizada, hacia múltiples entidades descentralizadas. [Colaboradores de los proyectos Wikimedia (2023a)]

3.1.1 Ventajas

La descentralización ayudaría a solucionar muchos de los problemas descritos en los apartados anteriores, al permitir que varias entidades tengan acceso a la información del vehículo, a través de un sistema de registro descentralizado de vehículos. En lugar de depender de una sola entidad gubernamental o privada para recopilar y mantener información sobre un vehículo, varias entidades podrían tener acceso y contribuir a la información sobre un vehículo en un sistema descentralizado. Esto podría ayudar a reducir la posibilidad de manipulación o alteración de la información, ya que varias entidades estarían revisando y verificando la información, permitiendo una mayor transparencia y trazabilidad, ya que la información estaría disponible para cualquier entidad que la necesite. [Cuadros Choez (2020)]

3.2 La Blockchain

3.2.1 Introducción

La blockchain es un registro digital distribuido que se utiliza para el registro de transacciones y mantener un historial inmutable y seguro de cada dato. Permitiendo a sus usuarios la transmisión de valor e información de forma segura, sin la necesidad de intermediarios. [Bib (2023)]

3.2.2 Funcionamiento

La blockchain, por su traducción *cadena de bloques*, funciona mediante la formación de bloques que contiene una serie de transacciones, cada uno de estos bloques se conecta a un bloque anterior mediante funciones criptográficas. Una vez añadido un bloque a esta cadena, los datos que contiene son permanentes y no se pueden modificar, como se puede ver en la figura 3.1. Esta seguridad y transparencia se deben a la naturaleza descentralizada de la blockchain, que permite que muchos nodos en una red validen y mantengan una copia de la blockchain. Esto significa que no hay un punto único de falla o control, y que la blockchain es resistente a la manipulación o la censura.

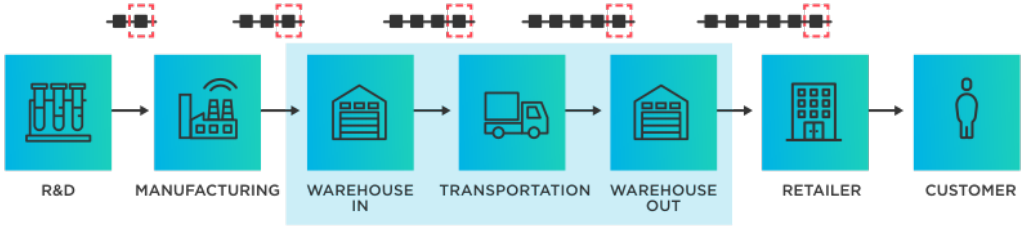

Figura 3.1: Ejemplo Adición de bloques a blockchain

3.2.3 Ventajas

A raíz de los siguientes trabajos [Golosova and Romanovs (2018); Gatteschi et al. (2018)], he extraído las principales ventajas de la tecnología:

1. **Seguridad:** La blockchain es un sistema seguro y resistente a la manipulación debido a su estructura distribuida y a la utilización de criptografía avanzada para mantener la integridad de los datos.

2. **Transparencia:** La blockchain permite una mayor transparencia en la gestión de transacciones y datos, ya que todos los nodos en la red tienen una copia del registro y pueden verificar su integridad.

3. **Descentralización:** La blockchain es un sistema descentralizado, lo que significa que no hay un intermediario central en control. Esto puede reducir los costos y mejorar la eficiencia en muchos casos.

4. **Protección de la privacidad:** Algunas implementaciones de blockchain permiten proteger la privacidad de los usuarios mediante la utilización de técnicas como la criptografía de direcciones o el anonimato.

5. **Automatización:** Los contratos inteligentes en la blockchain permiten la automatización de procesos mediante la codificación de reglas y acuerdos en código.

3.2.4 Desventajas

Al igual que las ventajas, las desventajas también se exponen en la siguiente documentación [Golosova and Romanovs (2018); Gatteschi et al. (2018)], estas son las principales desventajas del uso de la tecnología blockchain:

1. **Escalabilidad:** Limitaciones en la capacidad de procesamiento pueden ser un obstáculo para la adopción a gran escala.

2. **Latencia:** Validación y registro de transacciones pueden ser más lentos debido a la naturaleza descentralizada.
3. **Costo:** Minería y validación de transacciones pueden requerir grandes cantidades de energía y recursos, lo que puede resultar en costos elevados.
4. **Dificultad para cambiar las reglas:** Puede ser difícil cambiar las reglas y el funcionamiento una vez implementado, obstaculizando su evolución y mejora.
5. **Falta de regulación:** Aunque está ganando adopción, todavía existen barreras para su adopción generalizada, incluyendo la falta de comprensión y regulaciones claras.
6. **Interoperabilidad:** La falta de interoperabilidad entre implementaciones diferentes puede ser un obstáculo para la integración y uso conjunto.

3.3 Qué es Ethereum

Ethereum es una plataforma descentralizada basada en blockchain que permite a los usuarios crear y ejecutar aplicaciones descentralizadas o contratos inteligentes en un entorno seguro y sin posibilidad de manipulación, además utiliza su propia criptomoneda, Ether (ETH), por su símbolo 4.3, como moneda de cambio y para compensar a los mineros que verifican y validan las transacciones en la red. La plataforma Ethereum permite a los desarrolladores crear contratos inteligentes, que son programas que se ejecutan automáticamente cuando se cumplen ciertas condiciones específicas. Estos contratos inteligentes pueden ser utilizados para una amplia variedad de aplicaciones, como la automatización de procesos empresariales, la creación de tokens no fungibles (NFTs) y la creación de mercados descentralizados. [Sol (2023); Vujičić et al. (2018)]

Figura 3.2: Símbolo Ethereum

3.3.1 Por qué el uso de Ethereum

Haciendo referencia al apartado anterior 3.3, a diferencia de la red de Bitcoin, que se basa en un sistema de contabilidad de transacciones no programable, Ethereum es una plataforma que admite la ejecución de código programático en una máquina virtual basada en la blockchain. Para ello, Ethereum soporta lenguajes de programación Turing completos como Solidity, que permiten a los desarrolladores escribir contratos inteligentes que se ejecutan en la plataforma de Ethereum.

La característica principal de Ethereum es la capacidad de programar y ejecutar contratos inteligentes de forma distribuida y descentralizada, lo que permite la creación de aplicaciones

descentralizadas (dApps) que operan sin la necesidad de un intermediario centralizado. La red de Ethereum ofrece un entorno seguro y transparente para la ejecución de los contratos inteligentes y dApps, y los participantes de la red pueden confiar en la inmutabilidad y la ejecución determinista de los contratos.

3.4 Smart Contracts

3.4.1 Solidity

Solidity es un lenguaje de programación utilizado para escribir smart contracts en la plataforma Ethereum. Solidity permite a los desarrolladores de software crear y desplegar contratos inteligentes en la red de Ethereum y es uno de los lenguajes de programación más populares para desarrollar aplicaciones descentralizadas.

3.4.2 Que es un Smart Contract

Un smart contract es un programa que se ejecuta automáticamente cuando se cumplen ciertas condiciones específicas. Estos contratos son utilizados en la blockchain de Ethereum para automatizar y hacer más eficientes ciertos procesos empresariales.

Para crear un smart contract, se debe escribir un programa en el lenguaje de programación Solidity que especifica las condiciones y la lógica del contrato, además de Solidity existen muchos otros lenguajes de programación de Smart Contracts, pero este es el más extendido.

El programa se compila en un bytecode [1] que es desplegado en la blockchain de Ethereum, lo que permite a los usuarios de la red interactuar con el contrato a través de la ejecución de sus funciones. Cabe recalcar que para que los usuarios u otros contratos puedan interactuar con las funciones del contrato, se debe de conocer el *address*, este se refiere a un identificador único que se utiliza para representar un contrato inteligente o una cuenta en la red Ethereum.

Es importante tener en cuenta que las direcciones son públicas y cualquier persona puede verlas. Por lo tanto, es importante proteger la seguridad de los contratos inteligentes y no revelar la dirección a personas no autorizadas. [Modi (2018)]

3.4.3 Despligue de Smart Contracts

Existen varias maneras de desplegar estos contratos. Pero debemos recordar que en Ethereum ”nada es gratuito” y toda alteración o ejecución en la blockchain requiere de un pago dependiendo de su costo computacional, a este costo se le denomina *Gas*[Mohanty and Mohanty (2018).]

3.4.3.1 El Gas

En Ethereum, el ”gas” se refiere a la unidad de medida que se utiliza para calcular el costo de ejecutar una transacción o un contrato inteligente en la red. En otras palabras, el gas es una medida del trabajo que se necesita para realizar una tarea en la red Ethereum.

[1] El bytecode es en general un conjunto de instrucciones de bajo nivel que la máquina virtual o el intérprete de la plataforma puede ejecutar directamente. El bytecode es utilizado en la blockchain para garantizar la inmutabilidad de los smart contracts, es decir, que el código original no se puede modificar después de ser desplegado.

Cada transacción o contrato inteligente tiene un costo en gas, que se determina por la complejidad y la duración de la tarea que se debe realizar. El costo del gas se mide en ether, la criptomoneda nativa de la red Ethereum.

El gas se utiliza para incentivar a los "mineros" [2] de Ethereum a validar y procesar las transacciones en la red. Los mineros reciben una recompensa en ether por procesar las transacciones, y cuanto más trabajo tenga que hacer un minero para procesar una transacción, mayor será la cantidad de gas necesaria para completarla y, por lo tanto, mayor será la recompensa que recibe.

3.4.3.2 Test Networks

Las test networks de Solidity son redes de prueba que se utilizan para probar y depurar programas escritos en Solidity, que es el lenguaje de programación utilizado para crear contratos inteligentes en la plataforma Ethereum.

Estas redes permiten a los desarrolladores probar su código antes de desplegarlo en la red principal de Ethereum, lo que les permite detectar errores y problemas de seguridad antes de que los usuarios finales interactúen con los contratos inteligentes en la red real. Usando tokens sin valor, que los programadores pueden reclamar en las faucets[3], para poder subir los smart contracts a la red.

Existen diferentes test networks de Solidity, como Rinkeby, Kovan, Ropsten, entre otras, cada una con sus propias características y propósitos. En general, las test networks de Solidity son un recurso valioso para los desarrolladores que trabajan en la plataforma Ethereum, ya que les permiten depurar y mejorar sus contratos inteligentes antes de lanzarlos a la red principal. [Iyer et al. (2018)]

3.4.3.3 Main Net

La "Mainnet" de Ethereum se refiere a la red principal y pública de Ethereum, que es donde se llevan a cabo todas las transacciones y contratos inteligentes reales que utilizan ether, la criptomoneda nativa de Ethereum, es la red real, mantenida por una red global de nodos y mineros. En esta red, se pueden crear y desplegar contratos inteligentes y tokens, que luego se pueden intercambiar y transferir entre usuarios en todo el mundo. Es el lugar donde ocurren las transacciones "reales" en la red, lo que significa que todas las transacciones y contratos inteligentes tienen un costo en ether y deben ser verificadas y validadas por los nodos y los mineros de la red. [Iyer et al. (2018); Robinson (2020)]

[2] los "mineros" son los nodos de la red que validan y procesan las transacciones y los contratos inteligentes en la blockchain de Ethereum.

[3] Un faucet permite a los desarrolladores y usuarios obtener ether falso sin tener que comprarlo o minarlo. Por lo general, los faucets de las redes de prueba requieren que los usuarios ingresen su dirección de cuenta de la red de prueba para poder recibir el ether falso. Luego, el faucet envía una cantidad pequeña de ether falso a la dirección de cuenta del usuario.

3.5 Aplicaciones descentralizadas: dApps

3.5.1 Web3

Web3 es una abreviación de "Web 3.0", que se refiere a la evolución de Internet hacia un Internet descentralizado y basado en blockchain.

Web3 se enfoca en la creación de una red descentralizada en la que los usuarios pueden interactuar directamente entre sí sin la necesidad de intermediarios, utilizando tecnologías de blockchain, criptomonedas y contratos inteligentes. El objetivo de Web3 es proporcionar una Internet más segura, privada y resistente a la censura.

En lugar de depender de una serie de proveedores de servicios centralizados, Web3 busca crear un ecosistema descentralizado en el que los usuarios puedan controlar su propia identidad, datos y finanzas. Algunas aplicaciones de Web3 incluyen el intercambio de criptomonedas, la creación de contratos inteligentes y la implementación de sistemas de votación descentralizados.

Para interactuar con la red descentralizada de Web3, se requiere un navegador web especializado conocido como "navegador Web3" o "wallet Web3", que permite a los usuarios interactuar con aplicaciones descentralizadas y enviar y recibir criptomonedas. [Belk et al. (2022); Wang et al. (2022)]

3.5.2 Carteras

Como explican los siguientes estudios y libros relacionados con la tecnología blockchain [Sheridan et al. (2022); Wang et al. (2022); Voshmgir (2020)], En el contexto de Web3, una "wallet" o "cartera" se refiere a una aplicación o software que se utiliza para almacenar, enviar y recibir criptomonedas y tokens basados en blockchain, así como para interactuar con contratos inteligentes en la red.

Las wallets son esenciales para los usuarios que desean participar en el ecosistema de Web3, ya que les permiten gestionar su identidad y sus activos digitales de forma segura y descentralizada. Las wallets pueden funcionar en diferentes plataformas, como aplicaciones web, aplicaciones de escritorio o aplicaciones móviles.

Una wallet Web3 es diferente de una wallet tradicional en que no almacena dinero físico, sino activos digitales basados en blockchain, como criptomonedas y tokens. Las wallets Web3 también permiten a los usuarios interactuar con contratos inteligentes, lo que les permite participar en aplicaciones descentralizadas, intercambios de criptomonedas y otros servicios de Web3.

Es importante destacar que, a diferencia de las wallets tradicionales, las wallets de Web3 no son controladas por intermediarios centralizados. En su lugar, las wallets de Web3 utilizan tecnologías de criptografía [4] y blockchain para proporcionar un alto nivel de seguridad y privacidad a los usuarios.

[4] La criptografía es una práctica que consiste en proteger información mediante el uso de algoritmos codificados, hashes y firmas.

3.6 IPFS: InterPlanetari File System

El InterPlanetary File System (IPFS) es un sistema de archivos descentralizado y de código abierto que utiliza tecnología blockchain para permitir la distribución de archivos de manera descentralizada. IPFS es una alternativa a la World Wide Web y a los sistemas de archivos centralizados.

En lugar de almacenar archivos en un solo servidor central, los archivos en IPFS están distribuidos entre nodos de la red, lo que permite una mayor resistencia a la censura y una mayor disponibilidad de los datos. IPFS utiliza una dirección hash única para cada archivo, lo que permite que los archivos se encuentren y se distribuyan de manera eficiente en la red.

IPFS se basa en una red de pares [5], donde cada nodo puede actuar como cliente y servidor al mismo tiempo, lo que significa que cada nodo puede descargar y distribuir archivos. Además, IPFS utiliza un sistema de incentivos para alentar a los nodos a mantener y distribuir archivos, lo que ayuda a garantizar la integridad y disponibilidad de los datos en la red. Daniel and Tschorsch (2022); Chen et al. (2017)

3.7 Herramientas

En el ámbito del desarrollo web3, el conjunto de herramientas a utilizar puede variar dependiendo de la naturaleza del proyecto y las habilidades y conocimientos del desarrollador. Sin embargo, hay algunas herramientas comunes que son ampliamente utilizadas en este campo.

Una de las herramientas principales es Solidity, un lenguaje de programación orientado a objetos diseñado específicamente para la creación de contratos inteligentes en la red Ethereum. Solidity es un lenguaje de alto nivel que permite a los desarrolladores escribir código de forma sencilla y legible, lo que facilita la creación de contratos inteligentes complejos. De esto se ha indagado más a lo largo de 3.4.1

3.7.1 Herramientas de desarrollo

3.7.1.1 Remix IDE

Remix IDE es una herramienta muy popular utilizada en la plataforma Ethereum para el desarrollo de contratos inteligentes. Como una IDE que se puede utilizar en el navegador web o en el escritorio, proporciona una interfaz gráfica de usuario para escribir, depurar y probar contratos inteligentes. Además, cuenta con un compilador integrado que permite compilar contratos en diferentes versiones de Solidity.

Para ampliar su funcionalidad, Remix IDE también ofrece una serie de plugins útiles. Por ejemplo, el plugin "Debug" permite depurar contratos paso a paso, lo que ayuda a los desarrolladores a identificar errores y solucionar problemas. Además, el plugin "Testing" permite escribir y ejecutar pruebas automatizadas para los contratos inteligentes. En resumen, Remix IDE es una herramienta completa y segura para el desarrollo de contratos inteligentes en Ethereum. [Khandelwal et al. (2021); Amir Latif et al. (2020)]

[5] Una red peer-to-peer, red de pares, red entre iguales o red entre pares es una red de ordenadores en la que todos o algunos aspectos funcionan sin clientes ni servidores fijos, sino una serie de nodos que se comportan como iguales entre sí Colaboradores de los proyectos Wikimedia (2023b)

3.7.1.2 Next JS

Next.js es un framework basado en React que permite a los desarrolladores crear aplicaciones web escalables y de alto rendimiento. Con su enfoque moderno y su amplia variedad de herramientas para el desarrollo web, Next.js ofrece una experiencia de desarrollo eficiente y optimizada.

Al utilizar una arquitectura de servidor y cliente, Next.js permite el procesamiento del lado del servidor (SSR) para una rápida carga inicial y un rendimiento óptimo, y también la opción de renderizado del lado del cliente (CSR) para interacciones dinámicas con el usuario. Además, Next.js cuenta con características integradas, como enrutamiento dinámico, optimización de imágenes y soporte para API externas, lo que simplifica el proceso de desarrollo de aplicaciones web complejas.

Next.js es una herramienta ampliamente utilizada en la industria y cuenta con una gran comunidad de desarrolladores que lo apoyan. Su enfoque en la escalabilidad y el rendimiento lo convierte en una herramienta valiosa para la construcción de aplicaciones web modernas y eficientes.[Thakkar and Thakkar (2020); Gackenheimer and Gackenheimer (2015)]

3.7.1.3 Hardhat

Hardhat es una herramienta de desarrollo de contratos inteligentes que se utiliza en la plataforma Ethereum. Ofrece un entorno de prueba integrado y una amplia variedad de plugins para tareas comunes de desarrollo de contratos inteligentes. Hardhat ha ganado popularidad debido a su enfoque en la simplicidad, la flexibilidad y la escalabilidad en comparación con Truffle, que es una herramienta más antigua y consolidada.

La herramienta de desarrollo de contratos inteligentes Hardhat es capaz de compilar contratos en cualquier versión de Solidity, lo que permite a los desarrolladores trabajar en proyectos que necesiten versiones antiguas o nuevas de Solidity. Hardhat también ofrece una serie de características útiles para trabajar con testnets de Ethereum, como la posibilidad de ejecutar pruebas en múltiples redes de prueba simultáneamente, la capacidad de automatizar la creación de cuentas y la asignación de tokens de prueba, y la integración con los principales proveedores de infraestructura de prueba. [Jain (2022)]

Entre las testnets más populares que se pueden utilizar con Hardhat, se encuentran Ropsten, Rinkeby y Kovan. Además de las testnets públicas, Hardhat también es compatible con la configuración de redes de prueba privadas, lo que puede ser útil para probar aplicaciones en un entorno controlado antes de lanzarlas en producción. En general, Hardhat es una herramienta muy útil para los desarrolladores de contratos inteligentes de Ethereum que buscan una solución escalable, flexible y fácil de usar para desarrollar y probar sus contratos inteligentes. Entre las testnets más populares que se pueden utilizar con Hardhat, se encuentran Ropsten, Rinkeby, Kovan y Goerli. [Iyer et al. (2018)]

3.7.1.4 Mumbai TestNet

La testnet de Mumbai, lanzada en marzo de 2021, es una red de prueba de la plataforma de contratos inteligentes de Ethereum conocida como Polygon (antes Matic Network), que permite a los desarrolladores probar sus aplicaciones y contratos antes de lanzarlos en la red principal. Es una réplica de la red principal de Polygon, pero con la diferencia clave de

que en la testnet de Mumbai, los tokens MATIC se emiten de forma gratuita para que los desarrolladores puedan experimentar con la plataforma sin preocuparse por los costos de las transacciones.[Iyer et al. (2018); Wöhrer and Zdun (2021)]

3.7.1.5 OpenZeppelin

OpenZeppelin es una biblioteca de contratos inteligentes seguros y reutilizables para Ethereum. La biblioteca ofrece una amplia gama de contratos inteligentes que los desarrolladores pueden utilizar para construir aplicaciones descentralizadas en la red Ethereum, lo que reduce el tiempo y los esfuerzos necesarios para implementar funcionalidades comunes de los contratos inteligentes. Entre los contratos inteligentes que ofrece OpenZeppelin se encuentran el token ERC20, el contrato inteligente de control de acceso RBAC y el contrato inteligente de recuperación de ether, entre otros. La plataforma de seguridad para contratos inteligentes Defender de OpenZeppelin ofrece monitoreo, alertas y automatización de operaciones críticas.

3.7.1.6 QuickNode

QuickNode es una plataforma de infraestructura blockchain que ofrece nodos de Ethereum y otras criptomonedas como servicio. Proporciona una solución de alojamiento de nodos que permite a los desarrolladores, empresas y usuarios individuales conectarse a la red blockchain sin preocuparse por la configuración, el mantenimiento y la seguridad de los nodos. QuickNode ofrece nodos de alta disponibilidad y escalables en múltiples regiones, lo que garantiza una conectividad y una latencia mínima para los usuarios. Además, ofrece una interfaz de usuario intuitiva para administrar los nodos, monitorear su rendimiento y acceder a las métricas de la red.[Qui (2023)]

3.7.1.7 Metamask

Metamask es una extensión de navegador que se utiliza para interactuar con aplicaciones descentralizadas y blockchain. Proporciona a los usuarios una billetera de criptomonedas que les permite almacenar y gestionar sus tokens y monedas digitales. Además, Metamask permite a los usuarios conectarse a diferentes redes de blockchain, como Ethereum y Binance Smart Chain, para interactuar con aplicaciones descentralizadas en esas redes. La función principal de Metamask es permitir que los usuarios firmen transacciones en la cadena de bloques. Cuando los usuarios realizan una transacción en una aplicación descentralizada, Metamask solicita al usuario que apruebe la transacción y, una vez aprobada, la transacción se envía a la cadena de bloques para ser procesada. Esto hace que las transacciones en la cadena de bloques sean seguras y fiables, ya que cada transacción debe ser aprobada por el propietario de la billetera. [Lee and Lee (2019)]

4 Plataforma

4.1 Plataforma de seguimiento de activos

Ahora que hemos revisado el marco teórico y analizado diversas soluciones existentes para el seguimiento de activos en blockchain, es el momento de profundizar en el desarrollo de nuestro trabajo. En esta sección, realizaremos un análisis general de las soluciones presentadas, identificando los elementos comunes y características clave que son esenciales para el seguimiento efectivo de activos en la tecnología blockchain. Abordaremos temas como los roles de los usuarios que interactúan en estas soluciones, las estructuras de datos empleadas para almacenar información de manera segura e inmutable, y cómo se gestionan las transacciones y la propiedad de los activos. Al comprender estos elementos fundamentales, podremos establecer una base sólida para diseñar y desarrollar una solución innovadora y eficaz que aborde las necesidades y desafíos específicos en el ámbito del seguimiento de activos en blockchain.

4.1.1 Roles y actores en el seguimiento de activos en blockchain

4.1.1.1 Emisores y actualizadores de los activos

Los emisores de activos son aquellos actores clave involucrados en la creación, distribución, y gestión de los activos que se van a trazar. Estos emisores desempeñan un papel fundamental en garantizar la autenticidad, integridad y transparencia de la información asociada a los activos. Entre ellos, se encuentran fabricantes y productores, responsables de generar la información inicial de los activos y asignar identificadores únicos. Las entidades de transporte, proveedores de servicios y mantenimiento, que registran actividades de reparación y actualización de activos a lo largo del tiempo, además proporcionan información sobre el estado y ubicación de los activos.

4.1.1.2 Propietarios de activos

Los propietarios de los activos son individuos o entidades que poseen o ejercen derechos sobre bienes y servicios que requieren trazabilidad en el tiempo. Estos propietarios son fundamentales en el ecosistema de trazabilidad, ya que son responsables de mantener y actualizar la información relevante de los activos durante su periodo de propiedad. Los propietarios pueden ser tanto consumidores finales como empresas o instituciones que adquieren, utilizan y transfieren activos a lo largo de su vida útil. La información de trazabilidad es esencial para los propietarios de activos, ya que les permite tomar decisiones informadas sobre la gestión, mantenimiento, y eventual venta o transferencia de sus bienes. Además, la transparencia y accesibilidad de esta información les otorgan mayor confianza en la autenticidad y calidad de los activos, lo que a su vez puede influir en el valor percibido y la demanda de estos bienes en el mercado. Para garantizar la integridad y veracidad de la información de trazabilidad, los propietarios de activos deben colaborar estrechamente con emisores, intermediarios

y otras partes involucradas en el ecosistema, asegurando un flujo de información transparente y actualizado en todo momento.

4.1.1.3 Role Based Access Control OpenZeppelin

La implementación de Role-Based Access Control (RBAC) en soluciones de trazabilidad basadas en blockchain, como las proporcionadas por OpenZeppelin, puede ser fundamental para garantizar la seguridad y eficiencia en la gestión de activos. RBAC permite asignar roles específicos a los diferentes actores involucrados en el ecosistema, como emisores de activos y propietarios de activos, lo que facilita la interacción y colaboración entre ellos al definir claramente sus responsabilidades y permisos.[Chatterjee et al. (2020)]

Por ejemplo, los emisores de activos podrían tener el rol de crear y emitir nuevos activos en la cadena de bloques, mientras que los propietarios de activos podrían tener permisos para actualizar la información relacionada con los bienes que poseen y transferir la propiedad de los mismos a otros propietarios. Con RBAC de OpenZeppelin, es posible establecer políticas de acceso granulares y dinámicas que aseguran que cada actor solo tenga acceso a las funciones y datos que le corresponden, protegiendo la integridad del sistema y minimizando el riesgo de abusos o manipulaciones indebidas.

Código 4.1: Ejemplo definición de roles

```
1   //Role para creadores de automoviles y matriculadoras
2   bytes32 public constant CARFACTORY_ROLE = keccak256("CARFACTORY_ROLE");
3   //Role para garages y ITVs
4   bytes32 public constant GARAGE_ROLE = keccak256("GARAGE_ROLE");
5   //Role para administradores del sistemas
6   bytes32 public constant SYSADMIN_ROLE = keccak256("SYSADMIN_ROLE");
7
8   constructor() {
9     // Asigna el rol CARFACTORY_ROLE al creador del contrato
10    _setupRole(CARFACTORY_ROLE, msg.sender);
11    // Asigna el rol GARAGE_ROLE al creador del contrato
12    _setupRole(GARAGE_ROLE, msg.sender);
13    // Asigna el rol SYSADMIN_ROLE al creador del contrato
14    _setupRole(SYSADMIN_ROLE, msg.sender);
15  }
```

4.1.2 Estructuras de datos y almacenamiento de información

Al abordar la implementación de soluciones de trazabilidad y gestión de activos en Solidity y en el contexto de la blockchain, es crucial considerar las estructuras de datos y enfoques de almacenamiento utilizados para gestionar la información de manera eficiente y segura. Solidity ofrece diversas estructuras de datos, como arrays, mappings y structs, que pueden utilizarse para modelar y almacenar la información de los activos y sus interacciones en el tiempo.

4.1.2.1 Tipos de datos válidos

Los más comúnmente utilizados son los tipos de datos primitivos como los números enteros (uint, int), las cadenas de caracteres (string), las direcciones (address), y los booleanos (bool).

También hay tipos de datos más complejos, como las estructuras (struct), que permiten agrupar varios campos de datos en una sola entidad, y los mapeos (mapping), que son colecciones clave-valor y funcionan de manera similar a los diccionarios o los hash maps. Al seleccionar los tipos de datos, es esencial considerar factores como el coste del gas asociado, la legibilidad y la seguridad del contrato. [Dannen (2017)]

4.1.2.2 El uso del IPFS

Dado que almacenar grandes cantidades de datos directamente en la cadena de bloques puede ser costoso en términos de gas, IPFS ofrece una solución descentralizada y eficiente para almacenar y recuperar datos. Los archivos en IPFS se almacenan en una red de nodos distribuidos y se accede a ellos a través de identificadores únicos basados en el contenido. En el contexto de la trazabilidad de activos, IPFS puede utilizarse para almacenar información detallada del activo, como descripciones, historiales de transacciones, imágenes, entre otros. Luego, estos identificadores únicos pueden ser almacenados en la blockchain junto con el resto de la información del activo, proporcionando un enlace seguro y permanente a la información completa del activo.

4.1.2.3 Estructuras de datos en Solidity y su paridad con la realidad

Las estructuras de datos en Solidity permiten una modelización precisa y eficiente de los activos y sus interacciones en el mundo real. A través de los tipos de datos y estructuras ofrecidos por Solidity, podemos representar los diversos aspectos y características de un activo.

Por ejemplo, una estructura (struct) puede ser utilizada para modelar un activo, con campos correspondientes a sus características distintivas. Un vehículo, por ejemplo, puede tener campos para su número de VIN, marca, modelo, año de fabricación, y el historial de propietarios. Los mapeos (mapping) pueden ser utilizados para rastrear la propiedad de un activo, mapeando una dirección de Ethereum a un activo específico, o viceversa.

Código 4.2: Ejemplo definición de de estructuras

```
library CarData {
    struct Repair {
        string repairType;
        uint repairDate;
        string description;
    }

    struct Accident {
        string accidentType;
        uint accidentDate;
        string description;
    }

    struct Photos {
        string frontalPhoto;
        string rightSidePhoto;
        string leftSidePhoto;
        string backSidePhoto;
    }
}
//Mapping de matriculas y address de los contratos
mapping(string => address) public licensePlateToCar;
```

4.1.3 Patrón Contract Factory

El patrón Contract Factory en Solidity es un patrón de diseño que permite la creación dinámica de contratos dentro de otros contratos. Una ”fábrica de contratos” es, en esencia, un contrato que actúa como un generador de otros contratos. Cada activo puede estar representado por su propio contrato, y la fábrica de contratos sirve como un directorio centralizado para la gestión y el seguimiento de todos estos contratos de activos. Con el patrón Contract Factory, es posible desplegar nuevos contratos de activos a medida que se necesiten, sin tener que modificar o actualizar los contratos existentes. Esto es especialmente relevante en escenarios de trazabilidad de activos, donde la cantidad de activos puede variar y aumentar con el tiempo. Cada contrato de activo puede mantener su propio historial de estados, lo que facilita la trazabilidad. Cada vez que se actualiza el estado de un activo, esta actualización se registra en la cadena de bloques como una transacción en el contrato del activo correspondiente. Esto proporciona un registro inmutable y comprobable de la vida útil del activo, lo que es esencial para la trazabilidad.

Aunque desplegar un nuevo contrato consume más gas que realizar una operación en un contrato existente, el patrón Contract Factory puede seguir siendo más eficiente en términos de gas en el contexto de la trazabilidad de activos. Esto se debe a que cada contrato de activo es independiente y sólo necesita manejar la lógica y los datos relacionados con su activo específico. Esto contrasta con un enfoque en el que se utilizan estructuras de datos complejas en un contrato monolítico para rastrear múltiples activos, lo que puede resultar en operaciones de gas costosas. [Wohrer and Zdun (2018)]

4.1.3.1 ¿Por qué no usar NFTs?

Aunque los NFTs (Non-Fungible Tokens) son excelentes para representar la propiedad única de activos en la blockchain, presentan limitaciones para la trazabilidad de activos. Están restringidos por el estándar ERC721, lo que limita la personalización y puede causar problemas de escalabilidad cuando el número de activos aumenta. Además, albergan todos los datos en un solo contrato, aumentando la exposición a errores y ataques, y su diseño no es óptimo para rastrear la historia de un activo a lo largo del tiempo. Por el contrario, el patrón Contract Factory ofrece mayor flexibilidad, escalabilidad, segregación de datos y capacidades de rastreo histórico, haciéndolo una opción más efectiva para la trazabilidad de activos.

4.1.3.2 Tabla comparativa

4.1.4 Escalabilidad y rendimiento

En blockchain, la escalabilidad se refiere a la capacidad de la red para manejar y procesar un número creciente de transacciones. En el contexto de Ethereum y Solidity, este es un aspecto crítico debido a los límites de gas para las transacciones y la capacidad de la red para procesarlas. Si la implementación no está optimizada, las transacciones pueden consumir grandes cantidades de gas y ser prohibitivas en términos de costos.

Característica	Contract Factory	NFTs (ERC721)
Personalización	Alta: permite estructuras de datos y lógica personalizada.	Baja: restringida al estándar ERC721.
Escalabilidad	Alta: cada activo es un contrato individual, lo que permite un número ilimitado de activos.	Media: puede haber problemas con un gran número de activos.
Segregación de datos	Alta: cada activo tiene su propio contrato y almacena sus datos de forma independiente.	Baja: todos los datos se almacenan en un solo contrato.
Rastreo histórico	Alta: cada contrato puede llevar un registro completo de su historia.	Baja: no está diseñado para rastrear la historia de un activo a lo largo del tiempo.
Resistencia a errores y ataques	Alta: el fallo en un contrato no afecta a los demás.	Media: un error o ataque puede comprometer todos los activos en el contrato.
Representación de propiedad	Sí: cada contrato representa un activo.	Sí: cada token representa un activo único.

El rendimiento, por otro lado, se refiere a la velocidad a la que se procesan las transacciones y se confirman en la red. En Ethereum, esto se ve influenciado por factores como la congestión de la red y la cantidad de gas ofrecida por las transacciones.

4.2 Caso particular: plataforma de seguimiento de vehículos

Una vez explicados los conceptos necesarios para cualquier seguimiento de activos en la blockchain utilizando Ethereum y sus herramientas de desarrollo, vamos a aplicarlo a un caso práctico, construyendo una aplicación para la trazabilidad de los vehículos, para ello crearemos los contratos inteligentes necesarios, los desplegaremos y verificaremos para mayor transparencia, una vez hecho esto generaremos un front-end utilizando el framework de Nextjs, para que los usuarios de esta cadena puedan interactuar de manera cómoda con los activos.

4.2.1 Agentes implicados

Los agentes implicados dentro de nuestra solución, serán todos aquellos interesados en participar en el uso de la aplicación, ya sea de forma activa en la creación de nuevos activos, a forma de intermediario, modificando los activos y ofreciendo actualizaciones de los mismos, a forma de consultoría gratuita para poder consultar el estado de un activo, o por último como dueño de un activo pudiendo transferir los mismos.

- **Casa de automóviles (CarFactory):** Las casas de automóviles o fabricantes de los mismos serán los únicos capaces de crear nuevos activos en la blockchain, al crear el automóvil estos podrán asignárselos tanto así mismos como a un dueño.
- **Dueño del vehículo:** Los dueños de los vehículos tendrán la potestad de transaccionar con su vehículo, cambiando la dirección de la wallet de dueño
- **Taller (Garage):** Estos talleres tendrán la potestad de añadir reparaciones y accidentes a los datos del vehículo, solo aquellos talleres de confianza serán capaces de añadir esta información.
- **Usuarios:** Los usuarios de la aplicación serán capaces de visualizar toda la información deseada sobre el vehículo en cuestión, para ello solo tendrán que ingresar la matrícula del vehículo.
- **Administrador (SysAdmin):** Los administradores de la red tendrán todas las capacidades, este rol es necesario que sea desempeñado por una persona o grupo de personas de confianza.

4.2.2 Funcionalidades

Las siguientes tablas presentan un resumen estructurado de las funcionalidades de un contrato inteligente basado en la plataforma de blockchain Ethereum. Este contrato inteligente en particular está diseñado para administrar una serie de operaciones relacionadas con automóviles.

4.2.2.1 Parte pública

Las funciones públicas son aquellas que cualquier usuario puede ejecutar, como obtener información sobre un automóvil específico a través de su matrícula, comprobar el propietario actual del automóvil, o consultar la lista de accidentes o reparaciones que ha tenido un automóvil.

Tabla 4.1: Descripción de las Funcionalidades del Contrato Inteligente

ID	Funcionalidad	Descripción
1	isAFactory	Permite verificar si el remitente es una fábrica de automóviles
2	isAGarage	Permite verificar si el remitente es un garaje
3	isAAdmin	Permite verificar si el remitente es un administrador

Continúa en la página siguiente

Tabla 4.1 – continúa desde la página anterior

ID	Funcionalidad	Descripción
4	getCarByLicensePlate	Devuelve la dirección del contrato del automóvil asociado a la matrícula dada
5	getNumberOfCars	Devuelve el número total de automóviles en el sistema
6	getMaker	Devuelve el fabricante del automóvil por su matrícula
7	getModel	Devuelve el modelo del automóvil por su matrícula
8	getRegistrationDate	Devuelve la fecha de registro del automóvil por su matrícula
9	getKilometrajeHistory	Devuelve el historial de kilometraje del automóvil por su matrícula
10	getYear	Devuelve el año del automóvil por su matrícula
11	getReparationOfCar	Devuelve la lista de reparaciones del automóvil por su matrícula
12	getAccidentOfCar	Devuelve la lista de accidentes del automóvil por su matrícula
13	getActualOwnerOfCar	Devuelve el propietario actual del automóvil por su matrícula
14	getPhotosOfCar	Devuelve las fotos del automóvil por su matrícula
15	setNewOnwerOfCar	Permite al propietario actual de un automóvil establecer un nuevo propietario

4.2.2.2 Parte privada

Las funciones privadas, están restringidas a ciertos roles definidos en el contrato. Estos roles incluyen el de ”CARFACTORY_ROLE”, ”GARAGE_ROLE” y ”SYSADMIN_ROLE”, que

corresponden a diferentes tipos de usuarios con diferentes niveles de acceso al sistema. Los usuarios con estos roles pueden realizar acciones más privilegiadas, como crear un nuevo automóvil en el sistema, añadir reparaciones o accidentes a un automóvil, y establecer nuevas fotos o fechas de registro para un automóvil.

4.2.2.3 Funciones para CARFACTORY_ROLE y SYSADMIN_ROLE:

Tabla 4.2: Descripción de las Funcionalidades para los roles CarFactory y SysAdmin

ID	Funcionalidad	Descripción
1	createCar	Permite a los usuarios con el rol CARFACTORY_ROLE o SYSADMIN_ROLE crear un nuevo automóvil y añadirlo al sistema
2	getAllLicensePlates	Devuelve un array con todas las matrículas de los automóviles en el sistema
3	setPhotosOfCar	Permite a los usuarios con el rol CARFACTORY_ROLE o SYSADMIN_ROLE establecer fotos para un automóvil
4	setRegistrationDate	Permite a los usuarios con el rol CARFACTORY_ROLE o SYSADMIN_ROLE establecer una nueva fecha de registro para un automóvil

4.2.2.4 Funciones para GARAGE_ROLE y SYSADMIN_ROLE

Tabla 4.3: Descripción de las Funcionalidades para los roles Garage y SysAdmin

ID	Funcionalidad	Descripción
Continúa en la página siguiente		

Tabla 4.3 – continúa desde la página anterior

ID	Funcionalidad	Descripción
1	addRepairToCar	Permite a los usuarios con el rol GARAGE_ROLE o SYSADMIN_ROLE añadir una reparación a un automóvil
2	addAccidenteToCar	Permite a los usuarios con el rol GARAGE_ROLE o SYSADMIN_ROLE añadir un accidente a un automóvil
3	addKilometrajeToCar	Permite a los usuarios con el rol GARAGE_ROLE o SYSADMIN_ROLE añadir kilometrajes a un vehículo según su año

4.2.2.5 Funciones para SYSADMIN_ROLE

Tabla 4.4: Descripción de las Funcionalidades para el rol SysAdmin

ID	Funcionalidad	Descripción
1	getAllCars	Devuelve un array con todas las direcciones de los contratos de automóviles
2	deleteCar	Esta función hace un soft delete de la eliminación del vehículo en la blockchain

4.3 Implementación

Como se ha explicado a lo largo de los apartados anteriores, la implementación constará de tres contratos inteligentes, los cuales se relacionarán usando el patrón anteriormente descrito.

4.3.1 Car Factory

Este contrato será el único desplegado de manera directa en la blockchain, y será verificado [1] para mayor confianza de los usuarios tras su despliegue. CarFactory mantiene un registro de todos los contratos Car creados y proporciona funciones para crear nuevos contratos Car, así como para consultar y eliminar contratos existentes. Además, implementa un sistema de roles para restringir ciertas acciones, como la creación y eliminación de contratos Car, a usuarios autorizados como administradores del sistema o fabricantes de automóviles.

4.3.1.1 Herencia Openzeppelin

AccessControl es una biblioteca proporcionada por OpenZeppelin que proporciona una estructura para controlar el acceso a las funciones de los contratos inteligentes. Esto es especialmente útil para limitar la funcionalidad sensible a ciertos roles.

En el contrato CarFactory, AccessControl se utiliza para establecer roles que tienen permisos específicos. Al utilizar AccessControl, puedes establecer diferentes roles, asignar esos roles a las cuentas y proteger las funciones de tu contrato para que sólo las cuentas con el rol adecuado puedan ejecutarlas.

Por ejemplo, en el contrato CarFactory, puedes ver líneas como esta:

Código 4.3: Ejemplo de require dentro de Factoria

```
require(hasRole(CARFACTORY_ROLE, msg.sender)|| hasRole(SYSADMIN_ROLE,msg.sender), "You can
    ↪ not create a Car");
```

Esta línea está utilizando la función *hasRole* de AccessControl para comprobar si la cuenta que está intentando ejecutar la función tiene el rol CARFACTORY_ROLE o SYSADMIN_ROLE. Si no es así, la transacción se revertirá con el mensaje "You can not create a Car".

4.3.2 Car

Almacena detalles importantes del vehículo como la marca, el modelo, el año de fabricación, la placa de matrícula y la fecha de registro. También mantiene un registro de la dirección de la billetera del dueño actual y un historial de los dueños anteriores. Además, tiene la funcionalidad para actualizar y almacenar las imágenes del coche y cambiar la propiedad del coche

4.3.3 Car Data

Se utiliza para almacenar y gestionar los datos relacionados con un coche específico en la cadena de bloques. Estos datos incluyen la historia del kilometraje del coche, los detalles de los accidentes y las reparaciones que ha tenido. Este contrato permite agregar nuevas entradas a la historia del kilometraje, accidentes y reparaciones del coche, proporcionando un registro descentralizado y verificable de la historia del vehículo. El contrato CarData se

[1] Verificar un contrato en PolygonScan (o Etherscan para la red Ethereum) significa subir el código fuente de tu contrato inteligente a la plataforma. Esto permite a cualquier persona leer y comprender las funciones y operaciones que el contrato puede realizar.

relaciona directamente con un contrato Car específico, proporcionando una capa adicional de detalles y funcionalidad a cada coche en la plataforma.

4.3.4 Aclaraciones sobre la implementación

4.3.4.1 Autodestrucción de contratos

selfdestruct es una función en Solidity, el lenguaje de programación de los contratos inteligentes en Ethereum. Esta función permite destruir un contrato y enviar sus fondos a una dirección especificada. Cuando se llama a la función **selfdestruct**, el contrato deja de existir en la cadena de bloques después de que se ha ejecutado completamente la transacción actual. Esto significa que ya no se puede interactuar con él, no se pueden ejecutar más funciones y el código de bytecode se elimina del estado de la blockchain. Esta sería la manera más segura de eliminar los datos de los vehículos pero reduciría en gran medida la transparencia completa de la aplicación.

4.3.5 Flujo de trabajo de interacción con la aplicación

4.3.5.1 Flujo creación de activos

El rol *CarFactory*, como se ha explicado en secciones anteriores, es el único con la potestad para generar activos dentro de la lógica de los contratos inteligentes desarrollados, para generar un nuevo vehículo de manera sencilla, el usuario/cartera con el rol asignado, podrá visualizar dentro de la web un formulario para generar un nuevo automóvil, donde se introducirán los datos pertinentes al vehículo, una vez enviada la información se le abrirá la wallet de metamask donde deberá confirmar la transacción y pagar por ella. Una vez se ha enviado la transacción, el contrato inteligente denominado *CarFactory*, este utilizando el patrón *contract factory*, generará un nuevo contrato donde se almacenará la información del activo automovilístico, este contrato generado contendrá los métodos para actualizar y modificar la información pertinente al activo. A modo de interfaz nuestro contrato factoría contendrá todos los métodos pertinentes para interactuar con el resto de los contratos, almacenando las direcciones en un *mapping* a las cuales accederemos mediante las matrículas pertinentes a cada vehículo.

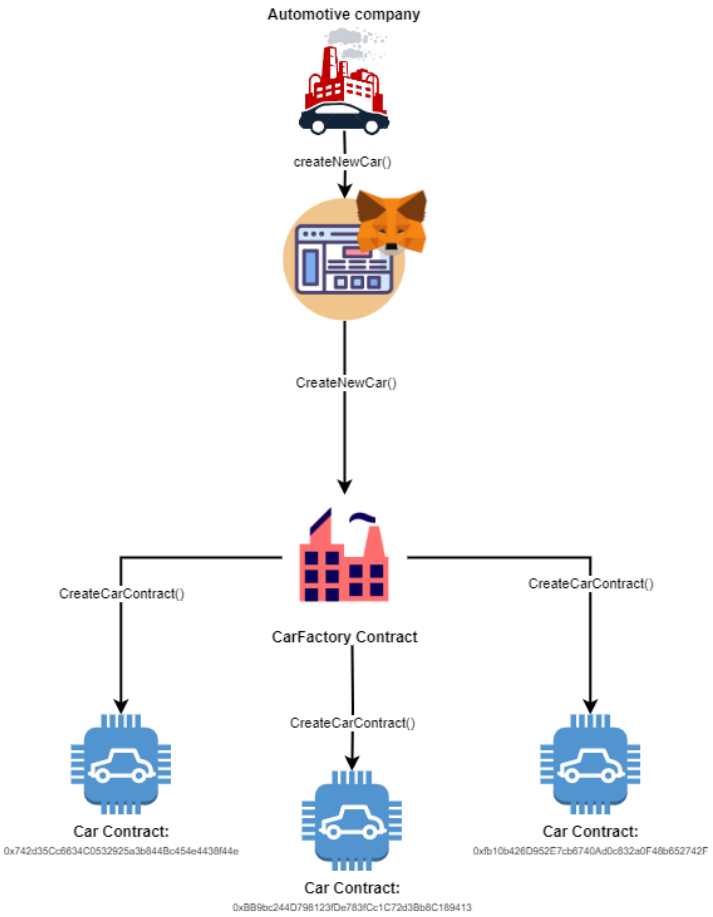

Figura 4.1: Flujo de trabajo rol CarFactory

4.3.5.2 Flujo de modificaciones de activos

El rol de *Garage*, será el que utilicen los talleres autorizados para añadir información a los vehículos, para ello al igual que las factorías, utilizarán unos formularios que solo serán capaces de visualizar aquellos usuarios/carteras con el rol asignado. Se han proporcionado tres formularios, para realizar las acciones de: añadir reparaciones, añadir accidentes y añadir kilometrajes, al igual que con la adición de nuevos activos, el contrato factoría actúa como interfaz para añadir información a los vehículos guardados en el *mapping*

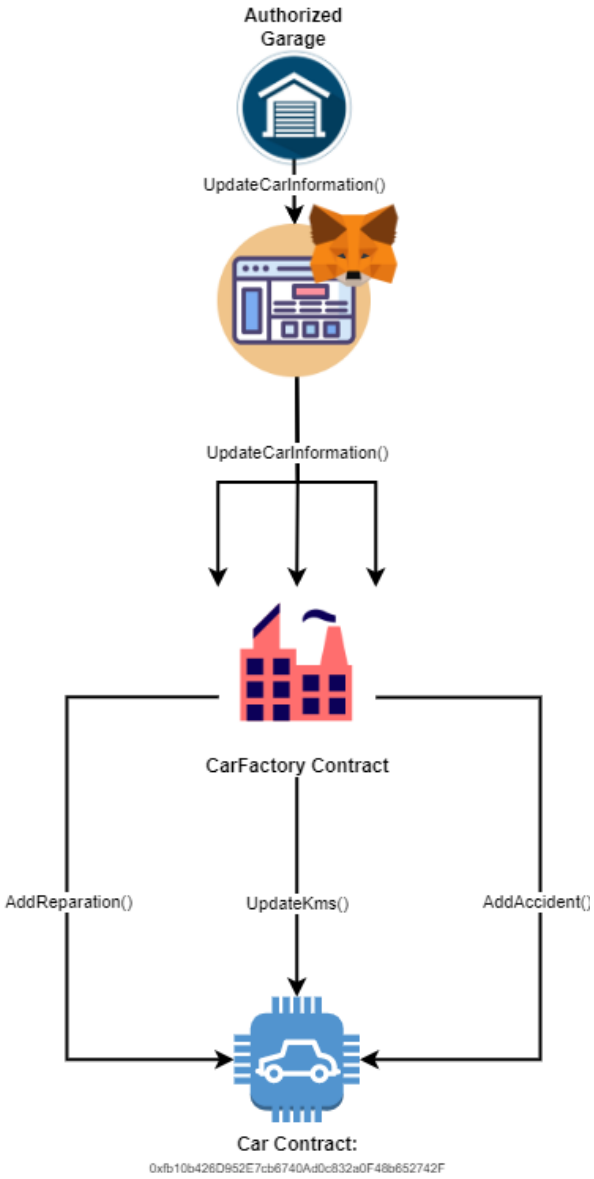

Figura 4.2: Flujo de trabajo rol Garage

4.3.5.3 Flujo extracción de información de activos

Los usuarios para poder acceder a la información pública almacenada en los contratos inteligentes, deberán conectar su cartera de metamask utilizando la red de mumbai, los usuarios podrán escribir la matrícula del vehículo que desean obtener información, el contrato factoría de nuevo actuará de interfaz buscando en el *mapping*, si la información del vehículo se encuentra disponible se dispondrá en la interfaz del usuario. Además, si el usuario es el dueño del vehículo que ha buscado en la aplicación, podrá transferir el vehículo a otro usuario perdiendo la potestad del mismo.

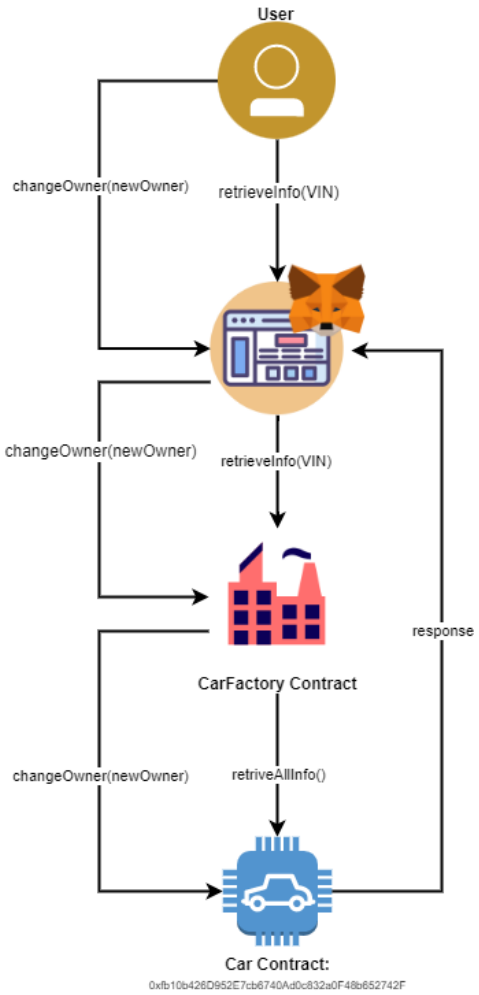

Figura 4.3: Flujo de trabajo rol usuario

4.4 Diseño

4.4.1 Mockups

Uno de los objetivos definidos en 1.3.3 es la creación de una una interfaz sencilla y segura, para que los usuarios se sientan cómodos interactuando con la blockchain, para ello se ha realizado una interfaz web sencilla y práctica para los usuarios, tanto con los roles anteriormente mencionados, como para los usuarios que solo quieren conocer información sobre ciertos vehículos.

Esta será la vista inicial de la aplicación, una vez los usuarios hayan entrado, al pulsar sobre el botón de "Connect your Wallet" se les abrirá su cartera de Metamask, desde la cual podrán interactuar con la Blockchain y las funcionalidades de los contratos desarrollados.

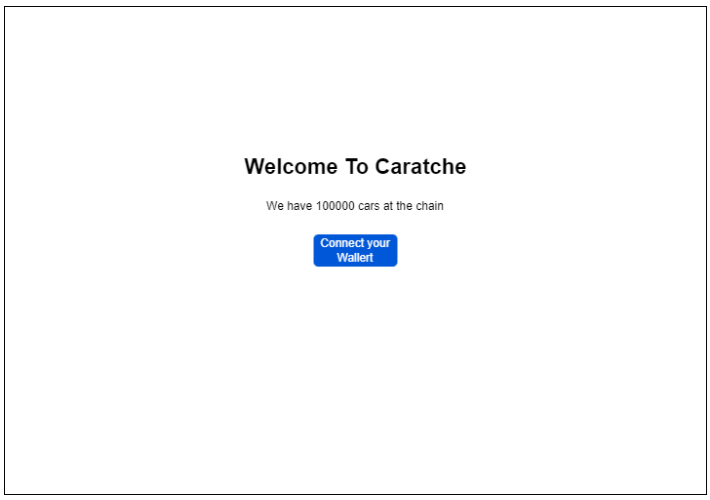

Figura 4.4: Mockup cartera sin conectar, inicio web

Una vez se haya conectado la Wallet del usuario al contrato, se dispondrá ante él, si no tiene ningún rol asignado, la siguiente barra de búsqueda, donde podrá escribir la matrícula del vehículo del cual quiere saber su información detallada. Si el vehículo se encuentra registrado, se dispondrá ante el usuario la información.

Figura 4.5: Mockup cartera conectada, barra búsqueda

De esta forma se dispondrá la información del vehículo delante del usuario, mostrando datos como: modelo, fabricante, año de matriculación, año de registro en la plataforma, una gráfica de como ha evolucionado el historial de kilometrajes del vehículo a lo largo de los años, el historial de accidentes, historial de reparaciones e imágenes del vehículo.

Figura 4.6: Mockup información del vehículo

Por último se ha realizado el mockup de una vista con roles definidos, en ella se dispondrán formularios en la aplicación y dependiendo de los roles asignados se podrán ver unos u otros.

Figura 4.7: Mockup cartera sin conectar, inicio web

4.4.2 Infraestructura y Despliegue

Esta es una configuración típica para el desarrollo de aplicaciones de blockchain con la cadena de bloques Ethereum (o una cadena compatible con Ethereum como Polygon/Matic).

Código 4.4: Infraestructura necesaria para desarrollo

```
require("@nomicfoundation/hardhat-toolbox");
require("dotenv").config({ path: ".env" });

const QUICKNODE_HTTP_URL = process.env.QUICKNODE_HTTP_URL;
const PRIVATE_KEY = process.env.PRIVATE_KEY;
const POLYGONSCAN_API_KEY = process.env.POLYGONSCAN_API_KEY;

module.exports = {
  solidity: {
    version: "0.8.17",
    settings: {
      optimizer: {
        enabled: true,
        runs: 200
      },
      outputSelection: {
        "*": {
          "*": ["!revertStrings"]
        }
      }
    }
  },
  networks: {
    mumbai: {
      url: QUICKNODE_HTTP_URL,
      accounts: [PRIVATE_KEY],
    },
  },
  etherscan: {
    apiKey: POLYGONSCAN_API_KEY,
  },
};
```

- **QUICKNODE_HTTP_URL:** Esta es la URL de tu proveedor de nodos de blockchain, en este caso QuickNode. QuickNode es un servicio que proporciona nodos de blockchain ya montados y listos para usar en diferentes redes de blockchain. Esto elimina la necesidad de montar y mantener tu propio nodo de Ethereum, lo que puede ser complicado y costoso. Se necesita esta URL para que tu aplicación pueda interactuar con la red de blockchain.

- **PRIVATE_KEY:** Es la clave privada de una cuenta de Ethereum. En este caso, se está utilizando para configurar una cuenta que se utilizará para desplegar contratos inteligentes en la red de Ethereum y realizar otras transacciones. Es importante mantener la clave privada segura, ya que cualquiera que la posea puede realizar transacciones desde esa cuenta de Ethereum.

- **POLYGONSCAN_API_KEY:** PolygonScan es una herramienta que permite explorar y buscar transacciones, direcciones, contratos, tokens y otros datos en la red de Polygon (antes Matic). La API de PolygonScan permite integrar esta funcionalidad en

otras aplicaciones. Se necesita una clave de API para autenticar las solicitudes a esta API.

Este script realiza varias tareas relacionadas con el despliegue de un contrato inteligente CarFactory en la red Ethereum (o una red compatible con Ethereum) usando la biblioteca de JavaScript ethers.js y el paquete de desarrollo de contratos inteligentes Hardhat.

Código 4.5: Script de despliegue

```
const { ethers } = require("hardhat");
const fs = require('fs');
require("dotenv").config({ path: ".env" });
require("@nomiclabs/hardhat-etherscan");

async function main() {
  const CarFactoryContract = await ethers.getContractFactory("CarFactory");

  // deploy the contract
  const deployedCarFactoryContract = await CarFactoryContract.deploy();

  // Wait for it to finish deploying
  await deployedCarFactoryContract.deployed();

  // If contract is not being deployed to the localhost network, verify.

  console.log("Verifying contract, waiting 6 tx for propagation...");

  await deployedCarFactoryContract.deployTransaction.wait(6);

  await hre.run("verify:verify", {
    address: deployedCarFactoryContract.address,
    constructorArguments: [],
  });

  // print the address of the deployed contract
  console.log(
    "CarFactory contract address:",
    deployedCarFactoryContract.address
  );



  const data = deployedCarFactoryContract.address;

  fs.writeFile('address.txt', data, (error) => {
    if (error) {
      console.error('Ocurrio un error al escribir en el archivo:', error);
    } else {
      console.log('El archivo address.txt ha sido creado y actualizado exitosamente.');
    }
  });

}

// Call the main function and catch if there is any error
main()
  .then(() => process.exit(0))
  .catch((error) => {
    console.error(error);
    process.exit(1);
  });
```

1. Primero, obtiene la ”factoría de contratos” para CarFactory. Esta es una clase que puede crear instancias de contratos de este tipo.

2. Luego despliega una instancia de CarFactory en la red blockchain utilizando la función deploy(). El contrato se despliega desde la cuenta Ethereum predeterminada establecida en la configuración de Hardhat.

3. Luego espera a que se complete el despliegue del contrato utilizando deployed(). Una vez que se ha desplegado el contrato, se puede interactuar con él a través de la red Ethereum.

4. A continuación, el script espera a que se hayan confirmado al menos 6 bloques después de la transacción de despliegue. Esto se hace para asegurar que la transacción de despliegue está suficientemente ”profunda” en la cadena de bloques y es poco probable que sea revertida debido a una reorganización de la cadena.

5. El script luego verifica el contrato en Etherscan (o un explorador de blockchain similar) utilizando el plugin hardhat-etherscan. La verificación hace que el código fuente del contrato sea visible en Etherscan, lo que facilita la interacción con el contrato y permite que otros puedan auditar el código del contrato.

6. Después de eso, el script imprime la dirección en la que se desplegó el contrato CarFactory.

7. Por último, el script escribe la dirección del contrato desplegado en un archivo address.txt para que sea fácil de recordar y acceder en el futuro.

8. Si se encuentra algún error en cualquiera de estos pasos, el script capturará el error, lo imprimirá en la consola y terminará el proceso con un código de error. De lo contrario, el proceso termina con éxito.

5 Despliegue y Resultados de la Implementación

Este capítulo está dedicado a discutir los resultados y observaciones tras la implementación y despliegue de los contratos inteligentes y la interfaz de usuario de nuestra aplicación. Después de meticulosas fases de diseño y desarrollo, finalmente hemos desplegado nuestros contratos inteligentes en la red de pruebas de Mumbai en la cadena de bloques de Polygon. Este capítulo está dedicado a discutir los resultados y observaciones tras la implementación y despliegue de los contratos inteligentes y la interfaz de usuario de nuestra aplicación. Después de meticulosas fases de diseño y desarrollo, finalmente hemos desplegado nuestros contratos inteligentes en la red de pruebas de Mumbai en la cadena de bloques de Polygon, y hemos conectado nuestros smart contracts con la aplicación.

La finalidad de este capítulo es analizar de forma crítica el rendimiento de nuestros contratos inteligentes en un entorno de blockchain en vivo y cómo estos interactúan con la interfaz de usuario para lograr nuestros objetivos originales.

5.1 Imagenes de los resultados

Como ya hemos definido en apartados anteriores, lo primero que veremos al iniciar la aplicación será la pantalla de bienvenida, donde deberemos conectar nuestra wallet, en este caso Metamask para poder interactuar con la blockchain.

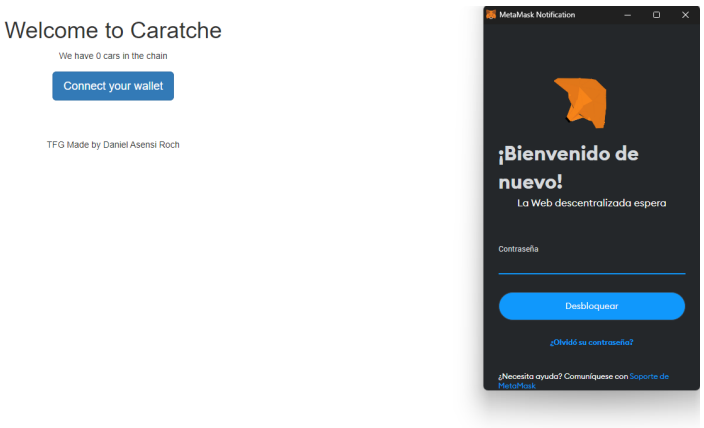

Figura 5.1: FrontEnd cartera sin conectar

Una vez iniciemos sesión con nuestra Wallet dependiendo de los roles que se hayan definido a nuestra address, se dispondrán ante nosotros una serie de formularios que podremos rellenar, si poseemos todos los roles como un administrador de la aplicación nuestra vista será la siguiente:

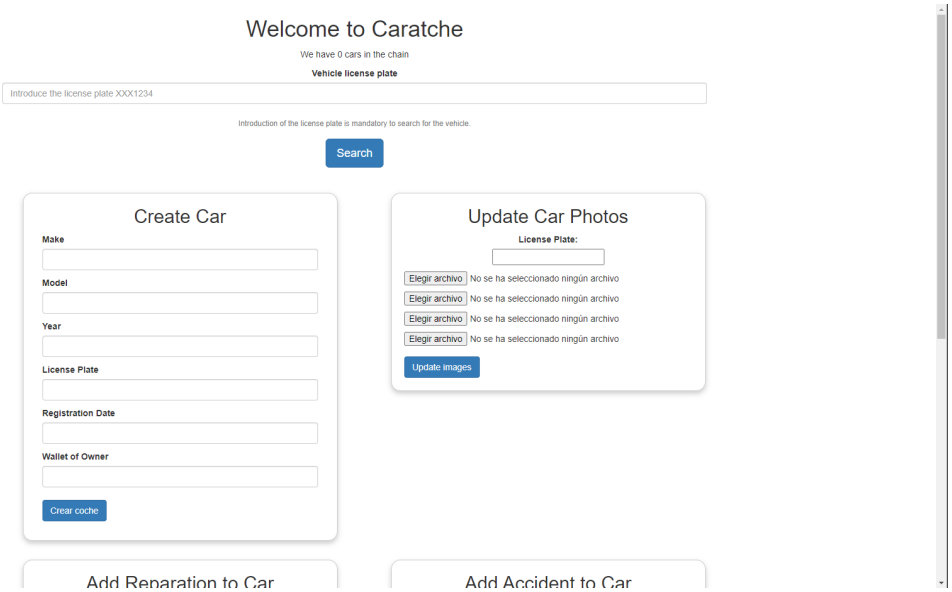

Figura 5.2: Vista administrador del sistema 1

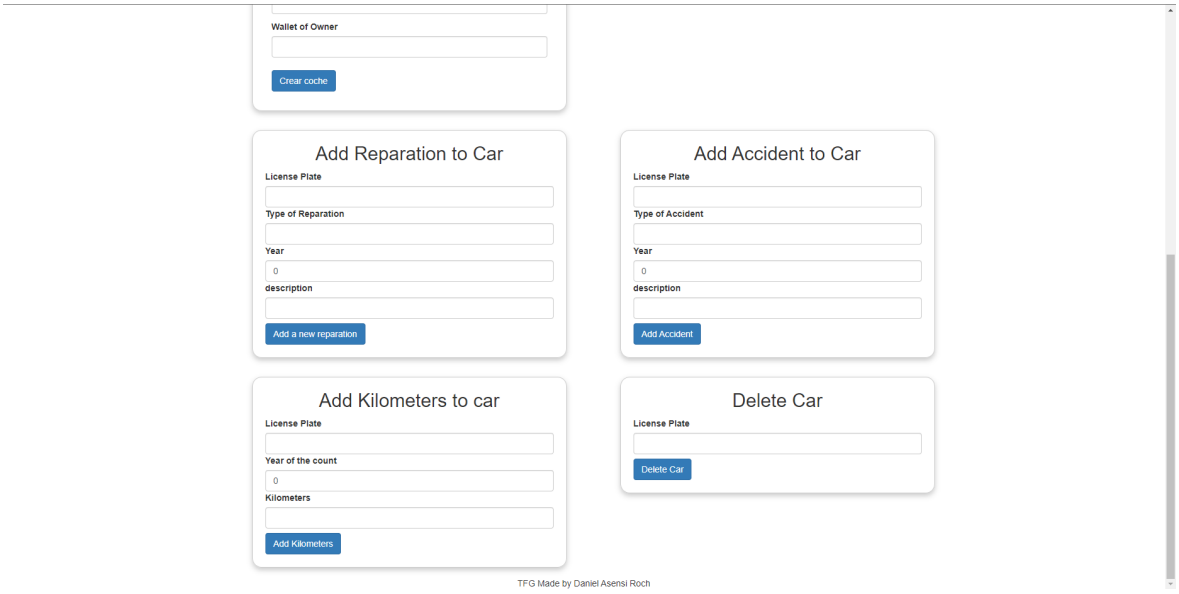

Figura 5.3: Vista administrador del sistema 2

Se ha rellenado la información completa de un vehículo, para mostrar que toda la información es accesible desde la blockchain. No se muestra como se han rellenado todos los formularios

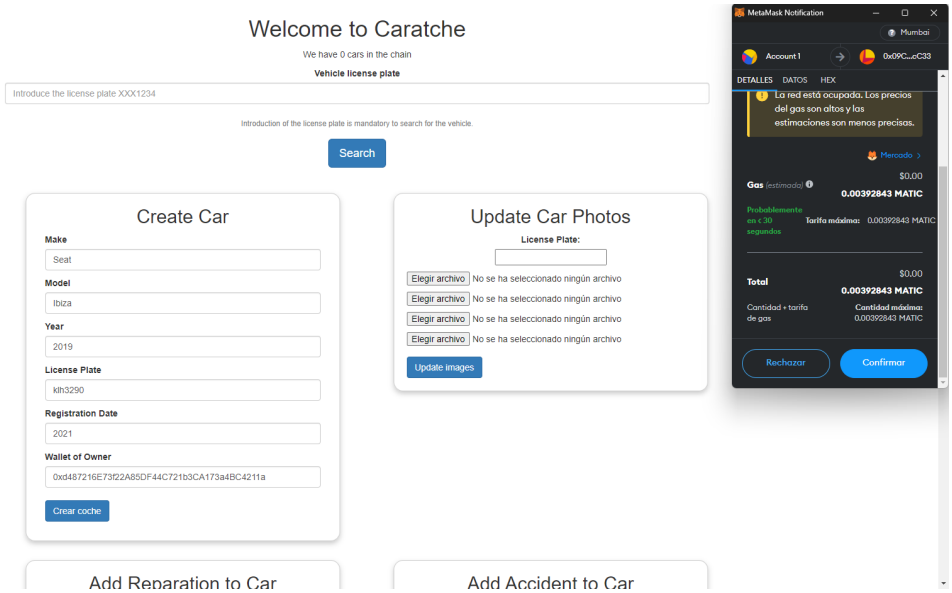

Figura 5.4: Creación de un vehículo

Una vez creado el vehículo podemos buscarlo usando la matrícula dentro de la blockchain, esta consulta de información es completamente gratuita para los usuarios, al contrario de la adición de información, la cual al requerir de computación dentro de la blockchain se ha de

pagar con la moneda que use la red, como ya hemos comentado en anteriores apartados, en este caso Matic.

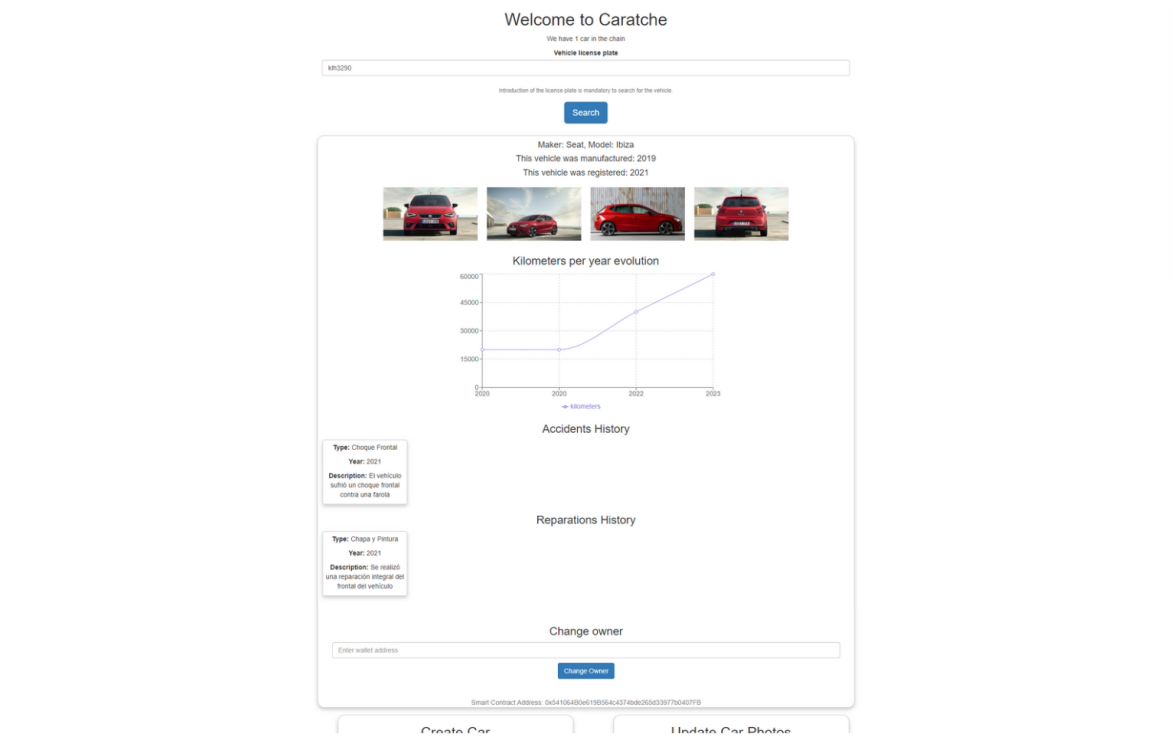

Figura 5.5: Consulta de información de un vehículo

Como funcionalidad adicional, se puede ver en la parte más baja de la figura 5.6, un formulario de cambio de dueño del vehículo, el cual solo será visible si la persona que realiza la consulta de la información del vehículo es el dueño del mismo.

Pues al igual que describe en el siguiente trabajo [Mendoza-Tello et al. (2021)], la confianza y transparencia son pilares fundamentales de la tecnología blockchain, por lo que haciendo referencia al apartado 4.4.2 se ha verificado el código del smart contract para que todos los usuarios que conozcan la dirección del contrato puedan ver las transacciones y el código que trata los datos de sus vehículos, así como los role definidos y las estructuras de información.

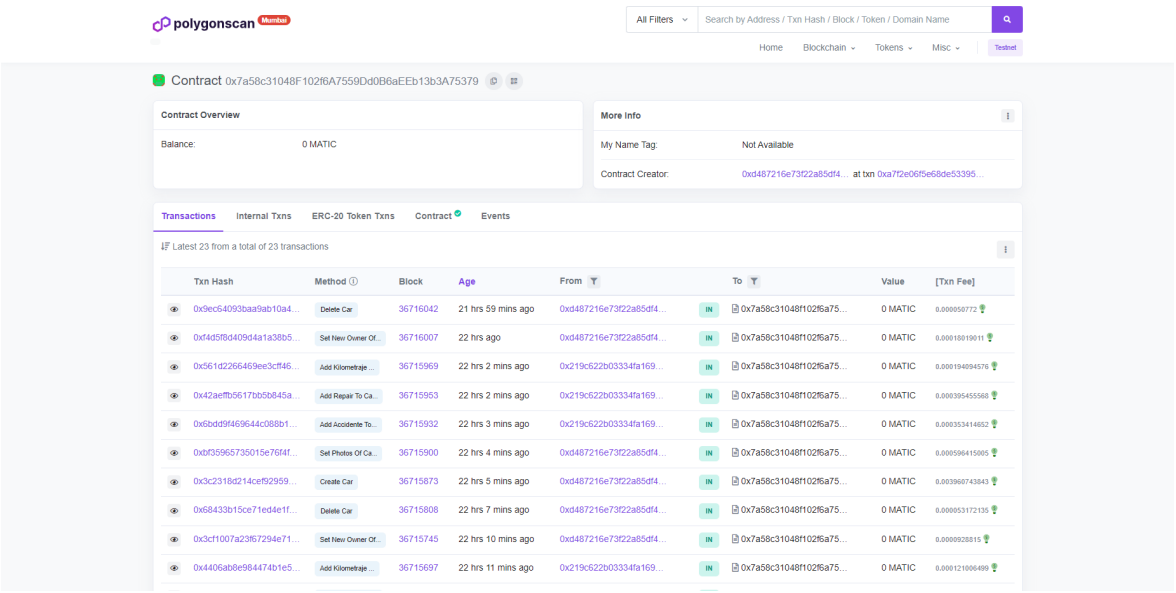

Figura 5.6: Consulta de contrato en PolygonScan

Se dispone a continuación un video representativo de la implementación realizada, donde se realiza el ciclo de vida de un activo. Video

1. Creación
2. Edición de sus propiedades
3. Transacción entre diferentes entes
4. Eliminación

6 Conclusiones

6.1 Conclusión final

En este trabajo, hemos demostrado que la tecnología blockchain tiene el potencial para permitir la trazabilidad de cualquier activo físico, desde un automóvil hasta un medicamento. La capacidad de blockchain para registrar, verificar y compartir información de forma segura y transparente la hace perfecta para este propósito.

A la hora de implementar un contrato inteligente para la trazabilidad de un activo, es crucial definir roles para las billeteras involucradas. Esto ayuda a asegurar que sólo las partes autorizadas tengan acceso a ciertas funciones y datos. Este enfoque garantiza la integridad de los datos y la privacidad de los usuarios, al tiempo que mantiene la transparencia y la verificabilidad de la blockchain.

Es esencial considerar cuidadosamente qué atributos de los activos deseamos rastrear. Para un sistema completamente descentralizado, se pueden requerir tecnologías adicionales como IPFS para almacenar datos que no se adaptan bien a la estructura de la blockchain, como imágenes o archivos grandes. Asimismo, se debe tener en cuenta que los datos una vez registrados en la blockchain no pueden modificarse, reforzando la necesidad de un diseño cuidadoso.

La adopción de patrones de diseño probados, como el modelo de "contract factory", puede ayudar a garantizar que el sistema es robusto, escalable y fácil de mantener. Este patrón, que permite la creación de múltiples contratos a partir de una "fábrica", es particularmente útil para rastrear grandes números de activos individuales.

Por último, es vital que la interfaz de usuario de la aplicación sea sencilla y clara, para facilitar su uso por parte de usuarios que no estén familiarizados con la tecnología blockchain. A medida que más y más usuarios interactúan con la blockchain, la accesibilidad y la usabilidad se vuelven cada vez más importantes.

En conclusión, la trazabilidad de activos físicos utilizando la tecnología blockchain es un campo prometedor, lleno de potencial, pero también presenta desafíos que requieren una consideración cuidadosa y un diseño inteligente. Este trabajo representa un paso en la exploración de estas posibilidades y desafíos.

6.2 Viabilidad de la dApp

Este proyecto demuestra que la aplicación propuesta para la trazabilidad de los automóviles utilizando la tecnología blockchain es absolutamente viable para su implementación en el día a día de nuestra sociedad. Actualmente, una gran parte de la población ya posee wallets digitales, lo que facilita enormemente la adopción de esta aplicación. Además, la familiaridad de los usuarios con los smartphones y las aplicaciones móviles también facilita la adopción de esta tecnología.

Una estrategia de implementación escalonada permitiría introducir gradualmente la aplicación en la industria automotriz. Inicialmente, los concesionarios y las fábricas podrían ser los primeros en adoptar el sistema, lo que proporcionaría trazabilidad desde el punto de producción. Posteriormente, se podría expandir su uso a los talleres oficiales y luego a los talleres de la calle. Este enfoque permitiría a la industria adaptarse progresivamente a la nueva tecnología, lo que facilitaría su adopción generalizada.

Para maximizar la eficiencia y la exactitud de los datos registrados, el sistema idealmente requeriría la mínima intervención humana posible. Esto podría lograrse mediante la implementación de tecnologías que permitan la recopilación automática de datos, como el uso de un dispositivo que se conecte directamente al coche para obtener los datos del kilometraje.

La transparencia y la accesibilidad del código abierto son otros aspectos fundamentales de la viabilidad de la aplicación. Gracias a la naturaleza transparente del ABI (Application Binary Interface), otros desarrolladores podrían crear sus propias versiones de la aplicación, incorporando nuevas funcionalidades o mejorando las existentes. Sin embargo, la seguridad y la integridad de la información seguirían garantizadas gracias a los roles definidos en el contrato inteligente.

Bibliografía

(2022). Ventajas y desventajas del Blockchain | BBVA Suiza. [Online; accessed 13. Feb. 2023].

(2023). LearnWeb3 DAO | Become a Web3 Developer for Free. [Online; accessed 13. Feb. 2023].

(2023). QuickNode | The Blockchain Development Platform. [Online; accessed 13. Apr. 2023].

(2023). ¿Qué es la tecnología Blockchain? - IBM Blockchain | IBM. [Online; accessed 12. Feb. 2023].

Almudí Coello, G. et al. (2022). El sector del automóvil en españa, importancia, situación actual y futura.

Amir Latif, R. M., Hussain, K., Jhanjhi, N., Nayyar, A., and Rizwan, O. (2020). A remix ide: smart contract-based framework for the healthcare sector by using blockchain technology. *Multimedia tools and applications*, pages 1–24.

Belk, R., Humayun, M., and Brouard, M. (2022). Money, possessions, and ownership in the metaverse: Nfts, cryptocurrencies, web3 and wild markets. *Journal of Business Research*, 153:198–205.

Calvão, F. (2019). Crypto-miners: Digital labor and the power of blockchain technology. *Economic Anthropology*, 6(1):123–134.

Chatterjee, A., Pitroda, Y., and Parmar, M. (2020). Dynamic role-based access control for decentralized applications. In *Blockchain–ICBC 2020: Third International Conference, Held as Part of the Services Conference Federation, SCF 2020, Honolulu, HI, USA, September 18-20, 2020, Proceedings 3*, pages 185–197. Springer.

Chen, Y., Li, H., Li, K., and Zhang, J. (2017). An improved p2p file system scheme based on ipfs and blockchain. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 2652–2657. IEEE.

Colaboradores de los proyectos Wikimedia (2023a). Descentralización - Wikipedia, la enciclopedia libre. [Online; accessed 12. Feb. 2023].

Colaboradores de los proyectos Wikimedia (2023b). Peer-to-peer - Wikipedia, la enciclopedia libre. [Online; accessed 13. Apr. 2023].

Costas, J. (2019). Tarjeta ITV electrónica, lo que necesitas saber. *https://www.motor.es/noticias/tarjeta-itv-electronica-201959941.html*.

Cuadros Choez, V. J. (2020). Análisis comparativo de un sistema de auditoría tradicional y un sistema de auditoría blockchain e ipes. B.S. thesis.

Daniel, E. and Tschorsch, F. (2022). Ipfs and friends: A qualitative comparison of next generation peer-to-peer data networks. *IEEE Communications Surveys & Tutorials*, 24(1):31–52.

Dannen, C. (2017). *Introducing Ethereum and solidity*, volume 1. Springer.

Diaz, C. L. (1985). *Cuentakilómetros con acelerómetro*. PhD thesis.

El Faqir, Y., Arroyo, J., and Hassan, S. (2020). An overview of decentralized autonomous organizations on the blockchain. In *Proceedings of the 16th international symposium on open collaboration*, pages 1–8.

Gackenheimer, C. and Gackenheimer, C. (2015). What is react? *Introduction to React*, pages 1–20.

Garankina, R., Zakharochkina, E., Samoshchenkova, I., Lebedeva, N., and Lebedev, A. (2018). Blockchain technology and its use in the area of circulation of pharmaceuticals. *Journal of Pharmaceutical Sciences and Research*, 10(11):2715–2717.

Gatteschi, V., Lamberti, F., Demartini, C., Pranteda, C., and Santamaria, V. (2018). To blockchain or not to blockchain: That is the question. *It Professional*, 20(2):62–74.

Golosova, J. and Romanovs, A. (2018). The advantages and disadvantages of the blockchain technology. In *2018 IEEE 6th workshop on advances in information, electronic and electrical engineering (AIEEE)*, pages 1–6. IEEE.

Huang, L., Röck, D., Murray, A., and Hofmann, E. (2019). modum. io: Funding a blockchain-based start-up's supply chain solution (teaching case).

Iyer, K., Dannen, C., Iyer, K., and Dannen, C. (2018). The ethereum development environment. *Building Games with Ethereum Smart Contracts: Intermediate Projects for Solidity Developers*, pages 19–36.

Jain, S. M. (2022). Hardhat. In *A Brief Introduction to Web3: Decentralized Web Fundamentals for App Development*, pages 167–179. Springer.

Khandelwal, P., Johari, R., Gaur, V., and Vashisth, D. (2021). Blockchain technology based smart contract agreement on remix ide. In *2021 8th International Conference on Signal Processing and Integrated Networks (SPIN)*, pages 938–942. IEEE.

Kim, H. M. and Laskowski, M. (2018). Toward an ontology-driven blockchain design for supply-chain provenance. *Intelligent Systems in Accounting, Finance and Management*, 25(1):18–27.

La Torre, J. S. Y. (2019). *Mejora de la gestión del Certificado de Identificación Vehicular, para enfrentar el delito de estafa en la compraventa de vehículos de segundo uso*. PhD thesis, Pontificia Universidad Catolica del Peru (Peru).

Lee, W.-M. and Lee, W.-M. (2019). Using the metamask chrome extension. *Beginning Ethereum Smart Contracts Programming: With Examples in Python, Solidity, and JavaScript*, pages 93–126.

Mendoza-Tello, J. C., Mendoza-Tello, T., and Mora, H. (2021). Blockchain as a healthcare insurance fraud detection tool. In *Research and Innovation Forum 2020: Disruptive Technologies in Times of Change*, pages 545–552. Springer.

Modi, R. (2018). *Solidity Programming Essentials: A beginner's guide to build smart contracts for Ethereum and blockchain*. Packt Publishing Ltd.

Mohanty, D. and Mohanty, D. (2018). Deploying smart contracts. *Ethereum for Architects and Developers: With Case Studies and Code Samples in Solidity*, pages 105–138.

Powell, L. M., Schwartz, J., and Hendon, M. (2021). The mobility open blockchain initiative: Identity, members, technologies, and future trends. In *Revolutionary applications of blockchain-enabled privacy and access control*, pages 99–118. IGI Global.

RACE (2023). Todo lo que debes saber sobre la ficha técnica de tu vehículo. *https://www.race.es/tarjeta-o-ficha-inspeccion-tecnica-vehiculo*.

Retamal, C. D., Roig, J. B., and Tapia, J. L. M. (2017). La blockchain: fundamentos, aplicaciones y relación con otras tecnologías disruptivas. *Economía industrial*, 405:33–40.

Robinson, P. (2020). The merits of using ethereum mainnet as a coordination blockchain for ethereum private sidechains. *The Knowledge Engineering Review*, 35:e30.

Romero, F. (2022). Interesante estudio europeo señala los coches que más trucan sus kilómetros. *https://www.motor.es/noticias/estudio-coches-importacion-kilometros-202284225.html*.

Rueda, A. (2023). Informe Carfax: qué es, cómo obtenerlo y cuánto cuesta. *https://motor.elpais.com/actualidad/informe-carfax-que-es-como-obtenerlo-y-cuanto-cuesta*.

She, Z. (2022). Vechain: A renovation of supply chain management–a look into its organisation, current acitivity, and prospect.

Sheridan, D., Harris, J., Wear, F., Cowell Jr, J., Wong, E., and Yazdinejad, A. (2022). Web3 challenges and opportunities for the market. *arXiv preprint arXiv:2209.02446*.

Starčević, I. and Puž, M. Cryptocurrencies and their implementation in everyday life4. In *Book of Proceedings*, page 5.

Testaj, V. (2022). Blockchain e digitalizzazione: una soluzione all'avanguardia per le criticità della supply chain.

Thakkar, M. and Thakkar, M. (2020). Next. js. *Building React Apps with Server-Side Rendering: Use React, Redux, and Next to Build Full Server-Side Rendering Applications*, pages 93–137.

Voshmgir, S. (2020). *Token economy: How the Web3 reinvents the internet*, volume 2. Token Kitchen.

Vujičić, D., Jagodić, D., and Ranđić, S. (2018). Blockchain technology, bitcoin, and ethereum: A brief overview. In *2018 17th international symposium infoteh-jahorina (infoteh)*, pages 1–6. IEEE.

Wang, Q., Li, R., Wang, Q., Chen, S., Ryan, M., and Hardjono, T. (2022). Exploring web3 from the view of blockchain. *arXiv preprint arXiv:2206.08821*.

Wohrer, M. and Zdun, U. (2018). Smart contracts: security patterns in the ethereum ecosystem and solidity. In *2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE)*, pages 2–8. IEEE.

Wöhrer, M. and Zdun, U. (2021). Devops for ethereum blockchain smart contracts. In *2021 IEEE International Conference on Blockchain (Blockchain)*, pages 244–251. IEEE.

Wüst, K. and Gervais, A. (2018). Do you need a blockchain? In *2018 crypto valley conference on blockchain technology (CVCBT)*, pages 45–54. IEEE.

Yoo, S. G. and Ahn, B. (2021). A study for efficiency improvement of used car trading based on a public blockchain. *The Journal of Supercomputing*, 77:10621–10635.

Lista de Acrónimos y Abreviaturas

DAC	organización que está dirigida a través de reglas codificadas en programas de ordenador llamados contratos inteligentes.
DAO	organización que está dirigida a través de reglas codificadas en programas de ordenador llamados contratos inteligentes.
DAPP	Aplicación creada en una red descentralizada que combina un contrato inteligente (smart contract) y una interfaz de usuario (frontend).
DGT	Dirección General de Tráfico.
IDE	(Integrated Development Enviorenmente o Entorno de desarrollo integrado).
IPFS	Sistema de archivos interplanetario.
ITV	Inspección Técnica de Vehículos..
NFT	(Non-Fungible Tokens, o Tokens No Fungibles en español).
RBAC	Sistema de acceso basado en roles.