Escuela Superior
de Ingeniería y Tecnología
Universidad de La Laguna

Trabajo de Fin de Grado

Grado en Ingeniería Informática

Blockchain para la Cadena de Custodia en Análisis Forense

Blockchain on the Chain of Custody

Diego Machín Guardia

La Laguna, 10 de septiembre de 2020

D. Cándido Caballero Gil, con N.I.F. 42.201.070-A profesor Ayudante Doctor adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutor

D. Néstor García Moreno, con N.I.F. 79.085.553-F, Ingeniero Informático, como cotutor

C E R T I F I C A (N)

Que la presente memoria titulada:

"Blockchain para la Cadena de Custodia en Análisis Forense"

ha sido realizada bajo su dirección por D. Diego Machín Guardia,

con N.I.F. 78539523-C.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 10 de septiembre de 2020

Agradecimientos

A mi familia por acompañarme durante toda la carrera
A mis amigos que han sido un gran apoyo
A mi tutor y cotutor por su comprensión y su acompañamiento

Licencia

Resumen

El objetivo de este trabajo se basa en una investigación sobre la tecnología Blockchain, en especial sobre la plataforma de Ethereum como implementación de la cadena de bloques y la integración de los contratos inteligentes. Dicha investigación se ha enfocado en la búsqueda de las posibles aplicaciones y ventajas que puede tener dicha tecnología en la cadena de custodia de pruebas forenses.

La investigación realizada se ha concretado en el desarrollo de una aplicación web descentralizada como método para la aplicación del conocimiento adquirido. Dicha aplicación permite la implementación de un sistema de control de la cadena de custodia de pruebas utilizando la plataforma Ethereum y los contratos inteligentes integrados en la misma. Dicha aplicación se encarga de asegurar la cadena de custodia implementando las características implícitas de una cadena de bloques en el proceso.

La realización de la aplicación materializa las distintas ventajas que puede aportar al proceso un sistema de Blockchain, así como sus principales puntos débiles para un análisis completo de la aplicación de esta tecnología a la cadena de custodia.

Palabras clave: Blockchain, Cadena de Custodia, Pruebas Forenses, Contratos inteligentes, Aplicaciones Descentralizadas, Ethereum.

Abstract

The objective of this project is a deep research about Blockchain Technology. Ethereum technology has been analysed as an implementation of blockchain and its integration of the smart contracts. The focus of the research has been set to the possible applications and advantages of the blockchain as an implementation of the chain of custody for forensic evidences.

The research has been materialized on the development of a decentralized web application in order to apply the acquired knowledge on the process. The application uses the platform Ethereum and its integration of smart contracts as a system to keep track of the chain of custody for each evidence. This application uses the inherent characteristics of blockchain technology to implement all the necessaries requirements ensuring the reliability of the process.

The development process that has been carried has shown different advantages and weakness of the blockchain. Thanks to a thoroughly analysis of the capabilities of the application the project encloses the analysis about the viability of the blockchain as a chain of custody for forensic evidences.

Keywords: Blockchain, chain of custody, forensic evidences, smart contracts, decentralized web applications, Ethereum.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1
Introducción

1.1 Motivación

La realización de este proyecto permite una oportunidad única para el aprendizaje de una tecnología en auge como es el Blockchain. La novedad de dicha tecnología unida con la gran variedad de posibles aplicaciones resulta altamente atrayente y permite ampliar mi conocimiento en un sector no tratado durante la carrera.

El auge de las criptomonedas ha aumentado la popularidad de la tecnología Blockchain lo que ha supuesto un incremento en su investigación y en el interés de la comunidad de desarrolladores. Este suceso ha conllevado tanto a la investigación como a la implementación de soluciones Blockchain para diversos usos.

Personalmente me atrae en gran medida el campo de la seguridad, así como el campo forense y la implementación de soluciones tecnológicas a diversos problemas cotidianos. Este proyecto recoge todos los campos antes mencionados y ha permitido aplicar la investigación sobre la tecnología para la implementación de una aplicación que mejore el sistema actual de la cadena de custodia, materializando así otro uso de la tecnología Blockchain.

1.2 Objetivos

Este trabajo se basa en dos objetivos fundamentales convergentes. El primer objetivo se basa en la comprensión de la tecnología y el funcionamiento de Blockchain en especial de las características específicas de la plataforma de Ethereum. Dentro de este ámbito se tendrá en cuenta el funcionamiento básico de una cadena de bloques y los contratos inteligentes. El segundo objetivo se basa en el análisis del estado actual de la cadena de custodia, así como la búsqueda de su automatización e incremento de seguridad.

La convergencia de dichos objetivos será una aplicación descentralizada que haga uso de los conocimientos obtenidos sobre Ethereum y los conocimientos del sistema de cadena de custodia. Los contratos inteligentes gestionarán las pruebas, los casos y los accesos a dichas pruebas de manera que todas las acciones queden registradas en la cadena de bloques. Esto supondrá un aumento en la seguridad, integridad y transparencia de la cadena de custodia.

1.3 Proceso de desarrollo

El desarrollo de este proyecto se ha realizado en dos grandes bloques, el bloque teórico y el bloque práctico, en los cuales se ha avanzado de manera simultánea.

El bloque teórico tiene una primera parte de aprendizaje sobre la tecnología Blockchain, donde se ha aprendido los conceptos fundamentales para el proyecto como cadena, bloque, nodo, transacción, etc. La segunda parte de este bloque corresponde con la búsqueda del estado del arte de

la cadena de custodia para su posterior investigación sobre las necesidades del sector para su implementación.

El bloque práctico tiene una etapa inicial de pruebas y experimentación, se han probado y seleccionado la tecnologías y herramientas, y se ha analizado la viabilidad de la solución final. En la segunda etapa se han realizado avances en el desarrollo de la aplicación en dos partes diferenciadas. La primera, donde se trabaja la funcionalidad de la cadena de bloques con sus contratos inteligentes, y la segunda, donde se ha desarrollado la parte visual de la aplicación. Una vez realizada esta etapa se ha procedido a comunicar ambas partes y a la realización de distintas funcionalidades y contratos más complejos.

Finalmente, una vez realizados los dos bloques, se ha analizado la aplicación final de manera que se identifiquen las ventajas y debilidades de la solución.

1.4 Estructura de la memoria

La estructura de esta memoria se ha dividido en 8 capítulos donde se ha expuesto el trabajo de investigación y desarrollo de la solución propuesta para la implementación de un sistema de cadena de custodia utilizando la tecnología Blockchain.

El primer capítulo expone una introducción al proyecto donde se especifica la motivación, los objetivos, el proceso que se ha seguido y el punto actual donde se explica la estructura de este documento.

El segundo capítulo se centra en un análisis de la tecnología Blockchain y cada uno de los conceptos claves para su comprensión. En este capítulo se ha entrado en detalle de la plataforma de Ethereum que se ha usado para construir la aplicación.

En el tercer capítulo se analiza es estado actual del tema centrándose en la cadena de custodia y las distintas implementaciones existentes actualmente. Así mismo, se trata el estado de la tecnología de la cadena de bloques y su relevancia.

En el cuarto capítulo se describe el diseño de la aplicación que se ha realizado y la implementación que se ha llevado a cabo. En el siguiente capítulo, el quinto, se analiza la solución aportada desde distintos enfoques y se explica el proceso de pruebas que se ha aplicado sobre los contratos.

En el sexto y séptimo capítulo se presentan las conclusiones obtenidas a raíz de la realización de este trabajo y se proponen unas líneas de trabajo futuro tanto para la mejora de la aplicación como para la mejora del uso de blockchain para la cadena de custodia.

Finalmente, en el punto octavo se describe y se propone un presupuesto en base al costo personal y de los componentes necesarios para hacer un despliegue de la solución propuesta.

Capítulo 2
Análisis de la tecnología Blockchain

En este capítulo se tratará Blockchain y todos los conceptos que han sido necesarios para el desarrollo de la aplicación descentralizada para la cadena de custodia.

2.1 Historia

La tecnología Blockchain surge en la búsqueda de una solución para el comercio electrónico sin la necesidad de una tercera parte confiable. Para ello se buscaba un sistema de pagos que no dependiera de la confianza entre las partes sino en pruebas criptográficas.

El concepto de la cadena de bloques se remonta a 1991 con el artículo de *"How to Time-Stamp a Digital Document"* [1]. En dicho documento se plantea la idea de la cadena de ficheros donde en cada elemento en el que se marca con la fecha se incluye la información sobre el fichero marcado anteriormente y la identificación del siguiente fichero a marcar. De esta manera se está creando una cadena que basa la fiabilidad en la estructura concatenada de los ficheros. Por otro lado, al año siguiente se publica un artículo donde se propone una mejora para el método de marcado de tiempo propuesto en el año anterior. En este artículo se hace referencia y se plantea el concepto de árbol de Merkle que tendrá gran importancia en el desarrollo de la cadena de bloques. [2]

Posteriormente, en el Libro Blanco *"Bitcoin: Un Sistema de Efectivo Electrónico Usuario-a-Usuario"* publicado por Satoshi Nakamoto en 2008 se plantean las bases sobre una implementación teórica para el problema del comercio electrónico resolviendo por primera vez el problema del doble gasto. En 2009 se lanza la criptomoneda conocida como Bitcoin, desarrollada bajo el seudónimo de Nakamoto y lanzada con licencia de código abierto donde se implementaban de manera concreta los conceptos publicados el año anterior. [3]

2.2 Funcionamiento

La tecnología Blockchain se basa en una sucesión de bloque en los cuales cada bloque se encuentra relacionado con su anterior de manera que se puede seguir un orden en el conjunto de bloques y cada uno sabe cual es el bloque anterior. La cadena que se forma se encuentra replicada en los nodos que componen la red distribuida en la que se basa la tecnología. Además, cuentan con mecanismos para añadir bloques, transmitir dichos bloques y asegurar la veracidad de la información de los bloques que componen la cadena.

Para poder entender correctamente el funcionamiento de esta tecnología es necesario abordar cada uno de los elementos que componen la cadena de bloques. Para ello se pondrá como ejemplo Bitcoin.

2.2.1 Bitcoin

La cadena de Blockchain Bitcoin es la primera cadena de bloques en ser entendida como Blockchain. Esta cadena fue creada con el propósito de servir para permitir el comercio digital creando su propio efectivo digital, que recibe el mismo nombre que la cadena. Esta solución permite el comercio digital entre usuarios sin la necesidad de que una entidad central confiable asegure la

fiabilidad del acuerdo de ambas partes, esto se asegura gracias a la solución del problema del doble gasto explicada a continuación.

2.2.2 Doble gasto

El mayor problema del comercio digital se basa en la dificultad existente en asegurar que un usuario dispone de los bienes que asegura tener. Para ello, la solución convencional se basa en la existencia de una tercera entidad confiable que certifica la veracidad de dicha afirmación, habitualmente una entidad bancaria. De esta manera la entidad confiable controla la existencia de los bienes de todas las partes y guardando registro de todos intercambios para poder certificar que los usuarios no han usado un bien que no disponen o que ya han intercambiado como pago.

La tecnología Blockchain elimina la entidad confiable y basa la veracidad de las operaciones en pruebas criptográficas. La solución propuesta propone que todas las personas dispongan de todas las operaciones para poder asegurar que los bienes del intercambio no han sido gastados. Esto supone que cada uno de los usuarios tendrá que disponer de todas las operaciones realizadas, sin embargo, esta solución sigue dependiendo de la confianza de que las transacciones que se reciban sean veraces.

2.2.3 Transacción

La transacción es la unidad mínima de la Blockchain y puede ser definida como cada una de las operaciones de escritura sobre la cadena. En el caso de Bitcoin cada una de las transacciones implica el envío de una moneda entre usuarios. Una transacción en una cadena de Bitcoin corresponde con una transferencia de una moneda, dicha transferencia implica que el emisor ha de firmar el hash de la transferencia anterior, la moneda, añadiendo la clave pública de su destinatario y firmándola con su clave privada.

Cada una de las transacciones estará formada por una o más entradas y habitualmente dos o más salidas. Las entradas serán las combinaciones de distintas transacciones anteriores que sumarán la cantidad a transferir y la salida será la transacción que irá al destinatario y el cambio que habitualmente volverá al emisor. Además, en las transacciones se puede añadir un incentivo de manera que los nodos de la red dedicados a minar los bloques prioricen la transacción y sea minada con mayor prioridad.

2.2.4 Bloque

Para la estructura de cadena de la Blockchain cada uno de los bloques que han de ser añadidos tienen que estar marcados de manera que contengan información del bloque anterior. En la creación de cada uno de los bloques se ha de utilizar el hash del bloque anterior para asegurar la continuidad de la cadena. Un bloque se forma con el conjunto de transacciones unidas al hash del nodo anterior y un campo nonce, que será utilizado en la prueba de trabajo requerida para la creación del bloque, proceso conocido como minado.

2.2.5 Pruebas de Trabajo

La eliminación de la confianza entre las partes se realiza como ha sido mencionado anteriormente mediante pruebas criptográficas, dichas pruebas son las pruebas de trabajo.

Para cada uno de los bloques que se crean se tiene que requerir que se realice un esfuerzo computacional. La prueba consistirá en que el hash del bloque ha de empezar con una cantidad establecida de 0s, para conseguir dicho suceso se tendrá que ir cambiando los datos dentro del bloque hasta que se encuentre un hash que cumpla con las especificaciones. Para obtener un cambio en el hash de salida será necesario modificar los datos del bloque, es por ello por lo que se modifica el nonce, campo dedicado a encontrar el número que hará que el hash comience por un número de 0s. Este empezará en 0 e irá incrementando hasta encontrar un número que cumpla las necesidades, minando así el bloque cuando se encuentre el valor. Para incentivar a los nodos de la red a minar los bloques se transfiere un incentivo por cada bloque minado además de los incentivos de las transacciones que han sido incluidas dentro del bloque.

Un bloque no podrá ser modificado sin volver a realizar todo el trabajo invertido en su minado y conforme se vayan añadiendo bloques a continuación de este, el bloque no podrá ser modificado sin que se modifiquen todos los bloques posteriores.

2.2.6 Consenso

Como la tecnología se compone de una red distribuida de nodos, se tendrá que llegar a un acuerdo para saber cuál es la cadena correcta y cuáles son los bloques válidos. Como las transacciones son anunciadas públicamente y los nodos tienen que actualizar la información, será necesario establecer un consenso para establecer un historial único del orden de sucesos de la cadena. Por ello se seleccionará siempre la cadena que contenga mayor cantidad de trabajo invertido, por lo que la cadena con mayor número de bloques será la correcta.

Dicho proceso produce que mientras más de la mitad del poder de procesado de la red sean honestos, la cadena seguirá creciendo, estableciendo un único orden y los bloques incorrectos serán descartados. Los nodos continuarán la generación de nuevos bloques utilizando el último hash correcto recibido.

2.3 Latencia

Uno de los mayores problemas de la tecnología Blockchain es la latencia. Una vez realizada una transacción no puede ser asegurado el éxito de esta. Para poder asegurar el éxito de la transacción se tiene que esperar a que la transacción sea minada en un bloque, tiempo que suele ser de 10 minutos. Además, una vez minado, se ha de propagar por los nodos de la red y esperar a que se añadan bloques a continuación de manera que se pueda asegurar que se encuentra en una cadena reconocida por los nodos de la red y dicho bloque no va a ser rechazado.

Esta latencia produce que la tecnología Blockchain no sea la indicada para acciones que han de transcurrir de manera inmediata como pueden ser acciones de bolsa.

2.4 Seguridad

La seguridad de la tecnología Blockchain se basa en dos ramas fundamentales, las funciones criptográficas y la red distribuida.

Las funciones criptográficas y la implementación de las pruebas de trabajo producen que al cambiar cualquier información sobre una transacción cambiará por completo el hash de la transacción y como resultado el bloque. Para que el bloque sea aceptado se tiene que realizar el nuevo minado del bloque que se ha cambiado para que encaje en la cadena.

Una vez que haya sido cambiado un bloque de la cadena, para que no sea rechazado por la red tendrá que producir nuevos bloques sobre el bloque fraudulento con una velocidad mayor a la del resto de la red. Esta velocidad de creación supone que se tenga que comprometer el 51% del poder de la red para poder llevar a cabo dicha tarea.

Por otro lado, los incentivos que se proporcionan a los nodos por el minado de un bloque suponen un beneficio para los nodos mineros. Este incentivo produce que los nodos mineros sean honestos ya que, siguiendo las normas de la red, tienen más posibilidades de incrementar sus ganancias.

2.5 Ethereum

Ethereum es una plataforma de código libre que utiliza la tecnología Blockchain como base. La filosofía, características propias de la implementación y uso de Ethereum se recoge en su libro blanco.[4]

Ethereum se caracteriza principalmente porque no es únicamente una plataforma de criptomoneda, sino que es programable, permitiendo así la realización de aplicaciones en la plataforma. Esto es posible gracias a que EVM, Ethereum Virtual Machine, máquina virtual disponible en cada uno de los nodos de Ethereum donde se ejecutan los contratos inteligentes es Turing completo. Esta funcionalidad permite que se puedan programar los contratos para la realización de numerosas funcionalidades.

La principal diferencia en la cadena de bloques de Ethereum respecto a Bitcoin subyace en el almacenamiento del estado. Para cada lista de transacciones que se aplican sobre la cadena Bitcoin solo almacena las transacciones realizadas mientras que Ethereum ha de almacenar no solo las transacciones sino el estado final.

Los estados están compuestos por objetos conocidos como cuentas, cada uno de ellos identificados con una dirección hexadecimal de 20 bytes. Además, contiene cuatro campos: el nonce, utilizado para que cada transacción solo sea procesada una vez, el saldo de Ether, el código del contrato si lo tiene y el almacenamiento de la cuenta. Las cuentas pueden ser de dos tipos, las cuentas externas, que son las controladas por clave pública y privada para su acceso y que permite la interacción de un usuario con la Blockchain, y las cuentas de contrato que son cuentas controladas por el propio código del contrato que se encuentra en la cadena y ejecutan las acciones programadas cuando son consultados.

Para la realización de las operaciones, al igual que en el minado de un bloque se reporta un beneficio al minero como recompensa por su trabajo computacional, en Ethereum, cada transacción dispone de un campo denominado *"startGas"* o gas inicial, donde se especifica el número de pasos computacionales máximos que puede realizar el contrato antes de que se anule la transacción. También existe un campo de precio del gas, o *"gasPrice"* donde se especifica la cuota a pagar por cada uno de los pasos realizados. De esta manera se paga de manera proporcional a los recursos consumidos.

La propia plataforma Ethereum incluye su propia criptomoneda, el Ether (ETH), que también es utilizada para el pago de los costes de las transacciones. Las unidades de subdivisión son las siguientes:

Unit	Wei Value
Wei	1wei
Kwei(Babbage)	1e3wei
Mwei(Lovelace)	1e6wei
Gwei(Shannon)	1e9wei
Microether(Szabo)	1e12wei
Milliether(Finney)	1e15wei
Ether	1e18wei

Tabla 2.1: Equivalencias de Ether

2.5.1 Contratos inteligentes

Los contratos inteligentes son un conjunto de instrucciones programadas que se ejecutan en las EVM. El contrato inteligente dispone del código que controla la cuenta en la que está desplegado dentro de la Blockchain y permite realizar distintas operaciones al recibir una transacción.

Con la realización de una transacción sobre un contrato se inicia la ejecución del contrato inteligente, dicha operación se realizará de manera atómica, aunque se llame a numerosos contratos. El éxito de la transacción dependerá de la finalización de la ejecución del contrato sin que se produzca ningún error. En el caso de que se produzca algún error o que se acabe el gas antes de la finalización del código del contrato entonces se revertirán todos los cambios a excepción de la transferencia del gas al minero que estaba ejecutando el contrato.

2.5.2 Proof of Stake

La prueba por depósito conocida como *"Proof of Stake (PoS)"*, es un tipo de algoritmo de consenso. Este tipo de algoritmo basa el consenso en la existencia de unos nodos validadores que sustituyen a los nodos mineros en el algoritmo de prueba de trabajo. Los validadores son elegidos por la Blockchain mediante una combinación del depósito y funciones aleatorias, que pueden ser combinadas con otros datos como la edad de la moneda. Una vez el validador es seleccionado este será el encargado de validar el bloque y añadirlo a la cadena, la recompensa que anteriormente se llevaba el minero se otorga al validador.

Este algoritmo mejora en alta medida ya que no se necesita una especialización de hardware para poder competir en el minado. Disminuye la electricidad y la computación gastada de manera innecesaria por el resto de los nodos que no consiguen minar el bloque.

Este algoritmo se ha implementado por Ethereum y su implementación se conoce como Casper. La implementación se basa en la prueba de depósito, los nodos que quieran convertirse en validadores tendrán que realizar un depósito de sus fondos. En el caso de que un nodo intente una validación incorrecta o fomente la validación de nodos en ramas distintas a la rama principal sus fondos serán requisados y perderá todos sus fondos. La implementación de PoS ha sido añadida en el paso a Ethereum 2.0 que ha sido realizado en agosto de 2020. [5]

Capítulo 3
Antecedentes y estado actual

Se aborda la cadena de custodia, el estado actual de dicho procedimiento y distintos métodos o implementaciones que son usadas. También se incluye el estado actual de Ethereum.

3.1 Cadena de custodia

La cadena de custodia es el proceso por el cual se asegura la legitimidad de las pruebas de manera que se asegure que la integridad de estas y su inmutabilidad desde el momento de su recogida. Este proceso es especialmente sensible en el procedimiento penal pero también puede ser utilizado con el fin de mantener la integridad de algún elemento en un proceso diferente. Un ejemplo podría ser la cadena de custodia sobre la información o sobre ficheros que puede proponer una organización para asegurar la restricción al acceso a documentos confidenciales.

Sin embargo, aunque la cadena de custodia pueda tener numerosos usos y casuísticas, el proceso conocido como cadena de custodia predomina por la protección de las evidencias en el proceso penal, como parte fundamental para la validez de estas. La importancia de este proceso y su definición se expone por el Tribunal Supremo como:

"En cuanto a la cadena de custodia el problema que plantea -hemos dicho en SSTS. 1190/2009 de 3.12 y 6/2010 de 27.1 - es garantizar que desde que se recogen los vestigios relacionados con el delito hasta que llegan a concretarse como pruebas en el momento del juicio, aquello sobre lo que recaerá la inmediación, publicidad y contradicción de las partes y el juicio de los juzgadores es lo mismo. Es a través de la corrección de la cadena de custodia como se satisface la garantía de la 'mismidad' de la prueba. Se ha dicho por la doctrina que la cadena de custodia es una figura tomada de la realidad a la que tiñe de valor jurídico con el fin de en su caso, identificar en todo la unidad de la sustancia estupefaciente, pues al tener que pasar por distintos lugares para que se verifiquen los correspondientes exámenes, es necesario tener la completa seguridad de lo que se traslada, lo que se mide, lo que se pesa y lo que se analiza es lo mismo en todo momento, desde el instante mismo en que se recoge del lugar del delito hasta el fomento final en que se estudia y destruye." [6]

Con ello se afirma la necesidad de un proceso por el cual se asegure la integridad de las evidencias para la validez de estas. Esto es necesario para la utilización de las pruebas en procesos judiciales ya que ante la demostración del incumplimiento de la cadena de custodia podría suponer la desestimación de las pruebas aportadas. A pesar de ello, no solo es necesario la presentación de una duda sobre la fiabilidad de la cadena de custodia, sino que serán necesarias las pruebas para asegurar su incumplimiento, para poder ser invalidada ha de existir la certeza de que no se han mantenido las garantías del proceso, como queda especificado en STS 777/2013 en el artículo séptimo. [7]

3.2 Estado del arte

Existen numerosos métodos y procedimientos de implementación de la cadena de custodia dependiendo de las necesidades concretas. Numerosas implementaciones y actuaciones son llevadas a cabo en función de las características propias del elemento al que se le ha de aplicar el procedimiento, así como los recursos disponibles y las directrices que reciba el organismo o persona responsable de su implementación.

Como se expone en el artículo de *"B-CoC: A Blockchain-based Chain of Custody for Evidences Management in Digital Forensics"* [8], aún existen numerosas implementaciones no digitalizadas de la cadena de custodia, así como el manejo de pruebas forenses digitales con implementaciones físicas donde no se tiene en cuenta el tipo de prueba o la posibilidad de que sea una prueba digital. Esta problemática no solo afecta a las pruebas digitales sino a todo el proceso de la cadena de custodia que podría ser altamente mejorado con su digitalización.

La simple digitalización de un sistema de cadena de custodia no es válida ya que se ha de poder asegurar la integridad de este. Dicha implementación ha de asegurar:

- Veracidad: La información que se añada al proceso de cadena de custodia tiene que ser cierto. No se puede permitir la inclusión de información que no esté verificada o que un usuario pueda manipular en su propio beneficio o interés.
- Integridad: La información que está recogida dentro de un proceso no ha de poder ser modificada, así como las pruebas que este protege. La alteración de la información vertida pondría en riesgo la seguridad de que los datos modificados no hayan sido editados para alterar el contenido o el procedimiento de custodia de una prueba. Si no se verifica la integridad un usuario podría manipular la cadena para borrar un rastro o su interacción con una prueba.
- Trazabilidad: La información referente al estado de la prueba, su ubicación o la persona encargada de su custodia ha de estar siempre presente y no puede ser indefinido. Por otro lado, se debe asegurar la posibilidad de verificar y seguir todos los procesos y todo el recorrido de la prueba desde que es recogida hasta el momento actual o el momento de destrucción de esta.
- Verificabilidad: Todas las personas asignadas al proceso tienen que poder comprobar que el proceso ha sido llevado de manera correcta y que se ha cumplido con los requerimientos propios.

Actualmente en España, dependiendo del organismo que implemente el proceso se pueden encontrar dos tipos principales de implementación para la cadena de custodia.

El primero, que es el más habitual, es la existencia de una cadena de custodia física en la que cada una de las pruebas es registrada en un papel. Cada vez que se registre un nuevo cambio en la cadena de custodia, es decir, una sesión de la custodia de una prueba a otra persona se registra en el papel el nuevo responsable de la custodia y que se quedará con el papel de la custodia que siempre ha de acompañar a la prueba. La persona que ha cedido la prueba se quedará con un resguardo que contendrá la cadena de custodia hasta el momento, de dicha manera, en el caso del extravío o desaparición de una prueba se podrá llegar hasta el responsable último de dicha evidencia.

El segundo tipo de procedimiento es la implementación digital. En este, se abre un expediente tanto en papel como digital, las pruebas cuando son procesadas, se añaden a una base de datos en la cual se podrá ver quien es la persona a cargo de dicha prueba, así como las personas que han entrado en contacto con la prueba o solicitar la interacción con la misma. En este tipo de procedimiento cabe destacar que las implementaciones son múltiples, desde un archivo compartido donde se anotan los cambios, una base de datos compartida o alguna aplicación propia diseñada para la cadena de custodia.

Una de las mejores implementaciones actuales a nivel español es BINCIPOL, Base de Datos de Inteligencia Científica Policial. En dicha base de datos, se garantiza la cadena de custodia de todos los vestigios y de quién, cómo y dónde se ha recogido y se ha almacenado. Este sistema comunica el Sistema de Denuncias Policiales, Control de Acceso a Usuarios de la Dirección General de Policía, Sistema de identificación por ADN…Además comunica la Dirección General de Policía y de la Guardia Civil. Este sistema está recogido en la orden INT/1202/2011[9]. En este sistema además se asegura el cumplimiento de las distintas recomendaciones emitidas por la Orden de JUS/1291/2010[10], donde también se puede encontrar un documento con el formulario de la cadena

de custodia entre el organismo remitente y el Instituto Nacional de Toxicología y Ciencias Forenses de España.

3.3 Antecedentes

La cadena de custodia, a pesar de ser un proceso requerido, no está regulada con una metodología concreta. A pesar de numerosos recursos que indican como han de ser recogidas y procesadas las pruebas digitales según la RFC 32227, las normas UNE 71505 y 71506 donde se indica como preservar, adquirir, documentar, analizar y presentar pruebas digitales y la orden INT/1202/2011 donde se especifica la preparación y la remisión de pruebas, no existe ninguna regulación concreta ante la metodología o requerimientos necesarios para la implementación de la cadena de custodia.

Por la necesidad de la cadena de custodia en los procesos judiciales, así como por su falta de regulación y disparidad de implementaciones tanto físicas como tecnológicas se propone una solución que analice los distintos sistemas actuales e implemente una mejora en la seguridad del proceso. La necesidad de la mejora del proceso ha quedado expuesta en numerosas ocasiones permitiendo la invalidez de pruebas debido a fisuras en su custodia o la propia desaparición de las pruebas. Una muestra es el caso "*Anonymous"* recogido en el procedimiento judicial Nº385/2015 [11], donde las pruebas obtenidas fueron desestimadas debido a la rotura total de la custodia de las pruebas de manera completa ya que los propios códigos hash de las pruebas digitales no coincidían con los códigos obtenidos al recoger las evidencias.

Ante un importante crecimiento de la tecnología Blockchain, cuyas bases principales son la eliminación de la confianza mediante su sustitución por pruebas criptográficas, la seguridad basada en el consenso y la fiabilidad aportada por una red de nodos distribuidos, se han presentado diferentes ideas sobre la posible sinergia de esta tecnología con la cadena de custodia. Dicha simbiosis puede hacer posible una implementación en la que se aumente en gran medida la seguridad y la fiabilidad del proceso. La idea ya ha sido presentada en diferentes artículos que valoran nuevas posibilidades de este proceso como puede ser *"B-DEC: Digital Evidence Cabinet based on Blockchain for Evidence Management"* [12] donde se exponen las ventajas del sistema de Blockchain para el tratamiento de la cadena de custodia, así como su organización. En *"B-CoC: A Blockchain-based Chain of Custody for Evidences Management in Digital Forensics"* [8], se exponen y estudian la viabilidad de una solución Blockchain analizando la latencia de generación de los bloques, así como la memoria requerida para el proceso y su crecimiento anual.

Capítulo 4
Diseño y desarrollo

En este capitulo se trata el diseño y el desarrollo de una aplicación que implemente la cadena de custodia en pruebas forenses. Durante el capítulo se discuten las decisiones técnicas realizadas, así como las tecnologías empleadas y el proceso de desarrollo de la aplicación.

4.1 Diseño

El diseño de la aplicación se ha basado en el estudio de distintas soluciones como método para la adquisición de las funcionalidades que han de ser requeridas y deseadas por la aplicación. Mediante el análisis de distintos artículos e implementaciones reales como puede ser la aplicación de BINCIPOL y formularios de cadena de custodia, se ha diseñado la aplicación Chain of Custody "CoC".

El diseño de CoC se mantendrá sencillo y minimalista, permitiendo que el tiempo de aprendizaje de los usuarios con la plataforma deba ser mínimo.

4.1.1 Funcionalidades

Al finalizar el análisis detallado de las distintas soluciones como del sistema actual y las posibles implementaciones, la aplicación CoC ha de disponer de las siguientes funcionalidades:

- **Agregación de pruebas:** Se han de poder registrar nuevas pruebas en el sistema. Cada una de las pruebas agregadas tendrá que contener toda la información de la prueba, la cual no podrá ser modificada. Con la creación de una nueva prueba dentro del sistema se tendrá que crear también la cadena de custodia asociada al mismo.
- **Visualización de la información de una prueba:** Se ha de poder visualizar toda la información referente a una prueba, esto permitirá saber que tipo de prueba es y la información recogida sobre ella.
- **Visualización de la cadena de custodia de una prueba:** Se ha de poder visualizar la cadena de custodia de cada una de las pruebas. Acceder a las personas que han tenido acceso y la persona que actualmente custodia la prueba.
- **Cambio de Custodia de una prueba:** Se ha de poder cambiar la persona que mantiene bajo custodia la prueba. Se ha de garantizar el acceso exclusivo a la prueba de manera que solo dispondrá de una persona que la custodie y será obligatorio que la prueba siempre esté bajo custodia.

En la búsqueda de una estructuración de la información más clara, así como para la visualización más fácil por parte del usuario, la aplicación implementará un sistema de casos. Las pruebas que se añadan a la aplicación tendrán que añadirse a un caso que tendrá que ser creado previamente y que contendrá todas las pruebas que se le asocien.

Finalmente, para el acceso a la aplicación será necesario una identificación de los usuarios de manera que no cualquier persona pueda tener acceso a la información que se almacena en la cadena.

4.2 Tecnologías

Se ha realizado un análisis de las distintas posibilidades para cada una de las necesidades de la aplicación y las posibles ventajas que puede aportar cada tecnología.

En el desarrollo de la aplicación web se han utilizado las siguientes tecnologías:

- **React**: Librería de JavaScript que permite la creación de aplicaciones web basadas en componentes reactivos. Esta librería ha sido utilizada para la creación de toda la aplicación.[13]
- **Next:** Librería de JavaScript que permite la creación de aplicaciones en React que sean renderizadas por el servidor de forma que estén optimizadas para los buscadores. Además, permite la opción de la generación de compilado de la aplicación para el despliegue de la página web estática.[14]
- **Tailwind:** Librería de CSS, esta librería ha sido utilizada como base para el diseño de todos los estilos utilizados por la aplicación. Esta librería de estilos permite una gran personalización y facilita la creación de los estilos para los componentes.[15]
- **Web3:** Librería de JavaScript que permite la interacción con nodos Ethereum remotos o locales. Esta librería se encargará de toda la comunicación de la aplicación entre la aplicación web y la cadena de bloques. [16]
- **Metamask:** Extensión del navegador que permite la conexión con las aplicaciones descentralizadas en Ethereum. Esta extensión inyecta la conexión de web3 permitiendo comunicación de la aplicación con el contexto de la cadena de bloques. [17]

Para el desarrollo del servidor de la aplicación se han utilizado las siguientes tecnologías:

- **Truffle:** Entorno de desarrollo y de las pruebas para cadenas de bloques usando la Máquina Virtual de Ethereum. Esta herramienta se utiliza en el desarrollo para la implementación, prueba y despliegue de los contratos inteligentes. Además, esta herramienta contiene una línea de comandos integrada que permitirá el compilado de contratos, así como la ejecución de scripts. [18]
- **Solidity:** Lenguaje de programación orientado a objetos influenciado por JavaScript y C++ cuya funcionalidad se basa en la escritura e implementación de contratos inteligentes. Este lenguaje se usa para la escritura de todos los contratos que aseguran la cadena de custodia en la aplicación. [19]
- **Ganache:** Herramienta de desarrollo utilizada para crear cadenas de bloques Ethereum y permitir el despliegue de contratos inteligentes. Esta herramienta se utiliza para la creación de la cadena de custodia propia para el despliegue de la aplicación y de la información vertida en la misma. Además, permite el análisis de todas las transacciones, bloques y registros de la cadena creada. [20]

4.3 Desarrollo

La aplicación CoC se ha ido desarrollando de manera paralela en sus dos componentes principales. El primero, la creación de la cadena de bloques y de los contratos inteligentes que han tenido que ser desarrollados para el funcionamiento de la aplicación descentralizada. El segundo elemento corresponde con la aplicación gráfica con la que ha de interactuar el usuario para poder acceder,

modificar e interaccionar con la cadena para poder realizar las funciones habilitadas en el proceso de cadena de custodia.

4.3.1 Contratos inteligentes

El funcionamiento de las aplicaciones descentralizadas está basado en la existencia de contratos inteligentes que contienen el código que ha de ser ejecutado cuando el contrato desplegado es llamado por una transacción. Para el desarrollo de la aplicación ha sido necesaria la creación de dos contratos, el primero controlará los usuarios que entren a la plataforma y los accesos que tengan dichos usuarios. El segundo contrato controlará los casos que se hayan abierto y las pruebas que se hayan registrado. Ambos contratos están desarrollados utilizando la herramienta de Truffle y en el lenguaje propio del desarrollo de contratos en Ethereum, ambos mencionados anteriormente.

El contrato de los usuarios será sencillo y asociará direcciones de cuentas Ethereum con los datos del usuario. Este contrato dispone de los métodos necesarios para poder registrar un nuevo usuario, que será utilizado para poder iniciar sesión en la aplicación descentralizada (dapp), consultar los datos de un usuario, necesario para recuperar la información que está utilizando la aplicación, un método para transformar direcciones Ethereum en nombres de usuarios, disponible para poder visualizar los datos de una manera fácil y comprensible por un usuario.

El contrato para el control de las pruebas conlleva mayor dificultad. El contrato ha sido diseñado para la creación de casos, donde serán creadas cada una de las pruebas y almacenada la cadena de custodia de cada una de esas pruebas. La primera parte del contrato es el caso, que se ha creado teniendo en cuenta un identificador por caso, una descripción, el número de pruebas asignadas y la dirección de la persona que abre el caso, la segunda parte estará basada en cada una de las pruebas de las que se recogerá el id, hash y la descripción. Cada una de las pruebas se almacenará en el caso y estará asignada a un único usuario.

El contrato del caso permitirá la obtención de toda la información del caso que se consulte, la posibilidad de añadir nuevas pruebas, registrando la persona que crea la prueba como persona responsable de su custodia. Para la implementación del cambio de usuario que custodia la prueba se ha tenido en cuenta que ha de existir un consenso para el correcto funcionamiento de la cadena de custodia. Para posibilitar que una prueba cambie la persona responsable de su custodia, un nuevo usuario ha de solicitar la custodia de dicha prueba y el usuario responsable de la prueba en ese momento debe aceptar dicha solicitud. De esta manera se asegura que un usuario no pueda pasar la custodia de una prueba a otro usuario sin el consentimiento de este, así como que no se desee pasar la responsabilidad de la custodia y dicho suceso se realice de manera automática sin consentimiento.

Un elemento fundamental en el desarrollo del contrato de los casos son los eventos. En la aplicación los eventos son utilizados para la realización de búsqueda de distintas ocurrencias de manera rápida y eficiente. Cada uno de los eventos tiene asociada una firma, dirección hexadecimal, que permite la búsqueda y filtrado de las ocurrencias de los eventos en toda la cadena de bloques. Cada uno de los eventos puede contener parámetros que aportan información y que pueden ser indexados. Los eventos por defecto no crean índices en la cadena de bloques con la información que almacenan en sus parámetros, sin embargo, se puede especificar que un parámetro sea considerado como índice para permitir la búsqueda del evento correspondiente con un valor concreto del parámetro que se ha seleccionado como índice, permitiendo así una búsqueda rápida y un filtrado del resultado deseado. El uso de índices aumenta el espacio de almacenamiento de los eventos por lo que su uso se recomienda solo en los parámetros para los que se realicen búsquedas.

4.3.2 Aplicación Web

El proyecto de la aplicación web ha sido realizado a partir de la plantilla propia generada por la librería de Next mencionada anteriormente. Además, toda la aplicación ha sido creada por componentes siguiendo el estilo de desarrollo de componentes de React. Los estilos visuales han sido añadidos usando la librería de Tailwind CSS por su facilidad de uso, baja curva de aprendizaje y su gran capacidad de personalización.

Para el desarrollo de la aplicación se han debido tener en cuenta distintas consideraciones para el buen funcionamiento de la aplicación. La primera, los usuarios han de tener una cuenta, cualquier usuario que consiga entrar a la plataforma ha de estar identificado.

La identificación se realizará mediante el contrato implementado de los usuarios, para poder acceder, el usuario debe tener una cuenta que esté almacenada en el contrato. En el caso de que el usuario que esté intentando acceder no tenga una cuenta registrada tendrá que registrar la cuenta de la dirección con la que se conecta. Para posibilitar la conexión con la blockchain, se ha de iniciar sesión en la extensión de metamask y seleccionar una cuenta con la que acceder a la cadena. En el caso de que la configuración de metamask no haya sido realizada no se podrá conectar con la aplicación, si ha sido correcta, el usuario habrá tenido que acceder con su clave privada por lo que será el responsable de la cuenta.

La segunda consideración es el acceso a la blockchain. Para el despliegue de los contratos, así como para cualquier funcionalidad de la aplicación se tendrá que conectar con la cadena de bloques para recuperar la información que esta tiene almacenada. Para posibilitar esta conexión se utiliza la librería de Web3 que se configura al acceder a la aplicación, y estará disponible desde todos los componentes envolviendo todos estos con un provider.

La obtención de los datos se realiza en el momento de la carga del componente que necesita la información llamando por medio de la librería de web3 a la cadena de bloques. Esto se realiza llamando al contrato mediante la función *'call()'* ya que los métodos de lectura de la cadena de bloques no conllevan gasto de gas. Por otro lado, todas las interacciones con la cadena de bloques que resulte en un cambio de estado de la cadena conllevan una escritura en la cadena de bloques y el cambio de estado de esta por lo que consumirá gas y será necesario enviar los fondos en la llamada. Para este tipo de interacciones se ha usado el método *'send()'* que permite el envío de gas necesario para la ejecución de los procesos requeridos por el contrato.

La tercera consideración que se ha de tener en cuenta son los fondos. Para la realización de funcionalidades básicas para la aplicación como puede ser la creación de una nueva prueba, se necesita la escritura de dicha información en la cadena de bloques. Como se ha explicado anteriormente, toda escritura conlleva una transacción que ha contener el gas necesario para asegurar la ejecución del contrato y el éxito de la transacción. Por ello, los usuarios que hayan de cambiar el estado de la cadena, tanto para crear un caso o una prueba, o para solicitar o conceder la custodia de una evidencia, así como para registrarse en la plataforma, han de tener gas suficiente para que éste sea enviado a la función.

4.3.3 Descripción CoC

El acceso a la plataforma CoC se realizará con la ayuda de la extensión de Metamask. En primer lugar, la aplicación buscará la red para conectarse a la cadena, para ello será necesario que la extensión de metamask esté instalada y desbloqueada mediante la ventana emergente de la figura 4.1. En segundo lugar, será necesario que el usuario inicie sesión con su clave privada dentro de la extensión y autorice a la aplicación a acceder a su cuenta como se puede ver en la figura 4.2. Una vez se hayan realizado estas operaciones la aplicación mostrará una ventana donde se permite el registro del usuario en la plataforma en el caso de que no haya sido registrado con anterioridad como se puede ver en la figura 4.3, o se le redirecciona a la página principal de la aplicación si se detecta que el usuario ya ha sido registrado.

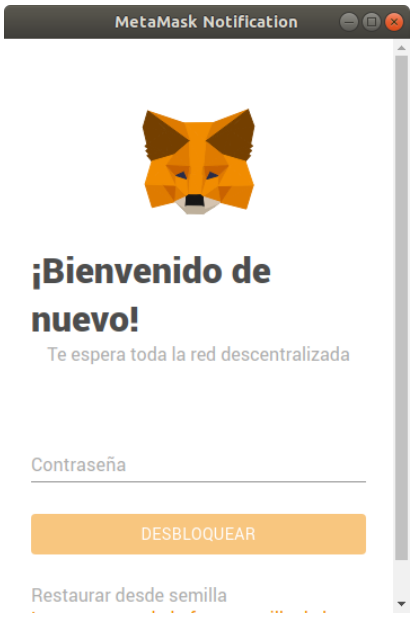

Ilustración 4.1: *Desbloqueo de Metamask*

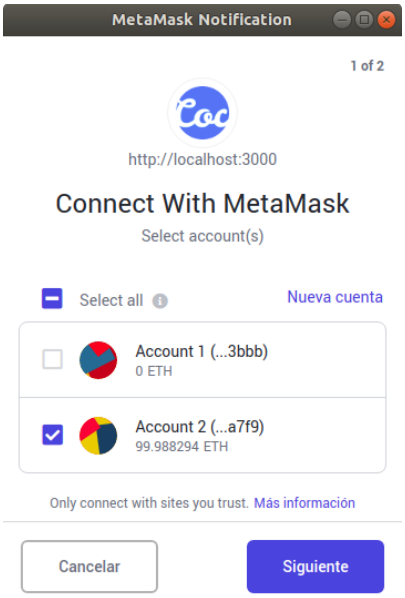

Ilustración 4.2: *Selección de cuenta*

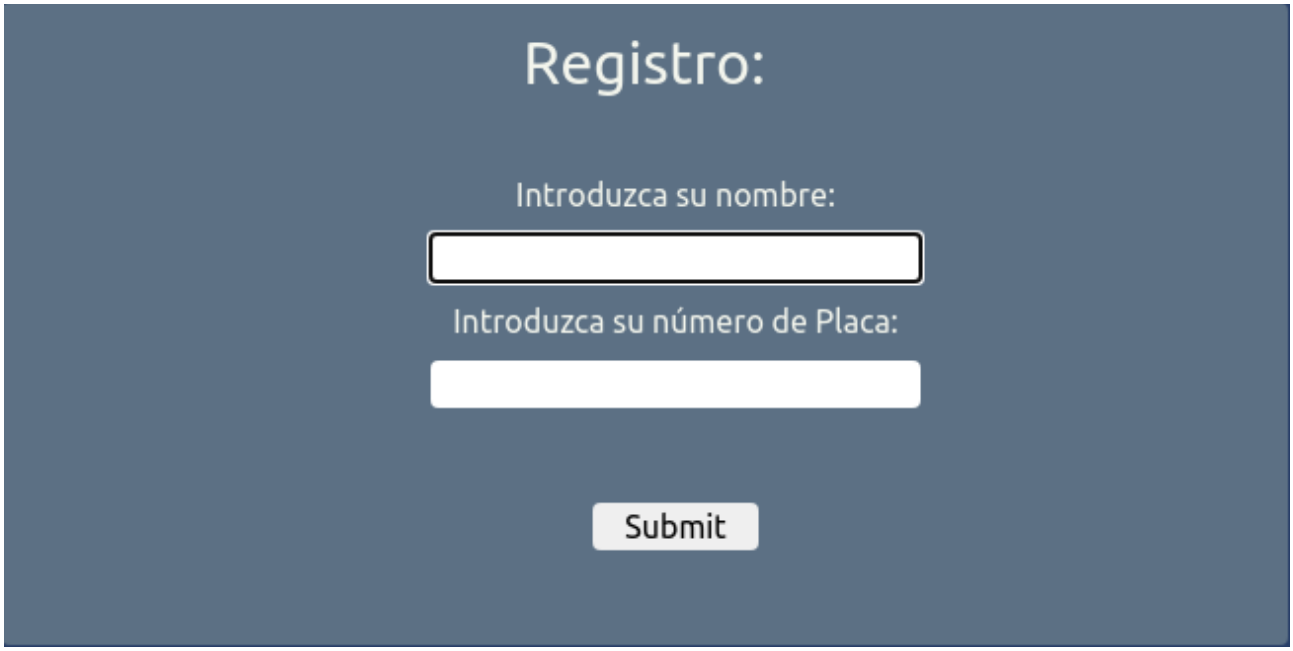

Ilustración 4.3: *Registro de la plataforma*

La aplicación CoC, como se puede ver en la figura 4.4 donde se muestra la página principal, está compuesta por una barra superior donde se visualiza el usuario con el que se encuentra identificado en la aplicación y la posibilidad de desconectar dicho usuario. Un menú lateral donde se permite un acceso rápido a las dos funcionalidades principales de la aplicación, la visualización de todos los casos abiertos y la visualización de todas las pruebas que han sido registradas.

Ilustración 4.4: *Página principal*

Desde la vista de casos, observable en la figura 4.5, se permite un filtrado para la rápida de localización del caso en el que se ha de trabajar. Por otro lado, mediante la utilización de botones se permite la apertura de un formulario para la creación de un nuevo caso y para la visualización de todos los casos. En un caso de uso, un policía a cargo de una investigación abriría un caso desde esta pestaña mostrando así el modal que se puede observar en la figura 4.6. Desde dicho modal se rellenará toda la descripción con la información referente al caso que se considere necesaria y se pulsará en el botón de añadir que lanzará una transacción que ha de ser aprobada desde metamask para la agregación del caso a la cadena de bloques.

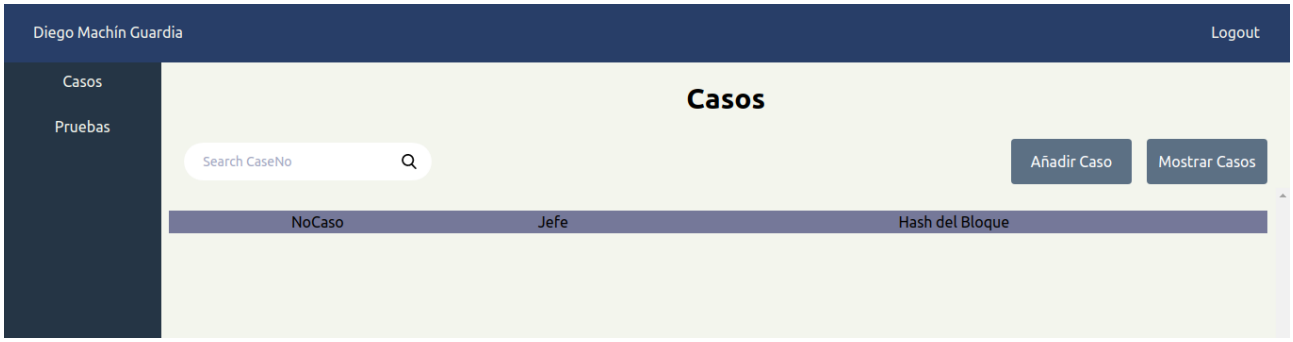

Ilustración 4.5: *Página de casos*

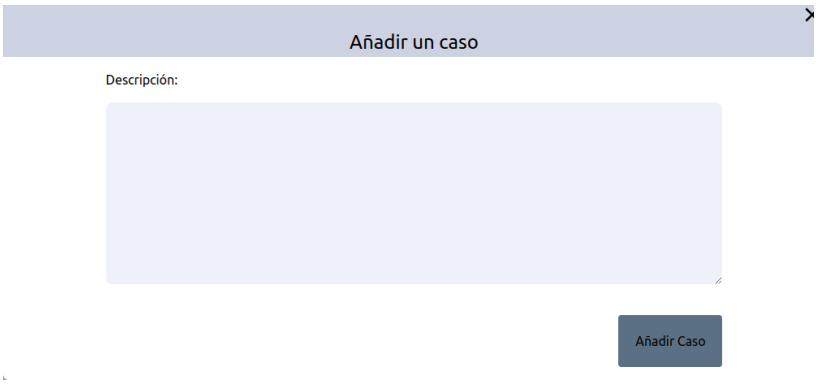

Ilustración 4.6: *Modal para crear un caso*

Desde la vista de evidencias, observable en la figura 4.7, se permite la visualización de todas las evidencias que están registradas en la plataforma para permitir el acceso a la información sobre la misma.

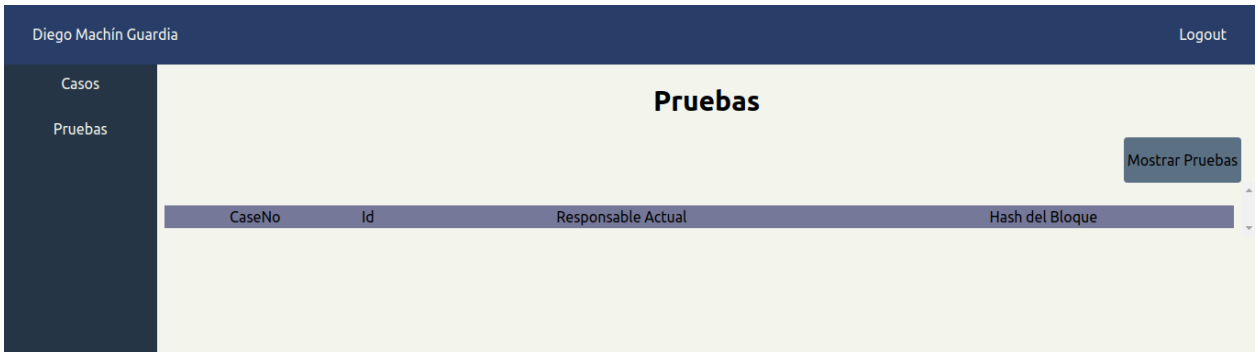

Ilustración 4.7: *Página de evidencias*

La visualización del detalle tanto de los casos como de las evidencias se realizan mediante la apertura de un modal personalizado donde se muestra la información necesaria sobre el objeto que se visualiza, así como las posibles acciones que se pueden realizar.

En el modal del caso, visible en la figura 4.8, se muestra la información referente al mismo, el número identificador del caso, el número de pruebas que dependen de dicho caso, la descripción de este, la persona que abrió el caso y un listado con todas las pruebas que depende del caso con sus identificadores y sus responsables. Además, desde el mismo se permite acceder al formulario para añadir una nueva prueba. Un ejemplo de uso será pulsar sobre la entrada en la tabla de los casos que se ha creado en el modal para añadir un caso mencionado anteriormente, dicha acción abrirá el siguiente modal y permitirá la visualización completa del caso. Además, desde esta vista se permite la adición de nuevas pruebas al caso desde un botón en la parte inferior izquierda que abre el modal del formulario de nueva prueba que se puede observar en la figura 4.9. En dicho modal se permite la introducción de una descripción de la prueba a añadir y la posibilidad de selección de prueba digital o física, debiendo añadir un hash en el caso de que la evidencia sea de tipo digital.

Ilustración 4.8: *Modal del caso*

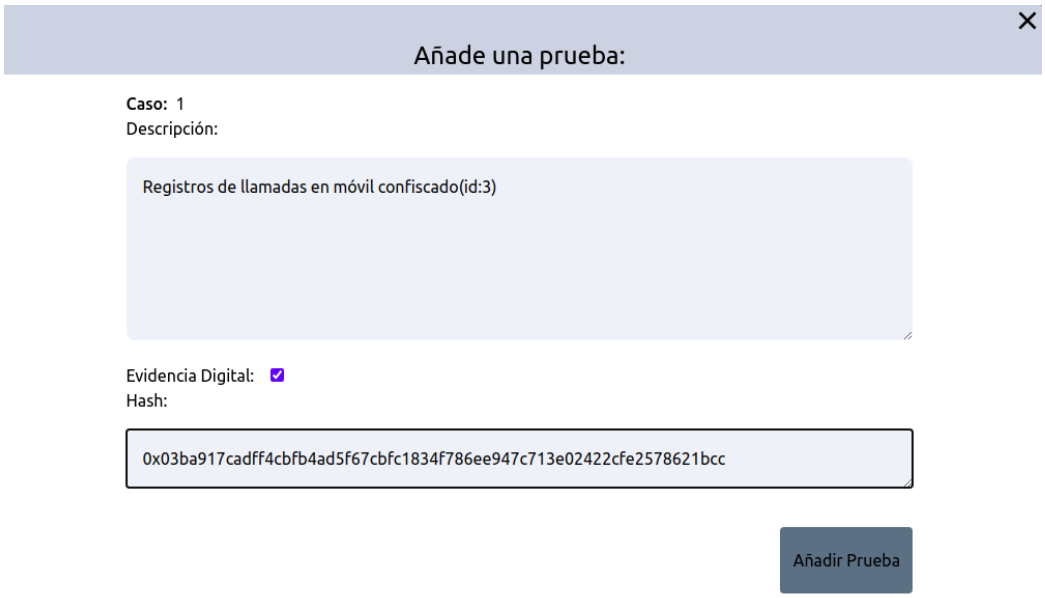

Ilustración 4.9: *Modal creación de prueba*

En el modal de la prueba, observable en la figura 4.10, se muestra la información referente a la misma, el número identificador de la prueba, el caso al que se encuentra asociada, la descripción aportada, el tipo de prueba recogida que puede ser digital o física, en el caso de que la prueba sea digital se mostrará también el hash de la misma, la persona responsable en el momento actual de la custodia de dicha prueba y finalmente una lista con la cadena de custodia donde se recogen todas las personas que han sido las responsables de la custodia de dicha prueba y la fecha y hora a la que empezó dicha custodia. En el caso de que la prueba haya sido solicitada por otra persona también se mostrará un apartado donde se visualiza la siguiente persona a la que se ha de pasar la prueba. Las acciones disponibles para la evidencia dependerán del estado de esta, en el caso de que la persona que visualice la prueba no sea la persona responsable de la custodia de la prueba en el momento actual se permite la solicitud de custodia, y en el caso de que ya se haya solicitado se mostrará un aviso de que ya ha sido solicitada. Por otro lado, en el caso de que la persona que visualice la prueba

corresponda con la persona responsable de la custodia se permite la sesión de la custodia de la prueba a la siguiente persona o la denegación de la solicitud como se puede observar en la ilustración 4.11.

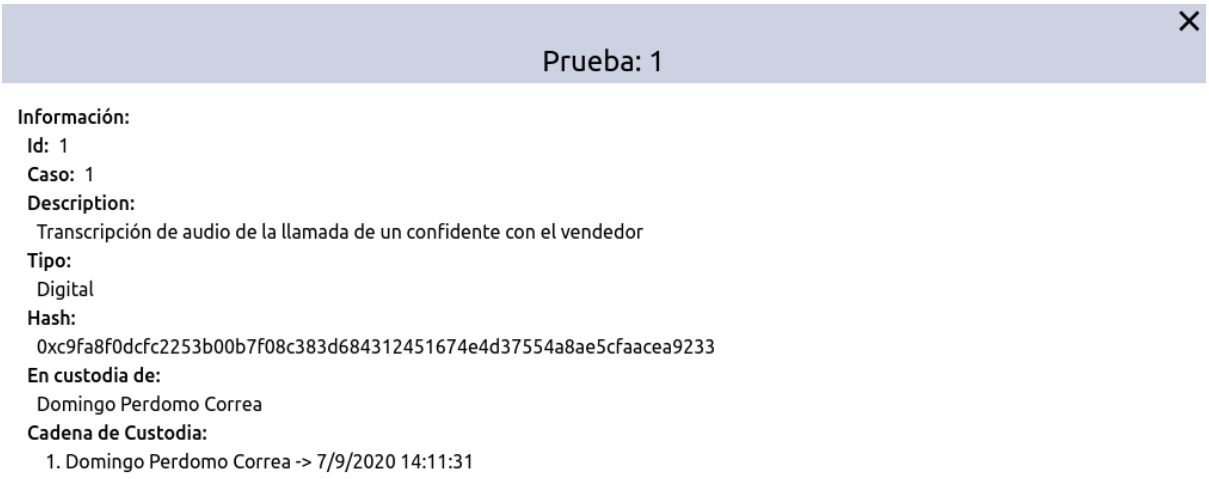

Ilustración 4.10: *Modal de la evidencia*

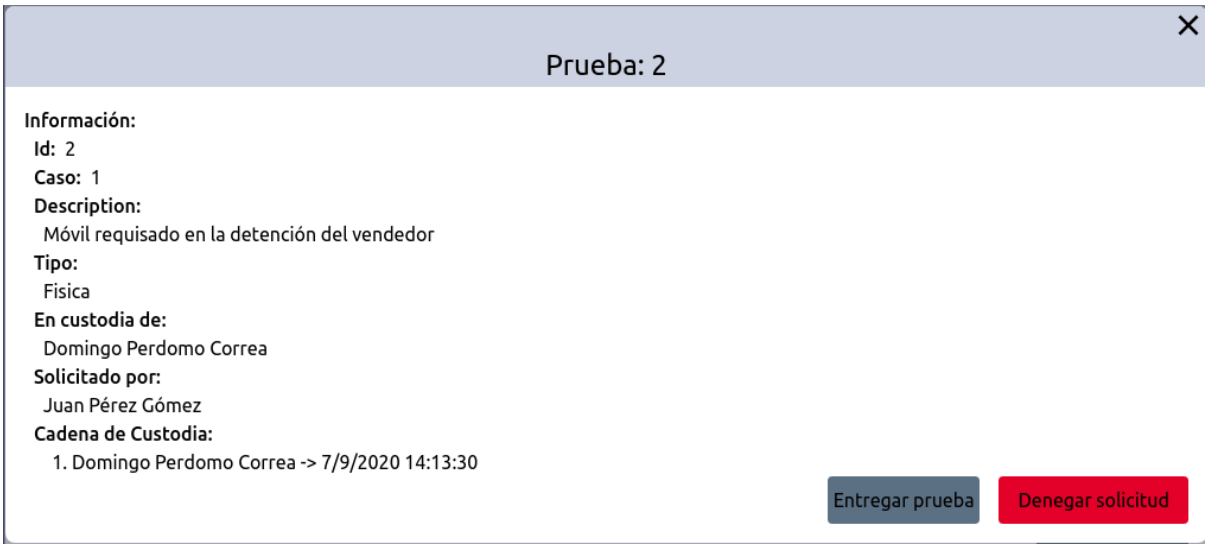

Ilustración 4.11: *Modal en caso de ser responsable de custodia*

Para la realización del intercambio de la custodia de una prueba entre dos usuarios, según lo explicado anteriormente, el usuario que quiere conseguir la custodia de la prueba deberá acceder al modal de la prueba que le gustaría inspeccionar. A continuación, será necesario que ese usuario seleccione en el modal de la prueba el botón de solicitar prueba que estará disponible. Una vez realizada esta acción, el usuario responsable de la custodia de la prueba en el momento deberá acceder a la prueba y aceptar la solicitud realizada por la otra persona mediante el botón de entregar prueba. Una vez realizado dicho proceso, la prueba cambiará la persona responsable de su custodia y se añadirá una nueva línea a la cadena de custodia donde se mostrará la nueva persona responsable de la prueba, así como a la hora a la que ha sido entregada, y a la persona que tenía antiguamente la custodia de la prueba se le mostrará otra vez el botón para solicitar la custodia de esta, como se puede ver en la figura 4.12. La cadena de custodia será siempre visible dentro de la prueba y registrará todas las personas que tengan su custodia.

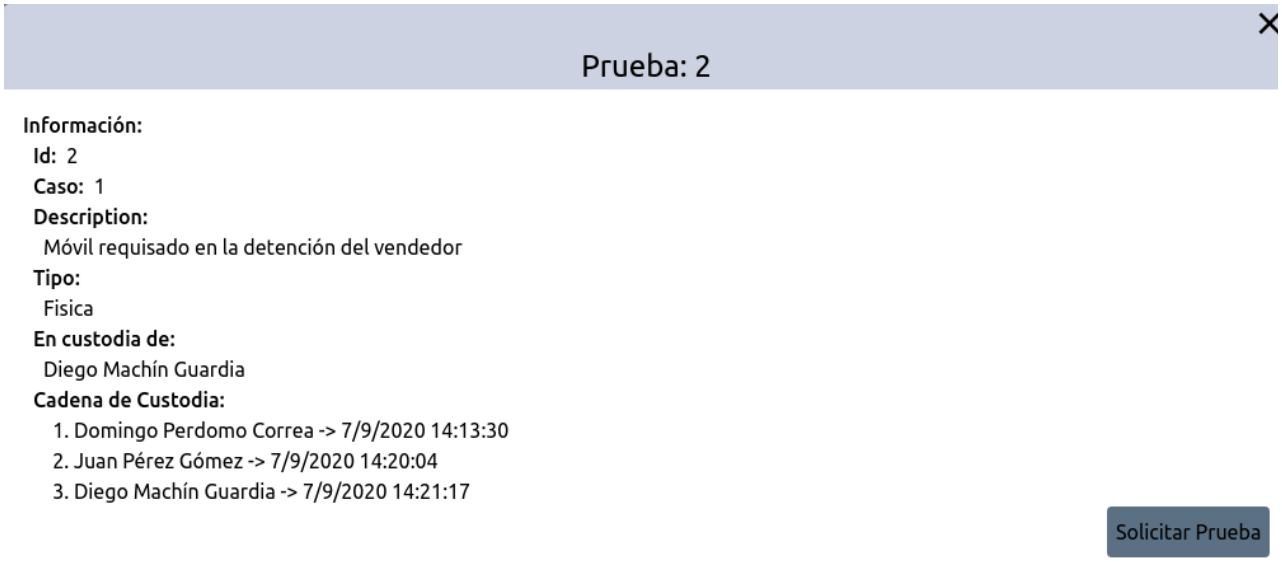

Ilustración 4.12: *Solicitud de custodia*

Para la realización de operaciones en la aplicación que no sean de consulta y que impliquen una modificación del estado de la cadena, como puede ser el registro de un usuario, la creación de un nuevo caso o de una nueva prueba o la realización o aceptación de una solicitud será necesario la creación de una transacción. Para la realización de transacciones la extensión de metamask detectará la solicitud de la transacción y abrirá una ventana emergente que el usuario deberá de confirmar para poder realizar la operación que ha solicitado. En ella se deberá de confirmar la transacción para el éxito de la operación.

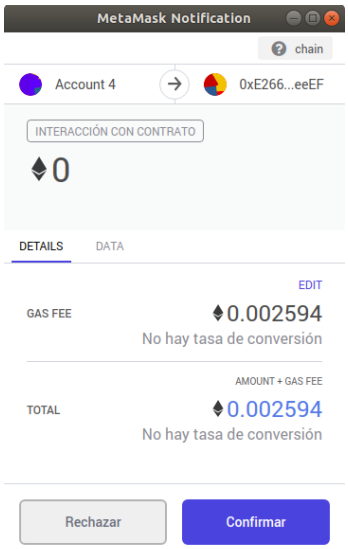

Ilustración 4.13: *Transacción Metamask*

Finalmente se permitirá la opción para cerrar sesión, esto reiniciará el estado de la aplicación y permitirá al usuario la redirección de nuevo a la aplicación. En el momento que se vuelva a iniciar la aplicación se volverá a cargar la información disponible desde la extensión de metamask, por ello, se ha de cambiar o cerrar sesión en metamask de manera paralela para que al inicio de la aplicación no se vuelva a seleccionar la cuenta que ya se encontraba seleccionada.

4.3.4 Implementaciones principales

En esta sección se muestran algunos fragmentos de código más relevantes para el funcionamiento de la aplicación que permite una mejor comprensión de la aplicación y de la interacción con la cadena de bloques.

En primer lugar, se mostrarán las distintas funciones y objetos más relevantes de los contratos inteligentes. Estos elementos están escritos en el lenguaje propio de los contratos inteligentes, Solidity. Las funciones han de definir su nombre, así como los parámetros con los que se invoca la función, los tipos de retorno y el tipo de acceso que ejecuta, así como la visibilidad de la función. Las funciones en Solidity tienen tres tipos en función de su interacción con la cadena, las funciones etiquetadas como *'view'* son aquellas que no modifican la información de la cadena por lo que no necesitan de una transacción ni gas para poder ser invocadas. Las funciones etiquetadas *'pure'* son aquellas que no acceden al estado de la cadena por lo que simplemente realizan operaciones con los parámetros requeridos, estas funciones tampoco necesitan gas para poder ser invocadas. Finalmente, las funciones etiquetadas con *'payable'* son aquellas que necesitan gas para operar y permiten la modificación del estado de la cadena de bloques para su operación.

1. Las evidencias, contiene la información del identificador, hash en el caso de que la prueba sea digital, descripción, responsable de registro, el número de solicitudes para la custodia de la prueba y un mapa que almacena las direcciones de estos solicitantes.

```
struct Evidence {
    uint256 id;
    bytes32 hashed;
    string description;
    address keeper;
    uint256 requests;
    mapping(uint256 => address) requesters;
}
```

***Ilustración 4.14**: Estructura de la prueba*

2. Para añadir una prueba se tendrá que añadir en el contrato de los casos un nuevo objeto evidencia que se inserta en un mapa donde se almacenan todas las pruebas asociadas a ese contrato. Además, cabe mencionar que la identificación de cada una de las pruebas se realiza de manera automática en función de la última prueba que haya sido insertada en el caso. Finalmente, para posibilitar todas las funcionalidades de la aplicación se han de emitir distintos eventos, en el caso de que se registre una prueba será necesario registrar los eventos de 'evidence' y 'newKeeper' para que la aplicación pueda controlar estas adiciones a su estado previo.

```
    function addEvidence(uint256 caseId, string memory newDescription,
        bytes32 newHash) public payable returns (uint256) {
        require(caseId < caseCount, "There is no such case");
        Evidence memory proof = Evidence(
            cases[caseId].evidenceCount,
            newHash,
            newDescription,
            msg.sender,
            0
        );
        cases[caseId].evidences[cases[caseId].evidenceCount] = proof;
        cases[caseId].evidenceCount++;
        emit evidence(caseId, proof.id, msg.sender);
        emit newKeeper(caseId, proof.id, msg.sender);
    }
```

Ilustración 4.15: *Función para añadir evidencia*

3. La función para solicitar ser el responsable de la custodia de una prueba modifica el estado de la cadena de bloques por lo que ha de tener una transacción asociada y ha de estar etiquetada como 'payable'. Para asegurar el éxito de la operación se tienen que cumplir los requisitos especificados dentro de las funciones 'require', la prueba que se solicita tiene que existir, la persona solicitante de la prueba no puede ser la responsable en el momento de su custodia y la persona que solicite la custodia no puede tener más de una solicitud activa en el momento. Una vez las solicitudes sean aceptadas se eliminará la solicitud del mapeado de las solicitudes pendientes. Finalmente, siguiendo el esquema de las demás funciones se emitirá el evento de solicitud de la prueba.

```
function requestEvidence(uint256 caseId, uint256 requestedEvidence)
        public payable {
        require(caseId < caseCount, "No such case");
        require(
            requestedEvidence < cases[caseId].evidenceCount,
            "There is no such evidence"
        );
        require(
            msg.sender != cases[caseId].evidences[requestedEvidence].keeper,
            "You are already the keeper"
        );
        bool requested = hasRequested(caseId, requestedEvidence, msg.sender);
        require(!requested, "Already requested");
        cases[caseId].evidences[requestedEvidence].requesters[cases[caseId]
            .evidences[requestedEvidence]
            .requests] = msg.sender;
        cases[caseId].evidences[requestedEvidence].requests++;
        emit request(
            msg.sender,
            cases[caseId].evidences[requestedEvidence].keeper,
            requestedEvidence,
            caseId
        );
    }
```

Ilustración 4.16: *Función para solicitar evidencia*

Por otro lado, la aplicación de CoC, esta aplicación como ya se ha mencionado anteriormente se ha escrito usando la librería de React. A continuación, se mencionan las principales funciones que permiten su funcionamiento:

Conexión con Metamask, en esta función se realizará la primera comunicación con la cadena de bloques. Gracias a la interacción con la extensión se podrá acceder a la cuenta del usuario permitiendo la realización las transacciones con la cadena. Esta conexión se realiza con el enlace de la librería de Web3 con Metamask. Además, una vez realizada la conexión se inicializará el contrato de los usuarios y el contrato de los casos que nos permitirá tener tanto la información del usuario, los casos y el objeto de web3 y el contrato específico de los usuarios disponibles en toda la aplicación por medio de la inicialización del estado de un 'provider' que será común a toda la aplicación.

```
async localBlockchainData() {
    const web3Init = new Web3(window.ethereum)
    await window.ethereum.enable().catch((error) => console.log(error))
    const accounts = await web3Init.eth.getAccounts()
    var USER_BLOCKCHAIN_ADDRESS = ''
    var USER_BLOCKCHAIN_ABI = []
    var CASE_BLOCKCHAIN_ADDRESS = ''
    var CASE_BLOCKCHAIN_ABI = []
    await $.getJSON('/contracts/Users.json', function (data) {
      USER_BLOCKCHAIN_ABI = data.abi
      USER_BLOCKCHAIN_ADDRESS = data.networks['5777'].address
      return true
    })
    await $.getJSON('/contracts/Chain.json', function (data) {
      CASE_BLOCKCHAIN_ABI = data.abi
      CASE_BLOCKCHAIN_ADDRESS = data.networks['5777'].address
      return true
    })
    const Users = new web3Init.eth.Contract(
      USER_BLOCKCHAIN_ABI,
      USER_BLOCKCHAIN_ADDRESS
    )
    const Evidences = new web3Init.eth.Contract(
      CASE_BLOCKCHAIN_ABI,
      CASE_BLOCKCHAIN_ADDRESS
    )
    this.context.setValues(web3Init, Users, Evidences, null)
    const name = await this.context.web3.userContract.methods
      .userData()
      .call({ from: accounts[0] })
    console.log(this.context.web3.caseContract)
    this.context.setValues(null, null, null, {
      id: name[0],
      name: name[1],
      account: accounts[0],
    })
  }
```

Ilustración 4.17: *Conexión con la Cadena de bloques*

Para la obtención de todas las pruebas relacionadas con un caso, de todos los casos o de un caso en concreto se utilizan los registros. Los registros permiten acceder a la cadena de bloques y hacer búsqueda de cada uno de los eventos lanzados en cualquier momento dentro de cualquier contrato de la cadena de bloques. Esto posibilita la búsqueda sencilla y eficiente de todos los eventos de un tipo y el filtrado de la información para la recuperación del estado de la cadena de bloques en el momento que el evento fue lanzado. Se muestra a continuación la primera parte de la función encargada de

recuperar todos los casos, para ello se obtienen todos los registros de los eventos que han sido emitidos desde el bloque inicial, en la sección de *"topics"* se ha de especificar la firma del evento que se busca, en este caso, el evento *'newCase'* que usa su firma especificada por el cifrado de *'newCase(uint256 indexed, address)'*.
Como el primer argumento es indexado en la búsqueda de la siguiente función se ha de incluir un segundo *"topic"*, en el caso de que se deseen todos los valores, se dejará a nulo y en el caso de que se desee filtrar por el valor del id del caso se especificará codificado el valor de dicho parámetro.

```
getCase(search = null) {
  //Topic de los casos
  this.context.web3.web3.eth
    .getPastLogs({
      fromBlock: 0,
      toBlock: 'latest',
      topics: [
        '0x267df8e01bdd6250522178053eaf57c85929dbaa1861b164d913d5ff4fc00506',
        search
          ? this.context.web3.web3.eth.abi.encodeParameter('uint256', search)
          : null,
      ],
    })
```

Ilustración 4.18 *Función para obtener los casos*

Una vez recibidos los datos, estos han de ser decodificados utilizando la firma que se ha utilizado para su codificación, por ello, se ha de especificar los parámetros esperados en el cuerpo de los datos sin incluir los datos indexados, esto es visible en la segunda parte de la función anterior, que se puede observar en la siguiente figura:

```
.then((result) => {
  this.setState({
    evidences: result.map((pcase) => {
      let element = this.context.web3.web3.eth.abi.decodeLog(
        [{ type: 'address', name: 'chief' }],
        pcase.data
      )
      return {
        id: pcase.topics[1]
          ? this.context.web3.web3.eth.abi.decodeParameter(
              'uint256',
              pcase.topics[1]
            )
          : 0,
        chief: element.chief,
        contract: pcase.address,
      }
    }),
  })
})
```

Ilustración 4.19 *Filtrado de los casos*

Finalmente, se muestra la función que permite la visualización de la cadena de custodia, esta función al igual que la anterior busca por eventos, en este caso filtra solo los eventos relacionados con el contrato del caso que se consulte. En el momento de la decodificación de los datos de cada una de las pruebas se accede al hash del bloque donde se encuentra dicha transacción y se busca dicho

bloque y la hora a la que ha sido minado para poder relacionar el momento en el que la transacción fue añadida a la cadena para visualizar la hora a la que el cambio del responsable de la cadena de custodia ha cambiado.

```
.then(async (result) => {
        this.setState({
          custody: await Promise.all(
            result.map(async (proof) => {
              let block = await this.context.web3.web3.eth.getBlock(
                proof.blockHash
              )
              return {
                keeper: proof.returnValues.newKeeper,
                timestamp: block.timestamp,
              }
            })
          ),
        })
      })
```

Ilustración 4.20: *Función para obtener la cadena de custodia*

4.3.5 Registros o Datos

Los eventos son una manera eficiente de acceder a los registros de los que dispone la EVM. Estos registros estarán asociados al contrato que los ha emitido y vivirán en el bloque de la transacción donde hayan sido emitidos mientras el bloque esté disponible, hasta el momento, siempre que continúe existiendo la cadena de bloques.

Los datos son almacenados en el estado del contrato en el que se crean, este almacenamiento, al igual que la emisión de los eventos, consume gas para poder almacenar la información. Estos datos serán accesibles o no en función de la visibilidad que se les haya definido, por otro lado, el acceso a estos datos será siempre a través del acceso al estado del contrato donde reside la información. Para el almacenamiento de conjunto de objetos, predominan los vectores y los mapas, el acceso a todos los elementos de un vector o de un mapa son altamente complicados y en el caso de los mapas no es posible ya que haya que especificar el elemento al que se desea acceder. Esto dificulta acceder a toda la información almacenada en un mapa en el caso de que no se sepan todos los índices de este.

Debido a la dificultad del acceso a todos los elementos almacenados como conjuntos de datos en un contrato y por la necesidad de existir en el mismo contrato se ha optado por el uso mixto de información almacenada en los contratos con información almacenada en los registros de la cadena. De esta manera se puede hacer una búsqueda rápida de todos los eventos de un tipo sin saber todos los índices o hacer múltiples llamadas y asegurando el acceso a toda la información independientemente del contrato en el que se encuentre la información. Por ello se ha seguido el enfoque de emitir en la aplicación los eventos de *'evidence', 'newCase'* y *'newKeeper'* permitiendo así el acceso a todas las pruebas y a todos los casos en una búsqueda sencilla, así como un registro de los usuarios responsables de la custodia de una prueba y en el momento que se hicieron cargo de dicha responsabilidad.

4.3.6 Actualizaciones de Metamask

Durante la implementación de CoC se ha debido tener en cuenta distintos cambios en las librerías usadas que son de gran importancia para el funcionamiento de la aplicación. Metamask ha lanzado distintos cambios que conllevan el desuso de distintas funciones que han sido implementadas en primeras iteraciones de la aplicación. Además, otras funcionalidades dejarán de ser soportadas a finales de 2020 por lo que se ha tenido que proceder a la modificación de las distintas funciones para adaptar el código a las nuevas características de las nuevas versiones de Metamask.

Los cambios más importantes que se han tenido en cuenta han sido el cambio de la activación del acceso por parte del usuario a la cuenta almacenada en Metamask, antes permitido con la función *'.isEnable()'* y ahora mediante *'.request({method: "eth_requestAccounts"})'*. El otro cambio de gran importancia es debido a que la extensión dejara de insertar el objeto de web3 dentro de la aplicación, por ello, los desarrolladores serán los encargados de seleccionar, incluir e inicializar la librería que necesiten para la comunicación con su aplicación. Estos cambios se deben a distintos cambios mayores en las nuevas versiones de web3 por lo que gracias a los nuevos cambios de metamask se incluirá compatibilidad entre las distintas versiones de web3 dejando al desarrollador la selección de la versión.

Capítulo 5
Pruebas y análisis

En este capítulo se procede a describir los problemas encontrados durante la realización de la aplicación, las pruebas que se han realizado y un análisis de la solución creada.

5.1 Pruebas

En la realización de CoC se han utilizado las librerías de 'mocha' y 'chai' para la creación de tests que permitan probar el correcto funcionamiento de cada una de las funciones de los contratos y la obtención de los resultados obtenidos.

Debido a la comunicación por medio de Metamask entre la aplicación y la cadena de bloques se hace altamente complicado la depuración de los contratos inteligentes. Uno de los mayores inconvenientes es la difícil depuración, en la realización de una transacción en el caso de que dicha operación resulte fallida Metamask mostrará un error comunicando que no se ha realizado correctamente la operación, pero no detalla los problemas por los que esta operación ha fallado. Esto también dificulta encontrar el error ya que no se especifica si es un problema de codificación de la transacción en la aplicación o un fallo en el procesado de la información en el contrato, por ello los tests se convierten en una herramienta fundamental en el desarrollo de los contratos. En la ilustración 5.1 se puede observar el test para la comprobación de la correcta adición de una prueba a un caso y su correcta recuperación.

```
it('Should add evidence to a case', async () => {
  await Chain.deployed().then(async (instance) => {
    await instance.addEvidence(
      0,
      'new gun 9mm',
      '0x0b894166d3336435c800bea36ff21b29eaa801a52f584c006c49289a0dcf6e2f',
      { from: accounts[0] }
    )
    let evidence = await instance.getEvidence(0, 0)
    assert.equal(evidence[0], 0)
    assert.equal(evidence[1], 0)
    assert.equal(evidence[2], 'new gun 9mm')
    assert.equal(
      evidence[3],
      '0x0b894166d3336435c800bea36ff21b29eaa801a52f584c006c49289a0dcf6e2f'
    )
    assert.equal(evidence[4], accounts[0])
  })
})
```

Ilustración 5.1: *Test para añadir pruebas*

En la consola de visualización que ofrece Ganache donde se muestran los registros de la cadena de bloques tampoco se recoge información de las transacciones que no son incluidas en la cadena por lo que no quedan registrados los fallos. Estos dos inconvenientes hacen que el desarrollo de tests sea imprescindible para asegurar el buen funcionamiento de la aplicación, así como para ahorrar tiempo en el desarrollo de los contratos.

Para la implementación de las pruebas se ha usado la librería de *'truffle-assertions'* [21] que permite la prueba y el control de las distintas excepciones. La librería permite atrapar las excepciones de una transacción permitiendo las pruebas para que las transacciones devuelvan la excepción correcta y el resultado esperado cuando la transacción es correcta. Se comprueba que en el caso de que la persona sea el responsable de la custodia se lance una excepción y que falle la transacción ya que ya es la persona responsable de la cadena de custodia de dicha prueba.

```
it('Should request a evidence', async () => {
  await Chain.deployed().then(async (instance) => {
    await truffleAssert.fails(
      instance.requestEvidence(0, 0, {
        from: accounts[0],
        gas: 1500000,
        gasPrice: '30000000000',
      })
    )
    await truffleAssert.passes(
      instance.requestEvidence(0, 0, {
        from: accounts[1],
        gas: 1500000,
        gasPrice: '30000000000',
      })
    )
  })
})
```

Ilustración 5.2: *Test de solicitud de custodia*

5.2 Problemas encontrados

En el desarrollo de este proyecto se han encontrado las siguientes dificultades:

1. La documentación no es precisa en la obtención del tipo de atributos que se necesitan. En la mayoría de los apartados se especifica el tipo de parámetro que ha de ser incluido, siendo habitualmente una dirección hexadecimal, sin embargo, no se especifica como obtener dicha dirección o como puede ser calculada. Además, debido a ser una plataforma relativamente nueva en un sector en crecimiento, pero con una comunidad que todavía está en desarrollo, el soporte y el conocimiento compartido sobre implementaciones es todavía escaso.
2. Limitaciones de lenguaje, Solidity, es un lenguaje específico relativamente nuevo y no dispone de todos los tipos de datos o funcionalidades de otros lenguajes por lo que dificulta la escritura de los contratos.
3. Actualizaciones recientes, tanto la librería de Web3 como la extensión de Metamask están en un proceso de actualización que ha forzado el cambio del código a lo largo del desarrollo de la aplicación.
4. Dificultad de depuración, ante un fallo por parte de la aplicación, en especial de las transacciones, se dificulta en exceso el descubrimiento del fallo debido a la poca información aportada por las librerías a la hora de depurar. Se ofrece todo el contenido de la transacción en hexadecimal, pero no se puede comprobar el punto de la ejecución donde se detuvo la transacción.

5.3 Análisis de CoC

La aplicación se ha desarrollado en aras del análisis de la solución Blockchain para la cadena de custodia, en base a la solución aportada se pueden analizar los siguientes aspectos.

5.3.1 Ventajas

El uso de una solución como CoC facilita en gran manera el acceso a la información. Cualquiera de las personas implicadas en el proceso tendrá acceso a la cadena de custodia y podrá ver de una manera clara y sencilla el estado de la prueba en cada momento del tiempo desde que la prueba ha sido recogida. Esto permite un acceso a la cadena de custodia con menor burocracia y la eliminación de un control físico de la cadena de custodia.

Con una solución Blockchain se mejora respecto a una aplicación de cadena de custodia como la implementada, permitirá el no repudio de la información asegurando que todas las acciones de cambio en la cadena de custodia quedan registradas en todos los nodos de la red.

También se mejora en gran medida la seguridad en el mantenimiento de la cadena de custodia, como consecuencia a la eliminación de la cadena de custodia física y su implementación digital el sistema siempre tendrá la cadena de custodia de cada una de las pruebas sin depender de la pérdida de un elemento físico por parte del responsable de la custodia de la prueba. Debido a la propia estructura de las aplicaciones descentralizadas la información estará replicada en varios nodos de la red por lo que además de siempre estar la información disponible en un sistema, la información estará duplicada y disponible en múltiples sistemas, asegurando la disponibilidad de la información incluso ante la pérdida de un sistema o nodo.

El uso de esta herramienta puede seguir siendo desarrollado para su combinación con el control de acceso a las pruebas, con dicha combinación se automatizaría el proceso completamente y se aseguraría el tipo de acceso que se tiene a dicha prueba.

5.3.2 Seguridad

Las implementaciones de seguridad de la aplicación están basadas en las propias características de la cadena de bloques. Estas son las siguientes:

- No repudio: un nodo no puede negar la veracidad de la información que ha transmitido ni negar las transacciones realizadas ya que gracias al sistema de consenso todos los nodos de la red escuchan y confirman dichas transacciones. Una vez las transacciones sean aceptadas y los bloques incluidos en la cadena, no se podrá negar la existencia de dicha transacción. Esto para la cadena de custodia implica que una persona no puede alterar un registro para borrar su rastro de acceso a la prueba.
- Accesos por criptografía de clave privada: los accesos a la plataforma se realizarán usando cuentas basadas en la criptografía de clave privada y clave pública por lo que será necesario para un usuario conocer la clave privada de la cuenta para su acceso. Esto supone un aumento en la seguridad con respecto al acceso de usuarios por contraseña.
- Redundancia: el sistema de aplicación distribuido asegura que la información estará duplicada en cada uno de los nodos de tal manera que la información sea resistente a fallos. En caso del mal funcionamiento de un sistema o de la desconexión de este por problemas técnicos el resto de la red podrá seguir operando sin ninguna dificultad. Esto produce una gran resistencia ante ataques de denegación de servicios ya que la red seguirá operando con el resto de los nodos restantes.
- Resistencia a ataques: la aplicación de CoC, gracias a la implementación como cadena de bloques permite una gran resistencia hacia los ataques informáticos. Para poder comprometer la red no basta con conseguir alterar un nodo de la red o emitir transacciones

fraudulentas, sino que será necesario comprometer más del 51% del poder computacional de la red para poder alterar la información almacenada en la cadena.

- Transparencia: Todas las transacciones serán visibles por todos los usuarios y nodos y no podrán ser eliminadas, siendo accesibles por todos los usuarios conectados. Por lo que la información será pública a todos los usuarios de la red que podrán comprobar el estado de todas las transacciones. Debido a la sensibilidad de la información que trata la cadena de bloques será necesario que esta aplicación resida en una cadena de bloques privada que sea accesible solo por los nodos autorizados a su uso.

Capítulo 6
Conclusiones y líneas futuras

En este capítulo se describen las conclusiones obtenidas a partir del desarrollo de este trabajo y las líneas futuras del trabajo para la obtención de una mejor herramienta para la implementación del sistema de cadena de custodia utilizando la tecnología Blockchain.

6.1 Conclusiones

La eficacia del proceso de cadena de custodia es imprescindible para la validez de las pruebas en un proceso judicial. En base a la no regulación de dicha implementación tanto como a la disparidad de los métodos adoptados por las distintas instituciones se dificulta tanto la comunicación entre las entidades que participan como la seguridad del proceso. Debido a estas dificultades se debería llevar a cabo una unificación que mejore la colaboración entre instituciones implicadas en el proceso mejorando así la eficiencia del sistema y la capacidad de colaboración de los organismos.

En base a los problemas existentes con dichas implementaciones y teniendo en cuenta todas las ventajas que aporta la tecnología conocida como cadena de bloques, una solución con esta tecnología permite una mejora eficiente del sistema. La implementación de este sistema ha de seguir las bases establecidas para las aplicaciones descentralizadas en Blockchain para asegurar todas las ventajas de dicha tecnología suponiendo una mejora respecto a otra implementación digital que no utilice esta tecnología.

Una aplicación como CoC permitirá asegurar y controlar la cadena de custodia asegurando el proceso, estandarizando la comunicación entre las distintas entidades con una única plataforma. Se asegurará el acceso a la información de una manera más eficiente, la información estará validada y no será posible la alteración de esta por ninguna persona interna o externa al proceso. La eliminación de la confianza a los usuarios o nodos conectados y la realización de pruebas de trabajo como método para validar la información permite la seguridad e inmutabilidad de la información. Estos sistemas conllevan la elevación de la dificultad de los ataques informáticos o la realización de acciones fraudulentas contra el proceso de cadena de custodia.

Debido a la sensibilidad de la información con la que se está tratando se hace necesario que la aplicación esté desplegada dentro de una cadena de bloques privada para asegurar el control de acceso a la información que está en la misma solo por los nodos autorizados, es decir, los intervinientes en los procesos de cadena de custodia.

Gracias a una solución como CoC se disminuirán los recursos necesitados para asegurar y demostrar la cadena de custodia y los recursos empleados para proteger la información ante modificación por parte de usuarios o atacantes.

6.2 Líneas futuras

El crecimiento de la tecnología blockchain, así como el crecimiento del propio Ethereum produce un aumento en la comunidad que los soporta. Ethereum está en un proceso de crecimiento y de actualización, en este momento migrando a Ethereum 2.0 que permitirá nuevas funcionalidades y cambiará implementaciones en uso. Debido a todos estos cambios que está sufriendo dicha plataforma

se hará necesario en un futuro cercano cambiar alguna de las implementaciones de la propia aplicación para aprovechar y mejorar el rendimiento usando las nuevas funcionalidades.

Una de las principales mejoras que podría sufrir la aplicación es la integración del propio sistema de control de acceso a las pruebas, asegurando el acceso a dichas pruebas solo durante el periodo que se pueda acceder a la prueba mientras se disponga de su custodia. En el caso de pruebas informáticas, que el control a la plataforma para visualizar o trabajar con las propias evidencias necesite de la comprobación a la propia aplicación de cadena de custodia de que eres la persona actualmente responsable de esa prueba. En el caso de evidencias físicas, que se relacione de manera automática el almacén de pruebas con la aplicación para asegurar de manera automática que la persona que está sacando la prueba es la responsable de su custodia y se cambie de manera automática la custodia de la prueba al sacarla del almacén. La integración completa con el sistema de almacenamiento de evidencias aseguraría la integridad de las pruebas digitales, estas podrían ser almacenadas para su descarga, dejando el registro de las personas con acceso a la evidencia, así como la seguridad al no permitir la actualización de las pruebas, por lo que dicha prueba permanecerá intacta. Cabe destacar que dicho control de pruebas no ha de implementarse utilizando la tecnología de blockchain, la información de la cadena de bloques es pública a todos los nodos que estén conectados por lo que no es recomendable el almacenamiento de las propias evidencias en la cadena. El almacenamiento podría llegar a realizarse, pero habría que tener en cuenta la necesidad de cifrar las pruebas antes de ser introducidas en la cadena.

Por otro lado, una línea interesante para seguir trabajando en el futuro es el control de acceso a toda la información sobre la prueba. Aunque la cadena de custodia y la existencia de una prueba sea pública para todas las personas que acceden a la plataforma, se puede implementar un control de acceso a los datos más privados sobre dicha prueba de manera que solo puedan ser visualizados por las personas correspondientes. En el contrato de los usuarios se puede implementar y relacionar de tal manera que se controle el acceso a dicha información de manera más sencilla. A pesar de que el control esté implementado y el acceso a las funciones para obtener la información no esté disponible en el contrato, la información seguirá estando en la cadena por lo que seguirá siendo accesible.

En relación con posibles mejoras habría que realizar pruebas de usabilidad en ambientes reales de tal manera que se pudieran observar de manera evidente las carencias de la aplicación y realizar un análisis de las funcionalidades extras que requieren los usuarios.

Capítulo 7
Conclusions and Future works

In this chapter it is shown the conclusions extracted from the project. Some future lines are proposed as well in order to focus on the key topics in which the development of the tool would have a high improvement on the application for the chain of custody based on a blockchain.

7.1 Conclusions

The efficiency of the chain of custody is a requirement for the ratification of the evidence in any court process. Based on the lack of regulations about the process implementation, as well as the wide variety of methods adopted by the different institutions, a unification of the methodology should be carried out. In order to improve the efficiency of the process and its reliability, one method should be implemented which will end on a better and easy communication between institutions that collaborate on the process.

Based on the existent problems of the nowadays implementation and taking into consideration the advantages which are inherent to the blockchain technology, a solution implemented from this technology will improve the chain of custody system. The implementation of the solution has to take into account the foundations of the decentralized web apps to ensure the advantages given by the blockchain that are used in contrast to other digital implementation of the process without blockchain technology.

An application as CoC will allow to ensure and control the chain of custody. It will standardize the process and the communications between different organizations using one common platform. Access to the information of the process will be more efficient, information will be validated, and it will not allow any possible alteration or modification by any internal or external person to the process. The deletion of the trust to the connected users and nodes and the realization of the proof of work as method to validate the information will ensure the security and the immutability of the information. This method increases the difficulty of the perpetrations of digital attacks or bad intentioned actions against the chain of custody process.

Due to the high sensibility of the information which is managed it is necessary for the blockchain in which the application is deployed to be a private blockchain to ensure a control on the access to the information which is stored on it. On that way, only authorized nodes will be able to access to the details and the data of the process.

Thanks to a solution as CoC, the resources required to ensure and demonstrated the chain of custody and the resources used to prevent the information to be changed can be decreased at the same time that the security increase.

7.2 Future works

The growth of the blockchain technology as well as Ethereum have produced an important increase on the community that they have. Nowadays, Ethereum is on a growing and updating process, it is migrating right now to the Ethereum 2.0. New functionalities are being realised producing actual implementations to be deprecated. Due to all changes on the framework and on the platform, it will

be necessary to change some implementations of the CoC to take advantage of this features and improve the performance of the application.

One of the main features which can take place on the application is its integration with an access control system to the evidence. This can improve the app ensuring that the evidences are only accessible to the person responsible for the custody and they can only access them the time which are allowed. In the case of digital evidences, the system will connect directly to the blockchain to ensure that your account is the responsible for custody before showing the content of the evidence. In the case of physical evidence, the app should be connected to the evidence depository, on a request of an evidence the depository give the evidence and change the state of the blockchain in order to give the custody to the person which took the evidence. At the returning of the evidence the change will be registered as well changing the custody to the depository. The complete integration with the depository system which ensure the integrity of digital evidence, will be stored and could be downloaded but never overwritten, its hash could always be checked on the chain in order to ensure that they have not be modified in a security failure. The system of evidence depository should not be implemented on the blockchain, as blockchain are distributed and public adding the content of evidences to the chain would be a security failure. It could yet be implemented but the evidences should be encrypted before they are stored on the blockchain and decoded outside the blockchain.

Another interesting line of research is to implement access control to the information stored from an evidence. Even if the information stored on the blockchain on is public, the contracts can define methods to easily give access to the information of an evidence only to the people who has access. However, people without access from the contract could search on the blockchain for the information that they need. This could be implemented on the user contract and connecting both contracts in order to see the person access to a specific evidence or case.

In relation with possible improvements it will be needed to create a test environment in order to check the usability of the application and to observe the necessity of the users. In that way, improvements can be done personalizing the app to the users which are going to use and implement extra features that are required for them.

Capítulo 8
Presupuesto

En este capítulo se calculará una estimación sobre el coste de la realización de este Trabajo de Fin de Grado para la creación de la aplicación y los costes de los materiales necesarios para implementar y desplegar la solución final en un entorno real.

8.1 Presupuesto Personal

En este apartado se estiman los costes humanos en horas de trabajo y formación que han sido invertidas en el estudio, comprensión, documentación e implementación de la solución final.

Concepto	Cantidad de Horas	Coste	Total
Estudio tecnología Blockchain	30	18€	540€
Estudio tecnología Ethereum	30	18€	540€
Estudio de los Contratos Inteligentes	20	18€	360€
Estudio Dapps	10	18€	180€
Estudio de Cadena de Custodia	30	18€	540€
Entrevistas	2	18€	36€
Estudio de React	25	18€	450€
Aprendizaje Ganache y Truffle	10	18€	180€
Diseño de la aplicación	5	18€	90€
Implementación de Contratos	30	18€	540€
Aprendizaje Web3 y Metamask	15	18€	270€
Implementación del Front-End	50	18€	900€
Comunicación entre Front-End y Back-End	40	18€	720€
Total:	**297 horas**		**5346€**

Tabla 8.1: *Presupuesto de Personal*

8.2 Presupuesto Componentes

En este apartado se calculan los costes de los distintos componentes necesarios para lanzar la aplicación.

En una primera propuesta, se propone la creación de una red Ethereum privada como solución para la Policía Nacional de Tenerife. Existiendo seis comisarías se propone la existencia de al menos un nodo completo que participe activamente de la blockchain y sirva la aplicación web por cada una de las comisarías que estarán en servidores de manera que aseguren la disponibilidad del servicio. Finalmente se necesitan ordenadores en las distintas comisarías para el acceso a la aplicación.

Concepto	Cantidad	Coste	Total por elemento
Servidores	6	700€	4200€
Ordenador	6	600€	3600€
Total:			7800€

Tabla 8.2 *Presupuesto de componentes completo*

En una segunda propuesta se propone la creación de una red Ethereum privada también, para la Policía Nacional de Tenerife, pero basándose en los servicios web de Amazon. Existiendo seis comisarías se propone la existencia de al menos un nodo completo por cada una de las comisarías que se sirve desde un EC2. Al igual que en el presupuesto anterior, se hacen necesarios los ordenadores para acceder al servicio y además un dominio común a todos.

Con esta solución se tendrían todos los nodos en la misma red creada fácilmente desde Amazon que permite la facturación por el uso de esta.

Concepto	Cantidad	Coste	Total por elemento
EC2(t2.medium)	6	0.067€/h	3434€/año
Dominio	1	15€/año	15€/año
Ordenador	6	600€	3600€
Total:			7049€

Tabla 8.3 *Presupuesto de componentes con AWS*

Ambas redes propuestas serían altamente escalables incluyendo nuevos nodos asegurando el crecimiento de la red y la seguridad de esta. A pesar de las ventajas aportadas por AWS y su precio competitivo, se recomienda el uso de los servidores locales para asegurar un control total sobre la información y almacenamiento sobre la información que se trata.

8.3 Presupuesto Final

El presupuesto final teniendo en cuenta el presupuesto personal y utilizando el presupuesto de componentes que se basa en el control local de la red es el siguiente:

Concepto	Precio
Costes Humanos	5346€
Coste Componentes	7800€
Total:	13246€

Tabla 8.4 *Presupuesto Final*

Capítulo 9
Anexo. Instalación

En este capítulo se muestran las instrucciones para la instalación de la aplicación. Se mostrarán las indicaciones para un sistema Linux.

9.1 Instalación

El primer paso será instalar las herramientas de Truffle[17] y Ganache[19], las instrucciones para su descarga se encuentran en la documentación de las propias herramientas, ambas necesitan de la instalación previa de Node v8.4.9 o posterior. La instalación de Truffle se realiza con la ejecución del siguiente comando:

```
npm install -g truffle
```

La instalación de Ganache se realiza por medio de la descarga de la aplicación de escritorio desde su página oficial [21].

Para el funcionamiento de la aplicación será necesaria la instalación de la extensión en el navegador de Metamask, en el caso de Google Chrome se puede hacer desde la tienda de extensiones del navegador [22].

Finalmente será necesario la descarga y la instalación de CoC, para ello se ha de clonar el repositorio de GitHub [23]. Se ha de realizar la instalación de todas las dependencias con la ejecución del siguiente comando dentro del directorio del proyecto:

```
npm install
```

Para el lanzamiento de la aplicación será necesaria la apertura de una sesión de Ganache en la que se especifique como fichero de configuración el fichero 'truffle-config.js' que se encuentra dentro del proyecto. A continuación, importar en la extensión de metamask una de las cuentas que ofrece Ganache.

Finalmente, el lanzamiento de la aplicación puede realizarse ejecutando los siguientes comandos dentro del directorio de la aplicación.

```
npm run compile

cd app

npm run dev
```

La aplicación ya está disponible en 127.0.0.1:3000

Bibliografía:

[1] Haber, S., Stornetta, W.S. How to time-stamp a digital document. J. Cryptology 3, 99–111 (1991)
[2] Bayer D., Haber S., Stornetta W.S. (1993) Improving the Efficiency and Reliability of Digital Time-Stamping
[3] Nakamoto, Satoshi. (2009). Bitcoin: A Peer-to-Peer Electronic Cash System
[4] ethereum.org. 2020. Ethereum Whitepaper. [online] Disponible en: https://ethereum.org/en/whitepaper/ [Accessed 30 August 2020].
[5] Medium. 2020. Ethereum 2.0: A Complete Guide. Casper And The Beacon Chain.. [online] Disponible en: https://medium.com/chainsafe-systems/ethereum-2-0-a-complete-guide-casper-and-the-beacon-chain-be95129fc6c1 [Accessed 31 August 2020].
[5] ATS 6629/2014 - ECLI: ES:TS:2014:6629A [2014] 1227/2014 (Tribunal Supremo. Sala de lo Penal).
[6] STS 5677/2013 - ECLI: ES:TS:2013:5677 [2013] 10448/2013 (Tribunal Supremo. Sala de lo Penal).
[7] Bonomi, S., Casini, M. and Ciccotelli, C., n.d. B-CoC: A Blockchain-Based Chain of Custody for Evidences Management in Digital Forensics. Schloss Dagstuhl - Leibniz-Zentrum für Informatik
[8] Orden INT/1202/2011 «BOE» núm. 114, de 13 de mayo de 2011, páginas 48748 a 48961 (214 págs.)
[9] Orden JUS/1291/2010 «BOE» núm. 122, de 19 de mayo de 2010, Sec. I. Pág. 43459
[10] SJP 39/2016 - ECLI:ES:JP:2016:39 [2016] 385/2015 (Juzgado de lo Penal, Gijón 3).
[11] Yunianto, E., Prayudi, Y. and Sugiantoro, B., 2019. B-DEC: Digital Evidence Cabinet based on Blockchain for Evidence Management. International Journal of Computer Applications, 181(45), pp.22-29.
[12] es.reactjs.org. 2020. React. [online] Disponible en: https://es.reactjs.org/docs/ [Accessed 31 August 2020].
[13] 2020. Next.Js. [online] Disponible en: <https://nextjs.org/docs/> [Accessed 31 August 2020].
[14] 2020. Tailwind CSS. [online] Available at: <https://tailwindcss.com/> [Accessed 31 August 2020].
[15] 2020. Web3.Js. [online] Disponible en: <https://web3js.readthedocs.io/en/v1.2.11/> [Accessed 31 August 2020].
[16] 2020. Metamask Docs. [online] Disponible en: <https://docs.metamask.io/guide/> [Accessed 31 August 2020].
[17] 2020. Truffle. [online] Disponible en: <https://www.trufflesuite.com/docs/truffle/overview> [Accessed 31 August 2020].
[18] 2020. Solidity. [online] Disponible en: <https://solidity-es.readthedocs.io/es/latest/> [Accessed 31 August 2020].
[19] 2020. Ganache. [online] Disponible en: <https://www.trufflesuite.com/docs/ganache/overview> [Accessed 31 August 2020].
[20] npm. 2020. Truffle-Assertions. [online] Disponible en: <https://www.npmjs.com/package/truffle-assertions> [Accessed 31 August 2020].

[21] 2020. Ganache. [online] Disponible en: <https://www.trufflesuite.com/docs/ganache > [Accessed 31 August 2020].
[22] 2020. Metamask Chrome Extension. [online] Disponible en: <https://chrome.google.com/webstore/detail/metamask/nkbihfbeogaeaoehlefnkodbefgpgknn?hl=es> [Accessed 31 August 2020].
[22] 2020. Coc Github Repository. [online] Disponible en: <https://github.com/diego-algomas/CoC> [Accessed 7 September 2020].