Título

PABLO Blanco Peris

Máster en INGENIERÍA Informática, Facultad de Informática,

Universidad Complutense de Madrid



Trabajo Fin de Máster en Ingeniería Informática

Fecha

Director:

Adrián Riesco Rodríguez

Autorización de Difusión

Pablo Blanco Peris

Fecha

El abajo firmante, matriculado/a en el Máster en Investigación en Informática de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Máster: “TÍTULO”, realizado durante el curso académico 20XX-20XX bajo la dirección de XXXX [y con la colaboración externa de dirección de YYYY] en el Departamento de Sistemas Informáticos y Computación, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

Resumen en castellano

[Enter abstract here, no longer than 350 words. Be sure to retain the Section Break below.]

Palabras clave

Resumen en inglés

[Enter abstract here, no longer than 350 words. Be sure to retain the Section Break below.]

Keywords

Índice de contenidos

[Autorización de Difusión ii](#_Toc9205706)

[Resumen en castellano iii](#_Toc9205707)

[Palabras clave iii](#_Toc9205708)

[Resumen en inglés iv](#_Toc9205709)

[Keywords iv](#_Toc9205710)

[Índice de contenidos 1](#_Toc9205711)

[Agradecimientos 2](#_Toc9205712)

[Capítulo 1 - Introducción y motivación 3](#_Toc9205713)

[Capítulo 2 - Preliminares 6](#_Toc9205714)

[2.1 Blockchain 6](#_Toc9205715)

[Historia 6](#_Toc9205716)

[¿Qué es blockchain? 7](#_Toc9205717)

[Tipos de blockchain 9](#_Toc9205718)

[Proof-of-work 12](#_Toc9205719)

[2.2 Ethereum 13](#_Toc9205720)

[Gas y pagos en Ethereum 14](#_Toc9205721)

[2.3 Smart Contracts 15](#_Toc9205722)

[Capítulo 3 - Diseño e implementación 18](#_Toc9205723)

[3.1 Front-end 20](#_Toc9205724)

[3.2 Herramientas de conexiones 20](#_Toc9205725)

[3.3 Back-end o DApp 20](#_Toc9205726)

[Capítulo 4 - Demo 21](#_Toc9205727)

[Capítulo 5 - Conclusión 22](#_Toc9205728)

[Capítulo 6 - Bibliografía 23](#_Toc9205729)

Agradecimientos

The Acknowledgement page is optional. If you include it, retain the Acknowledgements heading and enter your text here. If you do not include it, delete the entire page. Be sure to retain the Page Break that occurs after the List of Table page above.

# Introducción y motivación

A pesar de que el término blockchain naciera entre los años 2008 y 2009 por parte de la persona u organización llamada Satoshi Nakamoto [1], empezó a escucharse con más fuerza con la revolución del *bitcoin* y las criptomonedas en el año 2017 en el que llegó a valer 20.000$ cada unidad de la moneda virtual *bitcoin* o también conocida BTC [2].

Además de los aspectos técnicos que representa este concepto es una palabra que está de moda en los últimos años. Aunque blockchain significa “cadena de bloques”, el propio nombre no deja muy claro qué es exactamente. Sin embargo, es un nuevo concepto en la informática que plantea una revolución en una gran cantidad de ámbitos de la vida cotidiana como por ejemplo la economía.

Uno de los principales objetivos de esta tecnología es eliminar a los intermediarios de las transacciones de dinero. Un ejemplo muy sencillo podría ser:

Una persona A, quiere enviar dinero a una persona B; una de las soluciones más cómodas que existe a día de hoy sería mediante una transferencia bancaria. Sin embargo, para ello es necesario la intervención de terceros, en este caso los bancos, unas entidades centralizadas de las que se depende a la hora de realizar transacciones de capital de este tipo.

Esta gestión necesariamente cuenta con roles que podrían no aparecer en caso de que esta operación se hubiera llevado a cabo a través de una cadena de bloques, en donde se eliminan los intermediarios y se descentraliza la gestión. Aplicando estas operaciones en una *blockchain* provoca que el control del proceso sea de los usuarios y no de entidades externas.

Se podría entender de manera resumida y sencilla como una enorme base de datos o libro de cuentas en el que todos los registros están entrelazados entre sí y cifrados para mantener la seguridad y privacidad de las transacciones así como la identidad de los propios usuarios que participan en ellas. Para que todo este sistema sea estable, fiable y funcione correctamente necesita una red P2P de nodos comunicados entre ellos. Todos estos nodos contienen la misma *blockchain*, de manera que los bloques que se añaden a la cadena quedan registrados de forma permanente en ella y no pueden ser modificados. Cada vez que se crea un bloque válido nuevo se añade a la cadena y se sincroniza con el resto de nodos que comprueban la validez del mismo. Al depender todos los bloques del anterior se garantiza la seguridad ante un intento de fraude por parte de algún nodo, ya que el resto de nodos no aceptarían el nuevo bloque.

Sin embargo, las transacciones de dinero no son el único ámbito donde puede aplicarse el *blockchain.* La red Ethereum propone una solución en la que las transacciones pueden ser *Smart contracts* o contratos inteligentes, que permiten definir transacciones de distintos tipos sustituyendo productos y servicios que dependen de terceros.

Desde el 2017 la investigación de esta tecnología ha ido incrementando progresivamente y ya está presente en una gran cantidad de proyectos de diferentes ámbitos. Estos *Smart contracts* ofrecen soluciones útiles a día de hoy y ya se están usando en múltiples ámbitos con diferentes objetivos:

* El gobierno japonés cuenta con un proyecto basado en registros de propiedad a través de *blockchain* [3].
* La conocida aplicación Spotify compró en 2017 una empresa especializada en esta tecnología para abordar una solución en la gestión de pagos y autorías de su plataforma [4].
* En sanidad, la mayoría de tecnólogos y profesionales de la salud ven en las cadenas de bloques la opción ideal para registrar historiales médicos de una manera segura y confidencial [5].
* En servicios públicos y gubernamentales, ya existen plataformas *blockchain* orientadas a contabilizar los votos en unas elecciones presidenciales de manera transparente y evitando fraudes [6].

El objetivo principal de este trabajo es combinar distintas tecnologías punteras actuales para desarrollar un proyecto útil y estable que propone una solución de negocio que puede ser utilizada a día de hoy en un ámbito concreto.

El resultado que se presenta en este proyecto implementa una plataforma completa que proporciona un servicio de alquiler de coches. A través de un *Smart contract* dentro de la red *blockchain* Ethereum [7] como *back-end*, que se encarga de la gestión de la información dentro de la red de pruebas llamada Rinkeby [8]. La comunicación del usuario con el *back-end* se realiza a través de una aplicación móvil para dispositivos móviles con sistema operativos iOS como *front-end,* que se encarga de gestionar las interfaces visuales, y la interacción y conexión con el *Smart contract*.

La solución de negocio descrita se asemeja a las ya conocidas empresas Car2Go [9] y eMov [10]. Sin embargo, como novedad a estos sistemas existentes se propone una investigación de la optimización del sistema y las ventajas que podría suponer mantener la gestión de la información y los datos del alquiler de los vehículos a través de un contrato inteligente.

En el Capítulo 2 se encuentran descritas las tecnologías utilizadas en este proyecto sin entrar en detalle en como se han usado para la implementación de la plataforma.

En el Capítulo 3 se describe el diseño y la implementación del proyecto, es decir, la arquitectura global que sigue el proyecto y los detalles de como se han conectado las tecnologías empleadas para el correcto funcionamiento de la plataforma al completo. También aparecen ejemplos de las interacciones con el *Smart contract* desplegado y con el que se trata toda la gestión de la información acerca del alquiler de vehículos.

En el Capítulo 4 se muestra una demo visual de la aplicación así como flujos que sigue y su comportamiento ante distintas situaciones.

En el Capítulo 5 se detallan las conclusiones finales una vez implementada la plataforma al completo y probada.

# Preliminares

En esta sección se incluye una descripción detallada de las tecnologías que han sido utilizadas para llevar a cabo la realización del proyecto descrito en este documento.

La tecnología principal en la que se basa este proyecto es *blockchain* [11], también conocida como cadena de bloques. El término blockchain se aplicó por primera vez en 2009 como parte del Bitcoin [1], aunque no hace mucho que ha empezado a escucharse con fuerza en la sociedad. Sin embargo, la mayoría de la gente no sabe exactamente qué es el blockchain, para qué sirve o cómo funciona. En este trabajo se persigue el objetivo de explicar el blockchain con la mayor claridad posible para que cualquier persona con unos conocimientos básicos en informática sea capaz de entender sus beneficios y los usos que se le puede dar.

Esta tecnología se utiliza dentro de este proyecto para soportar la estructura de datos y la información, a través de un Smart contract [12] en el que se va a mantener la información dentro de la blockchain Ethereum [13], concretamente en Rinkeby [8], una cadena de bloques de *testing* proporcionada por Ethereum enfocada a desarrolladores. Los Smart contracts son scripts que residen en la cadena de bloques que permiten la automatización de procesos de varios pasos.

## Blockchain

### Historia

El concepto de moneda digital descentralizada, así como las aplicaciones alternativas o los registros de propiedad, han existido durante décadas. Los protocolos anónimos de efectivo electrónico de los años 80 y 90, que dependían principalmente de una primitiva criptográfica conocida como el cegamiento de Chaumian, proporcionaron una moneda con un alto grado de privacidad, pero los protocolos en gran parte no lograron ganar terreno debido a su dependencia de un intermediario centralizado. En 1998, Wei Dai se convirtió en la primera propuesta en introducir la idea de crear dinero mediante la resolución de acertijos computacionales así como en un consenso descentralizado, pero la propuesta fue escasa en detalles sobre cómo se podría implementar realmente el consenso descentralizado. Posteriormente, En 2005, Hal Finney introdujo un concepto de un sistema para crear un concepto de criptomoneda, pero no alcanzó el ideal al confiar en la computación confiable como *back-end*. En 2009, Satoshi Nakamoto implementó por primera vez una moneda descentralizada en la práctica, combinando las primitivas establecidas para administrar la propiedad a través de la criptografía de clave pública con un algoritmo de consenso para realizar un seguimiento de quién posee las monedas, conocida como prueba de trabajo *o proof of work* [1].

El mecanismo detrás de la prueba de trabajo fue un gran avance en el espacio porque resolvió simultáneamente dos problemas. Primero, proporcionó un algoritmo de consenso simple y moderadamente efectivo, que permite a los nodos de la red acordar colectivamente un conjunto de actualizaciones canónicas del estado del libro mayor de Bitcoin. En segundo lugar, proporcionó un mecanismo para permitir la libre entrada en el proceso de consenso, resolviendo el problema político de decidir quién puede influir en el consenso y, al mismo tiempo, evitar los ataques de sibila. Lo hace sustituyendo una barrera formal a la participación, como el requisito de estar registrado como una entidad única en una lista particular, con una barrera económica: el peso de un solo nodo en el proceso de votación por consenso es directamente proporcional a la potencia de cálculo. que trae el nodo. Desde entonces, se ha propuesto un enfoque alternativo *llamado proof of stake*, que calcula el peso de un nodo como proporcional a sus tenencias de moneda y no a recursos computacionales [13].

### ¿Qué es blockchain?

Blockchain es una estructura de datos en la que la información contenida se agrupa en secuencias de bloques a los que se va agregando información relacionada con el bloque anterior, que, a su vez, contiene información relacionada con el bloque anterior a éste y así recursivamente, de manera que, debido a técnicas criptográficas de clave pública con un algoritmo de consenso, si un bloque es modificado modificaría todos los bloques posteriores, quitando así su validez. Estas cadenas de bloques se reparten entre nodos distribuidos en la red, comúnmente llamados mineros. Esta estructura persigue el objetivo de mantener una descentralización de la información y evitar dependencias con ciertos nodos, de manera que ningún nodo está por encima de otro y ningún nodo es indispensable para el correcto funcionamiento de la cadena, manteniendo así una estructura jerárquica horizontal. El intercambio de información se realiza de manera que, cuando un nodo añade un nuevo bloque, se lo comunica al resto de mineros de la red para que actualicen su cadena de bloques y así contengan todos la misma información.

De esta manera se crea un entorno distribuido en donde la cadena de bloques ejerce como una base de datos pública no con capacidad de asegurar la veracidad del contenido. Constantemente este libro digital de registros va creciendo añadiendo bloques a la cadena, las operaciones se van registrando de manera cronológica en la propia cadena, y se permite realizar un seguimiento sin la necesidad de registros centrales. Es importante destacar que los bloques tienen un orden cronológico dentro de la cadena y esto no se puede modificar ya que los bloques están interrelacionados entre ellos. Por ello, una vez que un bloque es considerado válido dentro de la red de nodos, siempre permanecerá en la cadena de bloques.

La tecnología blockchain es muy adecuada para los entornos en los que se quiera almacenar información de manera cronológica y se quiera garantizar la integridad de la información, imposibilitando la modificación de la información con el objetivo de garantizar una confianza distribuida en lugar de que quede depositada en una entidad certificadora de confianza que se encarga de garantizar la seguridad de las comunicaciones y las transacciones digitales como podría ser una entidad financiera. Sirven para verificar las identidades de los participantes y evitar fraudes mediante claves públicas y privadas [14]. Un ejemplo de autoridad certificadora española podría ser la Fábrica Nacional de Moneda y Timbre (FNMT).

Gracias a las funciones hash[[1]](#footnote-1) y la criptografía asimétrica que se emplean es posible implementar un registro contable distribuido, llamado *ledger*, que garantiza la seguridad y la integridad del dinero digital. Es por ello por lo que se está apostando a día de hoy en esta tecnología con las criptomonedas.

A día de hoy es habitual encontrar noticias relacionadas con la tecnología blockchain y las DApps (aplicaciones descentralizadas) y la fuerte inversión que se está movilizando en este sector. Por ejemplo, las DApps movilizaron 6.700 millones de dólares en 2018 [15]. El periódico digital Criptonoticias cubre los hechos más relevantes a las tecnologías Bitcoin ofreciendo noticias, avances y tendencias sobre estos temas [16].

Para evitar una entidad de confianza que centralice la información a la hora de garantizar la integridad de los datos por parte de todos los participantes de la red es necesario seguir un protocolo adecuado para todas las operaciones que se ejecuten sobre la blockchain. Gracias a esto se dice que la seguridad y la confianza de la tecnología dentro de todo el sistema se genera, se establece y se consolida por los propios miembros (los mineros). Incluso en un entorno en el que exista una minoría de nodos maliciosos por la red (nodos sybil) sería necesario que un atacante cubriera en mayoría la potencia de cómputo y presencia en la red que la que sumaría el resto de nodos combinados.

Gracias a estas características de confianza distribuida y mantenimiento de la integridad de los datos esta tecnología es útil en diferentes escenarios, como por ejemplo:

* Almacenamiento de la información, mediante la replicación de la información de la cadena.
* Confirmación de datos, mediante un protocolo de consenso entre los nodos participantes. El tipo de algoritmo más utilizado es el de prueba de trabajo (en inglés *proof of work*), en el que existe un proceso de validación de los nuevos bloques llamado minería.

Sin embargo, el blockchain está abarcando la mayoría de su peso en el ámbito financiero, en transacciones de dinero digital. Es por ello que no hace mucho comenzó la tendencia de las criptomonedas, que ha tenido y tiene mucha influencia a nivel mundial.

### Tipos de blockchain

Existen diferentes tipos de blockchain: blockchains públicas, blockchains privadas y blockchains híbridas. Cada una de ellas tiene características diferentes y, por tanto, tienen usos diferentes entre sí.

#### Blockchains públicas

Las *blockchains* públicas son accesibles para todo el mundo, lo único que se necesita para acceder a ellas es un ordenador y una conexión a internet. Bitcoin fue la primera *blockchain* pública, con la que nació en 2009 esta tecnología [1]. De hecho, a día de hoy, esta criptomoneda es la más fuerte y las más consolidada de todas las *blockchains* públicas que se encuentran en activo.

Las blockchains públicas se caracterizan por ser:

* **Descentralizadas**: evitando de esta manera la necesidad de una entidad central de confianza como podría ser un banco.
* **Distribuidas:** ya que, cada nodo de la red cuenta con una copia exacta de la cadena de bloques.
* **Consensuadas**: hay un consenso generalizado marcado por unas reglas para que las operaciones sean tomadas como válidas. Como por ejemplo un usuario no podría ejecutar una transacción en la que tratara de enviar más dinero del que realmente contiene.
* **Abiertas**: de manera que todo usuario que quiera es libre de participar de manera sencilla descargando el software necesario y realizando transacciones.
* **Seguras**: su seguridad es representada con la “verdad” que se encuentra en la integridad de los datos que se encuentran en ellas, así como la imposibilidad de modificación de datos anteriores.

Las cadenas de bloques públicas son mantenidas por todo aquel que quiera participar. En el caso de los Bitcoin, esto es posible gracias a los mineros, que deben contar con equipos con mucha capacidad de cómputo y por lo tanto gastar electricidad para poder mantenerlo. Sin embargo, esto no lo hacen gratuitamente, sino que existen ciertas recompensas cuando un nodo crea un bloque nuevo y lo añade a la cadena, por lo que se trabaja por incentivos. Esta parte se explicará un poco más adelante en la sección de Bitcoin.

Este proyecto utiliza la *blockchain* Ethereum. , que es una de las *blockchains* públicas más conocidas, ya que, ha sido el caso más exitoso de blockchain pública por detrás de Bitcoin. Cabe mencionar cuando se habla sobre blockchains públicas un grupo de criptomonedas que son mundialmente conocidas como Litecoin [17] o Monero [18], ya que tienen una gran repercusión en el mercado de las criptomonedas y están continuamente moviéndose entre los usuarios de la red a través de transacciones. A este tipo de cadenas también se las conoce como cadenas de bloques sin permisos.

#### Blockchains privadas

Estas cadenas de bloques se caracterizan porque el proceso de consenso que contienen, así como su participación, están limitados. De esta manera, solo ciertos usuarios tienen los derechos necesarios para acceder a ellas.

En estas cadenas es necesario contar con permisos para llevar a cabo transacciones. Por lo que, la lectura de la información de la blockchain esté limitada a estos usuarios. Existen diferentes tipos de permisos, por ejemplo pueden existir usuarios que sólo sean capaces de leer la información y pueden existir por otro lado usuarios con la capacidad de acceder a la información y de realizar transacciones.

Sin embargo, estas cadenas de bloques dejan de lado inevitablemente la descentralización del poder por lo que ya no se representaría el sistema como totalmente descentralizado.

Dentro de este grupo de blockchain existen variaciones, de manera que una cadena de bloques privada puede contar con un nivel de descentralización mayor o menor según la cantidad de entidades o grupos que formen parte del consenso.

Cuanta mayor sea la cantidad de figuras que forman parte del consenso, mayor nivel de descentralización. Por lo que existen blockchain privadas con una sola figura o entidad en el consenso que tenga permisos de escritura provocando así una cadena completamente privada y parcialmente centralizada.

Estas cadenas cuentan con las siguientes características:

* **Privadas:** no son accesibles para todo el mundo por lo que para mantenerlas es necesario definir unas entidades preseleccionadas**.**
* **Intereses:** los usuarios que mantienen estas bases de datos lo hacen por intereses propios como puede ser reputación.
* **Jerarquía:** existe una jerarquía de poder y permisos dentro de la cadena por lo que no todos los usuarios tienen por defecto los permisos de escritura. Además, el contenido de los bloques no es accesible para todos los participantes.

Algunas de las blockchains privadas más famosas a día de hoy son:

* Hyperledger [19], para la fundación Linux.
* R3 [20], que se trata de un consorcio de bancos a nivel internacional para desarrollar soluciones bancarias a través de una blockchain privada.
* Ripple [21], una criptomoneda para realizar transferencias de dinero digital a nivel internacional.

#### Blockchains híbridas

Este tipo de cadenas de bloques son una combinación de las públicas y las privadas. En estas blockchains los nodos que participan han sido invitados previamente, sin embargo, todas las transacciones son públicas. De esta manera, los nodos se encargan de mantener y proporcionar seguridad a esta cadena, a pesar de que las transacciones sean públicas para el resto de usuarios.

Algunos ejemplos de blockchains híbridas son:

* BigchainDB [22], que permite a los desarrolladores y empresas implementar la prueba de conceptos, plataformas y aplicaciones de blockchain con una base de datos de blockchain, admitiendo una amplia gama de industrias y casos de uso..
* Evernym [23], que ofrece la posibilidad de presentar información de identidad de cualquier tipo a cualquier otra persona en el mundo.

### Proof-of-work

La prueba de trabajo implica la búsqueda de un valor que, cuando se aplica el hash, como por ejemplo con SHA-256, el resultado debe comenzar con un número de ceros definido en el convenio, el cual puede ir variando. El trabajo promedio requerido es exponencial con respecto al número de ceros requeridos y se puede verificar ejecutando un solo hash. Una vez que la CPU ha generado un esfuerzo para cumplir con la prueba de trabajo, el bloque no se puede cambiar sin rehacer el trabajo. Como los bloques posteriores se encadenan después de esto, el trabajo para cambiar el bloque incluiría rehacer todos los bloques posteriores.

La prueba de trabajo también resuelve el problema de determinar la representación en la toma de decisiones mayoritaria. Si la mayoría se basara en que una única dirección IP representa un voto, podría ser falsificada por cualquier persona que pueda asignar muchas IPs. La prueba de trabajo es esencialmente de un voto por CPU. La decisión mayoritaria está representada por la cadena más larga, que cuenta con el mayor esfuerzo de prueba de trabajo invertido en ella. Si la mayoría de la potencia de la CPU está controlada por nodos honestos, la cadena honesta crecerá más rápido y superará a cualquier cadena competidora.

Para modificar un bloque anterior, un atacante tendría que rehacer la prueba de trabajo del bloque y todos los bloques posteriores [1].

## Ethereum

Ethereum es una red pública y descentralizada de cadenas de bloques capaz de ejecutar código de programación de cualquier aplicación descentralizada. Esta plataforma permite compartir información con todo el mundo manteniendo su integridad ya que dicha información no puede ser manipulada ni modificada.

El objetivo de esta *blockchain* es crear un protocolo alternativo para la creación de aplicaciones descentralizadas, proporcionando un conjunto diferente de concesiones que pueden ser muy útiles para una gran clase de aplicaciones descentralizadas, con especial énfasis en situaciones en las que el rápido desarrollo, la seguridad para las aplicaciones y la capacidad de diferentes aplicaciones para interactuar de manera muy eficiente son importantes. Ethereum hace esto construyendo lo que es esencialmente la última capa fundamental abstracta: una cadena de bloques con un lenguaje de programación Turing completo incorporado, que permite a cualquiera escribir contratos inteligentes y aplicaciones descentralizadas donde pueden crear sus propias reglas arbitrarias de propiedad, formatos de transacción y funciones de transición de estado. Los contratos inteligentes, o *Smart contracts*, también se pueden construir sobre la plataforma, con mucho más poder que el que ofrecen las secuencias de comandos de Bitcoin debido a las características adicionales de Turing [24].

Esta *blockchain*, en su conjunto, puede verse como una máquina de estado basada en transacciones. En primer lugar está el estado de génesis, que es el estado inicial de la *blockchain* y con el tiempo se van ejecutando incrementalmente transacciones. Los estados puede incluir información como saldos de cuentas, acuerdos, datos pertenecientes al mundo físico, etc. En resumen, todo lo que actualmente puede ser representado por una computadora es admisible. También existen los cambios de estado no válidos como por ejemplo, reducir el saldo de una cuenta sin un aumento igual y opuesto en otras cuentas, o una transacción de un importe superior al importe que tiene el remitente.

Ethereum tiene una moneda intrínseca denominada *Ether*, también conocida por las siglas ETH. La subdenominación más pequeña del *Ether*, y por lo tanto aquella en la que se cuentan todos los valores enteros de la moneda, se hace llamar *Wei*. Un *Ether* se define como 1018 *Wei*. Existen otras subdenominaciones con sus correspondientes valores respectivos al Ether, sin embargo *Ether* y *Wei* son las más comunes y las más utilizadas [25].

### Gas y pagos en Ethereum

Para evitar problemas de uso indebido de la red todos los cálculos programables en Ethereum están sujetos a tarifas. La tabla de tarifas se especifica en unidades de gas. **Cualquier fragmento de cómputo programable como crear contratos, hacer llamadas a funciones, utilizar y acceder a la cuenta, ejecutar operaciones o realizar transacciones tiene un costo universalmente acordado en términos de gas.** Cada transacción tiene una cantidad específica de gas asociada denominada *gasLimit*. Esta es la cantidad de gas que se compra implícitamente del saldo de la cuenta del remitente. La transacción es considerada inválida si el saldo de la cuenta no puede soportar la compra del gas necesario. Se llama *gasLimit* ya que cualquier gas no utilizado al final de la transacción se reembolsa (a la misma tasa de compra) a la cuenta del remitente. El gas no existe fuera de la ejecución de una transacción.

En general, el *Ether* utilizado para comprar gas que no se reembolsa se entrega a la dirección del beneficiario, la dirección de una cuenta que generalmente está bajo el control del minero. Los operadores pueden especificar el precio del gas que deseen, sin embargo, los mineros pueden ignorar las transacciones que elijan. Por lo tanto, un precio de gas más alto en una transacción le costará más al remitente en términos de *Ether* y le dará un mayor valor al minero y, por lo tanto, será más probable que sea seleccionado por más mineros para ser finalmente el bloque correcto. Por regla general, los mineros elegirán anunciar el precio mínimo del gas para el cual ejecutarán las transacciones y los operadores serán libres de cubrir estos precios para determinar qué precio del gas ofrecerá. Dado que habrá una distribución (ponderada) de los precios mínimos aceptables del gas, los mineros tendrán necesariamente una compensación entre bajar el precio del gas y maximizar la posibilidad de que su transacción se realice de manera oportuna [25].

## Smart Contracts

Los *Smart contracts* son *scripts* que residen en la blockchain de manera que son capaces ejecutar código dentro de la propia cadena bloques una vez desplegados. Estos contratos inteligentes aprovechan las propiedades de las blockchain como la confiabilidad dentro de la cadena y las técnicas criptográficas empleadas para las interacciones con el fin de ofrecer flujos de trabajo distribuidos y automatizados. Esto provoca que los *Smart contracts* sean un punto fuerte en la investigación dentro del dominio del Internet de las Cosas para nuevos casos de uso.

Al pertenecer a la cadena y estar en un bloque concreto estos *Smart* *contracts* tienen una dirección única, la cual se necesita para la conexión y la interacción con ellos. Estos contratos se ejecutan de manera independiente y automática en cada nodo de la red, por lo que cada nodo de la red habilitado para ello ejecuta una máquina virtual con la capacidad de ejecutar el código que se encuentra en ellos. De esta manera la *blockchain* actúa como una máquina virtual distribuida [12].

Los Smart contracts se programan en el lenguaje Solidity, que se asemeja a las clases en los lenguajes orientados a objetos pero en lugar de utilizar la palabra class se usa contract.

Existen funciones *get* y *set* para leer y escribir en variables, estructuras de datos y tipos enumerados. También puede utilizarse la herencia entre contratos, y existe lo que se llama modificadores (de funciones), que sirven para que las funciones comprueben que se cumple cierta condición como requisito para ejecutarse. También cuenta con eventos que se pueden interpretar como *triggers* que reaccionan a algún comportamiento con algún objetivo.

Todos los contratos necesitan indicar en la primera línea de su archivo la versión del compilador para la que está escrito, ya que, al estar esta tecnología en continua evolución es muy cambiante por el momento y salen nuevas versión de compilación cada pocos intervalos de tiempo.

Ethereum proporciona un compilador online llamado Remix [26] en el que permite desarrollar *Smart contracts*, probarlos, depurarlos y desplegarlos [27].

Cuando se despliega un contrato o se interactúa con uno ya desplegado en la red se gasta gas, ya que los nodos deben emplear energía de computo para ejecutar las operaciones necesarias y este coste debe ser pagado por el usuario que está interactuando. Existe una tabla de costo de gas dependiendo de la instrucción que se tenga que ejecutar. Este gas se podría ver como una comisión de los nodos por realizar las operaciones necesarias a la hora de desplegar un contrato o de interactuar con él. De esta manera se “paga” a los mineros de la red como recompensa por los recursos utilizados (*hardware*, electricidad y tiempo) [28].

Estos fragmentos de código tienen ciertas ventajas, como por ejemplo: seguridad, confianza y autonomía. Gracias a ellos no es necesario recurrir a terceros para ciertas transacciones. Esto implica evitar la gestión de papeles que ocasionaría un tercero, por lo que optimizan la velocidad de las transacciones. Además es transparente para todas las partes ya que se encuentran en una red pública.

Sin embargo, cuenta con la desventaja o ventaja, según el punto de mira desde el que se enfoque, de ser inmodificable, una vez el contrato esta desplegado no puede cambiarse, es por ello que antes de desplegar el *Smart contract* definitivo tiene que ser analizado muy detalladamente y exhaustivamente probado para evitar cualquier tipo de fallo que pueda ocurrir en un futuro.

Algunas de las soluciones más conocidas, por el momento, para automatizar operaciones y gestiones gracias a este tipo de software son:

* Automatización de pagos
* Registros y cambios de propiedad
* Propiedades intelectuales
* Apuestas
* Compras automáticas
* Votaciones

A pesar del protagonismo que están ganando estos contratos inteligentes hoy en día, queda mucho por evolucionar en esta tecnología para explotar aún más sus cualidades y beneficios, por lo que, con el tiempo se abordarán nuevos casos de uso gracias a las investigaciones e inversiones que existen a día de hoy sobre los *Smart* *contracts*.

# Diseño e implementación

En esta sección se describe técnicamente el diseño completo de la plataforma ofreciendo una visión de la arquitectura del proyecto al completo y las tecnologías que han sido utilizadas para su implementación. Tal y como se ha mencionado anteriormente, se han desarrollado dos secciones bien diferenciadas, que podrían dividirse en dos subproyectos: por una parte el *front-end* y por otra parte el *back-end*.

El *front-end* del proyecto consiste en la parte visual del proyecto, en este caso la aplicación móvil para dispositivos iOS. La totalidad del proyecto ha sido implementado con el lenguaje Swift en el entorno de desarrollo Xcode.

Swift, creado por Apple y presentado en 2014, es un lenguaje de programación orientado a objetos que cuenta con ciertas características que le hacen ser un lenguaje muy potente aunque restrictivo a la hora de programar con el fin de evitar código inseguro. Comprueba automáticamente desbordamiento de enteros y administra automáticamente la memoria a través de ARC (*Automatic Reference Counting*) eliminando la necesidad de utilizar punteros. Este lenguaje está fuertemente tipado y es por ello que permite al propio compilador inferir el tipo de variables en tiempo de compilación sin asignarle un tipo por defecto. Permite enviar funciones o bloques de código como parámetros de otras funciones. Cuenta con un tipo de variable llamado Opcional que obliga al desarrollador a implementar un código seguro a partir de valores por defecto para poder compilar el código [29].

La parte de *back-end* está desarrollada a través de un *Smart contract* en la red *blockchain* de pruebas de Ethereum llamada Rinkeby. El contrato inteligente con el que se realiza toda la gestión de alquiler de coches está desarrollado en el lenguaje de programación Solidity. Solidity es un lenguaje de programación de alto nivel Turing completo.

Cuenta con una sintaxis similar a *Javascript* y con un tipado estático, que admite herencia y polimorfismo. Los *Smart contracts* se estructuran dentro de este lenguaje de manera similar a la programación orientada a objetos. Dentro de Solidity se utilizan variables y funciones como en la programación imperativa tradicional [31].

La arquitectura global de este proyecto se puede ver representada detalladamente en la siguiente imagen:

Figure 3.1 Arquitectura global del proyecto

En esta imagen que representa la arquitectura global del proyecto se pueden apreciar las diferentes partes que forman el proyecto al completo. Como se ha introducido anteriormente por una parte está la parte de *front-end* y por otro lado está la parte de *back-end*.

En primer lugar, en la parte izquierda de la figura está representada la parte de *front-end*, en este caso la app móvil para dispositivos con sistema operativo iOS, donde se encuentra la interfaz de usuario del proyecto. En la siguiente sección se describe detalladamente la totalidad del proyecto de la app, como la arquitectura que sigue el proyecto, las dependencias con las que cuenta para hacer todas las conexiones posibles y las complejidades que existen a la hora de integrar el *framework* de Web3.

En segundo lugar, en el centro de la figura se encuentran las herramientas Infura y Web3. Web3 es el *framework* que se está utilizando en la app que hace posible la conexión con la red Ethereum, en este caso con la red de pruebas Rinkeby. Realmente el framework de Web3 a través de Infura es capaz de conectarse a la blockchain Ethereum. Infura es la herramienta que hace posible la conexión y la comunicación con los nodos de la red, para poder utilizarse es necesario registrarse en la plataforma y cumplir una serie de requisitos como por ejemplo tener un proyecto creado. A través de una serie de *tokens* y *endpoints* Infura hace posible la comunicación con el Smart contract desplegado en la red *blockchain* de Rinkeby.

Por último, en la parte de la derecha de la figura se encuentra la parte de *back-end* del proyecto, que representa lo que se denomina un *DApp*. *DApp* es el acrónimo de *Decentralized Application*, es decir, una aplicación descentralizada. Estas aplicaciones consisten en un código *back-end*, en este caso el *Smart contract* que se ejecuta en una red P2P (peer-to-peer). Este tipo de programas de software no necesitan de terceros de confianza o una autoridad central para funcionar, ya que, permite la interacción directa entre usuarios y proveedores [32].

## Front-end

## Herramientas de conexiones

## Back-end o DApp

# Demo

# Conclusión

# Bibliografía

[1] S. Nakamoto, «Bitcoin: A Peer-to-Peer Electronic Cash System», p. 9.

[2] «46 días tomó Bitcoin en saltar de $6900 a $20000 en 2017, ¿Se repite la historia? – CRIPTO TENDENCIA». [En línea]. Disponible en: https://criptotendencia.com/2019/05/14/46-dias-tomo-bitcoin-en-saltar-de-6900-a-20000-en-2017-se-repite-la-historia/. [Accedido: 18-may-2019].

[3] «Gobierno japonés reinventará su registro de propiedades con blockchain», *CriptoNoticias - Bitcoin, blockchains y criptomonedas*, 23-jun-2017. [En línea]. Disponible en: https://www.criptonoticias.com/aplicaciones/gobierno-japones-reinventara-registro-propiedades-blockchain/. [Accedido: 18-may-2019].

[4] J. Pastor, «Spotify sí que cree en blockchain: así funciona Mediachain, la empresa que acaba de comprar», *Xataka*, 08-may-2017. [En línea]. Disponible en: https://www.xataka.com/empresas-y-economia/spotify-si-que-cree-en-blockchain-asi-funciona-mediachain-la-empresa-que-acaba-de-comprar. [Accedido: 18-may-2019].

[5] M. Orcutt, «Blockchain technology will revolutionize medical records—just not anytime soon», *MIT Technology Review*. [En línea]. Disponible en: https://www.technologyreview.com/s/608821/who-will-build-the-health-care-blockchain/. [Accedido: 18-may-2019].

[6] «Transparencia electoral: 5 plataformas blockchain para votaciones», *CriptoNoticias - Bitcoin, blockchains y criptomonedas*, 06-may-2018. [En línea]. Disponible en: https://www.criptonoticias.com/colecciones/transparencia-electoral-5-plataformas-blockchain-para-votaciones/. [Accedido: 18-may-2019].

[7] «Ethereum», *ethereum.org*. [En línea]. Disponible en: https://ethereum.org. [Accedido: 19-may-2019].

[8] «Rinkeby: Ethereum Testnet». [En línea]. Disponible en: https://www.rinkeby.io/#stats. [Accedido: 15-may-2019].

[9] «car2go carsharing España», *car2go*. [En línea]. Disponible en: https://www.car2go.com/ES/es/. [Accedido: 19-may-2019].

[10] «Muévete de forma sostenible e inteligente por Madrid con emov». [En línea]. Disponible en: https://www.emov.eco/. [Accedido: 19-may-2019].

[11] D. Tapscott, A. Tapscott, y J. M. Salmerón, *La revolución blockchain: descubre cómo esta nueva tecnología transformará la economía global*. Barcelona: Deusto, 2018.

[12] K. Christidis y M. Devetsikiotis, «Blockchains and Smart Contracts for the Internet of Things», *IEEE Access*, vol. 4, pp. 2292-2303, 2016.

[13] BUTERIN, Vitalik, «Ethereum white paper». ethereum, 2013.

[14] I. Mendívil, «El ABC de los Documentos Electrónicos Seguros», p. 28.

[15] «Dapps movilizaron US$ 6.700 millones en 2018», *CriptoNoticias - Bitcoin, blockchains y criptomonedas*, 18-ene-2019. [En línea]. Disponible en: https://www.criptonoticias.com/comunidad/arte-entretenimiento/dapps-movilizaron-usd-6700-millones-2018/. [Accedido: 15-may-2019].

[16] «Criptonoticias», *CriptoNoticias - Bitcoin, blockchains y criptomonedas*, 27-abr-2019. [En línea]. Disponible en: https://www.criptonoticias.com. [Accedido: 15-may-2019].

[17] «Litecoin - La moneda electrónica». [En línea]. Disponible en: https://litecoin.org/es/. [Accedido: 15-may-2019].

[18] «Monero», *getmonero.org, The Monero Project*. [En línea]. Disponible en: https://getmonero.org/index.html. [Accedido: 15-may-2019].

[19] «Hyperledger – Open Source Blockchain Technologies», *Hyperledger*. [En línea]. Disponible en: https://www.hyperledger.org/. [Accedido: 15-may-2019].

[20] «r3.com», *r3.com*. [En línea]. Disponible en: https://www.r3.com/. [Accedido: 15-may-2019].

[21] «Ripple - One Frictionless Experience To Send Money Globally», *Ripple*. [En línea]. Disponible en: https://ripple.com/. [Accedido: 15-may-2019].

[22] «BigchainDB • • The blockchain database.», *BigchainDB*. [En línea]. Disponible en: https://www.bigchaindb.com/. [Accedido: 15-may-2019].

[23] «The Solution», *Evernym*. [En línea]. Disponible en: https://www.evernym.com/solution/. [Accedido: 15-may-2019].

[24] «Cuestiones básicas de Ethereum | Billetera de Blockchain». [En línea]. Disponible en: https://www.blockchain.com/es/learning-portal/ether-basics. [Accedido: 09-may-2019].

[25] D. G. Wood, «ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER», p. 32.

[26] «Remix - Solidity IDE». [En línea]. Disponible en: https://remix.ethereum.org/#optimize=false&version=soljson-v0.5.1+commit.c8a2cb62.js. [Accedido: 19-may-2019].

[27] «Estructura y elementos de un contrato», *APRENDE BLOCKCHAIN*, 28-feb-2018. .

[28] «¿Qué es y para qué sirve el “Gas” en Ethereum?», *Ethereum*, 25-ene-2018. .

[29] C. G. García, J. P. Espada, B. C. P. G. Bustelo, y J. M. C. Lovelle, «Swift vs. Objective-C: A New Programming Language», *IJIMAI*, vol. 3, n.o 3, pp. 74-81, 2015.

[30] A. Teller, «Turing completeness in the language of genetic programming with indexed memory», en *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*, Orlando, FL, USA, 1994, pp. 136-141.

[31] M. Wohrer y U. Zdun, «Smart contracts: security patterns in the ethereum ecosystem and solidity», en *2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE)*, Campobasso, 2018, pp. 2-8.

[32] W. Warren y A. Bandeali, «0x: An open protocol for decentralized exchange on the Ethereum blockchain», p. 16.

asdasd

###### Enter Your Appendix Title Here

Appendices must be identified by letters (A, B, etc.) rather than by numbers. For this reason, different style headings are used with appendices. (The style at the top of this page is “Appendix A - Heading 6.”)

First-level Subhead (Heading 7 style)

Within an appendix, Heading 7 is the style to use for all first-level subheads. If you need to add another subhead level within Heading 7, use Heading 8 as shown below.

Second-level Subhead (Heading 8 style)

Use Heading 8 for all second-level subheads within an appendix. If you need to add another subhead level within Heading 8, use Heading 9 as shown below.

Third-level Subhead (Heading 9 style)

If you need a third-level subhead in an appendix, use Heading 4.

Figures and Tables Within Appendices

When you first add a figure or table to an appendix, it will be numbered as though it were in a regular chapter. For example, when the figure below was first inserted, it became “Figure 4.1”. As a figure within Appendix A, it should be “Figure A.1”

To make this change, the codes in the caption labels must be modified, and it’s best to wait until all figures and tables have been added to appendices.

Figure A.1 First Figure in Appendix A

MP00640_

###### Enter Your Appendix Title Here

If you need additional appendices, use style “Appendix A – Heading 6” for the appendix heading. This will label appendices in alphabetical order (A, B, C, etc.).

1. Una función *hash* es una función computable mediante un algoritmo. Es una operación criptográfica que genera una especie de firma digital de un contenido. [↑](#footnote-ref-1)