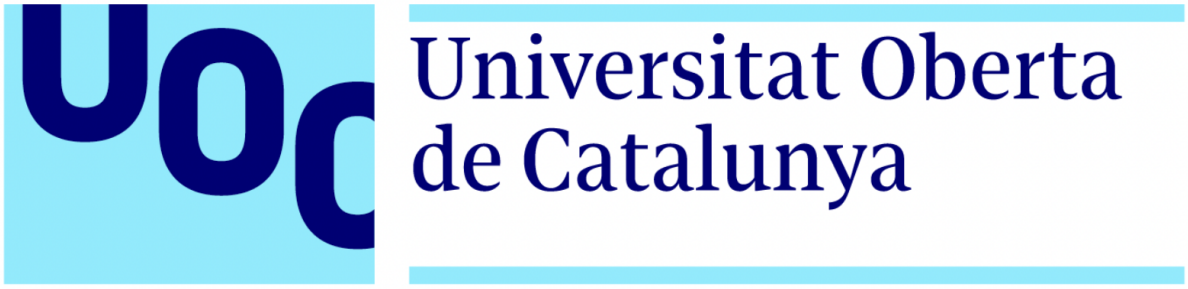


Máster en Ingeniería de Telecomunicación

Área de Sistemas de Comunicación

Curso académico 2023-2024

Trabajo Fin de Máster

Desarrollo de una aplicación descentralizada basada en blockchain para el gobierno de una ciudad inteligente

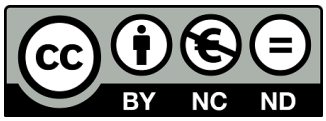
---

Gabriel Domínguez Camarero

Tutor

Víctor Monzón Baeza

FECHA: Enero de 2024



FICHA DEL TRABAJO FINAL

Título del trabajo	Desarrollo de una aplicación descentralizada basada en blockchain para el gobierno de una ciudad inteligente
Nombre Autor/a	Gabriel Domínguez Camarero
Nombre Tutor/a TFM	Víctor Monzón Baeza
Nombre del/de la PRA	Carlos Monzo Sánchez
Fecha de entrega	Enero 2024
Titulación o Programa	Máster en Ingeniería de Telecomunicación
Área del trabajo final	Smart Cities
Idioma del trabajo	Castellano
Palabras Clave	Ciudad Inteligente, Smart cities, Blockchain, Identidad descentralizada

Resumen

Este trabajo estudia el campo de la identidad descentralizada de los ciudadanos para facilitar las gestiones administrativas dentro del contexto de las ciudades inteligentes. Se plantea el uso de nuevas tecnologías como el blockchain, una base de datos y aplicaciones descentralizada, como plataforma única para procesar y almacenar la información de los ciudadanos. A lo largo de este documento, se analizará la situación actual de las ciudades inteligentes y como la tecnología blockchain puede usarse para registrar, validar y comprobar datos en múltiples sectores, como industria, agricultura, transportes, medicina o educación, garantizando la trazabilidad, integridad y anonimato.

Como resultado de todo este análisis se propondrá un prototipo de aplicación descentralizada de aplicación en los campos de medicina y educación para registrar prescripciones de recetas y titulaciones académica, respectivamente. Además, se usará la tecnología NFT para representar visualmente los títulos académicos.

Abstract

This project is included in the scope of the decentralized citizen identity to facilitate administrative processes in the context of smart cities. The paper analyzes the use of blockchain, a decentralized database and application, to use the same platform for storing and processing data.

Throughout this document, the current situation of smart cities will be analyzed, and how the blockchain technology could be used to register, validate, and check data in the fields of industry, transport, agriculture, medicine and education, ensuring the traceability, integrity, and anonymity.

As a result of this analysis, a prototype of a decentralized application will be proposed in the fields of medicine and education. This app will manage data such as prescriptions and titles, the academic title will be represented as NFT to visulize it in a wallet.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

LISTA DE ACRÓNIMOS:

- **AES:** Advanced Encryption Standard
- **API:** Application Programming Interface
- **C-V2X:** Cellular-Vehicle-to-Everything
- **CID:** Content Identifiers
- **DApp:** Aplicación descentralizada
- **DID:** Identidad descentralizada
- **DNS:** Domain Name System
- **DNI:** Documento Nacional de Identidad
- **EAS:** Ethereum Attestation Service
- **ENS:** Ethereum Name Service
- **ETH:** Ethereum
- **EVM:** Ethereum Virtual Machine
- **HTTP:** Hypertext Transfer Protocol
- **IBAN:** International Bank Account Number
- **ILP:** Interplanetary Linked Data
- **IOT:** Internet of Things
- **IPFS:** InterPlanetary File System
- **JPEG:** Joint Photographic Experts Group
- **JSON:** JavaScript Object Notation
- **KYC:** Sistema de validación de identidad
- **LOPD:** Ley Orgánica de Protección de Datos
- **NFT:** Non-Fungible Token
- **PDF:** Portable Document Format
- **PKI:** Public-key Infrastructure

- **PoH:** Proof of Humanity
- **PoS:** Proof of Stake
- **PoW:** Proof of Work
- **RBAC:** Role-Based Access Control
- **RFID:** Radio Frequency Identification
- **RSA:** Rives, Shamir y Adleman
- **SHA:** Secure Hash Algorithm
- **TCP:** Transmission Control Protocol
- **TTL:** Time To Live
- **URL:** Uniform Resource Locator
- **VM:** Virtual Machine

1. INTRODUCCIÓN

En este capítulo se presenta una introducción al desarrollo llevado a cabo en este Trabajo Fin de Máster (TFM). Esto incluye un contexto y justificación sobre las ciudades inteligentes y su necesidad para incluir aplicaciones de Blockchain en sus diferentes sectores. Seguido por los objetivos, fases, metodología y planificación llevadas a cabo para alcanzar la solución final.Ademas, se realiza un analisis del impacto etico-socila, asi como el enfoque y metodo seguido en la ejecucion. Terminamos con un breve resumen de los productos obtenidos al final de este TFM.

1.1. Contexto y justificación

Los gobiernos ofrecen a la ciudadanía mecanismos para interactuar con los servicios gubernamentales, sanidad, educación, registros, etc. Estos mecanismos han evolucionado a medida que lo hacía la tecnología, inicialmente se realizaban en oficinas físicas pero con la llegada de internet y la tecnología de la información apareció un nuevo mecanismo telemático más rápido y eficiente.

Las ciudades están evolucionando hacia lo que se conoce como ciudades inteligentes (smart cities). Una ciudad es la combinación de múltiples sectores transporte, sanidad, educación, industria, energía, telecomunicaciones, agricultura, ganadería y pesca, etc.

En las ciudades existen infraestructuras o servicios críticos, como producción o distribución de energía, redes de comunicaciones, agua, salud o transporte. La información utilizada para mantener y ofrecer estos servicios tiene que protegerse y garantizar que no ha sido manipulada ni accesible por personas no autorizadas para garantizar la evolución de cualquier infraestructura de la ciudad hacia inteligente [1], [2].

La ciudad inteligente se puede monitorizar con sensores cuya información es almacenada y procesada para tomar decisiones automáticas, mejorando la eficiencia de la actividad económica y los servicios.

El problema de estos sistemas es la obsolescencia y el mantenimiento de los sistemas, por lo general, cada administración implementa sus propias plataformas con distintos mecanismo de autenticación. Esta diversidad obliga a la ciudadanía a aprender el funcionamiento de cada una y, desde el punto de vista técnico, el mantenimiento es muy complicado.

Las ciudades inteligentes recogen datos de la población, movilidad, consumo, sanidad, infracciones, etc. El tratamiento y protección de esta información es fundamental, de hecho en 2018 se aprobó la ley de protección de datos LOPD [3] para definir el marco que regula el tratamiento de los datos de los usuarios.

Los mecanismos de protección y el uso que se puede dar a la información es diferente dependiendo de las características de cada conjunto de datos, por ejemplo, un informe médico es información sensible y sólo el paciente y los médicos autorizados deberán tener acceso a dicho dato. En cambio, un título académico es información no sensible, cualquier persona podría verlo pero, sin embargo, lo importante en este tipo de información es garantizar la autenticidad de dicho título, quien lo ha emitido, fecha de emisión y firma.

Como hemos podido ver las ciudades recogen datos que son almacenados, generalmente en servidores de entidades públicas o privadas, y manipulados por sistemas muy dispares que deben ser mantenidos, protegidos, dar garantías de la autenticidad, no repudio e integridad de la información. Además, cualquier cambio en los datos debe quedar registrado para tener una trazabilidad de las modificaciones y las personas involucradas.

La tecnología blockchain ayuda a solucionar estos problemas porque ofrece ventajas como sistema único de autenticación, confianza en la información, automatización de acciones (contratos inteligentes), una plataforma unificada tanto para el almacenamiento como el procesado y el uso de criptografía al generar los bloques y enlazarlos formando una cadena permite que la información se mantenga inmutable, una vez ha sido validada la cadena, la integridad de la información es muy importante en sectores críticos de la sociedad.

En la Figura 1.1 se muestra un diagrama de una solución para un ciudad inteligente donde se expone mediante un sistema de autenticación único una serie de aplicaciones para la ciudadanía, también, se expone el uso de la tecnología blockchain para almacenar la información garantizando su integridad y no repudio al realizar todas las acciones con la identidad descentralizada asignada a cada ciudadano.

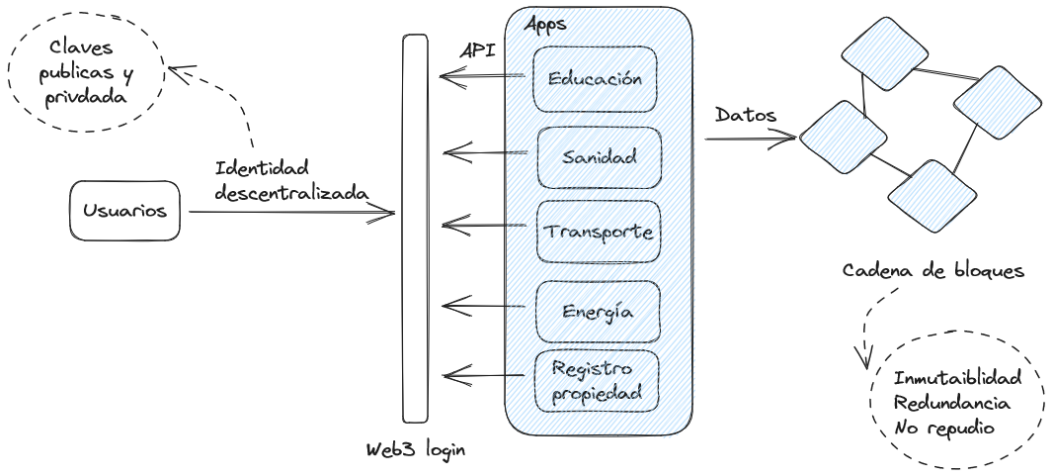


Fig. 1.1. Esquema solución gubernamental con identidad descentralizada en blockchain

Estas ventajas que ofrece la tecnología es la motivación que me ha llevado a realizar este trabajo, desarrollar una solución blockchain para una ciudad inteligente con el objetivo de mejorar los procesos administrativos, registrar la información con garantía de su integridad y aplicar el concepto de identidad descentralizada asociada a identidad real como único

mecanismo de autenticación.

1.2. Objetivos

En esta sección se expondrán los objetivos que se desean conseguir en este trabajo para mejorar los sistemas de una ciudad inteligente, primero se analizarán el uso de blockchain en diferentes sectores, se desarrollara una solución blockchain con contratos inteligentes y NFT para registrar información de salud y educación y se publicará la aplicación en una red blockchain pública.

Para ello, este trabajo establece los siguientes objetivos:

1. Análisis de blockchain en diferentes sectores

   Blockchain puede aplicarse a multitud de casos de uso en la sociedad. En este trabajo fin de máster se analizará la necesidad y ventajas de usar solución blockchain en los sectores analizados en el objetivo anterior.

   - Educación: Determinar la información relacionada con el progreso académico del ciudadano
   - Sanidad: Determinar la información relacionada con la salud del ciudadano

2. Desarrollo de los contratos inteligente que componen la aplicación descentralizada

   En este proyecto se diseñará e implementará una aplicación descentralizada (DApp), desplegada en una red blockchain, para controlar y registrar información de los ciudadanos en los sectores educación y sanidad. Esto incluye:

   - Obtener especificación técnica: Definición de requisitos para la aplicación y los smart contracts
   - Desarrollo de los diferentes bloques de los smart contracts
   - Implementación software: Ensamblado de todos los bloques
   - Pruebas y verificación software

3. Creación de NFT

   En este proyecto se incluirá la creación de NFT que representen los certificados académicos de los estudiantes al finalizar los estudios de cada etapa.

   - Diseñar el modelo JPG o PDF del certificado
   - Automatizar la creación de NFT y acuñarlo en la cadena de bloques
   - Visualizar los NFT en la billetera

4. Integración y distribución

   Una vez que la aplicación ha sido desarrollada y probada en el entorno local se publicará en la red de pruebas de Avalanche Testnet (Fuji) para hacer una prueba de concepto.

5. Publicación en revista

1.3. Impacto en sostenibilidad, ético-social y de diversidad

La agenda 2030 es un plan de acción para alcanzar una serie de objetivos que mejoren la vida en el mundo, luchar contra la desigualdad y el cambio climático. En el ámbito de las Smart Cities se han analizado los conocidos Objetivos de Desarrollo Sostenible (ODS) en [4]. Dentro del ámbito del Blockchain, ya existen algunos estudios como "Blockchain applications and the Sustainable Development Goals" [5] y "Blockchain for Sustainable Development: Promising use cases for the 2030 Agenda" [6] publicados en 2018.

En el ámbito ético-social la blockchain ayudará a democratizar el acceso a la información y su transparencia porque toda las acciones en la red quedan registradas en la cadena de bloques de cada uno de los nodos participantes de la red, no existe un único custodio de los datos, por otro lado, la transparencia de la información se garantiza porque la cadena de bloques guarda la trazabilidad de todos los registros y puede ser revisada en cualquier momento mediante un analizador de bloques.

En el ámbito de la diversidad esta tecnología podría ayudar a proteger a todos los colectivos ya que no tendría que ser necesario conocer el nombre ni el género de la persona; la identidad digital y descentralizada permite interactuar por medio de contratos inteligentes, sin que terceras personas deban tener acceso a la información personal pudiendo valorar o juzgar por razones de género, nacionalidad, edad u orientación sexual.

1.4. Enfoque y método seguido

El presente trabajo analizará la necesidad de proteger la información y su integridad en el ámbito de una ciudad inteligente mediante el uso de la tecnología blockchain. Como resultado de este análisis se implementará una aplicación en una cadena de bloques para almacenar y controlar la información de los sistemas de sanidad y educación.

Esta aplicación se construirá siguiente una metodología práctica donde primero se diseñará la arquitectura software para después desarrollar los contratos inteligentes que conforman la aplicación final utilizando el framework de desarrollo Hardhat y el lenguaje de programación Solidity, por último, la solución será desplegada en una red blockchain pública Avalanche y se digitalizará un activo real como un título académico en formato NFT.

Etapa 1: Análisis

Primero se analizará la situación actual de las ciudades inteligentes, las infraestructuras críticas y la información que almacenan y procesan para ofrecer los servicios. Se analizará la tecnología blockchain, soluciones que ofrece, ventajas y desventajas.

1.4.1. Etapa 2: Preparación entorno

En esta etapa se preparará un entorno de desarrollo y pruebas local, este entorno servirá para implementar la solución, ejecutarla y probar su correcto funcionamiento.

1.4.2. Etapa 3: Diseño y desarrollo

La fase de diseño es las más importante porque es el momento en el que se define la estructura de datos que se almacenará en la blockchain. Los contratos inteligentes se despliegan en la red blockchain y son inmutable por lo que es fundamental planificar el diseño de la aplicación y las relaciones de las clases y los datos.

El diseño incluye:

- Diagramas de clases
- Diagrama de flujo
- Diagrama con el modelo de datos

En la fase de desarrollo se desplegará el entorno de desarrollo, se escribirán los contratos inteligentes, con el lenguaje de programación Solidity, que se ejecutarán en la máquina virtual de Etherum (EVM). Los contratos inteligentes se probarán durante la fase de desarrollo en una red local de pruebas.

1.4.3. Etapa 4: Evaluación y diseminación

Una vez se haya implementado y probado en el entorno local se desplegarán en la red de Avalanche, utilizando la red de pruebas de Avalanche (Fuji).

1.5. Planificación

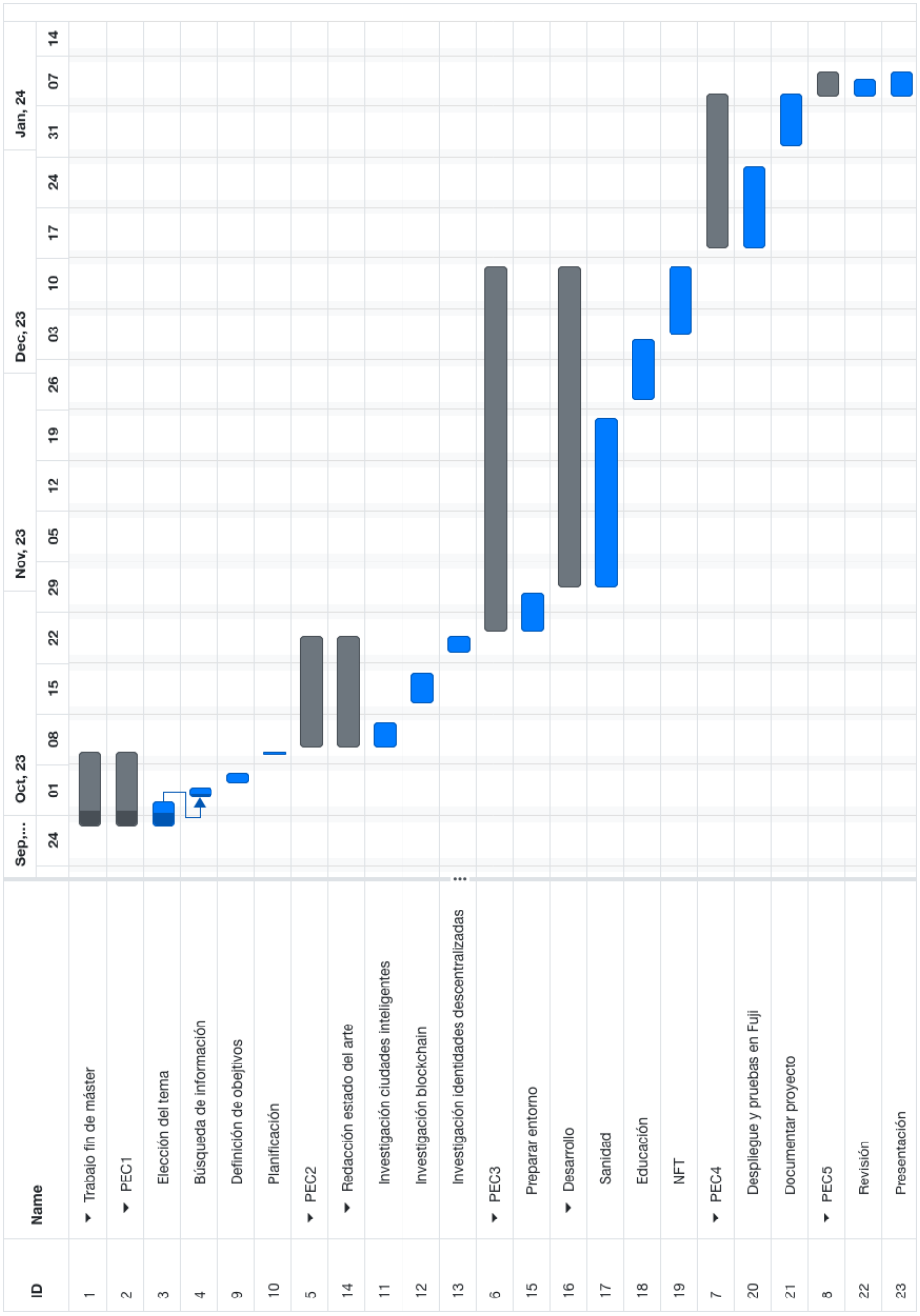


Fig. 1.2. Diagrama planificación de tareas

1.6. Breve sumario de los productos obtenidos

La realización de este trabajo ha permitido desarrollar un solución blockchain en el ámbito de las ciudades inteligentes para manejar la información de los ciudadanos en el ámbito sanitario y educativo.

La información debe ser custodiada y protegida para evitar que sea modificada o robada por personas malintencionadas, la tecnología blockchain evita que la información sea modificada sin autorización y está redundada en múltiple nodos automáticamente al ser un sistema descentralizado.

Los ciudadanos tiene una identidad descentralizada creada en el momento del nacimiento que les permite utilizar los servicios gubernamentales y registro de toda la actividad de esa persona durante su vida.

En este proyecto se ha realizado una prueba de concepto de un posible gobierno basado en blockchain para automatizar los trámites administrativos y digitalizar la información de los ciudadanos asociando su identidad con toda la información que generan durante su vida.

En la aplicación descentralizada desarrollada en este trabajo se muestra la seguridad que ofrece el uso de credenciales basadas en criptografía y evitar la suplantación de una identidad privilegiada. Se implementa un sistema de permisos e identidades basado en roles (RBAC) centralizado y común para toda los contratos, esta DApp de control de acceso tiene unos permisos muy estrictos y muy pocas identidades (owner) tienen capacidad para modificar los registros del contrato.

1.7. Breve descripción de otros capítulos de la memoria

1.7.1. Capítulo 2: Estado del arte

En este capítulo se expone un estudio sobre la situación actual de las ciudades inteligentes, las tecnologías que se han implementado en este tipo de ciudades en los distintos sectores que afectan en la sociedad, energía, industria, transporte, sanidad o educación. Tras haber analizado el las ciudades inteligentes se analiza la tecnología blockchain y las posibilidades que ofrece a los gobiernos y los ciudadanos de una ciudad inteligente.

1.7.2. Capítulo 3: Avalanche Network

En este capítulo se define los principales puntos relevantes de la tecnología blockchain desarrollada por el equipo de Ava Labs. Se analiza la red blockchain de Avalanche y las tres diferentes redes que la conforman X-Chain, P-Chain y C-Chain o incluso redes personalizadas con mayor control. Por último, se expone la relación entre Avalanche y la máquina virtual Etherum lo que hace compatible el desarrollo de aplicaciones descentralizadas en

Solidity con Avalanche.

1.7.3. Capítulo 4: Entorno de desarrollo

El desarrollo de aplicaciones descentralizadas requiere de configurar un entorno de desarrollo de que nos ayude a programar, desplegar y probar aplicaciones en una red local de pruebas. Este capítulo explica los pasos necesarios para configurar un entorno basado en Docker con una red local Avalanche personalizada de 5 nodos y el framework Hardhat para desplegar y probar los contratos en la red.

1.7.4. Capítulo 5: Aplicación propuesta

En este capítulo se explica la aplicación descentralizada desarrollada para el sector de sanidad y educación en una ciudad inteligente gobernada por una red blockchain. Se muestran los diagramas de clase, flujo y el modelo de datos de las aplicaciones.

1.7.5. Capítulo 6: Resultados

En este capítulo se muestran los resultados alcanzados al desplegar los contratos inteligentes y como sería la interacción de los ciudadanos con su identidad digital con la blockchain y la solución desplegada.

1.7.6. Capítulo 7: Conclusiones

Después de mostrar los resultados en el capítulo anterior se exponen las ventajas y desventajas de la solución desarrollada y los pasos futuros que completan el trabajo realizado en este proyecto.

2. ESTADO DEL ARTE

En este capítulo se expondrá la necesidad de evolución de las ciudades hacia ciudades más inteligentes, auto gestionables y cómo la tecnología blockhain puede ayudar a conseguir este hito.

Comenzamos con un repaso de las características de las ciudades inteligentes, analizando los diferentes sectores candidatos a mejorar con tecnología blockchain. Se explicará en que consiste esta tecnología, mostrando una visión general de las principales plataformas blockchain y su uso en el desarrollo de aplicaciones descentralizadas e identidades descentralizadas.

2.1. Smart cities

Las ciudades no son estáticas, evolucionan, al igual que la población que habita en ella, las necesidades que tenían los ciudadanos en 1950 es distinta a las que tenemos en 2023 y será muy distinta a las que puedan tener las personas en 2050. [7]

La humanidad avanza, cada generación es más longeva, hay más habitantes en el planeta consumiendo recursos, por lo que la eficiencia en el uso de los recursos limitados disponibles se ha convertido en uno de los objetivos principales de la sociedad actual y futura.

El concepto ïnteligente"(smart) aplicado a elementos cotidianos implica el uso de la tecnología para dotar a estos objetos de la capacidad de monitorizarse, notificar cambios y actuar aplicando los criterios establecidos. Este concepto ïnteligente"se comenzó a popularizar con la telefonía (smartphone) y, desde ese momento, se ha expandido a televisiones, automóviles, electrodomésticos, etc. Prácticamente todo lo que nos rodea está evolucionando y está adquiriendo capacidades nuevas.

La tecnología es uno de los medios que disponemos para optimizar los recursos, concretamente, el Internet de las cosas (Internet of Things, IoT) etc, [7] nos ayuda a captar los cambios del entorno en tiempo real, procesar la información y tomar decisiones automatizadas. IoT necesita de muchas otras tecnologías para ser efectiva como sistemas de comunicaciones, procesamiento de datos y almacenamiento de la información.

En este ámbito de ciudades inteligentes la ciudad de Madrid ha iniciado un proyecto de innovación MiNT [8] que permitirá al ayuntamiento gestionar los servicios de basura, alumbrado, control del tráfico o notificaciones de posibles incidencias a los ciudadanos,

En una ciudad interactúan muchos sectores como industria, energía, transporte, residuos, servicios de emergencias, etc. En la Figura 2.1 se muestra un diagrama de una ciudad inteligente y los sectores en los que aplicar esta tecnología.

Fig. 2.1. Sectores y concepto de una ciudad inteligente [9].

2.1.1. Industria

La industria 4.0 [10] ayuda a la empresas a fabricar los productos con un margen en coste inferior a la industria tradicional, para ello se apoyan en el uso de sensores y tecnologías como cloud computing, big data e inteligencia artificial. Una fábrica inteligente está equipada con maquinaria capaz de tomar decisiones en tiempo real en base a los datos que recogen los sensores, mejorando y agilizando la toma de decisiones.

El objetivo principal de cualquier empresa es generar los mayores beneficios con el menor coste posible, por este motivo, la eficiencia en los procesos de producción es crucial. La cuarta revolución industrial tiene como objetivo mejorar la eficiencia de los procesos de producción y actualmente, todas las empresas están iniciando su camino para lograrlo.

2.1.2. Energía

La electricidad es la sociedad actual es un servicio esencial, para hacer llegar la energía a los hogares, empresas, fábricas, etc es necesario desplegar una red de suministro por toda la geografía, mantenerla y coordinarla. En el negocio de la energía conviven múltiples agentes que ayudan a este objetivo, estos agentes son productores, distribuidores, comercializadoras, transportista y operadores. Los productores son empresas encargadas de producir la energía y entregarla al transportista, el transportista se encarga de distribuir esa energía hacia las subestaciones localizadas cerca de los consumidores y conectarse con la red de la distribuidora para transportar la energía hasta el usuario final.

Como se puede ver la distribución de la energía es complicada e intervienen múltiples entidades que tienen que estar coordinadas, sin embargo, lo más difícil en todos este proceso es cumplir con la regla básica de la energía, "la energía producida debe ser igual a la demanda", es decir, los productores deberán generar la cantidad exacta de energía necesaria para todos los consumidores en tiempo real. Predecir el consumo de energía futuro es complicado y esa es la función de la operadora, que en el caso de España esta responsabilidad recae en Red Eléctrica.

Red Eléctrica [11] mediante modelos matemáticos es capaz de predecir el consumo de energía futuro para informar a los productores de si deben aumentar o disminuir su producción. La tecnología puede ayudar a obtener más información del consumo real de los usuarios y mediante inteligencia artificial o big data predecir de forma mas precisa cual es será el consumo.

La transición energética hacia una producción más renovable ha provocado que los consumidores se conviertan en pequeños productores, complicando aún más la gestión de la energía. En el nuevo marco energético, como apunta Repsol en su informe [12], el negocio de la energía se va a descentralizar permitiendo a cualquier usuario convertirse en productor.

Las redes inteligentes [13] o smart grid [14] pretenden solucionar este problema y, además, hacer el sistema tradicional mucho más eficiente en términos de producción, distribución y mantenimiento de la red. Las redes inteligentes incorporan sensores y contadores inteligentes para predecir el consumo y posibles incidencias en la red.

2.1.3. Transporte

El transporte de mercancías también esta sufriendo importantes cambios con la llegada de las comunicaciones 5G y la tecnología IoT. Con estas tecnologías los transportistas y los consumidores podrán conocer no sólo la ubicación del producto en tiempo real si no también pueden conocer las condiciones en las que ese producto está siendo transportado. En algunas industrias como la industria del vino es muy importante conocer cierta información que garantice la calidad del producto, por ejemplo, la temperatura en la que se transporta. La empresa eProvenance [15] etiqueta con NFC o RFID cada botella de vino en origen y de esta manera durante el transporte puede conocer el estado de temperatura de cada una, estos datos se registran en una plataforma propietaria y así el consumidor podrá tener acceso a estos datos y comprobar la cadena del frío.

El transporte o movilidad en las ciudades también esta evolucionando, el objetivo que tienen en este caso la tecnología es lograr la comunicación entre la vía y los vehículos [16] aumentando la seguridad, o hacer la movilidad más eficiente gracias al proceso de los datos en tiempo real para modificar la frecuencia de los semáforos o buscando rutas alternativas.

La gestión del tráfico en las grandes ciudades ayudará a reducir la contaminación consiguiendo reducir el tiempo que los vehículos están emitiendo CO2. Los sistemas IoT [17] permiten recoger datos y ajustar en tiempo real la frecuencia de los semáforos, los límites de velocidad o incluso habilitar carriles rápidos para el transporte público.

De nuevo, Telefónica es una de las empresas pioneras en diseñar e implementar los primeros prototipos como el túnel de Cereixal (Lugo) [18] donde se está probando el concepto C-V2X (Vehículo a todo, Cellular-Vehicle-to-Everything, por sus siglas en inglés). El túnel dispone de sensores y cámaras que junto con 5G es capaz de enviar información a los conductores avisando de cualquier incidencia: condiciones meteorológicas a la salida, obras,

avisos de vehículo lento, posible congestión, accidente, obstáculo en carretera, presencia de peatón, vehículo en sentido contrario o frenada brusca durante su trayecto por el mismo, además del aviso de entrada de un vehículo de emergencias, son algunos de estos posibles mensajes.

El coche autónomo es otro de los grandes avances que empresas como Tesla están desarrollando, Tesla es un fabricante de coches disruptivo por muchos factores, dispone de un sistema de navegación abordo único, los vehículos utilizan la energía eléctrica en lugar de motores de combustión y un sistema de conducción autónoma (autopilot) [19]. La tecnología de conducción autónoma de Tesla utiliza cámaras y sensores del vehículo para recoger información del exterior y con inteligencia artificial es capaz de reconocer las señales, el tráfico, la vía y cualquier elemento que afecte a la circulación del vehículo, con todo esto, el ordenador de abordo toma decisiones para acelerar, frenar, cambiar de carril, girar, etc.

2.1.4. Sector primario

La agricultura, ganadería y pesca también han tenido que evolucionar, el la era de la automatización adaptaron las granjas y áreas de cultivo con robot y maquinaría que automatizaban las tareas, ahora abordan una nueva revolución con la llegada de la era de la información. Al igual que hemos visto en los sectores anteriores, el sector primario utilizará los sensores para recopilar, almacenar y procesar datos que ayuden a ser más eficientes.

El concepto smart agro [20] aporta importantes novedades para el sector:

- Asesoramiento o automatización de una gestión más eficiente del riego y la fertilización.
- Monitorización de cultivos desde drones, satélites o sensores en las fincas o animales.
- Integración automática y bidireccional de datos con maquinaria agrícola, que convierten la agricultura de precisión en una realidad práctica.
- Control individual de cabezas de ganado, con múltiples parámetros vitales.
- Gestión de trazabilidad detallada desde el campo hasta el consumidor.

Munich Reinsurance América [21] ha realizado un informe en el que concluye que 74 de cada 100 agricultores utiliza drones para evaluar, controlar y gestionar su explotación. Los drones permiten sobrevolar grandes áreas, grabar o medir con sistemas incorporados en el dron datos de interés para el ganadero o agricultor y así actuar en consecuencia.

Telefónica, una empresa líder en servicios de comunicaciones, ha presentado Vertical Green, un proyecto en colaboración con Onubafruit para hacer más sostenible su producción, se trata de un cultivo interior con acceso remoto y controlado a través de IoT, 5G, ciberseguridad y machine learning.

2.1.5. Sanidad

El sector sanitario está incorporando nuevas tecnologías, principalmente inalámbricas, para mejorar los sistemas de monitorización de los pacientes [22]. Estos nuevos dispositivos IoT permiten mejorar la atención del paciente, mejorar los resultados de los diagnósticos y pruebas y reducción de costes al realizar las tareas más rápido y eficiente.

Uno de los dispositivos, sin ser propiamente un dispositivo de salud, que más datos y servicios de salud puede ofrecernos a las personas es el Apple Watch [23]. Este reloj incorpora sensores que miden nuestras constantes vitales y nos ofrecen información para llevar una vida más saludable, además, es capaz de detectar infarto [24] [25] o accidentes de vehículos y alertar directamente a los servicios de emergencias [26].

Con estos sistemas la sociedad está generando datos constantemente, estos datos sanitarios deben protegerse adecuadamente y garantizar que no han sido modificados para diagnosticar lo mejor posible las enfermedades de los pacientes.

2.1.6. Educación

El IoT está revolucionando el sistema educativo creando entornos más dinámicos, colaborativos y fomentando la innovación [27]. Los alumnos cuentan con dispositivos digitales para acceder a un abanico de recursos más amplio, las dinámicas de aprendizaje están cambiando con la llegada de la realidad aumentada y la realidad virtual, convirtiendo el entorno físico o crean aulas virtuales a medida de la asignatura y el temario.

La enseñanza en línea ha sido otra de las disrupciones, la deslocalización del profesor y el alumno y el no requerir un espacio físico ahorran los costes de estudiar, por lo que, más personas pueden tener acceso a una educación de calidad.

En educación es importante recopilar y analizar información sobre los estudiantes, profesorado y materias para mejorar la calidad de la enseñanza. Esta información debes protegerse y sobre todo garantizar su integridad. La integridad de los registros del sistema educativo es fundamental para garantizar que un estudiante ha obtenido una calificación o título.

2.2. Tecnología blockchain

2.2.1. Inicios del blockchain

En el año 1991 el informático y criptógrafo Stuart Haber junto con el físico W.Scott Stornetta [28] empezaron a trabajar en un proyecto que consistía en crear un cadena de documentos cuya marca de tiempo estuviera protegida e inalterable. En los inicios de su investigación cada documento se almacenaba en un único bloque de la cadena pero más tarde mejoraron el diseño inicial para poder guardar múltiples documentos en un mismo

bloque introduciendo una estructura de datos conocida como árbol de Merkle.

Un árbol de Merkle es un modelo de datos que basa su estructura en nodos identificados por funciones hashes. Una función hash se caracteriza por tener dimensión fija, independientemente de la longitud de la entrada, por lo que es muy útil cuando se quiere manejar grandes cantidades de datos en los que no se necesita conocer el contenido.

El árbol de Merkle [29] se crea en forma piramidal construyendo nodos formados por la suma de hashes de los nodos inferiores hasta llegar al nodo raíz que tendrá el hash creado a partir de los hashes dos penúltimos nodos del árbol, este nodo raíz servirá para validar todo el árbol.

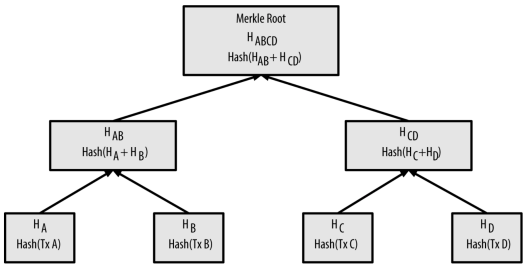


Fig. 2.2. Árbol de Merkle

En Bitcoin los árboles de Merkle son fundamentales para agilizar el proceso de minado ya que todas las transacciones de un bloque son usadas para calcular el árbol y obtener el hash del nodo raíz. La cabecera de cada bloque [29] tiene asociado un hash que lo identifica y sirve para garantizar la integridad de los datos que contiene.

La base de los algoritmos de minado consiste en resolver problemas de cálculo del Hash que identifica cada uno de los nodos, existen multitud de algoritmos usados en distintas redes blockchain [30]. En el caso de Bitcoin el algoritmo debe calcular un hash que cumpla una cierta condición, el hash se calcula a partir de de los datos de las transacciones del bloque (parte variable del bloque) + los datos de la cabecera, como se explica en la Sección 2.2.2.

El hash del nuevo bloque incluye el hash del bloque anterior por lo que cada bloque permite validar la integridad de todos los bloques anteriores, ya que cada cambio en cualquiera de los bloques anteriores generaría un hash radicalmente distinto que daría como resultado un hash en el bloque root distinto. Es esta propiedad [30] la que ha permitido a las cadena de bloques ofrecer un sistema de registro rápido que garantice la integridad de todos los datos.

La investigación llevada a cabo por estos dos científicos fue la base de lo que posteriormente en el año 2008 Satoshi Nakamoto publicó el paper "Bitcoin: A Peer-to-Peer Electronic Cash System"[31].

2.2.2. Bitcoin: A Peer-to-Peer Electronic Cash System

Sathoshi [31] propone un nuevo mecanismo para realizar transacciones económicas, transferencias y pagos entre usuarios o entidades de forma descentralizada. Esta nueva tecnología cambia el paradigma del sistema financiero centralizado actual, donde los bancos y los procesadores de pago controlan y dan garantías de la legitimidad de todas las transacciones.

El sistema que propone Sathoshi consiste en crear un activo escaso y limitado que pueda ser transferido a través de un red descentralizada. Los usuarios dispondrán de una billetera, compuesta por una dirección pública y una clave privada, la clave privada se usará para firmar la transacción y las direcciones públicas son conocidas por otros usuarios, como las direcciones IBAN de un banco tradicional.

El problema que plantea Sathoshi con este sistema es que no se puede garantizar que un usuario transfiera el "mismo dinero.ª otro usuario, como solución, Sathoshi propone un mecanismo de minado en el que los mineros deberán realizar operaciones criptográficas para validar la cadena de transacciones o bloques. Cada vez que un nuevo bloque se quiere añadir a la cadena, el primer minero capaz de solucionar el problema criptográfico y añadir el nuevo bloque a la red será recompensado.

Proof of Work o mecanismo de prueba de trabajo consiste en calcular el hash del bloque que cumpla con la condición de comenzar por N ceros. Para conseguir el hash de bloque se usarán los datos de la cabecera + los datos del cuerpo del bloque, que contiene todas las transacciones. Todos los campos del bloque son estáticos salvo el campo 'nonce' de la cabecera que es modificado iterativamente hasta encontrar un valor que genere a la salida el hash deseado.

Calcular el hash de las transacciones en cada iteración tiene un coste computacional muy elevado y no es necesario porque son datos que no varían. Los árboles de Merkle ayudan a hacer el proceso más rápido ya que tan solo hay que calcular el hash root del árbol de todas las transacciones del bloque, como se muestra en la Figura 2.3, y usar ese hash como representación de todas las transacciones en cada iteración junto con la cabecera y el nonce.

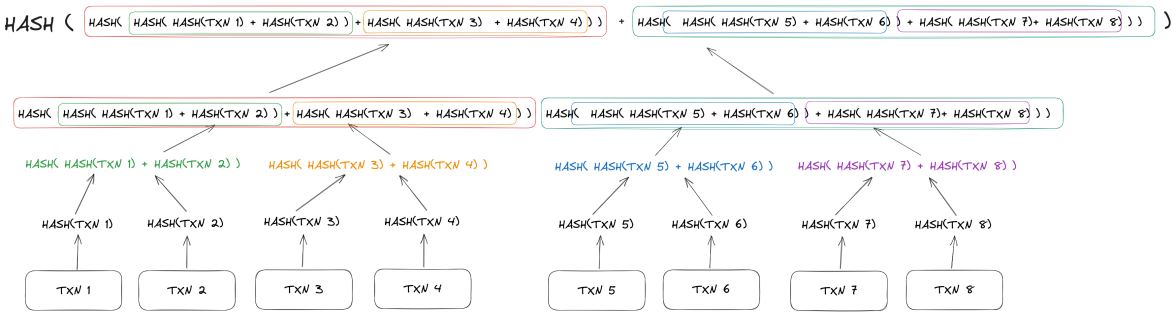


Fig. 2.3. Proceso cálculo del Hash root del árbol de Merkle de las txns

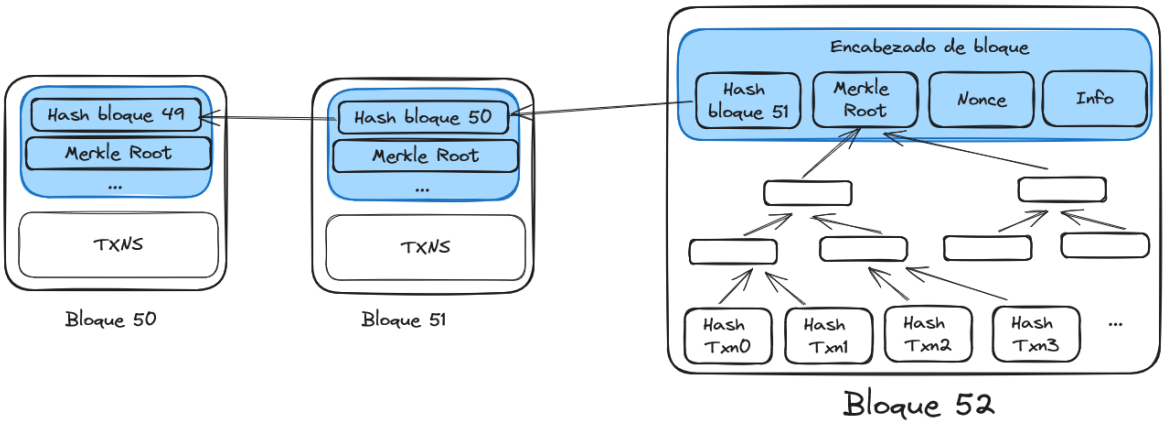


Fig. 2.4. Creación de un bloque a partir de sus transacciones

El JSON de la Tabla 2.1 corresponde al bloque "000000000000000000002024ab333efdd857dd19c172c21994fcf5256e8308245"de la red Bitcoin, se observa como cada bloque incluye mucha información como el nonce que se ha calculado para generar el Hash de bloque, la referencia al hash del bloque anterior, el tamaño del bloque (depende del número de transacciones), el minero que añadió al bloque, etc.

A medida que se generan nuevas transacciones en la red, pendientes de ser aprobadas, son difundidas a todos los nodos de minería, en tiempo real esos nodos intentan calcular el hash del nuevo bloque con las transacciones pendientes, cuando un nodo encuentra un hash que cumpla con la prueba de trabajo notificará a la red que un nuevo bloque ha sido añadido y en ese momento la cadena de bloques incluirá el nuevo bloque añadiendo incluyendo la referencia del hash del bloque anterior.

La red de Bitcoin ha servido para demostrar al mundo que es posible transferir activos (monedas virtuales) entre dos personas en un sistema descentralizado y seguro.

```
{
    "hash": "00000000000000000002024ab333efdd857d...",
    "height": 794738,
    "merkleroot": "6a1bc61c8fc0bf03aeee94a727bef9eb093789e7d49ccd129...",
    "time": 1686992952,
    "mediantime": 1686989854,
    "nonce": 662087795,
    "difficulty": 52350439455487.47,
    "nTx": 3186,
    "previousblockhash": "00000000000000000046a64ac48790e...",
    "strippedsize": 676443,
    "size": 1963671,
    "weight": 3993000,
    "tx": "See 'Transaction IDs'"

    ...

    "totalFees": "0.26127925",
    "miner": {
        "name": "Foundry USA",
        "link": "https://foundrydigital.com/",
        "identifiedBy": "coinbase tag 'Foundry USA Pool'"
    },
}
```


Código 2.1. Ejemplo bloque

2.2.3. Principios de la tecnología blockchain

2.2.3.1. Modelo de datos

La estructura de datos de una red blockchain es una lista enlazada, 'linked list', de nodos denominados bloques, como se muestra en la Tabla 2.2. Cada uno de estos bloques contiene un enlace al bloque anterior y los datos que contiene es una lista de transacciones.

La novedad que aporta blockchain radica en la forma en la que se eligen los bloques que son añadidos a la lista enlazada mediante un algoritmo de consenso, los participantes de la red son los encargados de mantener una copia del histórico de bloques, crear y añadir nuevos bloques.

En los inicios del blockchain las transacciones sólo permitían transferir el activo subyacente de la red, moneda virtual, pero Etherum revolucionó las posibilidades de la tecnología permitiendo programar acuerdos entre partes que ejecuten acciones automáticas cuando se cumplan ciertas condiciones pre-programadas. Básicamente, Etherum hizo la blockchain una tecnología de uso general para cualquier caso de uso, dotando a los desarrolladores de las herramientas necesarias para descubrir todas las posibilidades de ejecutar programas y almacenar datos en una red descentralizada.

```
typedef struct block {
    struct block *prev;
    tx data;
} node_t;
```

Código 2.2. Estructura en C de un bloque

2.2.3.2. Criptografía

La criptografía es una técnica creada para proteger la información de persona que no tienen autorización para conocer su contenido, utilizando algoritmos matemáticos. La historia de la criptografía se remonta muchos siglos atrás, en una era más moderna durante la Segunda Guerra Mundial se crearon máquinas para proteger la información militar, la más conocida fue la máquina Enigma.

Máquina Enigma

Enigma [32] fue un invento, utilizado por el ejército Alemán, similar a una máquina de escribir pero añade un componente adicional, un sistema de tres rotores. Los rotores son extraíbles y en su canto tienen dibujadas las 26 letras del alfabeto, dependiendo de como se coloquen el mensaje resultante varía, es decir, los rotores son el 'algoritmo' de cifrado y descifrado.

El funcionamiento interno de la máquina consiste en conectar las teclas del teclado con el primer rotor, con cada pulsación el rotor gira N posiciones, donde N es configurable, y envía un pulso eléctrico al segundo rotor y este al tercer rotor cuya salida es una letra distinta a la pulsada en el teclado. La existencia de tres rotores es para aumentar la seguridad de la máquina, ya que por cada vuelta completada de cada rotor el siguiente rotor avanza una posición.

El mensaje cifrado por la máquina de cifrado se puede enviar por cualquier medio de transmisión hacia el receptor que deberá conocer la configuración de la máquina emisora y posicionar los rotores en el mismo orden para descifrar el mensaje. Por tanto, la seguridad de este sistema radica en proteger la configuración y posición de los rotores.

Criptografía por ordenador

La llegada de la tecnología y los primeros ordenadores cambio la forma en la que la información se protegía, ya no era necesaria el uso de máquinas con mecanismos mecánicos [33] para generar permutas de letras; ahora los ordenadores tenía capacidad de cálculo mucho más rápida por lo que se pudieron desarrollar algoritmos de cifrado más sofisticados y complejos.

En la era de la computación los algoritmos basan el proceso de cifrado en operaciones matemáticas e iterativas [34] para generar el mensaje cifrado, normalmente se necesitan dos elementos como entrada del algoritmo, el mensaje en claro y la clave de cifrado. El algoritmo matemático realiza una serie de permutas entre la clave y el mensaje para generar en su salida el mensaje cifrado, como se observa en la Figura 2.5.

En el mundo de la informática existen dos tipos de criptografía: simétrica y asimétrica.

La criptografía simétrica [34] utiliza una única clave para cifrar y descifrar la información, esta clave es compartida con todos los interlocutores de la comunicación y por tanto, debe ser protegida para que otras personas no autorizadas puedan disponer de ella. Existen multitud de algoritmos de cifrado simétrico pero uno de los más utilizados es Advanced Encryption Standard (AES) [35]

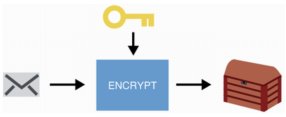


Fig. 2.5. Flujo cifrado simétrico [36]

El mensaje cifrado se envía al receptor que utilizará junto con la clave única y compartida como entrada del algoritmo simétrico en modo descifrado, el algoritmo realizará el procedimiento de descifrado y generará el mensaje en claro, como se muestra en la Figura 2.6.

En la Figura 2.7 se observa como mediante algoritmos simétricos cualquier persona que no disponga de la clave de cifrado no podrá conocer el contenido original del mensaje, una vez el mensaje ha sido cifrado puede enviarse por un medio público y no seguro.

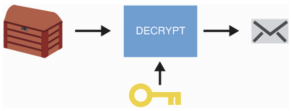


Fig. 2.6. Flujo descifrado simétrico [36]

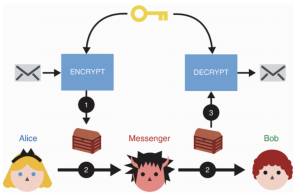


Fig. 2.7. Entorno de seguridad de un algoritmo simétrico [36]

La criptografía asimétrica [34] se diferencia de la simétrica porque en lugar de usar la misma clave para las operaciones de cifrado y descifrado, se utiliza una pareja de claves. La pareja de claves son complementarias, es decir, lo que es cifrado por una clave se descifra con la contraria, estas claves reciben el nombre de clave privada, conocida sólo por el propietario de la pareja de claves, y clave pública, conocida por cualquier usuario que pertenezca a ese entorno de seguridad. Uno de los algoritmos más utilizados en criptografía asimétrica es Rives, Shamir y Adleman (RSA) [37].

El principal uso de la criptografía asimétrica son los entornos de infraestructura de clave pública (PKI) [38]. Un entorno PKI garantiza las dimensiones de seguridad de confidencialidad, autenticidad, integridad y no repudio. Confidencialidad porque la pareja de claves permiten cifrar los mensajes, aunque el cifrado y descifrado es más lento que los algoritmos simétricos, por lo que no suelen utilizarse para este propósito. Autenticidad, integridad y no repudio porque al utilizar la clave privada, conocido sólo por el propietario del certificado, para firmar el mensaje y el hash asociado sólo su clave pública podrá validarlo y garantizar quién es la persona que realizó la acción.

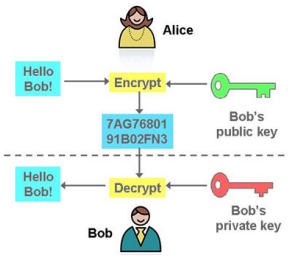


Fig. 2.8. Entorno de seguridad de un algoritmo asimétrico (cifrado) [39]

Como se observa en los diagramas de las Figura 2.9 para firmar un mensaje se calcula el HASH del mensaje y luego se cifra con la clave privada, aplicando el algoritmo de cifrado RSA. El mensaje en claro, junto con el hash cifrado y el certificado digital, que contiene la clave pública, se envía al receptor o receptores.

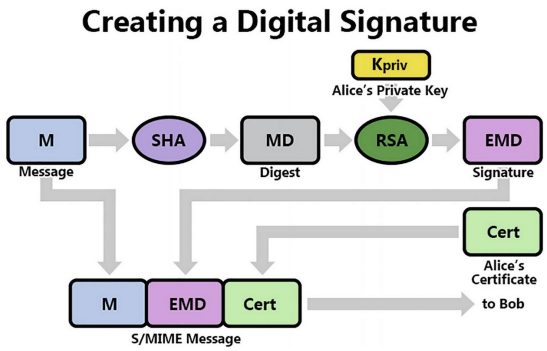


Fig. 2.9. Proceso de firma de un documento [38]

Como se observa en los diagramas de la Figura 2.10, para validar la firma de un mensaje se calcula el HASH del mensaje. El hash cifrado. se descifra, con la clave pública del certificado y se compara con el hash calculado; si son iguales se puede garantizar la identidad del emisor y la integridad del mensaje.

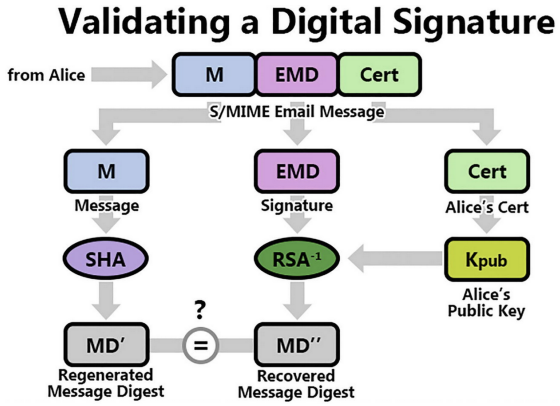


Fig. 2.10. Proceso de validación de la firma [38]

La criptografía asimétrica y el algoritmo RSA son las base de las cadenas de bloques, en blockchain las parejas de claves se denominan billeteras (wallet) y se utilizan para firmar las transacciones garantizando la integridad, autenticidad y no repudio.

2.2.3.3. Creación de billeteras

Para acceder a los recursos de las redes blockchain es necesario crearse una billetera. Una billetera es un conjunto de clave publico - privada generada mediante un algoritmo criptográfico asimétrico como Rives, Shamir y Adleman (RSA).

Existen varios tipos de billeteras

- Billetera software
- Billetera en papel

- Billetera hardware

Las billeteras software pueden ser aplicaciones de escritorio, móviles o en el navegador. En el mercado existen multitud de billeteras software, por ejemplo, Phantom [40], Solflare [41], Core [42], Exodus [43], Atomic Wallet [44]; pero la más conocida es Metamask [45], que se instala como complemento en el navegador y mediante una lista de palabras se crean o se desbloquea la clave privada con la que se accede a los activos o los contratos inteligentes.

Las billeteras en papel son las menos usadas porque las claves publicas y privadas son calculadas mediante un algoritmo y se imprimen directamente en papel para ser custodiadas, si se desea operar con la blockchain es necesario escribir las claves continuamente para autorizar las transacciones.

Las billeteras por hardware son las más seguras, generalmente, son unidades extraíbles USB que integran un chip con una zona segura protegida con un algoritmo de cifrado simétrico mediante una clave, definida por el usuario. En esta zona segura se almacenan las claves de las billeteras y mediante un software que actúa como interfaz con la blockchain permite autenticarse. En este tipo de billeteras destacan Ledger [46] y Trezor [47].

Si se desea crear una billetera software para Etherum lo primero que se debe hacer es instalar Metamask en nuestro navegador. Una vez ha sido instalada abrir la extensión y aparecerá la opción de crear una billetera, como se muestra en la Figura 2.11, elegir la contraseña para proteger el acceso a Metamask, esta contraseña se puede olvidar porque tan sólo protege el acceso a Metamask en este dispositivo.

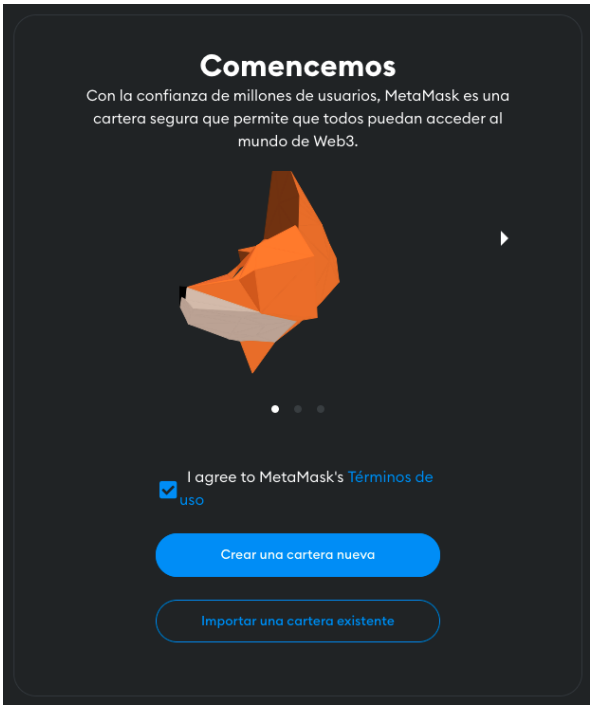


Fig. 2.11. Metamask ventana inicial

Seleccionar la opción de mostrar lista de palabras (seed) asociada a nuestra billetera

Etherum, como se muestra en Figura 2.12. Esta lista de palabras es lo que permite crear la pareja de claves de nuestra billetera por lo que debe almacenarse en un lugar privado y seguro; en caso de perder estas 24 palabras se perderá el acceso a la billetera y a todos los activos asociados.

Fig. 2.12. Metamask seed

Completando los pasos anteriores la billetera se habrá creado con éxito, como se muestra en las figuras 2.13 y 2.14.

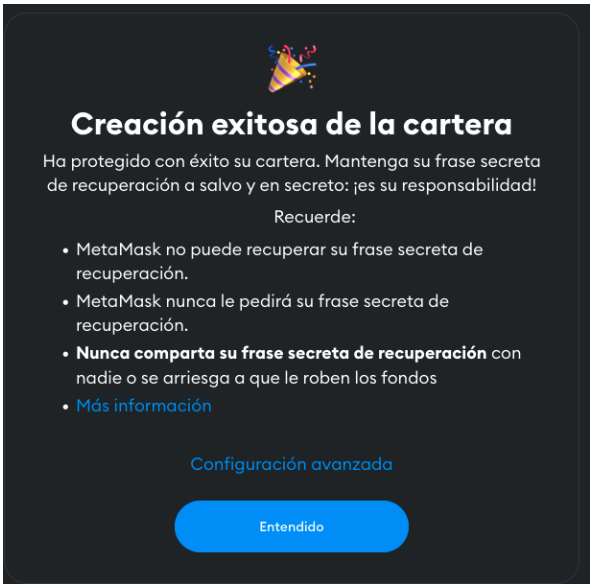


Fig. 2.13. Pantalla de creación de billetera con éxito

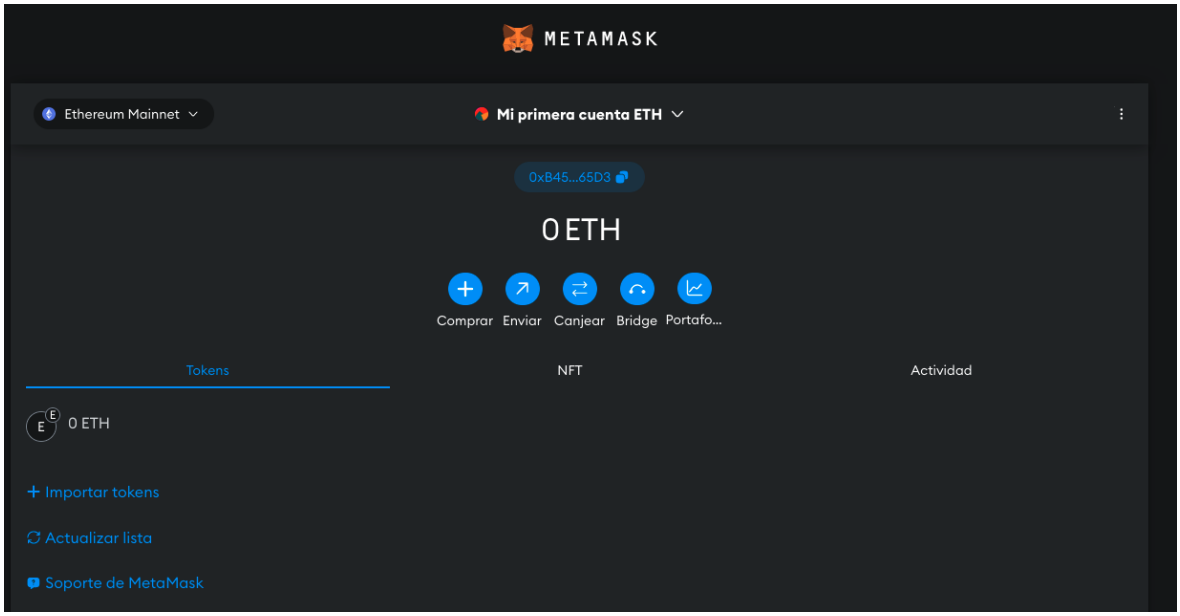


Fig. 2.14. Cuenta Etherum

2.2.4. De la centralización a la descentralización

Las aplicaciones que se usan actualmente se desarrollan en tres capas, interfaz de usuario o front-end, lógica de negocio o backend y almacenamiento de datos (bases de datos). La mayoría de aplicaciones almacenan sus datos en bases de datos centralizadas y replicadas en varias regiones del mundo utilizando los servicios de proveedores de nube pública [48] como Amazon Web Services, Microsoft Azure o Google Cloud.

Blockchain irrumpe en este paradigma [48] de desarrollo de aplicaciones permitiendo que cualquier persona o empresa participe del mantenimiento de la base de datos y ofreciendo un punto de acceso a la información. La blockchain por diseño ofrece **redundancia geográfica, alta disponibilidad** y mecanismos de consenso descentralizados para validar la información almacenada en la base de datos, se ha democratizado el dato. Las diferencias entre el modelo centralizado y descentralizado se representan en la Figura 2.15.

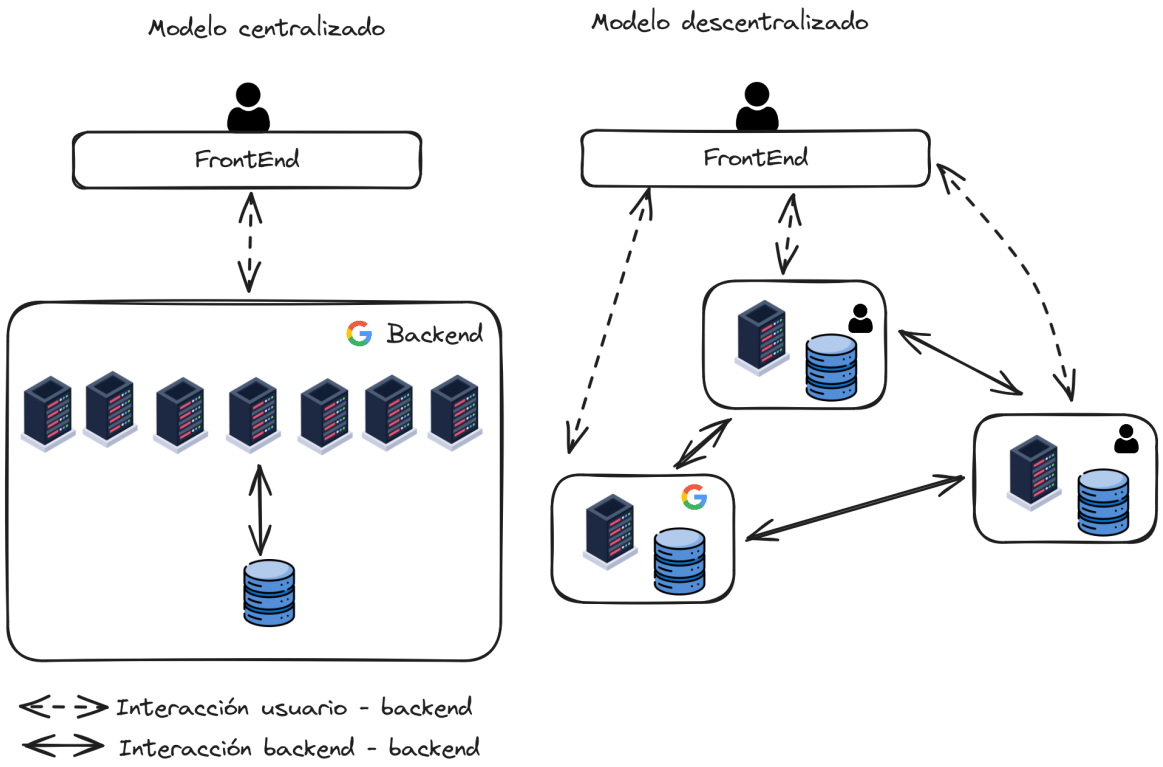


Fig. 2.15. Diferencias entre centralización y descentralización

2.2.5. Etherum y los smart contracts

Etherum fue creado por Vitalik Bulterin [49], un joven de 19 años, en 2014 para añadir una nueva funcionalidad a las redes de blockchain. Esta nueva característica, denominada contratos inteligentes, permitía crear un ordenador descentralizado capaz de ejecutar aplicaciones en la red.

Los contratos inteligentes [50] son las piezas de código, escrito en Solidity, que componen una aplicación descentralizada, por ejemplo, aplicaciones de compraventa, procesos legales, acuerdos entre particulares, videojuegos, sistemas de puntos y fidelización, etc.

La aparición de Etherum cambió radicalmente el mundo del blockchain, ya no sólo era una cadena de bloques que permitía transferir activos entre personas sino la capacidad para desplegar aplicaciones descentralizadas. A partir de este momento, Vitalik Bulterin ofrecío a la sociedad las características de inmutabilidad, confianza, sin intermediarios pero permitiendo nuevos casos de uso.

Etherum fue la segunda red de blockchain siendo la precursora de una oleada de nuevos proyectos.

Uno de los problemas que han tenido las redes Bitcoin y Etherum han sido la velocidad de generación de bloques y la comisiones de la red, estas dos debilidades son las que los nuevos proyectos buscan explotar y mejorar. De este forma surgieron otros proyectos y redes blockchain Cardano [51], Polygon [52] o Avalanche [53].

2.2.6. Mecanismos de consenso: Proof of work vs Proof of Stake

PoW es el mecanismo de consenso por el que Bitcoin y Etherum añaden nuevos bloques a la red, dando validez a las transacciones incluidas en ese bloque .Este mecanismo no es eficiente ni sostenible porque a medida que la red incorpora nuevos nodos la dificultad del cálculo se incrementa, Bitcoin incrementa la dificultad cada 4 años (halving), en cada halving los recursos de cómputo y energía necesarios para que un minero logre resolver el problema se incrementan.

En este mecanismo de consenso el minero pone como garantía de su legitimidad el coste del equipamiento informático y el coste energético que implica añadir bloques.

PoS es el mecanismo que utilizan la mayoría de redes de nueva generación, como Cardano [51], Polygon [52] o Avalanche [53], para aprobar y añadir nuevos bloques a la red. PoS no necesita de "mineros"que dispongan de grandes recursos informáticos para resolver un problema matemático, en estas redes, los validadores deberán depositar como garantía en la red fondos monetarios o el token de la red (criptomoneda).

El algoritmo de una red PoS elige de entre todos los validadores de forma pseudo-aleatoria, atendiendo al número de monedas bloqueadas, para validar y añadir el nuevo bloque de la red. Como recompensa por su trabajo el validador es recompensado con el token de la red pero si el nodo validador intenta añadir o corromper la cadena de bloques es sancionando y todo los fondos depositados son retirados por la red.

La tecnología PoS es la más utilizada actualmente quedando muy pocas redes que mantienen el sistem PoW. Etherum en 2023 consiguió finalizar el cambio de un modelo PoW a PoS mediante un hard fork que bifurcó la cadena de bloques en Etherum Classic (PoW) y

Etherum 2.0 (PoS), siendo esta última la cadena de bloques oficial.

2.2.7. Non-Fungible Token NFT

2.2.7.1. Estándar ERC721

El estándar ERC-721 define la norma que debe cumplir el contrato inteligente para que pueda usarse como NFT y pueda ser minteado. Un token no fungible (NFT) es un activo digital, generalmente un fichero, que representa de forma única un elemento u objeto del mundo real, por ejemplo, décimos de lotería, entradas de eventos, colecciones, arte digital o títulos académicos. Los NFT permite asociar una imagen a la identidad de una persona y que pueda mostrarse visualmente en una aplicación, como la wallet de una red blockchain, para validar la tenencia de ese NFT por la esa persona.

La estructura básica de un contrato inteligente, según el estándar ERC-721, debe contener la variable tokenId y baseURI, como se observa en Código 2.3. La primera variable debe ser de tipo uint256 y su valor deberá ser el identificador de la imagen alojada en un servicio de almacenamiento, como Pinata [54], Onedrive [55] o Amazon S3 [56]; por otro lado, la variable baseURI deberá ser de tipo string y contendrá la dirección URL de la web donde estén almacenados todas los ficheros (NFT) del contrato inteligente.

2.2.7.2. Alojamiento de archivos

El alojamiento de los ficheros NFT, que representan el activo real, debe tener la condición de garantizar la inmutabilidad e integridad del fichero; sin esta garantía el concepto de NFT pierde su utilidad porque si el almacenamiento donde se guarde el fichero permite actualizarlo sin modificar el identificador (URL) el contrato inteligente ERC-721 apuntará a un tokenID correcto pero su contenido ha sido modificado.

Para comprender este problema más fácil se puede plantear el caso de uso de las entradas de un evento donde el acceso dependa de la fecha y la hora impresas en la entrada, minteada en una blockchain, en la entrada del evento el personal de seguridad solicita a los cliente el acceso mediante sus billeteras blockchain, todas aquellas persona que puedan mostrar en sus billeteras la imagen del NFT con la fecha y hora de ese evento pueden acceder. Si el desarrollador de este modelo de acceso en blockchain guarda los NFTs en un almacenamiento no seguro son controles de integridad e inmutabilidad se podría alterar la imagen, cambiando la fecha y la hora, pudiendo acceder en cualquier momento al evento.

Este problema ha llevado a los desarrolladores de blockchain a diseñar el protocolo IPFS [57] que permite almacenar ficheros garantizando la inmutabilidad e integridad del contenido y de forma descentralizada en una red 'peer to peer'. Cuando un fichero se sube a la red de almacenamiento se calcula, en base a su contenido, un identificador único denominado CIDS, si un usuario desea actualizar el fichero deberá subir un nuevo fichero y se generará

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.19;

import "@openzeppelin/contracts/token/ERC721/ERC721.sol";

contract NFT is ERC721 {
    uint256 private _nextTokenId;

    constructor()
        ERC721("MyNFT", "NFTTOKENID")
    {}

    function _baseURI() internal pure override returns (string memory) {
        return "https://jade-negative-emu-951.mypinata.cloud/ipfs/
        QmZ4BdPhm5FKcZpWTd6H2Sf6MYhahaDKG6xqVrteRj8LWy/";
    }

    function safeMint(address to, uint256 tokenId) public {
        _safeMint(to, tokenId);
    }
}
```

Código 2.3. Ejemplo de restricción de una función con un modificador

un nuevo CID asociado, ahora la red tendrá ambas versiones y todos los contratos ERC-721 cuyo tokenID apuntaba al primer CID seguirán mostrando el mismo contenido dado que el fichero no se ha modificado.

IPFs es un protocolo libre que puede desplegarse en entornos locales, privados, empresariales o público, como es el caso de Pinata [54], una plataforma que ofrece almacenamiento en la nube basado en el protocolo IPFS con compatibilidad API Rest para subir y visualizar los ficheros alojados. La principal desventaja de utilizar la red pública de Pinata es que los ficheros son accesibles por lo nodos que los alojan, sin embargo, Pinata ofrece la posibilidad de crear redes privadas donde sólo aquellos nodos autorizados puedan intercambiar y alojar los ficheros.

2.2.8. Análisis de plataformas existentes

2.2.8.1. Cardano

Cardano es la red creada en 2017 [58] por el cofundador de Etherum Charles Hoskinson con la idea de solucionar tres problemas relacionados con sostenibilidad, escalabilidad e interoperabilidad. La tecnología de Cardano ofrece la capacidad de crear cadenas de bloques laterales permitiendo interactuar entre ellas a través de la cadena principal, también, permite crear token personalizados por lo que sería posible tokenizar activos fungibles y no fungibles.

2.2.8.2. Polygon

Polygon Matic [59] es uno de los proyectos con mayor soporte por parte de partners y desarrolladores, se dice que Polygon es la solución a los problemas de Etherum. Los inicios de Polygon fueron como una red lateral de Etherum pero su desarrollo ha ido evolucionando tán rápido que actualmente es una plataforma completa con multitud de servicios.

Polygon es compatible con las aplicaciones descentralizadas(dApps) de Etherum ofreciendo la interoperabilidad entre redes laterales independientes por lo que el procesado de transacciones es superior al ofrecido por Etherum.

Polygon no sólo ofrece una red blockchain basada en EVM y PoS sino que ofrece otros productos como redes zkEVM, supernets (redes blockchain privadas) y Polygon ID (Identidades descentralizadas).

2.2.8.3. Avalanche

Avalanche Network implementa un mecanismo de consenso PoS basado en el algoritmo Snowball [60] y, a diferencia de Polygon, desde sus inicios el proyecto ha dividido la red principal en tres subredes C-Chain, X-Chain y P-Chain, como se muestra en la Figura 2.16

C-Chain: [61] Esta red se utiliza para construir y desplegar aplicaciones descentralizadas (dApps). Avalanche permite el desarrollo de aplicaciones en múltiples lenguajes, incluido Solidity e implementa EVM por lo que es compatible con cualquier aplicación de Etherum.

X-Chain: [61] Esta red se utiliza para operar con activos inteligentes. Un activo inteligente puede ser un token o la representación de un activo real como acciones, inmuebles, propiedades, etc.

P-Chain: [61] Esta red se utiliza para crear nuevas redes blockchain personalizadas y permisionadas. Estas redes pueden definir algunos parámetros como las comisiones, los nodos validadores permitidos, el algoritmo de consenso y control de acceso.

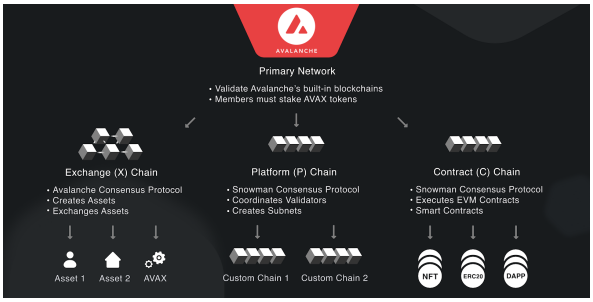


Fig. 2.16. Avalanche networks

2.2.9. Identidad descentralizada

La tecnología blockchain utiliza algoritmos criptográficos, como se explicó en Subsección 2.2.3.2, para generar las cuentas de los usuarios o dirección de red con las que los usuarios podrán identificarse y realizar acciones, como el intercambio de activos o la ejecución de contratos inteligentes.

En un país es imprescindible disponer de una identidad que identifique a cada uno de los ciudadanos permitiéndoles realizar acciones con las administraciones y otras entidades. La interacción con estas entidades genera una red de conexiones que completa y representa su identidad. Ejemplos de estas conexiones pueden ser el histórico de registros médicos (vacunas, enfermedades, medicamentos, tratamientos, etc), información académica (entidades educativas, registro de evaluaciones, certificaciones, títulos oficiales, etc), registro catastral y de propiedades.

Cada administración desarrolla sus propios sistemas y soluciones con mecanismo de autenticación y credenciales propias. La forma de realizar los trámites y de autenticarse son muy diversas, desde asistir de forma presencial a una oficina hasta usar el certificado digital para validar tu identidad.

Los certificados digitales [38] son ficheros digitales que contienen la clave pública y están firmados por una entidad certificadora, en España esta entidad es la Real Casa de Moneda y timbre [62], utilizados para identificarse y validar la identidad del usuario que esta realizando

el trámite.

En un entorno blockchain el funcionamiento sería similar al certificado digital, de hecho, las claves podrían ser guardadas en un fichero protegido similar al certificado digital para no requerir que los usuarios tengan que recordarlas y su uso sea rápido y sencillo.

A diferencia de los sistemas actuales, accesibles mediante certificado digital, radica en que en un sistema blockchain el desarrollo de los servicios es común, el mantenimiento y los costes se simplifican, además, la información almacenada en la cadena de bloques estará disponible para todos los partícipes, protegida por sistemas de roles RBAC, garantizando la confidencialidad, inmutabilidad, trazabilidad y confianza.

El uso de identidades descentralizadas está en desarrollo y existen distintas soluciones en el mercado en fases poco maduras de desarrollo como Etherum, Dock o Microsoft.

2.2.9.1. Etherum DID

La fundación Etherum esta desarrollando soluciones relacionadas con identidades descentralizadas, ha creado varios proyectos ambiciosos Etherum Name Service (ENS), Etherum Attestation Service (EAS), Proof of humanity (POH) or Proof of personhood Passport.

Etherum Name Service

Es un servicio abierto y distribuido en la red blockchain de Etherum que ofrece los servicios típicos de DNS pero en la Web3. Este servicio permite asociar nombres de dominio con direcciones de billeteras de múltiples cadenas de bloques, direcciones webs descentralizadas, hashes o metadata de las redes blockchain.

El registro ENS consiste en un smart contract desplegado en la red de Etherum que mantiene un mapa de todos los dominios activos con tres datos (propietario del dominio, entidad encargada de resolver el dominio y el TTL de los registros).

La resolución de nombres se divide en dos etapas, como se observa el diagrama de la Figura 2.17, la primera etapa en la que se consulta en el registro ENS la dirección de red blockchain de la entidad encargada de resolver el dominio (resolver), el resolver es un contrato inteligente. La segunda etapa el usuario consulta al resolver la dirección de red del dominio, accesible desde la dirección blockchain obtenida en la etapa anterior.

Etherum Attestation Service (EAS)

Es un servicio para garantizar el no repudio de un testimonio o una declaración. Un testimonio (attestation) es un registro criptográfico que contiene información sobre un determinado asunto firmado; la firma se realiza con la clave privada del usuario garantizando el no repudio. [64]

Algunos ejemplos de testimonio puede ser el nivel financiero de una entidad, la nacionalidad, votos, cumplimientos normativos, acreditaciones, certificados, etc.

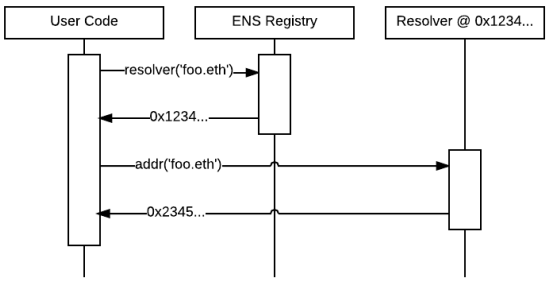


Fig. 2.17. Flujo resolución de nombre en ENS [63]

En la red de Etherum los testimonios pueden ser on-chain cuando toda la información se almacena en la blockchain u off-chain cuando la blockchain solo almacena un enlace o un hash de la información.

Las ventajas de utilizar el mecanismo off-chain son reducir la cantidad de datos almacenados en la red Etherum para reducir el coste de las comisiones de red. Almacenar la información fuera de la red también permite tener mayor control y privacidad, los usuarios tan sólo deben conocer que ese testimonio ha sucedido y ha sido validado por la red. [65]

El servicio EAS permite que una identidad quede asociada con un testimonio por lo que a medida que el usuario interactúa con los servicios de la red la identidad va generando nuevos vínculos que podrán ser validados por terceras entidades. [66]

Proof of Humanity (POH)

Es un sistema social de verificación de identidades de personas en la red Ethereum. Combina redes de confianza, test de Turing inversos y resolución de disputas para crear una lista de personas a prueba de ataques Sybil. [67]

Un ataque sybil es un tipo de ataque en el que la reputación de un sistema o servicio es alterado con identidades ficticias para conseguir más influencia.

La plataforma proof-of-humanity.id permite crear un un perfil usando una billetera Etherum, a partir de este momento tu perfil deberá recibir la aprobación por otros participantes de la red para incrementar tu reputación.

Sistemas como el propone proof of humanity permitiría crear un sistema de reputación de los individuos en función del cual un individuo podría recibir ayudas econónimcas, accesos a un trabajo de seguridad, acceso a créditos financieros, etc.

En la actualidad ya se pueden encontrar sistemas de reputación, por ejemplo, Linkedin permite que otros profesionales puedan valorar las capacidades o conocimientos de otro profesional.

2.2.9.2. Dock DID

Los colaboradores de Dock [68] estan desarrollando una plataforma blockchain llave en mano que facilita la implementación de servicios basados en DID. Dock ofrece servicios como Instant Veritification o Paswordless login.

Instant Verfication es un servicio que permite validar y verificar datos personales, titulaciones, sanciones o propiedades de forma privada sin necesidad de publicar el dato, simplemente se muestra que dicha información es cierta (Zero Knowledge Proofs), además, cuando una entidad desea solicitar la validez de la información, Dock propociona un mecanismo de cerrojos que limita la información que la entidad puede consultar. En Figura 2.18 se muestra la funcionalidad de cerrojo de la identidad descentralizada.

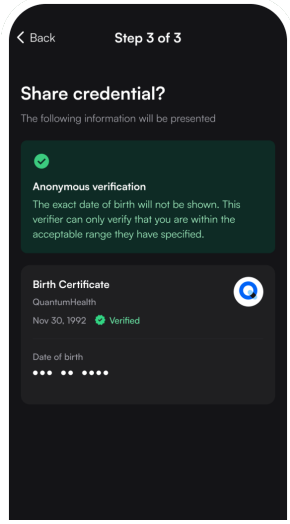


Fig. 2.18. Cerrojos de la DID en la plataforma Dock

Passwordless Login es un servicio de autenticación delegada para plataformas externas, como se muestra en Figura 2.19, similar al servicio de autenticación con Google. Este tipo de servicio es muy utilizado para acceder a los servicios de otras plataformas reutilizando la identidad de una entidad de confianza como Google, en este caso, se utiliza la tecnología Web3 y blockchain para proveer esta confianza.

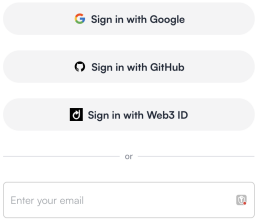


Fig. 2.19. Autenticación delegada usando Dock

2.2.9.3. Microsoft DID

En [69], Microsoft analiza los cambios en la sociedad sobre el uso de las identidades, expone la necesidad de utilizar mecanismos de identidad únicos de manera global en todos los servicios. Microsoft identifica que el futuro de las plataformas y la identidad de los usuario en internet tiene que cambiar a un modelo donde el usuario pueda controlar los datos que se almacenan en estas plataformas y la necesidad de crear un mecanismo universal para identificarnos a nivel global, como puede ser la licencia de conducir o el numero de identidad.

Microsoft explica que una persona tiene tantas identidades como entidades con las que interactúa, debe tener al menos un correo electrónico que luego usará para crear IDs de usuario en cada una de las plataformas sociales, comercios y cualquier otro tipo de servicios con los que interactúa, esto supone que los usuarios tienen que duplicar sus credenciales en cada una de las plataformas. La tecnología blockchain, como se verá a continuación, permite generar confianza y reutilizar la identidad única del usuario en cualquier entidad.

Es necesaria la investigación de nuevos mecanismos para generar las claves publico-privadas si se desea adoptar el blockchain como mecanismo para interactuar en la sociedad. Asumiendo este contexto se podrían diseñar mecanismos para utilizar la biometría para calcular y acceder a los servicios de la blockchain.

En [70], se explican distintos mecanismos para generar claves RSA a partir de las huellas dactilares de una persona u otro elemento biométrico. En el caso de las huellas se explica que las diferencias de cada individuo genera un string binario que tras un proceso de corrección de error, codificación usando el algoritmo Reed-Salomon y haciendo un Hash-SHA256 se obtiene los elementos de entrada para generar dos parejas de claves únicas.

En [70], se explica el procedimiento basado en tres algoritmos para calcular el seed, este procedimiento consiste en generar dos hashes SHA-256, el primero calculado a partir de la huella biométrica y el segundo a partir de una contraseña, los hashes resultantes se utilizan para calcula el seed aplicando un función 'bit-wise OR' entre los dos. Una vez se ha calculado el seed, se utiliza para generar la pareja de claves, a diferencia de como se explica en 2.2.3.3.

2.3. Situación actual del blockchain, países pioneros

En Malta se ha creado un grupo de trabajo para construir una solución blockchain para registrar los certificados académicos, con este sistema los estudiantes y las instituciones académicas podrán beneficiarse de tener todos los certificados académicos accesibles con garantías de su legitimidad. [71]

En España se ha constituido el proyecto Alastria [72], impulsado por empresas de banca, energía y telecomunicaciones. Entre las más destacadas se encuentran nombres [73] como Telefónica, Banco Santander, CaixaBank, BBVA, Repsol, Cepsa, Indra, Barceló Viajes

(Ávoris) y BME.

Su objetivo es desarrollar esta tecnología de registros contables compartidos para acelerar la transformación digital. Además, el equipo de Alastria cuenta con notarios y abogados que velarán por la seguridad y veracidad de la información a través de la identificación de las personas físicas y jurídicas. El objetivo es diseñar un sistema de identidad descentralizada en Alastria que permitirá a los ciudadanos tener el control sobre su información personal.

Otros países como **Japón, Suecia o Reino Unido** están investigando la tecnología blockchain para agilizar los procesos de registro de propiedades.[71]

2.4. Ventajas e inconvenientes del blockchain

2.4.1. Consumo energético

Como se menciona en la Sección 2.2, los algoritmos de prueba de trabajo (PoW) consume importantes recursos de computación y energía para calcular la solución del algoritmo y añadir el bloque en la cadena de bloques. Se estima que la minería de Bitcoin requiere anualmente 121 TWh y esta cifrá se incrementará a medida que la dificultad del algoritmo se incremente en cada halving.

En este aspecto las compañías de computación en la nube, que gestionan grandes centros de datos por todo el mundo, están desarrollando programas para utilizar energía renovable que alimente los centros de datos y los sistemas de refrigeración.

Blockchain está intentado reducir su huella de carbono mediante algoritmos de prueba de participación (PoS) donde los nodos de la red no compiten por resolver lo antes posible un acertijo criptográfico, simplemente, bloquean la moneda de la red como garantía y compromiso de que es un nodo legítimo para validar transacciones y crear nuevos bloques.

Las cadenas de bloques que utilizan este mecanismo de consenso no requieren de grandes recursos de computación ni consumo energético, por lo que, cualquier persona puede utilizar un ordenador antiguo y colaborar en la descentralización y correcto funcionamiento de la red.

2.4.2. Almacenamiento de datos

Las cadenas de bloques permiten almacenar información pero el coste de almacenar grandes cantidades de datos es muy elevado, no han sido diseñadas para este uso.

Sin embargo, si entendemos Blockchain como lo que es: una base de datos descentralizada, entonces podremos imaginarnos un sistema que utilice esta tecnología para almacenar punteros o direcciones que recojan la ubicación real de los datos, almacenada físicamente en servidores distribuidos geográficamente y conocidos como 'nodos'. Añadiéndole un sistema

informático capaz cifrar, segmentar, redundar y distribuir la información hacia estos miles de nodos de almacenamiento, obtenemos un soporte extraordinariamente resiliente.

La solución a este problema puede ser IPFS (InterPlanetary File Systems) [57] un protocolo y sistema de archivos distribuido. Este mecanismo de archivos ofrece mejoras con respecto a los sistemas de almacenamiento centralizados actuales, como velocidades de descarga más rápidas, reducción de los costes de ancho de banda y el riesgo de pérdida de datos o censura.

Los sistemas de almacenamiento actuales solucionan este problema con mecanismo de replicación y direccionamiento DNS para ubicar geográficamente el servidor que tiene el fichero. Además, para mejorar la latencia y la velocidad de entrega de contenido se utilizan redes de distribución de contenido (Content Delivery Network, CDN).

IPFS segmenta los ficheros en partes más pequeñas (blocks), cada bloque se identifica con hash único (Content Identifier, CID). IPFS determina la ubicación física de los bloques con distintos algoritmos de rutas como Kademlia Distributed Hash Table (DHT), Bitswap, mDNS o Delegated routing over HTTP.

Los múltiples CIDs que constituyen un fichero se relacionan a través del mecanismo Interplanetary Linked Data (ILP). IPFS permite crear jerarquía de ficheros y carpetas por lo que IPFS puede usarse como sistema de ficheros, no sólo como almacenamiento no estructurado.

Una vez que los datos se agregan al IPFS, no se pueden cambiar, se vuelven inmutables, lo que hace que sea un sistema de ficheros óptimo para usarlo en soluciones blockchain. En IPFS actualizar un fichero implica crear una nueva versión, similar a los sistemas de control de versiones como Git, cada versión tiene un enlace a la versión anterior, por lo que, siempre IPFS conoce la versión vigente pero también permite conocer las versiones anteriores.

3. AVALANCHE NETWORK

En este capítulo se analizará en profundidad la tecnología de subredes de Avalanche, desarrollada por el equipo de Ava labs, Exchange Chain (X-Chain), Contract Chain (C-Chain) y Platform Chain (P-Chain).

3.1. Sobre avalanche

Avalanche es un proyecto blockchain que busca mejorar las deficiencias de otras redes blockchain, para ello ha dividido su red principal en tres subredes diferentes X-Chain, C-Chain y P-Chain para distintos casos de uso. La subred X-Chain se utiliza para la creación de nuevos activos digitales, mientras que la C-Chain implementa la Máquina Virtual Ethereum (EVM) de Avalanche para desplegar contratos inteligentes y crear aplicaciones descentralizadas basadas en Solidity, por último, la P-Chain enfocada a funcionalidades de gestión como coordinar los validadores, gestionar las recompensas y crear otras subredes personalizadas.

Avalanche ha desarrollado su propio algoritmo de consenso, Avalanche DAG, con el objetivo de ofrecer una red blockchain con una alta capacidad en el procesado de transacciones pero manteniendo la seguridad y la integridad de los bloques, sin embargo, las subredes C-Chain y P-Chain no utilizan Avalanche DAG, implementan una modificación de Avalanche DAG, denominada Snowman Consensus Protocol, para garantizar que las transacciones se añaden a los bloques por orden secuencial.Según Ava Labs, la plataforma puede gestionar unas 4.500 transacciones por segundo, frente a las 7 tx/seg de Bitcoin y las 14 tx/seg de Ethereum.

Una de las ventajas que ofrece Avalanche es la flexibilidad de su plataforma para crear nuevas subredes personalizadas pudiendo configurar cualquier parámetro como los costes de gas, aplicar mecanismos de control de acceso y lista de nodos validadores permitidos o denegados, incluso, permite utilizar algoritmos de consenso personalizados.

3.2. Mecanismo de consenso

El consenso permite a la red blockchain determinar el estado válido de la cadena de bloques entre todos los nodos validadores que participan en la red. Estos validadores serán los encargados de llegar a un acuerdo sobre las transacciones y bloques a añadir en la red. El algoritmo de consenso es uno de los puntos críticos en una red blockchain porque determina el rendimiento y seguridad de la red.

Avalanche ha desarrollado su propio algoritmo de consenso, Avalanche Directed Acyclic

Graph (DAG), basado en una estructura de datos de grafo acíclico dirigido. Los grafos acíclicos dirigidos se caracterizan en que cada vértice del grafo no puede tener un camino que empiece y acabe en él (evita los ciclos). [60]. junto con el protocolo Snowball que utiliza un mecanismo de muestreo iterativo entre todos los nodos para encontrar un punto común como resultado de consenso válida.

El algoritmo de Avalanche utiliza una conjunto de parámetros configurables en cada subred atendiendo a cada uno de los requisitos y casos de uso de los proyectos que utilicen la tecnología blockchain de Avalanche.

- Numero de validadores (p)
- Tamaño de muestras (k): Numero de validadores a los que preguntar por su resultado
- Cantidad de nodos que llegan a la misma solución (Alfa)
- Decisión Threshold (Beta) : El número de veces que se consigue el mismo resultado en el consenso de los nodos

Cuando se añaden nuevas transacciones en la red los validadores empiezan a realizar cálculos para crear el nuevo bloque y así determinar el estado de la cadena. Cuando un validador genera un bloque deberá preguntar a un número (k) de validadores si han llegado a la misma solución, si el número de nodos que llegan a la misma solución es mayor o igual que (alfa), se sigue iterando (snowball) hasta que ese consenso se repite al menos (beta). Cuando alfa validadores repiten beta veces la misma preferencia se añade el nuevo bloque y se informa al resto de validadores del nuevo estado de la cadena.

3.3. C-Chain

La subred C-Chain implementa la máquina virtual Etherum (EVM) y se encarga de ejecutar los contratos inteligentes escritos en Solidity. Para interactuar con la subred C es necesario establecer una conexión con uno de los servidores (nodos) que proveen la API mediante el protocolo JSON RPC o Websocket, estos nodos pueden ser los oficiales de Avalanche aunque existe la posibilidad de utilizar otros nodos privativos o personalizados por el usuario, dependiendo del uso de la aplicación descentralizado puede ser interesante utilizar servidores dedicados.

El protocolo JSON RPC (remote procedure call) es un protocolo de comunicaciones cliente servidor, sobre el protocolo HTTP, el cliente envía una petición HTTP cuyo cuerpo es un JSON que contiene tres parámetros (method, id y params). En el Código 3.1 se muestra un ejemplo de una petición HTTP al servidor oficial C-Chain API de Avalanche, solicitando conocer el estado de la transacción (avm.getTxStatus) indicada en el campo params.

```
curl -X POST --data '{
    "jsonrpc":"2.0",
    "id"     :4,
    "method" :"avm.getTxStatus",
    "params" :{
        "txID":"2QouvFWUbjuySRxeX5xMbNCuAaKWfbk5FeEa2JmoF85RKLk2dD"
    }
}' -H 'content-type:application/json;' https://api.avax.network:9650/ext/bc/C
```

Código 3.1. Petición HTTP JSON RPC

3.4. Máquina virtual

La máquina virtual es la base de las redes blockchain donde se ejecutan los contratos inteligentes, es la encargada de definir la lógica de la aplicación de la cadena de bloques, mantener el estado de la blockchain, gestionar la transición de los estados, ejecutar las transacciones y proporcionar mecanismos para interactuar con la red (API). Una máquina virtual puede ser utilizada por múltiples cadenas de bloques y una cadena de bloques se ejecuta sobe una única maquina virtual. En Avalanche un validador ejecuta tres máquinas virtuales:

- CoreETH: Define la C-Chain y es la encargada de ejecutar los contratos inteligentes y es compatible con la máquina virtual ETH, que ejecuta los contratos escritos en Solidity.
- Platform VM: Define la P-Chain y es la encargada de procesar las operaciones de soporte y el staking
- Avalanche VM: Define la X-Chain y es la encargada de procesar las operaciones de los tokens nativos de Avalanche

3.5. Redes personalizadas

Una subred es una cadena de bloques que define sus propias reglas y políticas, aplica controles de acceso, define el plan económico del token nativo y establece el mecanismo de consenso para validar las transacciones. Todas estas configuraciones requieren crear una nueva maquina virtual y disponer de un determinado número de validadores que ejecuten la máquina virtual, procesen y mantengan la red.

Como se ha explicado en la Sección 3.1 la red primaria de Avalanche está compuesta por tres subredes C-Chain, X-Chain y P-Chain, cada una ejecutando su propia maquina virtual independiente. Avalanche ofrece a cualquier usuario la capacidad para crear su propia red

basada en las VM oficiales de Avalanche o crear su VM personalizada, hay que tener en cuenta que no todos los proyectos requieren una maquina virtual personalizada.

Una subred con su propia máquina virtual es útil para proyectos donde se desean aplicar mecanismos de control de acceso y privacidad de la información para que sólo aquellos validadores aceptados tengan la capacidad para procesar transacciones y exponer la red a través de una API a terceras partes. Otro posible caso de uso de subredes son aplicativos que deban cumplir con ciertas normativas como la protección de datos y la ubicación geográfica de la información o normativas de blanqueo de capitales y sistemas de validación de identidad (KYC).

4. ENTORNO DE DESARROLLO

En este capítulo se explicará el entorno de desarrollo y pruebas de los contratos inteligentes de la aplicación descentralizada. Como se ha explicado en el capitulo anterior la red C-Chain de Avalanche es donde se despliegan las aplicaciones descentralizadas, los nodos que conforman esta red ejecutan las máquinas virtuales Etherum donde se ejecutan los contratos inteligentes programados en Solidity [74].

4.1. Máquina virtual Etherum (EVM)

Una aplicación requiere de almacenamiento, memoria volátil y el binario de la aplicación, la maquina virtual Etherum, como se muestra en Figura 4.1, incluye estos tres elementos máquina de estado, Virtual ROM, y almacenamiento, respectivamente.

- Virtual ROM es una zona de memoria no volátil donde se almacena el binario de la aplicación (bytecode), inmutable una vez ha sido desplegado en en la red y creada la maquina virtual de la aplicación descentralizada.
- La máquina de estados es la memoria volátil que contiene el contador del programa, las tarifas de gas de la aplicación y los dos bloques de memoria de la aplicación para almacenar variables stack y memory.
- La zona de almacenamiento de datos no volátil (storage) es donde se guarda la información de las cuentas de los usuario como los balances u otros datos relativos a las aplicaciones descentralizadas, la estructura de datos de esta memoria es en mapas clave - valor de 256 bits.

El flujo de ejecución de un contrato inteligente, que se muestra en la Figura 4.2, comienza con una llamada a una de las funciones del contrato y el contador de programa se posiciona

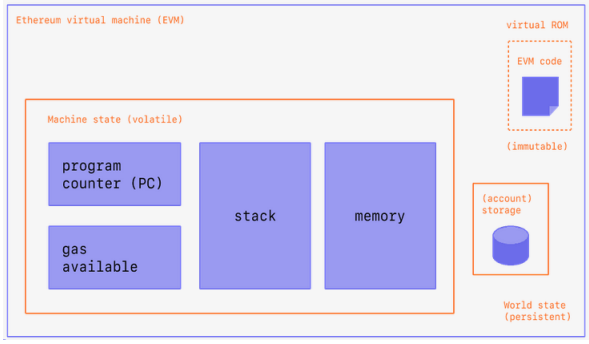


Fig. 4.1. Etherum Virtual Machine [75]

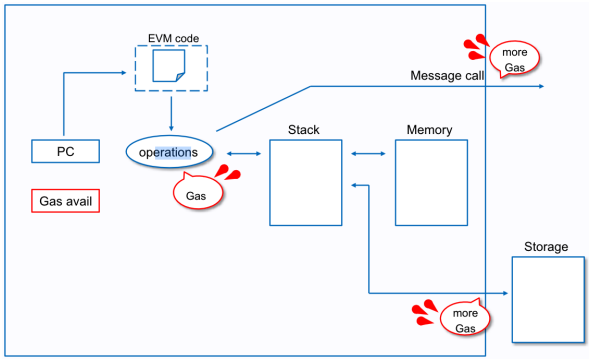


Fig. 4.2. Diagrama de flujo de ejecución de un programa en EVM [76]

en la primera instrucción de la función que guardará los datos temporales en stack o en memoria y los datos no volátiles en la zona de almacenamientos de datos. El coste de gas asociado a la ejecución de estas funciones dependerá del número de instrucciones que se ejecuten y del la cantidad de datos almacenados en memoria.

4.2. Herramientas de trabajo (Toolchains)

Avalanche proporciona tres entornos con los que desarrollar y probar los contratos inteligentes Hardhat, Truffle y Foundry.

4.2.1. Truffle

Truffle es el entorno por excelencia para el desarrollo de aplicaciones descentralizadas en la red Etherum, su uso está muy extendido y proporciona funcionalidades para el desarrollo y despliegue de las aplicaciones en múltiples redes. Truffle requiere NodeJS instalado en la máquina de desarrollo para poder ejecutarse y habrá que configurar el entorno de Truffle crear una red blockchain para pruebas en local.

4.2.2. Foundry

Foundry es un entorno escrito en Rust, en contraposición con Truffle está menos extendido aunque proporciona las mismas herramientas que Truffle. También requiere NodeJS instalado en la máquina de desarrollo.

4.2.3. Hardhat

Hardhat es el entorno recomendado por la comunidad Avalanche, al igual que Truffle, ha sido creado para la red Etherum pero es totalmente compatible con Avalanche. En la maquina de desarrollo se tendrá que instalar NodeJS y el paquete yarn, encargado de ejecutar las tareas

de compilación, despliegue y pruebas. Hardhat ofrece un plugin para VSCode que ayuda con autocompletado de código, formato, funciones de Go To y detección de errores sin compilar.

4.3. AvalancheGo and Network Runner

Avalanche Network Runner es una herramienta tanto para desarrolladores como para integradores de sistemas, permitiendo ejecutar nodos Avalanche personalizados, subredes y probar código en local antes de desplegarlo en la red de pruebas Fuji o en la red oficial C-Chain.

En este proyecto el entorno de pruebas se desplegará en un contenedor Docker a partir de una plantilla Dockerfile sobre una imagen Linux de NodeJS.

4.3.1. Creación de la imagen Docker

Un fichero Dockerfile contiene todas las instrucciones necesarias para construir imágenes Linux desde cero, las principales instrucciones que se usan en este proyecto son FROM, RUN, COPY y WORKDIR. La instrucción FROM define el kernel base sobre el que se desea construir la imagen, RUN ejecuta comandos en el kernel, WORKDIR se utiliza para crear carpetas en el sistema de ficheros y mover el cursor a ese directorio, por último, COPY permite copiar archivos desde el sistema local a la imagen para que en tiempo de ejecución el contenedor pueda utilizarlos.

El Dockerfile que se muestra en el Código 4.1 crea una imagen base para desplegar un entorno de desarrollo Avalanche con una red de pruebas local de 5 nodos. Esta imagen se usará para crear un contenedor donde se compilarán y ejecutarán los contratos inteligentes de la cuidad inteligente. Cada uno de los cinco nodos correrá la maquina virtual Etherum de la aplicación descentralizada, ejecutarán el algoritmo de consenso y mantendrán el estado de la cadena de bloques.

```
FROM node:20
RUN apt update -y && apt install nano sudo systemctl -y
RUN useradd -G sudo -m avalanche
RUN echo avalanche:avalanche | chpasswd
WORKDIR /avalanche

# Avalanche CLI is a tool to simulate a network of 5 nodes.
# In this network we can create custom subnet for testing purpose
# After test in local network we can deploy
# the subnet to the Fuji testnet and connect our wallets to it

COPY avalanche-cli.sh .
RUN chown avalanche:avalanche *.sh && chmod 755 *.sh
RUN chown avalanche:avalanche /avalanche
USER avalanche
WORKDIR /home/avalanche
COPY .avalanche-cli.json .
WORKDIR /avalanche
RUN git clone https://github.com/ava-labs/avalanche-smart-contract-quickstart.git
RUN cd avalanche-smart-contract-quickstart && yarn
RUN ./avalanche-cli.sh
RUN echo "export PATH=~/bin:$PATH" >> /home/avalanche/.bashrc
RUN echo "alias deploy=\"avalanche subnet deploy spain\"" >> /home/avalanche/.bashrc
RUN echo "alias ava=\"avalanche\"" >> /home/avalanche/.bashrc
RUN echo "alias avan=\"avalanche network\"" >> /home/avalanche/.bashrc
RUN echo "alias avas=\"avalanche subnet\"" >> /home/avalanche/.bashrc
RUN echo "alias pbu=\"cd /home/avalanche/citizenchain/src/\"" >> /home/avalanche/.bashrc
RUN echo "alias deploy=\"yarn deploy\"" >> /home/avalanche/.bashrc
RUN echo "alias main=\"yarn main\"" >> /home/avalanche/.bashrc
RUN echo Yes | /home/avalanche/bin/avalanche subnet list
WORKDIR /home/avalanche
USER root
COPY genesis/sidecar.json /home/avalanche/.avalanche-cli/subnets/spain/sidecar.json
COPY genesis/spain.json /home/avalanche/.avalanche-cli/subnets/spain/genesis.json
RUN chown avalanche:avalanche -R /home/avalanche/

USER avalanche
RUN mkdir /home/avalanche/citizenchain && mkdir /home/avalanche/citizenchain/src
CMD bash
```

Código 4.1. Código Docker para la creación de imagen del entorno de pruebas

Los comandos Código 4.2 y Código 4.3 permiten crear la imagen Docker y comprobar que la imagen se ha añadido al repositorio local de imágenes Docker disponibles.

```
$ docker build . -t <IMAGE_NAME>
$ docker images | grep <IMAGE_NAME>
```

Código 4.2. Comando para crear la imagen del entorno de pruebas

```
$ docker build . -t tfm/citizenchain:1.0
$ docker images | grep tfm/citizenchain:1.0
```

Código 4.3. Comando para crear la imagen del entorno de pruebas

4.3.2. Creación del contenedor a partir de la imagen

Un contenedor Docker es una instancia en ejecución basada en la plantilla de la imagen usada como base. Los contenedores son similares a las máquinas virtuales, son entornos de ejecución y almacenamiento independientes, que se ejecutan en un mismo dispositivo físico pero se diferencia en que son más livianos porque reutilizan el núcleo Linux del equipo HOST. Al crear un contenedor hay que tener en cuenta otros aspectos como la conectividad de red o el almacenamiento permanente en disco, a continuación, se explicará el procedimiento para exponer puertos y carpetas compartidas con el equipo HOST.

Los comandos que se muestran en Código 4.4 y 4.5 crean un contenedor llamado citizenchain basado en la imagen tfm/citizenchain:1.0. El parámetro -p redirecciona los puertos TCP 9651 y 9650 desde el equipo HOST hacia el contenedor, donde está escuchando el servicio API Rest de la red blockchain de pruebas; el parámetro -v comparte el directorio 'TFM/Proyecto/src' del equipo HOST con el contenedor en la ruta interna ' /citizenchain/src', por último, el parámetro -d se utiliza para que el contenedor se ejecute en modo detach y no se conecte a la terminal del contenedor, una vez esté en ejecución.

4.3.3. Acceso y crear la red de pruebas

Si el contenedor está en ejecución se puede acceder a la terminal utilizando el comando 'docker exec', una vez se haya accedido a la terminal del contenedor se crea la red de 5

```
$ docker run -d --name <CONTAINER_NAME> -p 9650:9650 -p 9651:9651 -it
-v <DIRECTORY_HOST>:<CONTAINER_DESTINATION_DIRECTOY> <IMAGE_NAME>
```

Código 4.4. Comando crear el contenedor

```
$ docker run -d --name citizenchain -p 9650:9650 -p 9651:9651 -it
-v /..../TFM/Proyecto/src:~/citizenchain/src tfm/citizenchain:1.0
```

Código 4.5. Comando para crear el contenedor

```
$ docker exec -it <CONTAINER_NAME> bash
$ avan start
```

Código 4.6. Código Docker para la creación de imagen del entorno de pruebas

nodos ejecutando el comando 'avan start' (el comando avan es un alias que se define en Código 4.1), como se muestra en Código 4.6.

Si la red se ha creado correctamente, al ejecutar el comando 'avan status', se debería visualizar la información que se muestra en Código 4.7.

```
Network is Up. Network information:
=====================================================================
Healthy: true
Custom VMs healthy: true
Number of nodes: 5
Number of custom VMs: 0
===================================== Node information ==============
node2 has ID NodeID-MFrZFVCXPv5iCn6M9K6XduxGTYp891xXZ and endpoint http://127.0.0.1:9652
node3 has ID NodeID-NFBbbJ4qCmNaCzeW7sxErhvWqvEQMnYcN and endpoint http://127.0.0.1:9654
node4 has ID NodeID-GWPcbFJZFfZreETSoWjPimr846mXEKCtu and endpoint http://127.0.0.1:9656
node5 has ID NodeID-P7oB2McjBGgW2NXXWVYjV8JEDFoW9xDE5 and endpoint http://127.0.0.1:9658
node1 has ID NodeID-7Xhw2mDxuDS44j42TCB6U5579esbSt3Lg and endpoint http://127.0.0.1:9650
================================== Custom VM information =============
```

Código 4.7. Código Docker para la creación de imagen del entorno de pruebas

En este punto el contenedor ya tiene creada la red blockchain con 5 nodos, que exponen la red por el endpoint indicado, el siguiente paso será crear la subred de pruebas que se utilizará para desplegar los contratos inteligentes, similar a la red oficial C-Chain pero en el entorno local. El comando Figura 4.8 permite crear la subred mediante un asistente para configurar los parámetros de la subred, se pueden dejar por defecto todos los valores salvo ChainId, Network Name and Token ID.

Si todo ha ido bien el comando list debería mostrar la información indicada en Código 4.9

Al crear la subred se ha creado una carpeta que contiene los ficheros génesis y sidecar de la subred, estos fichero se pueden usar como plantilla para recrear la subred en otros nodos sin necesidad de ejecutar de nuevo el asistente, hay que tener en cuenta que al recrear la subred

```
$ avas create <NETWORK_NAME>
$ avas list
```

Código 4.8. Comando para crear la subred de contratos inteligentes

```
+--------+--------+---------+----------------------------------------------------+------------+
| SUBNET | CHAIN  | CHAINID |                        VMID                        |    TYPE    |
+--------+--------+---------+----------------------------------------------------+------------+
| spain1 | spain1 |    5555 | sqja5hH2Dv3mEtt9iqdkoShzEGhKbGdP1RZ1y6bTDuewWTjLs | Subnet-EVM |
+--------+--------+---------+----------------------------------------------------+------------+
```

Código 4.9. Comando con el listado de subredes

todo el histórico de la cadena de bloques se elimina. La carpeta donde se encuentra este directo es '/home/<username>/.avalanche-cli/subnets/<networkname>' y debería contener los ficheros que se muestran en Código 4.10.

```
-rw-r--r-- 1 avalanche avalanche 1440 Dec 17 13:43 genesis.json
-rw-r--r-- 1 avalanche avalanche  409 Dec 17 13:43 sidecar.json
```

Código 4.10. Comando con el listado de subredes

4.4. Creación del proyecto

Hardhat [77] es el framework de desarrollo de Etherum que contiene diferentes componentes para edición, compilación, pruebas y despliegue de los contratos inteligentes y las aplicaciones descentralizadas, gracias a este framework disponemos de una suite de desarrollo completa que cubre todo el proceso para el desarrollo de soluciones blockchain.

Está basado en javascript y hace uso del sistema de paquetes npx, que ejecuta los paquetes sin necesidad de instalarlos directamente en tu dispositivo. Para comenzar a usar hardhat lo primero que se debe hacer es crear la estructura del proyecto ejecutando los comandos Código 4.11.

```
$ npm install --save-dev hardhat
$ npx hardhat init
```

Código 4.11. Comando con crear un proyecto hardhat

Tras finalizar el proceso de creación del proyecto se habrá creado un directorio que contiene la estructura de directorios y ficheros que se muestra en el Código 4.12, de los cuales los más importantes son:

- Contracts: Contiene un contrato inteligente de ejemplo
- Scripts: Contiene ficheros javascript o typescript que se usan para compilar, desplegar y un main para interactuar con los contratos inteligentes
- hardhat.config.js: Fichero que contiene la configuración de las subredes sobre las que desplegar la DApp y las cuentas de pruebas
- package.json: Fichero que contiene la configuración del proyecto

Avalanche ha publicado en Github un proyecto básico [78] basado en Hardhat con la configuración, scripts y contratos de ejemplo desarrollados por el de Ava Labs para Avalanche, es recomendable descargar y copiar el contenido del repositorio en el contenedor. En este proyecto se ha optado por descargar el repositorio en el equipo Host y haciendo uso de los volúmenes de Docker, como se muestra en Código 4.5, compartirlo con el contenedor para poder modificar los ficheros y construir la aplicación descentralizada de la ciudad inteligente desde el entorno de desarrollo Visual Studio Code instalado en el equipo Host.

Para nuestro entorno de desarrollo se ha modificado el fichero hardhat.config.js añadiendo los bloques de código que se muestran en Código 4.13 y Código 4.14. Las cuentas que aparecen en testAccount han sido creadas a partir de 24 palabras generadas con la herramienta online BIP39 mnemonic [79] y los datos de la subred se corresponden con los definidos al crear la subnet, como se indica en Código 4.5 y Código 4.9.

```
drwxr-xr-x   1 avalanche avalanche   4096 Dec 17 16:36 .
drwxr-xr-x   1 avalanche avalanche   4096 Dec 17 12:53 ..
-rw-r--r--   1 avalanche avalanche    352 Dec 17 16:36 README.md
drwxr-xr-x   2 avalanche avalanche   4096 Dec 17 16:36 contracts
-rw-r--r--   1 avalanche avalanche    146 Dec 17 16:36 hardhat.config.js
drwxr-xr-x 381 avalanche avalanche  12288 Dec 17 16:36 node_modules
-rw-r--r--   1 avalanche avalanche 270400 Dec 17 16:36 package-lock.json
-rw-r--r--   1 avalanche avalanche    106 Dec 17 16:36 package.json
drwxr-xr-x   2 avalanche avalanche   4096 Dec 17 16:36 scripts
drwxr-xr-x   2 avalanche avalanche   4096 Dec 17 16:36 test
```

Código 4.12. Comando con crear un proyecto hardhat

```
export const testAccounts = [
  "0x56289e99c94b6912bfc12adc093c9b51124f0dc54ac7a766b2bc5ccf558d8027", //SPAIN
  "0x74ec9b3f18c6d42703eaa339b9abf89e1a6006a9a42d75e2b27e31d620ccc598", // Director
  "0x61779b3f52f56378133a2c98a7142b34cae9312efd9283be62d9e84805ae59ea", // Citizen1
  "0x73425173342a1124e99ca23a9ef57641f95d074a9d1fe63964839564f9d3f294", // Funcionario

  "0x1bc06678006b33cf10d7f4a1631a0e03dd7f05e5c1348f884bb0b3af8f3b70c4", // Doctor1
  "0xe8f87a1b739da2e11b02423375193eca2f138d82a931f853304deb181017d0e4", // Pharmacist1
  "0xcad40a4fef10d0dd49ebb54cf25efa2938832918fbef9aea018a80d302814fdf", // Patient1
  "0xa86a48435e77d1ecdba60437dac0a70b3111d243a0f2ba91ef48b92fe3841b54", // Student1
  "0xd73f2e545cc78a9e36c8d5d0385d844b9c6b862b60db479ade4845d8d9552c8e", // Student2

  "0xe5c0cae23babdbe8571f51645f5296e41f3f4658012a34e81871f6cae8fd33e3", // Owner Bayer
  "0xc376880e3bf3cff02fefbcdb3376458f60988cf78c9c3f2cbd93369733486831", // Owner Pfizer
  "0x376966873b777648bf3b41e6053b105118c0dee10427eb5b556f30a16ab1f2f6", // Owner Grifols
  "0xad9358fd315d707e48f2a8e2982fd298de0e5d85ff1b0679a8728b79e95ae974", // Bayer
  "0x514b1e0556dd49738f50694689c3c2065e3e0b31528aed1a100228642eaca6fc", // Pfizer
  "0x10a6c5a0054663a1cc7b1d52233c4111cf225cc0b49ecfcd8fbb00c8e71049df", // Grifols
];
```

Código 4.13. Cuentas de prueba

```
    local: {
      url: 'http://localhost:9650/ext/bc/<subnet_name>/rpc',
      gasPrice: 225000000000,
      chainId: <subnet_chainid>,
      accounts: testAccounts
    },
```

Código 4.14. Datos de conexión subred

4.5. Compilación y despliegue de un contrato de prueba

El fichero package.json contiene aliases a los comandos de npx que permiten compilar, desplegar y ejecutar aplicaciones cliente para utilizar las DApps programadas, en nuestro caso, se ha modificado el fichero package.json y se han creado los ficheros main.js y deploy.js, como se muestran en Código 4.15, Código 4.16 y Código 4.17,

```
"scripts": {
  "compile": "npx hardhat compile",
  "main": "npx hardhat run main/Main.ts --network local",
  "deploy": "npx hardhat run scripts/deploy.ts --network local",
  "accounts": "npx hardhat accounts",
  "balances": "npx hardhat balances",
}
```

Código 4.15. Aliases de comandos hardat

```
import {
  Contract,
  ContractFactory
} from "ethers"
import { ethers } from "hardhat"

const main = async(): Promise<any> => {
  const Coin: ContractFactory = await ethers.getContractFactory("ExampleERC20")
  const coin: Contract = await Coin.deploy()

  await coin.deployed()
  console.log(‘Coin deployed to: ${coin.address}‘)
}

main()
.then(() => process.exit(0))
.catch(error => {
  console.error(error)
  process.exit(1)
})
```

Código 4.16. Código para desplegar el contrato inteligente

```
const { expect } = require('chai');
const { BigNumber } = require('ethers');
const { ethers } = require('hardhat');
const Web3 = require('web3');
import {testAccounts} from "../hardhat.config";
const readLineSync = require('readline-sync')

const main = async(): Promise<any> => {

// This code returns all the accounts defined in testAccounts in hardhat.config.js
const [admin] = await ethers.getSigners();

// Insert subnetwork name to connect with the endpoint
var url = 'http://localhost:9650/ext/bc/<subnet_name>/rpc';
var customHttpProvider = new ethers.providers.JsonRpcProvider(url);
console.log(await customHttpProvider.getBalance(admin.address));
console.log(await customHttpProvider.getBlockNumber());

var factory = await ethers.getContractFactory("ExampleERC20");
factory = factory.connect(admin);

// Insert contract address already deployed
const dapp = await factory.attach("0x");

await dapp.mint("0x", 1000)

}

main()
.then(() => process.exit(0))
.catch(error => {
  console.error(error)
  process.exit(1)
})
```

Código 4.17. Aplicación cliente main.js

4.6. Solidity, conceptos clave

Avalanche es compatible con la maquina virtual Etherum para desarrollar contratos inteligentes utilizando el lenguaje de programación Solidity. Este lenguaje ha sido diseñado exclusivamente para web3 e incluye conceptos ya existentes en otros lenguajes como estructuras, mapas y funciones pero, también, añade nuevos conceptos enfocados a redes blockchain y aplicaciones descentralizadas.

4.6.1. Address

Las redes blockchain identifican a los usuarios y contratos inteligente mediante identificadores, denominados direcciones (address), que se utilizan para hacer llamadas a las funciones de los contratos, transferir tokens entre usuarios o asociar la identidad de una persona con su información almacenada en los contratos de la red blockchain. En la siguiente Figura 5.1 se muestra un mapa de datos de tipo clave-valor cuya clave es la dirección (ID de una persona) y su valor es una estructura de datos denominada tPerson, que contiene los atributos que se muestran en Código 4.19; este mapa es el almacenamiento permanente de todos los ciudadanos que han sido dados de alta en la aplicación descentralizada.

```
contract MyContract {

    mapping(address => tPerson) people_;
}
```

Código 4.18. Tipo de dato Address

```
struct tPerson { // Struct
  address id;
  address nif;
  string name_;
  string surname1_;
  string surname2_;

  bool alive_;
}
```

Código 4.19. Tipo de dato persona

```
contract MyContract {

    address constant private owner;
    constructor()
    {
        owner = msg.sender
    }

    modifier isOwner()
    {
        require(owner == msg.sender,
        "ERROR: The signer is not the owner of the contract. Action denied.");
        _;
    }


    function SecureFunction() public isOwner()
    {
        // This code is executed only if the signer is the owner.
    }
}
```

Código 4.20. Ejemplo de restricción de una función con un modificador

4.6.2. Modifier

Los modificadores son funciones especiales que se utilizan como restricciones o comprobaciones antes de ejecutar las funciones. Los contratos inteligentes y los modificadores pueden acceder a datos de la transacción, uno de los más relevantes es 'msg.sender' que se corresponde con la dirección de la persona que ha ejecutado y firmado la transacción.

Por ejemplo, se puede usar un modificador para denegar la ejecución de una función a todos aquellos usuarios que ejecuten una función y firmen la transacción con una clave privada no autorizada. En el Código 4.20 se muestra el modificador que compara la dirección msg.sender, incluida en la transacción que representa al usuario que ha ejecutado la función, con la variable constante owner, creada al instanciar o desplegar el contrato en la red blockchain, de esta forma se garantiza que sólo el dueño de la clave privada que desplegó el contrato inteligente puede ejecutar esta acción.

4.6.3. Events

Los eventos permiten enviar notificaciones y mensajes de log a las aplicaciones cliente de forma asíncrona. La aplicación cliente escucha los eventos y cuando el evento es emitido desde el contrato inteligente se ejecuta el código asociado, actualizando el estado de la aplicación cliente y notificando al usuario de el evento emitido.

4.6.4. Gestión de memoria

El paradigma de las aplicaciones descentralizadas implica un cambio en la forma en la que se gestiona la memoria y, sobre todo, los mecanismos por los que se recompensa a los nodos por ofrecer sus recursos de cómputo y almacenamiento. En las aplicaciones tradicionales los recursos de cómputo y almacenamiento es responsabilidad de la empresa propietaria, sin embargo, en un entorno descentralizado los nodos que participan de la red mantienen el estado de la 'base de datos' de las aplicaciones desplegadas en la red blockchain, este almacenamiento tiene un coste que debe ser pagado por el propietario del contrato inteligente, como recompensa por procesar y almacenar su información.

Es importante conocer los distintos tipos de memoria para reducir los coste de mantener un contrato inteligente activo, este coste se conoce como comisión de gas o 'gas fee'. Solidity ofrece dos tipos de memoria 'Storage' y 'Memory'.

Storage

Los datos almacenados en storage [80] son persistentes en la cadena de bloques, esto significa que los datos persistirán incluso después de que la función que los modificó haya terminado de ejecutarse. Son compartidos entre todas las funciones de un contrato y se mantienen entre llamadas de función. Este tipo de almacenamiento se utiliza para guardar los mapas de datos que almacenan toda la información de los usuarios.

Memory

Los datos almacenados en memory son temporales, se borran cuando finaliza la función y su uso principal es para almacenar variables temporales, como parámetros de funciones o variables internas de la función.

El almacenamiento storage es más caro en términos de gas que los datos temporales de la memoria, por este motivo, es importante definir que variables deben ser temporales y cuáles deben ser permanentes. El almacenamiento permanente es caro, por lo que, no es recomendable almacenar ficheros o grandes cantidades de datos en redes blockchain, para estos usos se utilizan algoritmos como IPFS [57] para almacenar objetos de gran tamaño y garantizar su integridad.

5. APLICACIÓN PROPUESTA

En este capítulo se explicará los casos de uso, el diseño y el despliegue de la aplicación para la ciudad inteligente. Los casos de uso incluyen los diagramas y una explicación de los puntos principales a tener en cuenta como los permisos necesarios para ejecutar las acciones. El diseño incluye los diagramas de clase y flujo de la la solución implementada. Por último, el despliegue explica el procedimiento para crear los contratos en la blockchain de pruebas local y Fuji.

5.1. Casos de uso

5.1.1. Autenticación de los usuario en la aplicación descentralizada

Un usuario antes de poder utilizar las funcionalidades que ofrecen las aplicaciones gubernamentales debe autenticarse con su identidad blockchain, para ello, deberá introducir la clave privada o las 24 palabras en la aplicación cliente. La aplicación cliente calculará la clave pública-privada asociada y usará esta identidad para autorizar y firmar todas las acciones que realice en las aplicaciones descentralizadas, como se muestra en Figura 5.1.

5.1.2. Control de acceso DApp

La aplicación control de acceso es la encargada de gestionar los roles que el resto de aplicaciones descentralizadas pueden utilizar para autorizar o denegar las acciones de los usuarios, esta aplicación sigue un modelo centralizado de control de usuario basado en roles conocido como Role-Based Access Control (RBAC) para facilitar la gestión de usuarios y permisos en todo el sistema, no obstante, los desarrolladores de los contratos inteligentes pueden utilizar su propio sistema de roles para restringir accesos muy específicos de su caso de uso. En la Figura 5.2 se muestran los diagramas de los casos de uso de la aplicación control de acceso y roles.

Las funciones que se ofrecen en esta aplicación descentralizada son:

5.1.2.1. Mostrar todos los roles

Un administrador desea listar todos los roles dados de alta en el sistema.

- Control de acceso: Administrador

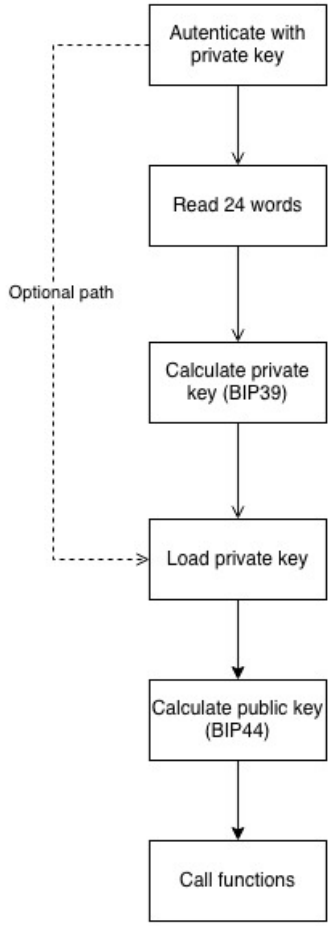


Fig. 5.1. Caso de uso del procedimiento de autenticación

5.1.2.2. Dar de alta rol

Cuando una aplicación descentralizada desea crear un nuevo rol específico deberá ponerse en contacto con un administrador y el administrador para dar de alta el nuevo rol.

- Control de acceso: Administrador

5.1.2.3. Dar de baja rol

Este caso de uso consiste en ofrecer la capacidad a los administradores para dar de baja un rol del sistema, invalidando todas las acciones permitidas.

- Control de acceso: Administrador

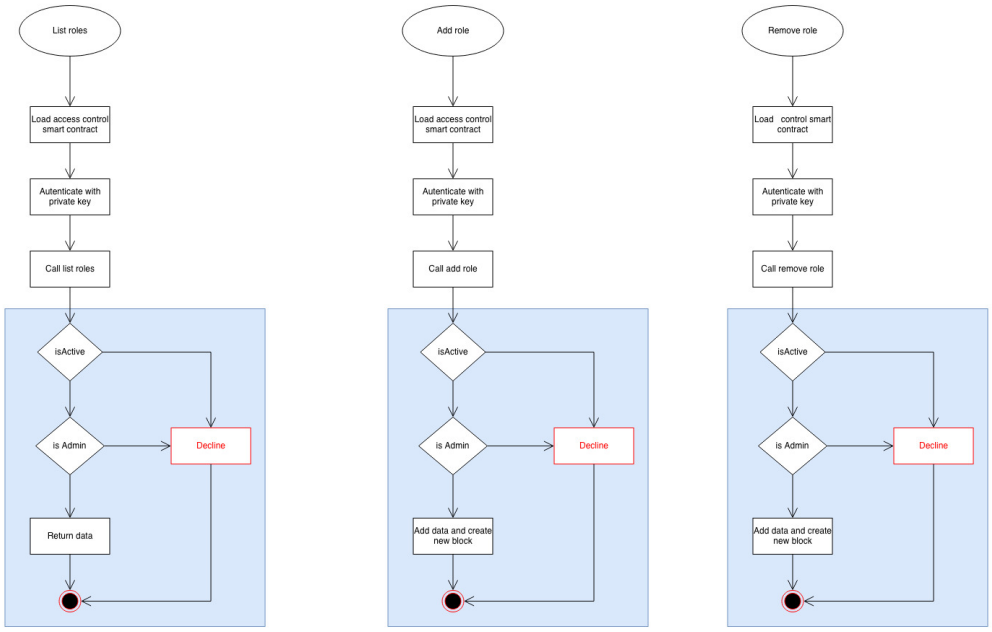


Fig. 5.2. Casos de uso de Access Control DApp

5.1.3. Civil DApp

La aplicación Civil DApp incluye las funcionalidades para dar de alta y consultar las personas registradas en la blockchain y, por tanto, con capacidad para interactuar con las DApps gubernamentales mediante el uso de las credenciales criptográficas asociadas a su identidad. En la Figura 5.3 y se muestran los diagramas de los casos de uso de la aplicación registro civil.

5.1.3.1. Mostrar todos los ciudadanos

Este caso de uso consiste en ofrecer a los funcionarios del registro civil la funcionalidad de consultar el listado de todos los ciudadanos dado de alta en la red blockchain.

- Control de acceso: Personal administrativo del registro civil

5.1.3.2. Dar de alta ciudadano

Este caso de uso permite dar de alta un nuevo ciudadano en la red blockchain, pudiendo ser un nuevo nacimiento o una persona que desee utilizar la aplicaciones descentralizadas que ofrece la administración del estado para reducir el tiempo al realizar trámites.

- Control de acceso: Personal administrativo del registro civil

5.1.3.3. Fallecimiento

Este caso de uso permite modificar el estado de un persona tras su fallecimiento, la variable 'alive' de la estructura tPerson se cambia a 'false'.

- Control de acceso: Personal administrativo del registro civil

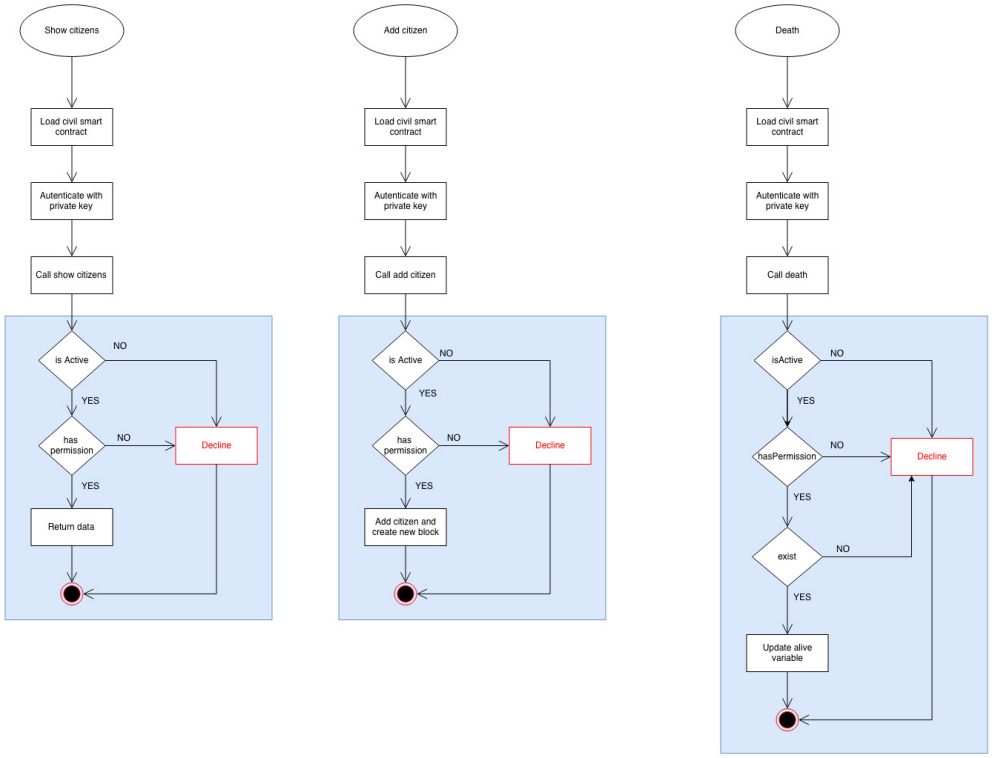


Fig. 5.3. Casos de uso de Civil DApp

5.1.4. Health DApp

Incluye las funcionalidades para gestionar personal sanitario como médicos o farmacéuticos, registros médicos de los pacientes, prescribir recetas, venta de medicamentos, registro de entidades sanitarias y laboratorios y el listado de medicamentos aprobados por los organismo de salud. En la Figura 5.4 y Figura 5.5 se muestran los diagramas de los casos de uso de la aplicación de sanidad.

5.1.4.1. Listar información

Este caso de uso consiste en ofrecer la función para listar información almacenada en los contratos inteligentes del sistema de salud, estos datos incluyen médicos, farmacéuticos, medicinas, laboratorios y el historial de un paciente. No todos los usuarios del sistema tendrán los mismos permisos y, por tanto, capacidad para realizar estas acciones de listado de registros.

A continuación se indican los permisos necesarios para consultar esta información:

- Listar médicos: Administrador
- Listar farmacéuticos: Administrador
- Listar medicinas: Administrador, médicos y farmacéuticos
- Listar laboratorios: Administrador, médicos y farmacéuticos

5.1.4.2. Mostrar historial del paciente

Este caso de uso consiste en ofrecer la función para que el médico pueda visualizar el historial clínico del paciente. Este historial incluye información sobre vacunas, enfermedades crónicas y grupo sanguíneo.

A continuación se indican los permisos necesarios para consultar esta información:

- Control de acceso: Médico

5.1.4.3. Dar de alta médico o farmacéutico

Este caso de uso cubre la funcionalidad de dar de alta médicos y farmacéuticos en el sistema de salud para que puedan realizar su actividad. El procedimiento para dar de alta un nuevo profesional debe ser realizado por una persona con el rol 'health.admin'.

Al igual que el caso de uso de añadir nuevos ciudadanos, los médicos o farmacéuticos que se den de alta pueden ser aquellos que ya estén ejerciendo en el sistema tradicional o los recién titulados.

Estas acciones requieren de varios controles adicionales:

- Requisitos: El farmacéutico o doctor deben estar dados de alta en el registro civil, en caso de que no estén la aplicación retornará un error indicando que deben ser añadidos, si no se cumple se declina la acción
- Control de acceso: Administrador

5.1.4.4. Dar de alta laboratorio

Este caso de uso cubre la funcionalidad de dar de alta laboratorios farmacéuticos que investigan y venden medicinas. El procedimiento para dar de alta un nuevo laboratorio, una vez haya pasado todo los requisitos legales y administrativos, será realizado por una persona con el rol 'health.admin'.

La entidad laboratorio deberá tener un administrador o propietario identificado con la clave pública de su identidad blockchain y, por tanto, esta persona deberá estar dado de alta en el registro civil.

Estas acciones requieren de varios controles adicionales:

- Requisitos: El propietario del laboratorio debe estar dados de alta en el registro civil, en caso de que no estén la aplicación retornará un error indicando que deben ser añadidos, si no se cumple se declina la acción
- Control de acceso: Administrador

5.1.4.5. Dar de alta medicina

Este caso de uso cubre la funcionalidad de dar de alta un medicamento, autorizado previamente para ser recetado. El registro será realizado por una persona con el rol 'health.admin'.

Estas acciones requieren de varios controles adicionales:

- Requisitos: El propietario del laboratorio debe estar dado de alta en el registro civil, en caso de que no estén la aplicación retornará un error indicando que deben ser añadidos, si no se cumple se declina la acción
- Control de acceso: Administrador

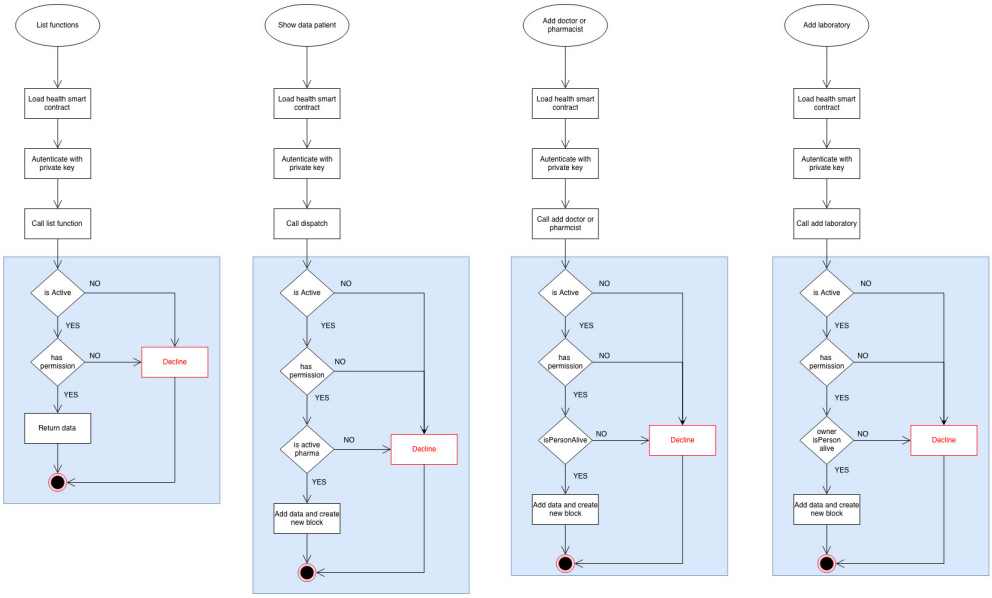


Fig. 5.4. Casos de uso de Health DApp

5.1.4.6. Recetar una medicina

Cuando un paciente visita un médico y el médico prescribe un medicamento deberá identificarse en el sistema con su clave privada, una vez identificado podrá seleccionar las medicinas y asignarlas al paciente usando su clave pública. La identidad del médico y del paciente se anonimizan con sus claves pública, ninguna persona o entidad podrá conocer a partir de una receta los datos personales del médico o del paciente asociados en la receta.

Estas acciones requieren de varios controles adicionales:

- Requisitos: El médico debe estar dado de alta en el registro de médicos con licencia vigente, si no se cumple se declina la acción Requisitos: El paciente debe estar dado de alta en el registro civil, si no se cumple se declina la acción
- Control de acceso: Médico

5.1.4.7. Expedir una medicina

El paciente podrá acudir a una farmacia, con la receta guardada en el contrato inteligente, y solicitar al farmacéutico la venta de los medicamentos. Para ello el farmacéutico se autenticará en el sistema con su clave privada y solicitará al paciente que se autentique con su clave pública y así visualizar todas la recetas vigentes.

Al realizar la venta se creará un nuevo registro en la blockchain asociando los medicamentos con las claves públicas del farmacéutico y del paciente, de nuevo, al asociarse con la clave pública el registro queda totalmente anonimizado, puesto que sólo los funcionarios del registro civil pueden conocer la relación de clave pública y la identidad asociada.

Estas acciones requieren de varios controles adicionales:

- Requisitos: El farmacéutico debe estar dado de alta en el registro de farmacéuticos con licencia vigente, si no se cumple se declina la acción Requisitos: El paciente debe estar dado de alta en el registro civil, si no se cumple se declina la acción
- Control de acceso: Farmacéuticos

5.1.4.8. Dar de baja un médico o farmacéutico

Este caso de uso cubre la necesidad de dar de baja a un profesional del sistema sanitario para bloquear sus permisos especiales en el sistema y denegando las operaciones que realice. Se modificará la variable 'status', de la estructura tDoctor o tPharmacist, con el valor 'INACTIVE'. Esta acción sólo podrá realizarla un usuario con el rol 'health.admin'.

Estas acciones requieren de varios controles adicionales:

- Requisitos: El doctor o farmacéutico debe estar dado de alta en el registro de profesionales del sistema salud.
- Control de acceso: Admninistrador

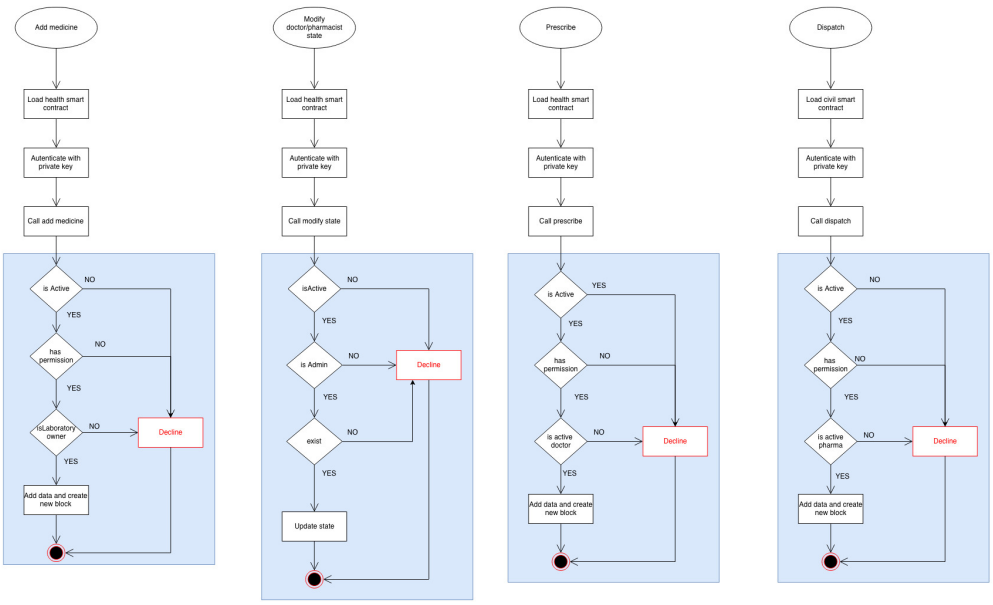


Fig. 5.5. Casos de uso de Health DApp

5.1.5. Academic App

Es una aplicación descentralizada enfocada al ámbito académico para digitalizar y automatizar la emisión de los títulos académicos que permitirán no sólo ahorrar los costes administrativos sino también ofrecerá la posibilidad de desarrollar nuevas funcionalidades para validad automáticamente estos títulos por terceras entidades. En la Figura 5.6 y Figura 5.7 se muestran los diagramas de los casos de uso de la aplicación de educación.

5.1.5.1. Listar información

Este caso de uso consiste en ofrecer funcionalidades para listar información almacenada en los contratos inteligentes del sistema de educación, estos datos incluyen instituciones académicas, estudiantes, titulaciones y títulos.

A continuación se indican los permisos necesarios para consultar esta información:

- Listar instituciones: Público
- Listar titulaciones: Público
- Listar estudiantes: Público
- Listar títulos: Publico

5.1.5.2. Dar de alta institución académica

El administrador del sistema podrá dar de alta una nueva institución académica, una vez haya pasado todo los requisitos legales y administrativos. La institución académica tendrá un director que será identificado con su clave pública y, por tanto, deberá estar dado de alta en el registro civil.

Estas acciones requieren de varios controles adicionales:

Requisitos: El director debe estar dado de alta en el registro civil, si no se cumple se declina la acción Control de acceso: Administrador

5.1.5.3. Dar de alta titulación

El administrador del sistema podrá dar de alta una nueva titulación académica y asociarla con la escuela que la imparte.

Estas acciones requieren de varios controles adicionales:

- Requisitos: No debe pertenecer ya a esa titulación en la institución
- Control de acceso: Administrador

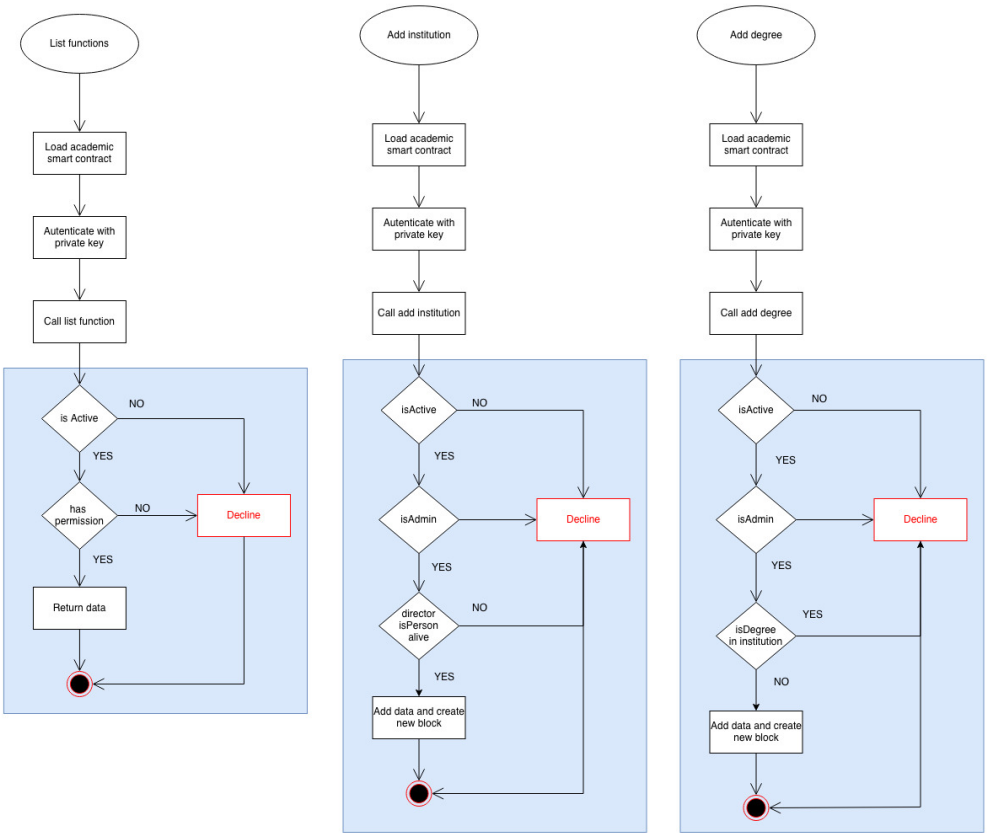


Fig. 5.6. Casos de uso de Academic DApp

5.1.5.4. Dar de alta estudiantes

El director podrá dar de alta en su institución nuevos alumnos y asociarlos con una de las titulaciones disponibles en la escuela. Para ello el director se autenticará con su clave privada y usará la clave pública de los estudiantes para darlos de alta en la titulación.

Estas acciones requieren de varios controles adicionales:

- Requisitos: El estudiante debe estar dado de alta en el registro civil, si no se cumple se declina la acción
- Requisitos: No debe pertenecer ya a esa titulación, si no se cumple se declina la acción
- Requisitos: La titulación debe estar dada de alta en esa institución, si no se cumple se declina la acción
- Control de acceso: Director

5.1.5.5. Emitir títulos

El director podrá emitir títulos académicos y asociarlos con los alumnos que hayan completado la formación, generando un NFT en formato imagen para que el estudiante pueda

presentarlo en las entidades que se lo soliciten. El título será registrado en el sistema asociándolo con el NFT y con la clave pública del estudiante asociado.

Estas acciones requieren de varios controles adicionales:

- Requisitos: El estudiante debe pertenecer a esa titulación, si no se cumple se declina la acción
- Requisitos: La titulación pertenece a escuela, si no se cumple se declina la acción
- Requisitos: El título no ha sido emitido previamente, si no se cumple se declina la acción
- Control de acceso: Director

5.1.5.6. Cancelar título emitido

Un título emitido puede ser cancelado por el director si se ha detectado algún problema en su emisión o si se demuestra que la obtención del título por esa persona no ha sido lícita. Se cambia la variable 'status', en la estructura tTitle, al valor 'INACTIVE'.

Estas acciones requieren de varios controles adicionales:

- Requisitos: El título existe
- Control de acceso: Director

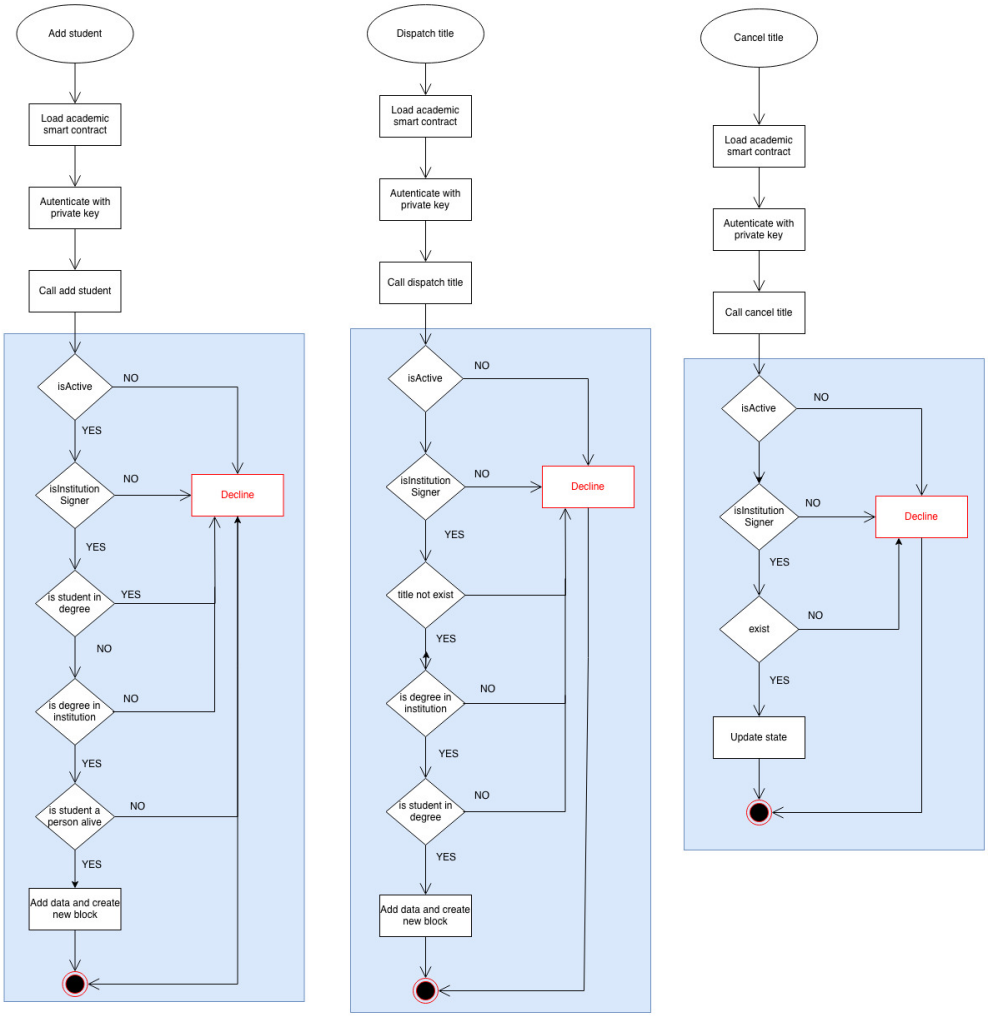


Fig. 5.7. Casos de uso de Academic DApp

5.2. Diseño

5.2.1. Diagramas de clases

En la Figura 5.8 se muestra el diagrama de clases y la estructura de datos de la aplicación descentralizada, la aplicación esta compuesta por una serie de contratos inteligentes (civil registry, dns, access control, health data, health app, academic data y academic app) y estructuras de datos (tPerson, tRole, tDispatch, tDoctor, tPatient, tlaboratory, tMedicine, tPharmacist, tPrescription, tInstitution, tDegree, tTitle y tStudent).

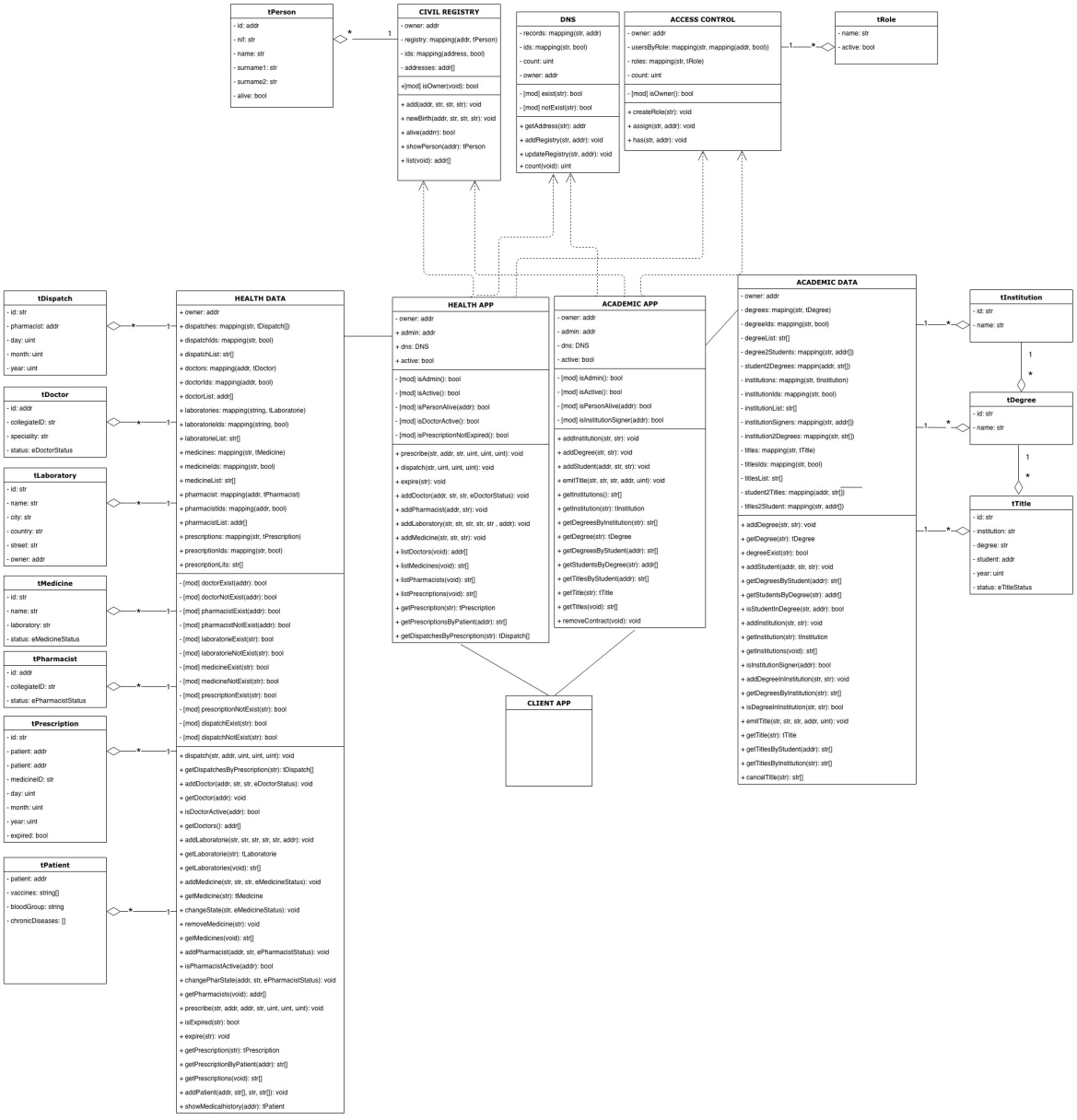


Fig. 5.8. Diagrama de clases

El software se ha diseñado aplicando el patrón proxy, como se muestra en Figura 5.9, para evitar que los contratos inteligentes que almacenan los mapas de datos tengan que ser actualizados por fallos o nuevas funcionalidades de la aplicación. Los contratos Health App y Academic App, actúan como proxy, el usuario interactúa con las funciones que expone el proxy y este se encarga de aplicar la lógica de negocio asociada y reenviar la llamada al contrato de datos, health data y academic data, para actualizar el estado de los datos guardados en los mapas.

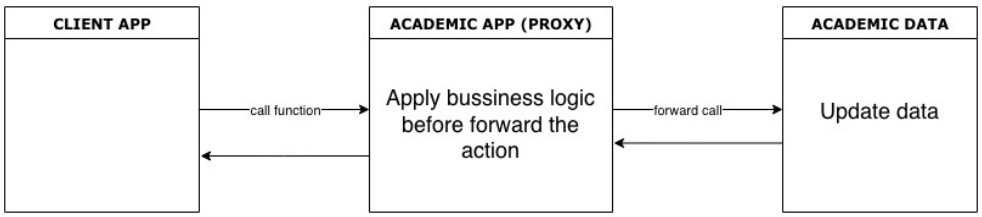


Fig. 5.9. Patrón de diseño proxy

Este patrón sigue teniendo limitaciones para abstraer a la aplicación cliente de posibles actualizaciones en los contratos proxy, por lo que, en caso de actualización de un contrato proxy la dirección del contrato desplegado será diferente y se tendría que actualizar la aplicación cliente (instalada en los dispositivos de los usuarios) en consecuencia.

En la web tradicional este problema se resuelve con el servicio DNS que asocia direcciones como www.google.es con una dirección dirección IP; para poder usar este servicio todos los dispositivos que usamos en internet conocen la dirección de su servidor DNS asociado, por ejemplo, el servidor DNS de Google tiene la dirección fija 8.8.8.8.

Este mismo concepto ha sido aplicado en la solución propuesta, se despliega un contrato inteligente (DNS) capaz de manejar una estructura de datos basada en un mapa clave-valor, donde la clave es el nombre del contrato inteligente y su valor es la dirección del contrato en activo, como se muestra en Código 5.1. La dirección del contrato DNS es conocido por el resto de contratos y por la aplicación cliente puede hacer llamadas a los diferentes contratos por su nombre sin necesidad de conocer la dirección. En el Código 5.2 se muestra la función listMedicines(), que devuelve el listado de medicamentos almacenados en Health Data, previo a llamar al método HelathData::getMedicines() se utiliza el objeto dns, para obtener la dirección del contrato Health data, ejecutando una llamada a la función getAddress, del contrato Dns, y pasando como parámetro el nombre del contrato que se desea consultar su dirección. En la cuarta línea del bloque de código se crea el objeto DNS pasando como parámetro la dirección fija del contrato DNS, previamente desplegado en la red blockchain.

```
contract Dns {

    mapping(string => address) records;
}
```

Código 5.1. Mapa de datos contrato DNS

```
contract HealthSystem
{
    address private owner_;
    Dns dns = Dns(0x55a4eDd8A2c051079b426E9fbdEe285368824a89);

    function listMedicines() external view returns (string[] memory)
    {
        return HealthData(dns.getAddress("HealthData")).getMedicines();
    }
}
```

Código 5.2. Llamada al servicio dns y reenvío de petición

Las estructuras de datos, nombradas con el prefijo 't', son la representación de los objetos del mundo real dentro de la blockchain y, por tanto, contienen una serie de atributos que identifican y/o caracterizan cada una de las entidades. En cada contrato inteligente se crean los mapas clave-valor, donde la clave es un identificador único para cada elemento y el valor es la instancia de la estructura de datos con los atributos cumplimentados. Por ejemplo, el contrato Academic Data define las estructuras de datos tInstitution, tDegree y tTitle que representan una institución académica, los grados universitarios que se ofertan y los títulos expedidos a los alumnos que hayan completado todas las materia, respectivamente.

Cada elemento representado por cada una de esas estructuras se identifica por un identificador en formato 'string' y se guarda en el mapa, como se muestra en Código 5.3, de esta forma la información queda almacenada en la 'base de datos' descentralizada, en cada uno de los nodos que pertenecen a la red blockchain. Los nodos garantizan la integridad de los datos mediante funciones matemáticas y criptográficas que generan los hash de bloque utilizado para enlazar los bloques y formar la cadena.

El contrato Civil registry contiene el registro de toda las personas dadas de alta en el sistema blockchain, el registro implica generar las credenciales de clave publico - privada, identidad blockchain, que les permitirá interactuar con la DApp gubernamental. Las personas se autenticarán en las DApp con su identidad blockchain, usando su clave privada, con la que posteriormente ejecutará y firmará todas las acciones.

El contrato Access control maneja el listado de roles permitidos en el sistema que se usan en el resto de contratos de la DApp para restringir las acciones que los usuarios pueden realizar con su identidad blockchain, este mecanismo de control basado en roles se conoce con el nombre de Role Based Access control (RBAC).

El sistema propuesto almacena toda la información de forma anónima con la clave pública del usuario; el único registro donde se relaciona la clave pública con los datos personales de cada individuo es en el contrato Civil Registry. El acceso a los registros de las personas está restringida exclusivamente a las personas con rol 'civil.administration'.

```
struct Title { // Struct
  string id;
  string institution_;
  string degree_;

  // The student is a person so the ID is the public address
  address student_;
  uint16 year_;
  eTitleStatus state_;
}

struct Degree {
  string id_;
  string name_;
}

struct Institution {
  string id_; // wallet institution
  string name_;
}

contract EducationData {
  address private owner_;

  mapping(string => Degree) degrees_;
  mapping(string => bool) degreesExist_;

  mapping(string => Institution) institutions_;
  mapping(string => bool) institutionsExist_;
  string[] institutionsIds_;

  mapping(string => string[]) institution2Degrees;

  mapping(string => Title) titles_;
  mapping(string => bool) titlesExist_;
  string[] titlesIds_;

  mapping(string => string[]) inst2titles_;
}
```

Código 5.3. Extracto del modelo de datos del contrato Academic Data

5.2.2. Diagrama de flujo

La Figura 5.10 muestra el diagrama de flujo de la interacción entre el usuario, la aplicación y los contrato inteligentes. El usuario puede utilizar las funciones de los contratos inteligentes a través de una aplicación tradicional, que puede estar en su móvil, ordenador o web; lo primero que debe hacer el usuario es autenticarse en la aplicación con su clave privada, como se muestra en Figura 5.1.

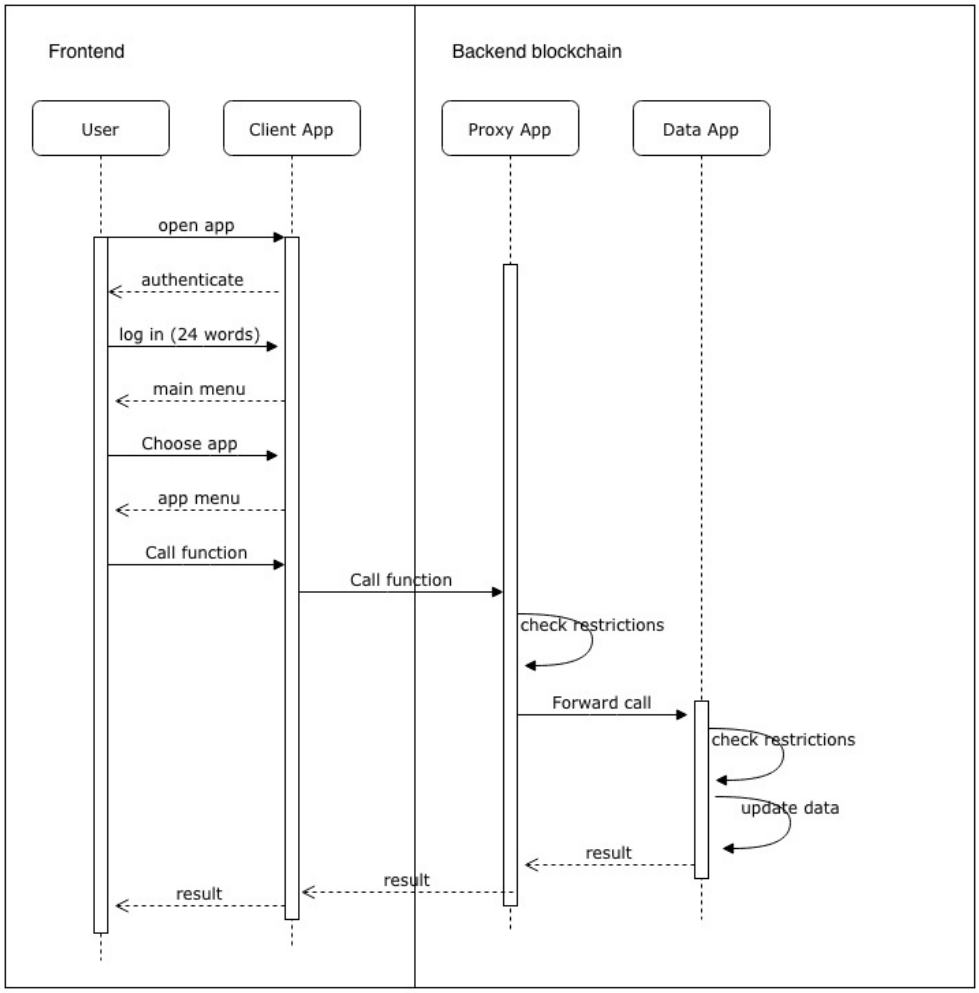


Fig. 5.10. Diagrama de flujo general

Tras realizar el proceso de autenticación se mostrarán las DApps disponibles (Academic App, Health App, Access control o Civil Register), dependiendo de la aplicación elegida por el usuario se mostrará un segundo menú con las funcionalidades disponibles en esa aplicación, como se muestra en la Figura 5.11.

El usuario podrá elegir la función que desea ejecutar y la aplicación cliente envía una petición al nodo blockchain para que ejecute el contrato inteligente (Proxy) que se encargará de realizar las comprobaciones, si la acción es autorizada se ejecuta la lógica de negocio asociada y se reenvía la acción al contrato inteligente de datos para su registro en la 'base de

datos' en blockchain.

El flujo finaliza una vez que los contratos han realizados todas las acciones y se retorna el resultado a la aplicación cliente para que el usuario lo visualice.

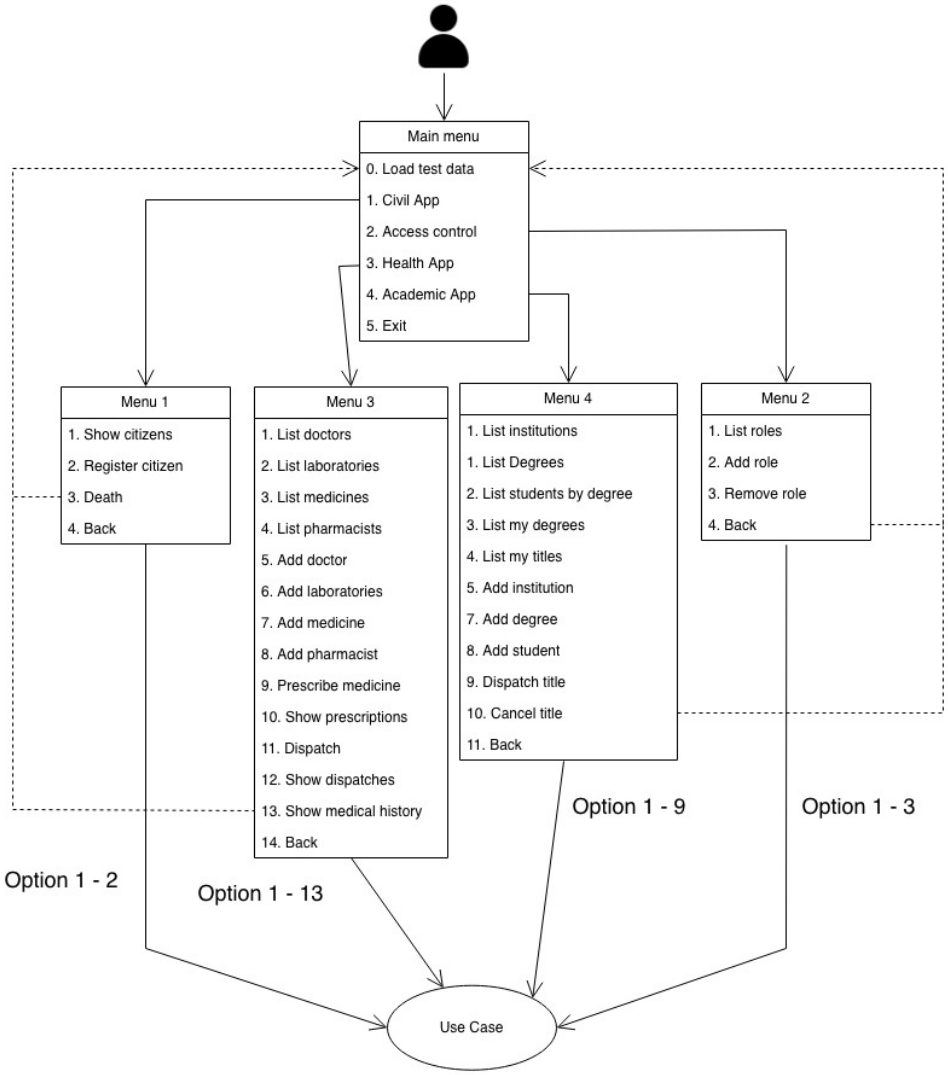


Fig. 5.11. Diagrama del menú de la aplicación cliente

5.2.3. Modelo de datos

Las estructuras de datos que se muestran en Sección 5.2.1 se utilizan para caracterizar los actores o elementos que participan en la aplicación, a diferencia de las bases de datos tradicionales, los datos no se relacionan mediante tablas y claves primarias. En blockchain las tablas son mapas de datos clave-valor en los contratos inteligentes, donde la clave actúa como clave primaria y el valor es la estructura de datos que representaría cada una de las columnas de una tabla tradicional, como se observa en la Figura 5.12.

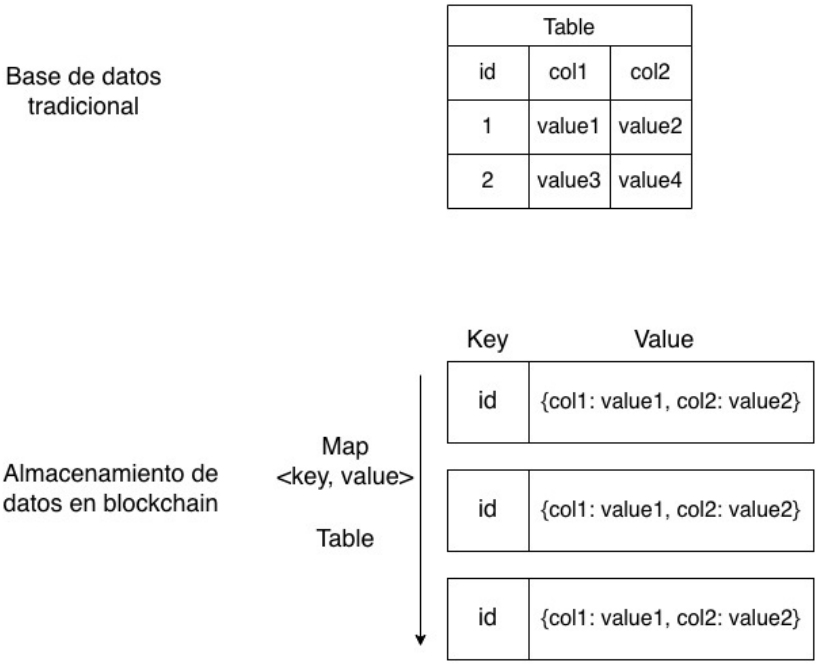


Fig. 5.12. Comparativa entre una base de datos tradicional y blockchain

En Código 5.3 se muestra un ejemplo de un contrato inteligente, los mapas de memoria y las estructuras que modelan la base de datos de educación, las tablas representadas en color azul en un base de datos tradicional no serían necesarias porque mediante sentencias JOIN se podrían consultar pero en blockchain no está disponible este tipo de consultas y es necesario crear mapas que relacionen tablas para relacionar datos entre los mapas.

En la Figura 5.13 y Figura 5.14 se representa el modelo de datos completo de una ciudad inteligente donde los ciudadanos usan su identidad digital en blockchain para registrarse en la escuela y registrar sus títulos académicos, trabajar como director o empresario, farmacéutico y médico, guardar el historial clínico y recetas.

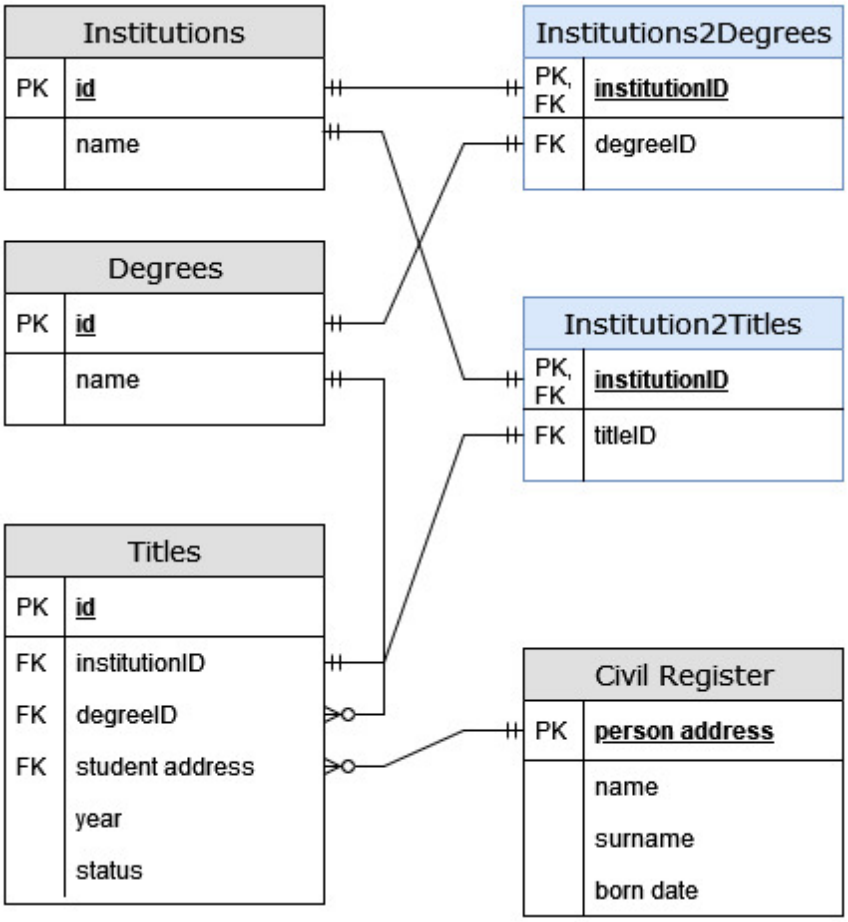


Fig. 5.13. Ejemplo modelo de datos para el caso de uso educación

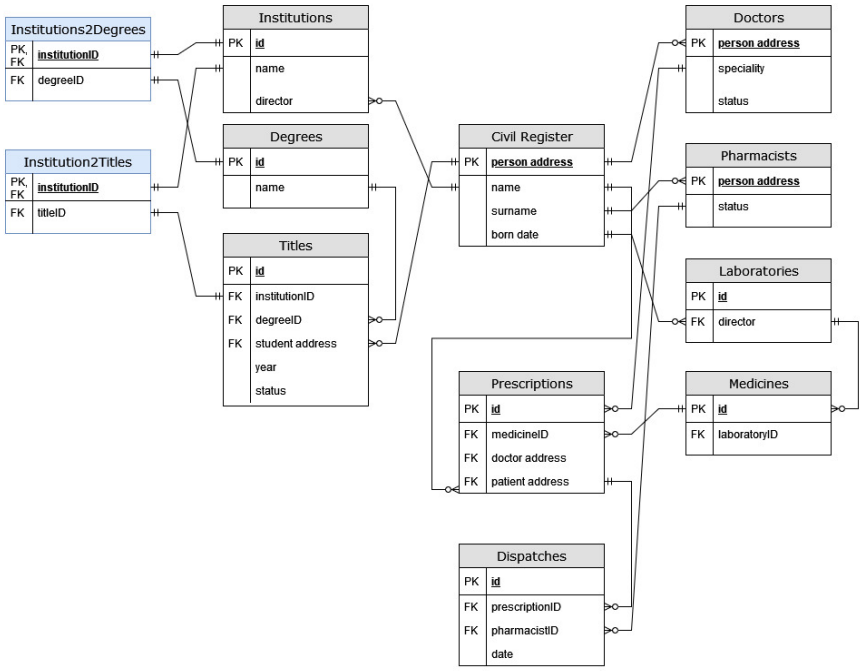


Fig. 5.14. Modelo de datos ciudad inteligente

5.3. Despliegue

5.3.1. Despliegue en entorno local

Para utilizar y probar los contratos inteligentes desarrollados hay que desplegarlos en el entorno de ejecución, la máquina virtual Etherum que se ejecuta en cada uno de los 5 nodos que conforman la red blockchain de pruebas local creada en Sección 4.3.2.

El framework Hardhat facilita la tarea de desplegar los contratos inteligentes, nos ofrece el script deploy.js como plantilla de código para desplegar cualquier contrato inteligente, como se muestra en el Código 5.5, el código utiliza el nombre de cada uno de los contratos principales y ejecuta la función deploy para desplegar el contrato.

El comando que se muestra en Código 5.4 ejecuta la función main del script deploy.js para desplegar el contrato en la red local configurada en el fichero hardhat.config.js como se muestra en Código 4.14 y se explica en Sección 4.3.3

```
$ npx hardhat run scripts/deploy.ts --network local
```

Código 5.4. Comando para desplegar contratos en red local

```
const main = async(): Promise<any> => {

  const [admin, citizen1] = await ethers.getSigners();

  // --------------------------------
  var StorageSC = await ethers.getContractFactory(<CONTRACTNAME>)
  StorageSC = StorageSC.connect(admin);

  var contract = await StorageSC.deploy();

  await contract.deployed()
  console.log(‘Contract <CONTRACTNAME> deployed to: ${contract.address}‘)

...
}
```

Código 5.5. Código javascript para desplegar contratos inteligentes

Por ejemplo, si queremos desplegar el contrato inteligente DNS se debería sustituir CONTRACTNAME por 'Dns' y la salida del script deploy.js mostrará por pantalla la dirección blockchain asociada a ese contrato, como se muestra en la Figura 5.15.

```
avalanche@9d52e7ea5291:~/citizenchain/src$ yarn deploy
yarn run v1.22.19
$ npx hardhat run scripts/deploy.ts --network local
Dns deployed to: 0x55a4eDd8A2c051079b426E9fbdEe285368824a89
```

Fig. 5.15. Captura contrato desplegado

5.3.2. Despliegue en entorno de pruebas Fuji

La red Fuji provee un entorno idéntico al que se ofrece en la red main pero los tokens usados para pagar los costes de transacción y despliegue de contratos no son reales y se solicitan a través de grifo de tokens https://test.core.app/tools/testnet-faucet. Antes de solicitar los tokens es recomendable instalar en el navegador la extensión Core App, la wallet oficial de Avalanche, y crear una nueva billetera a partir de 24 palabras, como se muestra en Figura 5.16.

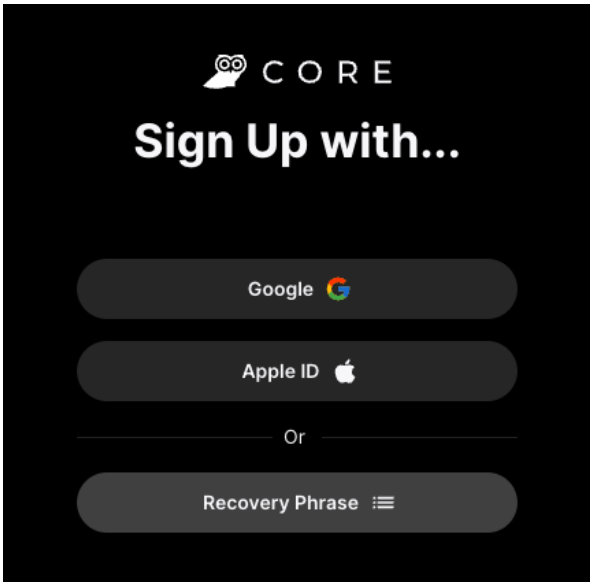


Fig. 5.16. Ventana principal Core App

Core App se abrirá con la billetera conectada a la red main, como las pruebas se van a realizar en la red pruebas Fuji, hay que conectar Core App a la red Fuji desde la pestaña de ajustes >Avanzado >Testnet Mode, como se indica en Figura 5.17, la aplicación mostrará la misma dirección de la billetera y un mensaje de aviso del estado de conexión con la red de pruebas Figura 5.18.

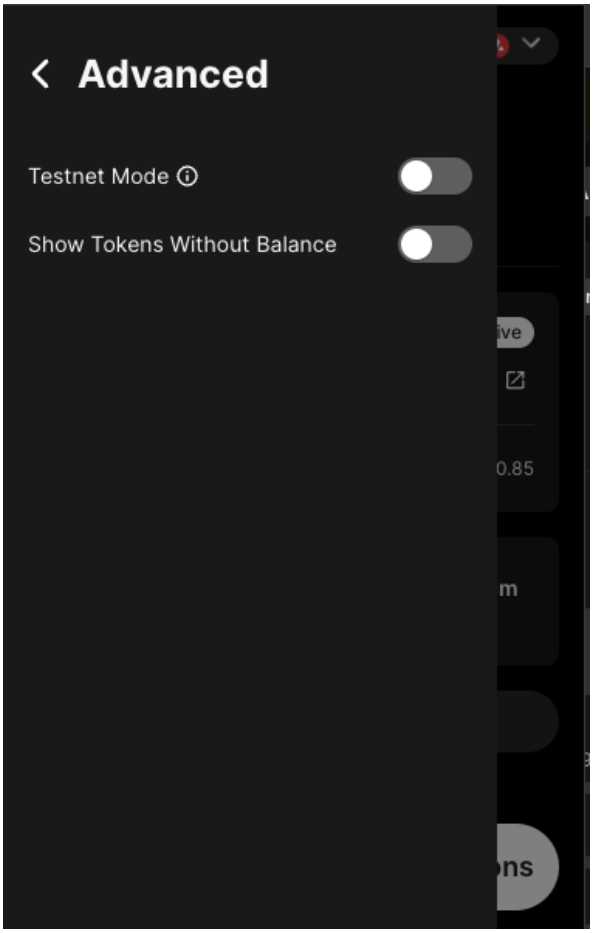


Fig. 5.17. Ajustes avanzados para activar el modo testnet

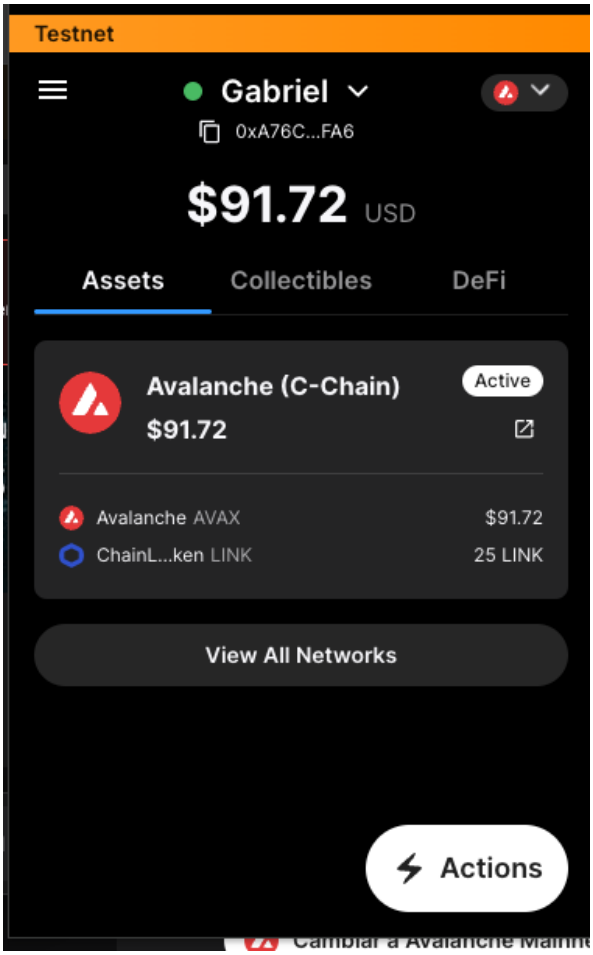


Fig. 5.18. Billetera en testnet

Para solicitar los token de prueba hay que acceder en la web https://test.core.app/tools/testnet-faucet y conectarnos con la billetera creada a través de la extensión Core App, el formulario de solicitud automáticamente detectará la dirección y se habilitará el botón solicitar AVAX, como se muestra en Figura 5.19, esta billetera será la que habrá que configurar en Hardhat para pagar los costes de despliegue de los contratos inteligentes.

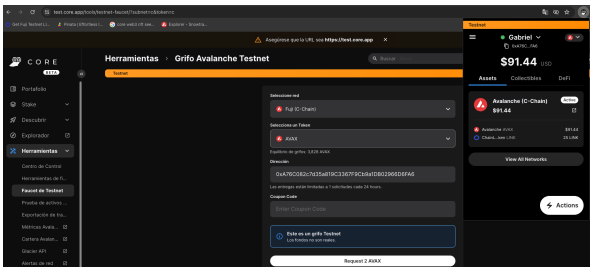


Fig. 5.19. Captura contrato desplegado

Para desplegar los contratos en el entorno de pruebas hay que configurar en el fichero hardhat.config.js los datos de conexión de un nodo endpoint de la red Fuji de Avalanche [81], como se observa en el Código 5.6.

```
networks: {
    local: {
      url: 'http://localhost:9650/ext/bc/spain/rpc',
      gasPrice: 225000000000,
      chainId: 8543,
      accounts: testAccounts
    },
    fuji: {
      url: 'https://api.avax-test.network/ext/bc/C/rpc',
      gasPrice: 225000000000,
      chainId: 43113,
      accounts: testAccounts
    }
}
```

Código 5.6. Configuración endpoint subred C-Chain en Fuji Testnet

El siguiente paso es preparar el script javascript, que se muestra en Código 5.7, para compilar y desplegar el contrato DNS. A continuación, se procederá a crear el script que se muestra en Código 5.8 para el resto de contratos.

La implementación en javascript utiliza las billeteras declaradas en la variable testAccounts del fichero hardhat.config.js para firmar las transacciones, en este caso será la primera cuenta del array testAccounts la que pagará los costes de despliegue, como se muestra en la línea "[admin] = await ethers.getSigners();".

```
//File: deploy-dns-fuji.ts

import {
  Contract,
  ContractFactory
} from "ethers"
import { ethers } from "hardhat"

const main = async(): Promise<any> => {

  "const [admin] = await ethers.getSigners();".

  // --------------------------------
  var sc = await ethers.getContractFactory("Dns")
  sc = sc.connect(admin);

  var dns = await sc.deploy();

  await dns.deployed()
  console.log(‘Dns deployed to: ${dns.address}‘)
}
```

Código 5.7. Script para compilar y desplegar DNS (deploy-dns-fuji.ts)

```
//File: deploy-fuji.ts

import {
  Contract,
  ContractFactory
} from "ethers"
import { ethers } from "hardhat"

const main = async(): Promise<any> => {

  const [admin] = await ethers.getSigners();

  // --------------------------------
  let contracts = ["AccessControl", "CivilRegistry",
                   "AcademicApp", "EducationData",
                   "HealthSystem", "HealthData"];

  for(var name of contracts)
  {
    StorageSC  = await ethers.getContractFactory(name)
    StorageSC = StorageSC.connect(admin);

    var storage = await StorageSC.deploy();

    await storage.deployed()
    console.log(`${name} deployed to: ${storage.address}`)
  }

}
```

Código 5.8. Script para compilar y desplegar el resto de contratos (deploy-fuji.ts)

Por último, se ejecutarán los comandos que se muestran en Código 5.9 para compilar y desplegar todos los contratos en la red Fuji, si el despliegue se ha realizado correctamente se mostrará el mensaje de la Figura 5.20 para cada uno de los contratos. En el analizador de bloques Snowtrace se pueden revisar los datos de la transacción el contrato desplegado, las imágenes Figura 5.21 y Figura 5.22 muestran campos relevantes de las transacciones como el campo 'From que se corresponde con la billetera que ha desplegado el contrato "From", el campo 'To' que identifica la dirección asociada al contrato, el coste en AVAX pagado en la transacción en el campo "Transaction Fee", el campo 'Block' con el número del bloque donde se ha incluido la transacción y el hash de la transacción en el campo "Transaction Hash".

```
$ npx hardhat run scripts/deploy-dns-fuji.ts --network fuji
$ npx hardhat run scripts/deploy-fuji.ts --network fuji
```

Código 5.9. Comando para desplegar contratos en red local

```
avalanche@9d52e7ea5291:~/citizenchain/src$ yarn deploy-dns-fuji
yarn run v1.22.19
$ npx hardhat run scripts/deploy-dns.ts --network fuji
Dns deployed to: 0xBCB1e8120153a14db7A0ba58340960FCA7A30A7F
Done in 14.73s.
avalanche@9d52e7ea5291:~/citizenchain/src$
```

Fig. 5.20. Ejemplo salida despliegue del contrato DNS

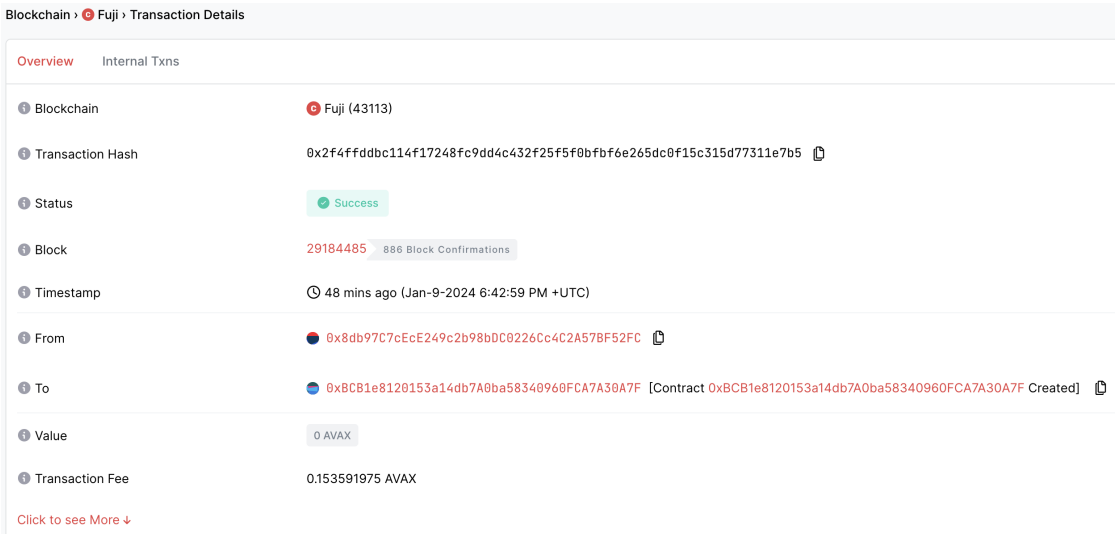


Fig. 5.21. Datos transacción despliegue contrato DNS

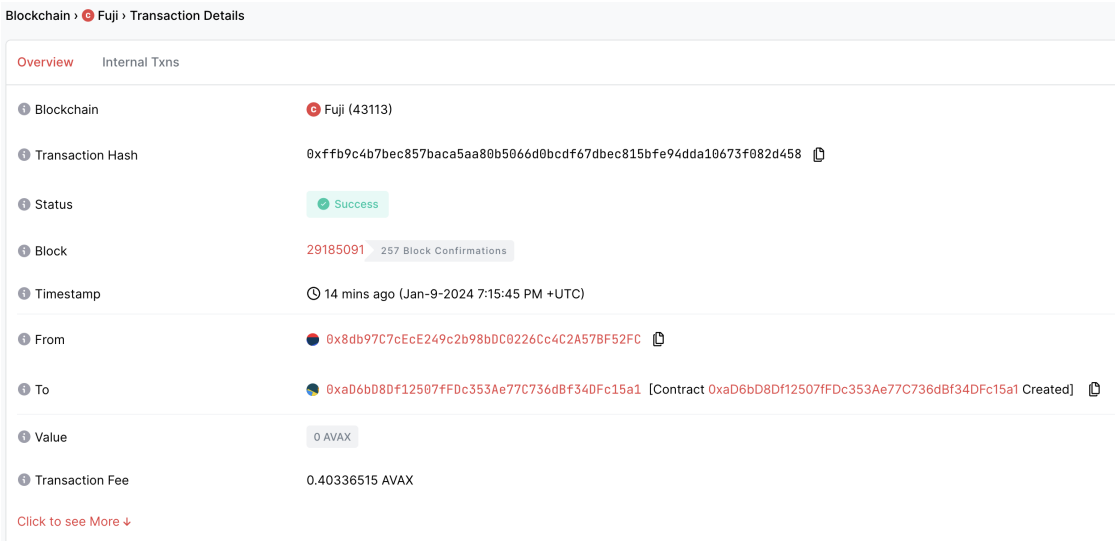


Fig. 5.22. Datos transacción despliegue contrato Civil

6. RESULTADOS

En este capítulo se exponen los resultados obtenidos al aplicar los conceptos de blockchain y aplicaciones descentralizadas para un uso gubernamental enmarcado en el campo de una ciudad inteligente.

6.1. Anonimización del dato

La blockchain y los contratos inteligentes son los encargados de garantizar que todas las acciones son realizadas por personas autorizadas por el estado, esta confianza está delegada en el funcionamiento intrínseco de una ciudad inteligente basada en la tecnología blockchain donde los funcionarios del estado registran a todos los ciudadanos y les otorgan su identidad descentralizada. Estas identidades, pareja de claves público-privada, evolucionan dentro de la blockchain a medida que lo hace el ciudadano, asociándola con nuevos roles dentro de la sociedad.

Gracias a esta confianza todas las operaciones pueden realizarse sin necesidad de exponer los datos personales de los participantes, lo único importante a registrar es la dirección que identifica al ciudadano en la aplicación descentralizada, como se muestra en el registro de venta de un medicamento de la Figura 6.1.

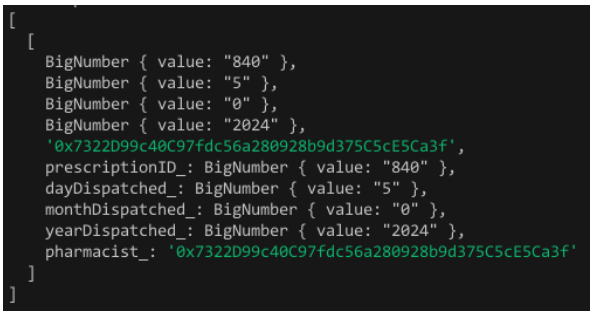


Fig. 6.1. Registro receta electrónica en blockchain

6.2. Integridad del dato

La tecnología blockchain garantiza la inmutabilidad de la información almacenada al calcular el hash de bloque en función de los datos de cada transacción incluida en dicho bloque, el algoritmo de consenso ejecutado por todos los nodos se encarga de determinar cuando un bloque puede ser considerado como válido y añadido a la cadena de bloques, esta actualización de la cadena es notificada al resto de nodos para que la incluyan en la copia de la cadena de cada uno de los nodos.

En la Figura 6.2 se observa el contrato del registro civil desde el analizador de bloques Snowtrace en la red Fuji, el contrato se ha ejecutado en 6 ocasiones generando 6 transacciones asociadas. A continuación, como se muestra en Figura 6.2 el usuario "Funcionario.ªñade un nuevo ciudadano en el registro civil.

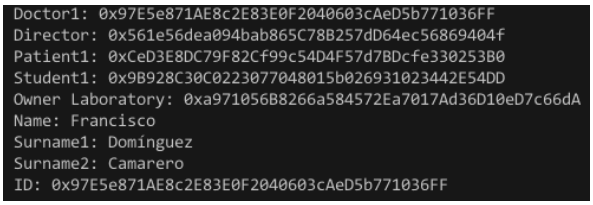


Fig. 6.2. Datos nuevo ciudadano añadido

El funcionario cuando realiza la acción de la Figura 6.2 genera una transacción con el hash '0x66e44a.....fbb59' incluida en el bloque "29253486", como se muestra en la Figura 6.3.

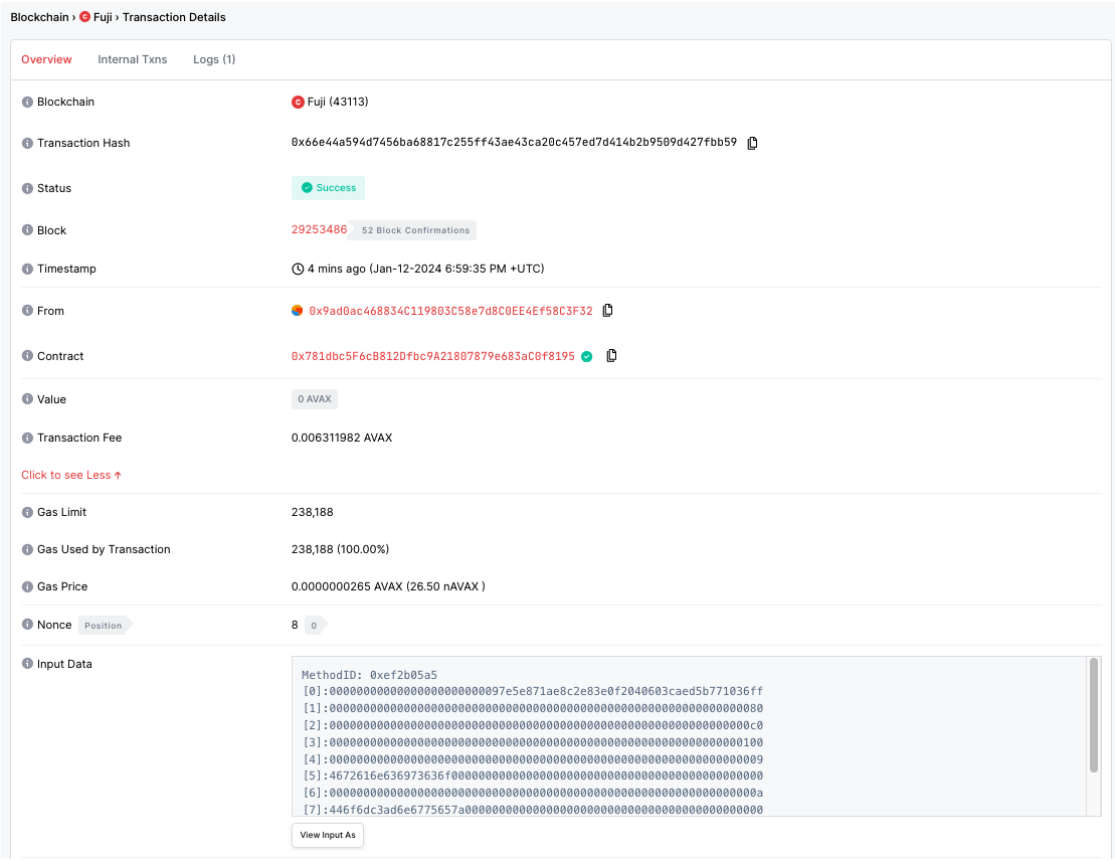


Fig. 6.3. Detalle de la transacción añadir ciudadano

Analizando el detalle de los datos 'Input Data' Código 6.1 se observa la información codificada en formato hexadecimal. Los datos de la transacción se dividen en múltiples campos: el campo [0] contiene dirección del ciudadano añadido "97e5e871ae8c2e83e0f2040603caed5b771036ff",

que coincide con los datos registrados por el funcionario en la Figura 6.2, los campos [5], [7] y [9] contiene el nombre y apellidos de la persona, codificados en hexadecimal.

```
MethodID: 0xef2b05a5
[0]:00000000000000000000000097e5e871ae8c2e83e0f2040603caed5b771036ff
[1]:0000000000000000000000000000000000000000000000000000000000000080
[2]:00000000000000000000000000000000000000000000000000000000000000c0
[3]:0000000000000000000000000000000000000000000000000000000000000100
[4]:0000000000000000000000000000000000000000000000000000000000000009
[5]:4672616e636973636f0000000000000000000000000000000000000000000000
[6]:000000000000000000000000000000000000000000000000000000000000000a
[7]:446f6dc3ad6e6775657a0000000000000000000000000000000000000000000000
[8]:0000000000000000000000000000000000000000000000000000000000000008
[9]:43616d617265726f0000000000000000000000000000000000000000000000000
```

Código 6.1. Datos extraídos del campo Input Data

Las evidencias anteriores demuestran que toda la información se almacena en la cadena de bloques en cada una de las transacciones, cada transacción se incluye en un bloque y este bloque es añadido y validado por todos los nodos de la red. En el campo "Block"de la Figura 6.3 también se puede ver la cantidad de nodos que han validado este bloque, cuantos más nodos validadores haya en la red más segura será la cadena de bloques y los datos que contiene. La integridad de la transacción y sus datos está garantizada por el algoritmo de consenso, ya que, para que la cadena comprometida, con el dato modificado, sea aprobada tendría que superar el consenso del resto de nodos.

6.3. Control de acciones y autenticación

La ciudad inteligente ofrece multitud de funcionalidades y acciones con la que los ciudadanos pueden interactuar, por tanto, es útil disponer de un mecanismo único de autenticación y comprobación de la identidad que permita o deniegue las acciones en función de los permisos asociados con cada identidad blockchain. En la Figura 6.4 se observa como un estudiante utiliza la aplicación académica para intentar emitir un título académico a su identidad, sin embargo, la aplicación deniega la emisión del título porque el estudiante no tiene los permisos necesarios, su identidad no tiene el rol 'director'.

La seguridad de la blockchain radica en la dificultad para robar las credenciales (clave privada o 24 palabras) de una identidad, por tanto, en blockchain es imprescindible que los propietarios sean responsables de la custodia de sus credenciales.

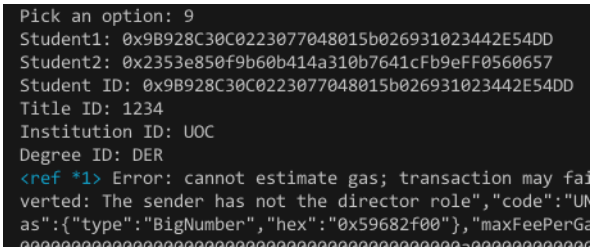


Fig. 6.4. Acción denegada: el estudiante no tiene permisos para crear títulos

6.4. Validación de un título académico

En la Figura 6.6 se muestra la identidad de un estudiante que ha finalizado los estudios universitarios y tiene asociado un NFT, que representa el título académico. En la Figura 6.5 se muestra el analizador de bloques Snowtrace donde se exponen todos los NFTs minteados en el contrato ERC721 de los títulos académicos; el contrato sólo tiene minteado un NFT con el tokenID ’223366’ asignado al ciudadano con la identidad ’0xCeD3E8D.....330253B0’, que se corresponde con la dirección del estudiante.

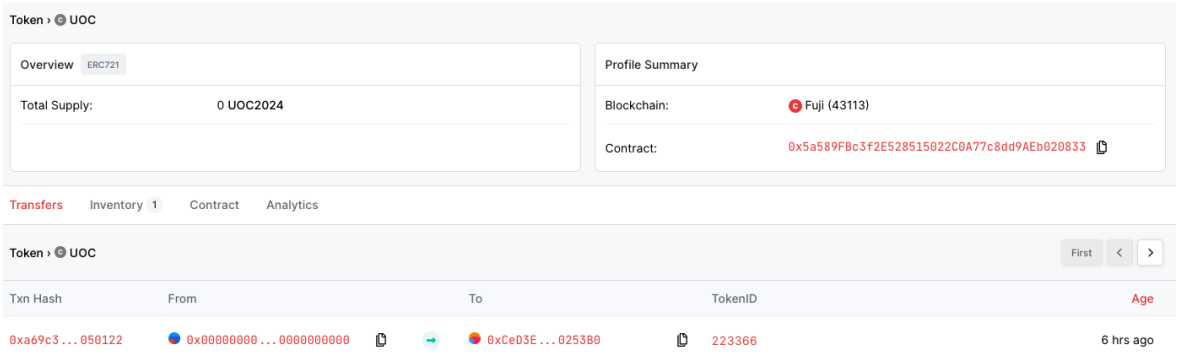


Fig. 6.5. Detalle contrato ERC721 de los títulos académicos en formato NFT

Para comprobar la veracidad del NFT, mostrado en Figura 6.6, se puede usar Snowtrace para acceder a los datos del estudiante usando su dirección ’0xCeD3E8D8....fe330253B0’, en la imagen Figura 6.7 se puede ver que el estudiante tiene un NFT minteado en el contrato UOC(UOC2024) identificado con la dirección de contrato

’0x5a589FBc3f2E528515022C0A77c8dd9AEb020833’.

Fig. 6.6. NFT de la estudiante Laura en la billetera de Core App

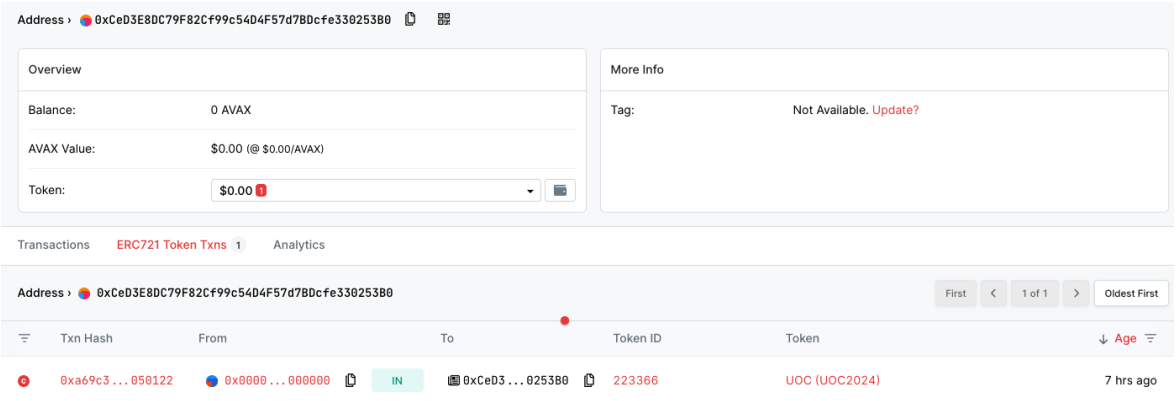


Fig. 6.7. NFT asociados al estudiante visto en Snowtrace

Gracias al uso de blockchain cualquier persona puede comprobar la veracidad de los títulos y confiar en que el registro ha sido realizado por una entidad o persona autorizada. Las funcionalidades que nos ofrece la blockchain permitiría comprobar toda esta información de forma automática y realizar acciones dependiendo del resultado de dicha comprobación, por ejemplo, en un proceso de concurso público se podría desplegar un contrato inteligente capaz de leer de todos los candidatos sus títulos universitarios (NFT), comprobar la validez de ese NFT revisando su emisor y seleccionar sólo a aquellos que posean una titulación concreta.

7. CONCLUSIONES

Las aplicaciones descentralizadas en blockchain tienen grandes ventajas para los gobiernos, identidad única con la que realizar cualquier trámite, automatizar los procesos administrativos, confiar en la integridad y veracidad de toda la información almacenada, digitalización de las administraciones cumpliendo con los planes de digitalización de la administración 2021-2025 [82].

7.1. Seguridad e integridad

7.1.1. Control de acceso mediante claves criptográficas

El sistema que se ha planteado con autenticación de usuarios utilizando mecanismos criptográficos difícil de usurpar protege la información, evita el uso de credenciales poco seguras como las contraseñas, especialmente contraseñas por defecto o sencillas, que la gran mayoría de las personas utilizan frecuentemente.

7.1.2. Identidad única

En informática e internet una de las acciones que todo usuario debe realizar siempre que desee utilizar los servicios de una plataforma es registrase con un correo electrónico o con algún sistema de autenticación delegada con proveedores externos de confianza, como Google, que emitan un token de confianza para autenticar al usuario en la plataforma. Utilizando la identidad descentralizada como identidad de confianza, creada por el gobierno a cada ciudadano, se facilita el acceso a todo tipo de servicios,la confianza de estas credenciales es ofrecida por el gobierno y la tecnología blockchain.

De esta manera los usuarios no tendrán que recordar el usuario y contraseña que han usado en cada plataforma, se estandarizan los mecanismos de autenticación y se eliminan los riesgos de robo de credenciales por actores malintencionados que atacan las bases de datos de las empresas para conseguir las contraseñas de los usuarios.

7.1.3. Autenticidad e Integridad

La red blockchain guarda información de todas las acciones que realizan los usuarios, cada transacción incluye metadatos que ayudan a mantener una trazabilidad de las interacciones entre todos los participantes del sistema. Los usuarios cuando ejecutan acciones deben firmarla con su clave privada y, por tanto, queda registrado en la cadena de bloques su identidad garantizando la dimensión de seguridad autenticidad y no repudio.

Otra dimensión de seguridad que se protege con el uso de cadenas de bloques es la integridad de la información, la redes blockchain utilizan mecanismos de hash para calcular y enlazar los bloques de la cadena y, por tanto, se garantiza que todo el contenido del bloque queda protegido por los árboles de Merkle y la relación entre bloques.

7.2. Almacenamiento de grandes cantidades de datos

Los propietarios de los contratos inteligentes deben pagar los costes por el almacenamiento de los nodos de la red, las redes blockchain no son aptas para este tipo de usos porque los costes del almacenamiento permanente de información es muy caro. Una forma de solucionar este problema es utilizar otros servicios como Pinata, que implementa el protocolo IPFS, para almacenar ficheros grandes en un sistema de nodos distribuidos y que garantizan la integridad de cada uno de los ficheros con identificadores CID basados en su contenido.

7.3. Consultas de datos

La tecnología blockchain no es una base de datos pensada para realizar consultas, no permite buscar información relacionada ni ofrece mecanismos de consultas rápidas como SQL, por lo que, para algunos casos de uso se deberán utilizar bases de datos tradicionales y mediante identificadores asociarla con información almacenada en blockchain. Blockchain es una tecnología con un uso muy concreto, almacenar información importante que deba garantizarse su integridad y autenticidad; el resto de casos de uso debería usarse soluciones tradicionales.

7.4. Automatización de procesos

Los contratos inteligentes ofrecen una gran versatilidad al poder acceder a los datos de los usuarios a través de una única identidad, por tanto, construir flujos de procesos mediante contratos inteligentes interconectados ayudará a los gobiernos a reducir lo costes administrativos, por ejemplo, se podría diseñar un contrato inteligente para el registro de personas que quieran aplicar a un proceso de oposición, en este contrato se incluiría un listado de títulos válidos para esta oposición pública y todos los usuarios, al aplicar con su identidad descentralizada, se podría comprobar automáticamente la posesión de dicho título de cada candidato, el contrato inteligente automáticamente podrá excluir del proceso de oposición a todos aquellos candidatos que no posean en su identidad descentralizada alguno de los títulos requeridos.

7.5. Sinergias entre web2 y web3

Las aplicaciones e internet tradicional (web2) y las aplicaciones descentralizas en blockchain (web3) deberán coexistir e interactuar mutuamente, como se ha expuesto en las conclusiones anteriores, las redes blockchain tiene un uso muy concreto y hay algunos usos para los que no fueron creadas como son el almacenamiento de grandes cantidades de datos o un sistema de base de datos para realizar consultas complejas, sin embargo, ambas tecnología convivirán y se utilizarán conjuntamente en futuras aplicaciones de forma totalmente transparente para el usuario.

7.6. Validación de objetivos

7.6.1. Análisis de blockchain en diferentes sectores

Se ha estudiado el uso de la tecnología blockchain en sectores para automatizar tareas y registrar información con garantías en distintos ámbitos como transporte, educación o sanidad.

7.6.2. Creación de contratos inteligentes

Se ha implementado una aplicación descentralizada básica para controlar información de los ciudadanos en una ciudad inteligente de forma anónimo pero garantizando la autenticidad e integridad de los datos guardados en la cadena de bloques. La solución obtenida permite asociar identidades descentralizadas con recetas y títulos académicos en una plataforma blockchain gubernamental, todas las interacciones en la ciudad inteligente se realizan mediante un mecanismo de autenticación único facilitado por el gobierno basado en criptografía y blockchain.

7.6.3. Creación de NFT

Para representar los títulos académicos y sustituir los títulos en papel tradicionales se ha diseñado un NFT que representa el mismo título pero en formato digital y protegido por la tecnología blockchain. Se ha conseguido mostrar el potencial de los NFT en un caso de uso real.

7.6.4. Integración y distribución

La aplicaciones descentralizadas tienen un problema a la hora de desplegar y distribuir la solución porque una vez desplegadas actualizarlas es imposible, para lograr un mecanismo de despliegue funcional y flexible se ha implementado la arquitectura en aplicando el patrón

proxy y un mecanismo similar al DNS tradicional que abstrae a la aplicación y a los usuarios de conocer la dirección del contrato.

7.7. Trabajos futuros

En este trabajo se ha desarrollado una solución gubernamental para los sectores de sanidad y educación donde se utiliza una única credencial criptográfica para la autenticación de los usuarios y asociar dicha identidad con otros datos como títulos académicos o recetas médicas. Se ha analizado la seguridad de la aplicación utilizando mecanismos de control de acceso basado en roles asociados a cada uno de los participantes de la red blockchain, además, se ha demostrado que el uso de las claves públicas permite proteger la identidad de los usuarios para terceras partes que deben interactuar con los datos generados por otros participantes.

7.7.1. Generación o almacenamiento de claves

En este trabajo no se ha explorado los mecanismos de autenticación de usuarios ni como generar las claves privadas en un entorno oficial, si se desea que toda la ciudadanía utilice este sistema, es necesario diseñar nuevas formas de generar o almacenar esta información, por ejemplo, se podría seguir manteniendo los algoritmos BIP39 y BIP44, que usan 24 palabras como entrada del algoritmo de generación de claves, pero guardarlos en el DNI electrónico y luego utilizar lectores que usen información del DNIe y autentiquen en el sistema a la persona, de esta manera se consigue que la ciudadanía no tenga que recordar el listado de palabras y se facilita su uso.

7.7.2. Arquitectura de la aplicación

Otra de las posibles acciones futuras a este trabajo es diseñar nuevos mecanismos de despliegue e interacción de los contratos inteligentes, en este proyecto se ha diseñado una primera aproximación de un sistema basado en DNS y arquitectura software usando el patrón Proxy para independizar los contratos de datos ïnmutablesz contratos de lógica de negocio "modificablesz así, poder adaptar la aplicación a nuevas funcionalidades o corrección de errores sin necesidad de migrar datos o actualizar todas las aplicaciones clientes, sin embargo, este sistema puede ser mejorado y rediseñado para ser más eficiente y más completo. Con nuevas arquitecturas evolucionadas podemos ampliar la aplicación a otras áreas y usos como en defensa [83], [84] o otros ámbitos y áreas dentro de las ciudades inteligentes como en hogares para detección de vulnerabilidades [85].

7.7.3. Implantación en una ciudad inteligente

Este sistema se puede implantar en todos los ámbitos de una ciudad inteligente, lo único que se necesita es un sistema gubernamental para la generación y custodia de las credenciales que identificarán a todos los ciudadanos en las aplicaciones descentralizadas. Los contratos inteligentes asociarán información con la identidad de cada persona, la información es compartida entre contratos inteligentes permitiendo automatizar procesos administrativos sin necesidad de que la persona aporte las evidencias manualmente, toda la información estará registrada y con garantías de su validez en la cadena de bloques.

BIBLIOGRAFÍA

[1] C. Villar Miguelez, V. Monzon Baeza, R. Parada y C. Monzo, "Guidelines for Renewal and Securitization of a Critical Infrastructure Based on IoT Networks," *Smart Cities*, vol. 6, n.º 2, pp. 728-743, 2023. DOI: 10.3390/smartcities6020035.

[2] C. Villar Miguélez, "Renovación y securización de una infraestructura crítica basada en redes IoT," 2021.

[3] "Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales," *Boletín Oficial del estado*, 2018.

[4] C. Tarazona Lizarraga, "Análisis de las necesidades de una Smart City en el marco de un desarrollo sostenible," 2020.

[5] A. R. Rocamora y A. Amellina, "Blockchain Applications and the Sustainable Development Goals," 2018. [En línea]. Disponible en: https://api.semanticscholar.org/CorpusID:169830454.

[6] *Blockchain for Sustainable Development: Promising use cases for the 2030 Agenda*, 2018. [En línea]. Disponible en: https://www.giz.de/de/downloads/giz2019-EN-Blockchain-Promising-Use-Cases.pdf.

[7] Reichental, *Smart Cities For Dummies*. For Dummies, 2020.

[8] *MiNT: Madrid impulsa su modelo Smart City*, https://www.madrid.es/portales/munimadrid/es/Inicio/Actualidad/Noticias/Madrid-impulsa-su-modelo-Smart-City/?vgnextoid=a5d4ca08566a1410VgnVCM1000000b205a0aRCRD&vgnextchannel=a12149fa40ec9410VgnVCM100000171f5a0aRCRD, Accedido Diciembre 2023.

[9] *Smart city infograph*, https://www.lanner-america.com/wp-content/uploads/2016/10/smart-city-infograph-v7.png.

[10] *IBM industria 4.0*, https://www.ibm.com/es-es/topics/industry-4-0, Accedido Diciembre 2023.

[11] *Quantum Fracture: Desde la sala de control de red electrica (España)*, https://www.youtube.com/watch?v=lmZwjxU0eRA, Accedido Diciembre 2023.

[12] *Repsol: ¿Qué son las Smart Grids o redes eléctricas inteligentes?* https://www.repsol.com/es/energia-futuro/tecnologia-innovacion/smart-grids/index.cshtml, Accedido Diciembre 2023.

[13] *Iberdrola: Redes inteligentes*, https://www.i-de.es/distribucion-electrica/redes-inteligentes, Accedido Diciembre 2023.

[14] *Endesa: Smart Grids*, https://www.fundacionendesa.org/es/educacion/endesa-educa/recursos/smart-grid, Accedido Diciembre 2023.

[15] *Una tecnología RFID que monitoriza el transporte del vino*, https://www.tecnovino.com/una-tecnologia-rfid-que-monitoriza-el-transporte-del-vino/, Accedido Diciembre 2023.

[16] *Introduction to smart transportation benefits*, https://es.digi.com/blog/post/introduction-to-smart-transportation-benefits, Accedido Diciembre 2023.

[17] *Gestión inteligente del tráfico: Optimizando el gasto en infraestructuras de su ciudad*, https://es.digi.com/blog/post/smart-traffic-management-optimizing-spend, Accedido Diciembre 2023.

[18] *Así es la movilidad por la que apuestan las ciudades inteligentes segura y sostenible*, https://elpais.com/tecnologia/2022-03-10/asi-es-la-movilidad-por-la-que-apuestan-las-ciudades-inteligente-segura-y-sostenible.html, Accedido Diciembre 2023.

[19] *Autopilot*, https://www.tesla.com/support/autopilot, Accedido Diciembre 2023.

[20] *Smart agro España gricultura4.0*, https://www.agrointeligencia.com/smart-agro-espana-agricultura-4-0/, Accedido Diciembre 2023.

[21] *Qué-significa 'Smart-Agro' para la agricultura 4.0*, https://www.interempresas.net/Horticola/Articulos/226562-Que-significa-'Smart-Agro'-para-la-agricultura-40.html, Accedido Diciembre 2023.

[22] T. BIO, "Internet of Things en la sanidad," (Accedido Diciembre 2023). [En línea]. Disponible en: https://www.tomorrow.bio/es/post/iot-en-la-sanidad-innovaciones-para-mejorar-la-atenci%C3%B3n-al-paciente-2023-06-4603446809-iot.

[23] *Apple watch salud*, https://www.apple.com/es/healthcare/apple-watch/, (Accedido Diciembre 2023).

[24] M. Fernández, "Apple watch detecta un infarto en EEUU," (Accedido Diciembre 2023). [En línea]. Disponible en: https://www.elespanol.com/omicrono/tecnologia/20210706/apple-watch-salva-mujer-aviso-dando-infarto/594441185%5F0.html.

[25] *Apple watch ayuda a detectar un infarto en España*, https://www.eldebate.com/sociedad/20221126/medico-residente-diagnostica-infarto-gracias-reloj-inteligente%5F75675.html, (Accedido Diciembre 2023).

[26] "Apple watch detecta accidentes y avisa a emergencias," (Accedido Diciembre 2023). [En línea]. Disponible en: https://support.apple.com.

[27] *IoT en eduación*, https://xtecnologica.com/iot-en-la-educacion/, (Accedido Diciembre 2023).

[28] *Blockchain History*, https://academy.binance.com/es/articles/history-of-blockchain, (Accedido Diciembre 2023).

[29] *Arbol de Merkle*, https://academy.bit2me.com/que-es-un-arbol-merkle/, (Accedido Diciembre 2023).

[30] *Top algoritmos de minería*, https://academy.bit2me.com/top-algoritmos-de-mineria-mas-utilizados/, (Accedido Diciembre 2023).

[31] *Bitcoin: A Peer-to-Peer Electronic Cash System*, https://bitcoin.org/en/bitcoin-paper, Accedido Diciembre 2023.

[32] *Cómo funciona la máquina Enigma*, https://www.curistoria.com/2020/04/como-funcionaba-la-maquina-enigma.html, Accedido Diciembre 2023.

[33] *Alan Turing, el arma secreta de los aliados*, https://historia.nationalgeographic.com.es/a/alan-turing-arma-secreta-aliados_16352.

[34] M. Bertaccini, *Cryptography Algorithms*. Packt Publishing, 2022.

[35] *Advanced Encryption Standard*, https://www.aescrypt.com/, Accedido Diciembre 2023.

[36] D. Wong, *Real World cryptography*. Manning, 2021.

[37] *Que son las claves RSA*, https://www.ionos.es/digitalguide/servidores/seguridad/claves-rsa/, Accedido Diciembre 2023.

[38] L. E. Hughes, *Pro Active Directory Certificate Services: Creating and Managing Digital Certificates for Use in Microsoft Networks*. Apress, 2022.

[39] *Alice y Bob, la más famosa pareja del mundo (de la criptografía)*, https://www.microsiervos.com/archivo/seguridad/alice-bob-famosa-pareja-criptografia.html, Accedido Diciembre 2023.

[40] *Website Phantom Wallet*, https://phantom.app, (Accedido Diciembre 2023).

[41] S. Community, *Website Solflare Wallet*, https://solflare.com/, (Accedido Diciembre 2023).

[42] C. Community, *Website Core Wallet*, https://about.core.app/, (Accedido Diciembre 2023).

[43] E. Community, *Website Exodus Wallet*, https://www.exodus.com/, (Accedido Diciembre 2023).

[44] A. Community, *Website Ledger Wallet*, https://atomicwallet.io/, (Accedido Diciembre 2023).

[45] E. Community, *Website Metamask Wallet*, https://metamask.io/, (Accedido Diciembre 2023).

[46] T. Community, *Website Ledger Wallet*, https://www.ledger.com/es, (Accedido Diciembre 2023).

[47] T. Community, *Website Trezor Wallet*, https://trezor.io/, (Accedido Diciembre 2023).

[48] S. M. Jain, *A Brief Introduction to Web3: Decentralized Web Fundamentals for App Development*. Apress, 2022.

[49] *Blockchain criptomonedas bitcoin y etherum, Etherum como nació y sus ventajas*, https://www.rankia.com/blog/blockchain-criptomonedas-bitcoin-ethereum/3683082-ethereum-que-como-nacio-cuales-son-sus-ventajas, Accedido Diciembre 2023.

[50] *Ethereum contratos inteligentes documentación para desarrolladores*, https://ethereum.org/es/developers/docs/smart-contracts/, Accedido Diciembre 2023.

[51] *Cardano: Proof of Stake*, https://docs.cardano.org/new-to-cardano/proof-of-stake/, Accedido Diciembre 2023.

[52] *Polygon: BFT consensus*, https://wiki.polygon.technology/docs/supernets/design/consensus/polybft/polybft-overview/, Accedido Diciembre 2023.

[53] *Avalanche consensus*, https://docs.avax.network/learn/avalanche/avalanche-consensus, Accedido Diciembre 2023.

[54] *Pinata Cloud*, https://www.pinata.cloud/, (Accedido Diciembre 2023).

[55] *Onedrive*, https://www.microsoft.com/es-es/microsoft-365/onedrive/online-cloud-storage, (Accedido Diciembre 2023).

[56] *Amazon S3*, https://aws.amazon.com/es/s3/, (Accedido Diciembre 2023).

[57] I. Community, *InterPlanetary File System (IPFS) almacenamiento de objetos descentralizado*, https://www.ipfs.tech/, (Accedido Diciembre 2023).

[58] *Cardano roadmap*, https://roadmap.cardano.org/en/byron/, Accedido Diciembre 2023.

[59] *Polygon web*, https://polygon.technology/, (Accedido Diciembre 2023).

[60] *Avalanche Consensus Whitepaper*, https://assets.website-files.com/5d80307810123f5ffbb34d6e/6009805681b416f34dcae012_Avalanche%20Consensus%20Whitepaper.pdf, Accedido Diciembre 2023.

[61] *Avalanche Network Documentation*, https://docs.avax.network, Accedido Diciembre 2023.

[62] *Certificado Electrónico de Ciudadano*, https://www.sede.fnmt.gob.es/certificados/persona-fisica, Accedido Diciembre 2023.

[63] E. Community, *Documentación Etherum Name Service*, https://docs.ens.domains/, (Accedido Diciembre 2023).

[64] *Documentation Etherum Attestation Service*, https://docs.attest.sh, Accedido Diciembre 2023.

[65] *On-chain vs Off-chain attestation*, https://docs.attest.sh/docs/learn/on-vs-off-chain-attestations, Accedido Diciembre 2023.

[66] *Identidades digitales usando EAS*, https://docs.attest.sh/docs/Use%20Cases/digital-identity, Accedido Diciembre 2023.

[67] *Definición POH*, https://blog.kleros.io/proof-of-humanity-que-es-y-como-funciona/, Accedido Diciembre 2023.

[68] D. Community, *Website Dock Descentralized Identity*, https://www.dock.io/, (Accedido Diciembre 2023).

[69] M. Security, "Decentralized identity and verifiable credentials," *Microsoft*, 2022.

[70] S. R. B. K. Suresh Rajarshi Pal, "Two-factor-based RSA key generation from fingerprint biometrics and password for secure communication," 2022.

[71] *Países pioneros en usar Blockchain para el registro de propiedad*, https://enzyme.biz/blog/blockchain-registro-propiedad, Accedido Diciembre 2023.

[72] *Más de 70 grandes empresas lanzan Alastria, un 'blockchain' para unirlos a todos*, https://www.elespanol.com/invertia/empresas/20171017/254974760_0.html, Accedido Diciembre 2023.

[73] *Socios alastria*, https://alastria.io/socios/, (Accedido Diciembre 2023).

[74] *Documentación Solidity*, https://docs.soliditylang.org/en/v0.8.23/, (Accedido Diciembre 2023).

[75] *Diagrama Etherum Virtual Machine*, https://ethereum.org/es/developers/docs/evm/, (Accedido Diciembre 2023).

[76] *Diagrama de flujo de ejecución EVM*, https://takenobu-hs.github.io/downloads/ethereum_evm_illustrated.pdf.

[77] *Hardhat overview*, https://hardhat.org/hardhat-runner/docs/getting-started#quick-start, (Accedido Diciembre 2023).

[78] *Avalanche Quickstart*, https://github.com/ava-labs/avalanche-smart-contract-quickstart, (Accedido Diciembre 2023).

[79] *BIP39 Mnemonic Code Online Tool*, https://iancoleman.io/bip39/, Accedido Diciembre 2023.

[80] *Solidity storage*, https://docs.soliditylang.org/en/v0.8.23/internals/layout_in_storage.html, (Accedido Diciembre 2023).

[81] *Datos endpoint fuji network*, https://docs.avax.network/tooling/rpc-providers.

[82] G. de España, "Plan de digitalización de las administraciones públicas 2021-2025," *Gobierno de España*, 2022.

[83] L. Concha Salor y V. Monzon Baeza, "Harnessing the Potential of Emerging Technologies to Break down Barriers in Tactical Communications," *Telecom*, vol. 4, n.º 4, pp. 709-731, 2023. DOI: 10.3390/telecom4040032.

[84] V. M. Baeza y L. C. Salor, "New horizons in tactical communications: An overview of emerging technologies possibilities," *IEEE Potentials*, vol. 43, n.º 1, pp. 12-19, 2024. DOI: 10.1109/MPOT.2023.3297326.

[85] S. Mulero-Palencia y V. Monzon Baeza, "Detection of Vulnerabilities in Smart Buildings Using the Shodan Tool," *Electronics*, vol. 12, n.º 23, 2023. DOI: 10.3390/electronics12234815.