Proyecto de Grado

Presentado ante la ilustre Universidad de Los Andes como requisito parcial para optar al título de Magíster Scientiae en Modelado y Simulación de Sistemas.

Modelo de un sistema de almacenamiento de datos médicos basados en Blockchain

Presentado por:

Ing. Rosa Adriana Hidalgo Guerrero.
Profesor Tutor: Jacinto Dávila.

Octubre 2023

Agradecimientos

Resumen

Este proyecto de investigación plantea una estrategia para la administración de historias médicas a través de la tecnología blockchain. En esencia, se propone un sistema que permite el almacenamiento seguro de datos críticos en una red blockchain descentralizada.

El núcleo de esta propuesta radica en la implementación de un "contrato inteligente" diseñado específicamente para la gestión de información médica. Este contrato inteligente cumple una doble función: en primer lugar, almacena los datos en la blockchain, lo que garantiza la integridad de la información. En segundo lugar, introduce la capacidad de definir reglas y condiciones personalizadas para acceder y actualizar estos datos. Esto se traduce en un nivel adicional de control, ofreciendo a los pacientes y profesionales de la salud la seguridad de que la información médica esté protegida y sujeta a políticas de acceso específicas. De este modo, se promueve tanto la confidencialidad como la integridad de los datos médicos, lo cual es esencial en el contexto de la atención médica.

Adicionalmente, se presenta un modelo de sistema que facilita la carga de archivos médicos mediante el aprovechamiento de tecnologías avanzadas, tales como el Sistema de Archivos Interplanetario (IPFS) y web3. Estas tecnologías brindan una solución eficiente y segura para cargar y acceder a archivos médicos en la red blockchain. Esto no solo optimiza el proceso, sino que también resuelve los desafíos de seguridad y privacidad asociados con la gestión de archivos médicos en línea.

Índice de Contenido

Índice de Figuras y Tablas

Capítulo 1

1.1. Introducción

Las historias medicas son indispensables en el ámbito de la medicina debido a múltiples razones fundamentales, ya que proporcionan una visión completa del paciente, lo que ayuda en el diagnóstico y tratamientos adecuados. Además, aseguran la continuidad del cuidado, promueven la seguridad del paciente y se convierten en una valiosa fuente de información para la investigación y la educación médica. Las historias medicas también desempeñan un papel crucial al brindar datos completos y precisos, garantizando una atención de calidad y mejorando el conocimiento médico en general.

"La historia médica es la exploración metódica que se practica a todo paciente y de la cual debe dejarse constancia escrita. Por lo tanto, es un documento que nos ilustra a la persona enferma desde que inicia su primer contacto con el médico, continuando por examen clínico detallado y sometido a exámenes especiales; llegándose finalmente a un diagnóstico. En otras palabras, podríamos decir que es la narración escrita, clara, precisa, detallada, ordenada de todos los datos y conocimientos, tanto anteriores (antecedentes personales y familiares), como actuales. Resume la herencia y hábitos de un ser humano, su constitución, fisiología y psicología." (Wuani, 2010). Antes de la informática, las historias médicas se manejaban principalmente en formato de papel y se almacenaban físicamente en archivos. Algunas prácticas comunes utilizadas para el manejo de las historias médicas antes de la era digital eran, el archivo de papel que registraba formularios de forma impresa o manual organizándose en archivos en los consultorios médicos, la transferencia manual de la historia cuando el paciente cambiaba de médico o de centro asistencial, el acceso limitado de la información contenida en las historias médicas estaba restringido a los profesionales de la salud autorizados. Los pacientes tenían que solicitar explícitamente la divulgación de su información médica a terceros.

Es importante tener en cuenta que estas prácticas pueden variar según la ubicación geográfica y el entorno de atención médica.

En la actualidad, las historias médicas suelen manejarse en formato electrónico, lo que facilita su acceso, almacenamiento y la manera que se comparte esta información entre los profesionales de la salud. Este enfoque digital ha reemplazado en gran medida el uso de registros médicos en papel, brindando una serie de ventajas significativas. Los sistemas electrónicos de registro médico suelen estar diseñados para permitir que los datos se compartan entre diferentes proveedores de atención médica. Esto significa que los médicos y profesionales autorizados pueden acceder a la historia médica de un paciente, independientemente de la ubicación física donde se haya generado el registro. Dado que la información médica es confidencial y sensible, se aplican rigurosas medidas de seguridad y privacidad en el manejo de las historias médicas electrónicas.

En algunos casos, todavía existen registros médicos en formato físico que deben ser digitalizados para su inclusión en el sistema electrónico. Esto implica la conversión de los documentos en papel a formato digital mediante escaneo o ingreso manual de datos.

Un sistema de almacenamiento de información médica debe cumplir con una serie de características para garantizar la integridad, seguridad y accesibilidad de los datos, debe contar con medidas de seguridad sólidas para proteger la información médica de accesos no autorizados, garantizando la confidencialidad y privacidad de los datos, la información almacenada debe ser precisa, completa y no modificada de forma no autorizada. El sistema debe ser capaz de proporcionar acceso a la información médica cuando sea necesario. Debe estar disponible y funcionando de manera confiable para garantizar la continuidad del cuidado del paciente. El sistema debe ofrecer funcionalidades de búsqueda y recuperación eficientes, permitiendo a los profesionales de la salud encontrar rápidamente la información relevante cuando la necesiten.

La historia médica también es un documento legal que solo puede ser conocido por terceros con la autorización del paciente o por algún mandato judicial. Comprende el conjunto de procesos asistenciales de cada paciente, siendo normal que se identifiquen los médicos y profesionales que han intervenido en estos procesos. Si la historia medica no es elaborada de manera correcta podría conducir a un mal diagnóstico. Es importante tener en cuenta que la transición hacia sistemas de almacenamiento digital o electrónicos de historias médicas puede ser limitada en Venezuela debido a restricciones económicas y falta de recursos tecnológicos. Sin embargo,

algunos centros de salud más grandes y modernos pueden haber implementado sistemas electrónicos de registro médico, que permiten almacenar y acceder a las historias médicas en formato digital. En Venezuela la implementación de un sistema de datos médicos basado en blockchain puede presentar desafíos. Sin embargo, si se implementa de manera efectiva, la tecnología blockchain tiene el potencial de brindar beneficios significativos en términos de transparencia, eficiencia y control de información. En este proyecto se desarrolla el modelo de un sistema almacenamiento de datos médicos basado en blockchain y su factibilidad económica.

Según (Starfield & Smith, 1994) Un modelo es una representación intencional de algún sistema real. Construimos y usamos modelos para resolver problemas o responder preguntas sobre un sistema o una clase de sistemas.

Un modelo puede ser una representación visual, un conjunto de reglas y especificaciones, o una combinación de ambos. Puede ser utilizado para comprender, analizar, comunicar y simular el funcionamiento de un sistema antes de su implementación real.

Para (Corvo, 2019) La factibilidad económica es el análisis de los costos e ingresos de un proyecto en un esfuerzo por determinar si resulta o no lógico y posible poder completarlo. Es un tipo de análisis de costo-beneficio del proyecto examinado, que evalúa si es posible implementarlo.

1.2. Contextualización del Problema

Estamos acostumbrados a creer que la práctica médica, en particular de primer nivel, ha venido actualizando sus soluciones, de aquellos tiempos donde un médico tenía algún registro en papel, por ejemplo, de la historia de un paciente, y que ha venido ocurriendo que la introducción de nuevas formas de almacenamiento de información, como las computadoras, se han ido incorporando y ahora tenemos que la información de las personas esta almacenada en formato electrónico. Pero no es tan simple porque una solución generosa al problema de almacenar la historia médica de una persona, tendría que permitirle a esa persona, el o la paciente, disponer de su información, no solo cuando el mismo médico le vuelva a atender, sino cuando otra situación

de emergencia o no de emergencia surja, y esta persona necesite acceder, o actualizar su registro médico, para seguir llevando con cuidado el seguimiento de todos los acontecimientos, que pueden pasar en un tratamiento cualquiera, de un problema médico o de atención de la salud. Así que se trata de repensar el problema desde el punto de vista del paciente y, desde el punto de vista de las nuevas tecnologías, que le permitirían a una persona disponer de esa información en cualquier lugar que la necesite.

1.3. Recuento Histórico

"Aunque se puede pensar que la historia clínica se trata de una herramienta reciente, lo cierto es que sus orígenes se remontan al año 2500 a.C. Antes de Hipócrates, la Medicina había sido una mezcla de empirismo y magia, influida por la visión religiosa de cada pueblo. Se consideraba que las enfermedades eran un castigo de los dioses. Gracias a Hipócrates, las enfermedades dejaron de tener un origen divino para considerarse fenómenos naturales, provocadas por causas ambientales. De ahí que las primeras historias médicas de las que se tienen noticias se encuentran en los tratados hipocráticos dentro de los libros I y II de las epidemias, escritos entre los siglos V y VI a.c" (Gonzales, 2021).

Estas primeras historias médicas y hasta hace muy poco, se han resguardado en papel. A partir de la segunda mitad del siglo XX, y con los avances en la tecnología informática, hospitales, clínicas y consultas privadas han optado por resguardar esta información, de manera electrónica, por lo general en computadores locales y de uso personal.

Es el caso el sector público venezolano, donde se siguen usando registros en papel ya que no pueden mantener el ritmo y el costo de la tecnología. Por otro lado, en el sector privado donde es probable que, si se tengan recursos, algunos médicos, llevan su propio registro en computadores personales, privando a el o la paciente del control de la información de su historia médica.

Se han realizado investigaciones y proyectos para el manejo de historias médicas sobre la red Blockchain y el uso de contratos inteligentes. Algunas de estas investigaciones son:

El Instituto de Tecnología de Massachusetts, propone el desarrollo de un sistema de gestión de registros descentralizado para manejar las historias clínicas, llamado MedRed, utilizando la tecnología Blockchain, que presenta a los pacientes un registro completo e inmutable y un fácil

acceso a su información médica en todos los proveedores y sitios de tratamiento. Al aprovechar las propiedades únicas de Blockchain, MedRec promete gestionar la autenticación, la confidencialidad, la responsabilidad y el intercambio de datos, consideraciones cruciales al manejar información confidencial. (Azaria, 2016)

MeDShare plantea un sistema que aborda el problema del intercambio de datos médicos entre los custodios de big data médicos en un entorno sin confianza. El sistema está basado en blockchain y proporciona procedencia, auditoría y control de datos médicos compartidos en repositorios en la nube entre entidades de big data. MeDShare propone monitorear entidades que acceden a datos para uso malicioso desde un sistema de custodia de datos. El diseño emplea contratos inteligentes y un mecanismo de control de acceso para realizar un seguimiento efectivo del comportamiento de los datos y revocar el acceso a las entidades infractoras al detectar una violación de los permisos sobre los datos. (Xia, 2017).

Médicalchain diseña un sistema que permita al usuario dar a los profesionales de la salud acceso a sus datos personales de salud, registrando las interacciones con estos datos de forma auditable, transparente y segura en una red Blockchain propia. Médicalchain propone una plataforma para que otros la utilicen para crear aplicaciones que complementen y mejoren la experiencia del usuario. Los usuarios podrán aprovechar sus datos médicos para impulsar una gran cantidad de aplicaciones y servicios, utilizando framework Hyperledger Fabric y la tecnología Blockchain. (Albeyatti, 2017)

La tecnología Blockchain puede ofrecer muchas ventajas en el tema de almacenamiento de datos médicos, pero deja abiertas algunas preguntas que en este proyecto se investigaron: ¿Cuál sería el costo de utilizar esta tecnología para almacenar y actualizar este tipo de datos?, ¿Qué tipo de blockchain utilizar?, ¿Qué implicación legal se enfrenta?

1.4. Objetivos de este proyecto

1.4.1. Objetivo General

1. Desarrollar un modelo de sistemas de gestión de historias médicas almacenado en la red Blockchain.

1.4.2. Objetivos Específicos

1. Diseñar y desarrollar un contrato inteligente que permita el almacenamiento de la información médica de una persona, dentro de una red Blockchain.
2. Evaluar los costos que implican almacenar datos en la red Blockchain.
3. Evaluar los costos que implican cada transacción de un contrato inteligente en la red Blockchain.
4. Analizar la factibilidad económica de un sistema de gestión de historias médicas almacenado en la red Blockchain.
5. Estimar costos en otras redes blockchain y analizar posibles alternativas para el despliegue.

Para alcanzar esos objetivos, se ha realizado el diseño y prueba de un modelo de contrato inteligente que permite almacenar datos médicos en una red Blockchain, un modelo de subida de imágenes a un sistema descentralizado, y un modelo de servicio de registro electrónico de datos médicos, también se evaluó la factibilidad económica de un sistema de gestión de historias médicas en la red Blockchain, información que está reflejada en los siguientes capítulos del documento de esta forma:

El capítulo 2 se muestra la metodología a seguir para el desarrollo y despliegue de un contrato inteligente en una red Blockchain y un sistema descentralizado de subida de archivos, también analizar la metodología a seguir para evaluar la factibilidad económica de un “Decentralized Applications” o “Aplicaciones descentralizadas” (Dapp).

El capítulo 3 se presenta el modelo de un servicio de registro electrónico de datos médicos, gestionados por los y las pacientes.

El capítulo 4 se presenta el modelo de sistemas de gestión de historias médicas almacenado en la red Blockchain.

El capítulo 5 se evalúa la factibilidad económica del sistema de gestión de historias médicas almacenado en la red Blockchain.

El Capítulo 6 las conclusiones obtenidas del trabajo realizado.

Capítulo 2

Metodología

Al proponer el modelo de un sistema de almacenamiento de datos en la red blockchain y su factibilidad económica, enfrentamos distintas ideas y conceptos que han venido desarrollando métodos de confianza, seguridad, transparencia y trazabilidad de datos compartidos.

El propósito de este trabajo de investigación es modelar un sistema que bajo la tecnología de redes blockchain y archivos descentralizados, permita almacenar datos médicos. El enfoque se centrará en analizar cómo la tecnología blockchain puede mejorar la seguridad, la accesibilidad y la integridad de las historias médicas y los posibles beneficios que esta solución puede brindar al ámbito de la salud en términos de eficiencia y confiabilidad de los datos médicos.

Desarrollar un modelo de sistemas de gestión de historias médicas almacenado en la red blockchain busca garantizar la integridad y confidencialidad de la información médica almacenada en la blockchain. Esto implica implementar mecanismos de control de acceso para proteger los datos sensibles de los pacientes y evitar cualquier tipo de manipulación. Garantizar que los pacientes tengan acceso y control sobre sus propias historias médicas, pudiendo otorgar permisos específicos para compartir su información con otros profesionales de la salud o instituciones, así como la posibilidad de realizar modificaciones o actualizaciones en su historial médico.

La tecnología blockchain ofrece un registro inmutable de todas las transacciones y cambios realizados en las historias médicas, lo que permitirá una trazabilidad completa de las acciones realizadas en los datos médicos, generando mayor confianza y seguridad en el sistema. Asimismo, almacenar las historias médicas en la blockchain mejorará la eficiencia en la gestión de la información y la administración de registros médicos físicos o sistemas centralizados.

2.1. Procedimiento

El procedimiento por seguir para desarrollar el modelo de un sistema de almacenamiento de datos médicos basado en Blockchain, se desarrolla en dos partes, primero se diseña el modelo de contrato inteligente, para luego diseñar el modelo de software basado en Web3 para cargar archivos y por último se estima la factibilidad económica de todo el sistema.

2.1.1 Modelo de Contrato Inteligente

Antes de hablar sobre un contrato inteligente, es conveniente hablar sobre la red Blockchain.

Blockchain es una red descentralizada y segura en la que se puede almacenar y compartir información de manera confiable sin depender de un solo servidor central. Es una tecnología que permite crear un registro digital compartido y distribuido entre múltiples participantes, llamados nodos. A diferencia de las bases de datos convencionales, la red blockchain no utiliza una estructura centralizada de almacenamiento de información. En lugar de ello, utiliza estos nodos interconectados, donde cada nodo tiene una copia completa y actualizada de todo el registro. Esto significa que la información se almacena y se distribuye en todos los nodos de la red en lugar de estar concentrada en un solo lugar.

La seguridad de la blockchain se basa en la criptografía, que protege los datos mediante algoritmos de encriptación. Además, al estar distribuida en múltiples nodos, la información se replica y respalda en toda la red, lo que la hace resistente a fallos y ataques.

Los contratos inteligentes son programas informáticos autónomos que se ejecutan en la red blockchain. Son como contratos tradicionales, pero en lugar de estar escritos en papel y ser ejecutados manualmente, los contratos inteligentes se programan en código y se ejecutan automáticamente cuando se cumplen ciertas condiciones predefinidas. Se ejecutan de forma descentralizada, por lo que no requieren de intermediarios o terceros para su ejecución. Una vez que se establecen las reglas y condiciones del contrato en el código, se convierten en inmutables y se ejecutan automáticamente al cumplirse las condiciones especificadas.

La red blockchain utilizada para probar el modelo de contrato inteligente fue Ethereum, una plataforma descentralizada que permite la creación y ejecución de aplicaciones descentralizadas,

también conocidas como dapps (aplicaciones descentralizadas). Ethereum se basa en la tecnología de la cadena de bloques y ofrece un entorno seguro y confiable para el desarrollo y despliegue de contratos inteligentes, ofrece herramientas y recursos que facilitan su implementación y prueba.

El modelo de contrato inteligente fue desarrollado utilizando el lenguaje de programación Solidity, que es ampliamente utilizado en la red blockchain Ethereum. Solidity es un lenguaje específicamente diseñado para la creación de contratos inteligentes y la ejecución de aplicaciones descentralizadas en la plataforma Ethereum. Al utilizar Solidity, se pueden definir las reglas y lógica del contrato inteligente de manera precisa y segura. Este lenguaje ofrece una serie de características y funcionalidades que facilitan la implementación de contratos inteligentes, como la capacidad de definir variables, estructuras de datos y funciones que interactúan con la red blockchain.

Además, al ser un lenguaje específico para contratos inteligentes, Solidity proporciona una sintaxis clara y concisa para la programación de la lógica del contrato, lo que facilita su comprensión.

El entorno de programación utilizado fue Remix, una herramienta diseñada para facilitar el desarrollo, prueba y despliegue de contratos inteligentes en la red Ethereum. Remix es una plataforma en línea que ofrece un editor de código integrado y una interfaz de usuario, permitiendo escribir, compilar y depurar contratos inteligentes de manera más eficiente. La herramienta es accesible a través de un navegador web. Una de las características de Remix es su capacidad para realizar pruebas y simulaciones de contratos inteligentes en un entorno virtual. Esto permite verificar el funcionamiento y la lógica de los contratos antes de desplegarlos en la red Ethereum.

2.1.2 Modelo de Carga de archivos

El desarrollo del modelo de la aplicación para cargar archivos se llevó a cabo utilizando una combinación de tecnologías, incluyendo IPFS, web3-react e INFURA, en un entorno de sistema operativo LinuxMint 20.

IPFS, significa InterPlanetary File System, es un protocolo descentralizado que permite el almacenamiento y distribución de archivos en una red peer-to-peer. En este caso, se utilizó IPFS como componente clave para cargar y gestionar los archivos en la aplicación. Proporciona una forma segura y eficiente de almacenar y compartir archivos sin depender de una ubicación centralizada.

Web3-react es una biblioteca de JavaScript que facilita la integración de aplicaciones con la red Ethereum. Permite interactuar con contratos inteligentes y realizar transacciones en la red blockchain. En el contexto de la aplicación para cargar archivos, web3-react se utilizó para conectar la aplicación con la red Ethereum y facilitar la comunicación con contratos inteligentes relacionados con la gestión de archivos.

INFURA es un servicio que proporciona acceso a nodos de la red Ethereum sin la necesidad de ejecutar un nodo completo. Actúa como un intermediario entre la aplicación y la red Ethereum, permitiendo interactuar con la red sin tener que configurar y mantener su propio nodo. En este caso, INFURA se utilizó para establecer la conexión con la red Ethereum y realizar las transacciones necesarias para la carga y gestión de archivos.

El sistema operativo LinuxMint 20 fue utilizado como el entorno de desarrollo para este modelo de aplicación. LinuxMint es una distribución de Linux basada en Ubuntu que ofrece un entorno amigable y estable para el desarrollo de aplicaciones. Proporciona herramientas y recursos útiles para la programación y ejecución de aplicaciones en diferentes lenguajes de programación.

Estas tecnologías permitieron la carga segura y descentralizada de archivos, la interacción con la red Ethereum y la gestión de transacciones necesarias para la aplicación.

2.1.3 Factibilidad Económica

Para estimar la factibilidad económica se seguirá las condiciones y tarifas de las redes blockchain.

Las transacciones en Ethereum se basan en parámetros específicos de la red blockchain. Estos parámetros incluyen el **base fee** o valor base, el **gas** utilizado, el **Max Priority Fee,** el valor actual de la criptomoneda de Ethereum, conocida como **ether**.

El **base fee** es una tarifa básica que se cobra por cada operación en la red Ethereum. Esta tarifa puede variar dependiendo de la demanda y la congestión de la red en un momento dado. Es importante tener en cuenta esta tarifa al realizar transacciones, ya que puede afectar el costo y la velocidad de procesamiento de la operación.

El **gas** es una medida utilizada para calcular la cantidad de recursos computacionales necesarios para ejecutar una operación en Ethereum. Cada operación requiere una cierta cantidad de gas, que a su vez está relacionada con la complejidad y el tiempo de procesamiento de la operación. Cuanto más gas se requiera, mayor será el costo de la transacción, para obtener la tarifa total de gas, necesitamos conocer **baseFeePerGas** que según (Ethereum, 2020) es una tarifa que varía en función de la congestión de la red y se quema, es decir, se retira de la circulación, lo que aporta a la economía deflacionaria de Ethereum y **maxPriorityFeePerGas e**s una tarifa que los usuarios pueden pagar para que su transacción sea procesada más rápidamente por los validadores (antes llamados mineros). Esta tarifa va directamente a los validadores como incentivo.

El valor actual de la criptomoneda de Ethereum, el **ethe**r, también afecta el costo de las transacciones. Dado que las tarifas en Ethereum se pagan en ether, si el valor del ether aumenta, las tarifas en términos de valor monetario también aumentarán.

Ejemplo de una transacción:

En Ethereum, cada operación, ya sea una simple transferencia o la ejecución de un contrato inteligente, requiere una cierta cantidad de "gas" para llevarse a cabo, estas transacciones siempre requieren 21.000 unidades de gas.

- **Para obtener la tarifa total de gas, calculamos de la siguiente de manera:**

  (base fee per gas + max Priority Fee per gas) ∗ Gas = Tarifa total

Supongamos que tenemos:

baseFeePerGas = 190 gwei

maxPriorityFeePerGas = 10 gwei

(190 + 10) * 21000 = 4,200,000 gwei

o

0,0042 ETH

De la cuenta **A** se debitará **1.0042 ETH** (Esto incluye el 1 ETH que envía a **B** más los 0.0042 ETH que se utilizan para pagar las tarifas de gas)

A la cuenta **B** se le acreditará **1 ETH**.

La tarifa base (quemada): La tarifa es **0.00399 ETH** que se quema, este total se obtiene de la resta de la tarifa total de gas 0.0042 ETH – 0.000210 ETH que es la tarifa que se le da a los validadores como recompensa por su trabajo.

El validador mantiene la propina **0.000210 ETH.**

Es importante considerar estos factores al realizar transacciones para garantizar una correcta estimación de los costos y tiempos de procesamiento. Cuando se realiza una transacción en Ethereum, los contratos inteligentes tienen la capacidad de emitir eventos y escribir registros en la cadena de bloques. Estos eventos y registros son utilizados para notificar a las interfaces de usuario sobre ciertos sucesos o cambios en el contrato inteligente.

Los eventos, como su nombre lo indica, son eventos específicos que pueden ser registrados en la Ethereum Virtual Machine (EVM). Estos eventos pueden contener información relevante que

puede ser procesada por las interfaces de usuario. Por ejemplo, un evento puede notificar cuando se realiza una transferencia de tokens o cuando ocurre un cambio de estado en el contrato.

La escritura de registros en la cadena de bloques permite almacenar información de manera permanente y transparente. Esto significa que cualquier cambio o acción realizada en un contrato inteligente queda registrado en la cadena de bloques y puede ser verificado por cualquier usuario.

Al utilizar eventos y registros, las interfaces de usuario pueden acceder a la información generada por los contratos inteligentes y tomar acciones en respuesta a esos eventos. Por ejemplo, una interfaz de usuario puede mostrar una notificación cuando se emite un evento de transferencia de fondos o actualizar la información mostrada en tiempo real cuando ocurren cambios en el contrato.

Al comprender estos parámetros y cómo interactúan entre sí, se puede tener un mayor control y comprensión de las transacciones realizadas en Ethereum. Esto es fundamental para optimizar el uso de recursos y estimar los costos asociados a las transacciones.

Para garantizar la ética en el uso de la información en blockchain, es fundamental obtener el consentimiento informado de las personas cuyos datos se están almacenando. Los usuarios deben tener pleno conocimiento de cómo se utilizarán sus datos y tener la opción de proporcionar su consentimiento o retirarlo en cualquier momento.

Capítulo 3

Modelo de un servicio de registro electrónico de datos médicos, gestionados por los y las pacientes

Para el sistema que se modeló, y que en el futuro se utilizará a través de una Dapp, se han identificado tres procesos fundamentales que cumplen las necesidades principales de los pacientes como usuarios:

Registro de usuario

Este proceso permite que el paciente se registre en el sistema proporcionando la información necesaria, como nombre, identificación y otros datos relevantes. A través de este proceso, se crea una cuenta única y segura para el paciente, que le permitirá acceder a sus servicios médicos y gestionar su historia clínica de manera confiable.

Consulta de historia médica

Una vez registrado, el paciente puede acceder a la función de consulta de su historia médica. A través de la Dapp, el paciente puede ingresar su clave de acceso y realizar consultas específicas sobre su historial médico. El sistema verifica la clave y proporciona al paciente acceso seguro a los registros y datos relacionados con su historial clínico.

Actualización de historia médica

El proceso de actualización de la historia médica permite al paciente realizar cambios o agregar nueva información a su historial clínico existente. A través de la Dapp, el paciente puede ingresar su clave de acceso, definir los datos a actualizar y el sistema se encargará de almacenar y registrar de manera segura y confiable los cambios realizados en la Blockchain. Esto garantiza la integridad y la trazabilidad de la información actualizada.

Se modelaron los procesos de gestión de historias médicas utilizando la plataforma Bonitasoft. Esta plataforma permitió crear de manera separada los procesos internos del sistema necesarios para gestionar el servicio electrónico de datos médicos. Cada modelo de proceso se divide en dos

carriles para representar la interacción del paciente con la aplicación y el flujo de proceso que el sistema debe seguir según las necesidades del usuario.

En el primer carril, se representa la interacción del paciente con la aplicación, es decir, cómo el paciente utiliza la plataforma para acceder a su historia médica, realizar consultas o actualizar su información personal. Este carril muestra la interfaz y la experiencia del paciente dentro del sistema.

En el segundo carril, se muestra el flujo de proceso que el sistema debe seguir para gestionar las historias médicas. Aquí se representan las diferentes etapas y acciones que el sistema realiza internamente para procesar las solicitudes del paciente, como la validación de datos, la actualización de registros, la generación de informes, entre otros.

Al dividir el modelo en dos carriles, se logra una representación clara y separada de la interacción del paciente y el flujo de proceso del sistema. Esto facilita el diseño, la comprensión y la visualización de los procesos de gestión de historias médicas. La plataforma Bonitasoft proporciona las herramientas necesarias para implementar estos modelos y gestionar eficientemente los datos médicos electrónicos.

3.1 Modelo para el registro de un nuevo usuario

El proceso comienza cuando el usuario abre la aplicación y realiza una solicitud para crear una cuenta, el usuario proporciona su información personal, como nombre, correo electrónico y contraseña.

A continuación, el sistema crea una nueva cuenta con una clave única generada automáticamente, esta clave asegura la autenticidad y la seguridad de la cuenta del usuario.

Una vez que la cuenta ha sido creada, el sistema procede a actualizar los datos en la Blockchain. Después de que los datos han sido actualizados en la Blockchain, el usuario tiene la opción de verificar sus claves y registros. Esto implica revisar la información almacenada en su cuenta,

como la clave de acceso y otros datos personales. La verificación permite al usuario confirmar que los datos se han guardado correctamente y que no ha habido modificaciones no autorizadas.

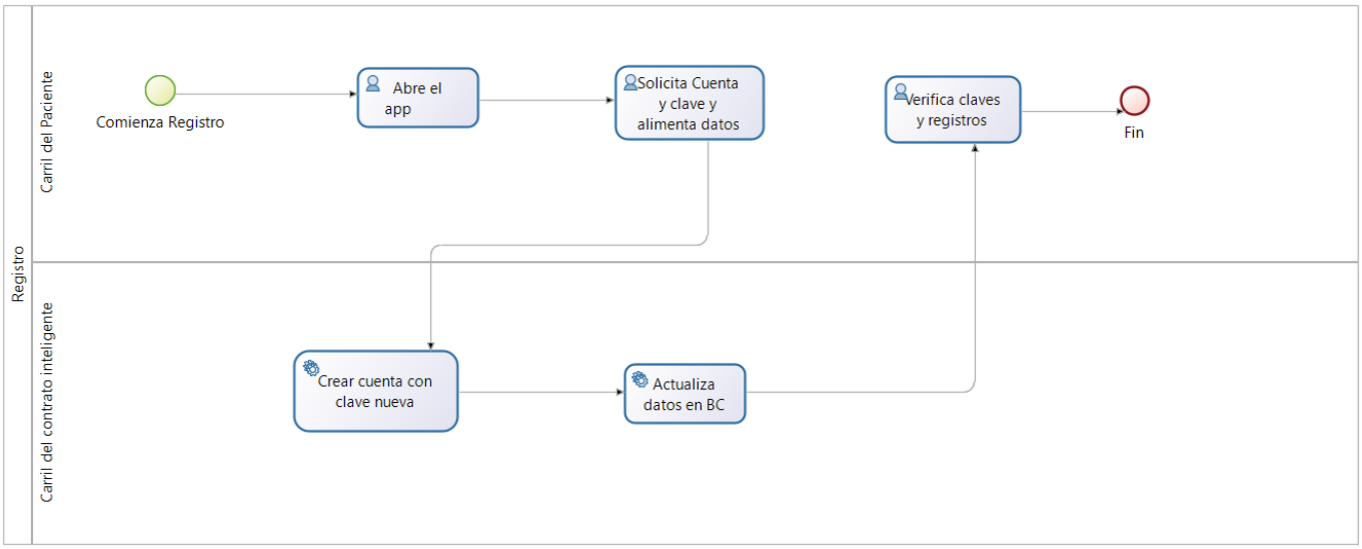


Figura 1 Modelo para el registro de un nuevo usuario

3.2. Modelo para consulta de historia médica

El proceso comienza cuando el usuario abre la Dapp y proporciona su clave de acceso. El sistema valida la clave del usuario para garantizar la autenticidad y seguridad de la consulta. Una vez validada la clave, el sistema prepara la consulta de acuerdo con las preferencias y requerimientos del usuario.

Luego, el usuario define los parámetros de la consulta, estos parámetros ayudan a filtrar y obtener resultados más relevantes para la consulta. El sistema realiza una búsqueda en los registros y logs almacenados en la Blockchain, recuperando la información médica correspondiente a la consulta definida por el usuario. Esta información puede incluir diagnósticos, tratamientos, medicamentos recetados y otros datos relevantes de la historia médica.

Finalmente, el usuario verifica los resultados de la consulta. Puede revisar y analizar la información médica obtenida para asegurarse de que cumple con sus necesidades y expectativas.

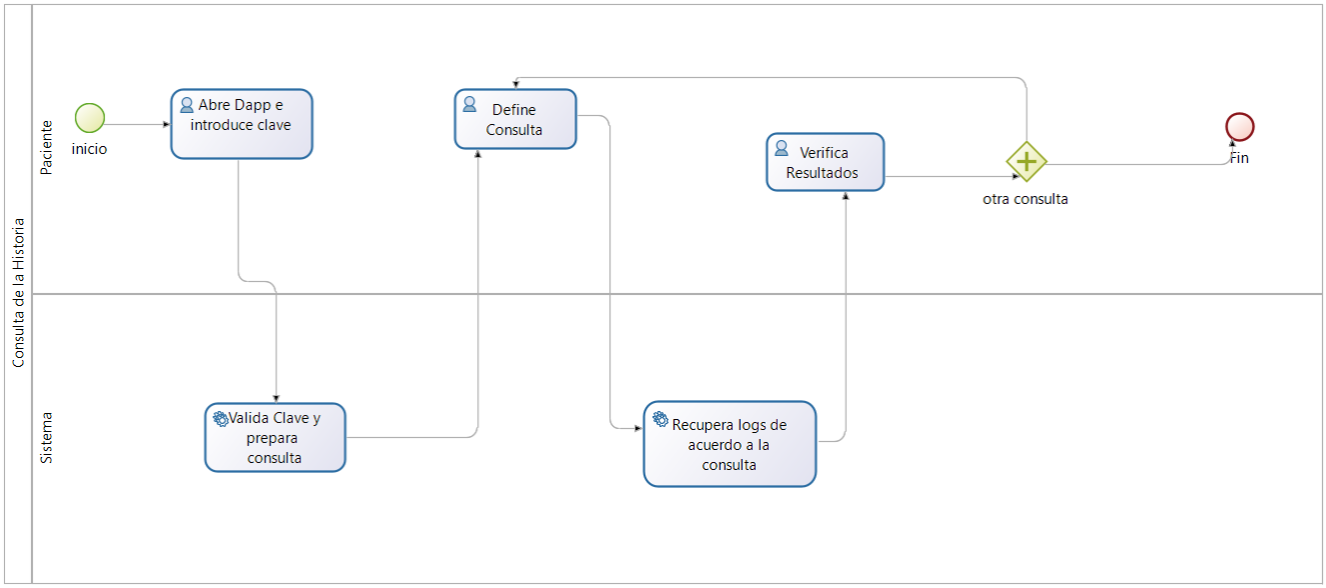


Figura 2 Modelo para consulta de historia médica.

3.3. Modelo para actualizar historia médica

El proceso comienza cuando el usuario abre la Dapp y proporciona su clave de acceso.

El sistema valida la clave del usuario para garantizar la autenticidad de la actualización de los datos. Una vez validada la clave, el usuario define los nuevos datos que desea agregar o modificar en su historia médica.

Luego, el sistema actualiza los datos en la Blockchain, asegurándose de que la información actualizada se almacene de forma segura y confiable. Después de actualizar la información, el sistema genera un nuevo hash, que es una representación única de los datos actualizados. El hash actúa como una especie de "huella digital" que verifica la integridad de los datos almacenados en la Blockchain.

Finalmente, el usuario verifica la actualización realizada en su historia médica. Puede revisar los cambios realizados y compararlos con los datos previos para asegurarse de que la actualización se haya realizado correctamente.

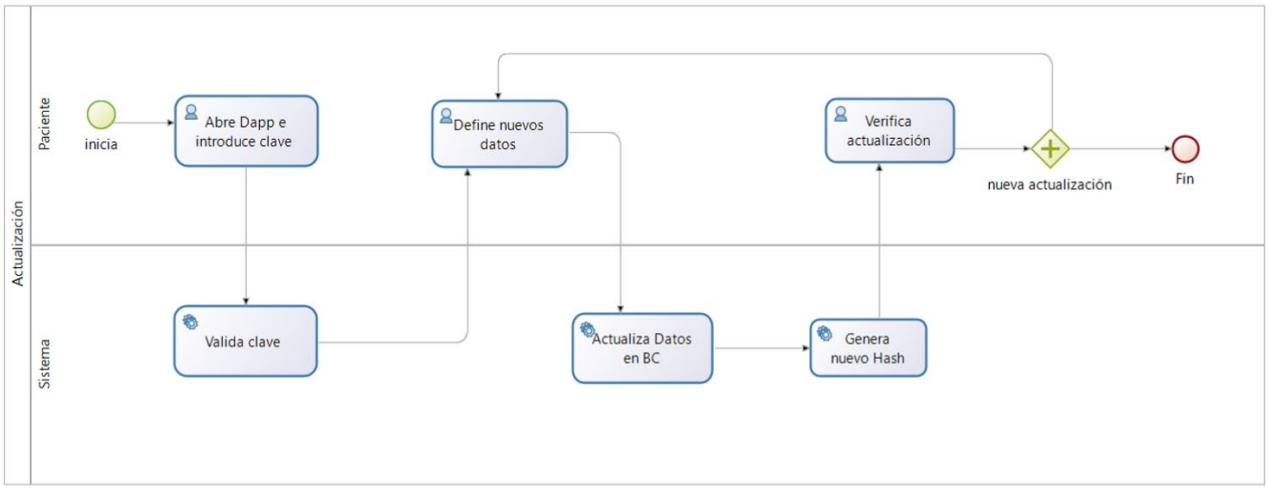


Figura 3 Modelo para actualizar historia médica.

Capítulo 4

Modelo de un sistema de gestión de historias médicas almacenado en una red Blockchain.

El diseño del modelo de gestión de historias médicas almacenado en la red Blockchain, para este trabajo de investigación, se hizo en dos partes. Un modelo de gestión de datos y un modelo de sistema de carga de archivos.

En la primera parte, se desarrolló un modelo para almacenar datos relevantes de las historias médicas, como el nombre del paciente, su edad, peso, síntomas y medicación. Estos datos son considerados fundamentales para la gestión de la historia médica y son gestionados mediante un contrato inteligente. El uso de un contrato inteligente permite que estos datos sean almacenados de forma descentralizada en la red Blockchain. Además, como se explica anteriormente, el contrato inteligente brinda la capacidad de establecer reglas y condiciones específicas para el acceso y actualización de estos datos, asegurando la integridad y la confidencialidad de la información médica.

En la segunda parte, se desarrolló un modelo de sistema para cargar archivos utilizando tecnologías como IPFS (InterPlanetary File System), web3-react e INFURA. Este modelo permite que se carguen y almacenen archivos adicionales relacionados con las historias médicas, como informes médicos, resultados de pruebas o imágenes médicas. El uso de IPFS garantiza que los archivos se almacenen de forma descentralizada y se puedan acceder de manera eficiente y segura. La integración de web3-react e INFURA permite la conexión con la red Blockchain y el acceso a los contratos inteligentes para la gestión de los archivos.

Ambos modelos se complementan para formar un sistema completo de gestión de historias médicas en la red Blockchain. Juntos, proporcionan una solución integral para la gestión segura,

confiable y eficiente de las historias médicas en un entorno descentralizado basado en la tecnología Blockchain.

4.1. Modelo de gestión de historias médicas electrónicas

Para el modelo de gestión, se desarrolló un contrato inteligente utilizando el lenguaje de programación Solidity y se probó en el entorno de desarrollo integrado Remix que permite compilar los códigos escritos en Solidity y realizar pruebas locales. Una vez que el contrato se ha compilado y funciona correctamente, se procede a desplegarlo en la red blockchain seleccionada. En este proyecto, las pruebas se llevaron a cabo en la red blockchain Ethereum.

El objetivo del contrato inteligente era establecer una estructura que permitiera crear una clave privada y almacenar datos como nombre, fecha, peso, enfermedad y tratamiento. Para realizar las pruebas en Remix, se necesitaba un proveedor de criptomonedas que permitiera trabajar en la red blockchain de Ethereum. Se utilizó Ganache, un simulador local de pruebas de Ethereum, que proporciona diez direcciones ficticias con un saldo de 100 ethers cada una. Estas direcciones nos permitieron realizar las pruebas necesarias y descontar el saldo correspondiente.

Home historia.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.16;

contract Historia {

    struct Privado {
        string nombre;
        uint clave;
    }

    struct Publico {
        uint fecha;
        uint peso;
        string enfermedad;
        string tratamiento;
    }

    struct Persona { //estructura con la información de cada persona
    address direccion;
        Privado id;
    }
```

Figura 4 Segmento de código parte 1

Una transacción en el contexto de despliegue de un contrato en la blockchain se refiere a una operación específica que implica el registro y ejecución de dicho contrato en la red. En este caso, la transacción se lleva a cabo utilizando la plataforma Remix.

Cuando se configura y se despliega el contrato en la blockchain, se inicia una transacción que implica el envío y procesamiento de información relacionada con el contrato. La transacción es generada por el usuario o por la plataforma Remix y se registra en la red blockchain. Durante este proceso, se deben indicar ciertos parámetros, como el proveedor de criptomonedas a utilizar (en este caso, Ganache).

Remix se encarga de gestionar el consumo de Gas necesario para la transacción. El Gas es una unidad que representa el costo computacional requerido para ejecutar operaciones en la Blockchain. En otras palabras, es la tarifa que se paga por utilizar los recursos de la red. Una vez que se completa la transacción, la consola devuelve información relevante, como el hash del contrato (una identificación única del contrato en la blockchain), su estado (confirmado, pendiente, etc.), la dirección desde la cual se realizó la transacción y la dirección de destino (donde se desplegó el contrato).

Los eventos, que son señales enviadas por los contratos inteligentes, pueden ser escuchados y respondidos por las dapps conectadas a Ethereum. Los eventos se declaran dentro del código del contrato y se emiten mediante la palabra clave **emit**. Cuando se llama a un evento desde el contrato inteligente, los argumentos se almacenan en un registro especial de la transacción. Esta estructura de datos se asocia a la dirección del contrato y se guarda en la blockchain, permaneciendo allí mientras sea accesible en el bloque. Al desplegar el contrato y llamar a la función correspondiente, Remix mostrará automáticamente la información relacionada con los eventos emitidos.

Cuando el usuario, quien es el paciente y propietario de la información, requiera actualizar los datos de su historia médica, tendrá la capacidad de hacerlo de manera segura y controlada. Para ello, le permitirá al profesional de la salud acceder a la información necesaria para realizar las actualizaciones pertinentes. Esta acción generará una nueva transacción en la dirección del contrato inteligente correspondiente a la historia médica del paciente.

La transacción realizada por el profesional de la salud implicará la modificación de los datos necesarios en la historia médica del paciente. Esta actualización se reflejará en la Blockchain y generará un nuevo hash, que es una representación única y segura de la información actualizada. El nuevo hash garantiza la integridad y la trazabilidad de los cambios realizados en la historia médica.

Este proceso permite que el paciente, como propietario de la información, mantenga el control sobre los datos de su historia médica y autorice las actualizaciones mediante el uso de su clave privada. De esta manera, se asegura la confidencialidad de la información y se evita cualquier modificación no autorizada. Al generar una nueva transacción y obtener un nuevo hash con los datos actualizados, se establece un registro confiable y verificable de los cambios realizados en la historia médica.

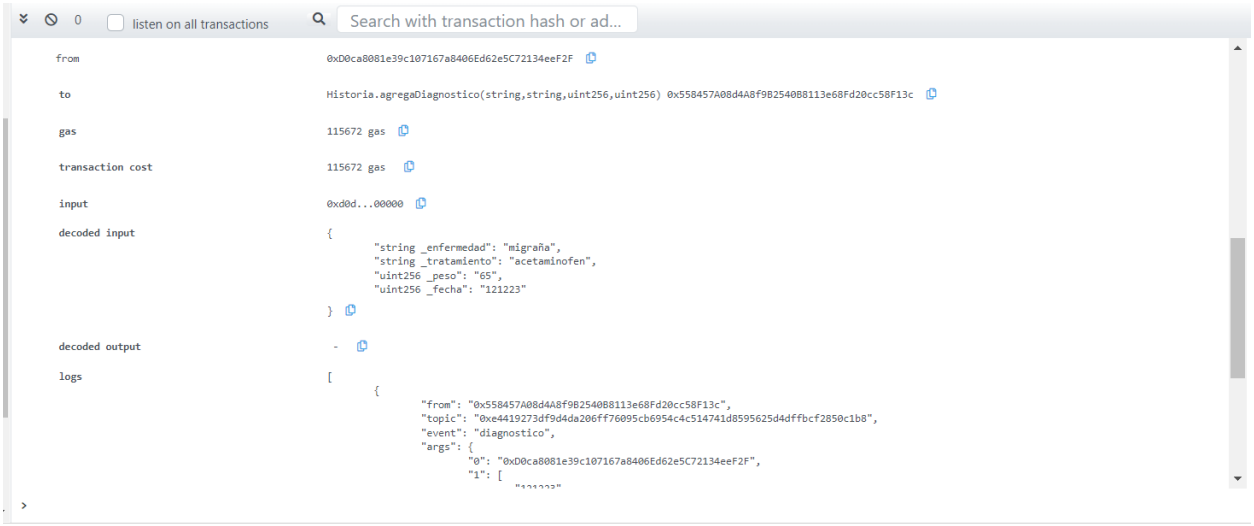


Figura 5 Eventos tomados luego de hacer Deploy al contrato

Al ser Remix una plataforma de desarrollo en línea que ofrece un entorno integrado para realizar pruebas exhaustivas y diversas al contrato inteligente previamente desarrollado, brinda la posibilidad de realizar una serie de pruebas y simulaciones del contrato inteligente sin la necesidad de conectarlo directamente a una Dapp. Esto significa que se puede verificar la funcionalidad y el comportamiento del contrato en un entorno controlado y seguro antes de implementarlo en una aplicación real.

El código completo de este modelo se mostrará en la sección de anexos.

4.2. Modelo para cargar archivos

Como ya se dijo, para desarrollar el modelo de la aplicación de carga de archivos, se utilizaron varias tecnologías y herramientas como IPFS, web3-react e INFURA, en conjunto con el sistema operativo LinuxMint 20.

El proceso comenzó instalando la plantilla de TypeScript, que proporciona un entorno de desarrollo adecuado para el uso de este lenguaje de programación. Luego, se procedió a instalar las dependencias de Node.js, que es el entorno de ejecución de JavaScript utilizado para el desarrollo de aplicaciones. Estas dependencias permitieron establecer la conexión con IPFS, una red descentralizada de almacenamiento de archivos.

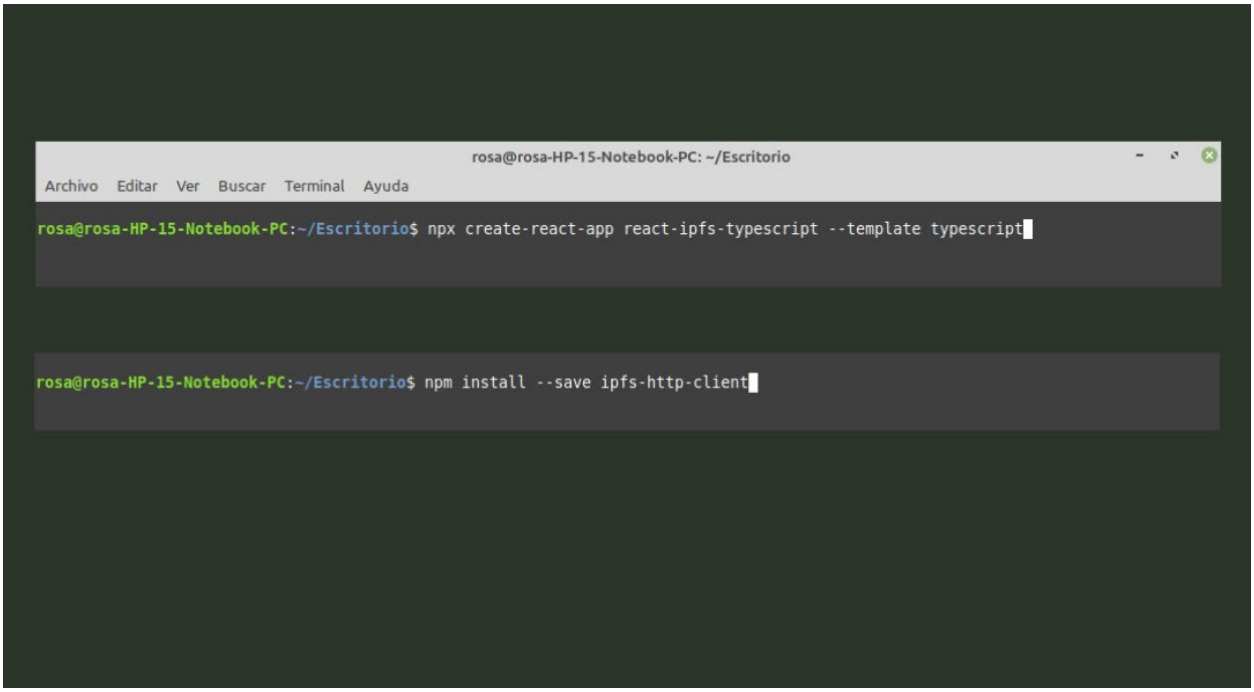


Figura 6 Instalación de React

Específicamente, se instaló el cliente HTTP de IPFS, que es necesario para interactuar con la red y cargar archivos en ella. Sin este cliente, sería imposible realizar la carga de archivos, ya que IPFS requiere que las operaciones se realicen a través de HTTP o en el entorno local (localhost).

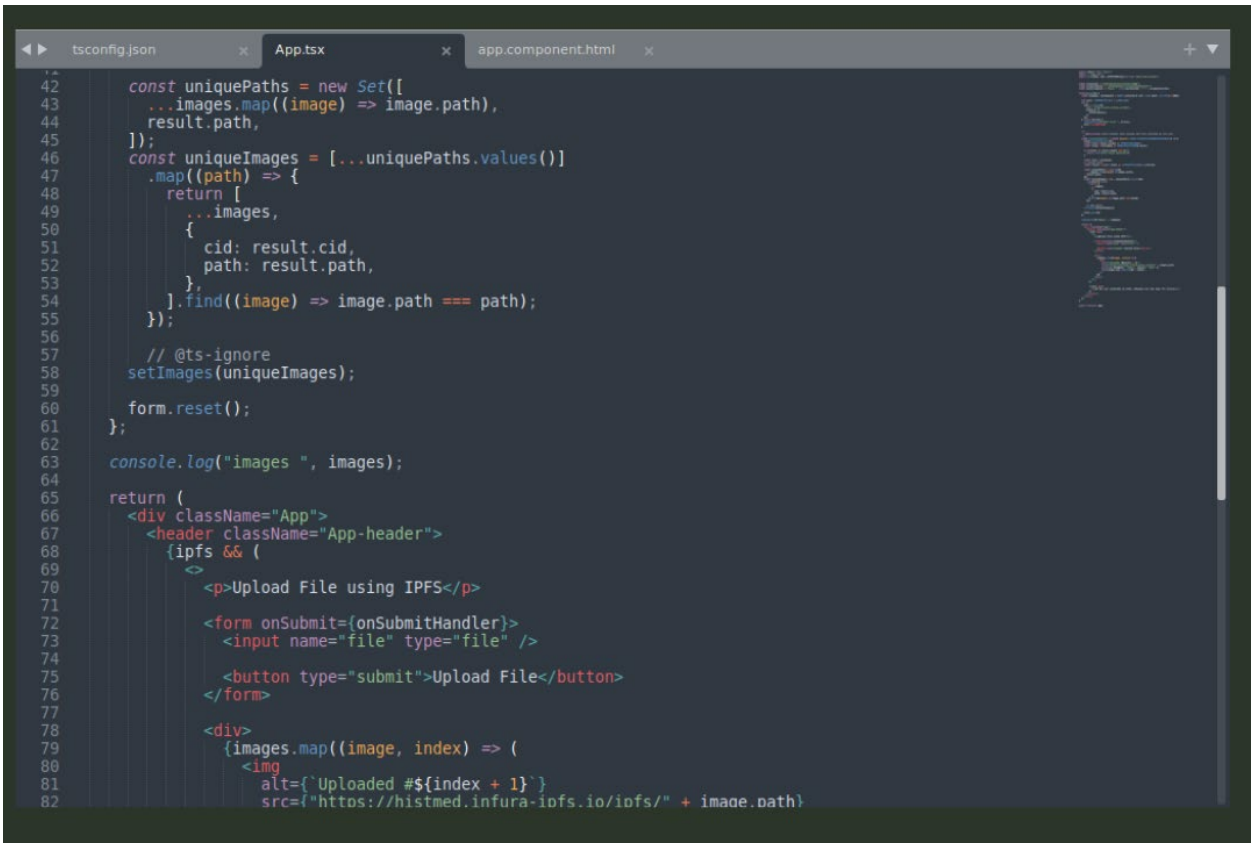


Figura 7 Código React

Además, se utilizó web3-react, una biblioteca que facilita la conexión con la red de Ethereum y la interacción con contratos inteligentes. Esto permitió integrar la funcionalidad de la aplicación de carga de archivos con la blockchain de Ethereum.

Para garantizar la conectividad y el acceso a la red Ethereum, se utilizó INFURA, un servicio que proporciona una API para interactuar con la red sin necesidad de ejecutar un nodo completo de Ethereum. Esto simplifica el proceso de desarrollo y permite centrarse en la implementación de la funcionalidad principal de la aplicación.

En primer lugar, la aplicación realizará una verificación para asegurarse de que el usuario ha seleccionado un archivo antes de continuar con el proceso. Esto evita que se realicen operaciones innecesarias o no deseadas en caso de que no se haya seleccionado ningún archivo.

Una vez que se ha confirmado la selección del archivo, la aplicación utilizará IPFS para cargar dicho archivo en la red descentralizada. IPFS es un sistema que permite almacenar y compartir archivos de forma distribuida y segura. Al cargar el archivo en IPFS, se generará un CID, que es

un identificador único que se asigna al archivo cargado. El CID permite acceder al archivo posteriormente de manera segura y confiable.

El código completo de este modelo se mostrará en la sección de anexos.

Teniendo lista la aplicación, se procedió a probarla creando un proyecto en INFURA, para poder crearlo es necesario ejecutar algunos pasos:

1. Se debe registrar un usuario en INFURA.io: De esta manera se obtendrá una cuenta que permitirá configurar un proyecto. La configuración de proyectos IPFS en INFURA ofrece de manera gratuita, pero con ciertas condiciones [5Gb de almacenamiento gratuito].

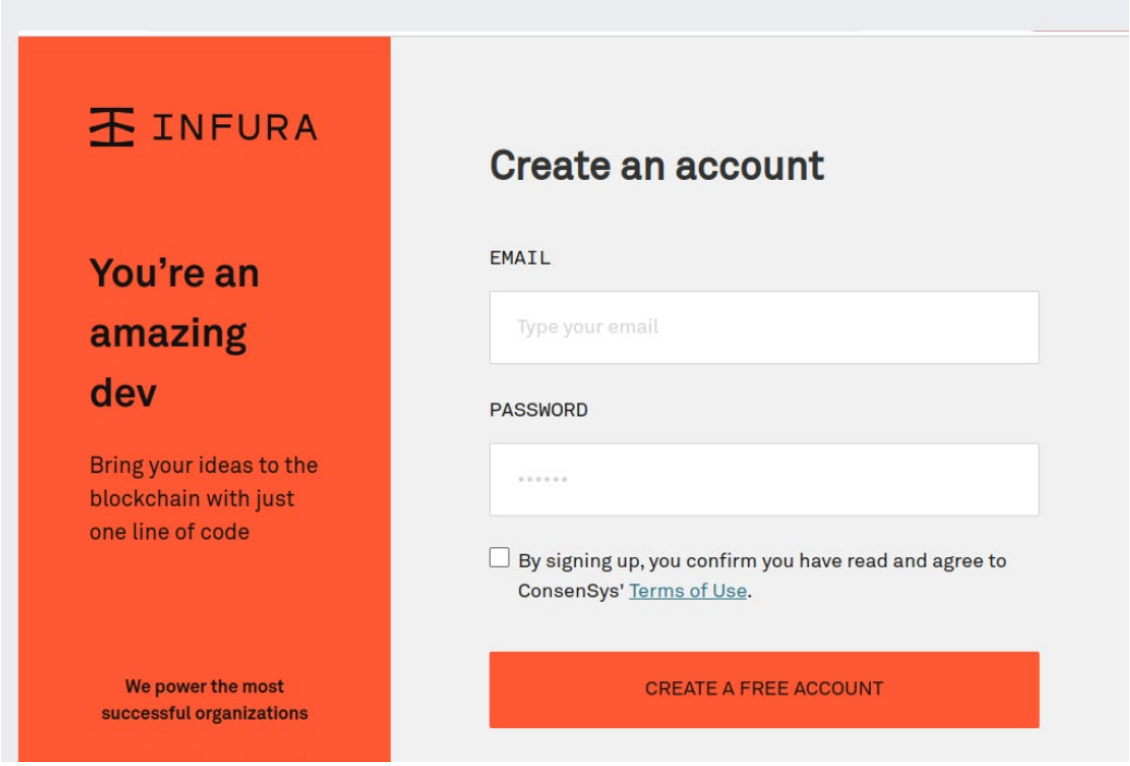


Figura 8 Crear cuenta INFURA

La principal condición de INFURA para configurar este tipo de proyectos es el requerimiento de una tarjeta de crédito activa y vigente. Para este proyecto no se contaba con este requerimiento ya que, por circunstancias de causa mayor, los bancos nacionales no ofrecen este servicio. Para solventar esta situación, se utilizó una aplicación llamada Zinli, que es una billetera virtual que te permite tener una Tarjeta Zinli Visa Virtual prepagada, pudiendo así obtener un número de

tarjeta real con una fecha de vencimiento. Ya que INFURA no solicita ningún pago antes de ocupar los 5Gb de almacenamiento, no fue necesario prepagar esta tarjeta.

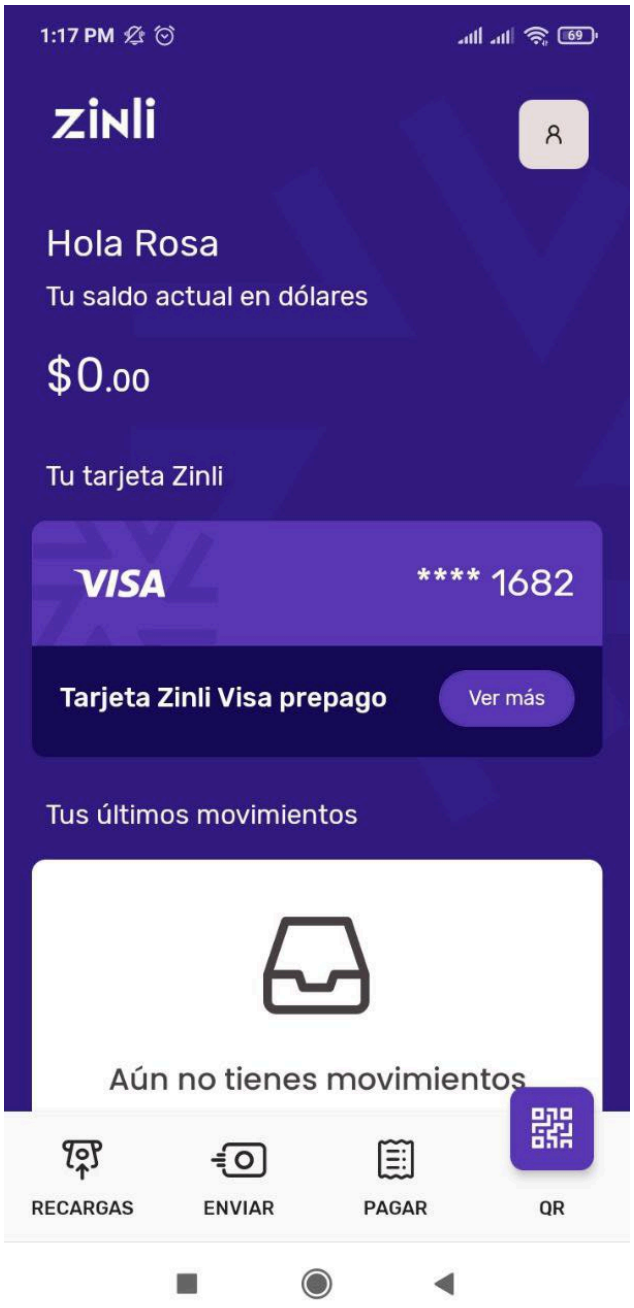


Figura 9 Captura 1, plataforma Zinli

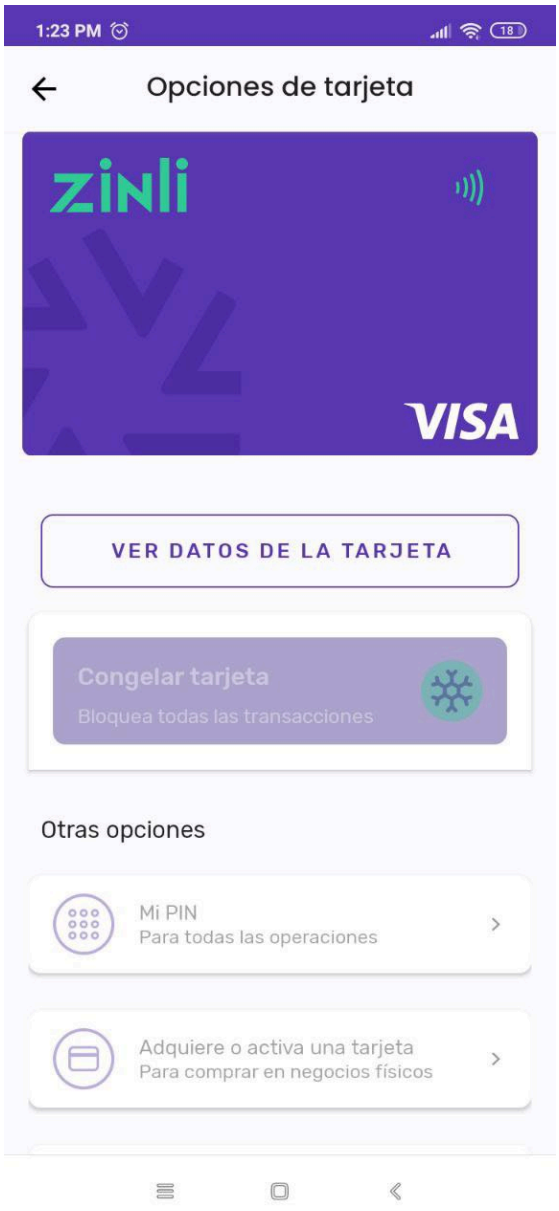


Figura 10 Captura 2, plataforma Zinli

Figura 11 Modelo para cargar archivos

2. En algunos casos navegadores como google chrome o firefox, restringen las solicitudes HTTP de origen cruzado que se inician desde secuencias de comandos que se ejecutan en el navegador, por lo cual es necesario instalar extensiones para que permita el acceso de estas solicitudes, en este caso se instaló para google chrome la extensión CORS.
3. Se instaló la extensión React Developer Tools para Google Chrome una herramienta para la biblioteca React JavaScript. Esta extensión agrega herramientas de depuración de React a las herramientas de desarrollo de Chrome permitiendo que la conexión entre la aplicación e INFURA funcione perfectamente.

Teniendo listos los pasos anteriores y configurado el proyecto de IPFS, INFURA nos proporciona ciertos datos que debemos incluir en el código del proyecto para poderlo conectar con INFURA.io, estos datos son:

- ID del proyecto
- Código del PROJECT SECRET
- SUBDOMINIO DE PUERTA DE ENLACE DEDICADA
- IPFS API ENDPOINT

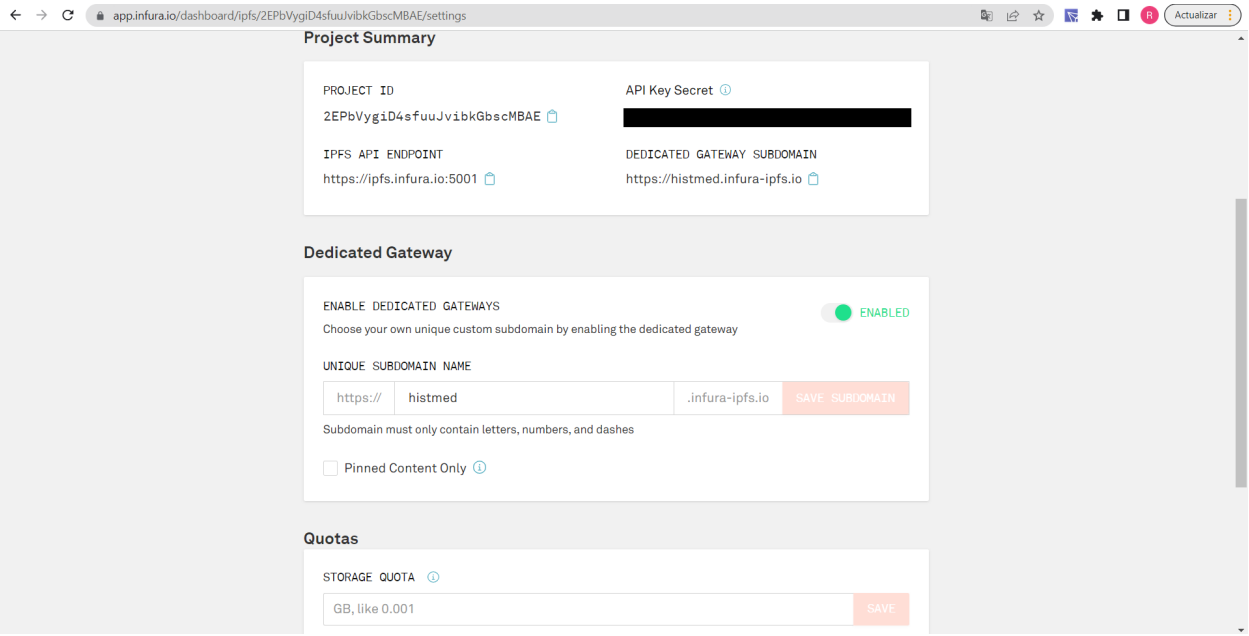


Figura 12 App INFURA

4. Configurado el proyecto pudimos cargar archivos. Las rutas se pueden ver desde la interfaz de INFURA.

Capítulo 5

Factibilidad económica del sistema de gestión de historias médicas almacenado en la red Blockchain

El White/Yellow Paper de Ethereum proporciona los fundamentos y conceptos básicos para comprender cómo se realizan las transacciones en la red de Ethereum. El costo de una transacción se calcula considerando varios parámetros específicos de la blockchain, estos parámetros incluyen el base fee o valor base, el gas utilizado, el Max Priority Fee y el valor actual de la moneda de Ethereum, el ether.

El cálculo total de la comisión por transacción se realiza multiplicando las unidades de gas utilizadas por la suma del base fee y el Max Priority fee. Es importante destacar que el valor del ether puede fluctuar en el mercado, y en este caso, se tomó el valor referencial del día 10 de enero de 2023 el cual era de 1600 dólares según el indicador coinmarketcap.com

El gas utilizado se obtuvo a través de una simulación realizada al compilar y desplegar el contrato inteligente utilizando las herramientas de Remix y Ganache, las cuales permiten probar y simular el funcionamiento del contrato inteligente en un entorno controlado.

Es importante acotar que las fees en Ethereum pueden variar según diversos factores, como la congestión de la red, la complejidad de la transacción y los parámetros específicos utilizados en el cálculo de las comisiones. Por lo tanto, las fees no son siempre las mismas y pueden fluctuar.

La primera transacción que se calculó es la de asignar un nuevo usuario y clave:

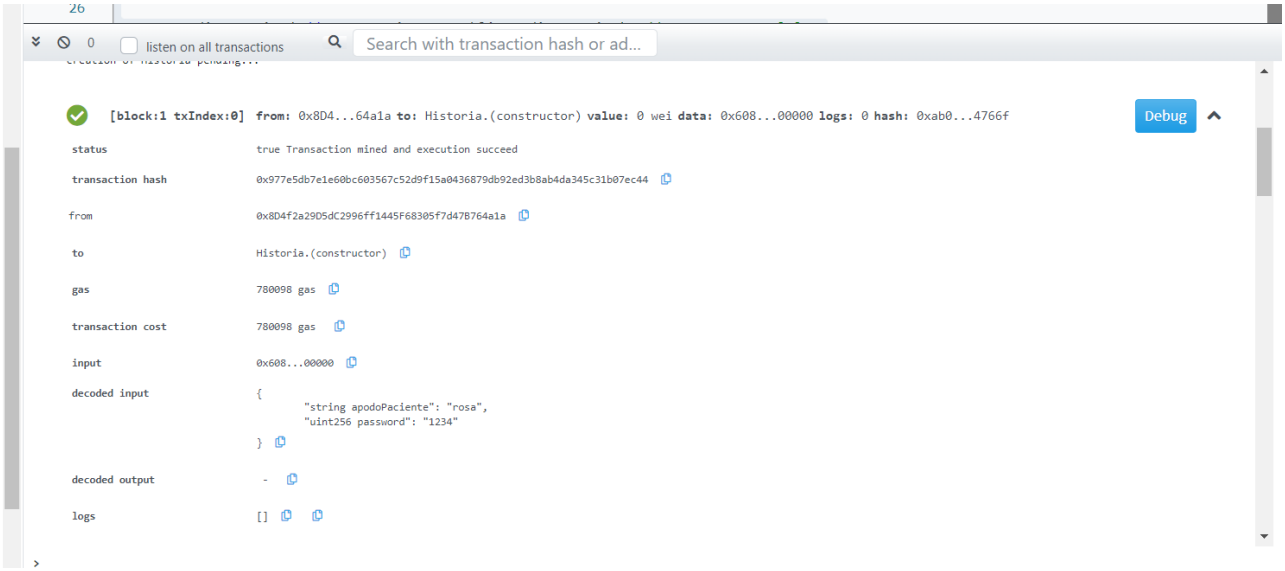


Figura 13 Consola Remix

GAS = 780098.

Base fee = 20 Gwei según el estimador virtual

Max Priority Fee = 2 Gwei

Valor del Ethereum para el 10 de enero de 2023 = 1600$

$$(20 + 2) * 780098 = 17{,}162{,}156\ Gwei$$

$$17{,}162{,}156\ Gwei = 0,\ 017162156\ Ether.$$

Por lo tanto, el coste de la transacción es de **0,017162156** Ether.

Si lo convertimos a dólares,

$$0,\ 017162156\ Ether * 1600\$ = 27,\ 45\$$$

La segunda transacción se calculó para agregar los datos de la historia médica:

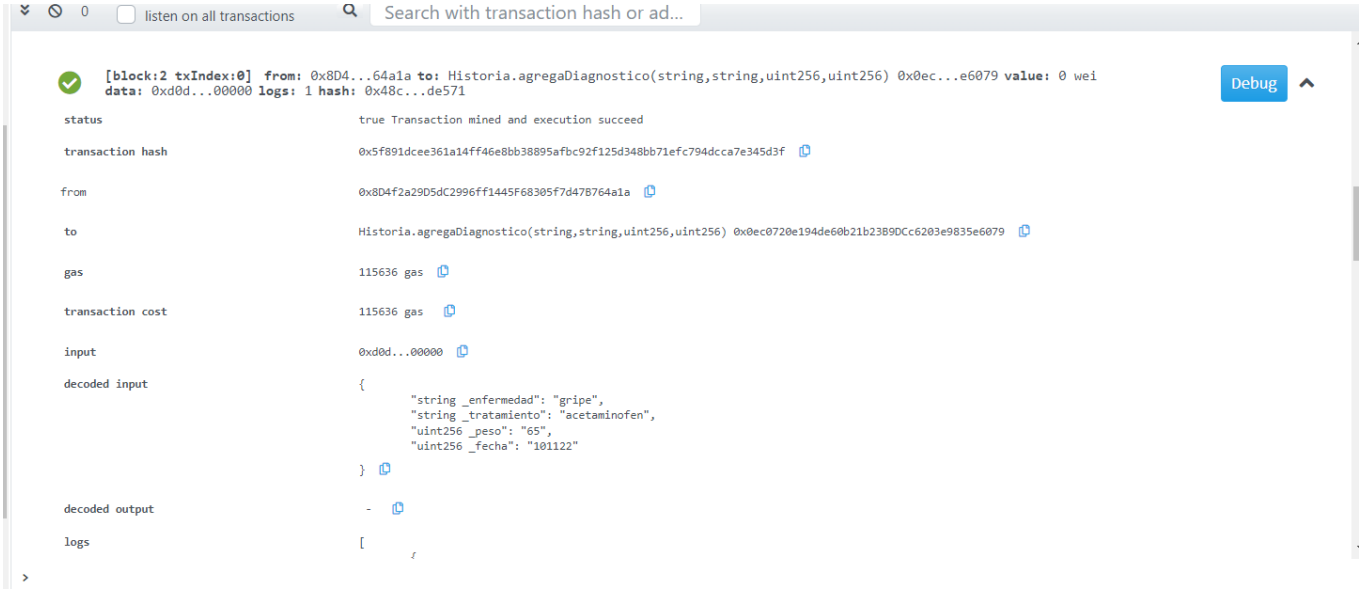


Figura 14 Consola Remix 2

GAS = 115636

Base fee = 20 Gwei según el estimador virtual

Max Priority Fee = 2 Gwei

Valor del Ethereum para el 10 de enero de 2023 = 1600$

$$(20 + 2) * 115636 = 2{,}543{,}992\ Gwei$$

$$2{,}543{,}992\ Gwei = 0,\ 002543992\ eth$$

Por lo tanto, el coste de la transacción es de **0,002543992** Ether

Si lo convertimos a dólares**,**

$$0,\ 002543992\ Ether * 1600\$ = 4,\ 07\$$$

El costo de una actualización de parámetros en la historia médica:

En este caso hemos actualizado el valor del peso.

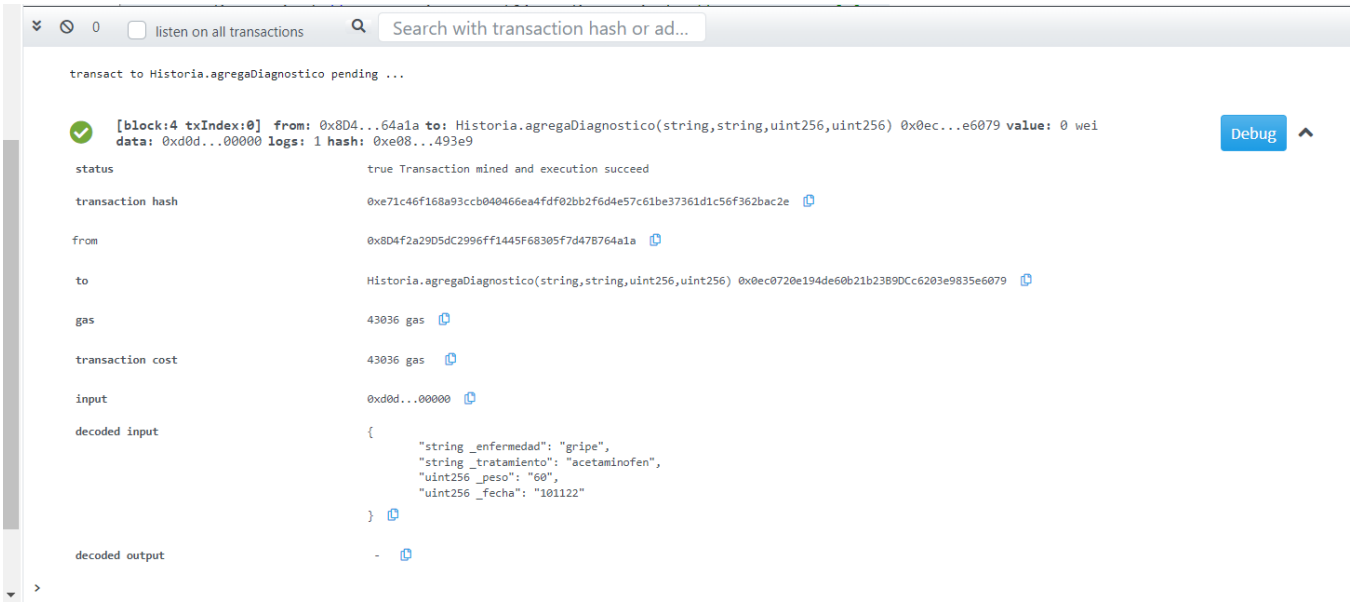


Figura 15 Consola Remix 3

GAS = 43036

Base fee = 20 Gwei según el estimador

Max Priority Fee = 2 Gwei

Valor del Ethereum para el 10 de enero de 2023 = 1600$

$$(20 + 2) * 43036 = 946{,}792\ Gwei$$

$$946{,}792\ Gwei = 0,\ 000946792\ eth$$

Por lo tanto, el coste de la transacción es de 0,000946792 Ether.

Si lo convertimos a dólares,

$$0,\ 000946792\ Ether * 1600\$ = 1,\ 51\$$$

Si solo se desea consultar la historia, haremos un llamado a los logs, que se crean cuando en el contrato hemos programado emitir eventos. Estos serán mucho más baratos ya que la cantidad de gas que se consume está estipulada en el Yellow Paper de Ethereum y tiene un valor de 375 gas.

GAS = 375

Base fee = 20 Gwei según el estimador virtual

Max Priority Fee = 2 Gwei

Valor del Ethereum para el 10 de enero de 2023 = 1600$

$$(20 + 2) * 375 = 8,250\ Gwei$$

$$\mathbf{8,250\ Gwei = 0,\ 00000825\ eth}$$

Por lo tanto, el coste de la transacción es de:

$$\mathbf{0,\ 00000825 Ether}$$

Si lo convertimos en dólares,

$$\mathbf{0,\ 00000825\ Ether * 1600\$ = 0,\ 0132\$.}$$

Observamos que el costo más alto, se presenta al crear un usuario nuevo, pero las transacciones comunes como, agregar, actualizar y consultar son económicas en comparación con la primera y única transacción de crear un usuario.

Tipo de Transacción	Precio $
Nuevo Usuario	24.45$
Agregar Datos	4.07$
Actualización de Datos	1.51$
Consultar Historia	0.0132$

Tabla 1 Tabla comparadora de precios

El concepto de "GAS" en Ethereum es esencial para comprender los costos asociados con las transacciones y la ejecución de contratos inteligentes en la plataforma, ya que sirve como una medida de trabajo o cálculo. Cada operación, ya sea una simple transferencia de Ether o la ejecución de un contrato inteligente complejo, requiere una cantidad específica de gas. El gas es esencialmente una forma de compensar a los mineros por el trabajo computacional y los recursos que emplean para validar y registrar las transacciones en la blockchain. El precio del gas es la cantidad de Ether que estás dispuesto a pagar por cada unidad de gas. No es un valor fijo, varía

según la demanda de cómputo en la red y cuánto están dispuestos a pagar los usuarios. En períodos de alta demanda o congestión, los usuarios tienden a ofrecer precios de gas más altos para que sus transacciones sean priorizadas y procesadas más rápidamente, lo que eleva el costo general para todos en la red.

En cuanto al modelo para cargar archivos que se realizó, se utilizó la tecnología IPFS. Esta nos permite el almacenamiento de datos pesados de forma descentralizada, basándose en las redes entre pares. IPFS es de código abierto, así que podemos usar distintas herramientas y plataformas para subir archivos.

Para este proyecto se utilizó la plataforma de INFURA que permitió alojar el proyecto para subir imágenes al sistema de archivos descentralizados IPFS.

5.1. Facturación de INFURA

El tráfico de la puerta de enlace dedicada, INFURA lo factura al mismo precio que el acceso a IPFS a través de la API de INFURA:

- 5GB de almacenamiento gratis
- Transferencia de para subida datos de 5GB gratis
- Transferencia de para datos descarga de 5GB gratis

Todo lo que supere ese límite lo cobrará en:

- Almacenamiento ilimitado a $0.08/GB
- Transferencia de datos ilimitada a $0.12/GB

Actualmente, para esta versión, la facturación de INFURA no se divide en API y tráfico de puerta de enlace. En las facturas, tanto el tráfico de la API como el de la puerta de enlace contarán como ancho de banda de descarga.

Podemos inferir que, dada la creciente disponibilidad tanto en los servicios de internet como en equipos e instrumentos tecnológicos, la viabilidad y factibilidad tecnológica de un proyecto como el de almacenamiento de historias médicas sobre la red Blockchain no puede sino

aumentar, ya que demás de la Blockchain de Ethereum, hay muchas otras que están desarrollando la funcionalidad de contratos inteligentes de mayor complejidad.

5.2. Análisis de sensibilidad

En el estudio de análisis de sensibilidad se consideraron dos escenarios. En el primer escenario, se analizó la variación del base fee. El base fee es determinado por la red Ethereum y no por los usuarios finales que desean realizar transacciones o los mineros que validan las transacciones.

El **Base Fee** objetivo es un 50% de bloques completos y se basa en el contenido del bloque confirmado más reciente. Dependiendo de qué tan lleno esté ese nuevo bloque, Base Fee automáticamente aumenta o disminuye.

En el segundo escenario, se examinó la variación del **Max Priority Fee**. A diferencia del base fee, el **Max Priority Fee** es opcional y es establecido por el usuario. Este fee se paga directamente a los mineros para priorizar la inclusión de la transacción en un bloque de la cadena de bloques.

Estos dos escenarios permitieron evaluar el impacto de las variaciones en los costos de transacción en Ethereum y entender cómo el **Base fee** y el **Max Priority Fee** pueden influir en el costo total de una transacción.

Para los dos escenarios se varió el valor de Ethereum, fijando un precio de **1860$** por cada unidad de ether.

5.2.1 Escenario 1

En el escenario 1, se experimentó con una característica específica de la red Ethereum relacionada con el "Base Fee". El Base Fee es establecido por la propia red Ethereum. Durante la

prueba, se llevaron a cabo transacciones modificando este valor, es decir, incrementándolo y reduciéndolo, para observar los efectos y comportamientos resultantes.

5.2.1.1 Aumentando el Base Fee a 26 Gwei

- **Asignar un nuevo usuario y clave:**

  Datos para la transacción:

  GAS = 780098

  Base fee = 26 Gwei

  Max Priority Fee = 2 Gwei

  Valor del Ethereum = 1860$

$$(26 + 2) * 780098 = 21,842,744 \text{ Gwei}$$

$$\textbf{21,842,744 Gwei = 0, 021842744 Ether.}$$

Por lo tanto, el coste de la transacción es de:

$$\textbf{0, 021842744 Ether}$$

Si lo convertimos en dólares,

$$\textbf{0, 021842744 Ether} * \textbf{1860\$} = \textbf{40, 62\$}$$

- **Agregar los datos de la historia médica:**

  Datos para la transacción:

GAS = 115636

Base fee = 26 Gwei

Max Priority Fee = 2 Gwei

Valor del Ethereum = 1860$

(26 + 2) ∗ 115636 = 3,237,808 Gwei

3,237,808 Gwei = 0, 003237808 Ether

Por lo tanto, el coste de la transacción es de

0, 003237808 Ether

Si lo convertimos en dólares,

0, 003237808 Ether ∗ 1860$ = 6, 02$

- **Actualización de parámetros en la historia médica:**

Supongamos que en esta transacción hemos se actualiza la variable peso del paciente:

Datos para la transacción:

GAS = 43036

Base fee = 26 Gwei

Max Priority Fee = 2 Gwei

Valor del Ethereum = 1860$

43036 ∗ (26 + 2) = 1,205,008 Gwei

1,205,008 Gwei = 0, 001205008 Ether

Por lo tanto, el coste de la transacción es de:

0, 001205008 Ether.

Si lo convertimos en dólares,

0, 001205008 Ether ∗ 1860$ = 2, 24$

- **Consultar Historia**

Si solo se desea consultar la historia, haremos un llamado a los logs, que se crean cuando en el contrato hemos programado emitir eventos. Estos serán mucho más baratos ya que la cantidad de gas que se consume está estipulada en el Yellow Paper de Ethereum y tiene un valor de 375 gas.

Datos para la transacción:

GAS = 375

Base fee = 26 Gwei

Max Priority Fee = 2 Gwei

Valor del Ethereum = 1860$

$$(26 + 2) * 375 = 10{,}500 \text{ Gwei}$$

$$10{,}500 \text{ Gwei} = 0{,}0000105 \text{ Ether}$$

Por lo tanto, el coste de la transacción es de:

$$0{,}0000105 \text{ Ether}$$

Si lo convertimos en dólares,

$$0,\ 0000105 \text{ Ether} * 1860\$ = 0,\ 01953\$.$$

5.2.1.2 Disminuyendo el Base Fee a 19

- **Asignar un nuevo usuario y clave:**

Datos para la transacción:

GAS = 780098

Base fee = 19 Gwei

Max Priority Fee = 2 Gwei

Valor del Ethereum = 1860$

$$(19 + 2) * 780098 = 16{,}382{,}058 \text{ Gwei}$$

16,382,058 Gwei = 0, 016382058 Ether.

Por lo tanto, el coste de la transacción es de:

0, 016382058 Ether

Si lo convertimos en dólares,

0, 016382058 Ether ∗ 1860$ = 30, 47$

- ## Agregar los datos de la historia médica:

  Datos para la transacción:

  GAS = 115636

  Base fee = 19 Gwei

  Max Priority Fee = 2 Gwei

  Valor del Ethereum = 1860$

  (19 + 2) ∗ 115636 = 2,428,356 Gwei

  2,428,356Gwei = 0, 002428356 Ether

Por lo tanto, el coste de la transacción es de:

0, 002428356Ether

Si lo convertimos en dólares,

0, 002428356 Ether ∗ 1860$ = 4, 51$

- ## Actualización de parámetros en la historia médica:

  Supongamos que en esta transacción hemos se actualiza la variable peso del paciente:

  Datos para la transacción:

  GAS = 43036

  Base fee = 19 Gwei

  Max Priority Fee = 2 Gwei

Valor del Ethereum = 1860$

$$(19 + 2) * 43036 = 903,756Gwei$$

903,756 Gwei = 0, 000903756 Ether

Por lo tanto, el coste de la transacción es de:

0, 000903756 Ether.

Si lo convertimos en dólares,

0, 000903756 Ether * 1860$ = 1, 68$

- ## Consultar historia:

GAS = 375

Base fee = 19 Gwei

Max Priority Fee = 2 Gwei

Valor del Ethereum = 1860$

$$375 * (19 + 2) = 7,875 Gwei$$

7,875 Gwei = 0, 000007875 Ether

Por lo tanto, el coste de la transacción es de:

0, 0000105 Ether

Si lo convertimos en dólares,

0, 000007875 Ether * 1860$ = 0, 0146475$.

5.2.2 Escenario 2

Para el escenario 2 dejaremos el cambio que se hizo en el escenario 1 y variaremos también el Max Priority Fee que es la cantidad máxima de gas que se incluirá como recompensa para el validador.

5.2.2.1 Aumentando el Base Fee a 26 y cambiando el Max Priority Fee a 1 Gwei

- **Asignar un nuevo usuario y clave:**

  Datos para la transacción:

  GAS = 780098

  Base fee = 26 Gwei

  Max Priority Fee = 1 Gwei

  Valor del Ethereum = 1860$

Si hacemos el cálculo de esta transacción sería:

$$(26 + 1) * 780098 = 21{,}062{,}646\ Gwei$$

21,062,646Gwei = 0, 021062646Ether.

Por lo tanto, el coste de la transacción es de:

0, 021062646 Ether

Si lo convertimos en dólares,

0, 021062646Ether * 1860$ = 39, 17$

- **Agregar los datos de la historia médica**

  Datos para la transacción:

  GAS = 115636

  Base fee = 26 Gwei

Max Priority Fee = 1 Gwei

Valor del Ethereum = 1860$

$$(26 + 1) * 115636 = 3,122,172\ Gwei$$

3,122,172 Gwei = 0,003122172 Ether

Por lo tanto, el coste de la transacción es de

0,003122172 Ether

Si lo convertimos en dólares,

$$0{,}003122172\text{Ether} * 1860\$ = 5,\ 80\$$$

- ## Actualización de parámetros en la historia médica:

En este caso hemos actualizado el valor de la variable peso del paciente:

Datos para la transacción:

GAS = 43036

Base fee = 26 Gwei

Max Priority Fee = 1 Gwei

Valor del Ethereum = 1860$

$$(26 + 1) * 43036 = 1,161,972\ Gwei$$

1,161,972 Gwei = 0,001161972 Ether

Por lo tanto, el coste de la transacción es de:

0,001161972 Ether.

Si lo convertimos en dólares,

0,001161972 Ether * 1860$ = 2,16$

- ## Consultar la historia

Datos para la transacción:

GAS = 375

Base fee = 26 Gwei

Max Priority Fee = 1 Gwei

Valor del Ethereum = 1860$

$$(26 + 1) * 375 = 10{,}125\ Gwei$$

10,125 Gwei = 0,000010125 Ether

Por lo tanto, el coste de la transacción es de:

0,000010125 Ether

Si lo convertimos en dólares,

0,000010125 Ether * 1860$ = 0, 0188325$.

5.2.1.1 Disminuyendo el Base Fee a 19 y cambiando el Max Priority Fee a 1 Gwei.

- **Asignar un nuevo usuario y clave:**

  Datos para la transacción:

  GAS = 780098

  Base fee = 19 Gwei

  Max Priority Fee = 1 Gwei

  Valor del Ethereum = 1860$

$$780098 * (19 + 1) = 15{,}601{,}960\ Gwei$$

15,601,960 Gwei = 0,01560196 Ether.

Por lo tanto, el coste de la transacción es de:

0,01560196 Ether

Si lo convertimos en dólares,

0,01560196Ether ∗ 1860$ = 29, 01$

- ## Agregar los datos de la historia médica:

Datos para la transacción:

GAS = 115636

Base fee = 19 Gwei

Max Priority Fee = 1 Gwei

Valor del Ethereum = 1860$

(19 + 1) ∗ 115636 = 2,312,720 Gwei

2,312,720 Gwei = 0,00231272 Ether

Por lo tanto, el coste de la transacción es de:

0, 00231272 Ether

Si lo convertimos en dólares,

0,00231272 Ether ∗ 1860$ = 4, 30$

- ## Actualización de parámetros en la historia médica:

En este caso hemos actualizado la variable “peso del paciente”.

Datos para la transacción:

GAS = 43036

Base fee = 19 Gwei

Max Priority Fee = 1 Gwei

Valor del Ethereum = 1860$

(19 + 1) ∗ 43036 = 860,720 Gwei

860,720 Gwei = 0,00086072 Ether

Por lo tanto, el coste de la transacción es de:

0,00086072 Ether.

Si lo convertimos en dólares,

0,00086072 Ether ∗ 1860$ = 1, 60$

- **Consultar la historia**

  Datos para la transacción:

  GAS = 375

  Base fee = 19 Gwei

  Max Priority Fee = 1 Gwei

  Valor del Ethereum = 1860$

(19 + 1) ∗ 375 = 7,500 Gwei

7,500Gwei = 0,0000075 Ether

Por lo tanto, el coste de la transacción es de:

0,0000075 Ether

Si lo convertimos en dólares,

0,0000075 Ether ∗ 1860$ = 0, 01395$.

5.2.1.2 Aumentando el Base Fee a 26 y cambiando el Max Priority Fee a 3 Gwei

- **Asignar un nuevo usuario y clave:**

  Datos para la transacción:

  GAS = 780098

  Base Fee = 26 Gwei

  Max Priority Fee = 3 Gwei

  Valor del Ethereum = 1860$

(26 + 3) ∗ 780098 = 22,622,84 2Gwei

22,622,842 Gwei = 0,022622842 Ether.

Por lo tanto, el coste de la transacción es de:

0,022622842 Ether

Si lo convertimos en dólares,

0,022622842 Ether ∗ 1860$ = 42, 07$

- ## Agregar los datos de la historia médica:

  Datos para la transacción:

  GAS = 115636

  Base fee = 26 Gwei

  Max Priority Fee = 3 Gwei

  Valor del Ethereum = 1860$

  (26 + 3) ∗ 115636 = 3,353,444Gwei

  3,353,444 Gwei = 0,003353444 Ether

Por lo tanto, el coste de la transacción es de:

0,003353444 Ether

Si lo convertimos en dólares,

0,003353444 Ether ∗ 1860$ = 6, 23$

- ## Actualización de parámetros en la historia médica:

En este caso hemos actualizado el valor de la variable “peso del paciente”.

Datos para la transacción:

GAS = 43036

Base fee = 26 Gwei

Max Priority Fee = 3 Gwei

Valor del Ethereum = 1860$

$$(26 + 3) * 43036 = 1,248,044 \text{ Gwei}$$

1,248,044 Gwei = 0,001248044 Ether

Por lo tanto, el coste de la transacción es de:

0,001248044 Ether.

Si lo convertimos en dólares,

0,001248044 Ether * 1860$ = 2, 32$

- ## Consultar la historia

  Datos para la transacción:

  GAS = 375

  Base fee = 26 Gwei

  Max Priority Fee = 3 Gwei

  Valor del Ethereum = 1860$

$$(26 + 3) * 375 = 10,875 \text{ Gwei}$$

10,875 Gwei = 0,000010875 Ether

Por lo tanto, el coste de la transacción es de:

0,000010875 Ether

Si lo convertimos en dólares,

0,000010875 Ether * 1860$ = 0,0202275$.

5.2.2.4 Disminuyendo el Base Fee a 19 y cambiando el Max Priority Fee a 1 Gwei

- **Asignar un nuevo usuario y clave:**

  Datos para la transacción:

  GAS = 780098

  Base Fee = 19 Gwei

  Max Priority = 3 Gwei

  Valor del Ethereum = 1860$

Si hacemos el cálculo de esta transacción sería:

$$(19 + 3) * 780098 = 17,162,156\ Gwei$$

17,162,156 Gwei = 0,017162156 Ether.

Por lo tanto, el coste de la transacción es de:

0,017162156 Ether

Si lo convertimos en dólares,

0, 017162156 Ether * 1860$ = 31, 92$

- **Agregar los datos de la historia médica:**

  Datos para la transacción:

  GAS = 115636

  Base fee = 19 Gwei

  Max Priority Fee = 3 Gwei

  Valor del Ethereum = 1860$

$$(19 + 3) * 115636 = 2,543,992\ Gwei$$

2,543,992 Gwei = 0,002543992 Ether

Por lo tanto, el coste de la transacción es de:

0, 00231272 Ether

Si lo convertimos en dólares,

0,002543992 Ether ∗ 1860$ = 4, 73$

- ## Actualización de parámetros en la historia médica:

En este caso hemos actualizado la variable “peso del paciente”.

Datos para la transacción:

GAS = 43036

Base fee = 19 Gwei

Max Priority Fee = 3 Gwei

Valor del Ethereum = 1860$

(19 + 3) ∗ 43036 = 946,792 Gwei

946,792 Gwei = 0,000946792 Ether

Por lo tanto, el coste de la transacción es de:

0,000946792 Ether.

Si lo convertimos en dólares,

0,000946792 Ether ∗ 1860$ = 1,76$

- ## Consultar la historia

Datos para la transacción:

GAS = 375

Base fee = 19 Gwei

Max Priority Fee = 3 Gwei

Valor del Ethereum = 1860$

375 ∗ (19 + 3) = 8,250 Gwei

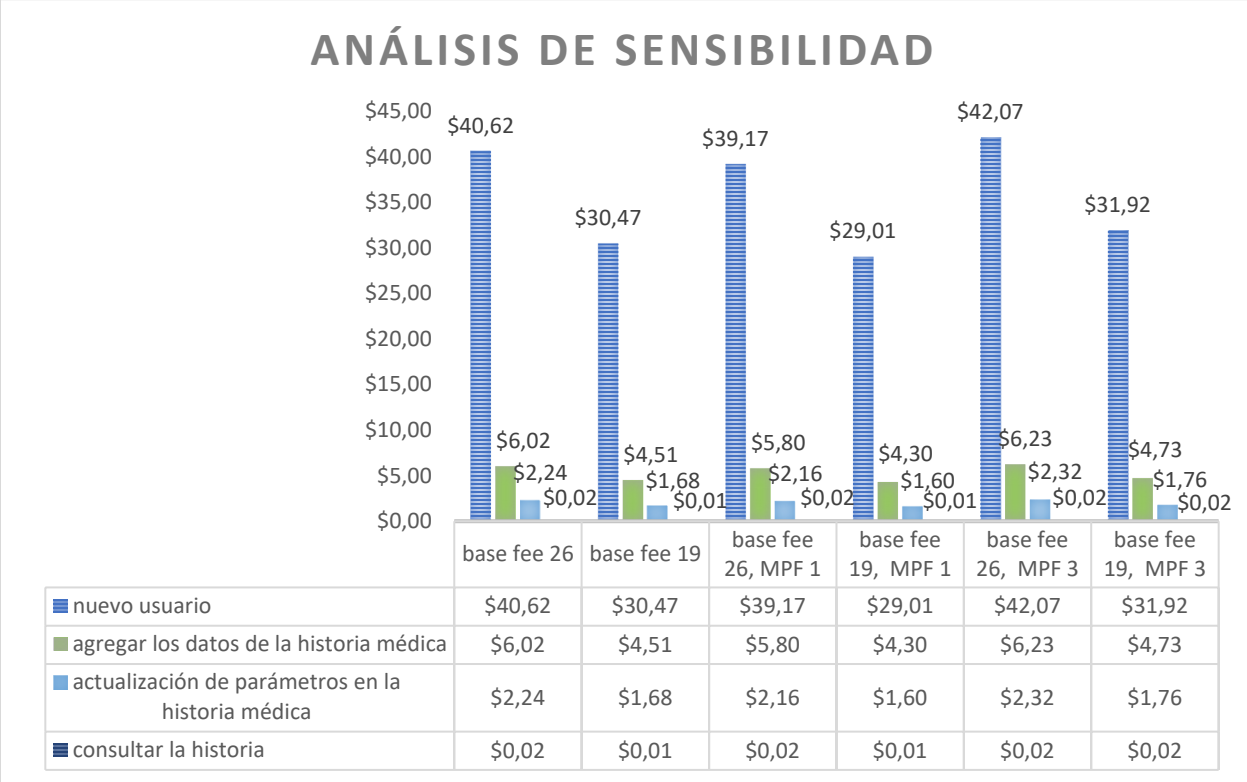
8,250 Gwei = 0,00000825 Ether

Por lo tanto, el coste de la transacción es de:

0,00000825 Ether

Si lo convertimos en dólares,

0,00000825 Ether ∗ 1860$ = 0,015345$.



	base fee 26	base fee 19	base fee 26, MPF 1	base fee 19, MPF 1	base fee 26, MPF 3	base fee 19, MPF 3
nuevo usuario	$40,62	$30,47	$39,17	$29,01	$42,07	$31,92
agregar los datos de la historia médica	$6,02	$4,51	$5,80	$4,30	$6,23	$4,73
actualización de parámetros en la historia médica	$2,24	$1,68	$2,16	$1,60	$2,32	$1,76
consultar la historia	$0,02	$0,01	$0,02	$0,01	$0,02	$0,02

Tabla 2 Tabla de análisis de sensibilidad

En este análisis de sensibilidad se ha evaluado la fluctuación de los costos de las transacciones en la red Ethereum. Se han considerado variables clave que influyen significativamente en el precio final de la transacción, destacando especialmente el valor actual del ether al momento de realizarla.

En todos los escenarios analizados, se ha observado que la primera transacción, que implica la asignación de un nuevo usuario y su correspondiente clave, representa el costo más elevado en comparación con las demás transacciones.

Esto se debe a que la asignación de un nuevo usuario y clave implica una mayor cantidad de gas utilizada en la transacción, lo cual se traduce en un mayor costo en términos de ether. Además, el valor del ether en el mercado también juega un papel importante, ya que afecta directamente al costo en dólares de la transacción.

Es importante tener en cuenta estas fluctuaciones en los costos de las transacciones al desarrollar y utilizar sistemas basados en la red Ethereum, ya que pueden tener un impacto significativo en la viabilidad y rentabilidad de dichas operaciones.

5.3 Estimar costos en otras redes Blockchain y analizar posibles alternativas para el despliegue.

5.3.1 Alternativa Bitcoin

Bitcoin fue concebido como una moneda digital descentralizada. Esto significa que, en lugar de depender de entidades centrales como bancos, permite transacciones directas entre individuos, a esto se le llama "peer-to-peer". Al eliminar intermediarios, se facilita un proceso de transferencia de dinero más directo y sin costes adicionales.

Bitcoin utiliza un lenguaje de programación llamado "script". A diferencia de algunos lenguajes de programación que ofrecen una amplia variedad de funciones, el "script" de Bitcoin es intencionadamente restrictivo. Esta restricción no es un descuido, sino una elección de diseño. Al ser menos complejo, se minimiza el riesgo de ataques, asegurando la integridad y seguridad de la red.

Precisamente debido a estas características de diseño centradas en la seguridad y simplicidad, Bitcoin no fue diseñado para albergar DApps (aplicaciones descentralizadas). Las DApps requieren un lenguaje de programación más versátil y complejo, capaz de ejecutar contratos inteligentes de variada naturaleza. Plataformas como Ethereum, que sí alojan DApps, poseen un lenguaje de programación Turing completo, permitiendo una mayor flexibilidad a costa de enfrentar mayores desafíos de seguridad. La esencia y diseño original de Bitcoin priorizan la seguridad y la función monetaria sobre la capacidad de alojar aplicaciones descentralizadas.

5.3.2 Alternativa TRON

TRON es una tecnología blockchain diseñada específicamente para permitir la creación y operación de contratos inteligentes y aplicaciones descentralizadas, conocidas como DApps. Funciona con un token llamado TRX, que actúa como la moneda principal dentro de esta red y se utiliza para transacciones, compras y ventas. Un punto destacado de TRON es su capacidad de procesamiento: puede manejar alrededor de 2,000 transacciones por segundo, según sus fundadores.

En lugar de utilizar el sistema de consenso de prueba de trabajo (PoW) que emplea Bitcoin, TRON opera mediante un mecanismo denominado Prueba de Participación Delegada (DPoS, por sus siglas en inglés). En DPoS, no todos los participantes (o nodos) compiten para validar y agregar el próximo bloque en la cadena. En su lugar, los propietarios de TRX seleccionan mediante votación a un grupo reducido de representantes. Estos representantes elegidos tienen la tarea de verificar las transacciones y añadir nuevos bloques a la blockchain.

Adicionalmente, TRON introduce dos conceptos fundamentales: "energía" y "banda ancha". Ambos son recursos esenciales para realizar transacciones y activar contratos inteligentes en la red. Como incentivo y para adquirir estos recursos, los usuarios pueden "congelar" sus tokens TRX, lo que implica ponerlos en una especie de modo de espera durante un tiempo determinado, a cambio de obtener energía y banda ancha.

Una característica notable de TRON es su compatibilidad con Ethereum en términos de desarrollo. Los creadores de aplicaciones y contratos en TRON utilizan Solidity, el mismo lenguaje de programación que se usa en la red Ethereum. Esta elección permite que los desarrolladores puedan trasladar o adaptar fácilmente sus DApps de Ethereum a TRON y viceversa.

TRON, al igual que Ethereum, ofrece un entorno de desarrollo integrado en línea, conocido como tronide.io. Es una plataforma o entorno de desarrollo específico para la blockchain de Tron. En esta herramienta, se pueden diseñar, desplegar y probar contratos inteligentes específicamente para Tron. Al llevar a cabo pruebas con estos contratos en tronide.io, se pueden

obtener métricas específicas, como los costos asociados a la ejecución de cada transacción dentro del contrato.

En el contexto de Tron, cuando hablamos de "costos de ejecución", nos referimos a la "energía" consumida en la blockchain durante el proceso de una transacción. La "energía" es un recurso vital en Tron que se utiliza para procesar operaciones y garantizar la correcta ejecución de contratos inteligentes.

Utilizando la compatibilidad existente entre Ethereum y Tron, se llevó a cabo una prueba con el mismo contrato inteligente originalmente diseñado para Ethereum, pero esta vez en Tron, a través de su herramienta de desarrollo tronide.io. Esto permitió realizar una comparación directa de los costos de transacción asociados al contrato en ambas blockchains. Una vez que se probó el contrato inteligente en ambas plataformas, se pudo comparar cuánto cuesta realizar una transacción usando dicho contrato en Ethereum y en Tron.

La finalidad de esta comparación es tener una visión clara de las diferencias económicas al ejecutar un mismo contrato inteligente en dos blockchains diferentes, lo que puede ser crucial para decidir en qué plataforma operar o migrar en el futuro.

Para el cálculo de estos costos en nuestro escenario, se partió de la premisa de que los usuarios optan por pagar por la energía directamente en lugar de "congelar" sus TRX.

La tasa actual de energía en la red TRON es de 0.002 TRX por unidad de energía.

- **Asignar un nuevo usuario y clave:**

Para calcular el costo en TRX, se multiplica la energía gastada por la tasa de energía:

Unidades de energía gastada = 791655

Tasa de energía = 0.002 TRX

(Unidades de energía gastada * Tasa de energía)

791655 * 0.002 = **1.583,31 TRX**

Si lo convertimos en dólares,

TRX = 1.583,31

Tasa $ por cada unidad de TRX = 0.07725 $

1.583,31TRX * 0.07725 = **122.31$**

- **Agregar los datos de la historia médica:**

Unidades de energía gastada = 115636

Tasa de energía = 0.002 TRX

(Unidades de energía gastada * Tasa de energía)

115636 * 0.002 = 231,272 TRX

Si lo convertimos en dólares,

TRX = 231,272

Tasa $ por cada unidad de TRX = 0.07725 $

231,272 TRX * 0.07725 = **17,86$**

- **Actualización de parámetros en la historia médica**

Unidades de energía gastada = 43012

Tasa de energía = 0.002 TRX

(Unidades de energía gastada * Tasa de energía)

43012 * 0.002 = 86,024 TRX

Si lo convertimos en dólares,

TRX = 86,024

Tasa $ por cada unidad de TRX = 0.07725 $

86,024 TRX * 0.07725 = **6,64$**

- **Consultar Historia**

Unidades de energía gastada = 28380

Tasa de energía = 0.002 TRX

(Unidades de energía gastada * Tasa de energía)

28380 * 0.002 = 56.76 TRX

Si lo convertimos en dólares,

TRX = 56.76

Tasa $ por cada unidad de TRX = 0.07725 $

56.76 TRX * 0.07725 = **4.38$**

A simple vista, se puede notar que el valor del token de Tron es inferior al de Ethereum. Sin embargo, al profundizar en los detalles técnicos, se descubre que Tron consume más "energía" para procesar transacciones en comparación con Ethereum. Esta mayor demanda de energía en Tron implica que, aunque su token pueda ser más barato, el costo total asociado con la ejecución de un contrato inteligente en esta plataforma puede ser más elevado que en Ethereum. En otras palabras, mientras que adquirir un token TRX puede ser menos costoso que un ETH, las tarifas acumuladas por usar ese token en transacciones y contratos en Tron podrían sumar un costo total mayor que el de realizar operaciones similares en Ethereum.

Tipo de Transacción	Precio en TRON $	Precio en Ethereum $
Nuevo Usuario	122.31$	24.45 $
Agregar Datos	17.86 $	4.07 $
Actualización de Datos	6.64 $	1.51 $
Consultar Historia	4.38$	0.0132 $

Tabla 3 Tabla de comparación de Blockchains

Capítulo 6

6.1 Conclusión

En este trabajo de investigación, se ha demostrado la factibilidad económica del modelo de sistema de almacenamiento de historias médicas basado en la red Ethereum. Los costos de las transacciones han sido evaluados cuidadosamente, considerando variables claves como el valor del ether y la cantidad de gas utilizada en cada transacción.

Una de las principales ventajas de utilizar la red Ethereum es su escalabilidad y bajo costo en comparación con otros métodos tradicionales de almacenamiento de datos. Aunque las transacciones iniciales de asignación de usuario y clave pueden tener costos más altos, esto se debe a la mayor cantidad de gas necesaria para ejecutar dicha operación. Sin embargo, una vez que el usuario está registrado en el sistema, las transacciones posteriores, como agregar datos a la historia médica o consultar registros, tienen costos significativamente más bajos.

Además, la fluctuación del valor del ether en el mercado puede afectar los costos de las transacciones en términos de dólares. A pesar de estas fluctuaciones, el modelo sigue siendo factible desde el punto de vista económico. La tecnología blockchain ofrece una forma segura y confiable de almacenar y acceder a los datos médicos, lo cual es de gran valor para los pacientes, los médicos y los sistemas de salud en general. La transparencia, la inmutabilidad y la interoperabilidad que proporciona la Blockchain son ventajas significativas que justifican los costos asociados con su implementación.

La implementación de un modelo de sistema de almacenamiento de historias médicas basado en blockchain ofrece diversos aspectos positivos. En primer lugar, proporciona una mayor seguridad y privacidad al acceder a los datos médicos al utilizar tecnología descentralizada. Esto garantiza la protección contra manipulaciones y accesos no autorizados, brindando a los pacientes un mayor control sobre su información personal.

Para los pacientes, este modelo ofrece ventajas significativas. Permite un acceso más rápido y eficiente a sus historias médicas, lo que facilita la coordinación de la atención y evita la

duplicación de pruebas y tratamientos. Mejora la interoperabilidad entre diferentes proveedores de atención médica, lo que contribuye a una atención más integral y centrada en el paciente. La transparencia y trazabilidad de los datos también fomentan la confianza en el sistema de salud.

Para los médicos, el sistema de almacenamiento de historias médicas basado en blockchain simplifica el intercambio de información y agiliza los procesos de toma de decisiones clínicas. Acceder a los datos médicos actualizados y precisos de los pacientes les permite brindar un mejor cuidado y personalizar los tratamientos.

En cuanto a las ventajas para el sistema de salud, este modelo mejora la eficiencia en la gestión de datos y reduce costos a largo plazo. La compartición segura de información y la eliminación de procesos manuales redundantes optimizan los recursos y mejoran la calidad de la atención. Facilita la investigación y el análisis de datos para mejorar la medicina basada en evidencias y el desarrollo de mejores prácticas clínicas.

Comparado con otros métodos de almacenamiento de historias médicas, el enfoque basado en blockchain ofrece mayor seguridad, privacidad y transparencia. La descentralización y la tecnología de cadena de bloques garantizan la integridad de los datos y evitan la dependencia de una única entidad centralizada.

Es importante destacar que este tipo de sistemas de almacenamiento basados en blockchain aún pueden evolucionar. A medida que la tecnología blockchain madura y se desarrollan estándares y regulaciones específicas para la gestión de datos médicos, se esperan mejoras en términos de escalabilidad, eficiencia y facilidad de uso. Además, la integración de tecnologías emergentes como la inteligencia artificial y el análisis de datos puede potenciar aún más los beneficios de modelos como el presentado.

En cuanto a su obsolescencia, es difícil predecir cuánto tiempo tardará en quedar obsoleto un sistema de almacenamiento basado en blockchain. La tecnología blockchain sigue siendo un área activa de investigación y desarrollo, y su adopción en la industria de la salud está en una etapa temprana. Sin embargo, a medida que se superen los desafíos técnicos y se resuelvan los aspectos regulatorios, es probable que este enfoque continúe evolucionando y brindando beneficios a largo plazo en la gestión de datos médicos.

Finalmente, para el complejo y dinámico mundo de las criptomonedas, la evaluación de costos no se limita solo al valor superficial de un token. Aunque Tron pueda presentar un valor nominal más bajo en comparación con Ethereum, esta ventaja puede ser engañosa al analizar las operaciones en su totalidad. Al observar más de cerca el mecanismo que impulsa cada blockchain, emergen diferencias técnicas significativas. La cantidad de "energía" que Tron requiere para llevar a cabo sus transacciones es, según se informa, mayor que la que Ethereum necesita. Esta intensidad energética de Tron, paradójicamente, podría llevar a costos operativos más altos, a pesar de su token de precio inferior. Por lo tanto, los usuarios y desarrolladores deben considerar no solo el valor inicial del token, sino también los costos asociados con su uso, especialmente cuando se trata de desplegar y ejecutar contratos inteligentes. Esta consideración es crucial para obtener una imagen completa y precisa de la eficiencia económica y operativa de ambas plataformas.

6.2 Recomendaciones

- Adaptar el modelo de sistemas de almacenamiento de datos médicos basado en blockchain a una Dapp.
- Realizar una revisión exhaustiva de las mejores prácticas de diseño de contratos inteligentes y patrones de seguridad en la red Blockchain para garantizar la integridad y la seguridad de la información médica almacenada.
- Considerar la interoperabilidad del contrato inteligente con otros sistemas de gestión de historias médicas existentes para facilitar la integración y el intercambio de datos entre diferentes proveedores de atención médica.
- Considerar la posibilidad de utilizar tecnologías de compresión de datos para reducir el tamaño de almacenamiento necesario y, por lo tanto, los costos asociados.
- Comparar los costos totales del sistema de gestión de historias médicas basado en Blockchain con los sistemas tradicionales de almacenamiento de historias médicas para evaluar los posibles ahorros a largo plazo.

Referencias

Academy. (2019). *Tipos de blockchain*. Obtenido de https://academy.bit2me.com

Albeyatti, A. (2017). Médicalchain Blockchain for electronic health records. *Médicalchain.*

Azaria, A. /. (2016). Using blockchain for médical data access and permission management. *MedRec 2nd International Conference on Open and Big Data.*

Bitcoin. (2015). *Bitcoin*. Obtenido de https://bitcoin.com

Bittorrent. (2022). *Bittorrent*. Obtenido de https://www.bittorrent.com/es/

Consensys. (2020). Eventos y registros en ethereum. *consensys*.

Corvo, H. S. (2019). Factibilidad económica: qué es y cómo se hace. *Lifeder*.

Cuchillero, M. (2022). Una guía definitiva para los cálculos de la tarifa de gas de Ethereum EIP-1559. *blocknative*.

Cuesta, P. (2020). ¿qué es cardano y por qué está teniendo tanto impacto en la tecnología blockchain? *Openexpoeurope*.

Diedrich, H. (2019). *Lexon: Digital Contracts.* Independently published.

Elgarte, F. (2022). Eventos en la Blockchain. Cómo emitirlos con Solidity y recepcionarlos con web3.js. *medium*.

Ethereum. (2020). *Ethereum*. Obtenido de https://ethereum.org/es/developers/docs

GIT. (2022). *GIT*. Obtenido de https://git-scm.com/

Github. (2020). *IPFS*. Obtenido de https://github.com/ipfs/ipfs

Gonzales, P. (2021). La historia clínica. Orígenes y evolución. *Ocronos - Editorial Científico-Técnica*.

Iberdrola. (2018). *Contratos inteligentes para formalizar acuerdos en la era digital.* Obtenido de https://www.iberdrola.com/innovacion/smart-contracts

Iproup. (2020). *Estas blockchain quieren destronar a ethereum: cuáles son y qué hacen para lograrlo.* Obtenido de https://iproup.com/economia-digital/20740-diez-blockchain-de-nueva-generacion-quieren-destronar-a-ethereum

joranhonig. (2022). *Fuse.* Obtenido de https://joranhonig.nl/stealing-info--using-ipfs-f-use/

Kademlia. (2022). Obtenido de https://es.theastrologypage.com/kademlia

Padilla, J. (2020). Blockchain y contratos inteligentes: aproximación a sus problemáticas y retos jurídico. *Revista de Derecho Privado,*, núm. 39, pp. 175-201.

Perez, J. (2021). React. *tribalyte*.

Preukschat, A. (2017). Ethereum es Turing completo. *eleconomista.es*.

Remix. (2016). *Remix*. Obtenido de https://remix.ethereum.org

Sap. (s.f.). *what is blockchain*. Obtenido de https://sap.com/latinamerica/

Starfield, A., & Smith, K. (1994). *How to Model It: Problem Solving for the Computer Age.* Burgess Intl Group.

Unpocodejava. (2022). *INFURA*. Obtenido de https://unpocodejava.com/2022/07/05/que-es-infura

Web3. (2022). Hash node. *https://web3.hashnode.com/*.

Wuani, H. (2010). La historia clínica. Evolución histórica, objetivos. Su importancia. La tecnología y la relación médico-paciente, hoy y mañana. *Medicina Interna. Vol. 26, Num. 3.*

Xia, Q. /. (2017). Data Sharing among Cloud Service Providers via Blockchain. *MeDShare: IEEE Access PP(99):1-1.*

Anexos

Código de contrato inteligente para almacenar datos médicos en la Blockchain

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.19;
contract Historia {
        struct Privado {
                string nombre;
                uint clave;
        }
        struct Publico {
                uint fecha;
                uint peso;
                string enfermedad;
                string tratamiento;
        }
        struct Persona { //estructura con la información de cada persona address direccion;
                Privado id;
        }
        Persona paciente;
        mapping(uint => Publico) public dataPublica; //crea una variable de estado publica
        event diagnostico (address _paciente, Publico _diagnostico); // Evento para el log
        constructor (string memory apodoPaciente, uint password) {
                paciente.direccion = msg.sender;
        paciente.id.nombre = apodoPaciente;
```

```
        paciente.id.clave = password;
    }
    // agregaDiagnostico agrega un reporte sobre una condición y tratamiento correspondientes // usando dos métodos de almacenamiento en la para comparar costos.
    function agregaDiagnostico(string memory _enfermedad, string memory _tratamiento, uint      // método uno: guardar en memoria del contrato
    dataPublica[_fecha] = Publico({
        fecha: _fecha,
        peso: _peso,
        enfermedad: _enfermedad,
        tratamiento: _tratamiento
    });
    // metodo dos: guardar en los logs.
    emit diagnostico(msg.sender, dataPublica[_fecha]);
    }
        function cambiarClave(uint _newPass, uint _oldPass) public { //funcion para cambiar clave if(msg.sender != paciente.direccion) {
    revert();
    }
    if(paciente.id.clave != _oldPass) {
    revert();
    }
    paciente.id.clave = _newPass;
    }
}
```

Glosario

Tecnología Blockchain

Blockchain es un conjunto de tecnologías que hacen las veces de un gran libro de registros, que contiene información de transacciones de datos o valor realizadas en la red y dicha información se guarda en bloques entrelazados. De ahí su nombre que en español traduce “Cadena de Bloques”.

Según (Sap, n.d.) El funcionamiento del blockchain se explica mejor comprendiendo el aspecto comunal. Se basa en lo que se denomina “tecnología de registro distribuido”. Todos los miembros de la red de pares que componen estos registros pueden ver la misma información en los bloques individuales.

Una transacción que se graba en una computadora o nodo es visible para cada una de las computadoras de la red digital. Todos pueden ver los mismos datos. Es más, pueden rechazar o verificar lo que ven. La información se comunica a todos los otros bloques de la cadena. Esto es lo que hace que la tecnología sea muy difícil de hackear. Ninguna computadora controla todos los datos, y modificarlos en un bloque significa que toda la cadena debería actualizar también. Todos tienen una copia que se actualiza automáticamente; las modificaciones deben ser verificadas por todos los usuarios de la red, y con la adición de código programable (sugerido por primera vez por el ruso canadiense Vitalik Buterin, cofundador de la Red Ethereum), la tecnología puede ser usada para crear contratos inteligentes que puedan ejecutar acuerdos cuando se cumplen ciertas condiciones.

Para el año 2022, se estima que hay más de 1000 blockchains, pero el número crece rápidamente, actualmente existen distintos tipos de blockchain con características y capacidades distintas entre ellas, algunas de las principales redes Blockchain son:

¿Qué es Bitcoin?

Según (Bitcoin, 2015), Bitcoin usa tecnología peer-to-peer o entre pares para operar sin una autoridad central o bancos; la gestión de las transacciones y la emisión de bitcoins es llevada a cabo de forma colectiva por la red. Bitcoin es de código abierto. Algunas de sus propiedades son:

Pública:

El código y todas las transacciones registradas en la red Bitcoin son verificables públicamente y están disponibles para todas las operadoras de la red Bitcoin.

Abierta:

Cualquiera puede unirse o salir de la red, validar transacciones y extraer nuevas monedas.

Peer to Peer:

Las transacciones entre pares se envían de una parte a otra sin la necesidad de que una autoridad centralizada o un intermediario las autorice.

Distribuida:

El almacenamiento y la validación de la cadena de bloques se realiza a través de una amplia variedad de participantes de la red independiente.

Seguro:

Siempre que más del 50% de los nodos sean honestos.

Confiable:

El tiempo de actividad de las redes se mantiene 24 horas al día, 7 días a la semana, durante todo el año.

Sin fronteras:

La red opera a través de fronteras geográficas, accesible prácticamente en cualquier lugar con una conexión a internet o por satélite.

Resistente a la censura:

Nadie tiene la potestad de bloquear la transferencia de fondos en función del tipo, origen o destino de cualquier transacción siempre que sea válida de acuerdo con las reglas de consenso.

Neutral:

La red bitcoin es independiente de las cuentas, las personas y los motivos, para enviar y recibir bitcoins donde sea, cuando sea y para quien sea.

Ethereum

Ethereum es la segunda red blockchain más popular, a diferencia de Bitcoin se especializa en contratos inteligentes, lo que representa múltiples herramientas adicionales. Fue propuesta a finales del 2013, se describió como una nueva plataforma revolucionaria para aplicaciones descentralizadas, que admite cualquier caso, desde votaciones, intercambios financieros y propiedad inteligente. (Ethereum, 2020).

En (Academy, 2019) se explica que estas aplicaciones descentralizadas (o "dapps") obtienen los beneficios de la criptomoneda y la tecnología blockchain. Son confiables y predecibles, lo que significa que una vez que se cargan en Ethereum, siempre se ejecutarán según lo programado. Pueden controlar los activos digitales para crear nuevos tipos de aplicaciones financieras. Se pueden descentralizar, lo que significa que ninguna entidad o persona los controla. Pero pueden ser controlados por una comunidad mediante algún mecanismo de gobernanza.

Algunas aplicaciones en Ethereum las cuales se puede usar hoy en día:

- Carteras de criptomonedas que le permiten realizar pagos baratos e instantáneos con ETH u otros activos.
- Aplicaciones financieras que le permiten pedir prestado, prestar o invertir sus activos digitales.
- Mercados descentralizados, que le permiten intercambiar activos digitales, o incluso intercambiar “predicciones” sobre eventos en el mundo real.
- Juegos donde tienes activos en el juego e incluso puedes ganar dinero.

Se caracteriza por ser una plataforma básica estandarizada. Tiene un lenguaje de programación llamado Solidity para facilitar la creación de aplicaciones descentralizadas y contratos inteligentes por cualquier persona.

Posee su propia moneda o activo nativo llamado ether, que se puede rastrear en la cadena de bloques y se usa más como combustible computacional que como moneda. A diferencia del bitcoin que tiene un suministro total de veintiún millones, el de ether permanece ilimitado.

Ethereum se basa en el concepto de contratos inteligentes autoejecutables. Un contrato inteligente en el contexto de Ethereum consiste en programas informáticos inmutables que se ejecutan de manera determinista, como parte del protocolo de red de Ethereum.

Cardano

Según (Cuesta, 2020) Es una cadena de bloques de código abierto, así como una plataforma para ejecutar contratos inteligentes. Cardano fue fundada en 2015 por el cofundador de Ethereum, Charles Hoskinson. El desarrollo del proyecto está supervisado por la Fundación Cardano, con sede en Zug (Suiza).

Es una de las criptomonedas que utiliza una blockchain de prueba de participación, que se considera una alternativa más ecológica a los protocolos de prueba de trabajo. Cardano utiliza la tecnología de prueba de participación llamada Ouroboros, en contraposición a Bitcoin y Ethereum que utilizan protocolos de prueba de trabajo.

La intención de cardano es ejecutar un alto volumen de transacciones garantizando la seguridad gracias a la criptografía. Además de ello, también pretende ser una plataforma preparada para que se puedan crear diferentes aplicaciones descentralizadas, llamadas DAPPs.

Polkadot

Para (Iproup, 2020) uno de los proyectos más sonados de los últimos tiempos, Polkadot está construido para conectar cadenas privadas y de consorcios, redes públicas y sin permisos, oráculos y “futuras tecnologías que aún están por crear". Polkadot es un proyecto Blockchain de próxima generación que conecta múltiples blockchains especializadas en una red unificada. Está asegurada por un mecanismo de consenso, denominado Prueba de participación nominada (NPoS) que permite que dos tipos de actores de la red -validadores y nominadores- aseguren la red.

Avalanche

Según (Iproup, 2020) La red Avalanche consta de múltiples blockchains y utiliza un mecanismo novedoso de consenso de prueba de participación para lograr un alto rendimiento. El proyecto asegura combinar los beneficios del consenso de Nakamoto (robustez, escala y descentralización) con los del consenso clásico (velocidad, finalidad rápida y eficiencia energética). Debido a su arquitectura, puede describirse como una "plataforma de plataformas", que consta de miles de subredes para formar una única red interoperable. AVAX, su token nativo, se usa para asegurar la red mediante staking, además de como mecanismo de intercambio peer-to-peer. Se trata además de la primera plataforma Blockchain en alcanzar tiempos de transacción inferiores a un segundo.

Contratos Inteligentes

Un contrato inteligente es un acuerdo entre dos personas o entidades en forma de código informático tiene la facultad de ejecutarse y hacerse cumplir por sí mismo, de forma autónoma y automática, a partir de una serie de parámetros programados. Es una “Cuenta que contiene un código que se ejecuta cada vez que recibe una transacción de otra cuenta” (Iberdrola, 2018) y se cumplen otras condiciones específicas. De la mano de la tecnología blockchain, su principal valor reside en reforzar la seguridad, la transparencia y la confianza entre los firmantes, evitando malentendidos, falsificaciones o alteraciones y prescindiendo de intermediarios. La idea fue propuesta en los años 90 por Nick Szabo, un pionero de la informática moderna, que los definió como un conjunto de promesas virtuales con unos protocolos asociados para hacer que se cumplan. El protocolo de Bitcoin, que básicamente registra la constancia de un pago, se puede considerar como una versión primitiva de un contrato inteligente.

Los contratos inteligentes se ejecutan en blockchain lo que implica que los términos se almacenan en una base de datos distribuida y no pueden modificarse (recordando que Blockchain, o cadena de bloques, es un libro de contabilidad digital distribuido que almacena datos de cualquier tipo. Una cadena de bloques puede registrar información sobre transacciones de criptomonedas, propiedad de NFT o contratos inteligentes). Las transacciones también son procesadas en blockchain, lo que automatiza pagos. (Iberdrola, 2018)

Según (Padilla, 2020), uno de los elementos principales de los contratos inteligentes es su capacidad de auto ejecución. Los contratos inteligentes son de ejecución automática en cuanto que automáticamente ejecuta una transacción ante la ocurrencia de eventos definidos de manera previa. Dicha característica pretende evitar que en la ejecución contractual intervenga el hombre, que se presume parcial y poco fiable, y puede rehusarse al cumplimiento de sus obligaciones. El código es imparcial y objetivo, lo cual garantiza que el contrato se ejecutará al pie de la letra, sin que se puedan presentar alteraciones o modificaciones a su contenido. En este orden de ideas, es razonable considerar que la necesidad de acudir al sistema judicial para solicitar el cumplimiento de obligaciones, si no es eliminada, es altamente reducida.

Los contratos inteligentes pueden proporcionar varias ventajas como:

- Hacer cumplir automáticamente la igualdad de poder de todas las partes involucradas.
- Proteger los derechos de un individuo haciendo cumplir expectativas razonables para el firmante.
- Eliminar la posibilidad de que algún signatario incumpla sus obligaciones.

También se enfrentan a varios desafíos:

- La legalidad y si un contrato inteligente es admisible en la Corte es cuestionable
- Aún son difíciles de entender para quienes no son programadores.
- La implementación de los contratos inteligentes sigue siendo un desafío en los casos en que se requiere el juicio humano, por ejemplo, interpelar las condiciones y circunstancias de un accidente automovilístico.
- Cualquier “error” en el código de los contratos inteligentes puede ser explotado por un actor malintencionado.

En este contexto, debe resaltarse que los contratos inteligentes no pueden ser modificados ni detenidos, o, lo que es lo mismo, son inmutables. Si bien la inmutabilidad puede resultar atractiva de manera preliminar, se trata de un atributo que puede presentar varios problemas.

(Iberdrola, 2018) El funcionamiento de un contrato inteligente es similar al de otras transferencias en blockchain.

Estos son los pasos necesarios:

1. Un usuario inicia una transacción desde su monedero en blockchain.
2. La transacción llega a la base de datos distribuida, donde se confirma la identidad.
3. Se aprueba la transacción, que puede ser una transferencia de fondos.
4. La transacción incluye el código que define qué tipo de transacción debe ejecutarse.
5. Las transacciones se añaden como un bloque dentro del blockchain.
6. Cualquier cambio en el estado del contrato sigue el mismo proceso para actualizarse.

Lenguaje de programación que usan los contratos inteligentes

Uno de los principales lenguajes es Solidity, un lenguaje de tipado estático de alto nivel que se puede usar para programar contratos inteligentes para la red Ethereum. Su sintaxis es muy similar a la de lenguajes conocidos como C ++ o JavaScript. El propósito de crear Solidity es permitir la escritura fácil de contratos inteligentes para la red Ethereum. Es un lenguaje diseñado para aprovechar al máximo la máquina virtual Ethereum y permite la creación y desarrollo de contratos inteligentes que se pueden ejecutar mejor en el EVM. Con este fin, los programadores pueden desarrollar aplicaciones en un lenguaje que sea fácil de usar, leer y mantener, de modo que una vez completado, el motor Solidity convierta códigos simples en códigos de máquina que el EVM pueda entender. (joranhonig, 2022)

En esencia, Solidity permite el desarrollo de contratos inteligentes en Ethereum. Aunque no es el único lenguaje, es el primero y más versátil, y se han desarrollado más contratos inteligentes en él. La característica principal de Solidity es que el tipo de lenguaje de programación es Turing Complete. Debido a Solidity y su relación con EVM, los desarrolladores pueden crear programas completos de Turing (Preukschat, 2017). Esto se debe a que EVM puede ejecutar cualquier código definido por el desarrollador dentro del alcance de su función.

Estructura de un contrato inteligente

Según (Diedrich, 2019)Un contrato inteligente debe escribirse de tal manera que un usuario sin conocimientos en lenguaje de programación lo pueda leer e interpretar de manera natural. En el siguiente ejemplo veremos cómo se puede estructurar el algoritmo de un contrato inteligente, donde los actores serán, el Pagador, el Beneficiario y el Árbitro, además de haber una tarifa, una garantía y unas clausulas.

El pagador es una persona

El Beneficiario es una persona

Árbitro es una persona

La tarifa es una cantidad

El pagador paga una cantidad en depósito en garantía, designa al Beneficiario, nombra al árbitro, y también fija la Cuota.

CLÁUSULA: Pago.

El Árbitro puede pagar los honorarios de la cuenta de depósito en garantía a sí mismo, y luego pagar el resto del depósito en garantía al Beneficiario. CLÁUSULA: Reembolso.

El Árbitro puede pagarse a sí mismo los honorarios del depósito en garantía y luego devolver el resto del depósito en garantía al Pagador.

Como se dijo anteriormente, Solidity es el lenguaje de programación más popular a la hora de escribir contratos inteligentes ya que es usado por la blockchain de Ethereum, que proporciona un entorno de desarrollo web, con análisis estático y una máquina virtual para hacer pruebas de blockchain este entorno se llama Remix (Remix, 2016).

Remix nos da acceso a un entorno amigable donde podemos hacer pruebas para compilar y desplegar código Solidity.

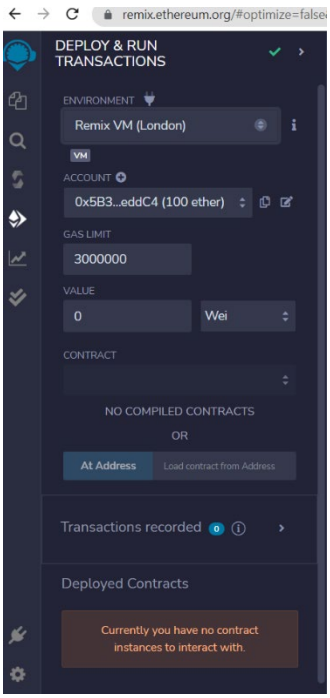


Figura 16 : Deploy Remix, tomado de remix.ethereum.org

Las pruebas de Remix se pueden hacer de manera local con el software de Ganache, que es un simulador local de pruebas de Ethereum. Ofrece una Blockchain personal para testear, compilar y desplegar contratos inteligentes. Ganache genera diez cuentas para interactuar con la Blockchain, creando claves privadas y dirección de Ethereum por cada una, muestra un saldo de cien ethers y las transacciones que se van haciendo con ese saldo.

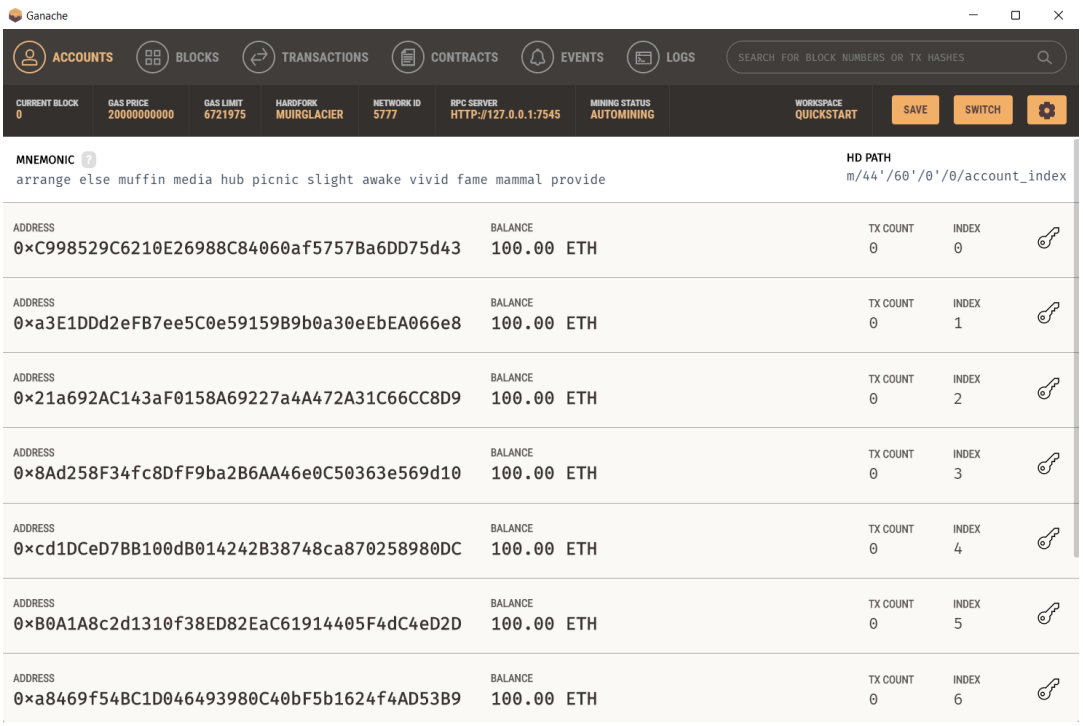


Figura 17 Software Ganache

Hash

Un hash es el resultado una operación criptográfica que genera identificadores únicos e irrepetibles a partir de una información dada. Estas funciones tienen como objetivo primordial codificar datos para formar una cadena de caracteres única. Todo ello sin importar la cantidad de datos introducidos inicialmente en la función. Estas funciones sirven para asegurar la autenticidad de datos, almacenar de forma segura contraseñas, y la firma de documentos electrónicos. Los hashes son una pieza clave en la tecnología blockchain (Academy, 2019).

IPFS

Para (Github, 2020) IPFS (el sistema de archivos interplanetarios) es un protocolo de distribución de hipermedios dirigido por contenido e identidades.

IPFS es un sistema de archivos distribuido que busca conectar todos los dispositivos informáticos con el mismo sistema de archivos. De alguna manera, esto es similar a los objetivos originales de la Web, pero IPFS en realidad es más similar a un solo enjambre de BitTorrent que intercambia objetos Git.

Con el almacenamiento en la nube típico, carga datos y obtiene un enlace, URL o una forma de ubicar dónde están los datos. Este es un identificador de ubicación. Por otro lado, IPFS utiliza un identificador de contenido (CID). Esto significa que cada vez que cargamos datos usando IPFS, IPFS usa un hash criptográfico para generar un CID. Este CID es único ya que se basa estrictamente en el contenido de los datos o archivos.

IPFS se está convirtiendo en un nuevo subsistema importante de Internet. Si se construye correctamente, podría complementar o reemplazar HTTP. Podría complementar o reemplazar aún más.

Vamos a ir punto por punto:

IPFS es un protocolo:

- Define un sistema de archivos direccionado por contenido.
- Coordina la entrega de contenido.
- Combina Kademlia (Kademlia, 2022)+ BitTorrent (Bittorrent, 2022) + Git (GIT, 2022).

IPFS es un sistema de archivos:

- Tiene directorios y archivos.
- Es un sistema de archivos montable (a través de FUSE (joranhonig, 2022))

IPFS es una web:

- Se puede utilizar para ver documentos como la web convencional.
- Se puede acceder a los archivos a través de HTTP en https://ipfs.io/.
- Los navegadores y las extensiones pueden aprender a usar los esquemas de ipfs://URL o ipns://URI directamente.
- El contenido con dirección hash garantiza la autenticidad.

IPFS es modular:

- Capa de conexión sobre cualquier protocolo de red.
- Capa de enrutamiento.
- Utiliza una capa de enrutamiento DHT (Kademlia/Coral).
- Utiliza un servicio de nombres basado en rutas.

- Utiliza un intercambio de bloques inspirado en BitTorrent.

IPFS usa criptografía:

- Direccionamiento de contenido hash criptográfico.
- Deduplicación a nivel de bloque.
- Integridad de archivos más control de versiones.
- Cifrado a nivel de sistema de archivos más soporte de firma.

IPFS es p2p:

- Transferencias de archivos peer-to-peer en todo el internet.
- Arquitectura completamente descentralizada.
- Sin punto central de falla.

IPFS es un CDN:

- Agregue un archivo al sistema de archivos localmente y ahora estará disponible para todo el mundo.
- Fácil de almacenar en caché (nomenclatura hash de contenido).
- Distribución de ancho de banda basada en BitTorrent.

IPFS tiene un servicio de nombres:

- IPNS, un sistema de nombres inspirado en SFS.
- Espacio de nombres global basado en PKI.
- Sirve para construir cadenas de confianza.
- Puede asignar DNS, onion, bit, etc. a IPNS.

React

Según (Perez, 2021)Es una librería open source de JavaScript para desarrollar interfaces de usuario. Como norma general, al diseñar una aplicación con React, lo que se hace es crear componentes independientes y reusables para, crear interfaces de usuarios más complejas.

Web3-react

Según (Web3, 2022)Web3-react es un framework de web3 para ayudar a los desarrolladores de blockchain a crear Dapps de Ethereum utilizando la arquitectura React.

Web3-react es una state machine que almacena ciertos bits esenciales de datos actualizados pertinentes a su Dapp. De forma predeterminada, admite proveedores como Meta Mask, Gnosis Safe, Frame y WalletConnect.

Web3-react es un marco adecuado para el desarrollo de dApp ya que, puede escribir complementos mediante programación en blockchain. No existe una arquitectura de datos tradicional, puede crear Dapps sistemáticas, potentes y extensibles.

INFURA

Para (Unpocodejava, 2022)INFURA es una infraestructura como servicio (IaaS), por lo que proporciona todo lo que un desarrollador necesita para construir en blockchains.

INFURA es una plataforma para conectar con redes Blockchain y permite a los desarrolladores construir Smart Contracts utilizando un API JSON-RPC para interactuar con múltiples blockchains. Aunque INFURA fue la primera solución API construida para Ethereum, también ofrece soporte para la red de almacenamiento descentralizada, IPFS, Polygon, Arbitrum y Optimism que son soluciones de escalado y capa 2 para la red Ethereum. Estas soluciones buscan abordar los desafíos de rendimiento y costo asociados con la cadena de bloques principal de Ethereum.

INFURA actúa como el punto de conexión con la blockchain, no hay necesidad de ejecutar un nodo propio para acceder a las cadenas de bloques. Esto reduce significativamente el tiempo de puesta en marcha, ya que no necesitas pasar horas sincronizando tus nodos con múltiples redes.

Puerta de enlace dedicada de INFURA IPFS

La puerta de enlace dedicada de INFURA IPFS permite acceder al almacenamiento descentralizado de proyectos a través de la API de INFURA IPFS. Una puerta de enlace permite que las aplicaciones, como los navegadores, que no admiten de forma nativa IPFS accedan al contenido de IPFS.

Estudio económico para una DAPP:

En este proyecto de investigación, las pruebas se realizaron sobre la blockchain de ethereum, por esto los datos con los que se trabajarán serán bajo las referencias de esta blockchain, según (Ethereum, 2020):

Ether

Ether (eth) es una criptomoneda. Es dinero digital limitado que puede usar en Internet, similar al bitcoin. ETH es el elemento vital de Ethereum. Cuando envía ETH o usa una aplicación Ethereum, pagará una pequeña tarifa en ETH por usar la red Ethereum. Esta tarifa es un incentivo para que un productor de bloques procese y verifique lo que está tratando de hacer.

Debido a que Ethereum es programable, los desarrolladores pueden dar forma a ETH de innumerables maneras.

Gas

El Gas es el coste que tiene el realizar una operación o un conjunto de operaciones en la red Ethereum. Estas operaciones pueden ser varias: desde realizar una transacción hasta ejecutar un contrato inteligente o crear una aplicación descentralizada. En otras palabras, más simples, el Gas es la unidad para medir el trabajo realizado en la EVM. Por lo que podemos inferir que, si el Gas es la medida de computación necesaria para la Blockchain, entonces la ejecución de contratos más complejos pagara más GAS.

Al igual que en el mundo físico, en Ethereum también hay trabajos que cuestan más que otros: si la operación que queremos realizar requiere un mayor uso de recursos por parte de los nodos que forman la plataforma, esto hará que el Gas aumente también y viceversa.

Una unidad de Gas corresponde a la ejecución de una instrucción, como, por ejemplo, un paso computacional.

Denominaciones del Ether

Ya que muchas transacciones en Ethereum son pequeñas, el ether tiene varias denominaciones, que pueden referenciarse para cantidades más pequeñas. De todas estas denominaciones, Wei y gwei son particularmente importantes.

GWEI

Gwei es una denominación de la criptomoneda ether (ETH), utilizada en la red Ethereum, es la mil millonésima parte de un ETH siendo la unidad de éter más utilizada porque es más fácil especificar los precios del gas Ethereum.

El precio de una (01) unidad de Gas, y su valor está dado en Ethereum, 1 GWEI= 0.00000001 Ether, o de otra manera 1 Ether = 1,000,000,000 GWEI.

El fee va a parar a manos de los mineros, por lo que darán prioridad a la ejecución de las transacciones o contratos de aquellos usuarios que hayan ofrecido más GWEI (recordemos que el GWEI es el precio del GAS).

Wei

Wei es el valor de ether más pequeño posible, y, como consecuencia, muchas implementaciones técnicas. El Libro amarillo de Ethereum basan todos sus cálculos en Wei. El Gwei, abreviación de giga-wei, se utiliza para describir los costes del gas en Ethereum.

Denominación	Valor en ether	Uso común
Wei	10^{-18}	Implementaciones técnicas
Gwei	10^{-9}	Comisiones de gas legibles por los humanos

Figura 18 Tabla de valor de Wei, Gwei, tomado de ethereum.com

Prueba de Participación

Para (Ethereum, 2020) la prueba de participación es un tipo de mecanismo de consenso que usan las redes de blockchain para lograr consensos distribuidos.

En Ethereum esto requiere que los usuarios participen con sus ethers para convertirse en un validador de la red. Los validadores no necesitan usar cantidades significativas de potencia informática, ya que a ellos se les selecciona de manera aleatoria y no están compitiendo. No

necesitan minar bloques, sino que únicamente precisan crear bloques cuando se les elige y validar los bloques propuestos cuando no lo son.

Transacción

Según Las transacciones serán agregadas a la blockchain de Ethereum ya no como consecuencia de una competencia que permite a su ganador obtener una recompensa en criptomonedas, sino que será a través de un proceso aleatorio por medio del cual se eligen participantes que podrán validar (registrar) transacciones y obtener una recompensa por ello.

Una transacción de Ethereum hace referencia a una acción iniciada por una cuenta de propiedad externa, en otras palabras, una cuenta administrada por una persona, no un contrato.

Las transacciones, que modifican el estado de la EVM, se deben transmitir a toda la red. Cualquier nodo puede transmitir una solicitud de una transacción que se va a ejecutar en la EVM; a continuación, un minero ejecutará la transacción y propagará la modificación de estado derivada al resto de la red.

Las transacciones implican el pago de una comisión y deben minarse para convertirse en transacciones válidas. Una transacción enviada incluye la siguiente información:

Destinatario:

La transacción destinataria (en caso de que sea una cuenta de propiedad externa, la transacción transferirá valor. Si se trata de un contrato, la transacción ejecutará el código del contrato)

Firma:

Identificador del remitente. Se genera cuando, mediante la clave privada, se firma la transacción y se confirma que el remitente la ha autorizado.

Valor:

Cantidad de ETH que el remitente transfiere al destinatario (en WEI, una denominación de ETH).

Datos:

Campo opcional en el que se incluyen datos arbitrarios.

Límite De Gas:

Cantidad máxima de unidades de gas que puede consumir la transacción. Las unidades de gas representan pasos computacionales.

Max Priority Fee Per Gas:

La cantidad máxima de gas que se incluirá como recompensa para el validador.

Max Fee Per Gas:

La cantidad máxima de gas que se quiere pagar por la transacción (incluidas baseFeePerGas y maxPriorityFeePerGas).

El gas es una referencia al cálculo requerido para procesar la transacción por parte de un validador. Los usuarios tienen que pagar una tarifa por este cálculo. Gas Limit y max Priority Fee Per Gas determinan la tarifa de transacción máxima pagada al validador.

Tipos de Transacciones

En Ethereum hay tipos diferentes de transacciones:

Transacciones regulares:

Una transacción desde una cartera a otra.

Transacciones de despliegue de contratos:

Una transacción sin la dirección <<a>>, donde el campo de datos se usa para el código de contrato.

Ciclo de vida de la transacción

Una vez que la transacción ha sido enviada ocurre lo siguiente:

Una vez que envíes una transacción, la criptografía genera un hash de transacción:

0x97d99bc7729211111a21b12c933c949d4f31684f1d6954ff477d0477 538ff017

Luego, la transacción se transmite a la red y se agrega a un conjunto de transacciones que consta de todas las demás transacciones pendientes de la red. Un validador debe elegir su transacción e incluirla en un bloque para verificar la transacción y considerarla exitosa".

A medida que pasa el tiempo, el bloque que contiene su transacción se actualizará a "justificado" luego a "finalizado". Estas actualizaciones hacen que sea mucho más seguro que su transacción fue exitosa y nunca será alterada. Una vez que se "finaliza" un bloque, solo se puede cambiar mediante un ataque a nivel de red que costaría muchos miles de millones de dólares.

Para enero de 2023, el feed a pagar por transacción en Ethereum es de 3,50 $

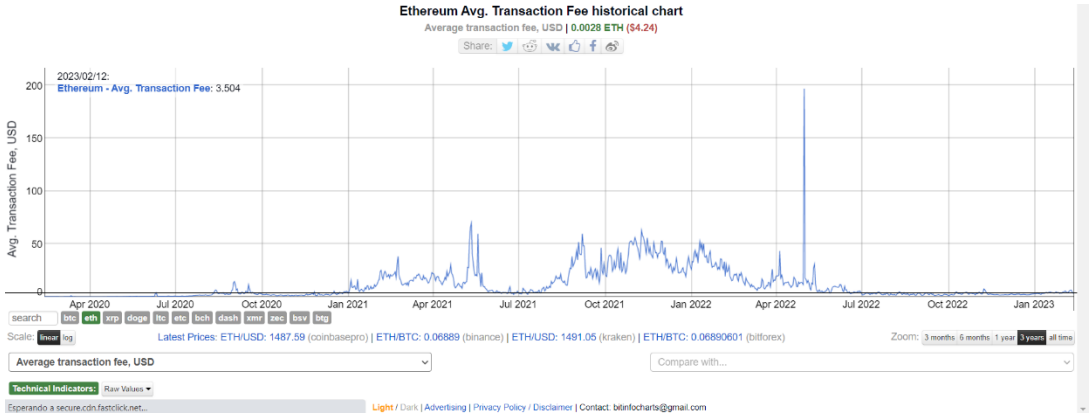


Figura 19 Fee de transacciones, tomado de https://bitinfocharts.com/comparison/ethereumransactionfees.html3y

Cálculo para una transacción sencilla en Ether (ETH)

Según (Cuchillero, 2022) Ethereum actualizó su mercado central de tarifas de gas con EIP-1559, un mecanismo de fijación de precios de transacciones que incluye una tarifa de red fija por bloque que se quema y expande/contrae dinámicamente los tamaños de bloque para lidiar con la congestión transitoria, las transacciones pasaron de una subasta de primer precio a un sistema híbrido que involucra tarifas base y propinas.

- El Base Fee, que viene determinado por la propia red. Y posteriormente se quema. Esto lo determina la red Ethereum en lugar de que lo establezcan los usuarios finales que buscan realizar transacciones o los mineros que buscan validar transacciones. El Base Fee objetivo es un 50% de bloques completos y se basa en el contenido del bloque confirmado más reciente. Dependiendo de qué tan lleno esté ese nuevo bloque, Base Fee automáticamente aumenta o disminuye.
- Max Priority Fee, que es opcional, determinado por el usuario, y se paga directamente a los mineros. Es técnicamente opcional, en este momento la mayoría de los participantes

de la red estiman que las transacciones generalmente requieren una propina mínima de 2.0 GWEI para ser candidatas para la inclusión.

- El Max Fee Per Gas, que es el máximo absoluto que está dispuesto a pagar por unidad de gas para que su transacción se incluya en un bloque.

Para obtener el coste de una transacción en dólares de ETH, calcularemos primero el coste de la transacción en Ether.

Para que de una cuenta Ethereum a otra, se transfiera 1 ether, debemos tener en cuenta:

baseFeePerGas = 10 gwei maxPriorityFeePerGas = 2 gwei unidades de Gas = 21000

(10 + 2) * 21000 = 252.000 gwei

o 0.000252 ETH

De la cuenta **A** se debitará 1.0042 ETH (donde 1ETH es para la cuenta **B** y 0.0042 ETH es el GAS)

A la cuenta **B** se le acreditará 1 ETH

La tarifa base se quemará -0.000252 ETH

El validador mantiene la propina +0.000210 ETH

Transacción de un contrato inteligente

Cuando los contratos son compilados se convierten en una serie de códigos de operación llamados ‘opcodes’ (operation codes). Estos códigos de operación (opcodes) se muestran bajo sus nombres mnemotécnicos tales como ADD (del inglés 'Addition'), o MUL (del inglés 'Multiplication'). Estos 'opcodes' los podemos encontrar en el Yellow Paper de Ethereum.

Name	Value	Description*
G_{zero}	0	Nothing paid for operations of the set W_{zero}.
G_{base}	2	Amount of gas to pay for operations of the set W_{base}.
$G_{verylow}$	3	Amount of gas to pay for operations of the set $W_{verylow}$.
G_{low}	5	Amount of gas to pay for operations of the set W_{low}.
G_{mid}	8	Amount of gas to pay for operations of the set W_{mid}.
G_{high}	10	Amount of gas to pay for operations of the set W_{high}.
$G_{extcode}$	700	Amount of gas to pay for operations of the set $W_{extcode}$.
$G_{balance}$	400	Amount of gas to pay for a BALANCE operation.
G_{sload}	200	Paid for a SLOAD operation.
$G_{jumpdest}$	1	Paid for a JUMPDEST operation.
G_{sset}	20000	Paid for an SSTORE operation when the storage value is set to non-zero from zero.
G_{sreset}	5000	Paid for an SSTORE operation when the storage value's zeroness remains unchanged or is set to zero.
R_{sclear}	15000	Refund given (added into refund counter) when the storage value is set to zero from non-zero.
$R_{selfdestruct}$	24000	Refund given (added into refund counter) for self-destructing an account.
$G_{selfdestruct}$	5000	Amount of gas to pay for a SELFDESTRUCT operation.
G_{create}	32000	Paid for a CREATE operation.
$G_{codedeposit}$	200	Paid per byte for a CREATE operation to succeed in placing code into state.
G_{call}	700	Paid for a CALL operation.
$G_{callvalue}$	9000	Paid for a non-zero value transfer as part of the CALL operation.
$G_{callstipend}$	2300	A stipend for the called contract subtracted from $G_{callvalue}$ for a non-zero value transfer.
$G_{newaccount}$	25000	Paid for a CALL or SELFDESTRUCT operation which creates an account.
G_{exp}	10	Partial payment for an EXP operation.
$G_{expbyte}$	50	Partial payment when multiplied by $\lceil \log_{256}(exponent) \rceil$ for the EXP operation.
G_{memory}	3	Paid for every additional word when expanding memory.
$G_{txcreate}$	32000	Paid by all contract-creating transactions after the *Homestead* transition.
$G_{txdatazero}$	4	Paid for every zero byte of data or code for a transaction.
$G_{txdatanonzero}$	68	Paid for every non-zero byte of data or code for a transaction.
$G_{transaction}$	21000	Paid for every transaction.
G_{log}	375	Partial payment for a LOG operation.
$G_{logdata}$	8	Paid for each byte in a LOG operation's data.
$G_{logtopic}$	375	Paid for each topic of a LOG operation.
G_{sha3}	30	Paid for each SHA3 operation.
$G_{sha3word}$	6	Paid for each word (rounded up) for input data to a SHA3 operation.
G_{copy}	3	Partial payment for *COPY operations, multiplied by words copied, rounded up.
$G_{blockhash}$	20	Payment for BLOCKHASH operation.
$G_{quaddivisor}$	100	The quadratic coefficient of the input sizes of the exponation-over-modulo precompiled contract.

Figura 20 Códigos de operación

Eventos y registros en Ethereum

Según (Consensys, 2020)Los eventos y registros en Ethereum facilitan la comunicación entre los contratos inteligentes y sus interfaces de usuario. En el desarrollo web tradicional, se proporciona una respuesta del servidor en una devolución de llamada a la interfaz. En Ethereum, cuando se extrae una transacción, los contratos inteligentes pueden emitir eventos y escribir registros en la cadena de bloques que luego puede procesar la interfaz.

Según (Elgarte, 2022) los Events, o eventos en español, permiten el uso de las funciones de logging que proporciona de manera nativa la Ethereum Virtual Machine (EVM) y que a su vez se pueden utilizar para retornar datos a nuestras dapps haciendo uso de JavaScript como handler de dichos eventos.

Cuando invocamos un evento dentro de nuestro contrato inteligente, los argumentos pasados se almacenan en un registro especial de la transacción. Esta estructura de datos especial se asocia a

la dirección del contrato y se incorpora a la blockchain y permanecen allí mientras se pueda acceder al bloque.