UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

DESARROLLO DE APLICACIÓN BLOCKCHAIN PARA PROYECTOS DE GENERACIÓN DISTRIBUIDA EN CHILE

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELÉCTRICO

RENÉ BASTIÁN SILVA VALDÉS

PROFESOR GUÍA:
RODRIGO PALMA BEHNKE

MIEMBROS DE LA COMISIÓN:
CESAR AZURDIA MEZA
MARCELO MATUS ACUÑA

SANTIAGO DE CHILE
2019

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO
POR: RENÉ BASTIÁN SILVA VALDÉS
FECHA: 2019
PROF. GUÍA: RODRIGO PALMA BEHNKE

DESARROLLO DE APLICACIÓN BLOCKCHAIN PARA PROYECTOS DE GENERACIÓN DISTRIBUIDA EN CHILE

El desarrollo de nuevas tecnologías en el sector energético como medidores inteligentes, generación distribuida, electromovilidad y el Internet de las cosas prometen traer beneficios tanto a a la sociedad como al sector privado. Sin embargo, esto también supone grandes y nuevos desafíos para la comunidad, ya sea en la búsqueda de soluciones que implementen estas tecnologías de manera eficiente o en la búsqueda de tecnologías unificadoras. En línea con esto último, el uso de la tecnología blockchain promete revolucionar el sector energético, integrando elementos como la digitalización, descentralización y la eliminación de intermediarios.

El presente trabajo desarrolla una aplicación blockchain para el sector energético chileno. Para esto se realiza una investigación de las aplicaciones de blockchain para el sector energía existentes a nivel global y local, así como un estudio del panorama energético chileno en torno a esta tecnología. De esta contextualización se corrobora que existen aplicaciones de blockchain para el sector energético tales como transferencia de energía *Peer to Peer* (P2P), aplicaciones para vehículos eléctricos y certificación de datos, siendo la primera la aplicación que explota de mejor manera las cualidades de la tecnología blockchain.

La aplicación desarrollada consiste en un sistema de economía colaborativa para agentes bajo un modelo organizacional de cooperativa, en pos de la realización de proyectos de generación distribuida. El desarrollo de esta plataforma se realiza utilizando la plataforma Ethereum y los resultados son obtenidos desplegando la aplicación en la testnet de Rinkeby, simulando el comportamiento de los agentes de forma aleatoria. Los resultados obtenidos prueban la viabilidad técnica de la aplicación y la estimación de un costo operacional de la aplicación inferior a 500 CLP por proyecto, con 100 agentes usando la plataforma.

La importancia de este trabajo consiste en mantener actualizado el sector energético chileno al explorar el desarrollo una tecnología nueva que promete ser revolucionaria. En el caso de blockchain, su potencial va en línea con el avance del sector energético nacional hacia la descentralización, digitalización y la generación distribuida en pos un futuro energético más sustentable. Se espera que en el futuro la existencia de aplicaciones blockchain en energía aumente a medida que proliferen otras tecnologías relativamente nuevas como las *smart grids*, la electromovilidad y el Internet de las cosas en Chile.

Tabla de Contenido

Índice de Tablas

Índice de Ilustraciones

Capítulo 1

Introducción

El presente trabajo está enfocado en el desarrollo de una aplicación basada en blockchain, o aplicación descentralizada (ĐApp), para el sector energético chileno. Para dicho desarrollo, se realiza una revisión de los avances existentes en torno a la tecnología blockchain y sus aplicaciones existentes relacionadas con el sector energético en el mundo, realizando un contraste con la realidad nacional. Luego, teniendo en cuenta aspectos que podrían limitar o favorecer la operación de estas aplicaciones en Chile, se realiza una propuesta ad hoc de ĐApp para el contexto energético chileno. Una vez definidas las características y el funcionamiento de la aplicación, se prototipa su implementación y se obtienen resultados mediante un caso de estudio de simulación, validando técnica y económicamente la ĐApp desarrollada.

1.1. Motivación

El desarrollo de tecnologías como la inteligencia artificial, la comunicación entre máquinas y el Internet de la cosas (IoT, por sus siglas en inglés) nos han llevado a la denominada cuarta revolución industrial, la cual está rompiendo las barreras entre los mundos físicos y virtuales a la gobernanza de blockchain como método criptográfico [1]. En una cadena de valor, una blockchain actúa como un protocolo que permite a las partes transferir activos de valor sin la necesidad de intermediarios. Por otra parte, la industria de la energía está experimentando un gran cambio hacia una producción y distribución de energía descarbonizada, descentralizada y digitalizada [2].

Tomando en cuenta estos antecedentes, resulta natural considerar la tecnología blockchain como una tecnología habilitadora para los cambios de paradigma proyectados para el sector energético. Sin embargo, la cantidad de trabajos existentes sobre blockchain en el sector energético es pequeña en comparación con los trabajos existentes en ambos tópicos por separado. En el caso de Chile, a la fecha, solo se tiene conocimiento de 2 aplicaciones de blockchain para el sector energético y se desconoce la existencia de trabajos de investigación en el ámbito académico. El presente trabajo busca contribuir al desarrollo de aplicaciones para el futuro del escenario energético nacional.

1.2. Alcances

Para el desarrollo del presente trabajo se recurrirá frecuentemente a la documentación existente sobre Bitcoin y Ethereum para explicar el funcionamiento de blockchain. Esto por ser los proyectos que abarcan la mayor parte del ecosistema de usuarios y desarrolladores de blockchain en la actualidad, además de ser las mejor documentadas. Asimismo, para la descripción del contexto energético nacional y la propuesta de aplicación, se considerarán proyectos solares de energía distribuida, a modo de acotar los contenidos tratados. A pesar de esto, las ideas tratadas en este trabajo serán, en general, extrapolables a energías renovables en general y a cualquier implementación de blockchain existente a la fecha.

1.3. Objetivos

1.3.1. Objetivo general

El objetivo general de esta memoria consiste en proponer y desarrollar aplicación de blockchain para el sector energético chileno, validándola técnicamente mediante una aplicación prototipo.

1.3.2. Objetivos específicos

1. Revisar la literatura existente sobre aplicaciones de blockchain en el sector energético y las plataformas disponibles para su desarrollo.
2. Estudiar el contexto de las cooperativas eléctricas y generación distribuida en Chile, relacionándolo con las aplicaciones blockchain existentes.
3. Proponer aplicación blockchain para el escenario energético nacional y una plataforma para su desarrollo.
4. Desarrollar implementación computacional de la aplicación blockchain propuesta.
5. Validar técnicamente la aplicación a través de un caso de estudio de simulación y comparar sus costos con los de una alternativa centralizada.

1.4. Metodología

La metodología propuesta para el cumplimiento de los objetivos del presente trabajo consta de seis etapas, las que se observan en el diagrama de la Figura 1.1 y se explican a continuación:

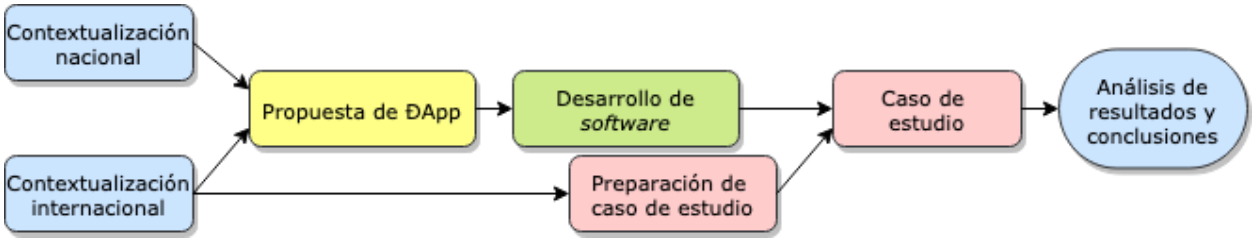


Figura 1.1: Metodología trabajo de título.

- **Contextualización internacional**: Investigación sobre la literatura y estado del arte sobre las aplicaciones basadas en blockchain en el sector energía. Mediante esta investigación se espera comprender las ventajas del uso de esta tecnología, así como las variables que influyen en su diseño e implementación.
- **Contextualización nacional**: Investigación sobre agentes del sector energético chileno y proyectos existentes que utilizan blockchain. Se espera poder aterrizar los contenidos de la literatura internacional y contrastarlos con la realidad nacional y las aplicaciones existentes.
- **Propuesta de ĐApp**: En base a los contextos nacional e internacional, a los enfoques existentes y los resultados de trabajos anteriores, se realizará una propuesta de ĐApp que se ajuste a las oportunidades identificadas en la utilización de blockchain en el sector energético chileno.
- **Desarrollo de software**: Considera primero una investigación de las plataformas existentes para el desarrollo de ĐApps con el fin de obtener nociones sobre el diseño de software de aplicaciones blockchain. Se continúa con la programación de la propuesta de ĐApps realizada previamente.
- **Preparación de caso de estudio**: Preparación del caso de estudio o simulación del software desarrollado. En esta etapa se realizan supuestos y se determinan todos los *inputs* necesarios para la obtención de resultados. Los datos proporcionados en esta etapa deben ser consistentes con el contexto energético chileno estudiado previamente.
- **Caso de estudio**: Simulación del software desarrollado. En esta etapa se obtendrán los resultados bajo las condiciones determinadas en las etapas anteriores, los cuales permitirán analizar y concluir sobre la propuesta de ĐApps realizada.
- **Análisis de resultados y conclusiones**: Se analizarán los resultados obtenidos en la etapa anterior para elaborar la conclusión del trabajo de título. También se harán recomendaciones para trabajos futuros que sigan la línea del mismo trabajo.

1.5. Estructura de la memoria

Esta memoria presenta las siguientes secciones:

- **Capítulo 1. Introducción**: Se introduce y motiva al lector en el tema investigado y se presentan los alcances, objetivos y estructura de la memoria.
- **Capítulo 2. Contextualización y marco teórico**: En este capítulo se describen los conceptos generales básicos que dan marco a la memoria.

- **Capítulo 3. Implementación**: Se describe la aplicación blockchain diseñada para el contexto descrito en el capítulo 2. Se detallan las herramientas utilizadas para su desarrollo y se explican los algoritmos implementados en los *smart contracts* creados.
- **Capítulo 4. Resultados y análisis**: Se lleva a cabo un análisis del funcionamiento de la aplicación desarrollada para un caso de aplicación simulado, observando variables de interés para una posible implementación de esta en el mundo real.
- **Capítulo 5. Conclusiones**: Se presentarán las conclusiones del trabajo a la luz de los resultados obtenidos. También se entregarán las directrices para el perfeccionamiento de la aplicación desarrollada.

Capítulo 2

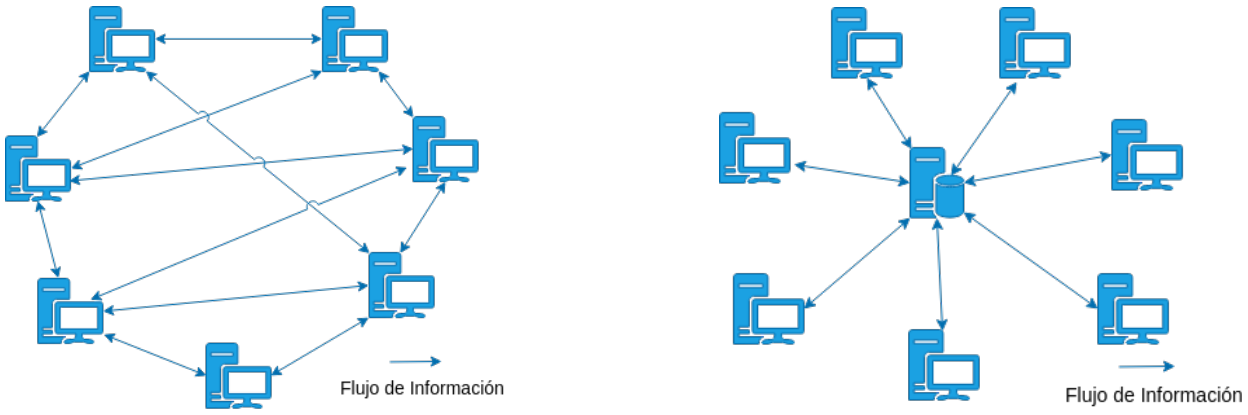
Contextualización y marco teórico

El objetivo del presente capítulo es ubicar al lector en el entorno en el cual se desarrolla este trabajo de título. Primeramente, se explican resumidamente los fundamentos teóricos que dan soporte a la tecnología blockchain, tales como redes *Peer to Peer (P2P)*, *hashes* criptográficos y arboles de Merkle. A continuación se explica en detalle el concepto de blockchain, pasando por su evolución desde su origen hasta la actualidad y detallando su funcionamiento. Luego se trata la temática de blockchain en el sector energía, indicando cuales son son sus potenciales aportes y las aplicaciones existentes en distintas zonas geográficas, indicando sus ventajas en comparación con los sistemas centralizados. Finalmente se describe de manera general el sector energético en Chile, describiendo a los agentes y la normativa existente que puedan influir o verse afectados con la implementación de blockchain en el sector eléctrico. También se describen a grandes rasgos las aplicaciones de blockchain existentes del sector energético chileno.

2.1. Fundamentos teóricos

2.1.1. Redes P2P

Los sistemas *Peer to Peer* son sistemas distribuidos que consisten en nodos interconectados capaces de autoorganizarse en topologías de red con el fin de compartir recursos como contenido, ciclos de CPU, almacenamiento y ancho de banda, sin requerir la intermediación o el soporte de un servidor o autoridad centralizada global [3]. Las redes *P2P* son una tecnología disruptiva para aplicaciones distribuidas a gran escala que ha ganado un gran interés últimamente debido al éxito que ha tenido el intercambio de contenido *P2P*, la transmisión de medios y las aplicaciones de telefonía [4]. Las arquitecturas subyacentes comparten características como la descentralización, el intercambio de recursos del sistema final, la autonomía, la virtualización y la autoorganización. La figura 2.1a muestra una arquitectura P2P, mientras que la figura 2.1 muestra la arquitectura cliente servidor.



(a) Arquitectura P2P.

(b) Arquitectura cliente servidor.

Figura 2.1: Arquitecturas P2P y cliente servidor

A diferencia de una arquitectura cliente servidor, en una arquitectura P2P hay una mínima dependencia de los servidores dedicados en *data centers*. En su lugar, las aplicaciones explotan la comunicación directa entre pares de *hosts* (o *peers*) conectados [5]. Los pares no son propiedad del proveedor del servicio, sino que son computadoras controladas por los usuarios. Muchas de las aplicaciones más populares y con uso intensivo de tráfico de la actualidad se basan en arquitecturas P2P. Estas aplicaciones incluyen intercambio de archivos (e.g BitTorrent), aceleración de descarga asistida por pares (e.g Xunlei), telefonía por Internet (e.g Skype) e IPTV (Internet Protocol Television). A pesar de las ventajas de las arquitecturas P2P, tales como la escalabilidad o rentabilidad, estas se enfrentan a los siguientes desafíos:

- La mayoría de los ISP residenciales se han dimensionado para el uso de ancho de banda "asimétrico", es decir, para mucho más flujo descendente que tráfico ascendente. Sin embargo, las aplicaciones de transmisión de archivos y transmisión de archivos P2P desplazan el tráfico ascendente de los servidores a los ISP residenciales. Las futuras aplicaciones P2P deben diseñarse de modo que sean amigables con los ISP [6].
- Debido a su naturaleza altamente distribuida y abierta, las aplicaciones P2P pueden ser un desafío para la seguridad [7].
- El éxito de las futuras aplicaciones P2P también depende de convencer a los usuarios para que ofrezcan recursos de ancho de banda, almacenamiento y computación para las aplicaciones, que es el desafío del diseño de incentivos [8].

2.1.2. Criptografía

Criptografía puede definirse como las técnicas matemáticas utilizadas para proteger la información digital, los sistemas y los cálculos distribuidos contra ataques adversos [9]. Estas técnicas permiten al emisor camuflar los datos del mensaje de modo que potenciales intrusos no puedan obtener ninguna información a partir de los datos interceptados, mientras que el receptor deberá ser capaz de recuperar los datos originales a partir de los datos ocultados [5]. Los elementos principales que se pueden identificar al hablar sobre criptografía son emisor, receptor, mensaje y llaves. Parte de esta terminología se ilustra en la Figura 2.2.

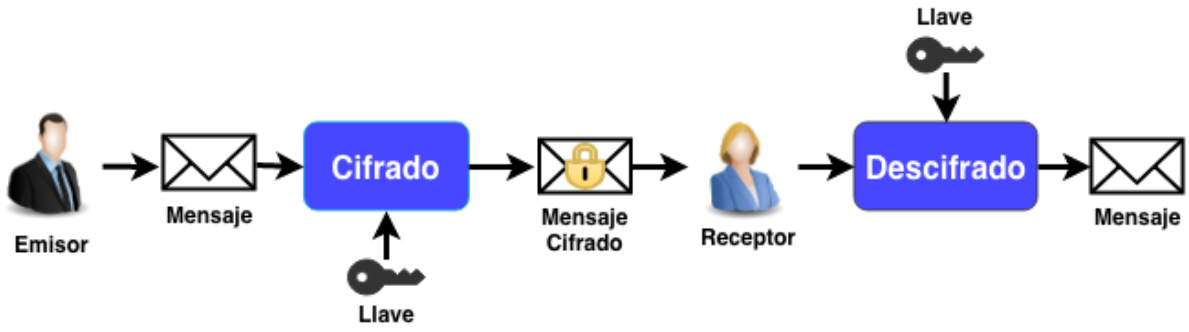


Figura 2.2: Esquema criptografía

Como se aprecia en la imagen, el emisor, quien desea que únicamente el receptor sea capaz de leer el mensaje a enviar, utiliza una llave de cifrado para alterar las representaciones lingüísticas del mensaje (encriptación) para luego enviarlo por un canal no confiable. Por su parte, el receptor utilizará una llave propia para recuperar el mensaje original a partir del mensaje encriptado (desencriptación). En el Anexo A se muestra una ejemplificación de como podría definirse un sistema críptográfico con los elementos mencionados. Según las llaves utilizadas por el emisor y el receptor se distingue entre criptografía simetrica y asimetrica.

Criptografía simétrica

La criptografía simétrica, también llamada criptografía de clave secreta, es un método criptográfico en el cual ambos socios de comunicación (emisor y receptor) utilizan la misma llave k para el cifrado y descifrado [10]. Para configurar un canal de comunicación seguro, emisor y receptor deben acordar primero una clave k, que mantienen en secreto. Antes de enviar un mensaje m al receptor, el emisor cifra m utilizando el algoritmo de cifrado E y la llave k, obteniendo el texto cifrado $c = E(k, m)$, el cual envía al receptor. Mientras que el receptor usa el algoritmo de descifrado D y la misma clave k para recuperar el mensaje $m = D\ (k, c)$. Este esquema se define formalmente como un mapeo.

$$E\ :\ K\ \times\ M\ \mapsto\ C,$$

, tal que para k ∈ K, el mapeo

$$E_k\ : M\ \mapsto\ C,\ m\ \mapsto\ E(k, m),$$

Es invertible. Los elementos m $\in M$ son los mensajes. C es el conjunto de textos cifrados, los elementos $k \in K$ son las llaves. E_k es la función de cifrado con respecto a la clave k. A la función inversa $D_k = E^{-1}$ se le llama función de descifrado. Se supone que existen algoritmos eficientes para calcular E_k y D_k.

Este esquema corresponde al mostrado en el Anexo A. Entre todos los algoritmos de cifrado, los algoritmos de cifrado de clave simétrica tienen las implementaciones más rápidas en hardware y software, por lo que son muy adecuados para el cifrado de grandes cantidades

de datos [10]. Un problema básico en un esquema simétrico es cómo emisor y receptor pueden ponerse de acuerdo sobre una clave secreta compartida k de una manera segura y eficiente. Este problema permaneció sin soluciones hasta que se descubrió la criptografía asimétrica o de clave pública.

Criptografía asimétrica

La criptografía asimétrica, también llamada criptografía de clave pública, es un método criptográfico en donde los socios de comunicación no comparten una clave secreta. Cada usuario tiene un par de claves: una clave secreta conocida sólo por él y una clave pública conocida por todos [10]. Supongamos que B tiene un par de llaves (p_k, s_k) y A quiere cifrar un mensaje m para B. Al igual que todos, A conoce la llave pública de B p_k. A calcula el texto cifrado $c = E(p_k, m)$ aplicando la función de cifrado conocida E, con la clave pública de B p_k. La función de cifrado E_{p_k} debe tener la propiedad de que la pre-imagen m del texto cifrado $c = E_{p_k}(m)$ sea fácil de calcular utilizando la llave secreta de B s_k. Como sólo B conoce la llave secreta, él es el único que puede descifrar el mensaje. Incluso A, que cifró el mensaje m, no podría obtener m de $E_{p_k}(m)$ si perdiese el mensaje m. En una blockchain se utiliza este esquema criptográfico con dos finalidades:

- **Identificar cuentas**: Blockchain necesita identificar usuarios o cuentas de usuarios para mantener el mapeo entre el propietario y la propiedad. La blockchain utiliza un enfoque "público a privado"de la criptografía asimétrica para identificar cuentas de usuarios y transferir la propiedad entre ellas. Los números de cuenta en la blockchain son en realidad llaves criptográficas públicas. Por lo tanto, los datos de transacción utilizan las llaves criptográficas públicas para identificar las cuentas involucradas en la transferencia de propiedad [11].
- **Validar transacciones**: Los datos de transacción siempre tienen que incluir un dato que sirva como prueba de que el propietario de la cuenta que transfiere propiedad está de acuerdo con la transferencia de propiedad descrita. El flujo de información implícito en este acuerdo comienza en el propietario de la cuenta que transfiere propiedad y se supone que llega a todos los que inspeccionan los datos de la transacción. Este tipo de flujo de información es similar al caso de uso "privado a público"de la criptografía asimétrica. El propietario de la cuenta que transfiere la propiedad crea un texto cifrado con su clave privada. Todos los demás pueden verificar esta prueba de acuerdo mediante el uso de la clave criptográfica pública, que es el número de la cuenta que entrega la propiedad (firma digital) [11].

Funciones *hash* criptográficas

Las funciones *hash* son programas computacionales que toman una entrada m y calculan un *string* de tamaño fijo $H(m)$ conocido como *hash* [5]. Los valores *hash* pueden tener ceros a la izquierda para proporcionar la longitud requerida. Hablamos de una **función *hash* criptográfica** cuando una función *hash* cumple las siguientes propiedades:

- Proporciona valores *hash* para cualquier tipo de datos rápidamente.
- Produce idénticos valores *hash* para datos de entrada idénticos.
- El valor *hash* devuelto cambia impredeciblemente cuando se cambian los datos de entrada.
- No proporciona ninguna forma de rastrear sus valores de entrada a partir de sus salidas.
- Es muy difícil encontrar dos o más datos distintos para los que arroja el mismo valor *hash*.

Debido a estas propiedades, las funciones *hash* criptográficas pueden crear *fingerprints* o "huellas digitales" para cualquier tipo de datos [11]. De esta manera, los valores *hash* pueden ser utilizados para comparar, referenciar y verificar cambios en los datos. Además, los valores *hash* pueden ser usados para la creación de *puzzles* criptográficos, en donde un computador desafía a otro(s) a encontrar el *hash* de algún dato que cumpla con ciertas condiciones especificadas previamente. A continuación se presentan algunos ejemplos de *hashes* obtenidos usando la función $Keccak - 256$, utilizada en la blockchain de Ethereum:

$$H(\text{ele}fante) \mapsto 381b770038ccb96b90f21842\text{d}97684866c6\text{e}240\text{e}9685c76\text{e}32\text{e}046c613\text{d}7a3c0$$
$$H(\text{ele}gante) \mapsto \text{e}92a1\text{d}46bf0a\text{e}31\text{d}c90795220a836c534698f\text{e}02f4\text{d}c51f5f41\text{e}10f55a073884$$
$$H(\text{ele}fantes) \mapsto 8\text{e}83a45\text{d}bd2a223183c9a2\text{d}9f1\text{e}8\text{e}23f0a0f099906\text{d}f67b96f\text{e}7f\text{d}36f111cfbf$$

Como se puede apreciar, a pesar de la similitud de las tres entradas, se obtienen *hashes* totalmente distintos y de un largo fijo. Para mayor seguridad, un emisor A podría obtener el *hash* h de su mensaje m y adjuntarlo. Luego cifrar este paquete (m, h) con su llave privada, y nuevamente con la clave pública de su receptor B. Cuando este reciba y descifre el texto cifrado de A, puede ejecutar el mismo algoritmo de *hash* usado por A sobre el mensaje. Si por alguna razón la "huella digital" del mensaje resulta diferente, entonces significa que el texto del mensaje real fue dañado o alterado en el camino [11]. Este principio constituye el principio de las firmas digitales, ilustradas en la figura 2.3.

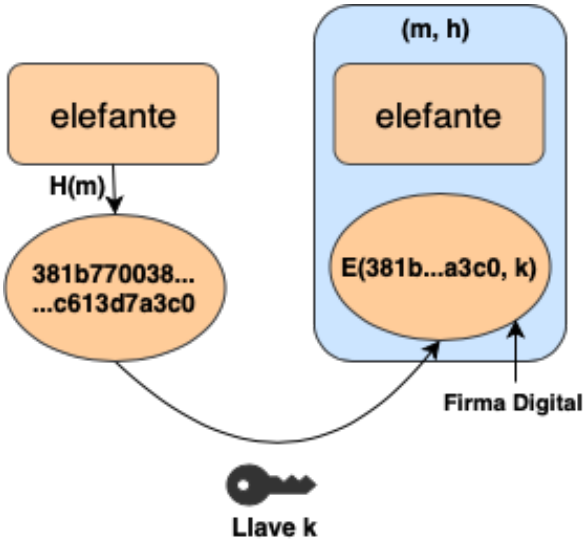


Figura 2.3: Esquema firma digital.

Dos de los principales valores que posee una blockchain para transacciones de activos económicos (como Bitcoin o Ethereum) son la seguridad y la inmutabilidad de los datos, cualidades que vienen dadas por el uso de la criptografía en su implementación, en particular de la criptografía asimétrica y los valores *hash* criptográficos. Mientras la criptografía asimétrica es utilizada para identificar a los usuarios y autorizar las transacciones utilizando las llaves pública y privada respectivamente, los valores *hash* son utilizados para almacenar la información de manera sensible al cambio, como huellas digitales de las transacciones y como forma de introducir costos computacionales a los procesos mediante *puzzles* criptográficos [11].

Arboles de Merkle

Un arbol de Merkle es una estructura de datos que se utiliza para resumir y verificar de manera eficiente la integridad de grandes conjuntos de datos [12]. Los árboles de Merkle son árboles, binarios o no, que contienen *hashes* criptográficos. El término "árbol" se usa en ciencias de la computación para describir una estructura de datos de ramificación, pero estos árboles generalmente se muestran al revés con la "raíz" en la parte superior y las "hojas" en la parte inferior en un diagrama, como se muestra en la Figura 2.4.

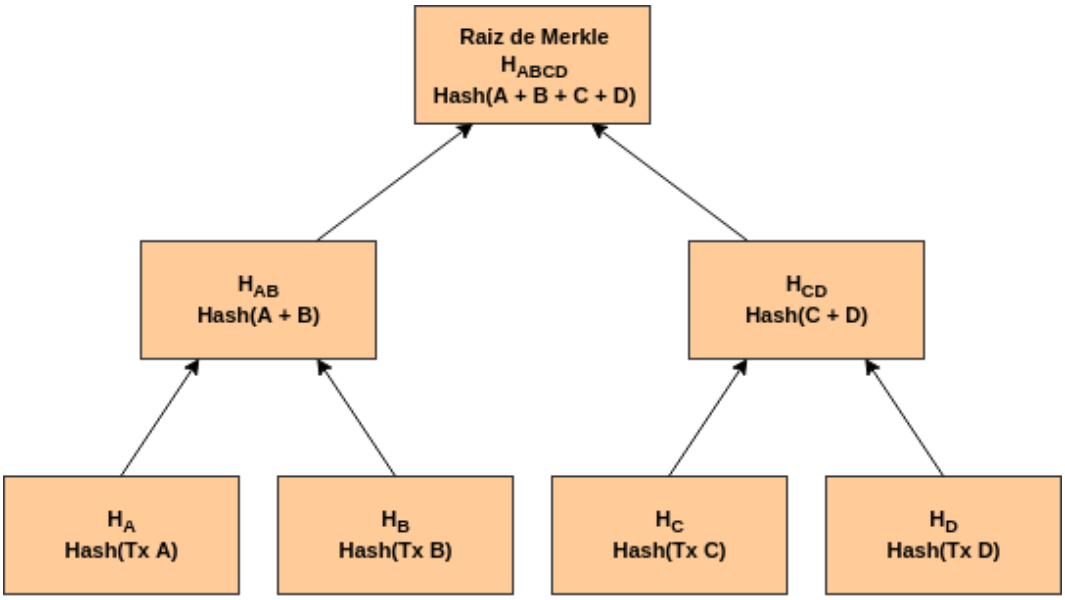


Figura 2.4: Arbol de Merkle.

Cuando N datos se incluyen y se resumen en un árbol de Merkle, se puede verificar si alguno de los datos está incluido en el árbol con un máximo de $2 * log2(N)$ cálculos, lo que hace que esta sea una estructura de datos muy eficiente. El árbol de Merkle está construido de abajo hacia arriba. En el ejemplo de la Figura 2.4, se comienza con cuatro transacciones A, B, C y D, que forman las hojas del árbol de Merkle en la parte inferior del diagrama. Las transacciones no se almacenan en el árbol de Merkle, sino que se copian sus datos y el *hash* resultante se almacena en cada nodo de hoja como H_A, H_B, H_C y H_D.

Los pares consecutivos de nodos hoja se agregan luego en un nodo padre, concatenando ambos *hashes* y *hasheandolos*. Por ejemplo, para construir el nodo padre H_{AB}, los dos *hashes* de 32 bytes de los hijos se concatenan para crear una cadena de 64 bytes. Luego, esa cadena tiene doble *hash* para producir el *hash* del nodo padre. El proceso continúa hasta que solo hay un nodo en la parte superior, el nodo conocido como la raíz de Merkle. Ese *hash* de 32 *bytes* se almacena en el encabezado del bloque y resume todos los datos en las cuatro transacciones.

2.2. Blockchain

Una blockchain (o cadena de bloques) es una red de software *Peer to Peer* distribuida que utiliza criptografía para alojar de forma segura aplicaciones, almacenar datos y transferir fácilmente instrumentos digitales de valor que representan dinero del mundo real [13]. En una blockchain, la información contenida es agrupada en bloques de datos, los cuales contienen metadatos (datos sobre los datos) de su bloque predecesor en una línea temporal. Las principales ventajas de una blockchain son la posibilidad de automatizar transacciones de activos y de almacenar datos de transacciones de forma distribuida, segura e inmutable. Esto es logrado mediante la utilización de redes *Peer to Peer*, criptografía y algoritmos de consenso. Cada uno de estos conceptos serán profundizados en esta sección, así como las blockchain de Bitcoin y Ethereum.

2.2.1. Desarrollo

El concepto blockchain apareció junto con las criptomonedas. Una Criptomoneda es un sistema de intercambio digital *Peer to Peer* en el que se utiliza la criptografía para generar y distribuir unidades de esta [14]. Las criptomonedas surgieron de la necesidad de tener un sistema eficiente, rentable, confiable y seguro para realizar y registrar transacciones financieras. Una de las primeras soluciones desarrolladas para abordar estas complejidades fue bitcoin, una criptomoneda lanzada en 2009. Bitcoin se define como una cadena de firmas digitales en un red *Peer to Peer* [15]. Algunas ventajas de Bitcoin por sobre otros sistemas de transacción son listadas a continuación:

- **Rentabilidad**: Elimina la necesidad de intermediarios, reduciendo costos. La minería de bitcoin descentraliza las funciones de emisión y compensación de divisas de un banco central y reemplaza la necesidad de cualquier banco central [12].
- **Eficiencia**: La información de la transacción se registra una vez y está disponible para todas las partes a través de la red distribuida.
- **Seguridad**: El libro contable subyacente es inviolable. Si los nodos honestos controlan la mayor parte de la capacidad de cómputo, un atacante no podrá modificar los registros.[15]

Un componente fundamental del sistema de Bitcoin es la cadena de bloques o blockchain, la cual funciona como un libro de contabilidad compartido de los bitcoin transados [16].

La Figura 2.5 muestra una estructura simplificada de los bloques de transacciones en la blockchain de Bitcoin. Este libro de contabilidad compartido que constituye la blockchain, puede generalizarse y ser empleado para registrar transacciones de cualquier tipo y rastrear el movimiento de cualquier activo, ya sea tangible, intangible o digital. Esta abstracción da lugar a una "separación conceptual" entre blockchain y bitcoin (o entre blockchain y criptomonedas) y a nuevos usos de la tecnología blockchain.

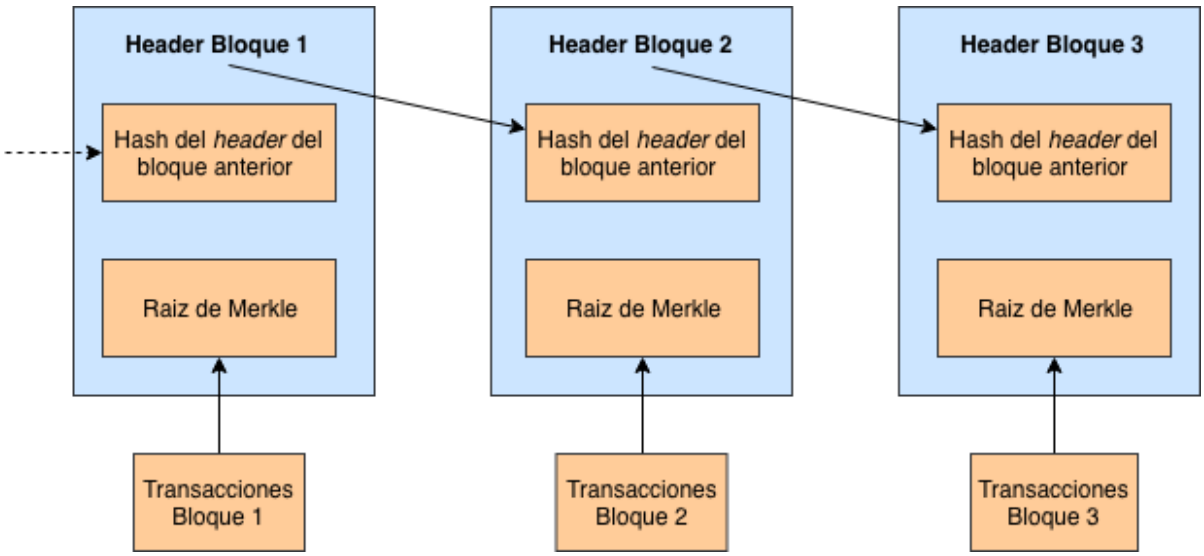


Figura 2.5: Cadena de bloques simplificada: Un bloque de una o más transacciones nuevas se recopila en la parte de datos de transacción de un bloque. Se *hashean* las copias de cada transacción, y los *hashes* se emparejan, se *hashean*, se vuelven a emparejar y se vuelven a *hashear* hasta que queda un solo *hash*, la raíz de Merkle. La raíz de Merkle se almacena en el encabezado del bloque. Cada bloque también almacena el *hash* del encabezado del bloque anterior, encadenando los bloques juntos. Esto garantiza que una transacción no se pueda modificar sin modificar el bloque que la registra y todos los bloques siguientes.

A medida el modelo de Bitcoin comenzó a ser reconocido, los desarrolladores intentaron ir más allá de las aplicaciones de criptomonedas. Fue así como en 2015 fue lanzada la blockchain de Ethereum [17]. La blockchain de Ethereum, al igual que Bitcoin, es una máquina de estados distribuida. Pero en lugar de monitorear solamente la propiedad de una moneda, Ethereum monitorea las transiciones de estado de los datos almacenados. Ethereum crea pequeños programas informáticos, llamados *smart contracts* que permiten representar instrumentos financieros, como préstamos o bonos, en lugar de solo el efectivo. Al igual que un computador de propósito general, Ethereum puede cargar código en su máquina de estado y ejecutar ese código, almacenando los cambios de estado resultantes en la blockchain [18]. La Tabla 2.1 muestra una comparación entre Bitcoin y Ethereum.

	Bitcoin	**Ethereum**
Fecha de lanzamiento	Enero, 2009	Junio, 2015
Utilidad	Comprar bienes y servicios, almacenar valor.	Crear ĐApps (aplicaciones descentralizadas).
Tiempo de emisión de cada nuevo *token*	Cada 10 minutos aprox.	Cada 10 - 20 segundos
Propósito	Criptomoneda creada para competir contra el estándar del oro y las monedas fiat.	Un *token* capaz de facilitar contratos inteligentes (e.g un intercambio de propiedad).

Tabla 2.1: Comparación entre Bitcoin y Ethereum.

Ethereum comenzó como una forma de hacer una blockchain de propósito general que podría programarse para una variedad de usos. Pero muy rápidamente, la visión de Ethereum se expandió para convertirse en una plataforma para la programación de ĐApps, que representan una perspectiva más amplia que los *smart contracts*. Una ĐApp es, en términos generales, una aplicación web que se construye sobre una infraestructura de servicios *peer to peer*, abiertos y descentralizados [18]. Estas aplicaciones buscan aprovechar las ventajas de esta estructura como la eliminación de terceros en una cadena de producción o la trazabilidad de activos. Una ĐApp se compone de al menos:

- Contratos inteligentes en una cadena de bloques
- Una interfaz de usuario web frontend

Además, muchos DApps incluyen otros componentes descentralizados, tales como:

- Un protocolo y plataforma de almacenamiento descentralizado (P2P).
- Un protocolo y plataforma de mensajería descentralizada (P2P).

Uno de los principales impedimentos para la implementación de estas ĐApps es la complejidad y el costo energético del algoritmo de minado empleado (Proof of Work (PoW) o prueba de trabajo en español), lo que ha dado lugar a algoritmos más eficientes como Proof of Stake (PoS), los cuales reducen significativamente el gasto energético en el minado de transacciones. Ambos algoritmos serán descritos en el presente trabajo. Otra limitación de estas tecnologías recae en la falta de escalabilidad, ya que en una blockchain, cada computadora de la red procesa cada transacción, lo cual es un proceso lento. Algunas de las tasas de transacciones de distintos sistemas de transacciones son listadas a continuación:

- Blockchain de Bitcoin: Aprox. 15 Transacciones por segundo.
- Blockchain de Ethereum: Aprox. 100 Transacciones por segundo.
- Red de Visa: Aprox 60.000 Transacciones por segundo

A pesar de lo anterior, un gran número de sectores de la economía están interesados en las ventajas que proporcionan las tecnologías basadas en blockchain, lo cual ha dado lugar a una creciente comunidad de desarrolladores y a la creación de distintas herramientas para el

desarrollo de ĐApps. Con respecto a la escalabilidad, se espera que una blockchain escalable aumente la tasa de transacciones, sin sacrificar la seguridad, al calcular cuántas computadoras son necesarias para validar cada transacción y dividir el trabajo de manera eficiente.

2.2.2. Terminología

La tecnología blockchain es un campo relativamente nuevo, el cual ha traído consigo una nueva terminología. Algunas de las palabras encontradas con más frecuencia al hablar de blockchain son listadas en el Anexo B del presente documento.

2.2.3. Algoritmos de consenso

En un sistema distribuido conformado por procesos $p_{\mathrm{i}}(\mathrm{i} = 1, 2, ..., N)$ que se comunican mediante *broadcast*, un requisito importante que se aplica en muchas situaciones es que se alcance un consenso entre los procesos [19]. Este problema se define de la siguiente manera:

"Para llegar a un consenso, cada proceso p_{i} *comienza en un estado indeciso y propone un solo valor* v_{i}*, extraído de un conjunto* $D(\mathrm{i} = 1, 2, ..., N)$*. Los procesos se comunican entre sí, intercambiando valores. Cada proceso luego establece el valor de una variable de decisión,* $\mathrm{d_i}$*. Al hacerlo, entra en el estado decidido, en el que ya no puede cambiar* $\mathrm{d_i}(\mathrm{i} = 1, 2, ..., N)$*."*

En una blockchain, el consenso no se logra explícitamente, pues no hay elección o momento fijo en el que se produce el consenso. Sino que éste es un artefacto emergente de la interacción asíncrona de los nodos independientes, todos siguiendo reglas simples [12]. El consenso lleva a que todos los nodos compartan exactamente los mismos datos de transacciones. Un algoritmo de consenso, por lo tanto, hace dos cosas:

- Asegura que los datos en el libro contable son los mismos para todos los nodos de la red y, a su vez,
- Evita que los actores malintencionados manipulen los datos.

Si bien las firmas digitales proporcionan parte de la solución al problema de consenso, sigue siendo necesario una herramienta para evitar el problema de doble gasto [15]. El problema de doble gasto es una falla potencial en una criptomoneda u otro esquema de efectivo digital por el cual el mismo *token* digital único puede gastarse más de una vez, lo cual es posible ya que un *token* digital es un archivo digital que puede ser duplicado o falsificado [20]. La Figura 2.6 muestra el problema de doble gasto en la estructura de una blockchain.

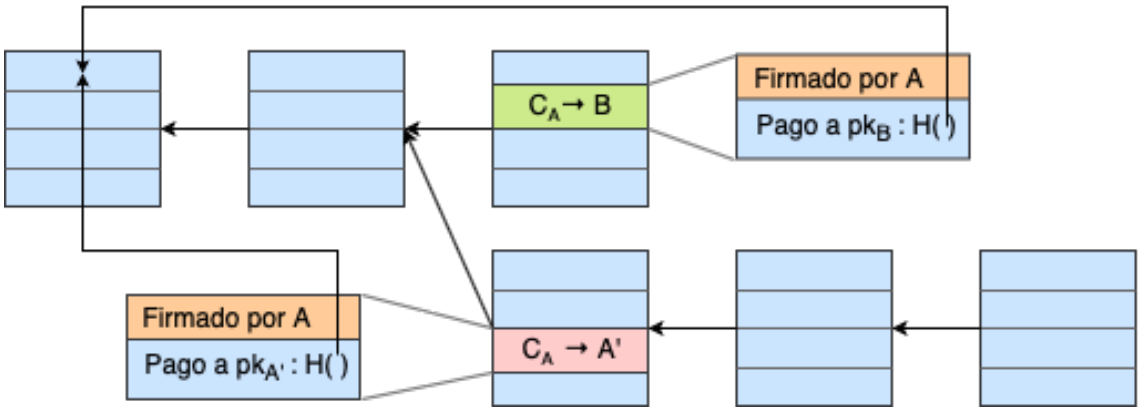


Figura 2.6: Intento de doble gasto: A crea dos transacciones: una en la que transfiere criptomonedas a a B y otra en la que transfiere esas criptomonedas a una dirección diferente que controla. Al gastar las mismas criptomonedas, solo una de estas transacciones puede incluirse en la cadena de bloques. Las flechas son punteros de un bloque al bloque anterior. C_A denota una moneda propiedad de A.

El algoritmo de consenso varía según las diferentes implementaciones de blockchain. En una blockchain de carácter pública como Bitcoin o Ethereum, en donde cualquier nodo puede agregarse a la red, es necesario dificultar la labor de posibles nodos deshonestos. En cambio, en una blockchain privada el proceso de consenso solo se puede alcanzar con un número limitado y predefinido de participantes.[21] Algunos de los procesos involucrado en el consenso en una blockchain son:

- Verificación independiente de cada transacción, por cada nodo completo, basada en una lista completa de criterios.
- Agregación independiente de esas transacciones en nuevos bloques por los *miners*.
- Verificación independiente y ensamblado en la cadena de los nuevos bloques por cada nodo.

En una blockchain pública existen nodos especializados llamados *miners* o mineros. Al igual que cualquier otro nodo completo, un *miner* recibe y propaga transacciones no confirmadas en la red. Sin embargo, los miners también agregan estas transacciones en nuevos bloques [12]. Después de ser validadas por un nodo, las transacciones son agregadas a una "memoria" o conjunto de transacciones, donde las transacciones esperan hasta que se puedan incluir (minar) en un nuevo bloque. Los bloques pueden ser agregados según distintas reglas según la implementación de la blockchain. Los algoritmos más utilizados son:

- Proof of Work (PoW): Este algoritmo aprovecha la naturaleza aparentemente aleatoria de los *hash* criptográficos. Si los datos se modifican y el *hash* se vuelve a ejecutar, se producirá un nuevo número aparentemente aleatorio, por lo que no hay forma de modificar los datos para hacer que el número de *hash* sea predecible. Para demostrar que se realizó un trabajo adicional para crear un bloque, se debe crear un *hash* del encabezado del bloque que no exceda un cierto valor llamado *target*. Agregar una sola letra, un signo de puntuación o cualquier carácter, llamado *nonce* producirá un *hash* diferente. De esta manera un *miner* ejecutará el algoritmo mostrado en la Figura 2.7.

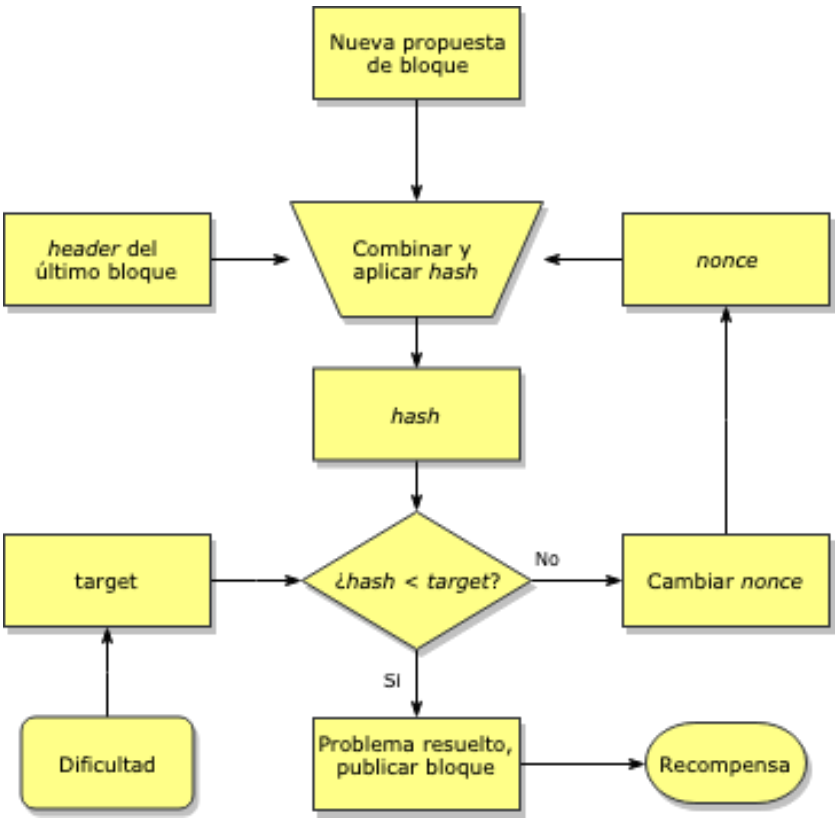


Figura 2.7: Algoritmo Proof of Work.

Por convención, la primera transacción en un bloque es una transacción especial que crea una nueva moneda que es propiedad del creador del bloque. Esto agrega un incentivo, el cual puede ayudar a alentar a los nodos a ser honestos [15]. Si un atacante codicioso es capaz de reunir más poder de CPU que todos los nodos honestos, tendría que elegir entre usarlo para defraudar a las personas robando sus pagos, o usarlo para generar nuevas monedas.

Consideramos el escenario de un atacante que intenta generar una cadena alternativa más rápido que la cadena honesta. Incluso si esto se logra, no abre el sistema a cambios arbitrarios, como crear valor de la nada o sacar dinero que nunca perteneció al atacante. Los nodos no aceptarán una transacción no válida como pago, y los nodos honestos nunca aceptarán un bloque que los contenga. Un atacante solo puede intentar cambiar una de sus propias transacciones para recuperar el dinero que gastó recientemente (doble gasto) [15]. La carrera entre la cadena honesta y una cadena de atacantes puede caracterizarse como:

$$q_z = \begin{cases} 1 & \text{si } p \leq q. \\ (q/p)^z & \text{si p >q.} \end{cases}$$

p = probabilidad de que un nodo honesto encuentre el siguiente bloque.

q = probabilidad de que el atacante encuentre el siguiente bloque.

q_z = probabilidad de que el atacante se recupere de una ventaja de z bloques.

Dado que suponemos que $p > q$, la probabilidad disminuye exponencialmente a medida que aumenta el número de bloques que el atacante tiene que alcanzar. Con las probabilidades en contra de él, sus posibilidades se vuelven cada vez más pequeñas a medida que se queda atrás. De esta forma, resulta computacionalmente impráctico que un atacante cambie el registro si los nodos honestos controlan la mayoría de la potencia de la CPU de la red [15].

- Proof of Stake (PoS): Proof of Stake es una categoría de algoritmos de consenso para blockchain que dependen del interés económico de un validador en la red (*stake*). La "prueba de interés" se basa en la antigüedad de la moneda y es generada por cada nodo a través de un esquema de *hashing* similar al de Bitcoin, pero sobre un espacio de búsqueda limitado. El historial de la cadena de bloques y la liquidación de transacciones están protegidos por un mecanismo de punto de control de difusión central [22]. La edad de la moneda se define simplemente como el monto de la moneda por el período de tenencia.

  Los nodos se seleccionan aleatoriamente para validar bloques, y la probabilidad de esta selección aleatoria depende de la cantidad de participación que se tenga. La "prueba de interés" en los bloques es una transacción especial llamada *coinstake* [22]. En ella, el propietario del bloque se paga a sí mismo, consumiendo su edad de la moneda, mientras obtiene el privilegio de generar un bloque para la red y acuñar para la prueba de juego. La primera entrada de coinstake se llama kernel y se requiere que cumpla con ciertos protocolos de *hash target*, lo que hace que la generación de bloques de proof of stake sea un proceso estocástico similar al de los bloques de prueba de trabajo. Sin embargo, una diferencia importante es que la operación de *hash* se realiza a través de un espacio de búsqueda limitado en lugar de un espacio de búsqueda ilimitado como en Proof of Work, por lo tanto, no hay un consumo significativo de energía involucrado [22].
- Proof of Authority (PoA): La generación de bloques con PoA requiere otorgar un permiso especial a uno o más miembros para realizar cambios en la cadena de bloques [23]. Por ejemplo, un miembro que tiene una clave especial puede ser responsable de generar todos los bloques. Esencialmente, PoA puede verse como un algoritmo de PoS modificado, donde la participación de los validadores es su propia identidad. Los miembros de la red depositan su confianza en nodos autorizados y se acepta un bloqueo si la mayoría de los nodos autorizados firma el bloqueo. Cualquier nuevo validador se puede agregar al sistema mediante una votación. Aunque el método representa un enfoque más centralizado y más apropiado para los órganos de gobierno o reguladores, actualmente también es popular entre las empresas de servicios públicos en el sector energético. El algoritmo de consenso puede ser útil en casos de uso especial donde la seguridad y la integridad no pueden ponerse en riesgo.

2.2.4. Ciclo de transacciones

Con todo lo anterior, podemos definir como funciona una blockchain y cual es el ciclo que cumplen las transacciones desde que son creadas por el emisor hasta que son almacenadas en la blockchain por ende recibidas por el emisor. Primeramente notamos que en lugar de definir la propiedad de todos los participantes de la red, en una blockchain se mantiene una lista de todas las transacciones en el libro contable de forma continua [11]. Cada transferencia de

propiedad se describe mediante datos de transacciones que señalan claramente qué propietario deja de ser propietario de qué artículo y a quién y en qué momento. La Figura 2.8 muestra el ciclo que cumple cada transacción en una blockchain.

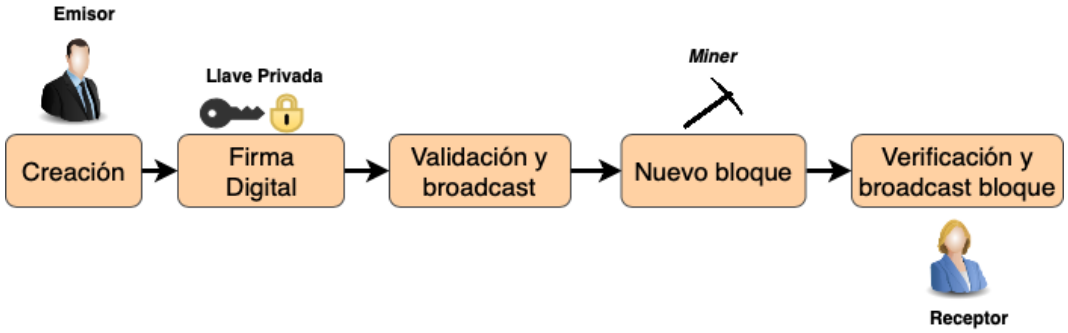


Figura 2.8: Ciclo de una transacción.

Tal como se indica en la figura, el procedimiento que rige cómo los nodos tratan con los nuevos datos de transacción y los bloques que reciben de sus pares consta de las siguientes etapas:

- El emisor (propietario de los activos de origen) crea una nueva transacción y la firma. Si se formó y firmó correctamente, la transacción firmada ahora es válida y contiene toda la información necesaria para ejecutar la transferencia de fondos [12]. Dado que la transacción está firmada y no contiene información confidencial, claves privadas o credenciales, se puede transmitir públicamente utilizando cualquier transporte de red subyacente que sea conveniente.
- Una vez que la transacción es recibida por algún nodo conectado a la red, la transacción será validada por ese nodo. Si es válida, ese nodo lo propagará a los otros nodos a los que está conectado y un mensaje de éxito se devolverá de forma sincronizada al originador. Si la transacción no es válida, el nodo la rechazará y devolverá un mensaje de rechazo al originador.
- La transacción válida debe llega a un minero de la red para su inclusión en el libro contable (la blockchain). Tan pronto como el *miner* termina de crear el nuevo bloque (PoW, PoS u otro), envía el bloque recién creado a todos los otros nodos [11].
- Cada nodo verifica los bloques nuevos y todos sus datos de transacción que contienen corrección formal, corrección semántica y autorización. Luego Cada nodo agrega bloques válidos a su propia copia de la blockchain. Si un bloque recién llegado se ha identificado como no válido, se descartará y los nodos continuarán procesando los datos de la transacción o creando un nuevo bloque.
- El nodo cuyo bloque fue aceptado recibirá los honorarios por todas las transacciones contenidas en el bloque como recompensa.

De esta manera, una blockchain logra almacenar de manera segura e inmutable los datos de las transacciones de los usuarios.

2.3. Contexto internacional

2.3.1. Revolución blockchain

Las cadenas de bloques o los libros de contabilidad distribuidos son una tecnología emergente que ha despertado un interés considerable entre las empresas de suministro de energía, las *startups*, los desarrolladores de tecnología, las instituciones financieras, los gobiernos nacionales y la comunidad académica [23]. Se espera revolucionar no solo la estructura técnica de nuestra comunicación y la tecnología de la información, sino también la estructura misma de las sociedades [24]. Algunos de los cambios que se esperan incluyen:

- La forma en que se llevan a cabo los negocios: Hasta ahora, las sociedades necesitaban intermediarios confiables para mediar en la mayoría de los tipos de transacciones comerciales (e.g bancos). Ya que blockchain proporciona confianza criptográfica a través de su diseño, las partes anónimas pueden realizar transacciones sin la posibilidad de hacer trampa, por lo que los intermediarios ya no serían necesarios más allá de la provisión de la plataforma técnica [24].
- La forma en que se regulan los negocios: Hasta ahora, las sociedades han requerido que los reguladores aseguren que las empresas operen dentro de los marcos legales (e.g auditores financieros). Como blockchain, junto con sus *smart contracts*, proporciona transparencia en el registro de transacciones, así como la imposición de reglas definidas dentro de los contratos en todas las transacciones, las verificaciones de cumplimiento normativo y legal se convierten en un requisito previo para la finalización de cualquier transacción [24].
- El rol de los individuos dentro de la sociedad: Hoy nos vemos como "sociedades de consumo", donde los individuos son generalmente consumidores pasivos. Sin embargo, la naturaleza transactiva entre pares de blockchain anima a los individuos a desempeñar roles productivos y de consumo. Los individuos ya no son consumidores pasivos, sino que son prosumidores activos (es decir, productores y consumidores). Ejemplos de esto son el intercambio entre pares y la microgeneración en el sector energético [25].

Se pueden distinguir tres etapas de desarrollo de la aplicación blockchain. Blockchain 1.0 corresponde a las criptomonedas (e.g, Bitcoin) que se utilizan como alternativa a las monedas reales. Blockchain 2.0 permite *smart contracts*, es decir, protocolos digitales que ejecutan automáticamente procesos de transacción predefinidos sin intermediarios (e.g, suministro de energía) de manera autónoma y segura. La próxima generación de Blockchain 3.0 alcanza un mayor grado de autonomía con una organización autónoma descentralizada basada en contratos inteligentes con registro de transacciones y reglas de programa mantenidas en la cadena de bloques [26].

Si bien las opiniones sobre el futuro a largo plazo de las criptomonedas pueden dividirse, varias aplicaciones clave han sido identificadas por numerosas fuentes. Un informe del gobierno del Reino Unido [27] indica que las blockchain podrían tener la capacidad de “reformar nuestros mercados financieros, cadenas de suministro, servicios de consumidores y de empresa a empresa, y registros públicos”. Las aplicaciones potenciales van desde los regis-

tros de activos y la transferencia de propiedad de activos duros hasta asegurar el registro de activos intangibles [28].

La variedad de aplicaciones de blockchain propuestas es tal que esta tecnología ha llegado a ser comparada con la llegada de Internet y algunos autores afirman que podría llegar a ser un avance tecnológico, generando una optimización significativa de procesos y modelos de negocios novedosos [29]. El potencial radica en el hecho de que las tecnologías de blockchain o de libro contable distribuido (DLT, por sus siglas en inglés) pueden redefinir la confianza digital y pueden eliminar a los intermediarios que forman un nuevo paradigma de gestión que potencialmente puede alterar las formas tradicionales de gobierno [30]. Su naturaleza disruptiva reside en el potencial de reemplazar el control de arriba hacia abajo por consenso y también en la filosofía subyacente de consenso distribuido, fuente abierta, transparencia y la toma de decisiones basada en la comunidad [27]. Según un informe reciente de Gartner [31], las tecnologías de cadena de bloques ya han superado el *peak* de expectativas en el ciclo de *hype* y se prevé que estén a dos o cinco años de la adopción general. Las etapas mencionadas se muestran en la Figura 2.9.

Figura 2.9: Curva de *hype* de Gartner.

2.3.2. Aplicaciones de blockchain en el sector energía

Los sistemas de energía están experimentando un cambio provocado por el avance de los recursos energéticos distribuidos y las tecnologías de la información y la comunicación (TICS). Uno de los principales desafíos es la descentralización y digitalización emergentes del sistema energético, que requiere la consideración, exploración y adopción de paradigmas novedosos y tecnologías distribuidas [23]. La tecnología Blockchain puede ser una de las soluciones potenciales para los futuros desafíos de estas nuevas tendencias, denominadas *Energy Internet*, o Internet de la Energía [32]. La idea de utilizar blockchain en el sector

energético está ganando un interés cada vez mayor [24]. Los principales usos de blockchain en el sector energía son mostrados en la Figura 2.10, mientras que las plataformas usadas se muestran en la Figura 2.11. Algunos ejemplos de aplicaciones existentes se muestran en la Tabla 2.2.

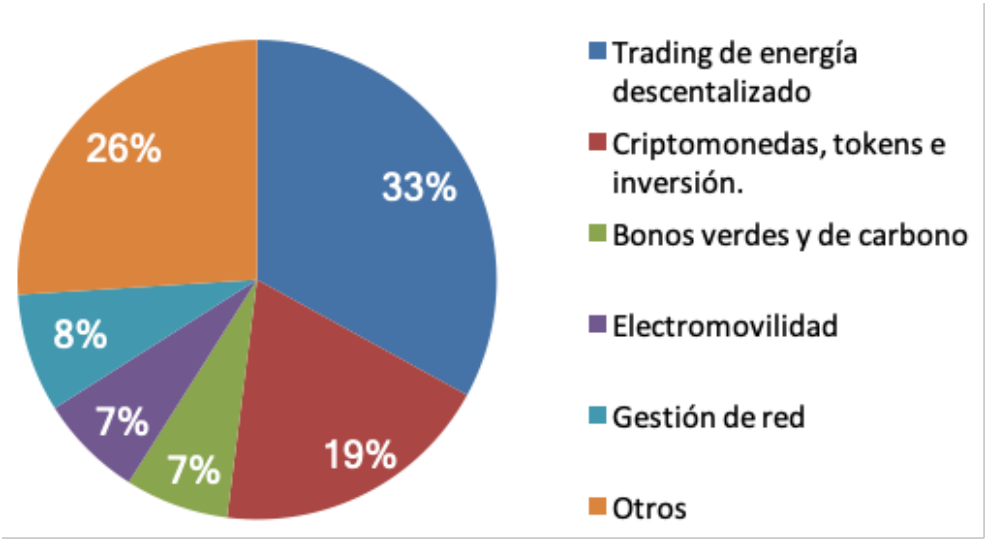


Figura 2.10: Aplicaciones de blockchain en el sector energético [23]. Clasificación de los casos de uso de blockchain según su campo de actividad: resultados obtenidos de un estudio sobre 140 iniciativas de blockchain en el sector energético que están siendo desarrolladas por empresas, *startups* e instituciones de investigación en países como Alemania, Estados Unidos, Reino Unido y Suiza, entre otros.

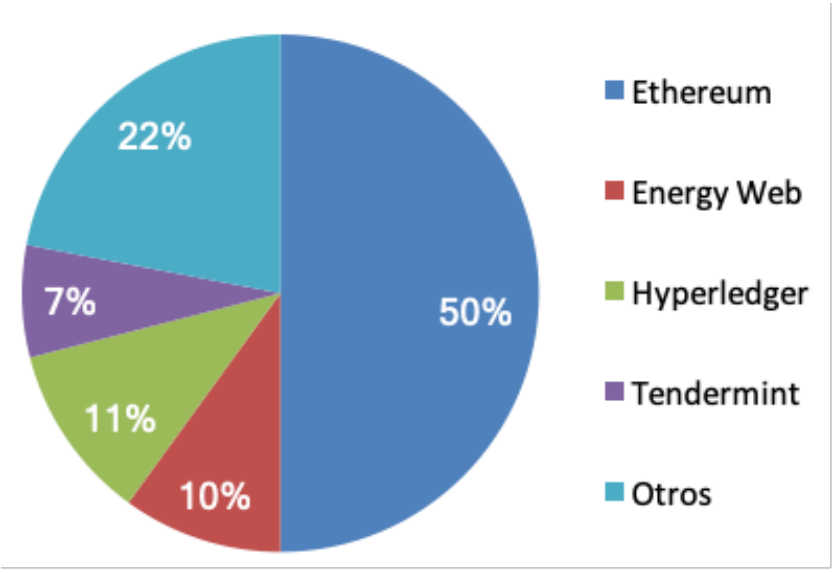
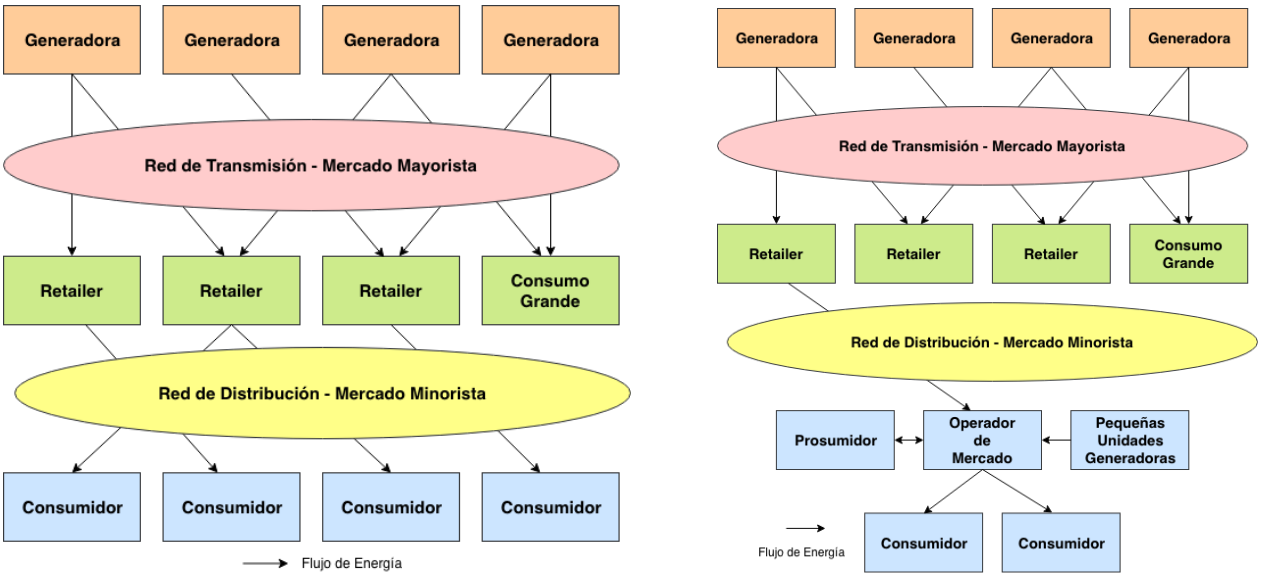


Figura 2.11: Plataformas blockchain usadas en el sector energía.[23]

Trading de energía descentralizado

El comercio tradicional de energía se gestiona a través de una organización centralizada. Con el acceso de un gran número de consumidores, el Internet de la energía se ha vuelto más complicado. Si se establece una organización centralizada, existen problemas como los altos costos operativos y la poca seguridad de la información. Si no hay una organización de gestión centralizada, habrá un problema de desconfianza en la entidad comercial. La introducción de la tecnología blockchain en el comercio de energía puede superar estos problemas. El modelo de comercio de energía P2P basado en blockchain puede proporcionar una plataforma de comercio eficiente, barata, abierta y confiable para el Internet de la Energía [33].

El comercio entre pares puede verse como una forma verdaderamente descentralizada de un mercado energético. Este es un dominio de aplicación donde los sistemas habilitados con blockchain se adaptarían más naturalmente, al permitir el comercio directo de energía entre los consumidores de energía (productores de energía / prosumidores y consumidores finales), quienes pueden usar este enfoque para tomar el control de su generación y demanda. Si bien esto generalmente se puede lograr en pequeñas comunidades y micro-redes, una pregunta clave es cómo encaja esto con el control y la operación de la red de distribución existente. Este modelo de mercado se muestra en la Figura 2.12 junto con el modelo tradicional de mercado.



(a) Mercado tradicional.

(b) Mercado local descentralizado.

Figura 2.12: Modelos de mercado centralizados y descentralizado. Se aprecia que en el modelo descentralizado aparece la figura del prosumidor. Un prosumidor se define como aquel agente que consume y produce electricidad [34].

Con los sistemas de comercio P2P, las unidades de electricidad generadas se registran dentro de una blockchain, lo que permite al propietario de esta generación comercializarla a otros. Esto permite que los generadores de energía y los compradores (grandes y pequeños) tomen posesión de sus productos, opciones y preferencias, en lugar de confiar únicamente en

la red como intermediario [35]. Algunos investigadores sobre el comercio de energía P2P se centraron en la creación del mercado de energía P2P, demostrando que el comercio de energía sin intermediarios basado en blockchain es posible y beneficioso para los generadores y compradores por igual [35] [36] [37]. Otros estudiaron la optimización de los recursos energéticos en el comercio P2P [38]. En [39], los autores han desarrollado un mecanismo de mercado de subasta cerrado basado en un contrato inteligente en el que los individuos pueden presentar ofertas y pedidos de venta para cada período de mercado y un contrato automatizado decidiría un precio de compensación de mercado para el período.

Electromovilidad

En el área de la defensa de los viajes ecológicos y la conservación de la energía, el uso generalizado de vehículos eléctricos será una de las soluciones efectivas para los problemas ambientales. Sin embargo, los usuarios de automóviles eléctricos se enfrentan a la dificultad de la carga [33]. Algunos investigadores proponen integrar blockchain con vehículos eléctricos (VES) [40] para que los VES puedan usar blockchain para encontrar estaciones de carga cercanas, mientras que las estaciones de carga pueden ofrecer la oportunidad de cargar VES. Este mecanismo ayudaría a encontrar el mejor precio y ubicación para los usuarios de VES y las estaciones de carga, al mismo tiempo que proporciona privacidad y seguridad a los VES [24].

La naturaleza descentralizada del transporte, con muchas partes (vehículos, conductores, estaciones de carga, pasajeros que utilizan servicios de movilidad *on demand*, como Uber of Lyft) se presta naturalmente a las implementaciones de blockchain [23]. Las ventajas de la descentralización en este caso incluyen: la eliminación de la necesidad de una infraestructura de carga de VES gestionada centralmente, la tolerancia a fallos, así como la eliminación de la fijación de precios y la colusión entre estaciones de carga o proveedores de transporte. Sin embargo, también en esta aplicación, las cadenas de bloques tendrían que superar serios problemas de privacidad y seguridad.

Las oportunidades para la innovación de blockchain en aplicaciones de movilidad eléctrica son importantes, sin embargo, es necesario abordar ciertos desafíos. Las cadenas de bloques son, por su naturaleza, libros de contabilidad públicos, por lo que la información sobre la ubicación diaria y el movimiento de los usuarios de VES debería ser anonimizada, para proteger su privacidad. Además, las cadenas de bloques en los sistemas de movilidad eléctrica deberían ser a prueba de falsificaciones, para evitar que los actores maliciosos pongan en peligro la seguridad de los vehículos eléctricos. Finalmente, dado que los VES pueden interactuar con el sistema de energía y cargar en varias ubicaciones, el desarrollo de estándares de interoperabilidad es crucial para lograr los beneficios que las cadenas de bloques pueden ofrecer en este espacio [32].

Emisión de certificados

En la actualidad, el mercado de emisiones de carbono está enfrentando problemas como la gran carga de trabajo de las certificaciones de cuotas de emisión y la dificultad para rastrear

los datos de transacciones. La tecnología Blockchain puede proporcionar una plataforma de gestión inteligente para la certificación y el comercio de los derechos de emisión de carbono, como se muestra en la Figura 11. A través de esta tecnología, se puede rastrear cada tonelada de carbono y toda la información de la transacción, evitando la manipulación y la asimetría de la información. Las Reducciones de Emisiones Certificadas de China (CCER, por sus siglas en inglés) se comercializarán en forma de activos digitales "boletos de carbono". Cada boleto de carbono tiene una identificación única, marcada y registrada en blockchain. El comercio de carbono se realizará automáticamente a través de contratos inteligentes [33].

Compañía	Aplicación Blockchain	Plataforma
Energy-Blockchain Lab & IBM [41]	Plataforma eficiente y transparente que permite a las organizaciones con altas emisiones monitorear sus huellas de carbono y cumplir con las cuotas al comprar créditos de carbono de emisores bajos.	Hyperledger Fabric
Grid+ [42]	Plataforma comercial P2P entre sus clientes	Ethereum
ImpactPPA [43]	Criptomonedas, fichas e inversiones.	Ethereum
LO3 Energy (Exergy) [44]	Plataforma de comercio de energía P2P.	Tendermint
MyBit [45]	Inversión P2P en hardware IoT, como paneles solares conectados. Un inversionista puede ser propietario de una parte del hardware con un *token* y obtener un retorno por parte de la propiedad.	Ethereum
Power Ledger [46]	Comercio de energía P2P, carga de EV, monitoreo de la red de transmisión, financiamiento de activos P2P e intercambio de *token* de propiedad.	Ethereum
WePower [47]	Una plataforma para el comercio P2P de energía renovable, así como la recaudación de fondos para proyectos renovables mediante la preventa de energía que se generará en el futuro.	Ethereum

Tabla 2.2: Empresas de blockchain que trabajan actualmente en el sector energético en el contexto internacional.

Como muestra la Tabla 2.2, existe una amplitud real de las áreas y propósitos dentro del sector de la energía donde se está empleando activamente blockchain. Varias observaciones emergen de la revisión de los negocios actuales basados en blockchain. Algunas de ellas se listan a continuación:

- Blockchain se utiliza para iniciar una nueva estructura empresarial y un ecosistema dentro del sector energético, compitiendo contra los operadores tradicionales. Por lo tanto, algunas empresas, como myBit [45], fomentan la inversión en hardware de generación renovable de tal manera que, incluso si una persona no puede comprar un dispositivo

completo, puede invertir en una parte de un hardware mediante *tokens* y obtener un retorno según su parte. Una vez generada, la producción de energía se registra en una cadena de bloques, sobre la cual el software (que actúa como comerciante mayorista o minorista) compra y vende energía.

- Debido a este nuevo ecosistema emergente, también comienzan a surgir nuevos tipos de servicios de energía (por ejemplo, equilibrando la oferta y la demanda dentro de la red o programando de forma remota el funcionamiento de dispositivos del consumidor, como la carga de vehículos eléctricos, el funcionamiento de las lavadoras, etc. En respuesta a la disponibilidad de energía).
- Las formas renovables de energía son el núcleo de los productos entregados a través de una infraestructura de blockchain. Algunas empresas fomentan el comercio de la energía renovable generada, mientras que otras apoyan la adopción y una mejor utilización de los activos de generación a nivel familiar;
- Muchas de estas empresas están enfocadas en el usuario final, con el objetivo de obtener mejores precios de energía para los usuarios finales, o acceso a / participación en el esfuerzo de generación de energía;
- Finalmente, las empresas de energía tradicionales también han identificado blockchain como una potente tecnología y han comenzado a utilizarla para la optimización de procesos empresariales y la comunicación entre empresas.

2.4. Contexto nacional

2.4.1. Normativa

En Chile, el sector eléctrico está normado por la ley 20.018 de servicios eléctricos. En ella, se dispone la normativa sobre generación, transmisión y distribución de electricidad. Ante las posibles aplicaciones de blockchain en el sector, toma especial importancia lo que diga la ley en aspectos en el aspecto de distribución y generación distribuida. A continuación se presentan algunos de los aspectos más importantes presentes en la ley con respecto a posibles usos de blockchain en el sector.

Regulación

El suministro de electricidad por parte de las distribuidoras es considerado un monopolio natural, pues en un esquema de mercado centralizado como el de la Figura 2.12a, los consumidores no pueden escoger a quien comprar la energía energía eléctrica. Es por esto que los mercados eléctricos son regulados por el estado.

Clientes regulados

Para suministros a usuarios finales cuya potencia conectada es inferior o igual a 2.000 kW, son considerados sectores donde las características del mercado son de monopolio natural y por lo tanto, la Ley establece que están afectos a regulación de precios.

Clientes libres

Para suministros a usuarios finales cuya potencia conectada superior a 2.000 kW, la Ley dispone la libertad de precios, suponiéndoles capacidad negociadora y la posibilidad de proveerse de electricidad de otras formas, tales como la autogeneración o el suministro directo desde empresas generadoras.

Zonas de concesión

El estado otorga concesiones a las compañías distribuidoras, las cuales les otorgan el derecho a usar bienes nacionales uso público para tender líneas aéreas y subterráneas destinadas a la distribución en la zona de concesión. La distribución de electricidad a usuarios ubicados en una zona de concesión sólo puede ser efectuada mediante concesión de servicio público de distribución, salvo por excepciones como los clientes libres.

Cooperativas eléctricas

Las cooperativas son asociaciones que de conformidad con el principio de la ayuda mutua tienen por objeto mejorar las condiciones de vida de sus socios. En el caso de las cooperativas eléctricas, esta asociación se constituye con el objeto de distribuir energía eléctrica entre sus socios. En una cooperativa, las ganancias se suelen reinvertir en infraestructura, pues se intenta proporcionar el mejor servicio al menor costo posible.

Además del concepto de cooperativa eléctrica descrito anteriormente, existe un modelo de cooperativa virtual en que un grupo de personas, bajo el mismo principio de ayuda mutua comercializan su energía en su condición de clientes (e.g vecinos dentro del mismo edificio). Las condiciones de estos acuerdos no se encuentran predefinidas en la ley y deben ser acordadas por los miembros de la asociación.

Netbilling

La Generación Ciudadana, establecida mediante la Ley 20.571, es un sistema que permite la autogeneración de energía en base a Energías Renovables No Convencionales (ERNC) y cogeneración eficiente. Esta Ley, conocida también como Netbilling, Netmetering o Generación Distribuida, entrega el derecho a los usuarios a vender sus excedentes directamente a la

distribuidora eléctrica a un precio regulado, el cual está publicado en el sitio web de cada empresa distribuidora.

2.4.2. Aplicaciones de blockchain existentes

En Chile, a pesar de que blockchain cuenta con cierta popularidad, debido principalmente al mundo de las criptomonedas, la investigación de posibles usos de esta tecnología para el sector energía es muy escasa, al igual que las aplicaciones existentes. A la fecha, existen en Chile únicamente dos aplicaciones relativamente consolidadas de blockchain en el sector energía, las cuales siguen la linea de los trabajos existentes en el contexto internacional. Ambas aplicaciones son tratadas a continuación.

Energía Abierta [48]

Energía Abierta es una iniciativa desarrollada por la Comisión Nacional de Energía (CNE), la cual consiste en un portal web multifuncional desarrollado para atender una amplia variedad de intereses y necesidades asociados al sector energético, enfocándose en reducir asimetrías de información, aumentar la transparencia y fomentar la participación ciudadana, mediante soluciones innovadoras. Energía Abierta utiliza blockchain para certificar datos de:

- Electricidad,
- Hidrocarburos,
- Energías renovables,
- Eficiencia energética, entre otros.

En esta plataforma, los datos a certificar son almacenado en la nube, para posteriormente utilizar funciones *hash* para crear firmas digitales de estos datos y almacenarlos en la blockchain en Ethereum, de manera segura, integra, trazable y transparente. Una vez que los datos forman parte de la cadena de bloques, cualquier persona puede acceder a los datos, al igual que puede acceder a datos de transacciones de *ether* (la criptomoneda de Ethereum).

Sellosol [49]

Sello Sol es un certificado, creado por la empresa Phineal, el cual se otorga a empresas que poseen sistemas fotovoltaicos y solar térmicos que permite a los clientes conocer la cantidad de energía solar generada por sus instalaciones. El Sello SOL permite diferenciar a las empresas que realizaron la inversión para cambiarse a las energías limpias y renovables, pudiendo capturar el valor del "marketing verde" a través de la visualización de la generación real de energía de su instalación solar.

En Sello Sol, la tecnología blockchain permite realizar transacciones de la información de la energía en forma segura, transparente y verificable a través de una plataforma en línea. Cada nuevo bloque de información es validado automáticamente por la comunidad conectada

a la red de Sello Sol, transformando esta información en un dato incorruptible y único en el tiempo, con el cual los usuarios finales pueden verificar a través de la misma plataforma el origen de la energía solar.

Sello Sol utiliza el protocolo de blockchain propio denominado GTIME, el cual ha sido desarrollado específicamente para la medición y trazabilidad de fuentes energéticas. GTIME incorpora en bloques de datos del blockchain las mediciones de energía cada 15 minutos, variables de geolocalización, tiempo y una huella de identificación de cada medición generada en las instalaciones asociadas, para la validación descentralizada de trazabilidad energética.

Capítulo 3

Propuesta de ĐApp

En este capítulo se describe la propuesta de aplicación blockchain elaborada para el sector energético chileno. La propuesta consiste en un *marketplace* administrado por *smart contracts*, en donde distintos agentes puedan asociarse para la realización de proyectos de generación distribuida (GD), de forma similar a como lo harían los socios de una cooperativa. Esta aplicación busca cambiar la forma en que se realizan los proyectos GD, pasando de un único agente multiespecialista involucrado a multiples agentes monoespecialistas, reduciendo asi los *soft costs* de los proyectos.

3.1. Consideraciones preliminares

Debido a que el número de combinaciones posibles para un proyecto GD puede ser muy grande (miles de posibilidades) dependiendo del número de variables consideradas (e.g tipo de energía, capacidad, ubicación, etc), el desarrollo de esta ĐApp contempla únicamente proyectos estandarizados en términos del tipo de energía a generar y su capacidad (e.g 1 kWp solar). Asimismo, se considera que las ofertas realizadas por los proveedores incluyen todo el *hardware* necesario para la realización del proyecto (e.g panel solar + inversor). De esta forma, para cada proyecto estandarizado definido dentro de la aplicación, se debe crear un conjunto de *smart contracts* que administre a los agentes involucrados en dicho proyecto.

3.2. Plataforma Ethereum

Como se trató anteriormente, Ethereum es la plataforma más utilizada para aplicaciones en el sector energía. Esta tendencia se aprecia no solamente en este sector, sino que, en general, Ethereum cuenta con un mayor ecosistema de desarrolladores e investigadores, así como una mejor documentación. Es por esto que para el desarrollo de esta ĐApp se utilizará esta plataforma. Al utilizar la red de Ethereum existen algunos conceptos que se deben tener en cuenta, los cuales se tratan a continuación.

3.2.1. E.V.M: Ethereum Virtual Machine

La EVM es una única "computadora" global de 256 bits compuesta por todas las computadoras de la red Ethereum [13]. Desde la perspectiva de un desarrollador de software, la EVM también es un entorno de ejecución para pequeños programas, llamados *smart contracts*, que pueden ser ejecutados por la red. Estos programas son desarrollados en un lenguaje llamado Solidity. Cuando un usuario envía una transacción, la EVM requiere un pequeño *fee* para procesar la transacción. Al obligar a los usuarios a pagar por las transacciones en la EVM, la probabilidad de que se ejecuten programas inútiles sin fin se reduce teóricamente. Estos costos se cotizan en una unidad llamada *gas*, el cual tiene un costo en *ether*. El costo en *gas* de las operaciones realizadas por la EVM dependerá de la complejidad de estas. Los costos en *gas* siempre se deben tener en cuenta, pues constituyen un "costo de operación" de las ĐApps de Ethereum.

3.2.2. *Main net* y *test nets*

Para el despliegue de los *smart contracts* existen alternativas a la red principal de Ethereum (*main net*), llamadas *test nets* (e.g Rinkeby). Estas cadenas se diferencian de la cadena principal en que sirven como entorno de pruebas de ĐApps con *ether* ficticio (sin valor), por lo que la ejecución de *smart contracts* no tiene ningún costo para el desarrollador. Otra alternativa, es que un computador simule su propia blockchain privada de Ethereum para realizar pruebas de manera centralizada. En este trabajo se realizan pruebas preliminares de funcionamiento en una red local y se despliegan los *smart contracts* en la *testnet* de Rinkeby de manera que los códigos desarrollados sean auditables desde cualquier computador conectado a Internet. Las ventajas de Rinkeby por sobre otras *test nets* son el uso de PoA como mecanismo anti-spam y la facilidad para obtener *ether* de prueba desde un *faucet* [50].

3.2.3. *Tokens* ERC20

Ya que la ĐApp a desarrollar involucrará pagos en algún momento, es necesaria la utilización de *tokens* para que se puedan llevar a cabo. Si bien Ethereum cuenta con su propia criptomoneda (ether), la controversia existente en base a estas ultimas, así como su valor, el cual en general es volátil, sumado a la dificultad de su uso desde el punto de vista del usuario, hacen que su uso sea algo impráctico. Es por esto que el presente trabajo crea un *token* propio ateniéndose al estándar ERC20. El estándar de *token* ERC20 describe las funciones y eventos que debe implementar un contrato de *token* de Ethereum. Las funciones definidas por este estándar se listan a continuación:

- Saldo *tokens*: Permite obtener el saldo en *tokens* de una cuenta de la red.
- Transferir *tokens*: Permite a una cuenta transferir parte de su saldo a otra.
- Autorizar: Permite a una cuenta autorizar a una dirección para que gaste una cantidad de *tokens*. Esta dirección puede corresponder a otra cuenta o a un *smart contract*.

- Transferir desde: Hace uso de la autorización de una cuenta para transferir su saldo a un destinatario.

Cabe destacar que por motivos de simplicidad, la cantidad total de *tokens* será fija en todo momento. De esta forma, definiendo la cantidad inicial, e implementando las funciones descritas anteriormente, se dispondrá de un medio de pago para la plataforma. Adicionalmente, debido a que algunos pagos se efectúan de forma posterior a la obtención del bien o servicio, se agregan las siguientes funciones al contrato de *token ERC20*:

- Deuda: Permite obtener la deuda de un usuario con otro.
- Deuda total: Permite obtener la deuda total de un usuario.
- Agregar deuda: Permite aumentar la deuda de *tokens* de una cuenta a otra.
- Pagar: Permite a un usuario pagar la deuda, o parte de la deuda con su acreedor.

Finalmente, por simplicidad, se fijará el valor de estos *tokens* como 1 *token* = 1 CLP. De esta manera, las cantidades tranzadas por los clientes serán más intuitivas y menos manipulables por parte de los agentes. Con esto, la interfaz del contrato de *tokens* se vería como muestra la Figura 3.1

Figura 3.1: Ejemplo de interfaz del contrato de *tokens*.

3.3. Economía colaborativa

Parte del alto valor de los proyectos de generación distribuida se debe a la alta especialización de los agentes (*soft costs*), quienes deben realizar, entre otras funciones, los roles de proveedores, instaladores y ser quienes obtienen los permisos correspondientes a los proyectos. A pesar de que estas funciones pueden ser realizadas por distintos agentes independientes, en la actualidad no existe un lugar de encuentro (físico o virtual) entre estos agentes a donde pueda acudir quien desee realizar un proyecto. Ante esta problemática se buscará, mediante blockchain, propiciar un lugar de encuentro entre los distintos agentes, formando un ecosistema transparente y competitivo. Este ecosistema se muestra en la Figura 3.2.

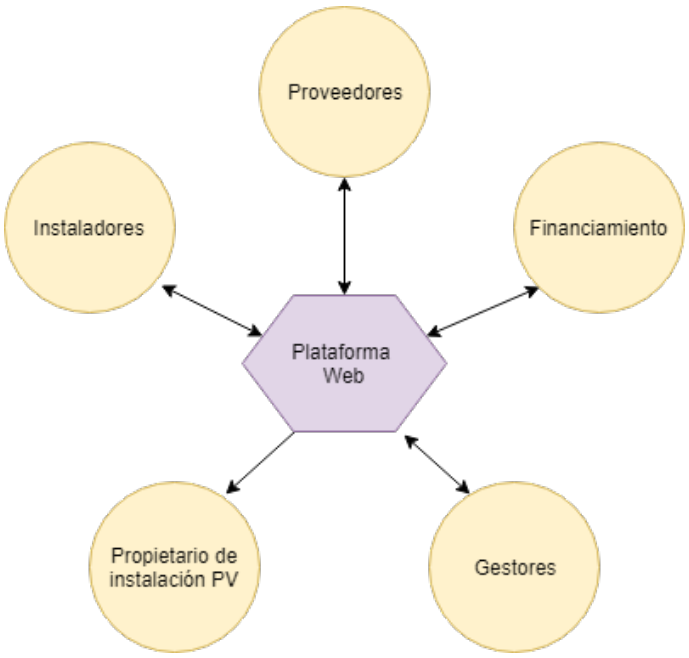


Figura 3.2: Ecosistema para proyectos GD.

En este ecosistema, cada uno de los agentes aporta *inputs* y obtiene *outputs* en función de estos. A continuación se describen los roles de los agentes en el marco de la plataforma a implementar:

- Propietarios: Son quienes deciden realizar un proyecto de generación distribuida. En base a algún precio de referencia determinan cual es el precio máximo que están dispuestos a pagar por dicho proyecto y si necesitan financiamiento para este.
- Proveedores o *vendors*: Son quienes ofertan los materiales para el proyecto (e.g paneles policristalinos, inversores, etc.).
- Instaladores: Son quienes llevan a cabo las instalaciones de los equipos proveídos por los *vendors*. En la plataforma deben ofertar cuanto están dispuestos a cobrar por su trabajo.
- Gestores: Son quienes realizarán todos los trámites necesarios para el proyecto, como la obtención de permisos. En la plataforma deben ofertar cuanto están dispuestos a cobrar por su trabajo.

- Financiadores: Son las entidades que están dispuestas a pagar los proyectos de los propietarios obteniendo beneficios en el mediano/largo plazo. Hacen aportes a proyectos que se encuentren en la etapa de *crowdfunding*.

De esta manera la plataforma cuenta con la información necesaria para realizar *matching* entre los agentes. Propietarios, proveedores, instaladores y gestores son administrados, cada cual mediante un *smart contract* similar, que implementa las funciones CRUD (*Create, Read, Update, Delete*) en una base de datos, junto con implementar un sistema de reputación para cada agente. Las funciones más importantes de estos *smart contracts* son listadas a continuación:

- Nueva oferta/solicitud: "Registra" a los agentes como tales, junto con su oferta/solicitud correspondiente.
- Obtener oferta/solicitud: Permite obtener el valor de las ofertas/solicitudes de los agentes.
- Actualizar oferta/solicitud: Permite a los agentes modificar los valores de sus ofertas/-solicitudes.
- Eliminar oferta/solicitud: Permite a los usuarios eliminar sus ofertas/solicitudes.

La información proporcionada por los agentes es almacenada en la blockchain de manera que sea auditable y transparente para los participantes de la plataforma. La Figura 3.3 muestra, a modo de ejemplo, una interfaz con un contrato de solicitudes (propietarios).

Figura 3.3: Interfaz contrato de solicitudes (propietarios).

Una vez disponible la información de oferta y demanda en la blockchain, periódicamente y de manera *off-chain* se ejecuta la función *matching*, descrita a continuación:

Algoritmo 1 *Matching* (*off-chain*)

```
1: for propietario in propietarios:
2:     maxPrice = demandContract.getDemand(propietario)
3:     for proveedor in proveedores:
4:         providerPrice = providerContract.getPrice(proveedor)
5:         for instalador in instaladores:
6:             installerPrice = installerContract.getPrice(instalador)
7:             for gestor in gestores:
8:                 managerPrice = managerContract.getPrice(gestor)
9:                 totalPrice = providerPrice + installerPrice + managerPrice + fee
10:                if totalPrice <= maxPrice:
11:                    projectContract.matching(propietario, proveedor, instalador, gestor)
12:                    return
```

Como se puede apreciar, al ser un algoritmo *off-chain*, este debe interactuar con los contratos desplegados en la blockchain. Cuando se cumplen las condiciones necesarias, la función *matching off-chain* interactua con el contrato de proyectos llamando a la función *matching on-chain* . Esto se hace para reducir los costos en *gas* en la creación de proyectos, pues el costo en *gas* de efectuar este algoritmo de manera centralizada es 0. A grandes rasgos, el algoritmo *matching on-chain* realiza las siguientes operaciones:

1. Verificar las condiciones necesarias para el *matching*. Esto debe hacerse, pues los inputs corresponden a información *off-chain*.
2. Se agrega la deuda del proyecto al propietario y se habilita a los agentes para calificarse entre ellos.
3. En caso de necesitar financiamiento, el proyecto pasa a la etapa de *Crowdfunding*.
4. En caso contrario, se agrega el nuevo proyecto a la blockchain con la información provista por cada uno de los agentes.

3.4. Financiamiento

Como se mencionó anteriormente, los propietarios tienen la opción de elegir si necesitan o no financiamiento para su proyecto. Para que los proyectos obtengan el financiamiento necesario se crea un *smart contract* de *crowdfunding* con plazo fijo para un tipo de proyecto, en donde cualquier usuario de la red puede usar sus *tokens* para financiar el proyecto. Para favorecer la realización de la mayor cantidad de proyectos posible, solamente podrá haber un proyecto en *crowdfunding* a la vez. Si se alcanza la meta del *crowdfunding*, los *tokens* reunidos son transferido al propietario del proyecto. En el caso de que la meta no se alcance, los *tokens* son devueltos a los financiadores.

3.5. Sistema de pagos

Una vez realizado el proyecto, se registra a los propietarios con su respectiva parte del proyecto. Si el propietario no optó por financiamiento, será el dueño del proyecto y la energía producida por este. En caso de necesitar financiamiento, el modelo de negocio cambia, pues el propietario deja de ser dueño del proyecto, el cual pasará a ser de los financiadores en forma proporcional a su aporte en *tokens*. Luego, serán los dueños del proyecto quienes vendan la energía producida por este al propietario (dueño del lugar físico donde se encuentre instalado el proyecto). Periódicamente, un *smart-meter* ubicado en el proyecto proporcionará a la blockchain cuánta energía ha generado durante dicho periodo. De acuerdo con dicha cantidad y con el valor de la energía, el cual es determinado previamente, se crea una deuda en *tokens* del usuario con los dueños del proyecto, la cual debe ser pagada cuando el usuario disponga los *tokens*, de manera similar a como se paga la energía eléctrica a una distribuidora.

3.6. Visión general

En resumen, la plataforma desarrollada busca reducir las barreras económicas de los proyectos de generación distribuida, a la vez que otorga transparencia y un ambiente competitivo a los procesos utilizando blockchain. Esto se logra mediante una base de datos, un sistema de *crowdfunding* y un sistema de pagos. El funcionamiento de la plataforma está sintetizado por la Figura 3.4.

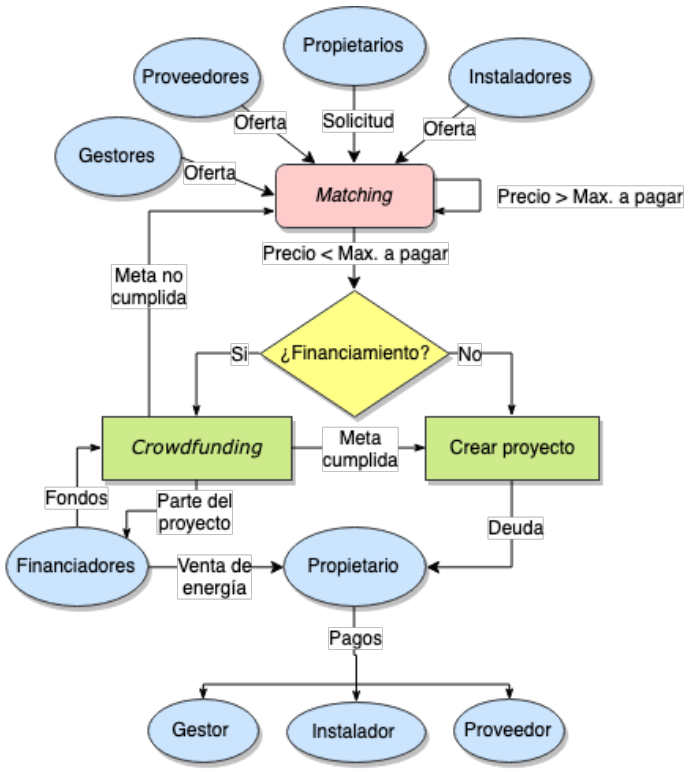
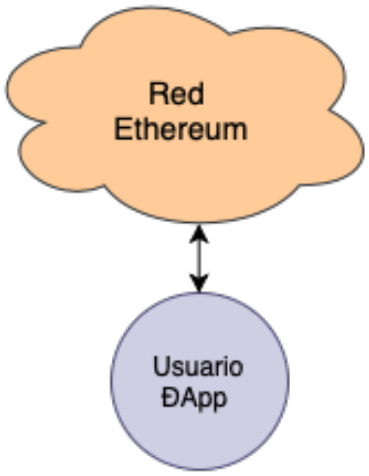
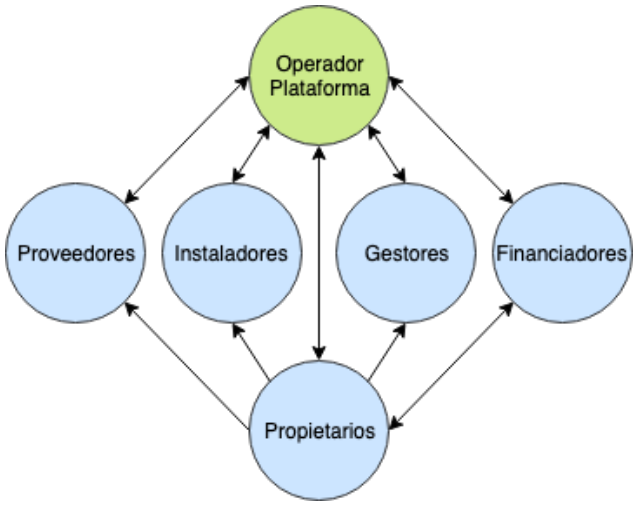


Figura 3.4: Diagrama de flujo aplicación blockchain.

Para el desarrollo de esta aplicación se utiliza la red de Ethereum, la cual introduce un costo por transacción cuantificado en *gas*, que debe ser pagado en *ether* por quien realiza la transacción. La conversión de *gas* a *ether* (*gas price*) la fija quien realiza la transacción, siendo los *miners* de la red quienes deciden que transacciones minar en base a este valor. Además, los pagos dentro de la plataforma se realizan con *tokens* con valor unitario fijo (1 CLP). Por lo tanto, el flujo de activos dentro de la aplicación se da según la Figura 3.5.



(a) Flujo de *ether*.



(b) Flujo de *tokens*.

Figura 3.5: Flujo de activos de usuarios de la ĐApp: Toda transacción en la red de Ethereum tiene un costo que se debe pagar a quien valida la transacción (*miner*), de manera que los usuarios de la ĐApp, incluyendo al operador de la plataforma, deberán pagar con *ether* cada acción que realicen (e.g crear oferta, pagar deuda). Cualquier usuario puede comprar/vender *ether* a otro usuario de la red Ethereum en cualquier momento. Por otro lado, el operador de la ĐApp cambia *tokens* ERC20 por CLP a los usuarios de la ĐApp, luego estos utilizan los *tokens* para realizar pagos dentro de la ĐApp o canjearlos de vuelta.

Capítulo 4

Resultados y análisis

4.1. Resultados

A continuación se presentan los resultados obtenidos del desarrollo de la ĐApp. Primeramente se muestran las direcciones de los contratos desplegados en la testchain de Rinkeby en la red Ethereum, los cuales son auditables en el sitio web Etherscan [51], con los costos en *gas* de su despliegue. Luego se muestra el funcionamiento de la plataforma mediante dos casos base de proyecto, incluyendo sus costos en *gas* y los *hashes* de las transacciones realizadas. Finalmente se muestra el funcionamiento simulado de la plataforma durante un periodo de tiempo y se realiza una análisis económico de esta en comparación con una aplicación tradicional.

4.1.1. Despliegue de *smart contracts*

Como se mencionó en el capitulo anterior, los *smart contracts* desarrollados fueron desplegados en la *testchain* de Rinkeby. La tabla 4.1 muestra las direcciones de los contratos utilizados y el costo en *gas* d e su despliegue en la blockchain. Tanto los contratos como las transferencias de los mismos pueden ser auditados en *etherscan* al buscarlos por su dirección.

Contrato	Direccion	*Gas*
Token ERC20	0x2c66b176962911D2ce40f5809cD52C73c8E78356	1.456.089
Propietarios	0xD5fbF619121824aCB4e7aAf66A1d86947CE87f1B	623.701
Proveedores	0x71eCEF369c041955C9993c635144a629c82CcD86	530.951
Instaladores	0xC0ed1D311963EFc99418C0C84Ff2661CF199f9b0	530.159
Gestores	0x23215E9FFaE52Eaf79c861d4F620A9748f3652C1	530.759
Crowdfunding	0x4Bac31B5056b1975D286d552F64F7962b8f2b2cc	1.654.850
Proyectos	0x05c24902c594c255E4d2FA9B5D8ca7c1d44A7E85	2.163.726
Total	-	7.490.235

Tabla 4.1: Despliegue de *smart contracts* en la blockchain de Ethereum

4.1.2. Casos base

Como primera prueba de la plataforma, se simulan dos casos simples de operación, los cuales suponen un mínimo número de operaciones y de costos en *gas*. Se considera un precio máximo a pagar de 2.190.000 *tokens* por parte del propietario y una combinación de ofertas tal que el valor total del proyecto, incluyendo un *fee* del 5 % por utilizar la plataforma, se encuentre por debajo de este valor. Los resultados de ambas simulaciones se muestran a continuación.

Caso base 1: Proyecto sin financiamiento

La simulación de este caso representa el costo en *gas* mínimo que puede tener un proyecto dentro de la plataforma, pues existe un único *match* posible entre los agentes, que además cumple la condición de precios para llevar a cabo el proyecto. Los agentes partícipes en este proyecto, así como las funciones que llama cada uno se muestran en la Tabla 4.2.

Agente	Dirección	Función	Valor [*tokens*]
Propietario	0x6C7...499	Nueva solicitud	2.000.000
Proveedor	0x520...d0BF	Nuevo proveedor	1.000,000
Instalador	0x6c2...6E8	Nuevo instalador	100.000
Gestor	0x29b...6C8	Nuevo gestor	10.000
Operador	0xCBE...E2F	*Matching*	55.500
Total	-	-	1.655.500

Tabla 4.2: Agentes caso base 1

Cuando cada agente realiza una transacción, obtiene el *hash* de esta como recibo, también debe pagar un *fee* por el costo en *gas* de la transacción. Las transacciones realizadas en el primer caso base son 5, siendo ejecutadas una por cada agente. Los detalles de estas transacciones, así como sus costos en *gas* se muestran en la tabla 4.3.

De	Para	Hash	Función	*Gas*
0x6C7...499	0xD5f...f1B	0x951...b47	Nueva solicitud	80.436
0x520...d0BF	0x71e...D86	0x46e...555	Nuevo proveedor	74.681
0x6c2...6E8	0xC0e...9b0	0xadb...257	Nuevo instalador	74.571
0x29b...6C8	0x232...2C1	0x064...aa9	Nuevo gestor	74.529
0xCBE...E2F	0x589...545	0xf5e...546	*Matching*	654.911
Total	-	-	-	959.128

Tabla 4.3: Transacciones caso base 1.

Una vez realizado el *match* entre los agentes, el proyecto se almacena en la blockchain y se le asigna una ID única. Al ingresar la ID del proyecto (ID = 0 en este ejemplo) al contrato de proyectos se obtendrá la siguiente información:

$$Propietario : 0x6C740120A054cecF209BCaCF62fC38Cc64D98499$$

$$Proveedor : 0x520597cF4C6f9aed321a39a12529911350c7d0BF$$

$$Instalador : 0x6c2a6C2F7B3ABf37575a423E72d91C78F72B36E8$$

$$Gestor : 0x29b93fdD9D9BE8c80d483aA94CA12496f09006C8$$

$$Valor : 1165500$$

Caso Base 2: Proyecto con financiamiento

Este caso es análogo al caso base 1 en cuanto al número de combinaciones posibles. Sin embargo, en este caso se incluye un agente financiador, quien financiará la totalidad del proyecto. Los agentes que participan en la realización, así como las funciones que llaman se muestran en la tabla 4.4

Agente	Dirección	Función	Valor [tokens]
Propietario	0x6C7...499	Nueva solicitud	2.000.000
Proveedor	0x520...d0BF	Nuevo proveedor	1.000,000
Instalador	0x6c2...6E8	Nuevo instalador	100.000
Gestor	0x29b...6C8	Nuevo gestor	10.000
Operador	0xCBE...E2F	*Matching* (*Crowdfunding*)	55.500
Financiador	0xF7a...358	Financiar	1.655.500
Beneficiario	0x6C7...499	Chequear + Retirar fondos	-

Tabla 4.4: Agentes caso base 2

A diferencia del caso base 1, este proyecto tiene interacción del propietario (o beneficiario) con el contrato de *crowdfunding*, pues el beneficiario debe chequear si se cumple la meta para posteriormente retirar los fondos. El historial de transacciones de la realización de este proyecto se muestran en la Tabla 4.5.

De	Para	Hash	Función	*Gas*
0x6C7...499	0xD5f...f1B	0x3d7...aaf5	Nueva solicitud	70.900
0x520...d0BF	0x71e...D86	0x204...372	Nuevo proveedor	65.081
0x6c2...6E8	0xC0e...9b0	0x156...96a	Nuevo instalador	64.971
0x29b...6C8	0x232...2C1	0xe81...b6f	Nuevo gestor	64.929
0xCBE...E2F	0x4Ba...2cc	0x40e...6f4	*Matching* (*Crowdfunding*)	61.768
0xF7a...358	0x2c6...356	0xfbe...9c1	Financiar	191.360
0x6C7...499	0x4Ba...2cc	0x821...edb	Chequear	32.093
0x6C7...499	0x4Ba...2cc	0x8a7...801	Retirar fondos	437.372
Total	-	-	-	988.474

Tabla 4.5: Transacciones caso base 2.

Como se puede apreciar, los costos en *gas* de las ofertas/solicitudes prácticamente no varían con respecto al caso base 1, mientras que el costo de la función *matching* es menor. Esto se debe a que cuando se conecta a los agentes, el proyecto aún no se puede crear, pues no se sabe si los fondos se obtendrán para su realización, por lo que en lugar de crear el proyecto se crea un *crowdfunding* con los datos de los agentes. Luego el proyecto se creará una vez que el *crowdfunding* es exitoso, cargando el costo en *gas* al beneficiario una vez que retira los fondos. Es por esto que el costo de la función *withdraw* es comparable con el costo de la función *matching* del caso base 1.

Al igual que el caso base 1, el proyecto obtiene una ID única en el contrato de proyectos, mediante la cual es posible consultar su información. Adicionalmente, en el caso base 2 también es importante obtener a los financiadores para la futura compra/venta de energía entre el beneficiario y el (los) financiador(es). Esta información puede obtenerse con la ID del proyecto (ID = 259 en este ejemplo) en el contrato de *tokens*. La información de financiamiento obtenida para el caso base 2 es la siguiente:

$$Propietario : 0x6C740120A054cecF209BCaCF62fC38Cc64D98499$$

$$Financiadores : 0xF7ab85D5391651B239873fE0Cf26B6B2f0E52358$$

$$Valor : 1165500$$

4.1.3. Simulación

Al agregar más participantes a la plataforma, la complejidad de ésta y el número de transacciones involucradas en la realización de cada proyecto van en aumento. La simulación de la plataforma considera la realización de proyectos con y sin financiamiento, con 100 cuentas de agentes usando la plataforma en el instante inicial (más del operador), a quienes se les entregó 0.5 *ether* para transacciones. La simulación considera 12 períodos a una tasa de 1 proyecto/período.

Aspectos técnicos

Técnicamente, los proyectos a realizar corresponden a paneles fotovoltaicos de 1 m^2 de superficie para *netbilling*, con una potencia *peak* de 1 kWp en la zona de Santiago de Chile. Estos proyectos tienen un valor referencial de 2.190.000 CLP en el mercado de multiespecialistas. En base a este valor de referencia, los agentes simulados crean ofertas/solicitudes de manera aleatoria. En el caso de los proyectos con financiamiento, se realizan pagos mensuales por la energía considerando un valor de 100 $tokens/kWh$ y una generación promedio de 5 kWh/día por cada proyecto.

Agentes

Los agentes que participan son:

- 24 Propietarios, quienes fijan y actualizan el monto máximo a pagar de manera aleatoria. Una vez que se crea su proyecto, utilizan la plataforma para pagar mensualmente la energía producida por el proyecto a sus acreedores en el caso de haber solicitado financiamiento.
- 20 Proveedores, quienes fijan y actualizan su oferta de manera aleatoria. Una vez que venden sus equipos no vuelven a ofertar en la plataforma.
- 20 Instaladores, quienes fijan y actualizan su oferta de manera aleatoria. Una vez entregados sus servicios no vuelven a ofertar en la plataforma.
- 20 Gestores, quienes fijan y actualizan su oferta de manera aleatoria. Una vez entregados sus servicios no vuelven a ofertar en la plataforma.
- 16 Financiadores, quienes financian los proyectos que en la fase de *crowdfunding*. Cada financiador financia un único proyecto aportando la totalidad de su costo.

Cada grupo de agentes es simulado en un *script* distinto en un mismo computador, resultando en 5 *scripts* ejecutándose en paralelo. Esto introduce cierta asincronía en la simulación, pues el nodo del operador no ve las transacciones de los procesos en paralelo hasta que son minadas por la red, pudiendo ocurrir que las transacciones de cada *script* no sean minadas secuencialmente, haciendo que la secuencia de transacciones entre los *scripts* deje de ser determinista, pudiendo variar entre un *script* y otro.

Saldos

Todas las cuentas comienzan con un mismo saldo inicial. Los propietarios que no requieren financiamiento pagan el costo total del proyecto a los demás agentes en el *step* que se hizo el *matching*. Por otra parte, los propietarios que optaron por financiamiento pagan en cada *step* la energía generada por este. La Figura 4.1 muestra los saldos de los agentes que participaron en proyectos sin financiamiento, mientras que la Figura 4.2 muestra los saldos de los agentes que participaron en proyectos con financiamiento.

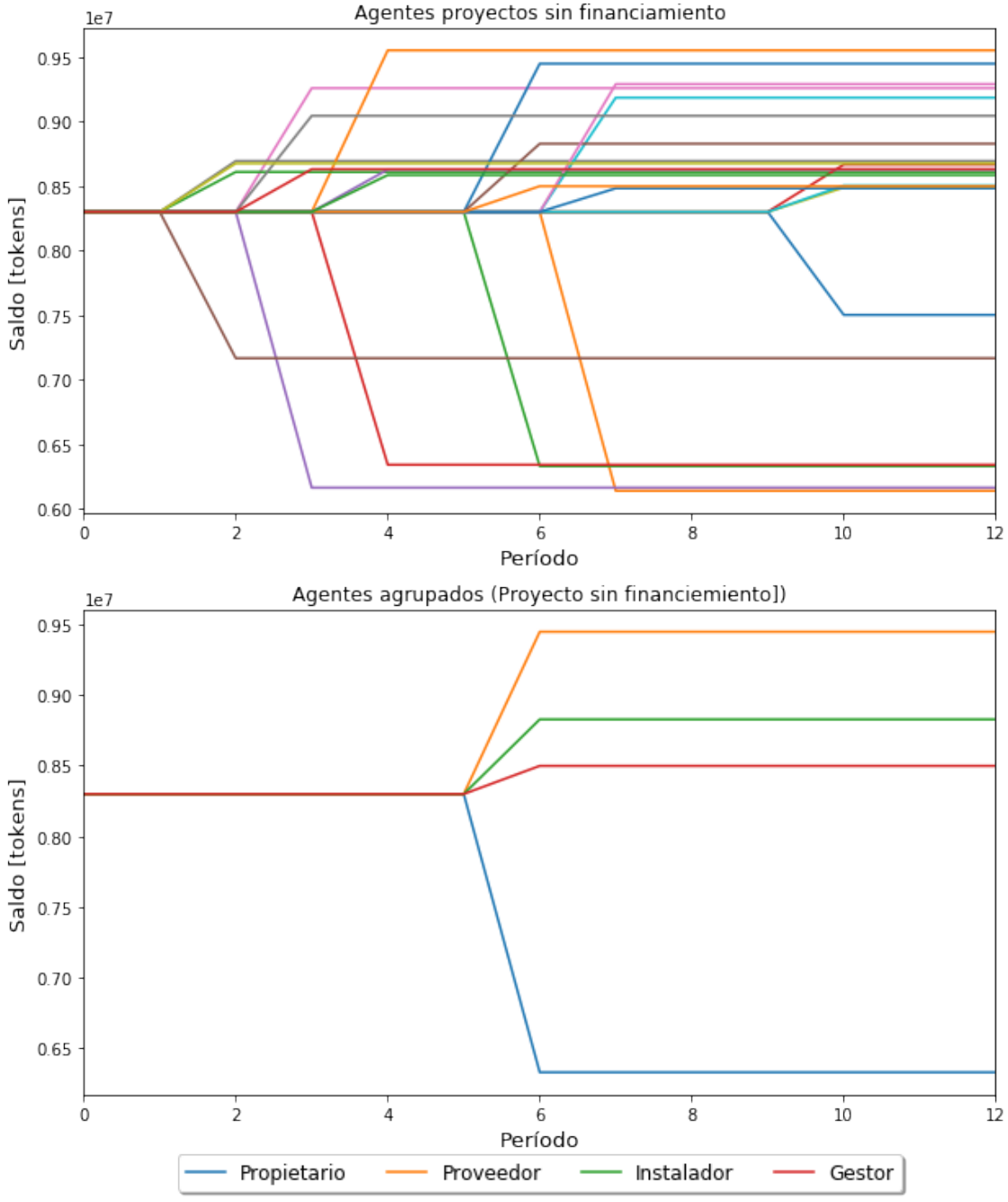


Figura 4.1: Saldos proyectos sin financiamiento

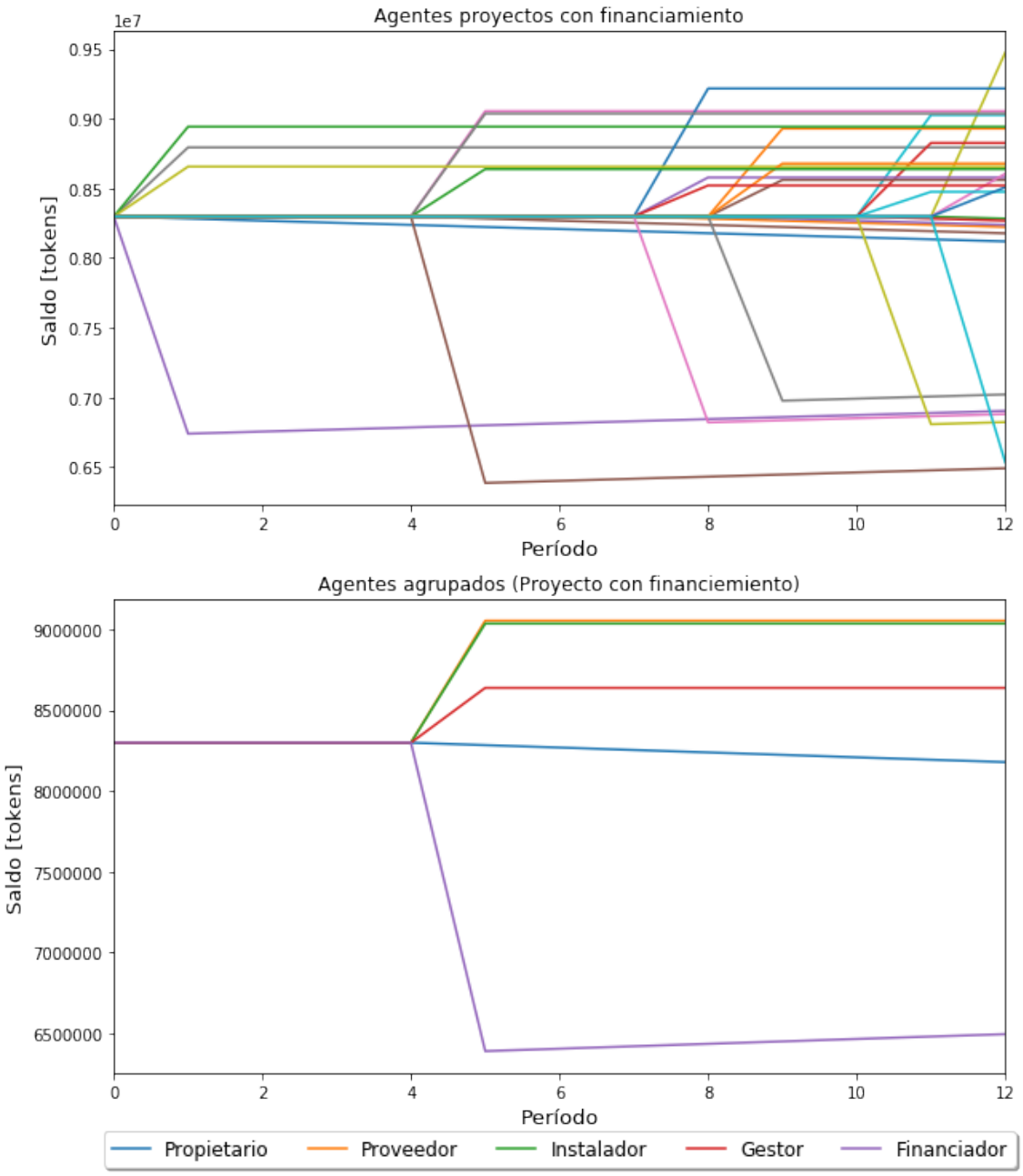


Figura 4.2: Saldos proyecto con financiamiento

Costos en *gas*

Los costos en *gas* de la ĐApp desarrollada son importes para este trabajo por dos razones. Primeramente porque Ethereum impone un límite de *gas* por transacción, de manera que si el *gas* requerido por alguna acción dentro de la ĐApp supera este límite, la ĐApp sería disfuncional, y por lo tanto, inviable. Y en segundo lugar, porque mientras más *gas* requiera una acción dentro de la ĐApp, mayor será su costo en *ether*, lo que implica un mayor costo operacional de la ĐApp, volviéndola menos competitiva. Las estadísticas de los costos en *gas* de las funciones utilizadas durante la simulación se muestran en la Tabla 4.6.

	Veces utilizada	Costo mínimo	Costo promedio	Costo máximo	Costo acumulado
Nueva solicitud	24	66.351	79.459,2	96.415	1.907.021
Nuevo proveedor	20	74.681	75.544,6	75.590	1.510.891
Nuevo instalador	20	74.571	75.434.6	75.480	1.508.691
Nuevo gestor	20	74.593	75.453,4	75.502	1.509.067
Match	7	774.911	819.874,4	834.911	5.739.121
Crowdfunding	5	73.736	73.761,6	73.768	368.808
Chequear	15	21.624	26.713,7	36.893	400.705
Financiar	5	200.960	203.960	215.960	1.019.800
Retirar fondos	5	626.972	710.972	746.972	3.554.860
Pagar	75	22.725	26.975,6	30.513	2.023.171

Tabla 4.6: Estadística de los costos en *gas* de las funciones utilizadas.

Como se puede observar, en general los costos en *gas* se mantienen similares a los de los casos base mostrados anteriormente, siendo la función *matching* la más cara, cuyo costo lo asume el operador de la plataforma. También se observa que todos los costos están por debajo del límite de *gas* por transacción de 7.000.000 *gas* impuesto por la red, dando cuenta que los contratos fueron programados con un nivel de eficiencia aceptable para la operación de la plataforma con cien usuarios haciendo uso de esta simultáneamente.

Para el límite de usuarios que pueden usar la ĐApp simultaneamente, se realizan pruebas de escalabilidad de los *smart contracts* aumentando gradualmente el tamaño de sus *inputs*. Primeramente se prueban las funciones que son independientes del financiamiento de los proyectos, los resultados obtenidos de esta prueba se muestran en la Figura 4.3.

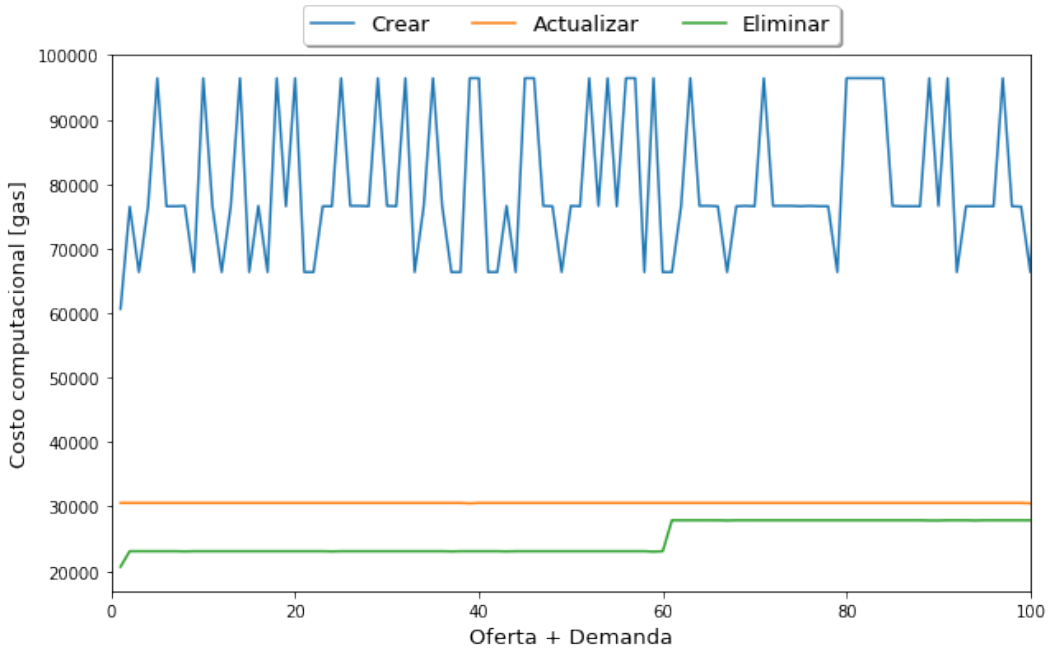


Figura 4.3: Escalabilidad funciones CRUD.

Como se puede apreciar, el costo computacional de las funciones CRUD varían en torno a un valor constante, es decir, las funciones son de complejidad constante (O(1)). La función *read* no se incluye en el gráfico, pues la lectura de datos (*getter*) no tiene costo en *gas*.

Viendo la Figura 3.4, se observa que algunas acciones dentro de la ĐApp cambian según el financiamiento escogido por el propietario. Este es el caso de la función *matching*, cuyo costo es menor cuando el proyecto es con financiamiento y mayor cuando es sin financiamiento como se mostró en el los casos base. Las Figuras 4.4 y 4.5 muestran la escalabilidad de la función *matching* con y sin financiamiento.

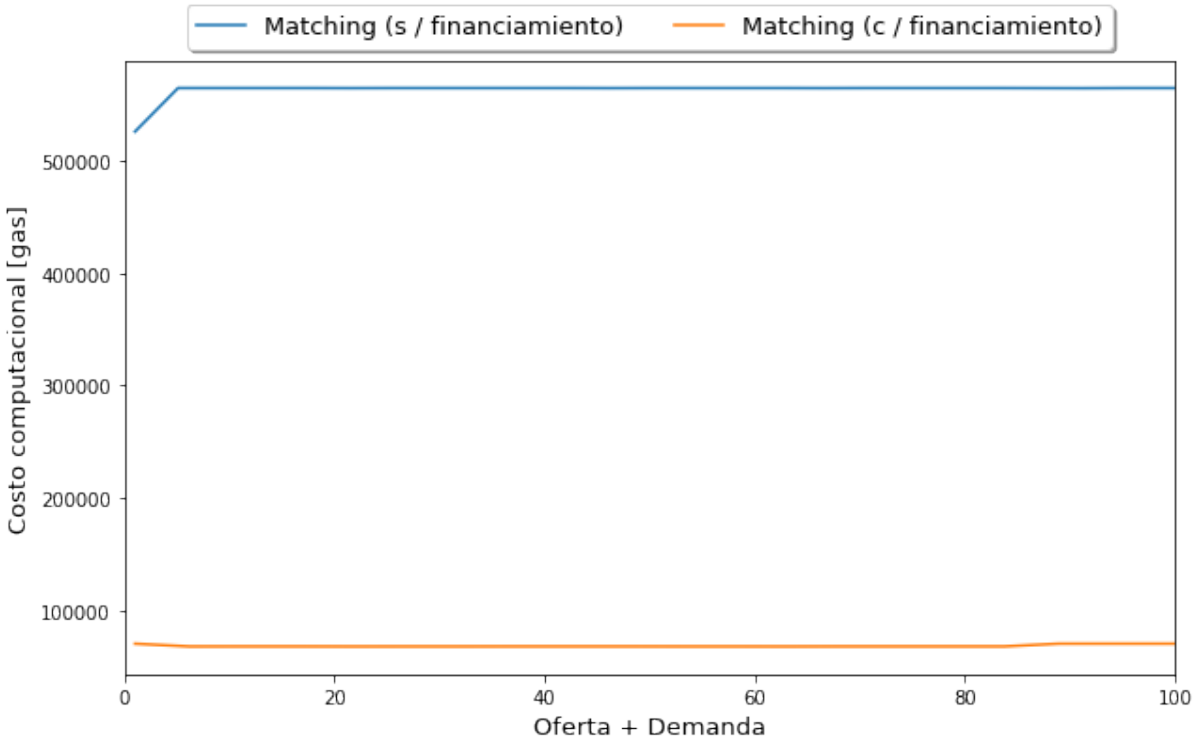


Figura 4.4: Escalabilidad función *matching* (oferta + demanda).

Como se puede apreciar, sin importar el financiamiento del proyecto, el costo de la función *matching* no varía con la cantidad de ofertas/solicitudes existentes (complejidad O(1)). Sin embargo, la necesidad de reiniciar el *crowdfunding* cada vez que se crea un proyecto con financiamiento, hace que la función *matching* deba manipular la memoria en donde se almacena la información de los financiadores del proyecto anterior, de modo que el costo de esta función aumenta proporcionalmente con el número de financiadores (complejidad O(n)). Este comportamiento se aprecia en la Figura 4.5.

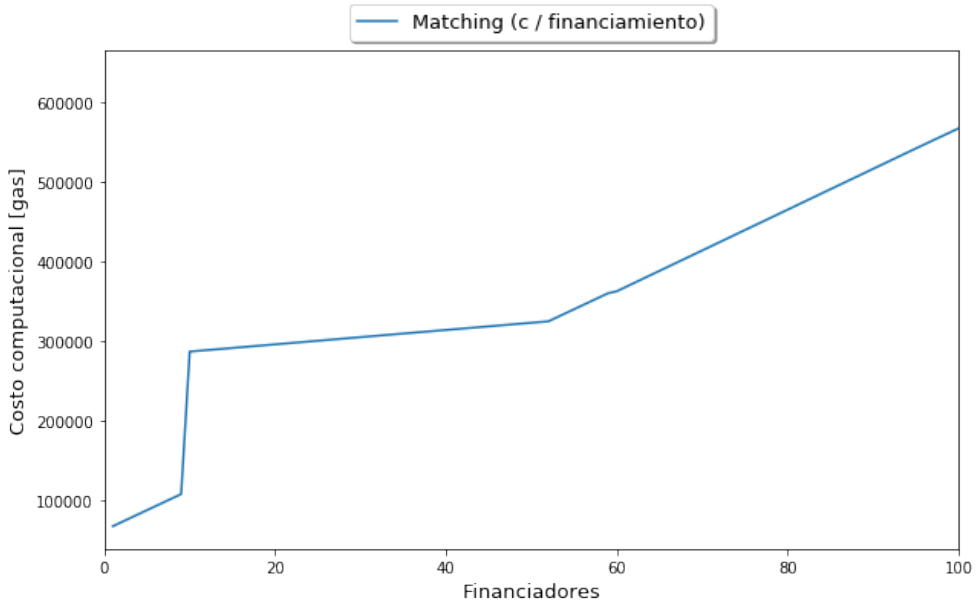


Figura 4.5: Escalabilidad función *matching* (financiadores).

Por último, se estudia la escalabilidad de las funciones *pay* y *whitdraw*. Estas funciones dependen directamente del numero de financiadores, pues la función *whitdraw* guarda en la blockchain la información de los financiadores con sus montos, mientras que la función *pay* recorre ambas listas para efectuar los pagos por la energía producida. La escalabilidad de estas funciones se muestra en la Figura 4.6

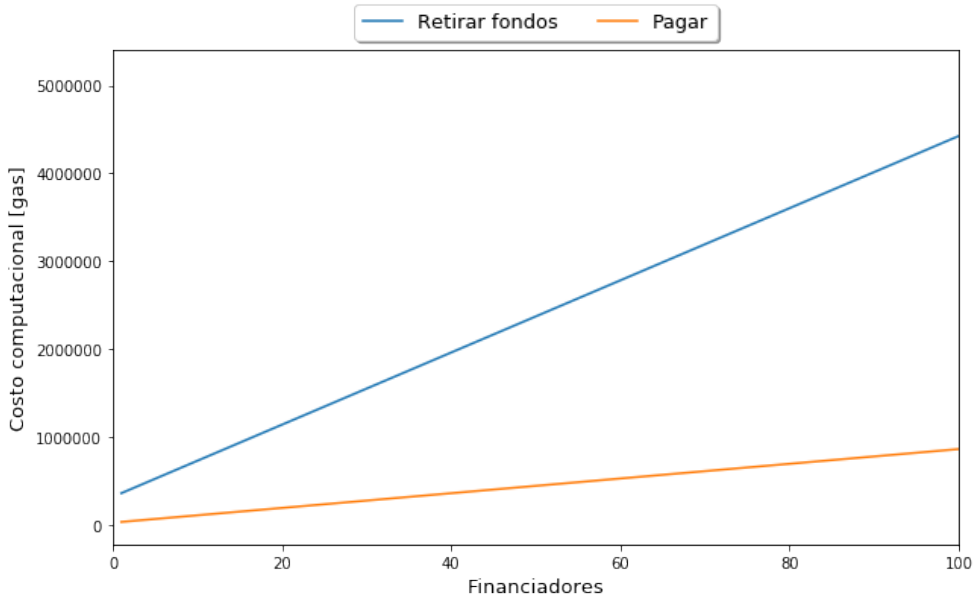


Figura 4.6: Escalabilidad funciones “retirar fondos” y “pagar”.

Como se puede apreciar, los costos de ambas funciones crecen de manera proporcional con el número de financiadores (complejidad O(n)), siendo la función “retirar” la que crece más

abruptamente. La complejidad de las funciones "chequear" y "financiar" es la misma de las funciones CRUD, pues "financiar" hace una operación análoga a "crear" (agregar un elemento a una lista), mientras que "chequear" cambia un valor con una clausula condicional (*if*).

4.2. Análisis

Una vez obtenidos los resultados del funcionamiento de la ĐApp, se realizan análisis, tanto de algunos aspectos generales como contemplando una futura implementación de la plataforma en Chile.

4.2.1. Viabilidad

Diseño

El desempeño de la ĐApp en simulación demostró que ésta es capaz de funcionar con datos reales como parte de una *beta*. Los costos en *gas* de las funciones utilizadas en los casos base y en la simulación se mantienen en niveles por debajo del límite de *gas* impuesto por la red, indicando que la ĐApp tiene buena escalabilidad, a pesar de que su objetivo sea tener la menor cantidad posible de agentes buscando un proyecto al mismo tiempo. En el peor de los casos, el operador puede optar por borrar ofertas/solicitudes de manera arbitraria, guiándose por factores como la reputación de los agentes para reducir para mantener estable el servicio de la aplicación.

Normativa

El funcionamiento de la plataforma no genera conflicto con la regulación chilena del sector energético. En el caso de la compra/venta de energía entre pares de los proyectos con financiamiento, esta se realiza desde el lado del consumidor, por lo que las compañías distribuidoras no perciben una transferencia de energía de un medidor a otro. En el caso de las áreas de concesión, los proyectos corresponden a proyectos de *netbilling* o autogeneración aislados físicamente entre ellos, por lo que la implementación de la plataforma no convierte al operador en una distribuidora. De manera general, los pagos por energía dentro de la plataforma pueden verse como una generalización de una cooperativa virtual, en donde la energía comercializada no sale del empalme del proyecto.

Escalabilidad

De acuerdo con las simulaciones realizadas, la ĐApp puede funcionar correctamente con 100 usuarios haciendo uso simultaneo de esta. Sin embargo, los costos computacionales alcanzan valores cercanos al límite impuesto por la red Ethereum cuando se hacen proyectos

con financiamiento con 100 financiadores. Por lo tanto, se recomienda un máximo de 120 financiadores por proyecto para no sobrepasar el límite de *gas* por transacción.

4.2.2. Blockchain pública vs privada

Las simulaciones realizadas pueden hacerse ya sea en una blockchain pública como en una blockchain privada, pues su funcionamiento es análogo. Sin embargo, la operación de la plataforma en una red u otra posee distintas ventajas y desventajas. En el caso de la blockchain pública, en donde se realizó la simulación, cuenta con la ventaja de seguridad provista por el número de *miners* y el número de nodos en la red. Gracias a esto funciones como el *crowdfunding* se verían beneficiadas, pues estarían abiertas a todos los participantes de la red Ethereum, quienes bajo incentivos económicos, podrían comprar/vender *tokens* y/o ser financiadores de proyectos, aumentando así el número de proyectos realizados.

Por otra parte, los incentivos para usar una blockchain privada yacen en que la plataforma está diseñada para operar en una zona geográfica definida, por lo que no es deseable que toda la red pueda acceder a funciones como "Nueva solicitud", pues no tiene sentido que un individuo fuera del territorio nacional manifieste deseo de realizar un proyecto. Por otro lado, una implementación en blockchain privada implicaría gastos en nodos que hagan de *miners* en la blockchain.

4.2.3. Factores económicos

Como se mencionó anteriormente, cada transacción realizada en la blockchain de Ethereum tiene un costo, que debe pagarlo (en *ether*) quien emite la transacción. En el caso de la red de prueba Rinkeby, la obtención de *ether* es gratuita, por lo que de operar en esta red, el uso de la plataforma no tendría costo alguno. Sin embargo, es deseable que en algún momento la plataforma migre de este entorno de prueba (Rinkeby) a la red oficial de Ethereum (*Mainnet*), lo cual introducirá costos por el *ether* usado en la plataforma.

El *fee* o costo en *ether* de una transacción es equivalente al producto entre el *gas* requerido y el *gas price* escogido. A mayor *gas price*, menor el tiempo que demora la transacción en ser agregada a la blockchain. El costo en CLP de la transacción equivale a su costo en *ether* multiplicado por el valor del *ether* al momento de la conversión. Por ejemplo, el costo promedio de la función "nueva solicitud" estaría dado por:

$$gas \text{ usado} : 79.459$$

$$gas\ price \text{ promedio} : 2\ [gwei] \text{ (proporcionado por la red)}$$

$$gas\ price \text{ escogido} : 1\ [gwei]$$

$$fee : 0.0000795\ [ether]$$

$$\text{valor } ether : 105.000\ [CLP]$$

costo transacción : 8,34 $[CLP]$

tiempo de verificación : 460 $[s]$

Es posible que el valor en CLP de una transacción sea volátil por depender del valor del *ether*. Sin embargo, en la ĐApp desarrollada, el tiempo de verificación no es una variable crítica, de modo que ul usuario tiene la posibilidad mitigar las variación de costos modificando el *gas price* de sus transacciones. A la fecha, el *gas price* promedio es de 2 *gwei* ($2 * 10^{-9}$ *ether*, ver Anexo C) por transacción y el valor del *ether* es de ∼105.000 CLP. Con estos valores, algunos precios relevantes de la plataforma se muestran en la tabla 4.7.

Acción	Costo promedio [/gas]]	Costo promedio [gwei]	Costo promedio [CLP]
Desplegar *smart contracts*	7.490.235	14.980.470	1.582
Nueva solicitud	79.459,2	158.918,4	17
Nuevo proveedor	75.544,6	151.089,2	16
Nuevo instalador	75.434,6	150.868	16
Nuevo gestor	75.453,4	150.906,8	16
Matching	819.874,4	1.639.748,8	173
Crowdfunding	73.761,6	147.523,2	16
Financiar	203.960	407.920	43
Retirar fondos	710.972	1.421.944	150

Tabla 4.7: Costos en CLP de acciones dentro de la ĐApp

Como se puede apreciar, los costos dentro de la ĐApp son órdenes de magnitud menores al precio de referencia del proyecto especificado, lo que implica un costo mínimo por concepto de intermediario para los proyectos, validando la ĐApp desarrollada como una opción competitiva en el mercado. En la Tabla 4.8 se comparan los costos anuales de la ĐApp desarrollada con los de una aplicación sin blockchain (base de datos centralizada) considerando 100 proyectos al año y los valores encontrados en el mercado para *hosting* y registro.

	ĐApp web	ĐApp móvil	App web	App móvil.
Hosting (página web)	20.000	-	40.000	40.000
Cuota registro (Play Store)	-	17.000	-	17.000
Proyectos (*gas*)	18.000	18.000	-	-
Total	38.000	35.000	40.000	57.000

Tabla 4.8: Comparación de costos entre alternativas centralizadas y descentralizadas.

Se observa que los costos de la ĐApp son menores a los de la opción centralizada para todas las alternativas. Por una parte, el costo del *hosting* de una ĐApp es menor, pues solo se necesita almacenar la interfaz de los *smart contracts*, ya que la base de datos se almacena en la blockchain. Y por otro lado, en la ĐApp aparecen los costos en *gas*, que aumentan proporcionalmente con el número de proyectos.

En la simulación realizada, para la eficiencia de los *smart contracts*, el *gas price* y el valor *ether* considerados, el uso de la ĐApp es más económico a una tasa de proyectos menor o igual a 100 proyectos/año. Sin embargo, a mayores tasas de proyectos, las opciones centralizada y descentralizada tendrán costos iguales. Si se diera este escenario, el operador debería optimizar los *smart contracts* o usar un *gas price* para reducir su costo operacional.

Como aspecto negativo de la ĐApp se encuentra el cobro por acciones dentro de la aplicación (transacciones) impuesto por la red Ethereum. No obstante, el impacto para los usuarios producto de este cobro es mínimo, pues el cobro es marginal con respecto al valor de los proyectos (cientos contra millones de CLP). De todas formas, el operador tiene la opción de modificar los *smart contracts*, ya sea para reducir los costos en *gas* de todas las operaciones, o para reasignar los costos de las transacciones con el fin de reducir sus propios costos a costa de los demás usuarios.

4.2.4. Pagos y *tokens*

Tratándose de una plataforma digital que habilita pagos, es necesario implementar una divisa digital o *token* para que estos se puedan realizar. Para no considerar la volatilidad de las criptomonedas, la implementación actual de la plataforma crea un *token* con valor fijo, el cual se distribuye inicialmente de manera centralizada. Sin embargo, es posible que lidiar con estos *tokens* resulte poco amigable con el usuario. Es por esto que el ciclo de los *tokens* desde la entidad centralizada hasta el proveedor de bienes/servicios debería hacerse de manera subyacente a un medio de pago tradicional con dinero fiat, de manera que la existencia del *token* pase desapercibida por los usuarios y la ĐApp se asemeje lo más posible a una aplicación tradicional. Alternativamente, podría plantearse el uso de este *token* para otras funciones, por ejemplo, como medio de pago dentro de una comunidad.

4.2.5. Eficiencia y seguridad

Si bien blockchain es considerada una tecnología segura, los *smart contracts* son desarrollados por personas, lo cual no libera a una ĐApp de tener falencias en la eficiencia de sus algoritmos o en sus medidas de seguridad. Si bien los costos en *gas* reflejan cierta eficiencia de la ĐApp es necesaria una revisión rigurosa de sus algoritmos, ya que es posible que esta sufra ataques DDoS producto de un algoritmo mal diseñado. Junto con lo anterior, si bien se implementan medidas de seguridad como función *matching* que son accesibles sólo para el operador de la plataforma, la simulación realizada considera el comportamiento ideal esperado por parte de los usuarios de la red, por lo que una futura operación de la plataforma con datos reales, necesitará una revisión en detalle de posibles fallas de seguridad para resguardar debidamente los activos de los usuarios.

Capítulo 5

Conclusiones y trabajo futuro

5.1. Conclusiones

Este trabajo de título desarrolla una aplicación blockchain o ĐApp para el sector energético chileno. La ĐApp desarrollada consiste en un *marketplace* que reúne a propietarios, proveedores, instaladores, gestores y financiadores en pos de la realización de proyectos GD para la disminución de los *soft costs* de estos proyectos. La aplicación además cuenta con un sistema de pagos por venta de energía generada por los proyectos.

A partir del contexto internacional, se corroboró que existen diversas aplicaciones de blockchain para el sector eléctrico, tales como transferencia de energía P2P, aplicaciones para vehículos eléctricos y emisión de certificados en países como Alemania, Estados Unidos, Reino Unido y Suiza, entre otros.

Por otro lado, la contextualización nacional mostró que existe un escaso desarrollo de aplicaciones blockchain en energía, siendo dos las aplicaciones conocidas existentes, ambas orientadas a la certificación de datos. Con respecto a las cooperativas eléctricas, su modelo organizacional guarda similitudes con el paradigma detrás de blockchain y los *smart contracts*.

Se demostró de forma empírica que es posible el desarrollo de un sistema de economía colaborativa basado en blockchain con opciones de financiamiento y de pago. Los costos operacionales de la ĐApp resultaron ser órdenes de magnitud menor que el valor de los proyectos, validando a la ĐApp como una herramienta útil para mediar entre los agentes involucrados en proyectos de generación distribuida.

Por último, la simulación realizada muestra que la plataforma es capaz de operar correctamente con un centenar de usuarios realizando proyectos con y sin financiamiento. Se observa que los costos en *gas* de las funciones de los agentes se mantienen por debajo del valor máximo impuesto por la red de Ethereum y que los costos en CLP para el operador de la plataforma son menores en comparación a la alternativa centralizada en la medida que los *smart contracts* estén programados de forma eficiente.

5.2. Trabajo futuro

Primeramente, la plataforma escogida para el desarrollo de este trabajo fue la red de Ethereum, sin embargo, existen otras plataformas blockchain para el desarrollo de ĐApps, incluyendo algunas diseñadas específicamente para el sector energía. Se propone como trabajo futuro hacer una comparativa entre las distintas plataformas existentes para la aplicación desarrollada. Alternativamente, se podría desarrollar una versión alternativa de la ĐApp desarrollada en este trabajo en otra plataforma para posteriormente escoger una sobre la cual articular todo el trabajo posterior en el desarrollo de la ĐApp.

Una segunda observación es que el presente trabajo utilizó blockchain como administrador de los datos ingresados por los usuarios, obviando la procedencia de dichos datos. Si bien los datos de oferta y demanda pueden ser ingresados por cualquier persona que participe de la plataforma, la energía generada por un proyecto debe subirse a la blockchain mediante un *smart meter*. Como extensión de este trabajo se propone el desarrollo de un *smart meter* que sea capaz de subir sus mediciones a la blockchain e interactuar con los *smart contracts* de la ĐApp desarrollada, pensando en su futura implementación.

Por otro lado, para que la aplicación desarrollada pueda operar en el mundo real, además de los *smart contracts*, es necesario el desarrollo de una interfaz gráfica para su uso por parte de los usuarios. Una interfaz para esta plataforma debe interactuar con la blockchain de Ethereum y los *smart contracts*, administrando las billeteras de los usuarios de manera amigable y segura. El desarrollo de esta interfaz constituye la siguiente etapa en el desarrollo de esta ĐApp para poder poder realizar pruebas con usuarios reales como parte de una versión beta.

Otra mejora importante para la ĐApp desarrollada es la optimización de los códigos desarrollados. Esta tarea es fundamental, pues un código optimizado puede reducir considerablemente los costos en *gas*, tanto de los usuarios como del operador de la plataforma, mejorando la rentabilidad y competitividad de la ĐApp frente a una aplicación centralizada. Si se logra obtener bajos costos en *gas* para los agentes, a la vez que se les delega parte del procesamiento realizado por la función *matching*, el costo por proyecto para el operador de la plataforma puede disminuir hasta en un orden de magnitud.

Y por último, pero no menos importante, se propone estudiar posibles fallas de seguridad dentro de la plataforma. Esto debido a que el carácter inmutable de la tecnología blockchain podría resultar en que sea imposible restablecer el funcionamiento de la plataforma una vez explotada una falla de seguridad. Un código seguro y testeado impediría la fuga indebida de activos hacia un tercero y mantendría operativa la plataforma la mayor parte del tiempo, proporcionando los permisos adecuados a cada persona que acceda a la plataforma.

Glosario

ĐApp Distributed Application. iii, vi, 1, 3, 13, 14, 29, 36, 43–45, 47, 49–52

CLP Peso chileno. i, 31, 36, 41, 48–51

CPU Central Processing Unit. 5, 16, 17

CRUD *Create, Read, Update, Delete*. vi, 33, 44, 45, 47

DDoS Distributed Denial of Service. 50

DLT Distributed Ledger Technology. 20

ERNC Energías Renovables No Convencionales. 26

EVM Ethereum Virtual Machine. 30, 60

GD Generación Distribuida. 29, 51

IoT Internet of Things. 1

IPTV Internet Protocol Television. 6

ISP Internet Service Provider. 6

P2P Peer to Peer. i, 5, 6, 11, 13, 22, 23, 51

PoA Proof of Authority. 17, 30

PoS Proof of Stake. 13, 17

PoW Proof of Work. 13, 15

TICS Tecnologías de la Información y Comunicación. 20

VES Vehículos Eléctricos. 23

Bibliografía

[1] R. Abdullah and M.A. Faizal. "Block Chain: Cryptographic Method in Fourth Industrial Revolution". *International Journal of Computer Network and Information Security*, 10:9–17, Nov 2018.

[2] V. Brilliantova and T. Thurner. "Blockchain and the future of energy". *Technology in Society*, Nov 2018.

[3] L. Kawulok, K. Zielinski, and M. Jaeschke. Trusted group membership service for jxme (jxta4j2me). In *WiMob'2005), IEEE International Conference on Wireless And Mobile Computing, Networking And Communications, 2005.*, volume 4, pages 116–121, Aug 2005.

[4] X. Shen, H. Yu, J. Buford, and M. Akon. *Handbook of Peer-to-Peer Networking*. Springer Publishing Company, Incorporated, 1st edition, 2009.

[5] J.F. Kurose and K.W. Ross. *Computer Networking: A Top-Down Approach*. Pearson, 6th edition, 2012.

[6] H. Xie, Y.R Yang, A. Krishnamurthy, Y.G. Liu, and A. Silberschatz. "P4P: Provider Portal for Applications". *SIGCOMM Comput. Commun. Rev.*, 38(4):351–362, Aug 2008.

[7] S. Le Blond, C. Zhang, A. Legout, K. Ross, and W. Dabbous. "I Know Where You Are and What You Are Sharing: Exploiting P2P Communications to Invade Users' Privacy". In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, pages 45–60, New York, NY, USA, 2011. ACM.

[8] Z. Liu, P. Dhungel, D. Wu, C. Zhang, and K. W. Ross. "Understanding and Improving Ratio Incentives in Private Communities". In *2010 IEEE 30th International Conference on Distributed Computing Systems*, pages 610–621, June 2010.

[9] J. Katz and Y. Lindell. *Introduction to Modern Cryptography (Chapman & Hall/Crc Cryptography and Network Security Series)*. Chapman & Hall/CRC, 2007.

[10] H. Delfs and H. Knebl. *Introduction to Cryptography: Principles and Applications*. Springer-Verlag, Berlin, Heidelberg, 2001.

[11] D. Drescher. *Blockchain Basics : A Non-Technical Introduction in 25 Steps*. APRESS, New York, 2017.

[12] A.M. Antonopoulos. *Mastering Bitcoin: Unlocking Digital Crypto-Currencies*. O'Reilly Media, Inc., 1st edition, 2014.

[13] C. Dannen. *Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners*. Apress, Berkely, CA, USA, 1st edition, 2017.

[14] R. Farell. "An analysis of the cryptocurrency industry". B.S. Thesis, University of Pennsylvania., 2015.

[15] S. Nakamoto. "Bitcoin: A Peer-to-Peer Electronic Cash System". *Cryptography Mailing list at https://metzdowd.com*, Mar 2009.

[16] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press, Princeton, NJ, USA, 2016.

[17] V. Buterin. "Ethereum: A next-generation smart contract and decentralized application platform". [en línea] `https://github.com/ethereum/wiki/wiki/White-Paper`, 2013. [consulta: 08 diciembre 2018].

[18] A.M. Antonopoulos and G. Wood. *Mastering Ethereum: Building Smart Contracts and Dapps*. O'Reilly Media, Incorporated, 2018.

[19] G. Coulouris, J. Dollimore, T. Kindberg, and G. Blair. *Distributed Systems: Concepts and Design*. Addison-Wesley Publishing Company, USA, 5th edition, 2011.

[20] U.W. Chohan. "The Double Spending Problem and Cryptocurrencies". *SSRN Electronic Journal*, Jan 2017.

[21] D. Guegan. "Public blockchain versus private blockhain". *Documents de travail du Centre d'Economie de la Sorbonne - ISSN : 1955-611X*, 2017.

[22] S. King and S. Nadal. "PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake". *Self-Published Paper*, 2012.

[23] M. Andoni, V. Robu, D. Flynn, S. Abram, D. Geach, D. Jenkins, P. McCallum, and A. Peacock. "Blockchain technology in the energy sector: A systematic review of challenges and opportunities". *Renewable and Sustainable Energy Reviews*, 100:143 – 174, 2019.

[24] R. Chitchyan and J. Murkin. "Review of Blockchain Technology and its Expectations: Case of the Energy Sector". *arXiv e-prints*, 2018.

[25] LO3 Energy. "Brooklyn Microgrid". [en línea] `http://brooklynmicrogrid.com/`. [consulta: 19 noviembre 2018].

[26] Kasey Panetta. "Top Trends in the Gartner Hype Cycle for Emerging Technologies, 2017". [en línea] `http://www.gartner.com/smarterwithgartner/top-trends-in-the-gartner-hype-cycle-for-emerging-technologies-2017/`.

[consulta : 19 diciembre 2018].

[27] M. Walport. Distributed ledger technology: beyond blockchain. [en línea] `https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/492972/gs-16-1-distributed-ledger-technology.pdf`. [consulta: 19 diciembre 2018].

[28] M. Swan. *Blockchain : blueprint for a new economy*. O'Reilly Media, Sebastopol, Calif., 2015.

[29] D. Tapscott and A. Tapscott. *Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World*. Brilliance Audio, 2016.

[30] O. Konashevych. "Advantages and Current Issues of Blockchain Use in Microgrids". *Èlektronnoe modelirovanie*, 38:93–104, Agosto 2016.

[31] Gartner. "Gartner identifies three megatrends that will drive digital business into the next decade". [en línea] `https://www.gartner.com/newsroom/id/3784363`. [consulta: 19 diciembre 2018].

[32] Y. Cao. "Energy internet blockchain technology". In *The Energy Internet*, pages 45 – 64. Woodhead Publishing, 2019.

[33] J. Wu and N. Tran. "Application of Blockchain Technology in Sustainable Energy Systems: An Overview". *Sustainability*, 10:3067, Aug 2018.

[34] Y. Parag and B. Sovacool. "Electricity market design for the prosumer era". *Nature Energy*, 1:16032, Mar 2016.

[35] J. Murkin, R. Chitchyan, and D. Ferguson. "Goal-Based Automation of Peer-to-Peer Electricity Trading". In *From Science to Society*, pages 139–151, Cham, 2018. Springer International Publishing.

[36] I. Kounelis, G. Steri, R. Giuliani, D. Geneiatakis, R. Neisse, and I. Nai-Fovino. "Fostering consumers' energy market through smart contracts". In *2017 International Conference in Energy and Sustainability in Small Developing Economies (ES2DE)*, pages 1–6, July 2017.

[37] A. Hahn, R. Singh, C. Liu, and S. Chen. "Smart contract-based campus demonstration of decentralized transactive energy auctions". In *2017 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–5, April 2017.

[38] E. Münsing, J. Mather, and S. Moura. "Blockchains for decentralized optimization of energy resources in microgrid networks". In *2017 IEEE Conference on Control Technology and Applications (CCTA)*, pages 2164–2171, Aug 2017.

[39] E. Mengelkamp, B. Notheisen, C. Beer, D. Dauer, and C. Weinhardt. "A blockchain-based smart grid: towards sustainable local energy markets". *Computer Science - Research and Development*, pages 1–8, 08 2017.

[40] F. Knirsch, A. Unterweger, and D. Engel. "Privacy-preserving blockchain-based electric vehicle charging with dynamic tariff decisions". *Computer Science - Research and Development*, Septiembre 2017.

[41] IBM. "Energy-Blockchain Lab". [en línea] `https://www.ibm.com/case-studies/energy-blockchain-labs-inc`. [consulta: 8 diciembre 2018].

[42] GridPlus. "Grid+". [en línea] `https://gridplus.io`. [consulta: 8 diciembre 2018].

[43] ImpactPPA. "ImpactPPA". [en línea] `https://www.impactppa.com`. [consulta: 8 diciembre 2018].

[44] LO3 Energy. "LO3 Energy". [en línea] `https://lo3energy.com`. [consulta: 8 diciembre 2018].

[45] MyBit. Mybit. [en línea] `https://mybit.io`. [consulta: 8 diciembre 2018].

[46] PowerLedger. "PowerLedger". [en línea] `https://www.powerledger.io`. [consulta: 8 diciembre 2018].

[47] WePower. "We Power". [en línea] `https://wepower.network`. [consulta: 8 diciembre 2018].

[48] CNE. "Energía Abierta". [en línea] `http://energiaabierta.cl/`. [consulta: 19 dicembre 2018].

[49] Phineal. "Sellosol". [en línea] `https://www.sellosol.com/`. [consulta: 19 diciembre 2018].

[50] Ethereum. "Rinkeby Authenticated Faucet". [en línea] `https://faucet.rinkeby.io/`. [consulta: 8 diciembre 2018].

[51] Etherscan. "Etherscan". [en línea] `https://rinkeby.etherscan.io/`. [consulta: 1 enero 2019].

[52] D.R. Stinson. *Cryptography: Theory and Practice, Third Edition*. Discrete Mathematics and Its Applications. Taylor & Francis, 2005.

Anexo A. Cifrado por desplazamiento

Sea x un carácter perteneciente al conjunto de letras del alfabeto, e_k un algoritmo de encriptación y d_k un algoritmo de desencriptación. Podemos definir el cifrado por desplazamiento como:

$$e_k(x) = (x + k) \ mod \ 26;$$
$$d_k(x) = (x - k) \ mod \ 26$$

En donde $(a+b) \ mod \ c$ es la suma módulo, operación que puede realizar cualquier computador. Es posible utilizar este algoritmo mediante la siguiente tabla de correspondencia:

A	B	C	D	E	F	G	H	I	J	K	L	M	N
0	1	2	3	4	5	6	7	8	9	10	11	12	13

O	P	Q	R	S	T	U	V	W	X	Y	Z
14	15	16	17	18	19	20	21	22	23	24	25

De esta manera, el mensaje "*holamundo*"puede reescribirse como:

7 14 11 0 12 20 13 3 14

Luego, el emisor y el receptor pueden ponerse de acuerdo en utilizar, por ejemplo, una llave K = 11, de modo que el mensaje numérico sea transformado a:

18 25 22 11 23 5 24 14 25

Para finalmente ser convertido a texto según la tabla de correspondencia en:

s z w l x f y o z

De esta manera. El receptor puede realizar la operación inversa (resta módulo 26) con la llave K = 11 conocida para recuperar el mensaje *holamundo*. Por otro lado, alguien que no conoce la llave escogida, se ve forzado a probar todos los K posibles(26 posibilidades). Debido a que esto puede realizarse de manera "fácil", para que un sistema sea seguro, el conjunto de llaves posibles debe ser muy grande [52].

Anexo B. Terminología Blockchain

- **Nodo**: Cualquier participante en la red de comunicación.
- **Sistema**: Un sistema es la red de comunicación completa. Por lo tanto, todos los nodos resultan en un sistema.
- **Transacción**: Es la información que se envía de un nodo a otro. Las transacciones se registran en cada bloque de la cadena de bloques.
- **Pago**: Transacción de valores económicos.
- **Blockchain Privada**: Cadena de bloques con acceso restringido para nuevos nodos, y donde los nodos deben tener permisos para realizar operaciones.
- **Cadena pública**: Cadena de bloques abierta para todos, en donde todos los nodos tienen los mismos derechos.
- **Consenso**: El consenso se logra cuando la mayoría de los nodos acepta una decisión. Normalmente se usa para validar transacciones en la blockchain.
- **Contratos inteligentes**: Las aplicaciones en el protocolo blockchain se llaman contratos inteligentes o *smart contracts*.
- **Solidity**: Lenguaje de programación utilizado para escribir *smart contracts* en la EVM. Similar a JavaScript.
- **Ethereum Virtual Machine (EVM)**: Los contratos inteligentes de Ethereum son ejecutados por todos los nodos participantes a través de esta máquina virtual.
- **Gas**: Medición utilizada para la complejidad de las transacciones. Un contrato inteligente más complejo proporciona un alto consumo de gas. Interesa minimizar el uso de gas de una transacción.
- **Ether**: "Combustible" necesario para operar la plataforma de aplicación distribuida Ethereum. Es una forma de pago realizada por los clientes de la plataforma a las máquinas que ejecutan las operaciones solicitadas.
- **Token**: Unidad de valor, emitida por una entidad privada, que tiene el valor que se le otorga dentro de una comunidad (e.g ether, bitcoin, fichas).

Anexo C. Unidades Ethereum

Unidad	Valor [*ether*]
Wei	$1 * 10^{-18}$
Kwei	$1 * 10^{-15}$
Mwei	$1 * 10^{-12}$
Gwei	$1 * 10^{-9}$
Szabo	$1 * 10^{-6}$
Finney	$1 * 10^{-3}$
Ether	1
Kether	$1 * 10^{3}$
Mether	$1 * 10^{6}$
Kether	$1 * 10^{9}$
Gether	$1 * 10^{12}$
Tether	$1 * 10^{15}$

Anexo D. Códigos

Todos los codigos programados para la realización del presente trabajo, incluyendo las simulaciones y pruebas de funcionamiento se encuentran disponibles en el siguiente repositorio de *github*:

$$https://github.com/rsilvav/blockchain\_gd$$