Universidad de La Habana
Facultad de Matemática y Computación

Sistema de subasta a ciegas sobre Quorum

Autor:
Alben Luis Urquiza Rojas

Tutor:
Dr.C. Yaidir Mustelier Ruiz

Trabajo de Diploma
presentado en opción al título de
Licenciado en Ciencia de la Computación

Fecha: Noviembre 2022

github.com/ic-matcom/blind-auction-quorum

Contents

Introducción **1**

1 Subastas y blockchain **4**
1.1 Subastas 4
1.1.1 Tipos de subastas 4
1.1.2 Mercado de deuda 6
1.1.3 Mercado de Deuda Pública en Cuba 7
1.2 Blockchain 8
1.2.1 Quorum 9
1.2.2 Subastas a ciegas sobre *blockchain* 10

2 Propuesta **13**
2.1 Ventajas de la propuesta 13
2.2 Condiciones iniciales 14
2.3 Proceso de Subasta 14
2.3.1 Desplegar contrato 15
2.3.2 Fase de ofertas 15
2.3.3 Revelación de ofertas 16
2.3.4 Verificación y publicación de los ganadores 17
2.3.5 Finalización 17
2.4 Seguridad 18
2.4.1 Obligación de pago del beneficiario 18
2.4.2 Keccak256 19

3 Experimentación y Resultados **21**
3.1 Detalles de Implementación 21
3.2 Experimentación 22
3.2.1 Tecnologías Utilizadas 22
3.2.2 Gas 22
3.2.3 Pruebas Realizadas 25
3.3 Resultados 28

Conclusiones **29**

Recomendaciones **31**

Bibliography **32**

List of Figures

1.1 Ejemplo de subasta holandesa en el mercado de deuda 8

3.1 Diagrama donde se resume, el funcionamiento interno de un contrato inteligente 24
3.2 26
3.3 27

Introducción

Cuando el 31 de octubre de 2008 Satoshi Nakamoto publicó el artículo *Bitcoin: A Peer-to-Peer Electronic Cash System* (documento técnico original de la bien conocida y primera criptomoneda: Bitcoin) , creó las bases de una tecnología que está revolucionando al mundo, y el autor no se refiere al que bien pudiera ser el sistema de pago que sustituya al dólar y al dinero *fiat* en un futuro, sino a la *blockchain* (Nakamoto, 2008).

Blockchain se traduce como cadena de bloques. Básicamente, es un conjunto de tecnologías que permiten llevar un registro seguro, descentralizado, sincronizado y distribuido de operaciones digitales, sin necesidad de la intervención de terceros (Solunion, 2021).

En ese sentido, la definición más completa es la dada por Don & Alex Tapscott en su libro Blockchain Revolution (Tapscott & Tapscott, 2016): "un libro de contabilidad digital incorruptible de transacciones económicas que se puede programar para registrar no solo transacciones financieras, sino prácticamente todo lo que tiene valor". Cada uno de los bloques de datos se encuentra protegido y vinculado entre sí criptográficamente. Las transacciones no las verifica un tercero, sino la red de nodos (computadores conectados a la red), que también es la que autoriza en consenso cualquier actualización en la *blockchain* (Solunion, 2021).

A finales de 2013, Vitalik Buterin publica el que luego se convertiría en el documento técnico (*white paper*) de Ethereum (Buterin, 2013). Este joven, quien hasta ese momento era uno de los programadores involucrado en el ecosistema Bitcoin, había notado el potencial de la criptografía para el desarrollo de aplicaciones descentralizadas. No obstante, su propuesta de crear un lenguaje de scripting para Bitcoin, que hiciera esto posible, no tuvo resonancia suficiente. Fue entonces cuando se propuso el desarrollo de una red independiente, con su propia infraestructura, para el desarrollo de un criptoactivo y una cadena de bloques capaz de soportar aplicaciones descentralizadas. Y el 30 de julio de 2015, Vitalik conjuntamente con otros programadores pusieron en línea la *blockchain* de Ethereum (Díaz, 2018).

La red de Ethereum llevó a la práctica un nuevo concepto, los contratos inteligentes, en inglés conocidos como *smart contracts*. La definición más simple al respecto es que se trata de contratos que tienen la capacidad de cumplirse de

forma automática una vez que las partes han acordado los términos. Su nombre hace recordar a los contratos legales firmados en papel. Pero a pesar de que tienen cosas en común, son totalmente diferentes.

Los contratos inteligentes son programas informáticos. No están escritos en lenguaje natural, sino en código virtual. Son un tipo de software que se programa, como cualquier otro software, para llevar a cabo una tarea o serie de tareas determinadas de acuerdo a las instrucciones previamente introducidas. Su cumplimiento, por tanto, no está sujeto a la interpretación de ninguna de las partes: si el evento A sucede, entonces la consecuencia B se pondrá en marcha de forma automática. Su implicación legal ha caído -como toda la tecnología relacionada a Bitcoin- en una zona gris. No se requiere de ningún intermediario de confianza (como una notaría), pues este papel lo adopta el código informático, que asegurará sin dudas el cumplimiento de las condiciones. Por tanto, se reducen tiempo y costes significativamente (Pérez, n.d.).

Las ventajas son obvias, y pueden reducirse a tres palabras: autonomía, seguridad y confianza. Utilizando contratos inteligentes ya no resulta necesario recurrir a un tercero —como un abogado o un notario—, que además de que pueden provocar errores, ocasiona gastos significativos. La *blockchain* es capaz de resguardar la información en una red cifrada que puede consultarse desde cualquier lugar del mundo, por lo que la velocidad y seguridad saltan a la vista.

Con esta nueva tecnología se puede crear una gran cantidad de nuevas aplicaciones para hacer trámites y transacciones hasta ahora difíciles de realizar con las tecnologías existentes, o simplemente mejorar servicios gracias a la descentralización de la *blockchain* y de los contratos inteligentes.

Una de estas aplicaciones de los contratos inteligentes está dada a las subastas. Una subasta es una venta generalmente pública en la que se adjudica una cosa, especialmente bienes o cosas de valor, a la persona que ofrece más dinero por ella. La *blockchain* puede y está cambiando, la manera en la que se hacen las subastas, ya sin necesidad de un subastador o de alguna entidad que haga de mediador. Ya muchas casas de apuestas han actualizado sus políticas, para adaptarse a los nuevos métodos, de hacer subastas.

Los mercados de deuda pública o de bonos soberanos(o del estado) siempre han hecho uso de subastas para hacer sus ventas. Con la *blockchain* se abre una nueva puerta para una forma segura, eficiente y sencilla de efectuar estas subastas.

Específicamente, en el presente trabajo se estudian las subastas a ciegas. Una subasta a ciegas es aquella en la que solo el ofertante sabe el monto de su oferta y nadie más. Él no conoce las ofertas de los demás y viceversa.

La implementación de subastas a ciegas sobre *blockchain* presupone una dificultad, pues toda información que se almacena es pública y verificable por cualquiera que esté conectado a la red de nodos. La solución a esto podría ser el cifrado de las ofertas

que hacen los pujadores. Por tanto, el problema consiste en la selección del algoritmo adecuado para el desarrollo de un sistema de subasta a ciegas.

El problema científico abordado en esta investigación es: analizar algoritmos, criptográficos o no criptográficos, para la realización de subastas a ciegas. Que se enmarca en el objeto de investigación: subastas electrónicas. El **objetivo** general de este trabajo es el diseño e implementación de un conjunto de algoritmos que permita el desarrollo de subastas a ciegas sobre Quorum (Quorum es una *blockchain* basada en Ethereum). Este objetivo delimita el siguiente campo de acción: implementación de subastas a ciegas sobre Quorum.

Para lograr el objetivo general se definen los siguientes objetivos específicos:

1. Identificar las soluciones técnicas y tecnologías que se emplean para el desarrollo de aplicaciones relacionadas con subasta electrónica.

2. Valorar las posibilidades de Quorum como plataforma para el desarrollo de aplicaciones de este tipo.

3. Estudio de Solidity como lenguaje de programación para el desarrollo de contratos inteligentes sobre Quorum.

4. Implementar contratos inteligentes (algoritmos) que permitan efectuar subastas a ciegas.

La memoria escrita está organizada en tres capítulos. En el Capítulo 1 se aborda el tema de las subastas, qué beneficios presenta la *blockchain* para el desarrollo de subastas y una comparación entre algunos tipos de protocolos de subastas a ciegas sobre *blockchain*. En el Capítulo 2 se explica la utilización de la *blockchain* de Quorum para el desarrollo de un sistema de subastas a ciegas. El Capítulo 3 está dedicado a la evaluación de los resultados y mostrar el desempeño del método propuesto. Finalmente, se dan las conclusiones de la investigación, recomendaciones, así como la bibliografía y los anexos necesarios para la mejor comprensión de la propuesta.

Chapter 1

Subastas y blockchain

1.1 Subastas

Una subasta es el proceso de comprar y vender bienes o servicios. Este proceso implica ofrecer artículos para vender, esperar que sean enviadas las ofertas y vender los bienes a la mayor oferta, bajo la supervisión de un subastador (Krishna, 2009).

Por la relevancia del término, se considera importante revisar qué definiciones formales de "subasta" existen:

- Definición de la RAE: 1. f. Venta pública de bienes o alhajas que se hace al mejor postor, y regularmente por mandato y con intervención de un juez u otra autoridad. 2. f. Adjudicación de una contrata, generalmente de servicio público, como la ejecución de una obra, el suministro de provisiones, etc., a quien presenta la propuesta más ventajosa (Real Academia Española, n.d.).

- Economipedia: Una subasta es un procedimiento de venta donde los interesados compiten entre sí para adjudicarse el bien o servicio a ser subastado (Roldán, 2017).

1.1.1 Tipos de subastas

Las subastas pueden clasificarse en diferentes tipos. A continuación se resumen las características de las más conocidas.

- *English Auction* (Subasta Inglesa o ascendente). Este es el tipo de subasta más conocido. Las pujas comienzan con un precio bajo, y se incrementan progresivamente a medida que se solicitan pujas más altas, hasta que se cierra la subasta o no se reciben pujas más altas. A menudo el vendedor fija un precio de reserva por debajo del cual

el artículo no se vende y la subasta se cancela. Permite a un vendedor asegurar el precio más alto para un artículo.

- *Dutch Auction* (Subasta holandesa o descendente). El precio empieza alto y va bajando hasta que algún participante está dispuesto a pagar el precio, y este es el que gana y paga el último precio que se menciona.

- *Blind Auction* (Subasta a ciegas o de sobre cerrado). También conocida en la literatura como *first-price sealed-bid auction (FPSBA)* En este tipo de subasta, todas las ofertas se envían simultáneamente y nadie sabe qué oferta hizo el resto de los participantes. Gana el que mayor oferta hizo y paga esa cantidad al vendedor.

- *Vickrey Auction* (Subasta Vickrey). Conocida también en la literatura en inglés como *sealed-bid second-price auction (SBSPA)* . Es un tipo de subasta de puja sellada, donde los oferentes presentan ofertas por escrito sin conocer la oferta de las otras personas en la subasta, y en la que gana el postor más alto, pero el precio que paga este es la segunda oferta más alta (2017).

- *All-pay auction* (Subasta americana). Es como la subasta inglesa, en este caso todos los postores deben pagar la oferta que hacen, pero solo el que realiza la mejor oferta obtiene el producto.

- *Silent auction* (Subasta Silenciosa). Las pujas se escriben en hojas de papel. Al final de la subasta, la puja más alta se adjudica la subasta. Este tipo de subasta se utiliza frecuentemente en eventos de beneficencia, en los que se subastan muchos objetos simultáneamente, y se "cierra" a una hora predeterminada común a todos los objetos. La subasta es "silenciosa" porque no hay subastador y los pujadores escriben sus pujas en una hoja que usualmente se deja en una mesa cercana al objeto. En las subastas de beneficencia, las hojas usualmente indican una puja inicial mínima, los incrementos que se pueden hacer sobre dicha puja mínima y una cantidad, llamada "puja garantizada" que si se paga se obtiene el objeto de forma inmediata. Otras variaciones de este tipo de puja pueden incluir pujas selladas. El pujador con la puja más alta paga el precio que indicó en su hoja y obtiene el bien (Investopedia.com, n.d.).

- *Reverse auction* (Subasta inversa o reversa). Es un tipo de subasta en la que se invierten los papeles de comprador y el vendedor. En una subasta ordinaria, los compradores compiten para obtener un bien o servicio, ofreciendo precios cada vez más altos. En una subasta inversa, los vendedores compiten para obtener negocio del comprador y los precios suelen disminuir a medida que los vendedores hacen sus ofertas.

- *Candle Auction* (Subasta de velas). Es una variación de la subasta típica inglesa que se hizo popular en los siglos XVII y XVIII. En una subasta de velas, el final de la subasta se indica con el vencimiento de la llama de una vela, que tenía la intención de garantizar que nadie pudiera saber exactamente cuándo terminaría la subasta y hacer una oferta de último segundo. A veces, se utilizaron otros procesos

impredecibles, como una carrera a pie, en lugar de la expiración de una vela (Patten, 1970).

- *Double Auction* (Subasta Doble). Una doble subasta es un proceso de compra y venta de bienes cuando los compradores potenciales y los posibles vendedores presenten simultáneamente sus ofertas, su demanda, los precios de una casa de subastas, y luego un subastador elige algún precio p que equilibra el mercado: todos los vendedores que solicitaron menos de p venden y todos los compradores que pujaron más de p compran a este precio p (Friedman, 1992).

Hay algunos otros tipos de subasta, pero mucho menos conocidas. Como fue explicado en la Introducción, la presente investigación se enfoca precisamente en subastas a ciegas.

1.1.2 Mercado de deuda

El mercado de deuda o bonos es donde se emiten y negocian los títulos de deuda, cuando los participantes no están en condiciones o no desean pedir préstamos o créditos a la banca. En él participan el Gobierno Federal, los gobiernos estatales o locales y las empresas paraestatales o privadas que necesitan financiamiento, ya sea para realizar un proyecto de inversión o para mantener sus propias actividades. Una parte de este mercado se conoce como mercado del dinero, que es donde se intercambian los bonos que por su corto plazo, liquidez y alta seguridad se pueden considerar sustitutos del dinero.

El mercado de deuda también se conoce con otros nombres, dependiendo del tipo de instrumentos de deuda negociado. Por ejemplo, si en el mercado se negocian principalmente instrumentos de deuda que pagan una tasa fija, entonces se denomina mercado de renta fija, mercado de renta variable, mercado de deuda internacional, de deuda pública, etc. En términos generales, para que una persona pueda comprar o vender títulos de deuda es necesario que acuda a un banco o a una casa de bolsa para que dichas instituciones puedan efectuar las transacciones necesarias a nombre de esta persona (Banco de México, n.d.).

La presente investigación se enfoca específicamente en el mercado de deuda pública o el mercado de bonos del Estado. Un bono del Estado es un tipo de inversión basada en deuda, en la cual se le presta dinero a un gobierno a cambio de una tasa de interés acordada. Los gobiernos los utilizan para generar fondos que pueden gastar en nuevos proyectos o infraestructura, mientras que los inversores pueden usarlos a fin de que se les pague un retorno establecido en intervalos periódicos.

Cuando se compra un bono del Estado, se le presta al Gobierno una cantidad acordada de dinero durante un período también acordado. A cambio de esto, el Gobierno devuelve el dinero con un nivel establecido de interés de forma periódica,

lo que se conoce como cupón. De esta manera, los bonos conforman un activo de ingreso fijo.

Cuando el bono venza, se devolverá la inversión original (denominada principal). El día en el que se recibe el valor de capital adeudado se conoce como la fecha de vencimiento. Diferentes bonos tienen distintas fechas de vencimiento; por ejemplo, se puede comprar un bono que vence en menos de un año o uno que vence en 30 años o más.

Algunos inversores dicen que los bonos del Estado son inversiones libres de riesgo. Dado que un gobierno siempre puede imprimir más dinero para saldar sus deudas, teóricamente hablando, siempre se devolverá tu dinero cuando venza el bono.

En la realidad, esto es mucho más complicado. En primer lugar, los gobiernos no siempre pueden producir más capital. Incluso si es que pudieran hacerlo, esto no evita que incumplan el pago de los préstamos. No obstante, aparte del riesgo crediticio, existen otros posibles problemas de los cuales preocuparse con los bonos del Estado, por ejemplo, el riesgo de las tasas de interés, la inflación y las divisas (ig.com, n.d.).

1.1.3 Mercado de Deuda Pública en Cuba

Cuando en la economía doméstica los gastos superan a los ingresos no queda más remedio que pedir prestado, de lo contrario no se pudiera pagar lo que necesitamos o debemos. En otras palabras, se tendría que disminuir o aplazar las compras. Al presupuesto del Estado le sucede exactamente lo mismo y sus ausencias de dinero se llaman déficit fiscal. La deuda pública es la suma del déficit fiscal del año, más las garantías que se activen en esos 12 meses (porque el presupuesto del Estado es garante de determinadas operaciones económicas como inversiones en sectores priorizados) y las amortizaciones de deudas anteriores provenientes de los bonos que se colocaron en lo años anteriores.

Pero, ¿cómo se financia el déficit presupuestario? Esto ocurre mediante bonos soberanos de la República de Cuba que emite el Ministerio de Finanzas y Precios (MFP) y que hasta el momento han sido adquiridos por el sistema bancario nacional.

El Banco de Inversiones es el encargado de colocar, emitir y registrar la emisión. El MFP le solicita una emisión de bonos y este lo coloca en el sistema bancario, y cada banco comercial (Banco de Crédito y Comercio, Popular de Ahorro y Metropolitano) adquiere el monto que puede asumir. En caso de que los bancos no tengan suficiente liquidez para cubrir totalmente el bono, interviene el Banco Central de Cuba (BCC) con la emisión de nuevos billetes.

Hasta 2013, el déficit se financiaba solo de esta manera (con la emisión de monedas por el BCC), pero debido a la inflación que provoca el exceso de dinero en circulación fueron instrumentados los bonos soberanos y siempre se busca que la participación

del BCC sea en última instancia, porque es un dinero que se pone en circulación sin respaldo productivo (Tamayo & Ferrer, 2021).

A pesar de que ya desde 2014 estan siendo emitidos los títulos de deuda soberana y de que legalmente es posible transar estos valores, la inexistencia de incentivos que promuevan la demanda de los títulos públicos y de condiciones básicas institucionales y de infraestructura del mercado han condicionado que las primeras colocaciones de bonos se hayan realizado mediante asignaciones administrativas. Lo anterior demuestra que no resulta suficiente emitir los títulos, si ello no se acompaña de transformaciones y reformas institucionales, estructurales, normativas y regulatorias que permitan ordenar, organizar e incentivar el funcionamiento de un Mercado de Deuda Pública. El sistema empleado hasta ahora esta muy lejos de ser el ideal. Para profundizar en este tema, se puede revisar (Barceló, 2017).

Por parte del Instituto de Criptografía se desarrolla para el Banco Central de Cuba una plataforma para el Mercado de Deuda Pública de Cuba. Una de las propuestas a utilizar en esta plataforma es una variación de la subasta holandesa (como se utiliza en casi todo el mundo), pero en este caso, a ciegas sobre una *blockchain*, específicamente la *blockchain* de Quorum (Figura 1.1).

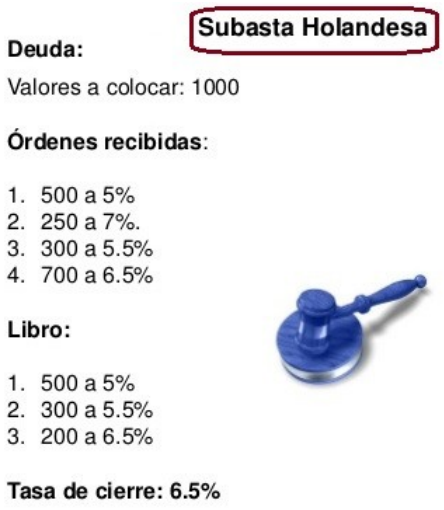

Figure 1.1: Ejemplo de subasta holandesa en el mercado de deuda

1.2 Blockchain

Las subastas tradicionales usualmente requieren una tercera persona, ya sea un subastador o una casa de subastas que maneje el proceso completo de subasta, lo cual puede llegar a tener muy altos impuestos por comisiones. También sufren de un punto de fallo, los subastadores pueden en ocasiones tener malas intenciones (Wu

et al., 2019). En este contexto, *blockchain* surge como una plataforma descentralizada que puede ser empleada para aplicaciones de subastas en línea confiables. En 2018, por primera vez en el mundo una colección de arte valuada en varios millones de dólares perteneciente a Andy Warhol fue tokenizada y subastada satisfactoriamente, usando la *blockchain* de Ethereum (Emem, 2018; Wood, 2021). Es también conocido que una de las mayores casas de subastas (e.g., Sotheby's and Christie's) está activamente trabajando en aplicar *blockchain* para subastas seguras y confiables (Neuendorf, 2018).

La tecnología *blockchain* elimina efectivamente los intermediarios, por lo tanto, reduce los costos de transacción y asegura la confianza entre las partes interesadas en la subasta. En general, la tecnología *blockchain* puede mejorar las subastas en los siguientes aspectos (Shi et al., 2021):

- Inmutabilidad de las transacciones de la subasta. Cada transacción ejecutada en la *blockchain* es pública, verificable e inmutable. Esto significa que la *blockchain* puede ser empleada como dispositivo de certificado de auditoría que previene que los participantes hagan trampa durante la subasta. El ganador puede usar la *blockchain* como prueba de la transacción.

- Automatización del proceso de subasta. Un contrato inteligente automatiza el proceso de la subasta en la *blockchain*. Casi toda la lógica de la subasta puede ser predefinida en contratos inteligentes para facilitar el intercambio de bienes y servicios, así como el pago de los tokens.

- Descentralización del manejo de la subasta. No hay necesidad de designar una tercera persona como subastador, que asegure la confiabilidad. Las tradicionales subastas centralizadas pueden ser muy costosas y sujeta a posibles subastadores tramposos; casas de subastas típicamente cargan del 8 al 20% como comisión.

- Flexibilidad en el pago de la subasta. Las criptomonedas existentes en la *blockchain* pueden mejorar la seguridad y flexibilidad del pago de la subasta. Al mismo tiempo, un sistema de pago descentralizado no necesita de intermediarios financieros, haciendo las transacciones más convenientes y menos costosas.

1.2.1 Quorum

Blockchain es de los términos más usados en el mundo de la tecnología en estos días. Mientras esta tecnología es el elemento núcleo de las criptomonedas que están llegando a los mercados financieros, su utilidad no es limitada a las criptos y tiene muchos más casos de uso. Con el pasar de los años, diferentes plataformas *blockchain* han sido desarrolladas con sus propios mecanismos de consenso [1] y métodos de

[1]Por ser la blockchain un libro contable distribuido y no centralizado, se requiere establecer un mecanismo para que todos los participantes en la red estén de acuerdo en el contenido sobre el contenido del libro contable. Esto es lo que se conoce como un mecanismo de consenso.

encriptación.

Una de estas plataformas es la *blockchain* de Quorum, la cual ha ganado popularidad en los últimos tiempos, como resultado de la larga lista de casos de uso que Quorum provee a sus usuarios. La red de Quorum está basada en una bifurcación[2] de la *blockchain* de Ethereum. Es un protocolo *blockchain* de código abierto especialmente diseñado para usar en redes *blockchain* privadas. Algo a destacar de esta red es su nueva característica, llamada *"private transaction identifier"*, que asegura la privacidad de los datos. El objetivo de construir Quorum es utilizar la tecnología existente tanto como sea posible. Por lo tanto, incluso si la red Ethereum se somete a diferentes actualizaciones en el futuro, habría pocos o ningún cambio en la *blockchain* de Quorum para mantener la sincronización entre estas redes.e

1.2.2 Subastas a ciegas sobre *blockchain*

Las subastas sobre *blockchain* han sido una área centro de muchas investigaciones en los últimos años, razón por la cual ya se han investigado algunos protocolos de subastas a ciegas sobre *blockchain*. Algunos de ellos son:

- Kosba et al. (2016) propusieron Hawk, una combinación de la privacidad de Zcash[3] con la programabilidad de Ethereum. Contratos inteligentes que preservan la privacidad fueron empleados para manejar la privacidad de las transacciones en la *blockchain* de Ethereum. Ellos se enfocan en presentar un *framework* que puede simultáneamente admitir varias aplicaciones como subasta de puja sellada, el juego de piedra, papel y tijera, aplicación de *crowdfunding*[4] e intercambiar instrumentos financieros. La principal limitación de Hawk es que depende de un administrador al que se le confía explícitamente que no filtre entradas secretas a los contratos inteligentes; dependiendo de esta manera de una tercera persona.

- Blass and Kerschbaum (2018) presentan Strain, un protocolo para construir una subasta de puja sellada sobre una *blockchain*, preservando la privacidad de las ofertas contra partes maliciosas. Se utiliza un tablón de anuncios para publicar la oferta ganadora que es determinada comparándolas por pares. Dos conocimientos de prueba cero (*zero-knowledge proofs (ZKP)*) diferentes son empleadas; una asegura que los participantes usan sus ofertas originales, bajo compromiso, y la otra ZKP asegura que el subastador declare el ganador, sin ninguna manipulación. Los participantes

[2]Se considera una bifurcación (en inglés *fork*) al desarrollo de un proyecto informático tomando como base un código fuente que ya existe e iniciar un desarrollo independiente, creando asi software distinto y separado.

[3]Zcash es una criptomoneda que utiliza criptografía aplicada avanzada para proporcionar una mayor privacidad a través de direcciones protegidas. Es la primera aplicación práctica de zk-SNARK, un tipo específico de prueba de conocimiento cero. (Zcash, 2022).

[4]Es un mecanismo colaborativo de financiación de proyectos, desarrollado sobre la base de las nuevas tecnologías.

maliciosos son sancionados por abrir su compromiso, ya que su clave privada está parcialmente compartida entre todos los participantes a través de un proceso de generación distribuida de claves.

- Galal and Youssef (2018b) propusieron una subasta de puja sellada en la *blockchain* Ethereum con contrato inteligente y pruebas de conocimiento cero (*zero-knowledge proofs*). Ofertantes envían sus pujas al contrato inteligente usando *Pedersen commitment* (Pedersen, 1991). Los compromisos (commitments) son secretamente revelados al subastador vía un esquema *Public Key Encryption (PKE).* Después de declarar el ganador, por cada oferta perdedora, el subastador tiene que participar en un conjunto de protocolos interactivos de compromiso-desafío-verificación para demostrar que la puja ganadora es mayor que las pujas perdedoras y, como consecuencia, la complejidad de la interacción depende del número de ofertantes. Luego (Galal & Youssef, 2018a) mejoraron este protocolo presentando un contrato inteligente con una verificablemente concisa *ZKP* que permite una sola prueba de verificación para todo el proceso de la subasta. Similar a Zcash, ellos emplearon *Multi-Party Computation (MPC)*[5] entre el subastador y pujantes. El esquema propuesto implementa la validez, imparcialidad y secreto de las transacciones en la subasta. Sin embargo, depende de un subastador externo.

- Sánchez (2020) propone un contrato inteligente privado y verificable, se introduce como una combinación de *Multi-Party Computation* segura y *Proof-carrying code*[6] enfocada principalmente en garantizar la correctitud, privacidad y verificabilidad para contratos inteligentes en la *blockchain*. Este enfoque puede ser utilizado para varios tipos de aplicaciones tales como *crowdfundings* privados y verificables, fondos de inversión y subastas dobles para intercambios descentralizados.

- Sharma et al. (2021) introdujeron un protocolo genérico para comercio anónimo de manera justa usando solamente estándar construcción de bloques criptográficos. Las propiedades confidencialidad y anonimato son alcanzadas utilizando *Designated Verifier Ring Signature (DVRS)* y la transparencia y auditabilidad de la plataforma *blockchain* se aprovecha para realizar el proceso de subasta públicamente verificable. Se asume la existencia de una *blockchain* que permita transacciones anónimas y confidenciales. También se analiza la eficiencia de emplear primitivas criptográficas en la *blockchain* de Ethereum e investigar la complejidad y vulnerabilidades que un entorno *blockchain* podría introducir durante la implementación.

- Li and Xue (2021) proponen un esquema de subasta electrónica de puja sellada basada en *blockchain* con algoritmo de compromiso, contratos inteligentes y *zero-*

[5]El objetivo de los protocolos de computación multiparte (MPC) es permitir a un conjunto de participantes calcular el resultado de una función de sus entradas privadas, revelando el mínimo de información

[6]Es un mecanismo de software que permite que un sistema host verifique las propiedades de una aplicación a través de una prueba formal que acompaña al código ejecutable de la aplicación.

knowledge proof (ZKP) para proteger la fuga de información de las pujas y verificar el resultado de la subasta con todos los ofertantes anónimamente, que implementa satisfactoriamente la seguridad y justicia de la subasta sin necesidad de un subastador externo. El esquema propuesto presenta limitaciones en cuanto al tiempo de corrida. El tiempo de ejecución de dos de la fases, la fase abierta y fase final, está determinado por el número de ofertantes. Esto significa que la subasta se convertiría en un trabajo que consume mucho tiempo en el caso de ser usado en plataformas abiertas como Internet.

De acuerdo a la revisión de la literatura hecha anteriormente, los estudios relacionados tienen algunos defectos para los esquemas de subasta de puja-sellada. Y en especial para el que se quiere implementar, una variación de la subasta holandesa con subasta a ciegas. En opinión del autor de esta investigación, tales dificultades serían:

1 - Riesgo del modo de transacción centralizado: como se discutió anteriormente, en subastas tradicionales todas las transacciones están en control de los subastadores. Ha sido demostrado que subastadores no confiables pueden causar filtración del precio de la puja y alterar el resultado de la subasta. En los estudios relacionados, la mayoría de subastas electrónicas, incluso las basadas en *blockchain*, todavía utilizan subastadores para controlar las transacciones. Por lo tanto, la equidad y fiabilidad de las subastas no están perfectamente garantizadas.

2 - Ocultación de precios incompleta: ocultar el precio de las pujas es el núcleo de las subastas de puja sellada. En la mayoría de los estudios relacionados, el precio de la puja es protegido por encriptación. Sin embargo, en la fase abierta, el precio de la puja es desencriptado y directamente revelado por verificación, lo cual puede causar filtración de precios.

3 - Solamente un ganador: todos los esquemas vistos basan sus algoritmos para encontrar un único ganador, la puja más alta. Sin embargo, para el problema en cuestión es posible y casi seguro que haya varios ganadores. Es necesario buscar una estrategia factible para este caso.

4 - Posibilidad de solamente una oferta por participante: todos los protocolos anteriores solamente permiten una única puja, por cada participante en la subasta, para el problema actual, es necesario que se puedan realizar varias ofertas, e incluso, de ser posible, retirar ofertas no deseadas.

Chapter 2

Propuesta

En este capítulo se describirá el algoritmo implementado para ejecutar una variante de subasta holandesa a ciegas sobre *blockchain*, en este caso para una red *blockchain* basada en Ethereum, programada en el lenguaje de programación solidity. Este algoritmo se hace con la finalidad de facilitar y realizar de manera segura la venta de bonos soberanos para el mercado de deuda pública.

2.1 Ventajas de la propuesta

Los esquemas y protocolos para subastas a ciegas de la literatura consultada no cumplen con los requerimientos necesarios para resolver el problema en cuestión (subasta a ciegas de bonos soberanos) o se hace difícil su adaptación al problema a resolver. Dado esto, se propone una solución que se adapta a las necesidades del problema y que además es escalable, segura y confiable.

Ventajas

1. No es necesario que los participantes de la subasta se registren en esta. Para participar solamente necesitan hacer dos transacciones, la oferta y luego la revelación de la oferta.

2. Posibilidad de hacer varias ofertas. Cada participante o postor de la subasta puede realizar cuantas ofertas estime convenientes, no tiene limitantes en cuanto al número de ofertas.

3. Permite retractar o cancelar ofertas. En la fase de revelación el usuario puede decidir no proceder con una oferta.

4. Admite múltiples ganadores. Los bonos se reparten entre los ganadores de la subasta, que en la mayoría de los casos será más de uno.

A pesar de las ventajas de la propuesta, también se presentan algunas desventajas, las cuales se describen en el siguiente apartado. Desventajas:

1. Las ofertas son reveladas. Las ofertas, si bien en la fase de ofertas son desconocidas, es necesario revelar la información real de la oferta para comprobar su validez.

2. Posibilidad de colisión. Al utilizar un algoritmo de *hash* para codificar las ofertas, existe la posibilidad (aunque bastante poco probable) de que la información de dos ofertas diferentes, den el mismo *hash*.

Teniendo en consideración las ventajas y desventajas anteriormente mencionadas, el autor cree que las desventajas no representan un problema tan significativo, dado que la seguridad y confiabilidad de la subasta se mantienen en un nivel alto. Y, por el contrario, tiene ventajas muy beneficiosas para resolver la problemática, incluso alguna de ellas no vistas en los esquemas de subasta que se han investigado con anterioridad.

2.2 Condiciones iniciales

Para mejor entendimiento del lector y lograr ser más específicos en algunos asuntos, la propuesta estará enfocada a la *blockchain* de Ethereum, y todo lo que se refiera a la implementación e interacción del contrato inteligente a partir de ahora, va a estar enfocada a esta *blockchain* en particular. A pesar de esto, la propuesta implementada se podrá utilizar en la red de Quorum o en otras redes basadas en Ethereum sin ninguna o muy pocas modificaciones.

Para participar en la subasta como postor solo es necesario tener una cuenta en la *blockchain* de Ethereum, la aplicación o extensión Metamask (o alguna otra que permita interactuar con contratos inteligentes) y una cantidad de ether suficiente para pagar la comisión de gas de las transacciones.

2.3 Proceso de Subasta

El proceso de la subasta va a estar compuesto por cinco fases principales: despliegue, ofertas, revelación, verificación y finalización.

Para mejor entendimiento, a partir de este momento al que oferta los bienes a subastar (en este caso los bonos) se le llamara beneficiario (para el problema en cuestión sería el gobierno), al que despliega el contrato inteligente en la *blockchain* le llamaremos subastador (a pesar de que no cumple el mismo objetivo ni funciona como los subastadores tradicionales). El beneficiario y el subastador pueden ser el mismo

o personas diferentes. Por último, a los participantes en la subasta, se les llamará postores. Beneficiario, subastador y postores van a ser los usuarios de la subasta para mayor comodidad.

2.3.1 Desplegar contrato

El contrato inteligente puede ser desplegado a la *blockchain* por el propietario de lo que se oferta o por una tercera persona que haga función de subastador. Esta persona no tiene ningún poder, ni ningún privilegio en el contrato inteligente, ni tampoco tiene acceso a retirar los activos del contrato. La única función del subastador es la de poner los parámetros iniciales de esta, dígase:

1. boneToSale: valor total de los bonos que se quieren vender
2. biddingTime: tiempo de duración de la fase de ofertas
3. revealTime: tiempo de duración de la fase de revelación de ofertas
4. beneficiaryAddress: dirección donde se quiere recibir el pago de los bonos vendidos.

Cada vez que se quiere hacer una nueva venta de bonos, es necesario volver a desplegar el contrato inteligente.

2.3.2 Fase de ofertas

Cada postor, puede realizar cuantas ofertas estime convenientes. Dado que lo que se oferta en la subasta son bonos, que como se explicó anteriormente, son una especie de préstamos por tiempo definido que se le hace al gobierno; cada puja está compuesta de dos partes, la cantidad que el postor está dispuesto a prestar y cual sería el interés a cobrar por ese préstamo en porciento.

Dada la necesidad de que las ofertas no sean conocidas por los demás postores, luego de escoger las condiciones de la oferta a efectuar, en vez de enviar los datos reales de esta, el postor codifica esa información (cantidad y porciento) en conjunto con una clave privada solo conocida por él, y está codificación es lo que se envía al contrato inteligente. Es importante destacar que la cantidad enviada en la transacción, en conjunto con la codificación de la oferta, no tiene que ser necesariamente la cantidad exacta de la oferta enviada, esta cantidad se deposita en el contrato inteligente y se añade al saldo de esa dirección en el contrato, para un posible uso posterior de este en ofertas venideras.

Para codificar la oferta se hace uso de la función de solidity: *keccak256*, la cual es un algoritmo o función de *hash* que toma como entrada un conjunto de datos y

devuelve un valor de longitud fija, 32 bytes. Esta función es una de las más utilizadas en la programación de contratos inteligentes, ya que permite la creación de *hash* de datos que no pueden ser revertidos a su valor original, es decir, que no se puede obtener la información original de un *hash* de esta función. Esta función es empleada para codificar la oferta, por el hecho de que al ser una función de hash, no se puede obtener la información original de la oferta, por lo que no se puede saber la cantidad y el porciento de la oferta que se realizó.

Luego de desplegado el contrato inteligente hay un tiempo hábil para mandar ofertas, luego de ese tiempo no serán recibidas más ofertas. Y da inicio a la fase de revelación de ofertas.

2.3.3 Revelación de ofertas

Cada postor que realizó ofertas debe enviar al contrato la información de las ofertas que quiere revelar. Es decir, tiene que enviar tres arreglos, que van a representar valor, porciento y clave secreta de las ofertas, respectivamente. Es necesario que los tamaños de los arreglos sean igual a la cantidad de ofertas enviadas, de lo contrario, no serán analizados, y será necesario una nueva transacción con la información completa. La posición del arreglo significa el número de la oferta, ordenada por tiempo de recepción de la oferta por el contrato inteligente. El contrato inteligente se encarga de codificar la información suministrada, con la misma función *keccak256* que fue utilizada por el postor y la comprueba con la codificación enviada por este en la fase de ofertas. Si las dos codificaciones coinciden exactamente, quiere decir, que los datos de la oferta son los mismos que el postor eligió en la fase de ofertas y la oferta se considera válida. Para que la oferta sea totalmente válida, es necesario que el valor depositado hasta ese momento en el contrato sea mayor que el valor de la puja de esa transacción. A cada oferta válida se le asigna un identificador único (id), por orden de revelación (las ofertas que primero se revelan tienen un menor id), que posteriormente será usado.

Si la oferta no es válida, ya sea por no coincidir los valores *hash* de las codificaciones o por no tener suficiente dinero disponible para ejecutar la oferta, se anula la oferta, sin embargo, el dinero depositado en esa transacción queda disponible para próximas ofertas, aunque también disponible para retirar en cualquier momento posterior. Cuando se solicita un retiro (*withdraw*) del contrato, si la dirección que lo solicita tiene algún fondo disponible para retiro, pues recibe el reembolso de todo lo disponible en el contrato.

Es necesario destacar que la revelación de las ofertas ocurre solamente una única vez por cada dirección, es decir, si una oferta es considerada válida o no válida (por alguna de las razones vistas anteriormente) pues no hay forma de cambiar ese veredicto. Por esto es necesario tener sumo cuidado con la información que se envía al contrato inteligente tanto en la fase de ofertas, como en la fase de revelación, ya

que una vez que se envía, no hay forma de cambiarla.

2.3.4 Verificación y publicación de los ganadores

En esta fase, se comprueban cuáles son los ganadores de la subasta, y se publican los resultados. Para esto, ya se tienen las ofertas válidas, determinadas por la fase de revelación, estas se ordenan crecientemente por el porciento de interés que ofrecen, es decir las que menor porciento de interés tiene el préstamo van primero, dado que son las más convenientes para el deudor. En caso de tener el mismo porciento se desempata por la oferta con menor id.

Luego de ordenadas las ofertas, se comienzan a aceptar ofertas hasta lograr la cantidad total que se necesita para cubrir el valor de los bonos ofertados. Para esto, se comienza con la oferta con menor porciento de interés, y se acepta la oferta, es decir, disminuye la cantidad de bonos ofertados. Luego se pasa a la siguiente oferta, y se acepta la oferta, y así sucesivamente hasta que se agoten los bonos ofertados o hasta agotar todas las ofertas válidas.

Se paga un único porciento de interés a todos los ganadores de la subasta, el cual es el porciento de interés de la última oferta aceptada, es decir, la oferta aceptada con mayor porciento de interés.

En caso de que para un porciento de interés se tengan varias ofertas con ese mismo porciento, se acepta primero la oferta con menor id. Por consiguiente, si en el último porciento de interés aceptado se tienen varias ofertas, el desempate está dado por el tiempo de revelación de la oferta (que es lo que determina el id de la oferta). Esto estimula a los postores a revelar sus ofertas lo más pronto posible, para tener una mayor probabilidad de estar entre los ganadores la subasta.

En caso de que la última oferta aceptada sobrepase la cantidad de bonos ofertados, se acepta la oferta parcialmente, es decir, se acepta solamente la cantidad que se necesita para cubrir el valor de los bonos ofertados.

Luego de que ya se tienen las ofertas ganadoras, el dinero bloqueado de las ofertas restantes (si quedara alguna) es puesto a disposición de sus respectivos postores, para que puedan retirarlo.

2.3.5 Finalización

Esta fase está estrechamente ligada con la anterior, y se hacen una a continuación de la otra. Pero para mejor entendimiento se puso en una fase aparte. En esta fase, se publican los resultados de la subasta, es decir, se publican las ofertas ganadoras (dirección, porciento y cantidad a prestar de cada una). Además, se publica el porciento de interés que se le pagará a los ganadores. Y seguidamente se transfiere

el dinero de los postores ganadores, en este caso convertidos en prestamistas, a la dirección del beneficiario de la subasta, que sería el deudor de los préstamos.

Por último el contrato inteligente activa una bandera que indica que la subasta ya ha finalizado, y que no se pueden hacer más ofertas, ni revelaciones, ni nada relacionado con la subasta, solamente retiros del dinero de los postores que tengan dinero disponible.

El contrato fue diseñado para subasta de bonos soberanos, pero puede ser empleado para cualquier otra subasta del mercado de deuda. Y también puede ser fácilmente adaptado para realizar otro tipo de subastas.

2.4 Seguridad

2.4.1 Obligación de pago del beneficiario

En la implementación realizada, se asume la fiabilidad del beneficiario de la subasta, a la postre deudor. No se llega a implementar ninguna medida para asegurar el pago a los prestamistas, dado que depende mucho de las condiciones de la subasta y se escapa un poco del alcance que pudiera tener el contrato inteligente, sin embargo, se propondrán algunas propuestas, que pudieran ser empleadas para garantizar o al menos mejorar la confianza de los postores hacia el beneficiario.

1. El beneficiario tenga que depositar algún activo como colateral en el contrato inteligente, que sea igual al valor total o al menos a los intereses a pagar por los bonos que se ofertan. De esta manera, si el beneficiario no paga, el contrato inteligente puede pagar a los prestamistas con el dinero depositado por el beneficiario.

2. Liberación de fondos en partes. En lugar de liberar el dinero de los prestamistas al final de la subasta, se puede liberar el dinero de los prestamistas en partes, es decir, liberar el dinero de los prestamistas en partes, proporcionalmente a los intereses que se van pagando. De esta manera, si el beneficiario no paga, el contrato inteligente puede pagar a los prestamistas con el dinero congelado en el contrato inteligente.

3. Tokenizar los bonos. En lugar de ofertar los bonos, se puede ofertar tokens que representen los bonos. De esta manera, estos tokens servirían como garantía de que el beneficiario pagará los intereses de los bonos. Quizás hasta se podrían legalizar estos tokens, de manera de que se pudieran incurrir en acciones legales en caso de que el beneficiario no pague los intereses de los bonos o la devolución total del préstamo luego del vencimiento de este. Estos tokens pudieran también servir para que los prestamistas puedan vender sus tokens en

el mercado secundario, y así recuperar parte de su dinero invertido. Creando así un mercado secundario para los bonos, que es algo que no existe en el mercado de deuda cubano actualmente y que fomentaría la liquidez de los bonos y el interés de los inversores en estos.

El autor opina que cada una de estas opciones puede ser viable en algún caso particular, sin embargo, piensa que la mejor y más conveniente opción es la tercera, ya que es la que mejor garantiza el pago de los intereses y la devolución del préstamo, y además, tiene múltiples beneficios adicionales, para los prestamistas y el mercado de deuda cubano en general.

2.4.2 Keccak256

Keccak está basada en una novedosa propuesta llamada construcción de esponja. La construcción de esponja está basada en una amplia función aleatoria o permutación aleatoria, y permite la entrada ("absorber" metafóricamente en terminología de esponja) de cualquier cantidad de datos, y la salida ("exprimir") cualquier cantidad de datos, mientras actúa como una función pseudoaleatoria con respecto a todas las entradas anteriores. Esto provoca una gran flexibilidad (Bertoni et al., 2007).

El algoritmo Keccak es el trabajo de Guido Bertoni, Joan Daemen, Michael Peeters y Gilles Van Assche. Está basado en los diseños de hash PANAMA y RadioGatún. En el año 2006 el *National Institute of Standards and Technology* (NIST) organizó una nueva edición de la competición para la creación de una nueva función para los estándares de Secure Hash Algorithm (SHA), el SHA-3. Al no existir un ataque significativo demostrado en SHA-2, el nuevo estándar SHA-3 no lo reemplazará. La NIST ha mencionado que debido a exitosos ataques a los estándares MD5, SHA-0 y SHA-1, es necesario una alternativa llamada SHA-3 (Stevens et al., 2017).

Las admisiones de proyectos comenzaron en el año 2008. Keccak fue aceptado como uno de los 51 candidatos. En julio de 2009, 14 algoritmos pasaron a la segunda ronda y Keccak avanzó a la ronda final en diciembre de 2010. Durante el periodo de la competición, se les permitió a los participantes corregir problemas descubiertos en sus algoritmos. Keccak hizo algunos cambios como el número de rondas, se amplió de $12-\ell$ a $12-2\ell$. En octubre de 2012 fue seleccionado como el ganador de la competición. En el año 2014 la NIST publicó la documentación técnica del algoritmo y fue aprobado en agosto de 2015, para así convertirse en el nuevo estándar SHA-3 (NIST, 2015).

En el caso de Bitcoin, el uso del algoritmo de hash SHA256 está bastante extendido, además es utilizado en multitud de implementación de funcionalidades de la cadena de bloques de Bitcoin. En cambio, Ethereum hace uso del algoritmo de hash Keccak- 256, aprobado por la NIST en agosto del 2015, convirtiéndose en el nuevo estándar SHA-3. Cabe destacar que debido a que la aprobación por parte

del NIST de este algoritmo fue más tardía que el desarrollo de Ethereum, el estándar final de SHA-3 adoptado por la NIST hace referencia al estándar "FIPS-202 SHA-3", el cual sufrió ligeros cambios en sus parámetros a diferencia del algoritmo Keccak-256 implementado en Ethereum.

```
Keccak256("")
= c5d2460186f7233c927e7db2dcc703c0e500b653ca82273b7bfad8045d85a470

SHA3("")
= a7ffc6f8bf1ed76651c14756a061d662f580ff4de43b49fa82d80a4b80f8434a
```

Como se produjeron ligeros cambios, las salidas de estos algoritmos no son iguales entre sí en el caso de que haya entradas iguales. Para comprobarlo, como se hace en el ejemplo anterior, se puede observar los valores en ambos casos de uno entrada vacía, lo que da como resultado dos salidas distintas. Por lo tanto, a pesar de que en múltiples papers y documentos de Ethereum se haga referencia a SHA-3, realmente se quiere hacer referencia al uso de Keccak-256 (Escarramán, 2022).

Chapter 3

Experimentación y Resultados

En este capítulo se explica cómo se desarrollaron los experimentos realizados, los elementos que formaron parte de los mismos, además se enuncia y justifica algunas de las decisiones tomadas en la implementación del algoritmo, y por último se analizan los resultados obtenidos, y si es factible o no la propuesta para el problema que se quiere resolver.

3.1 Detalles de Implementación

En la fase de revelación de ofertas, a medida que se van verificando las ofertas, a la par se van construyendo dos arreglos con las ofertas válidas. Uno de estos arreglos *bidsRevealed* contiene *<value, percent, biderAddress>* (valor, porciento y dirección del postor) de cada oferta. Y el otro arreglo *bidsRevealedPercentID* solo contendrá *<percent, id>* (porciento e identificador numérico) de cada oferta válida, cabe destacar que este *id* es la posición del arreglo donde se encuentra la oferta, esta información será utilizada más adelante.

Para determinar los ganadores de la subasta, es necesario que seleccionar las mejores ofertas, para el problema actual que es una variación de la subasta holandesa, pues las ofertas que ofrecen menores porcientos de interés, serán las ganadoras de esta. Para escoger estas ofertas, ordenamos el arreglo *bidsRevealedPercentID*, donde el porciento va a ser el valor por el que se va a hacer el ordenamiento de forma ascendente (de menor a mayor valor), en caso de dos ofertas con igual porciento se definirá su orden en dependencia de la que menor *id* tenga.

Para ordenar el arreglo se hace uso del algoritmo Quick Sort. ¿Por qué se usó Quick Sort y no otro algoritmo de ordenamiento? A pesar de que este algoritmo puede llegar a tener en su peor caso costo computacional $O(N^2)$, en el caso promedio se comporta como un algoritmo cuasi lineal $O(NlogN)$. No se empleo el *merge sort* por la complejidad del lenguaje Solidity en el manejo de la memoria y las variables a

utilizar, que se deben a las particularidades de la blockchain como nueva tecnología. El *merge sort* necesita de un arreglo adicional para ejecutarse y generaría costo computacional adicional, por esta razón no es conveniente su uso. Otra posible opción era el Heap Sort (que no necesita memora adicional), pero se prefirió el empleo del Quick Sort, por la sencillez de su implementación.

3.2 Experimentación

3.2.1 Tecnologías Utilizadas

El contrato inteligente que procesa la subasta fue programado en el lenguaje Solidity v0.8.4 (Solidity, n.d.). Y para las pruebas realizadas al contrato se utilizó Python 3.8.10 (Foundation, 2021) en conjunto con Brownie v1.19.2, este último es un framework basado en python para desarrollar y testear contratos inteligentes dirigidos a ejecutarse en la Máquina Virtual de Ethereum. Además se empleó Ganache v7.5 (Truffle, n.d.), que es una implementación local de la blockchain de Ethereum, que permite llevar a cabo pruebas de contratos inteligentes sin necesidad de tener una red real de Ethereum.

3.2.2 Gas

Uno de los conceptos más importantes en el mundo de Ethereum es el Gas. El Gas en Ethereum es una unidad de medida utilizada para medir el trabajo realizado por Ethereum para realizar transacciones o cualquier interacción dentro de la red.

Una forma sencilla ver que es el Gas en Ethereum sería la siguiente analogía: se quiere viajar de Madrid a Barcelona, el viaje se hará en coche. Se sabe de antemano que son 500 km de distancia y que el coche gasta 1 litro de gasolina cada 10 km (para hacer simple el cálculo), así que se necesitará 50 litros de gasolina para llegar al destino. Además, también se sabe que el litro de gasolina cuesta entre 1 € y 1,5 € dependiendo de la gasolinera donde se detenga a repostar.

Esto es lo mismo que pasa en Ethereum. Por un lado, cada tarea en Ethereum tiene un coste específico y no variable estipulado en Gas, lo que es equivalente al litro de gasolina que gasta el auto por cada 10 Km. Por supuesto, las operaciones en Ethereum están formadas por distintas funciones más pequeñas, cada una de ellas con un valor de Gas (o consumo de gasolina) específico y su sumatoria es lo que dirá el valor final en Gas de dicha operación (el total de gasolina a gastar para hacer nuestro viaje). Así solo nos queda una cosa, ¿Cuánto se pagará por ese Gas para poder llevar a cabo la operación en Ethereum?

En la analogía anterior la gasolina varía entre 1 y 1,5 €, se puede escoger donde repostar y pagar lo menos posible para adquirir los 50 litros de gasolina que se necesita

para el viaje. Lo mismo pasa en Ethereum, el Gas tiene un precio en ether que es dado por la demanda y oferta de operaciones en Ethereum. Es decir, el precio del Gas en ether es variable, aunque en este caso se puede elegir el valor que se va a pagar por ese Gas, y si un minero está de acuerdo con ese valor, tomará la transacción y la ejecutará.

Hay tres cosas que son importantes y vitales dentro de Ethereum, y que explicamos a continuación:

1. Unidad de Gas. La Unidad de Gas es la cantidad de Gas que se puede atribuir a una instrucción en específico, pero no tiene ningún valor monetario.

2. Precio de Gas *gas price*. El Precio de Gas por su parte, es el pago de comisión que hacemos por cada Unidad de Gas. Es el precio que se elige pagar por cada unidad y se hace usando unidades decimales de Ether, los llamados Gwei. Esta comisión es la que permite tener prioridad de atención. Si se paga más por cada Unidad de Gas que se use, más rápido los mineros tomarán tu transacción y la llevarán a un bloque.

3. Límite de Gas (*gas limit*). Este es un valor que indica la cantidad máxima de Unidades de Gas que la red Ethereum puede manejar en un momento dado. Es su límite máximo, y es un punto que los mineros no pueden sobrepasar en ningún momento.

Este último límite es interesante porque permite hacer frente al problema de la parada (*halting problem*). Este es un problema de computación que nos permite saber si un programa de computación se ejecutará en un bucle infinito con solo tener a la mano la entrada de datos y su programación. Esta situación plantearía un serio problema en la blockchain que podría llevar a una Denegación de Servicios (DoS). Sin embargo, debido a que Ethereum impone un Gas Limit por bloque, esto significa que ninguna operación en Ethereum por compleja que sea podrá exceder jamás dicho límite (Bit2Me Academy, n.d.). El límite de gas (*gas limit*) actual de la red de Ethereum es de 30 Millones (Ycharts, 2022).

Gas en Quorum

Por defecto, la red de Quorum (GoQuorum) es una red libre de gas, lo que significa que el gas no tiene precio (*gas price* igual a 0).

Las transacciones usan recursos computacionales, por lo que tienen un costo asociado. Gas es la unidad de costo y *gas price* es el precio por cada unidad de gas. El costo de una transacción es la cantidad de gas empleado multiplicado por el precio del gas.

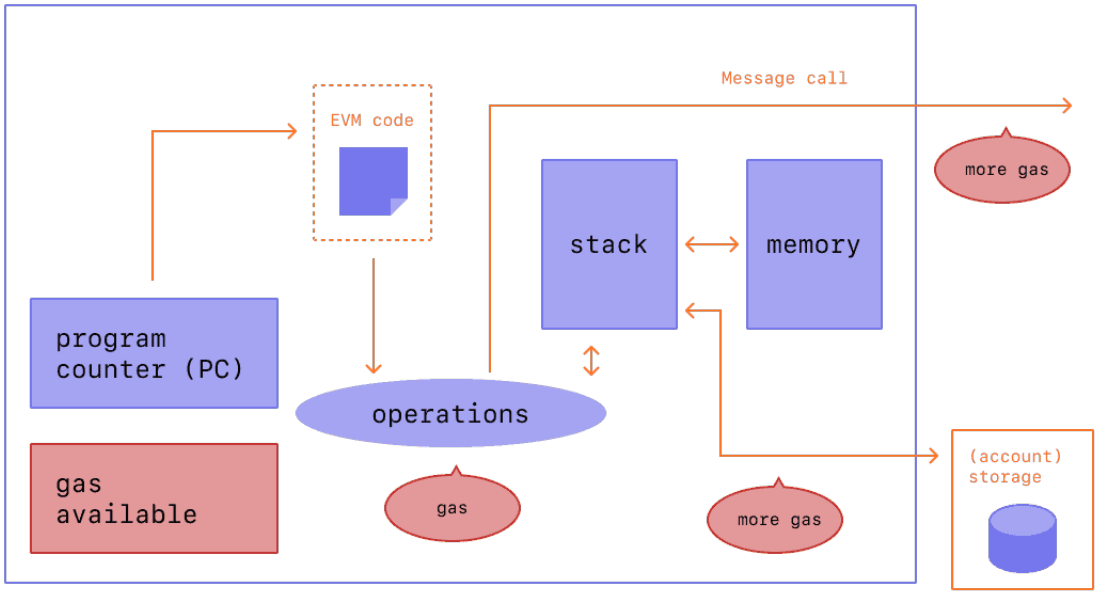

Figure 3.1: Diagrama donde se resume, el funcionamiento interno de un contrato inteligente

En redes públicas como Ethereum, la cuenta que envía la transacción paga el costo de la transacción, en Ether. El minero (o validador, en redes con prueba de autoridad) que incluye la transacción en un bloque, recibe el costo de la transacción como recompensa.

En muchas redes privadas, incluyendo GoQuorum, los participantes de la red ejecutan los validadores y no requiere gas como incentivo (como pasa en las redes públicas). Redes que no requieren gas como incentivo usualmente eliminan el precio del gas o lo configuran para que sea cero (que es, gas gratis). Algunas redes privadas pueden distribuir Ether y utilizar un *gas price* diferente de cero para limitar el uso de recursos.

En redes libres de gas, el precio del gas es cero, aunque las transacciones se mantengan usando gas, por consiguiente, el costo de la transacción (gas usado por el precio del gas) es cero.

En GoQuorum, precio del gas es completamente removido a menos que sea explícitamente habilitado. Precio del gas no está incluido entre los parámetros del objeto transacción en los métodos de la API privada de GoQuorum (Consensys, 2022).

Con lo visto anteriormente podemos concluir que el gas no es totalmente necesario en GoQuorum, ya que no se requiere como incentivo para los mineros, por lo tanto, no se paga por las transacciones realizadas en GoQuorum. No obstante esto, el gas

sigue siendo importante para el funcionamiento de los contratos inteligentes, por el hecho de que nos indica la cantidad de recursos computacionales que se requieren para ejecutar una transacción. Y con esto, podemos saber si el algoritmo es eficiente y escalable. En la siguiente sección se estará abordando este tema.

3.2.3 Pruebas Realizadas

Se realizaron varias pruebas al contrato inteligente, para verificar su funcionamiento. En las pruebas iniciales se comprobó que el contrato se desplegara correctamente, y que se pudieran efectuar las transacciones de manera correcta. Luego se hicieron varias pruebas con diferente cantidad de postores y estos a su vez con varias ofertas (válidas e inválidas). La cantidad de postores se escogía de forma aleatoria entre 1 y 12. Y la cantidad de ofertas de cada uno de estos, igual de forma aleatoria entre 0 y 8 ofertas. Finalmente, se efectuaron algunas pruebas de estrés para comprobar los límites máximos de postores y el límite máximo de ofertas válidas.

En el contrato inteligente, hay 5 métodos que necesitan de transacciones para ejecutarse. Estos son:

- deploy: Desplegar el contrato. Esta transacción se lleva a cabo solamente una vez, por el subastador.
- bid: Enviar una oferta. Esta función puede ser invocada varias veces por varios postores o por el mismo postor. En cada llamado a la función (transacción) se debe enviar un hash (oferta codificada) y opcionalmente hacer un depósito en el contrato (no tiene por el mismo valor de la oferta enviada, puede ser más o menos que esa cantidad).
- reveal: Revelar ofertas. Esta función se debe llamar solamente una vez por cada postor. Su objetivo es revelar las verdaderas ofertas y clasificarlas en ofertas válidas e inválidas.
- auctionEnd: Elegir y anunciar ganadores y dar por terminada la subasta. En este método se ordenan las subastas válidas, se eligen las mejores hasta completar la totalidad de los bonos ofertados.
- withdraw: Retirar fondos. Cuando se llama a esta función se retiran todos los fondos desbloqueados (depósitos de ofertas inválidas o válidas no ganadoras) pertenecientes a la dirección que envía la transacción.

En las pruebas realizadas se obtuvo que las funciones usaban la siguiente cantidad de gas para ejecutarse:

Tabla 3.1: Gas usado por cada método

metodo	gas usado
deploy	1227782
bid	[68688, 83700]
reveal	¿?
auctionEnd	¿?
withdraw	19709

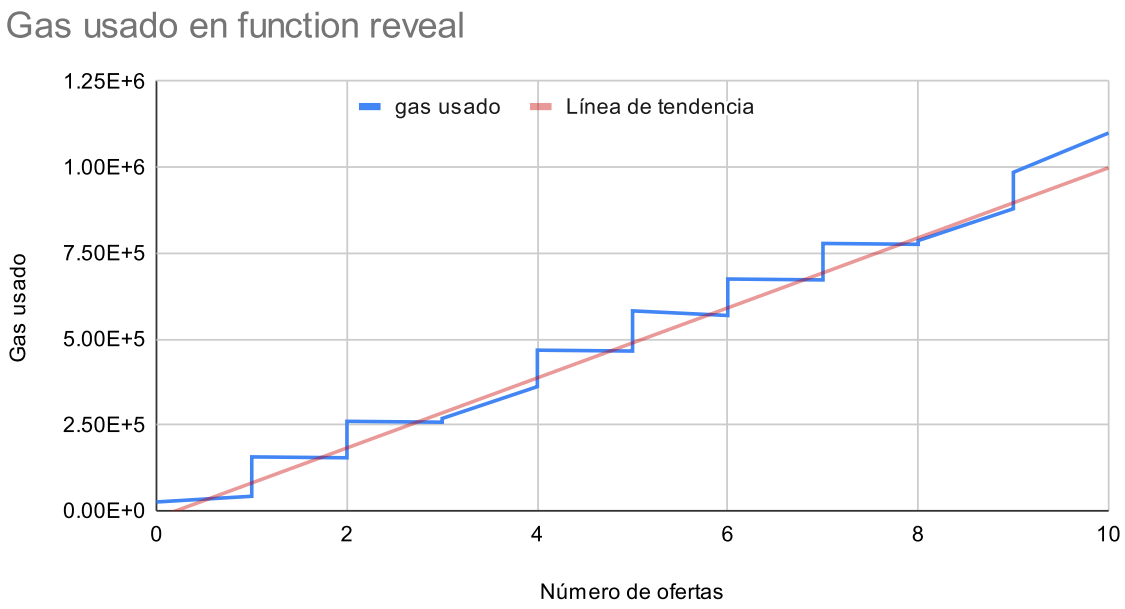

Figure 3.2

En las funciones *reveal* y *auctionEnd* el gas usado, es variable, depende de las condiciones particulares de cada subasta. Tabla 3.1.

En la función *reveal* el gas empleado va a depender de la cantidad de ofertas de cada postor. Es decir, mientras más ofertas tenga que revelar el postor mayor va a ser la cantidad de gas que consumirá esta. Figura 3.2.

En la función *auctionEnd* pasa algo similar a la anteriormente vista. El gas empleado va a depender en este caso de la cantidad de ofertas válidas. Es decir, mientras más ofertas válidas tenga la subasta mayor será la cantidad de gas consumido por esta función.

Para los dos casos anteriores, el gas usado depende cuasi linealmente de la cantidad de ofertas del postor y de la cantidad de ofertas válidas, es decir, si los datos de entrada

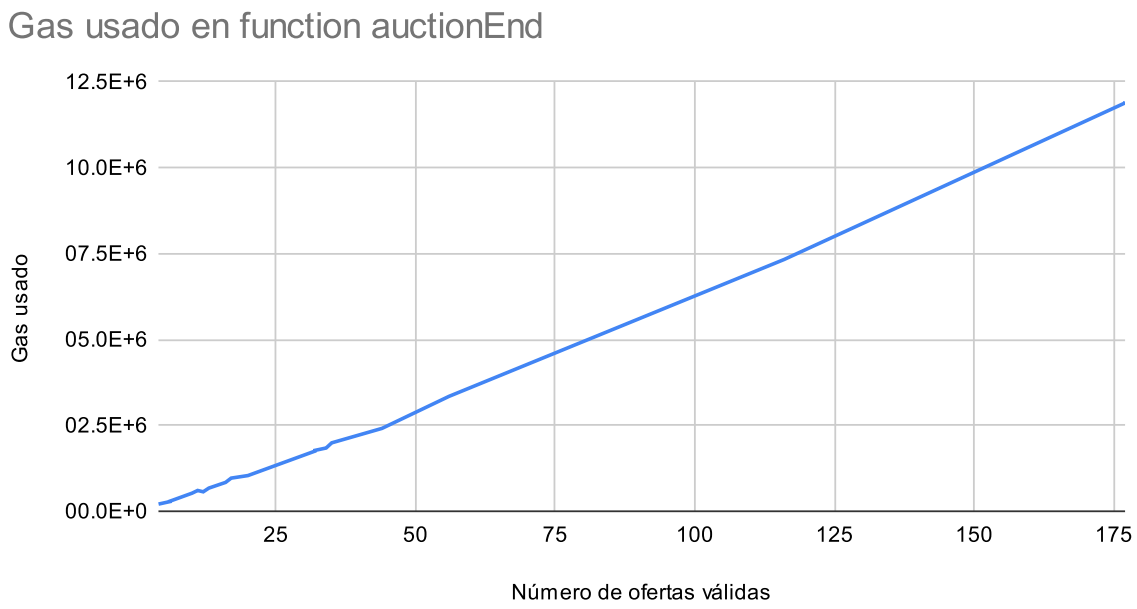

Figure 3.3

se duplican, el gas empleado también se duplica.

En la red local desplegada para probar el contrato inteligente, el límite de gas por bloque es de 12 millones (12M) de unidades. Dado este límite y las pruebas de estrés que se le hicieron al algoritmo en esta red local, los límites máximos que las funciones permitían sin exceder el límite de gas, y por lo tanto, sin afectar la correcta y completa ejecución del programa, se llegó a los siguientes resultados: cantidad máxima de ofertas por un mismo postor fue de 110 aproximadamente, y la cantidad de ofertas válidas en la subasta en general de alrededor de 175 ofertas.

El límite de gas de la red Ethereum actualmente es de 30M (Ycharts, 2022), por lo tanto, en esta, pudiera ser mayor aún los límites de una subasta realizada con el contrato inteligente propuesto. Para predecir los límites en esta red, se usó un algoritmo de regresión lineal, el cual, teniendo los datos de las pruebas en la red local, nos da un valor aproximado de cuánto consumiría una ejecución con una cantidad arbitraria de ofertas. Luego de utilizado el algoritmo, el resultado que se obtuvo es que en la red pública de Ethereum se podría efectuar una subasta donde cada postor pusiera a lo más 290 ofertas y que el total de ofertas válidas fuera a lo sumo 440.

3.3 Resultados

Teniendo en cuenta los resultados obtenidos en las pruebas realizadas, el algoritmo funciona correctamente y cumple con los objetivos planteados. El contrato inteligente propuesto es factible y útil para ejecutar subastas de bonos con ofertas a ciegas en la blockchain de Ethereum o cualquier otra blockchain basada en Ethereum.

A pesar de que los resultados fueron satisfactorios, también se presentan algunos inconvenientes. Las funciones de revelación de ofertas y de fin de subasta, presentan mucho costo computacional para ejecutarse en una blockchain, a medida que aumentan la cantidad de ofertas en la subasta aumenta proporcionalmente su costo computacional y por consiguiente el gas usado, lo que impide que se ejecute correctamente el contrato si se excede el límite de gas de la red utilizada.

El contrato puede ser empleado, pero a pequeña escala, es necesario que la cantidad de ofertas sea limitada, por lo tanto, la subasta se debe hacer en un ambiente cerrado, es decir, donde se tenga control sobre la cantidad de participantes en la subasta y/o de la cantidad de transacciones realizadas.

No se recomienda ejecutar la subasta en entornos abiertos como Internet, donde muchos tengan acceso al contrato, pues de efectuarse un número importante de ofertas en la subasta, el contrato podría tener fallos en su funcionamiento y comportamientos inesperados.

Conclusiones

Las subastas son un mecanismo de venta cada día más usado. Con el auge de las tecnologías de la información, sistemas de pago electrónico y de un mundo interconectado, a través de internet. Cada vez se hace un mayor uso de subastas electrónicas. Por esta razón, se hace necesario protocolos y mecanismos para incrementar la seguridad de estos sistemas y protegerlos de hackers que explotan vulnerabilidades de seguridad en las plataformas de subastas, pero también proteger a los usuarios de subastadores malintencionados que pueden terminar siendo estafadores, este último problema cada vez más latente en la actualidad.

La blockchain y los contratos inteligentes, surgen luego del 2015 y la aparición de la blockchain de Ethereum, como una herramienta eficaz para hacer frente a estos problemas, poniendo a nuestra disposición la capacidad de crear transacciones y programas incorruptibles, que se mantienen públicos a la vista de todos, haciendo que algunos procesos antes oscuros y cerrados, sean ahora totalmente transparentes y abiertos a cualquier persona.

Gracias a esta tecnología, surge la idea de crear subastas en la blockchain, aportándole así seguridad y fiabilidad al proceso. Pero con esta total transparencia de la blockchain surgen algunas dificultades para realizar algunos tipos de subasta. ¿Cómo lograr ejecutar una subasta a ciega, si toda la información es pública?

En la literatura consultada se presentan varias propuestas, casi todas relacionadas directamente con la criptografía para ocultar las ofertas de la subasta. En la propuesta implementada se utiliza con este objetivo la función *hash keccak256*, la cual es altamente probada y utilizada para proteger datos (en este caso ofertas de la subasta) de los demás usuarios de la subasta, incluyendo el subastador.

Para el desarrollo del contrato inteligente se utilizó el lenguaje Solidity, el cual es un lenguaje especializado en la creación de contratos inteligentes específicamente creado para la red de Ethereum, pero que redes posteriormente creadas también han adoptado su uso.

A pesar de no haber llegado a desplegarse el contrato en la red de Quorum, el autor cree que las características de esta red son ideales para desarrollar un contrato de este tipo. Como característica a destacar es que las transacciones realizadas en esta red, al ser una red privada y no pública (como Ethereum) es que las transacciones

son libres de costo, hecho que facilita la realización satisfactoria de todos los procesos y transacciones que requiere el contrato inteligente propuesto.

Además, el contrato inteligente implementado, según las pruebas realizadas, necesita que la cantidad de ofertas sean limitadas, para un correcto funcionamiento del algoritmo y el entorno de una red privada como Quorum permite tener más control sobre eso, sin afectar la seguridad y cumplimiento total de los procesos de la subasta en el contrato.

Recomendaciones

Luego de la investigación, la implementación y las pruebas realizadas en el presente trabajo. Se proponen algunas recomendaciones para analizar e investigar en futuros trabajos.

Se hace necesario un estudio profundo de los tipos de almacenamiento que admite Solidity y los tipos de datos usados para implementar el algoritmo, para optimizar el uso de la memoria empleada por el algoritmo propuesto, recurso muy costoso en las redes blockchain.

Se recomienda además la implementación de un Heap (Monticulo) en detrimento del Quick Sort, para escoger las ofertas ganadoras de la subasta y hacer una comparación, para ver con cuál de los dos algoritmos la fase de verificación de los ganadores utiliza menos gas, y por tanto, menos recursos.

Dado que las pruebas en la red blockchain local fueron satisfactorias, se recomienda el despliegue del contrato implementado en la red de Quorum.

Por último, se hace necesario la implementación de un mecanismo para asegurar el cumplimiento del beneficiario (ofertante de los bonos) de su parte en el acuerdo. La opción recomendada y más factible según el autor, es tokenizar los bonos, y darlos a los postores de las ofertas ganadoras, en proporción a la cantidad de bonos comprados por este.

Bibliography

Banco de México. (n.d.). Mercados financieros [Recuperado el 21 de Septiembre de 2022, de http://educa.banxico.org.mx/banco_mexico_banca_central/sist-finc-mercados-financiero.html]. (Cit. on p. 6).

Barceló, A. (2017). Mercado de Deuda Pública, una propuesta de acciones de política para Cuba. [Tesis de Maestría. La Habana, Cuba]. https://www.bc.gob.cu//storage/investigaciones/March2018/AGZ1q0hd9wboRpF1L93d.pdf. (Cit. on p. 8)

Bertoni, G., Daemen, J., Peeters, M., & Assche, G. V. (2007). Sponge Functions (E. H. Workshop, Ed.). (Cit. on p. 19).

Bit2Me Academy. (n.d.). ¿Qué es el Gas en Ethereum? [Recuperado el 20 Noviembre de 2022, de https://academy.bit2me.com/que-es-gas-en-ethereum]. (Cit. on p. 23).

Blass, E., & Kerschbaum, F. (2018). Strain: A Secure Auction for Blockchains. [In ESORICS; Springer: Berlin/Heidelberg, Germany, 2018; Volume 11098, pp. 87–110.]. (Cit. on p. 10).

Buterin, V. (2013). Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform [Recuperado el 16 de Octubre de 2022, de https://ethereum.org/en/whitepaper/]. (Cit. on p. 1).

Consensys. (2022). Free gas networks [Recuperado el 23 Noviembre de 2022, de https://consensys.net/docs/goquorum/en/stable/concepts/free-gas-network/]. (Cit. on p. 24).

Díaz, G. (2018). Ethereum: historia de la plataforma de contratos inteligentes más usada [Recuperado el 11 de Septiembre de 2022, de https://www.criptonoticias.com/tecnologia/ethereum-historia-plataforma-contratos-inteligentes-usada/]. (Cit. on p. 1).

Emem, M. (2018). Andy Warhol's Multi-Million Dollar Painting Tokenized and Sold on Blockchain. (Cit. on p. 9).

Escarramán, A. T. (2022). Seguridad en la Blockchain de Ethereum: explotación y mitigación de vulnerabilidades modernas en Smart Contracts (U. P. de Madrid, Ed.). (Cit. on p. 20).

Foundation, P. S. (2021). Python 3.8.10 documentation [https://docs.python.org/release/3.8.10/]. (Cit. on p. 22).

Friedman, D. (1992). The Double Auction Market Institution: A Survey [http://www.its.caltech.edu/~pbs/expfinance/Readings/FriedmanDA.pdf]. (Cit. on p. 6).

Galal, H., & Youssef, A. (2018a). Succinctly Verifiable Sealed-Bid Auction Smart Contract [In Data Privacy Management, Cryptocurrencies and Blockchain Technology; Springer: Berlin/Heidelberg, Germany]. (Cit. on p. 11).

Galal, H., & Youssef, A. (2018b). Verifiable Sealed-Bid Auction on the Ethereum Blockchain [In Financial Cryptography; Springer: Berlin/Heidelberg, Germany]. (Cit. on p. 11).

ig.com. (n.d.). ¿Qué son los bonos del Estado y cómo comerciar con ellos? [Recuperado el 21 de Septiembre de 2022, de https://www.ig.com/es/bonos/que-son-los-bonos-del-estado-y-como-comerciar-con-ellos]. (Cit. on p. 7).

Investopedia.com. (n.d.). (Cit. on p. 5).

Kosba, A., Miller, A., Shi, E., Wen, Z., & Papamanthou, C. (2016). Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts [In Proceedings of the IEEE Symposium on Security and Privacy, San Jose, CA, USA, 22–26 May 2016; pp. 839–858.]. (Cit. on p. 10).

Krishna, V. (2009). Auction Theory. (Cit. on p. 4).

Li, H., & Xue, W. (2021). A Blockchain-Based Sealed-Bid e-Auction Scheme with Smart Contract and Zero-Knowledge Proof. (Cit. on p. 11).

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. (Cit. on p. 1).

Neuendorf, H. (2018). Christie's Will Become the First Major Auction House to Use Blockchain in a Sale. (Cit. on p. 9).

NIST. (2015). Announcing Approval of Federal Information Processing Standard (FIPS) 202, SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions, and Revision of the Applicability Clause of FIPS 180-4, Secure Hash Standard. (Cit. on p. 19).

Patten, R. (1970). Tatworth Candle Auction (N. 2. (1. Folklore 81, Ed.). (Cit. on p. 6).

Pedersen, T. (1991). Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing [In CRYPTO; Springer: Berlin/Heidelberg, Germany]. (Cit. on p. 11).

Pérez, I. (n.d.). Qué son los contratos inteligentes [Recuperado el 11 de Septiembre de 2022, de https://www.criptonoticias.com/criptopedia-old/que-son-contratos-inteligentes-blockchain-criptomonedas/]. (Cit. on p. 2).

Real Academia Española. (n.d.). Subasta [Recuperado el 18 de Septiembre de 2022, de https://dle.rae.es/subasta]. (Cit. on p. 4).

Roldán, P. N. (2017). Subasta (Economipedia.com, Ed.) [Recuperado el 19 de Octubre de 2022, de https://economipedia.com/definiciones/subasta.html]. (Cit. on pp. 4, 5).

Sánchez, D. C. (2020). Raziel: Private and Verifiable Smart Contracts on Blockchains [https://eprint.iacr.org/2017/878]. https://eprint.iacr.org/2017/878. (Cit. on p. 11)

Sharma, G., Verstraeten, D., Saraswat, V., Dricot, J.-M., & Markowitch, O. (2021). Anonymous Sealed-Bid Auction on Ethereum. [Electronics 2021, 10, 2340. https://doi.org/10.3390/electronics10192340]. (Cit. on p. 11).

Shi, Z., de Laat, C., Grosso, P., & Zhao, Z. (2021). When Blockchain Meets Auction Models: A Survey, Some Applications, and Challenges [1v43521.0112:viXra]. (Cit. on p. 9).

Solidity. (n.d.). Solidity Documentation [https://docs.soliditylang.org/en/v0.8.4/]. (Cit. on p. 22).

Solunion. (2021). ¿Qué es y para qué sirve la tecnología blockchain? [Recuperado el 10 de Septiembre de 2022, de https://www.solunion.cl/blog/que-es-y-para-que-sirve-la-tecnologia-blockchain/]. (Cit. on p. 1).

Stevens, M., Bursztein, E., Karpman, P., Albertini, A., & Markov, Y. (2017). The first collision for full SHA-1 [Retrieved February 23, 2017]. (Cit. on p. 19).

Tamayo, E. C., & Ferrer, L. I. (2021). ¿Cómo se financia el déficit presupuestario de Cuba? (cubadebate.com, Ed.) [Recuperado el 21 de Septiembre de 2022, de http://www.cubadebate.cu/especiales/2021/06/08/como-se-financia-el-deficit-presupuestario-de-cuba/]. (Cit. on p. 8).

Tapscott, D., & Tapscott, A. (2016). *Blockchain Revolution: How the Technology Behind Bitcoin is Changing Money, Business, and the World*. Portfolio / Penguin. https://books.google.com.cu/books?id=L1QHjwEACAAJ. (Cit. on p. 1)

Truffle. (n.d.). Ganache Documentation [https://github.com/trufflesuite/ganache]. (Cit. on p. 22).

Wood, G. (2021). Ethereum: A Secure Decentralised Generalised Transaction Ledger. (Cit. on p. 9).

Wu, S., Chen, Y., Wang, Q., Li, M., Wang, C., & Luo, X. (2019). CReam: A smart contract enabled collusion-resistant e-auction (I. T. I. F. Security, Ed.) [vol. 14, no. 7]. (Cit. on p. 8).

Ycharts. (2022). Ethereum Average Gas Limit [Recuperado el 25 Noviembre de 2022, de https://ycharts.com/indicators/ethereum_average_gas_limit]. (Cit. on pp. 23, 27).

Zcash. (2022). The Basics [Recuperado el 25 de Noviembre de 2022, de https://z.cash/]. (Cit. on p. 10).