UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

TFG del Grado en Ingeniería Informática

ContractMe

Presentado por David Martínez Bahillo
en Universidad de Burgos — 9 de abril de 2024
Tutor: Sandra Rodriguez Arribas

UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

D. nombre tutor, profesor del departamento de nombre departamento, área de nombre área.

Expone:

Que el alumno D. David Martínez Bahillo, con DNI 71566321G, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado «ContractMe».

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 9 de abril de 2024

Vº. Bº. del Tutor:	Vº. Bº. del co-tutor:
D. nombre tutor	D. nombre co-tutor

Resumen

Este proyecto se enfoca en la creación de una solución innovadora para la contratación laboral, aprovechando la tecnología blockchain y los contratos inteligentes. Se busca agilizar el proceso de contratación en sectores como la agricultura y servicios domésticos mediante la automatización de la gestión de contratos desde su inició hasta su final. La solución propuesta incluye funcionalidades avanzadas como el seguimiento del tiempo de trabajo y el pago automático, integrando tecnologías de localización GPS para establecer zonas de trabajo específicas. El proyecto incorpora métodos de autenticación biométrica y escaneo de códigos QR, apoyándose en soluciones de identidad descentralizadas para proteger la privacidad de los usuarios.

El objetivo final es proporcionar una herramienta que promueva prácticas de empleo transparentes y legales, optimizando los procesos de contratación y verificación para empleadores y trabajadores garantizando la seguridad de todos los tramites.

Descriptores

Blockchain, contratos inteligentes, localización GPS, autenticación biométrica de teléfonos inteligentes, encriptación de datos, desarrollo de aplicación móvil, procesamiento de pagos.

Abstract

This project focuses on creating an innovative solution for employment hiring, leveraging blockchain technology and smart contracts. It aims to streamline the hiring process in sectors such as agriculture and domestic services by automating contract management from start to finish. The proposed solution includes advanced features such as work time tracking and automatic payment, integrating GPS technology to establish specific work zones. The project incorporates biometric authentication methods and QR code scanning, relying on decentralized identity solutions to protect user privacy.

The final goal is to provide a tool that promotes transparent and legal employment practices, optimizing hiring and verification processes for employers and workers, ensuring the security of all procedures.

Keywords

Blockchain, smart contracts, GPS tracking, smartphone biometric authentication, data encryption, mobile app development, payment processing.

Índice general

Índice de figuras

Índice de tablas

1. Introducción

La transformación digital se ha convertido en un pilar fundamental para el desarrollo y la eficiencia de diversos sectores económicos, entre ellos el mercado laboral. Sin embargo, a pesar de los avances tecnológicos, ciertos sectores como la agricultura, los servicios domésticos y el trabajo freelance enfrentan retos significativos en la gestión eficiente de contrataciones y la verificación de identidad y tiempo laboral. Estos desafíos se ven agravados por la presencia de la economía sumergida, donde la falta de transparencia y la informalidad laboral no solo perjudican la economía global, sino que también vulneran los derechos de los trabajadores. En este contexto, se pretende implementar una solución innovadora que permita superar las barreras existentes, haciendo frente a la economía sumergida mediante el empleo de tecnologías disruptivas como blockchain y contratos inteligentes. Esta iniciativa busca promover la formalización de empleos y asegurar prácticas laborales justas, abordando directamente los problemas de falta de transparencia y seguridad en los procesos contractuales. Se destacará el impacto social y económico de reducir la economía sumergida, apoyándose en datos que evidencian su magnitud y las implicaciones positivas de una mayor formalización laboral.

La adopción de contratos inteligentes en una red blockchain permite una gestión contractual transparente, segura y automática, asegurando que los términos acordados se cumplan eficazmente sin intervención manual. Esta tecnología también proporciona un registro inmutable y verificable de las transacciones y acuerdos labores, combatiendo así prácticas ilegales y fomentando un entorno de trabajo más equitativo.

Este proyecto no solo aborda la necesidad de un sistema de contratación y verificación más eficiente y seguro sino que también se anticipa a las demandas de un mercado laboral en constante evolución, ofreciendo una

herramienta adaptable y escalable. Con una interfaz de usuario diseñada para la simplicidad y la accesibilidad en dispositivos móviles, se busca fomentar prácticas de empleo legales y transparentes, contribuyendo a la creación de un entorno laboral más justo y seguro para todos. La inclusión de tecnologías GPS y la autenticación biométrica refuerza este objetivo, permitiendo un seguimiento preciso de las horas de trabajo y asegurando la identidad de los trabajadores de manera segura y privada.

Por tanto este proyecto no solo propone una solución práctica que beneficia tanto a trabajadores como empleadores, sino que también tiene el potencial de impactar positivamente en la economía global. marcando un paso adelante hacia la erradicación de la economía sumergida y el establecimiento de prácticas laborales más justas y transparentes a nivel mundial.

2. Objetivos del proyecto

En esta sección se muestran las metas que se pretenden alcanzar con el desarrollo del proyecto.

Objetivos técnicos

1. **Seguir estándares en el desarrollo de contratos inteligentes**: Alineándose con las mejores prácticas de la comunidad Ethereum y Solidity.
2. **Adoptar metodologías ágiles**: Para favorecer un desarrollo iterativo y eficiente, facilitando la adaptación a cambios y mejora continua del proyecto.
3. **Documentar del proyecto**: Utilizando herramientas visuales para explicar la arquitectura.
4. **Mantener un alto estándar de calidad en el código**: Empleando herramientas de revisión de código.
5. **Evaluar la calidad de la solución**: En términos de eficiencia.
6. **Planear un plan de implementeción y escalabilidad**

Objetivos del Software

1. **Integrar Ethereum en el proyecto**: Asegurando una implementación efectiva para el despliegue de contratos inteligentes

2. **Desarrollar contratos inteligentes con Solidity**: Asegurando un código eficiente y adaptado a las necesidades específicas de la contratación laboral.

3. **Desplegar en una red Ethereum real**: Logrando una convivencia con el resto de contratos de la comunidad.

4. **Implementar tecnologías de autenticación biométrica**: Utilizando métodos como la huella dactilar o el reconocimiento facial para la identificación del usuario.

5. **Integrar tecnología GPS**: Verificando la presencia del trabajador en el lugar adecuado y el momento correcto.

6. **implementar códigos QR**: Agilizando los procesos administrativos y operativos además de aplicar una capa adicional de seguridad.

7. **Integrar servicios de inicio sesión**: Proporcionando a los usuarios opciones de inicio de sesión descentralizadas a través de MetaMask.

8. **Desarrollar una interfaz de usuario intuitiva y accesible**: Priorizando la simplicidad y la usabilidad para garantizar una plataforma fácil de utilizar para todos los usuarios.

3. Conceptos teóricos

A continuación se sintetizarán algunos conceptos teóricos relevantes para la correcta compresión del proyecto.

3.1. Introducción a la Blockchain

La Blockchain proporcionando un registro inmutable y en tiempo real de transacciones, emergió como una solución al problema de doble gasto [15] en transacciones digitales, un desafío que había eludido a los criptógrafos durante décadas, el cual consiste en la posibilidad de gastar una misma moneda digital más de una vez. Esto ocurre debido a la falsificación o duplicación de archivos digitales que representan la moneda. En 2008 una persona o grupo anónimo, bajo el seudónimo de Satoshi Nakamoto, publicó un documento técnico para crear una moneda digital para contabilizar y transferir valor. Así nació una tecnología que se fundamenta en una base de datos descentralizada compuesta por bloques de información replicados y sincronizados en múltiples ordenadores. Esta premisa hizo que en enero de 2009 entrara en funcionamiento la primera red basada en el protocolo Bitcoin [14]. La BlockChain demostró su capacidad para contabilizar y transferir valor de manera segura sin la necesidad de intermediarios, transformando el sector financiero y encontrando aplicaciones en una variedad de industrias.

3.2. Tecnología subyacente

Nodo

Los nodos en Blockchain son dispositivos, generalmente computadoras, que participan en una red Blockchain. Estos dispositivos ejecutan el software del protocolo Blockchain, lo que les permite ayudar a validar transacciones y mantener la seguridad de la red [18].

Para lograr esto, los nodos se comunican entre sí por medio de protocolos peer-to-peer(P2P). En un sistema P2P, cada nodo se conecta directamente a otros nodos sin necesidad de intermediarios. Cada nodo actúa tanto como cliente como servidor, compartiendo la carga de procesamiento de datos y la transmisión de información. Esta arquitectura permite desarrollar un red robusta y descentralizada donde cada nodo contribuye al mantenimiento y seguridad de la red. Siendo extremadamente difícil censurar o bloquear el acceso a la red, ya que no hay un punto central de control.

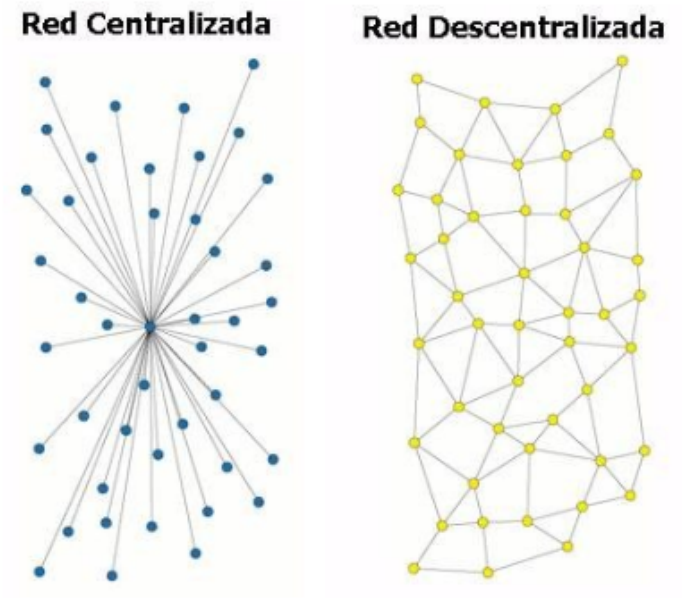

Figura 3.1: Diferencias entre una red centralidad y descentralizada

Resulta fundamental comprender los distintos tipos de nodos que conviven en una red y el papel específico que cada uno desempeña. A continuación, se enumerarán los tipos de nodos, destacando sus características y funciones únicas [26].

- **Nodos completos:** Estos nodos albergan una copia completa del libro mayor de la Blockchain. Al contener el registro completo de todas las transacciones, los nodos pueden verificar de forma independiente cualquier transacción sin necesidad de recurrir a información externa.
- **Nodos ligeros:** Han sido diseñados para dispositivos con recursos limitados, los nodos ligeros no almacenan el registro completo, sino que confían en los nodos completos para obtener dicha información.
- **Nodos mineros:** Son utilizados en las redes que utilizan el algoritmo de consenso Proof of Work(PoW), los nodos mineros compiten para agregar nuevos bloques a la Blockchain para obtener un recompensa por ello.
- **Nodos completos podados:** Estos nodos almacenan una versión recortada del registro, eliminando datos antiguos para ahorrar espacio, pero a diferencia de los nodos ligeros estos pueden seguir verificando de forma independiente cualquier transacción.
- **Nodos completos de archivo:** Almacenan todo el libro mayor del Blockchain, desde el principio de los tiempos. Los Nodos Completos de Archivo son la única fuente valiosa y fiable para verificar los datos de transacciones anteriores en la historia de una Blockchain, ya que no están afectados por el límite de tiempo o almacenamiento. A diferencia de los nodos completos, los nodos de archivo vas más allá al almacenar cada cambio de estado en la Blockchain.
- **Nodos de autoridad:** Utilizados en Blockchains con mecanismos de consenso como Proof of Authority(PoA), estos nodos son operados por entidades verificadas y de confianza dentro de la red. A diferencia de PoW, tienen el poder de validar bloques sin necesidad de competir entre ellos.
- **Nodos maestros:** Ofrecen funcionalidades adicionales como la ejecución de contratos inteligentes. Requieren de una garantía o "stake"para operar y suelen recibir incentivos por ofrecer servicios especializados.
- **Nodos de estaca:** Utilizados en Blockchains que funcionan con el algoritmo de consenso Proof of Stake(PoS) donde los nodos participan en la validación de bloques apostando una cierta cantidad de criptomonedas como garantía para operar.
- **Nodos Lightning:** Específicos de las soluciones de segunda capa como Lightning Network, estos nodos facilitan transacciones rápidas

y de bajo costo fuera de la cadena principal, ayudando a reducir la congestión.

- **Supernodos:** Son nodos con capacidades y responsabilidades adicionales, a menudo relacionadas con la gobernanza de la red, se crean bajo demanda para realizar tareas especializadas, como implementar cambios de protocolo o gestionar protocolos.

Para este proyecto se ha utilizado Ethereum la cual utiliza una gran variedad de nodos para satisfacer las necesidades específicas de su ecosistema, incluyendo nodos completos, nodos ligeros, y nodos de archivo. A partir de la evolución de Ethereum 2.0 y su progresiva migración se han introducido los nodos estaca, que poco ha poco van remplazando a los nodos mineros utilizados en Ethereum 1.0.

Libro mayor digital

La tecnología Blockchain funciona como un libro mayor digital (DTL) que registra transacciones en múltiples nodos de manera que cada registro es inalterable e irreversible. Este registro se organiza en bloques de datos que están interconectados de manera cronológica formando una cadena [27]. Una vez un bloque es añadido a la cadena, se distribuye a todos los nodos de la red, actualizando así el libro mayor en cada nodo. Esta distribución asegura una redundancia que la hace segura contra la manipulación, ya que alterar un registro requeriría cambiar el bloque correspondiente y todos los bloques posteriores en la mayoría de los nodos de la red, una tarea casi imposible debido a la demanda de cómputo [5].

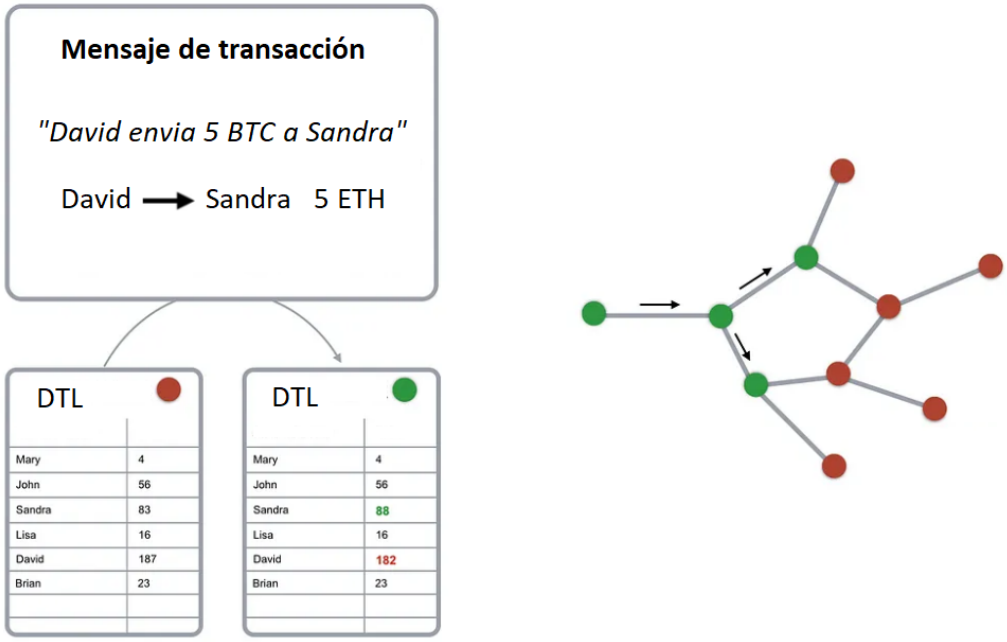

Figura 3.2: Proceso de actualización del libro mayor

Encadenamiento de bloques

Cada bloque contiene un numero determinado de transacciones y está formado por tres elementos principales; los datos de las transacciones, el hash del bloque anterior en la cadena y su propio hash único, generado a partir de la información contenida en el bloque. Los bloques se enlazan mediante el hash formado por los datos del bloque anterior. El algoritmo hash más usado en la Blockchain es "SHA-256-[20] desarrollado por la Agencia de Seguridad Nacional (NAS) en el año 1997. Es conocido por ser lento en comparación con otras funciones hash, pero a pesar de esto destaca por su seguridad por lo que lo hace adecuado para aplicaciones financieras. El hash consiste en una función criptográfica que produce una salida de longitud fija a partir de una entrada de longitud variable. Cualquier cambio en el contenido de un bloque anterior requeriría recalcular todos los hashes subsiguientes, lo cual es computacionalmente costoso y prácticamente imposible de realizar sin ser detectado

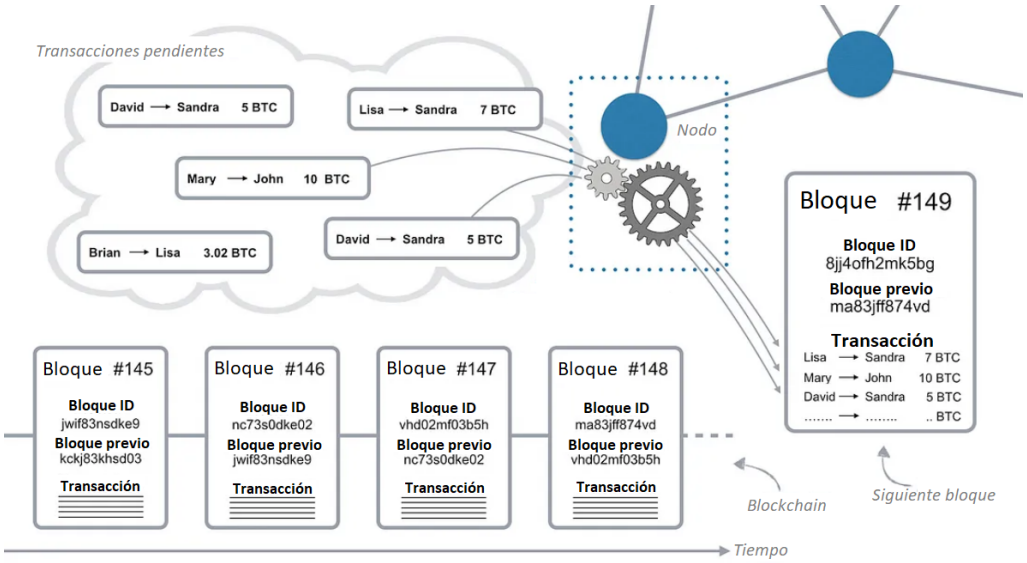

Figura 3.3: Encadenamiento de bloques en la Blockchain

Wallet y transacciones

Para iniciar una transacción en la Blockchain se debe de emplear una "wallet.º monedero electrónico, este es un software que permite almacenar y intercambiar activos digitales. Este monedero genera una y almacena un par de claves criptográficas, una clave pública que actúa como una dirección a la cual otros pueden enviar activos, y una clave privada, que se utiliza para firmar digitalmente las transacciones, asegurando que solo el propietario de la clave privada pueda autorizar la transferencia de sus activos.

Por tanto, cuando se desea realizar una transacción, el monedero electrónico firma la transacción utilizando su clave privada. Esta firma digital, generada a través de algoritmos de criptografía asimétrica como el RSA, es esencialmente un hash criptográfico de la transacción encriptado con la clave privada del monedero. Seguidamente, los nodos de la red al recibir la transacción, emplean la clave pública del firmante para descifrar la firma digital. Este proceso no solo autentifica que la transacción fue creada por el poseedor de la clave privada correspondiente, sino que también asegura que la transacción no haya sido modificada, ya que cualquier cambio en los datos de la transacción resultaría en una discrepancia al verificar la firma con la clave pública.

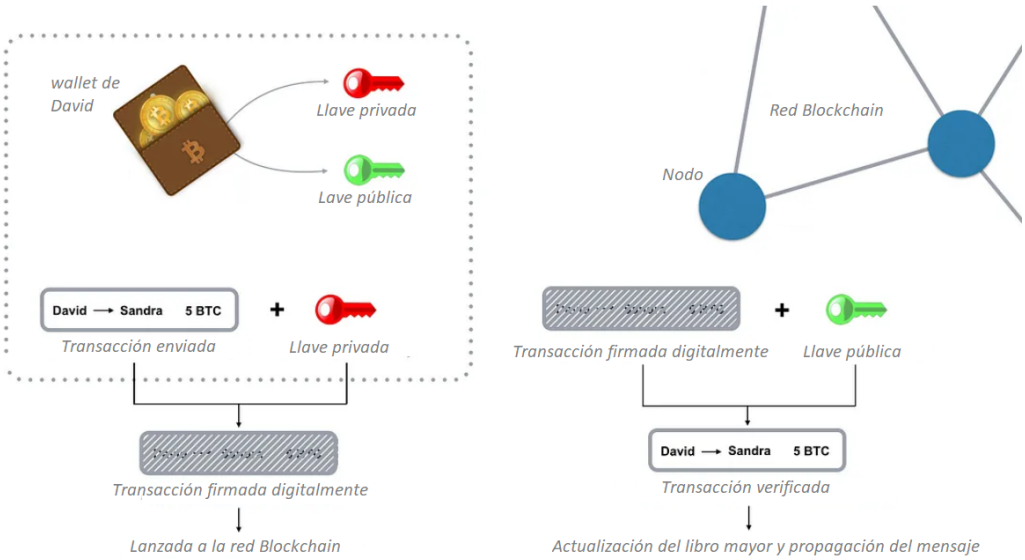

Figura 3.4: Transacción encriptada con firma digital

A diferencia de los sistemas financieros tradicionales, la Blockchain no registra los saldos de las cuentas de manera directa. En su lugar, mantiene un registro detallado de todas las transacciones que han sido verificadas y aprobadas en la red. Por tanto, para determinar el saldo de una wallet, es necesario analizar y verificar todas las transacciones asociadas a ella desde la creación de la red. Este enfoque asegura que la información sea transparente y auditada constantemente por todos los nodos de la red, lo que refuerza la seguridad y la integridad del sistema.

Por tanto si un usuario quiere generar una transacción para enviar un activo, este debe generar una solicitud que incluya unas referencias llamadas ïnputs", a transacciones entrantes previas que sumen la cantidad deseada. Los nodos de la red verifican estos inputs para asegurarse que no hayan sido gastados previamente. Por ende, una vez los inputs hayan sido referenciados, estos son invalidados para transacciones futuras, evitando el doble gasto de activos digitales.

Consenso

El mecanismo de consenso en una blockchain es fundamental para mantener la integridad y la seguridad de la red. Este proceso permite que todos los nodos de la red se pongan de acuerdo sobre el estado actual del libro

mayor digital, lo que significa que cualquier cambio en el blockchain debe ser validado y aceptado por más del 50 % de los nodos de la red.

Actualmente existen dos categorías amplias de protocolos de consenso, los de finalidad probabilística, representados fundamentalmente por **Proof of work** (PoW), implica que la confirmación de una transacción se vuelve más segura a medida que se van confirmando bloques sucesivos. Por otro lado, los protocolos con finalidad absoluta, como **Proof of Stake** (PoS) aseguran la finalización de las transacciones de manera definitiva una vez se agregan a la Blockchain.

El protocolo Proof of Work se destaca como el mecanismo de consenso más empleado en las Blockchains. En PoW, los participantes de la red, conocidos como mineros, compiten para resolver un problema criptográfico complejo, el cual requiere un considerable poder computacional. Este problema implica encontrar un valor llamado "nonce"que cuando se combina con los datos del bloque y se procesa a través de una función hash produce un resultado que cumple con una criterio especifico, generalmente que contenga un cierto número de ceros al principio del hash. Este proceso, conocido como minería, garantiza que alterar un bloque ya minado sea computacionalmente inviable, proporcionando así seguridad e inmutabilidad a la cadena de bloques. A su vez, la dificultad de este problema se ajusta periódicamente para mantener un tiempo objetivo entre la creación de bloques consecutivos, asegurando la estabilidad y previsibilidad de la generación de nuevos bloques. La labor de los mineros en la Blockchain no es altruista, este poder de cómputo es recompensado de tal manera que el primer minero en resolver el problema es recompensado con una cantidad fijada de criptomonedas [24].

A pesar de su amplia adopción y probada seguridad, La minería de criptomonedas requiere de grandes granjas de minado que conllevan un elevado consumo energético teniendo un gran impacto ambiental. Este aspecto ha impulsado la búsqueda de alternativas más eficientes y sostenibles como el Proof of Stake (PoS) que reduce el consumo eléctrico. Este protocolo se basa en seleccionar a los validadores en proporción a la cantidad de criptomonedas que poseen y están dispuestos a b̈loquearc̈omo garantía. La adopción de PoS por parte de proyectos líderes como Ethereum, con su transición a Ethereum 2.0, marca un hito importante y podría incentivar a otras criptomonedas a seguir un camino similar.

Hay que destacar que estos mecanismos de consenso únicamente son seguros en redes de una cierta magnitud ya que en contra pueden ser victimas del ataque del 51 %. El ataque del 51 % es una vulnerabilidad en la que un único minero o grupo de mineros controla más del 50 % del poder de

cómputo de la red. De tal forma que les permite influir en las confirmaciones de las transacciones, permitiendo de esta forma el fenómeno del doble gasto. Es por esto que los mecanismos de consenso únicamente son efectivos en redes grandes, donde el tamaño de la Blockchain es equivalente a la inversión necesaria para logar un control del 51 %.

3.3. Tipos de Blockchain

Existen diferentes tipos de redes, cada una diseñada para satisfacer unas necesidades en cuanto a la privacidad, gobernanza y accesibilidad. Se pueden clasificar en cuatro categorías principales [21]:

- **Blockchains públicas:** Son completamente abiertas y cualquier persona puede unirse. Bitcoin y Ethereum son buenos ejemplos de Blockchains públicas, donde las transacciones y los datos son visibles para todos, manteniendo al mismo tiempo el anonimato de los usuarios. Han pavimentando el camino para un ecosistema de aplicaciones descentralizadas (dApps) y finanzas descentralizadas (DeFi).
- **Blockchains semiprivadas:** Son operadas por una única entidad con la posibilidad de restringir el acceso, ofreciendo un equilibrio entre el control y la descentralización. A diferencia de las redes privadas, las semiprivadas pueden permitir la participación de partes externas bajo ciertas condiciones, manteniendo un nivel significativo de control sobre la red. Este tipo de redes las ofrecen empresas como IBM (Hyperledger Fabric) utilizada en sectores como la salud y la financiación, permite a las organizaciones configurar redes donde los datos se comparten solo con los actores autorizados, mejorando la eficiencia y la seguridad. Ofreciendo opciones personalizables para empresas, equilibrando la privacidad con la innovación [13].
- **Blockchains privadas:** Son operadas por una única organización, permiten un control total sobre quién puede participar en la red. Estas redes limitan el principio de la descentralización, pero ofrecen una solución eficaz para entornos empresariales que necesitan privacidad y eficiencia en procesos internos. Por ejemplo, para este proyecto se ha utilizado la herramienta Ganache, la cual simula una red privada, siendo de gran utilidad en la fase de desarrollo y pruebas.
- **Consorcio:** Representan un equilibrio entre los modelos públicos y privados, siendo operadas por un grupo de organizaciones en lugar

Característica	Pública	Privada	Consorcio	Semiprivada
Acceso	Abierto a todos	Restringido	Restringido a organizaciones	Control selectivo
Descentralización	Completa	Mínima	Parcial	Variable
Mecanismo de Consenso	PoW, PoS	Permisionados	Permisionados, personalizados	Combinación
Transparencia	Total	Limitada	Limitada a miembros	Configurable
Privacidad	Baja	Alta	Moderada	Alta en privado
Velocidad y Escalabilidad	Variable	Alta	Moderada	Configurable
Casos de Uso	DApps	Registros internos	Cadena de suministro, DeFi	Compartimentos privados

Tabla 3.1: Resumen de Tipos de Blockchain y sus Características.

de una única entidad. Esta posibilidad permite compartir la responsabilidad del mantenimiento de la red entre varios participantes, lo que las hace adecuadas para colaboraciones interempresariales. Un ejemplo real sería R3 Corda, que facilita transacciones eficientes y seguras entre instituciones financieras, reduciendo costos y tiempos de procesamiento [3].

3.4. Web3 y DApps

Web3 emerge como una propuesta revolucionaria en la evolución de Internet, promoviendo una arquitectura descentralizada que contrasta con las fases anteriores de la web. Desde los comienzos de Internet con Web 1.0, caracterizada por páginas estáticas y un flujo unidireccional de información, hasta la aparición de Web 2.0, que permitía a los usuarios interactuar con el contenido en línea y entre ellos. Sin embargo, estas etapas se han caracterizado por su centralización del poder y los datos en mano de unas

pocas plataformas dominantes, lo que a menudo cuestiona preocupaciones sobre la privacidad, seguridad y monopolizción de la información [11].

En este contexto nace Web3 como una solución prometedora para abordar estas preocupaciones centrándose en descentralización. Web3 a diferencia de sus antecesoras no se limita a ser un medio para compartir y crear contenido sino que pretende redefinir las dinámicas de poder en el espacio digital mediante la implementación de tecnologías Blockchain.

La aplicación de la Blockchain más reconocida a nivel mundial ha sido Bitcoin. Esta criptomoneda ha demostrado las capacidades de la Blockchain, permitiendo transacciones globales rápidas y seguras, caracterizadas por su alta liquidez, bajas comisiones y un nivel de anonimato que protege la privacidad del usuario [14].

Más allá de las criptomonedas, esta tecnología se destaca por su naturaleza descentralizada, donde cada individuo en la red tiene acceso a una copia del registro completo de transacciones, lo cual garantiza una transparencia sin precedentes. Siendo así que la Blockchain cuenta con la capacidad para ofrecer la trazabilidad completa en las cadenas de suministro. Cada producto puede ser rastreado desde su origen hasta el consumidor final, asegurando la autenticidad y facilitando la detección de cualquier problema en el proceso [21].

La seguridad es otro de los pilares fundamentales de la Blockchain, ya que la inmutabilidad del registro asegura que una vez la información ha sido añadida a la cadena, esta no podrá ser alterada, reforzando así la confianza en el sistema.

Desde el punto de vista operativo, la Blockchain ofrece eficiencias significativas al eliminar los intermediarios, reduciendo tanto los tiempos de procesamiento como los costos asociados, a parte de minimizar las posibilidades de error. Este aspecto es crucial en sectores como el financiero, donde los contratos inteligentes facilitan la ejecución automatizada y segura de acuerdos sin la necesidad de intermediarios, redefiniendo las prácticas comerciales y financieras en la era digital.

Dentro de este entorno de Web3, las aplicaciones descentralizadas(DApps) se presentan como un componente esencial, ofreciendo una alternativa a las aplicaciones centralizadas tradicionales. Aunque el concepto de DApp parece moderno, sus raíces se remontan a más de 20 años. Las primeras aplicaciones en este ámbito fueron las aplicaciones de redes P2P, siendo algunas tan conocidas como eMule o BitTorrent, las cuales democratizaron el acceso a

la información al distribuirla a través de una red de nodos(ordenadores) en lugar de centralizarla en servidores únicos [12].

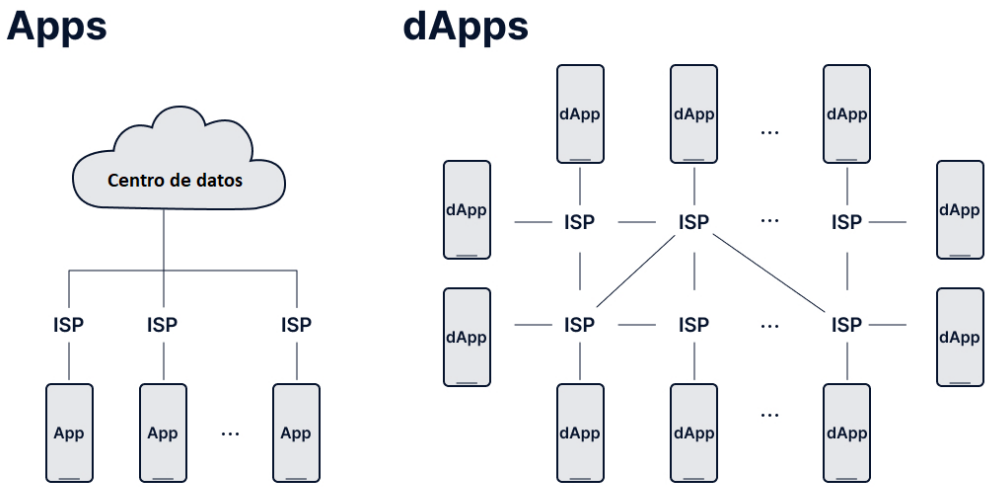

Figura 3.5: Estructura de las DApps

El mercado de las DApps ha experimentado un crecimiento impresionante en los últimos años, escalando de un valor de mercado de 10.5 mil millones de dolares en 2019 a más de 25 mil millones en 2022, este proyecto se proyecta a continuar a un ritmo acelerado, esperándose alcanzar una capitalización de 368 mil millones en 2027. Estas cifras subrayan la creciente importancia y potencial económico de las DApps en el ecosistema digital [1].

Hoy en día las DApps adquieren una nueva dimensión, ejecutándose en redes descentralizadas y apoyándose en la tecnología de contratos inteligentes para automatizar procesos y garantizar la ejecución de acuerdos sin intermediarios. Las Dapps destacan por su libertad y soberanía digital de los usuarios, ya que la ausencia de un punto central de control hace prácticamente imposible que se impongan restricciones arbitrarias por parte de entidades externas. A su vez, esta distribución de los datos a través de la red dificulta los ataques y manipulaciones. Adicionalmente, el carácter de código abierto de muchas DApps fomenta la continua revisión por parte de la comunidad e incrementa la seguridad y la confianza en estas aplicaciones.

Las DApps se pueden clasificar en tres niveles distintos: las de primer nivel se alojan en su propia Blockchain; las de segundo nivel se alojan en Blockchains ajenas; y finalmente las de tercer nivel, las cuales dependen de

Dapps de segundo nivel para funcionar. Este proyecto se enmarca en las DApps de segundo nivel, ya que se opta por desarrollar la DApp sobre la blockchain de Ethereum, dada su amplia aceptación y su papel predominante en el ecosistema de las aplicaciones descentralizadas.

3.5. Ethereum

Ethereum actualmente es la segunda red más grande, solo por detrás de Bitcoin y aunque normalmente se les compara, Ethereum y Bitcoin son dos proyectos totalmente distintos. Bitcoin introdujo una forma descentralizada de dinero electrónico, permitiendo realizar transferencias en una red segura y sin intermediarios. Ethereum por otro lado, lanzado en 2015 basándose en la base establecida por Bitcoin, expandió su funcionalidad.

La ambición de Ethereum no solo se limita a simplificar las transacciones financieras, su propósito es ampliar el alcance de las aplicaciones descentralizadas (Dapps) al proporcionar una infraestructura programable. A diferencia de Bitcoin, no cuenta con un suministro limitado de monedas para preservar su valor, además cuenta con tiempos de minado de bloques significativamente menores, por lo que puede ofrecer una experiencia más ágil y confirmaciones de transacciones más rápidas. Ethereum de esta manera se constituye como la columna vertebral de un Internet descentralizado(Web3) [4].

Ether

El Ether(ETH) es la criptomoneda que alimenta la red Ethereum, desempeña un papel fundamental tanto como activo digital como en la funcionaldiad de la red.

Similar a como ocurre con otras criptomonedas, ETH se puede enviar y recibir asegurando transacciones a cualquier parte del mundo y sin intermediarios. A su vez, cada operación que cambie el estado de la red, supone el pago de una pequeña tarifa en ETH. Desde una simple transferencia hasta la ejecución de complejos contratos inteligentes requieren computación y almacenamiento, siendo utilizado el ETH para pagar estas denominadas "Tarifas de gas". Finalmente como se introdujo el apartado mecanismos de consenso, Ethereum está implementando el mecanismo de consenso Proof of Stake(PoS) y el ETH constituye el activo que los participantes de la red pueden bloquear(stake) con el objetivo de dar seguridad a la red a cambio de recompensas en esta misma moneda.

Aunque ETH no tiene un suminsitro máximo fijo como Bitcoin, las políticas de emisión estan diseñadas para asegurar que el suminsitro de ETH crezca a un ritmo predecible y decreciente, lo que contribuye a la escasez y valor a largo plazo. El valor de ETH no se basa solo en su escasez digital como en el resto de criptomonedas, también adquiere un valor adicional al permitir a los usuario pagar las tarifas de gas y más recientemente ETH se ha vuelto válido para los usuario de aplicaciones financieras descentralizadas(DEFI) al poder usarse como garantía para prestamos criptográficos o como sistema de pago [8].

Ethereum Virtual Machine

La Ethereum Virtual Machine(EVM) constituye el núcleo de la red Ethereum, proporcionando un entorno de ejecución aislado y seguro para los contratos inteligentes y las aplicaciones descentralizadas. La EVM es una maquina virtual Turing completa, que facilita la ejecución de código en el contexto de la blockchain. La EVM funciona como una máquina de pila, con una profundidad de 1024 items, cada item es una palabra de 256 bits, seleccionado para su utilización con la criptografía de 256 bits. Se ejecuta a través de códigos de operación, que realizan operaciones estándar de pila como XOR, AND, ADD, SUB, etc [19].

La EVM funciona mediante la ejecución de bytecode, un conjunto de intrucciones de bajo nivel que la máquina es capaz de interpretar y ejecutar directamente. El bytecode se obtiene de la compilación de contratos inteligentes escritos en lenguajes de alto nivel. La EVM es capaz de ejecutar código en un entorno completamente aislado, por lo que los contratos pueden oprar sin comprometer la seguridad de la red. Manejando el acceso a los recursos de los computadores y limitando sus acciones en un ambiente controlado. Esta seguridad se ve reforzada por su naturaleza descentralizada y distribuida, por lo que cualquier intento de manipulación mediante un código malicioso deberá enfrentarse al consenso de la red. De esta forma, Ethereum funciona como un ordenador mundial descentralizado de una general en una red entre pares.

La EVM cuenta con gran flexibilidad en cuanto a su capacidad para soportartar una variedad de lenguajes de programación, facilitando así la adaptabilidad del sistema Ethereum. Además, la EVM permite que cualquier persona con acceso a Ethereum pueda desplegar sus propios contratos inteligentes, democratizando el acceso a la tecnología blockchain y creando una gran comunidad de desarrolladores [9].

Gas

El gas es una unidad que mide la cantidad de esfuerzo computacional requerida para ejecutar operaciones en la red Ethereum. La tarifa de gas es la cantidad de gas usado para hacer alguna operación, multiplicado por el coste unitario del gas. Es un mecanismo no solo de prevención contra ataques de spam, sino tambien que facilita una economía alrededor, estableciendo un sistema de compensación para los mineros que procesan y confirman las transacciones. El precio del gas suele expresarse en gwei, cada gwei equivale a $0,000000001$ ETH o 10^{-9} ETH.

Desde 2021 con la actualización L̈ondonsë introdujo el mecanismo EIP-1559, que hizo que el calculo de la tarifa de gas fuera más previsible, introduciendo los conceptos de tarifa base y tarifa prioritaria.

La tarifa base indica la cantidad mínima de gas a pagar para que la transacción se considere como válida para tramitar. La tarifa base se ajusta dinámicamente bloque a bloque basándose en la ocupación del bloque anterior. El sistema, calcula cuánto gas se utilizó en total para todas las transacciones en el bloque anterior y lo compara con el tamaño de bloque previamente definido por el protocolo. Este tamaño de bloque previamente definido es una medida de cuánto gas se espera que consuman las transacciones en un bloque ideal, cada bloque en la red Ethereum tiene un tamaño esperado de 15 millones de gas. Si el bloque anterior consumió más gas del tamaño previamente definido, la tarifa base aumentará para el siguiente bloque hasta un máximo de 12,5 %. El crecimiento exponencial en el costo de las transacciones cuando los bloques están constantemente sobreocupados actúa como un freno contra la congestión y alcanzando el tamaño un tamaño de equilibrio de 15 millones de gas de media por bloque , ya que solo las transacciones prioritarias de aquellos que estén dispuestos a pagar más serán ejecutadas.

Por otro lado, **la tarifa prioritaria** establece una propina que se añade a la tarifa base para que los validadores vean la transacción más atractiva para incluirla en el siguiente bloque, ya que es esta propina la que adquirirán en recompensa a su trabajo, a su vez la tarifa base es consumida y por lo tanto, eliminada. Por lo general, la tarifa base por si sola es insuficiente para que un validador se interese por ella, por lo que la elección de una transacción por parte del validador depende de la tarifa prioritaria, la cual tiene que ajustar su precio en base al uso de la red en el momento de enviar la transacción.

Bloque	**Total Gas**	**Incremento Tarifa (%)**	**Tarifa Base (gwei)**
1	15 M	0 %	100 gwei
2	30 M	0 %	100 gwei
3	30 M	12,5 %	112,5 gwei
4	30 M	12,5 %	126,6 gwei
5	30 M	12,5 %	142,4 gwei
6	30 M	12,5 %	160,2 gwei
7	30 M	12,5 %	180,2 gwei
8	30 M	12,5 %	202,7 gwei

Tabla 3.2: Evolución de la Tarifa Base en Ethereum.

Existe otro tipo de tarifa que se puede incluir de manera opcional en las transacciones, esta es la tarifa máxima, que es la que se ha usado para las transacciones dentro de este proyecto. Este parámetro opcional recoge la cantidad máxima que el usuario está dispuesto a pagar por la validación de una transacción, por lo que dependiendo de la importancia de la transacción se puede consentir pagar más a cambio de una mayor velocidad de validación. La tarifa total constituye la suma de la tarifa base y la tarifa prioritaria [22].

Faucet

Los faucets representan una herramienta diseñada para facilitar a los usuarios la obtención de pequeñas fracciones de criptomonedas a cambio de realizar actividades simples, como captcha o pequeñas tareas. El primer faucet de criptomonedas se remonta a 2010, creado por Gavin Andresen, quien entonces era el desarrollador principal de Bitcoin. Este faucet distribuyó 5 BTC a cada usuario que resolviera un captcha simple, totalizando una entrega de 19,715 BTC [**?**]. Este mecanismo pretendía expandir la educación y adopción del Bitcoin entre un público más amplio. A día de hoy los faucet no ofrecen recompensas tan elevadas debido a la apreciación en el valor de las criptomonedas pero siguen teniendo un papel vital en la atracción y educación de nuevos usuarios mediante una vía de acceso sencilla y de bajo riesgo.

Hoy en día este enfoque se ha expandido y los faucets se han popularizado entre las testnets. Estos faucets son esenciales para el desarrollo y la prueba de aplicaciones descentralizadas (dApps), facilitando un ambiente vital para

la innovación y el perfeccionamiento de nuevos productos y servicios en la Blockchain.

Contratos inteligentes

La idea fue conceptualizada por primera vez por Nick Szabo en 1993, visionando una nueva forma de establecer acuerdos digitales. Sin embargo, la falta de una plataforma adecuada mantuvo esta idea en teoría hasta la llegada de la Blockchain con Bitcoin en 2009, y mas notablemente con Ethereum en 2014, que los contratos inteligentes se materializaron prácticamente gracias a la infraestructura que esta tecnología proporciona [6].

Un contrato inteligente es un código que ejecuta automáticamente los términos de un acuerdo entre partes. Los códigos, almacenados en la Block-Chain, son ejecutados automáticamente, cuando se cumplen unas condiciones predefinidas, haciendo cumplir un acuerdo entre dos partes no confiables sin la necesidad de un tercero de confianza. Los contratos inteligentes utilizan la tecnología Blockchain para almacenar reglas, ejecutar automáticamente acciones cuando se cumplen esas reglas y almacenar los resultados en la blockchain. Debido a su naturaleza inmutable y distribuida, ofrecen un nivel de seguridad y confianza superior al de los sistemas tradicionales. Así mismo, al eliminar los intermediarios, ofrecen una reducción de costos y una mayor rapidez en la ejecución de acuerdos.

Sin embargo, se enfrentan a desafíos en cuanto a cuestiones legales, la necesidad de recursos externos a la cadena de bloques, su naturaleza inmutable que dificulta la corrección de errores, problemas de escalabilidad y limitaciones del mecanismo de consenso. Las soluciones de Capa 2, como la Lightning Network y Ethereum Plasma, se diseñaron para abordar los desafíos de escalabilidad y eficiencia de las blockchain de Capa 1. Operan sobre la cadena principal para permitir transacciones más rápidas y con menores costos, manteniendo la seguridad y descentralización [17].

3.6. Tokenización

La tokenización [2] es el proceso de convertir la información delicada o activos del mundo real en representaciones digitales denominadas "tokens", dentro del ecosistema Blockchain. Este procedimiento juega un papel crucial en la protección de datos confidenciales al reemplazar la información original con un token único, el cual no tiene valor fuera de su contexto específico de uso.

La información sensible se almacena en la "bóveda de tokenización-[23] una infraestructura de almacenamiento segura donde los datos originales se cifran y aíslan. El acceso a la bóveda es solo posible a través de rigurosos controles de seguridad y claves de descifrado específicas. A diferencia de los métodos de cifrado que utilizan un algoritmo matemático para transformar datos en un formato ilegible que puede ser revertido usando un clave concreta, la tokenizción no mantiene una relación algorítmica con los datos originales. En consecuencia, los tokens generados no pueden ser revertidos sin un acceso autorizado a la bóveda de tokenización, lo que proporciona una capa adicional de seguridad. Por lo tanto, mientras los datos originales se almacenan en una bóveda de tokens segura, los tokens se distribuyen en sistemas internos para su utilización diaria.

Un elemento importante de los tokens es que, fuera de la relación financiera específica para la que fueron creados, carecen totalmente de valor. Ya que una función de los mismos es representar un valor específico en una relación determinada. Esta característica los distingue de las criptomonedas y otros activos digitales que pueden tener un valor en el mercado abierto. En el ámbito de los pagos y transacciones, los tokens permiten a las organizaciones procesar transacciones y almacenar información de clientes sin exponer los datos críticos a riesgos de seguridad. La generación de un token se realiza mediante contratos inteligentes en la Blockchain, que definen las reglas y la lógica para su emisión, transferencia y anulación. Los contratos inteligentes aseguran que el token sea único y esté vinculado de manera inmutable a los datos o activos correspondientes en la bóveda.

En el ecosistema blockchain existen diversos tipos de tokens diseñados para propósitos específicos [16]:

- **Tokens de Seguridad:** Representa inversiones digitales en activos reales como acciones o bonos, respaldado por activos tangibles y regulado por entidades gubernamentales.

- **Tokens de Gobernanza:** Permiten a los poseedores participar en la toma de decisiones dentro de una plataforma o protocolo, votando en cambios o propuestas.

- **Tokens de Utilidad:** Proporciona acceso a productos o servicios dentro de una plataforma blockchain, sin ser considerado un valor financiero.

- **Tokens Comunitarios:** Recompensan la participación en una comunidad, ofreciendo beneficios como acceso exclusivo o descuentos a los miembros activos.
- **Tokens Vinculados a valores:** Son digitales pero están respaldados por activos físicos como metales preciosos, permitiendo a los inversores negociar activos reales de manera digital.

Todos los tipos de tokens existentes se pueden clasificar en dos grandes grupos, los tokens fungibles y los tokens no fungibles.

Tokens fungibles

La definición de fungibilidad es esencial para entender los aspectos fundamentales de un token fungible. Tomando como referencia una definición ofrecida por el Tesoro Público del Gobierno de España, la fungibilidad se describe como la "Propiedad de un conjunto de valores que los hace plenamente equivalentes entre sí a efectos legales-[7]. La fungibilidad es un concepto que nos rodea en la vida cotidiana, siendo el dinero uno de los mejores ejemplos. Cuando se intercambia un billete de cinco euros por otro billete de cinco euros, se entiende que ambos tienen el mismo valor y son aceptados de la misma manera.

Este principio se puede extrapolar al mundo digital y a la Blockchain. Los token fungibles actúan de forma similar al dinero físico, siendo indistinguibles y equivalentes entre unidades del mismo tipo. El ejemplo más conocido es Bitcoin, convirtiéndolo en una herramienta poderosa para las transacciones digitales.

Tokens no fungibles (NFT)

Los NFT han emergido como una innovación disruptiva en el ámbito del comercio electrónico, especialmente en el mundo del arte digital. A diferencia de los tokens fungibles, cada NFT es una certificación criptográfica que contiene información y códigos de identificación únicos que los hacen irremplazables e intercambiables. Esta característica los hace particularmente adecuados para representar activos digitales únicos y derechos de propiedad en el mundo digital. En el contexto de los contratos laborales, los NFT pueden ser utilizados para tokenizar y asegurar la autenticidad de contratos individuales, garantizando que los términos acordados sean únicos y vinculados inequívocamente a las partes involucradas. [25]

ERC-721

ERC-721 es un estándar propuesto por el desarrollador Dieter Shirley a finales de 2017 que introdujo el concepto de tokens no fungibles en la red Ethereum. [10] Abriendo las puertas a una nueva dimensión de activos digitales únicos, a diferencia de los tokens fungibles basados en el estándar ERC-20.

La creación del estándar ERC-721 fue motivada por la creciente demanda de tokens digitales que pudieran representar de manera única activos individuales. La singularidad de los tokens ERC-721 les dota de gran utilidad en aplicaciones donde el ámbito de autenticidad y la propiedad exclusiva son cruciales. Su uso mas popular se enfoca en el mundo del arte, asegurando la autenticidad y unicidad de diferentes obras, aunque también ha tomado gran relevancia en el ámbito legal. Poniendo de ejemplo este proyecto, con el uso del estándar ERC-721 se puede tokenizar y autenticar contratos laborales, asegurando la transparencia y la inmutabilidad de los términos acordados, verificando el cumplimiento de los acuerdos.

Este estándar cuenta con una serie de propiedades técnicas que lo hacen versátil. Algunas de estas propiedades son la asignación de un nombre, la definición de un balance de tokens dentro de una dirección y la implementación de funciones que permiten la transferencia segura de la propiedad.

4. Técnicas y herramientas

4.1. Herramientas

Se muestra a continuación las herramientas usadas a lo largo del desarrollo del proyecto.

React Native

La elección del framework adecuado juega un papel crucial en el éxito de un proyecto. Dicha elección se complica aún más cuando se carece de experiencia previa en el desarrollo de aplicaciones móviles. Dentro del gran abanico de posibilidades React Native y Flutter emergen como grandes líderes en el sector gracias a sus grandes comunidades y la abundancia de recursos en línea. Por otro lado se pueden descartar directamente opciones como el entorno de desarrollo de iOS(por ejemplo, el uso de Swift y Cocoa Touch) debido a su exclusividad para aplicaciones para iOS debido a que el objetivo de este proyecto es desarrollar una aplicación para Android.

React Native se presenta como mi elección favorita, este es un framework de código abierto creado por Facebook orientado a la creación de aplicaciones nativas tanto en iOS como en Android. React Native esta basado en JavaScript y React, una biblioteca de JavaScript destinada a la creación de interfaces de usuario. Fue lanzado en 2015 con el propósito de superar las limitaciones del desarrollo de aplicaciones móviles basado solo en HTML5, en las que simplemente se adapta aplicaciones web a un entorno móvil. React Native utiliza componentes nativos en lugar de WebViews para la interfaz de usuario, esto conlleva a que las aplicaciones se sientan y actúen como en una aplicación nativa. Uno de los elementos mas importantes de este framework es el React Native Bridge", el cual facilita la comunicación

entre el código JavaScript y los elementos nativos del dispositivos. El puente maneja de forma paralela dos flujos de trabajo, uno ejecuta la lógica de la aplicación en JavaScript y otro gestiona las operaciones de la interfaz de usuario nativa. Esto permite que las aplicaciones en React Native accedan a las características del dispositivo como la cámara o la ubicación, ofreciendo una experiencia fluida para el usuario gracias a características como el "hot reload"

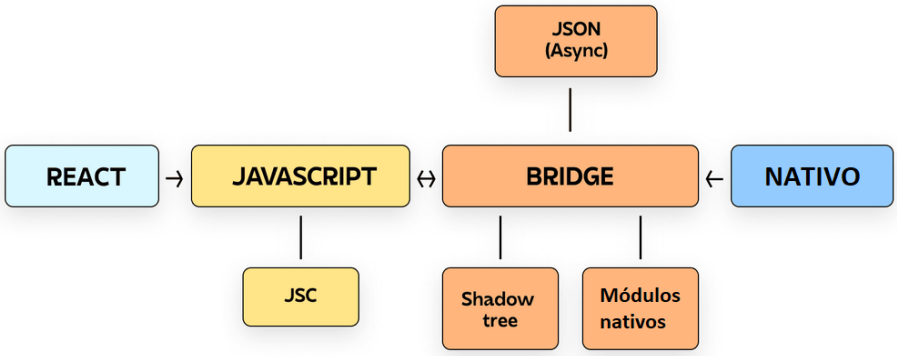

Figura 4.1: Flujo de trabajo y arquitectura de una aplicación React Native

Mi preferencia de este framework sobre el resto radica en mi familiaridad previa con la programación web y JavaScript, lo cual reducirá la curva de aprendizaje en la transición hacia React Native en contraste con otros frameworks que podrían requerir el aprendizaje de nuevos lenguajes de programación o paradigmas de programación. Por otro lado, uno de los grandes motivos de este framework es su gran popularidad, el cual se traduce en una gran riqueza de recursos disponibles, como bibliotecas, videotutoriales o foros, cruciales a la hora de enfrentar los desafíos que puedan surgir. Finalmente aunque para este proyecto no sea un requerimiento hay que tener en cuenta la gran versatilidad de React Native, el cual permite la creación de aplicaciones nativas tanto en Android como en iOS a partir de un único código base. Optimizando así el proceso de desarrollo permitiendo en un futuro poder dar cobertura a un mercado más amplio sin esfuerzos duplicados.

Como había nombrado anteriormente Flutter es otro framework que destaca sobre el resto y ofrece ciertas ventajas notables sobre React Native, como un rendimiento superior gracias al uso de su motor de renderizado propio y la gran capacidad de personalización de la interfaz de usuario. Aunque Flutter pudiera ser superior en algunos aspectos técnicos sigo decantándome por React Native debido a su gran comunidad y la familiaridad que tengo con las tecnologías web, priorizando así un aprendizaje más sencillo frente a posibles mejoras en el rendimiento. Por otro lado existen los frameworks Ionic y Xamarin que los he descartado debido a sus limitaciones en términos de acceso a funciones nativas y comunidades bastante inferiores comparado con React Native o Flutter. Kotlin era un opción con gran potencial para el desarrollo nativo de aplicaciones Android pero la descarté rápidamente por tener una curva de aprendizaje más pronunciada que su competencia, ya que representaría un obstáculo significativo en cuanto al tiempo de desarrollo sin garantizar beneficios proporcionales a dicho esfuerzo.

Respecto a Angular, si bien este framework ofrece un ecosistema robusto para el desarrollo de aplicaciones web dinámicas y complejas utilizando TypeScript, es importante mencionar que también es posible desarrollar aplicaciones móviles con Angular, ofreciendo una experiencia cercana a la nativa, aunque a través de un enfoque diferente al de las aplicaciones nativas desarrolladas con React Native. Angular no deja de ser una elección excelente, pero debido a la preferencia del proyecto de enfocarse en una aplicación móvil sin la necesidad de dar soporte de navegador, Angular puede no ser la opción más adecuada para el proyecto. Por lo tanto he descartado Angular buscando una solución más enfocada y optimizada para el desarrollo móvil, aprovechando las capacidades nativas de los dispositivos móviles.

Expo

Expo es un framework que simplifica el desarrollo de aplicaciones móviles con React Native, actuando como una capa de abstracción reduciendo las barreras de entrada al desarrollo móvil. Consta de un entorno y herramientas listas para usar, eliminando la necesidad de configurar sistemas de construcción nativos complejos. Ofrece un conjunto de APIs disponibles a través de módulos que se pueden invocar directamente desde el código JavaScript las cuales cubren funcionalidades comunes del dispositivo, como cámara, geolocalización o notificaciones sin requerir de configuración adicional.

Expo ha sido fundamental en el proyecto para desarrollar la interfaz de usuario permitiendo probar y visualizar cambios en tiempo real en diferentes

plataformas. Una de las funcionalidades significativas de Expo ha sido su capacidad para facilitar la ejecución y testeo de la aplicación directamente en dispositivos móviles personales sin necesidad de configurar emuladores. Esto se logra mediante el uso de la aplicación ExpoGo en el teléfono móvil y escaneando un código QR generado por el entorno de desarrollo Expo. Expo ha tenido un papel fundamental en las fases de prueba y depuración, permitiendo probar la aplicación en un entorno real y ajustar la interfaz de usuario con un ciclo de retroalimentación casi instantáneo.

Solidity

Solidity es un lenguaje de programación de contratos orientado a objetos y de alto nivel diseñado específicamente para escribir Smart Contracts que se ejecutan en la Ethereum Virtual Machine (EVM). Consta de una sintaxis que recuerda a lenguajes como JavaScript, C++ o Python, lo que facilita su aprendizaje. Es un lenguaje de tipado estático, asegurando que el tipo de cada variable se define en tiempo de compilación y no cambia durante al ejecución del contrato, aumentando así la seguridad y ayudando a la detección de errores antes de la ejecución.

Una de las características más destacadas de Solidity es su soporte para la herencia, una característica común en la programación orientada a objetos. Esto permite que los Smart Contracts hereden propiedades y comportamientos de otros contratos, beneficiándose de la reutilización de código y la organización de la lógica del mismo. En mi caso ha sido de gran utilidad para implementar las funcionalidades del estándar ERC-721 utilizando la biblioteca OpenZeppelin, que ofrece un conjunto de contratos inteligentes auditados y aprobados por su comunidad.

Aunque Solidity no sea el único lenguaje de programación destinado a la creación de Smart Contracts, mi preferencia por este lenguaje se fundamente en dos aspectos claves. En primer lugar, mi familiaridad previa con lenguajes como JavaScript y python hace que la transición a Solidity sea más intuitiva, a diferencia de Vyper cuya sintaxis presentan mayores diferencias. Por otro lado, uno de los requisitos fundamentales de este proyecto era la capacidad del lenguaje para soportar herencia, una característica fundamental para implementar el estándar ERC-721, requisito que Vyper no incluye.

Remix

Remix es un entorno de desarrollo integrado (IDE) diseñado para el desarrollo de Smart Contracts escritos en Solidity. Proporciona una interfaz

accesible y amigable para escribir, compilar, probar y desplegar Smart Contracts directamente desde el navegador, sin la necesidad de instalar software adicional. Remix también ofrece funcionalidades avanzadas como la compilación en tiempo real, el despliegue de contratos en diversas redes de prueba (testnets) y la interacción con SmartContracts ya desplegados. Una de las características más útiles de este IDE es su análisis estático del código, pudiendo así identificar posibles errores de programación o vulnerabilidades de seguridad. Crucial para el desarrollo de Smart Contracts donde los errores pueden suponer consecuencias financieras significativas.

Además, Remix se integra con herramientas y plugins adicionales, ofreciendo un entorna mas rico y extenso permitiendo conectarse con herramientas y servicios como MetaMask, Truffle y Ganache los cuales han sido de gran utilidad en mi proyecto. Remix ha tenido una gran protagonismo en el desarrollo de mi proyecto permitiéndome iterar rápidamente a través de diferentes versiones de contratos inteligentes. Pudiendo ejecutar pruebas unitarias con diversas redes Ethereum como Rinkeby y Goerli, indispensable para validar la lógica del Smart Contract antes de su despliegue final. Dándome una gran capacidad de testeo en un entorno controlado pero realista el cual ha sido crucial para la corrección de errores y la optimización del uso de gas, asegurando así la eficiencia y seguridad de los contratos inteligentes desarrollados.

Infura

Infura es una plataforma que proporciona una API escalable y de alta disponibilidad para acceder a la Blockchain de Ethereum, entre muchas otras. Esta plataforma permite desplegar contratos inteligentes en la Blockchain sin necesidad de mantener un nodo propio lo que representa un ahorro significativo en timpo y recursos. Para ello, Infura proporciona .[en]dPoints"de API que facilitan la lecutra y escritura de datos en la Blockchain. Infura es una propuesta de gran valor para el desarrollo de dApps como la presente, ya que facilita el despliegue y la interacción de contratos inteligentes, asegurando una alta disponibilidad. Esto garantiza que las aplicaciones puedan realizar y recibir transacciones en cualquier momento. Del mismo modo, a medida que el proyecto crece, Infura se adapta a las necesidades de escalabilidad, permitiendo que la aplicación pueda manejar un mayor número de solicitudes de manera eficiente.

Truffle

Tras una fase inicial de desarrollo de los Smart Contracts usando Remix, la transición al uso de Truffle ha marcado un punto de inflexión en la complejidad de mi proyecto. Truffle consiste en una suite de desarrollo avanzada para Ethereum, ofreciendo un conjunto de herramientas diseñadas para facilitar el desarrollo, gestión y despliegue de Smart Contracts. Truffle ofrece un entorno de desarrollo estructurado generando una jerarquía de carpetas para favorecer la implementación de proyectos blockchain complejos. Ahorra mucho tiempo con su sistema de migraciones y scripts de despliegue el cual automatiza y simplifica el proceso de lanzamiento de contratos.

Aunque Remix es una excelente herramienta, Truffle eleva la posibilidad de hacer pruebas unitarias a otro nivel con un marco de prueba mucho mas sofisticado, permitiendo ejecutar test en Solidity o JavaScript.Esta capacidad ampliada para realizar pruebas unitarias contribuye de manera crucial a la calidad y seguridad del código de los Smart Contracts.

Finalmente, uno de las mayores ventajas de usar Truffle en el desarrollo de aplicaciones descentralizadas es en cómo simplifica el proceso de unir el trabajo que se realiza en el backend con el frontend. Truffle proporciona herramientas que que promueven una conexión mas directa y menos propensa a errores entre Smart Contracts y las aplicaciones móviles.

Ganache

Ganache funciona como un nodo de Ethereum personal, permitiendo a los desarrolladores simular un entorno de blockchain que opera localmente ofreciendo un espacio seguro y controlado para experimentar sin ningún costo real ni tiempos de espera al contrario que pasa con las redes públicas de Ethereum. Por defecto, proporciona diez cuentas cargadas con 1000 ETH cada una junto con sus correspondientes claves públicas y privadas, aunque esta configuración puede ajustarse según las necesidades del proyecto.

Disponible tanto como una herramienta en línea de comandos o como una aplicación de escritorio. Ganache proporciona una vista de las transacciones, bloques y estado de la red, mostrando el feedback en tiempo real, vital para realizar una iteración rápida.

La integración de Ganache con Truffle facilita aún más su utilidad, estableciéndose como el entorno de desarrollo local predeterminado dentro del ecosistema Truffle. De esta forma se permite testear la eficiencia de los contratos ajustando parámetros como el tiempo de bloque para simular diferentes condiciones de red.

MetaMask

MetaMask es una extensión de navegador y una aplicación móvil que permite a los usuarios interactuar con la blockchain de manera segura y sencilla. Actúa como puente entre los navegadores web y la blockchain. Su funcionamiento es análogo a una cartera digital, permitiendo a los usuarios almacenar sus cuentas de Ethereum así como el envío de criptomonedas.

Al conectarse a dApps, los usuarios pueden usar sus cuentas de MetaMask para autenticarse eliminando la necesidad de ingresar claves privadas manualmente. Por otro lado, aplica una capa adicional de seguridad al encriptar la información del usuario y almacenar las claves privadas directamente en el dispositivo del usuario. Esto asegura que solo el usuario tenga acceso a sus fondos y datos.

La integración de MetaMask en mi proyecto no solo mejora la experiencia del usuario final, proporcionando una forma más segura e intuitiva de acceder a sus activos, sino que también agiliza el desarrollo de mi proyecto al simplificar la manera en que los usuarios se conectan a la dApp.

Finalmente una característica definitiva de Metamask a diferencia de otras billeteras, es su capacidad para agregar redes personalizadas, de gran utilidad para el testeo de la aplicación, permitiendo la conexión a entornos específicos como el entorno local proporcionado por Ganache o la tesnet de Sepolia.

Node.js

Node.js es un entorno de ejecución en JavaScript, que tradicionalmente ha sido un lenguaje de programación del lado del cliente, para desarrollar aplicaciones del lado del servidor. Es conocido por su capacidad para manejar operaciones asíncronas y por su escalabilidad siendo de gran popularidad en el desarrollo de aplicaciones web modernas. Una de las ventajas de de Node.js es su ecosistema de paquetes gestionados por npm (Node Package Manager), que proporciona acceso a miles de librerías.

En el proyecto se ha usado de manera frecuente, más allá de ser un requisito para utilizar Truffle, a través del comando ïnpm installïpara descargar librerías y paquetes necesarios tanto para el desarrollo del frontend como el del backend. Bien es así, que aunque React Native sea el framework principal del desarrollo del frontend, la gestión de sus dependencias, librerías adicionales se realiza mediante npm, el cual opera sobre Node.js. Por ejemplo el uso de la biblioteca Ëthers;, una de las mas importantes del proyecto ya

que es fundamental para interactuar con la blockchain, se ha instalado y gestionado usando npm. Por tanto, Node.js ha sido una pieza crucial en el desarrollo del proyecto, demostrando su valor no solo como una plataforma de desarrollo del lado del servidor, sino también como un eje central para la administración de paquetes y librerías en el desarrollo de aplicaciones completas.

EtherScan

EtherScan ofrece una plataforma web que permite explorar los bloques de la red de Ethereum. Esta herramienta permite seguir la actualización de la red Ethereum en tiempo real, observando cómo se añaden nuevos bloques y el conjunto de transacciones de cada uno. También permite buscar específicamente una transacción que se ha cometido a lo largo del tiempo mediante su función hash. A parte de mostrar las transacciones en tiempo real, EtherScan permite desglosar cada transacción para ofrecer detalles como el número de bloque, la hora exacta de la confirmación, y datos específicos sobre las acciones dentro de las transacción, tales como transferencias de tokenes ERC-721, el valor transferido, las tarifas de gas, etcétera. EtherScan también sirve como un centro de análisis, en su plataforma podemos encontrar información relevando sobre el estado de la red, cómo estadísticas en tiempo real y datos históricos sobre la dificultad del minado, el precio del gas y otras métricas que pueden ser da gran utilidad para entender el comportamiento de la app en un momento determinado. EtherScan ha tenido un gran protagonismo en la etapa final de mi aplicación donde se ha desplegado el contrato inteligente en una red real de prueba. Con esta herramienta se ha monitoreado la ejecución y rendimiento del contrato, asegurando su correcto funcionamiento y fiabilidad. Por otro lado, ha permitido experimentar y observar la eficiencia del contrato dependiendo de las fluctuaciones en el precio del gas y otras condiciones de la red.

Sepolia PoW Faucet

En las primeras implementaciones del proyecto se ha utilizado ETH ficticios que se generaban en un contexto local, pero en las versiones finales del proyecto y en su despliegue en una red de prueba real ha sido necesario obtener ETH de prueba reales para poder interaccionar con la Blockchain. Al ser ETH de prueba, estos carecen de valor y por lo tanto no se pueden comprar como se haría con ETH de la mainnet, por lo tanto las únicas formas de conseguirlos es minando o que otro usuario te los proporcione. Convertirse en un minero para obtener ETH de prueba es una opción más

compleja, por lo que el uso de las faucet reduce en gran cantidad esta barrera de entrada.

Existen diversas faucets que proveen una cantidad de ETH de prueba diaria, pero en la mayoría para verificar la autenticidad del usuario se requiere que dicho usuario cuente con una pequeña cantidad de ETH reales en su billetera y por lo tanto obligan a hacer una pequeña inversión. Sin embargo, existe una faucet llamada Sepolia PoW Faucet que verifica la autenticidad del usuario mediante el mecanismo de prueba de trabajo (PoW), eliminando la necesidad de poseer ETH reales. Este enfoque requiere que los usuarios realicen un trabajo de cómputo, específicamente generando hashes que cumplan con ciertos criterios preestablecidos, para demostrar su esfuerzo y recibir ETH de prueba a cambio.

4.2. Bibliotecas

Se muestra a continuación las bibliotecas usadas a lo largo del desarrollo del proyecto.

- **OpenZeppelin:** OpenZeppelin es una biblioteca para el desarrollo seguro de Smart Contracts en Ethereum. Ofrece implementaciones auditadas y probadas para minimizar riesgos de seguridad. Su uso ha sido crucial para el desarrollo de mi proyecto, usando el estándar ERC721 para la representación de los contratos como tokens no fungibles (NFTs) asegurando que la aplicación cumpla con los estándares de seguridad.
- **ethers:** Biblioteca caracterizada por su simpleza, ligereza y seguridad para interactuar con la red de Ethereum. Proporciona funcionalidades para crear clientes que requieren comunicarse con la blockchain de Ethereum, permitiendo la gestión de billeteras, la construcción y firma de transacciones, la conexión a diferentes nodos de Ethereum a través de varios proveedores como JSON RPC o Infura, y la interacción con contratos inteligentes.
- **axios:** Biblioteca cliente HTTP basada en promesas para el navegador y para Node.js. Permite hacer solicitudes HTTP a servidores externos de manera sencilla y eficaz.
- **WalletConnect:** protocolo estándar que facilita la conexión segura entre billeteras móviles y aplicaciones descentralizadas (dApps) mediante el escaneo de un código QR o un enlace directo.

- **walletconnect/react-native-compat:** Asegura la compatibilidad de WalletConnect con React Native, simplificando la integración de WalletConnect en aplicaciones móviles

- **Viem:**Ppermite a los desarrolladores interactuar con la blockchain de Ethereum, incluyendo abstracciones de la API JSON-RPC, interacción con contratos inteligentes, implementaciones de billeteras y firmas, utilidades de codificación/descodificación, y más. Viem actúa como una base sólida sobre la cual se construyen herramientas más complejas, como Wagmi.

- **Wagmi:** Wagmi Core es un envoltorio sobre Viem que proporciona funcionalidad multi-cadena a través de la configuración de Wagmi y manejo automático de cuentas mediante Conectores. Resuelve problemas comunes como la conexión de billeteras, soporte multi-cadena, envío de transacciones, escucha de eventos y cambios de estado, y refresco de datos de blockchain.

- **web3modal/wagmi-react-native:** dapta WAGMI y Web3Modal para React Native.

- **react-native-qrcode-svg:** Biblioteca de React Native que permite la generación de códigos QR en formato SVG.

- **expo-barcode-scanner:** Módulo de Expo que ofrece funcionalidades de escaneo de códigos de barras y QR en aplicaciones React Native.

- **react-native-get-random-values:** Polifill que proporciona valores aleatorios seguros para criptografía en React Native, crucial para generar claves y tokens únicos de forma segura.

- **ethersproject/shims:** Facilita la compatibilidad de ethers.js en React Native, permitiendo el desarrollo de dApps mediante la provisión de funciones de criptografía y blockchain en dispositivos móviles.

- **react-native-async-storage:** Biblioteca para almacenamiento asincrónico y persistente de datos, ideal para almacenar preferencias de usuario y configuraciones.

- **react-navigation/native-stack:** Ofrece una experiencia de navegación fluida y optimizada mediante el uso de APIs nativas para la navegación entre pantallas en aplicaciones React Native.

- **react-navigation/bottom-tabs:** Biblioteca para crear barras de navegación en la parte inferior de aplicaciones móviles, facilitando la navegación entre vistas

4.3. Otras herramientas

Esta sección incluye herramientas y programas que tienen una relevancia secundaria en el proyecto.

- **TEXStudio:** IDE para LaTeX.
- **Visual Studio Code:** Editor y depurador de código empleado junto con alguna extensión.
- **GitLab:** Plataforma de control de versiones,
- **Git Bash:** Emulador de línea de comandos para Windows que proporciona herramientas de Git.
- **Notion:** Organizador de tareas utilizado para tomar notas.
- **Mendeley:** Gestor de referencias para organizar investigación.

5. Aspectos relevantes del desarrollo del proyecto

Este apartado pretende recoger los aspectos más interesantes del desarrollo del proyecto, comentados por los autores del mismo. Debe incluir desde la exposición del ciclo de vida utilizado, hasta los detalles de mayor relevancia de las fases de análisis, diseño e implementación. Se busca que no sea una mera operación de copiar y pegar diagramas y extractos del código fuente, sino que realmente se justifiquen los caminos de solución que se han tomado, especialmente aquellos que no sean triviales. Puede ser el lugar más adecuado para documentar los aspectos más interesantes del diseño y de la implementación, con un mayor hincapié en aspectos tales como el tipo de arquitectura elegido, los índices de las tablas de la base de datos, normalización y desnormalización, distribución en ficheros3, reglas de negocio dentro de las bases de datos (EDVHV GH GDWRV DFWLYDV), aspectos de desarrollo relacionados con el WWW... Este apartado, debe convertirse en el resumen de la experiencia práctica del proyecto, y por sí mismo justifica que la memoria se convierta en un documento útil, fuente de referencia para los autores, los tutores y futuros alumnos.

6. Trabajos relacionados

Este apartado sería parecido a un estado del arte de una tesis o tesina. En un trabajo final grado no parece obligada su presencia, aunque se puede dejar a juicio del tutor el incluir un pequeño resumen comentado de los trabajos y proyectos ya realizados en el campo del proyecto en curso.

7. Conclusiones y Líneas de trabajo futuras

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.

Bibliografía

[1] ARTKAI. How much does it cost to create a dapp? https://artkai.io/blog/how-much-does-it-cost-to-build-a-dapp/, 2022.

[2] ASOBANCARIA. Hablemos de blockchain: ¿qué es la tokenización y para qué sirve? https://www.sabermassermas.com/hablemos-de-blockchain-que-es-la-tokenizacion-y-para-que-sirve/, 2021.

[3] André Carneiro. ¿what is r3 corda and how it works | bbchain? https://www.bbchain.com.br/en/blockchain-blog/what-is-r3-corda-and-how-it-works, 2022.

[4] cointelegraph. Qué es ethereum. https://es.cointelegraph.com/learn/what-is-ethereum-a-beginners-guide-to-eth-cryptocurrency/, 2022.

[5] Michele D'Aliessi. How does the blockchain work? https://onezero.medium.com/how-does-the-blockchain-work-98c8cd01d2ae, 2016.

[6] Universidad de Alcalá. Historia de los smart contracts. https://masterethereum.com/historia-smart-contracts/, 2019.

[7] Gobierno de España. Fungibilidad. https://www.tesoro.es/fungibilidad/, 2021.

[8] Ethereum. ¿qué es el ether (eth)? | ethereum.org. https://ethereum.org/es/eth/, 2024.

[9] Juan Fornell. Máquina virtual de ethereum (evm) | ethereum.org. https://academy.bit2me.com/que-es-ethereum-virtual-machine-evm/, 2019.

[10] Juan Fornell. ¿qué es un token erc 721? https://academy.bit2me.com/que-es-token-erc-721/, 2019.

[11] Wan Shicheng-Yu Philip S Gan Wensheng, Ye Zhenqiang. Web 3.0: The future of internet.

[12] Javier Sáez Hurtado. Qué son las dapps o aplicaciones descentralizadas y varios ejemplos. https://www.iebschool.com/blog/dapps-o-aplicaciones-descentralizadas-que-son-y-como-funcionan-finanzas/, 2022.

[13] IBM. ¿qué es hyperledger fabric? https://www.ibm.com/es-es/topics/hyperledger#:~:text=Hyperledger%20Fabric%20es%20una%20plataforma,con%20permisos%22%20(conocidos)., 2021.

[14] Lisa Institute. Qué es bitcoin: origen, usos, ventajas y riesgos. https://www.lisainstitute.com/blogs/blog/que-es-bitcoin-origen-usos-ventajas-riesgos, 2021.

[15] KeepCoding. ¿qué es el doble gasto? https://keepcoding.io/blog/que-es-el-doble-gasto/#:~:text=El%20doble%20gasto%20en%20las,digitales%20que%20representan%20la%20moneda./, 2023.

[16] KeepCoding. ¿qué tipos de tokens existen? https://keepcoding.io/blog/que-tipos-de-tokens-existen/, 2023.

[17] Ghedira-Guegan Chirine Khan Shafaq Naheed, Loukil Faiza and Bani-Hani Anoud Benkhelifa Elhadj. Blockchain smart contracts: Applications, challenges, and future trends. *Peer-to-Peer Networking and Applications*, 14(5):2901–2925, 2021.

[18] José Maldonado. QuÉ son los nodos y su papel clave en la tecnologÍa blockchain. https://observatorioblockchain.com/blockchain/que-son-los-nodos-y-su-papel-clave-en-la-tecnologia-blockchain/#:~:text=Los%20nodos%20en%20blockchain%20son,la%20seguridad%20de%20la%20red./, 2023.

[19] nhsz. Máquina virtual de ethereum (evm) | ethereum.org. https://ethereum.org/es/developers/docs/evm/, 2023.

[20] Scaling Parrots. Blockchain hash: what is it and what is it for. https://www.scalingparrots.com/en/blockchain-hash-what-is-it/#:~:text=A%20Blockchain%20hash%20is%20used,the%20end%20of%20the%20chain./, 2020.

[21] SAP. ¿qué es la tecnología de blockchain? https://www.sap.com/spain/products/artificial-intelligence/what-is-blockchain.html, 2023.

[22] socopower. Gas y tarifas | ethereum.org. https://ethereum.org/es/developers/docs/gas/, 2023.

[23] SoluLab. Vaultless tokenization vs. vault tokenization. https://www.solulab.com/vaultless-tokenization-vs-vault-tokenization/, 2019.

[24] Shamili Sriman, Ganesh Kumar. Blockchain technology: Consensus protocol proof of work and proof of stake. *Advances in Intelligent Systems and Computing*, 1172(6):395–406, 2021.

[25] Hamed Taherdoost. Non-fungible tokens (nft): A systematic review. *MDPI*, 14(26):26–, 2022.

[26] Utimaco. ¿cuáles son los tipos de nodos en blockchain? https://utimaco.com/es/servicio/base-de-conocimientos/tecnologia-blockchain/cuales-son-los-tipos-de-nodos-en-blockchain/, 2022.

[27] Wikipedia. Distributed ledger technology (dlt). https://es.wikipedia.org/wiki/Distributed_Ledger_Technology_(DLT), 2022.