

UNIVERSIDAD DE ALCALÁ



Escuela Politécnica Superior

MÁSTER EN ETHEREUM, TECNOLOGÍA BLOCKCHAIN Y CRIPTOECONOMÍA

Trabajo Fin de Máster

**DISEÑO E IMPLEMENTACIÓN DE UNA DAPP
PARA OFRECER SERVICIOS DE RENTING/LEASING
PARA EL SECTOR AUTOMOVILÍSTICO**

Pedro Cerón
Omar Lozano
Rafael Pérez
2019

UNIVERSIDAD DE ALCALÁ

Escuela Politécnica Superior

**DISEÑO E IMPLEMENTACIÓN DE UNA DAPP
PARA OFRECER SERVICIOS DE RENTING/LEASING
PARA EL SECTOR AUTOMOVILÍSTICO**

Trabajo Fin de Máster

Autor: Pedro Cerón / Omar Lozano / Rafael Pérez

Director: Alberto Ballesteros Rodríguez

Tribunal:

Presidente:

Vocal 1º:

Vocal 2º:

Calificación:

Fecha: de abril de 2019

A todos nuestros profesores, que nos han embarcado y guiado en un viaje a un mundo tan desconocido como apasionante, y a nuestros compañeros por habernos acompañado en el viaje.

Resumen

Blockchain es una de las tecnologías disruptivas que forman parte de lo que se conoce como Industria 4.0, y está siendo una de las palancas de cambio que están haciendo que todas las empresas, sean del sector que sea, estén cambiando sus negocios y maneras de operar.

Gracias a Blockchain se apuesta por la descentralización y por la inmediatez, así como por la seguridad y la inmutabilidad de la información. Blockchain apuesta por dar el control a los usuarios.

A través de las próximas líneas, se presenta el resultado de los conceptos técnicos y prácticos obtenidos durante el último año como resultado del estudio del Máster en Ethereum, Tecnología Blockchain y CriptoEconomía, y en el que se han abordado todos los temas claves para entender esta tecnología que tanto está dando que hablar, poco conocida o entendida en general por los usuarios, y con tanto recorrido durante los próximos años.

Además, se ha pretendido establecer un solución a un nicho de mercado como es el crédito al consumo de particulares y la financiación de empresas, que pasa por ser uno de los mercados donde las entidades financieras más foco está poniendo, con la aparición de numerosas iniciativas y opciones, y que va ganando peso en los balances contables y porcentajes de operaciones en curso de las empresas.

Así mismo, se presenta una plataforma con un primer alcance acotado, pero con mucho margen de desarrollo y mejora y, sobre todo, con la posibilidad de combinación con el resto de tecnologías disruptivas actuales, como IoT (para el cálculo y parametrización de ofertas “Insurance as a Service”), Data Analytics (para el análisis y estudio de los patrones de comportamiento de los clientes) o RPA (para la automatización de procesos y operativas manuales actuales).

Índice

Resumen	4
Índice	5
Introducción	7
Parte Teórica	8
Blockchain	8
Ethereum	10
Smart Contract	12
Renting/Leasing	13
Blockchain y Renting/Leasing	14
Objetivos y requisitos funcionales	16
Calendario Ejecución proyecto	17
Reparto de tareas y asignación de roles durante el proyecto	17
Primera aproximación de la solución tentativa	19
Módulos incluidos en el alcance del proyecto	20
Parte Técnica	21
Repositorio Aplicación	21
Explicación técnica	21
Modelo Económico	23
Modelo Económico Fase I	23
Presupuesto de la fase de desarrollo	23
Presupuesto de la fase de implantación y uso	25
Conclusiones	27
Trabajos futuros	28
Bibliografía	30
Anexo I - Manual del Usuario	31
Alta Usuarios	31
Usuarios Clientes	31
Usuarios Empresa Aseguradora	34
Usuarios Empresa financiera	36
Usuarios	38
Administrador/Owner	38
Administración de usuarios	39
Inicialización de Tokens	39
Panic Buttons	40
Renting	41
Alta Coches	41
Validación/Entrega Coches	43
Cliente	45

Comprar Tokens	45
Financiar Tokens	46
Pagar Financiación	47
Rentar Coche	48
Entregar Coche	50
Empresa Aseguradora	51
Modificar precios	51
Empresa financiera	53
Comprar Tokens	53
Anexo II - Detalle técnico	54
Mappings	54
Funciones de los contratos	56
Frontales	60

I. Introducción

Aunque quizá no seamos conscientes, nos encontramos siendo actores principales de un cambio en el paradigma sobre cómo hacer las cosas, fruto especialmente de dos acontecimientos que cambiaron el mundo:

- Por un lado, la fuerte crisis económica de los años 2.000.
- Por otro lado, la revolución tecnológica en la que estamos inmersos.

Como resultado del primer punto, la gente reclama más y más un mundo descentralizado y que nuestro dinero, acciones y riesgo no dependa de personas y entidades que puedan llevarnos a repetir situaciones anteriores como resultado de la mala praxis de estas. Es por ello por lo que se está creando un movimiento de descentralización siendo el sector financiero el mayor afectado por ello.

Consecuentemente, se ha experimentado un descenso en cuanto a demanda de crédito o deuda por parte de los particulares a las entidades tradicionales financieras para satisfacer sus necesidades. No obstante, este sigue siendo un mercado con un movimiento muy alto y las entidades financieras están trabajando en la explotación de fórmulas y nuevos productos para, no solo mantener los niveles, si no aumentarlos [1].

Y esto, combinado con este momento tan disruptivo a nivel tecnológico, está naciendo y consolidándose un nuevo modelo de negocio apoyado en las nuevas tecnologías (Blockchain, Cloud, IA, IoT...) y la aparición de nuevos actores que se salen de los patrones clásicos establecidos, ya sean en el negocio bancaria (empresas Fintech), en el sector asegurado (Insurtech), regulatorio (Regurtech), etc.

En este proyecto de fin de máster se presenta una aplicación para la gestión de operaciones de rentals/leasing de automóviles sobre blockchain, sentando además las bases para ampliar sus funcionalidades y operativa.

Dado que se trata de un diseño inicial, y lo que se desea presentar es la idea principal, se ha de ceñir el alcance del proyecto a las funcionalidades que se consideran básicas. A medida que avance el tiempo es posible complementar este diseño con más servicios y funcionalidades.

II. Parte Teórica

Blockchain

Blockchain es una de las tecnologías que están en el selecto grupo de tecnologías disruptivas sobre las que se tiene puesta las esperanzas para formar parte de la revolución tecnológica en la que nos encontramos.

Concretamente, se definiría como “**una base de datos que se halla distribuida entre diferentes participantes, protegida criptográficamente y organizada en bloques de transacciones relacionadas entre sí matemáticamente**” (Álex Preukschat. Blokchain: La revolución industrial de Internet. Cap1) [2]. Es decir:

- **Distribuida:** como otros servicios p2p existentes e históricos, la base de datos se encuentra compartida entre todos los nodos que conforman la red. De esta manera, se consigue una de sus grandes ventajas como es la tolerancia a los fallos, permitiendo que no sólo haya un punto de control reduciendo el riesgo de ataque y caída de todo el sistema.
- **Protegida Criptográficamente:** toda la información que se almacena está cifrada utilizando técnicas y algoritmos criptográficos, de tal modo que permite asegurar la incorruptibilidad, transparencia, trazabilidad y el fraude en cuanto a la información almacenada.
- **Relacionados entre sí matemáticamente:** el minado de los bloques y por tanto “cierre” de la información almacenada, deriva del uso de algoritmos matemáticos como la “Prueba de Trabajo” (Proof of Work en inglés).

Blockchain es una de las tecnologías innovadoras que han venido para cambiar el mundo y revolucionar todos los sectores económicos y, especialmente, el mundo financiero. ¿Por qué? Debido fundamentalmente a una serie de puntos que rompen con el modelo de negocio que hasta ahora se ha conocido, siendo los principales entre otros:

1. Desintermediación
2. Confiabilidad
3. Seguridad

Desintermediación

Quizá el punto más clave de todos. Como se ha avanzado antes, los acontecimientos sucedidos a nivel mundial durante los últimos años principalmente consecuencia de la brutal crisis económica, los usuarios han perdido confianza en los sistemas tradicionales y demandan una mayor propiedad de su información y, por consiguiente, su dinero.

Además, este sistema tan obsoleto, presenta una serie de retos por resolver [3]:

1. Costes excesivos de las operaciones y transacciones.
2. Tarifas poco ajustadas respecto a las expectativas de clientes y usuarios.

3. Frecuentes retrasos, que restan eficacia a los trámites y encarecen las gestiones.
4. Falta de seguridad en un sistema que queda vulnerable al fraude y los ataques.

Para ello, Blockchain se presenta como la solución perfecta. Eliminando a los intermediarios actuales, permite que sean los propios usuarios quienes (como pasa en el caso de las redes sociales) tejan las redes de conexión, y sean los propios usuarios quienes decidan qué hacer y cómo hacer las cosas.

Este último punto es quizás a día de hoy uno de las mayores barreras que juegan en contra de la evolución, tal y como se comenta en el siguiente artículo [4]: *"El problema al que se enfrenta BitCoin, y otras monedas similares, es el usual en este tipo de cambios de paradigma: la falta de confianza. La gente se siente razonablemente segura con el sistema actual, y los ahorros son algo demasiado serio para hacer apuestas. Una red financiera descentralizada de la que nadie (o todos) se hace responsable transmite la sensación de que o bien no hay ningún tipo de control o , aún peor, de que alguien la controla entre bambalinas sin nuestro conocimiento."* Y es aquí donde se enlaza con el siguiente punto clave.

Confiabilidad

Una de las características claves de Blockchain es que permite la interacción y la generación de transacciones entre pares, sin necesidad de intermediarios y sin necesidad de establecer lazos confiables entre ellos. Es decir, se da la posibilidad de hacer cualquier tipo de transacción de cualquier tipo (económica, datos, etc..) sin conocer a la otra persona pero con la seguridad de que *"los datos de las transacciones son imposibles de falsificar una vez registrados"* [5]. Y esto es porque Blockchain aporta además la seguridad necesaria para que no se puedan producir fraudes económicos, aportando la transparencia que se viene demandando en el sector financiero especialmente tras los acontecimientos acaecidos con motivo de la crisis financiera sufrida.

Seguridad

Aunque quizá este punto sea uno de los más pendientes de desarrollar, y por supuesto Blockchain no se libra de los numerosos ataques de robos de criptomonedas, Blockchain ofrece especialmente seguridad desde el punto de vista de fraude económico. Es decir, Blockchain evita lo que comúnmente se puede llamar "Contabilidad B". Es decir, seguridad económica. Todas las transacciones quedan registradas de manera inmutable gracias a la aplicación de técnicas criptográficas. Y no sólo para evitar fraude en cuanto a manipulación o malversación de transacciones, sino para prevención de modificación de datos, preservar la privacidad de la identidad digital, ...

Por todo esto, además por supuesto de otros puntos claves, el sector financiero al igual que otros sectores tiene que adaptarse a las nuevas posibilidades que ofrece Blockchain, en la medida que están experimentando en los últimos años una gran revolución con la aparición de estas nuevas tecnologías y la aparición de nuevas start-ups que está revolucionando un sector que se pensaba que era tan estabilizado en unos principios como hasta ahora. Un ejemplo de algunas de start-ups son Transferwise, Simple, Atom Bank...

Ethereum

Desarrollada por Vitalik Buterin como una evolución y mejora de Bitcoin, Ethereum es *un protocolo, tecnología o plataforma descentralizada y de código abierto que ejecuta programas llamados smart contracts, y que utiliza blockchain para sincronizar y almacenar los cambios de estado del sistema, junto con una criptomoneda llamada ether para medir y restringir los costos de los recursos de ejecución.* (Mastering Ethereum.Andreas Antonopoulos, Gavin Wood Ph.D) [6].

Como se ve en la definición, en esencia no deberá distar mucho respecto al protocolo Bitcoin más allá de la posibilidad de programación y uso de Smart Contracts además de utilizar una criptomoneda propia, pero en detalle se diferencian entre otros en [7][8]:

- Lo primero es el propio diseño u objetivo de los propios protocolos. Mientras que Bitcoin fue diseñado para convertirse en *un sistema de contabilidad seguro e immutable fuera del control del sector financiero*, Ethereum está diseñado como una "computadora descentralizada del mundo" donde la funcionalidad Turing-completa permite a los usuarios crear y ejecutar aplicaciones en la red a través de la Máquina Virtual de Ethereum. (EVM). Es decir, Ethereum está diseñado explícitamente para facilitar los contratos inteligentes completos de Turing y las aplicaciones descentralizadas en su red.
- A fecha de abril de 2019, Bitcoin crea 12.5 bitcoins nuevos cada 10 minutos, habiéndose establecido un límite fijo de 21 millones. Ethereum crea 3 ethers cada 15 segundos, pero sin tope fijo.
- Los bloques son creados, de media, cada 15 segundos mientras que en Bitcoin estos llevan un promedio de 10 minutos. Además, los bloques de Bitcoin tienen un tamaño máximo de 1 MB, pudiendo procesar 4 transacciones por segundo por las 15 de Ethereum.
- Los bloques son creados, de media, cada 15 segundos mientras que en Bitcoin estos llevan un promedio de 10 minutos. Además, los bloques de Bitcoin tienen un tamaño máximo de 1 MB, pudiendo procesar 4 transacciones por segundo por las 15 de Ethereum.
- El algoritmo de hash de Ethereum es Ethash, mientras que Bitcoin usa el SHA-256D.
- En cuanto al proceso de minería, Ethereum aún usa minería similar a Bitcoin en un esquema de PoW, está en proceso de decidir si cambiar por el uso de "Proof of Stake" (PoS).

En el siguiente cuadro comparativo se puede ver en detalle las diferencias entre uno y otro protocolo [9]

	BITCOIN	ETHEREUM
NACIMIENTO DE LA PLATAFORMA	18 de agosto de 2008 (registro del dominio 'Bitcoin.org'). 31 de octubre de 2008 fecha de su White Paper.	Diciembre de 2013
FECHA 1ER BLOQUE MINADO	3 de enero de 2009	30 de Julio de 2014
CREADOR DE LA PLATAFORMA	Satoshi Nakamoto, del cual no se sabe quién es o quiénes son (en caso de pseudónimo de una organización)	Vitalik Buterin; Otros co-fundadores incluidos Gavin Wood y Joseph Lubin
FUNCIÓN PRINCIPAL DE LA PLATAFORMA	Sistema de pago descentralizado, rápido y seguro, al igual que su propia moneda.	Plataforma de ejecución de contratos inteligentes y aplicaciones descentralizadas (dApps)
TECNOLOGÍA USADA	Blockchain (Cadena de bloques)	
REDES USADAS	Mainnet (Red principal) y Testnet (Red de prueba)	
ALGORITMO DE HASH	SHA256D	Ethash
HARDWARE CORRECTO PARA LA MINERÍA	ASIC	GPU y CPU
LENGUAJE DE PROGRAMACIÓN	C++	C++, Python, GoLang...
SE PUEDEN CONSIDERAR	Criptomonedas descentralizadas	
USO CRIPTOMONEDA	Pagos. Competir con las divisas fiat y el oro. También como inversión	Operar dentro de la red Ethereum: crear aplicaciones descentralizadas y ejecutar contratos inteligentes. También como inversión
CRİPTOMONEDA	Bitcoin (BTC)	Ether (ETH)
DECIMALES	8	18
CANTIDAD MÁXIMA A EMITIR DE CRİPTOMONEDA	21 millones de bitcoin en total, por lo tanto, deflacionaria	18 millones por año, por lo tanto, inflacionaria
CREACIÓN DE CRİPTOMONEDAS A TRAVÉS DE	Minería	
SISTEMA DE MINERÍA	Proof of Work (PoW) o Prueba de Trabajo	
CANTIDAD DE RECOMPENSA DE LA MINERÍA	Actualmente 12,5 bitcoin por bloque. Cada 210.000 bloque decrece a la mitad	3 Ether por bloque desde la introducción de la etapa Metrópolis. Anteriormente fue de 5 Ether por bloque
MÉTODO DE RECOMPENSA DE LOS MINEROS	Por validación de bloques	Por validación de bloques, de transacciones y por ejecución de contratos inteligentes
PROCESAMIENTO DE LOS BLOQUES	Cada 10 minutos (600 segundos)	Cada 16 segundos
TAMAÑO DE LOS BLOQUES	1 Mb como máximo	Sin definir, pero muy por debajo de 1 Mb. No obstante, el tamaño de los bloques está fijado por el límite del Gas del mismo
RECÁLCULO DE LA DIFICULTAD DE MINADO	Cada 2016 bloques minados	Cada bloque minado
COSTE TRANSACCIÓN	Depende del fee (comisión) por transacción	Depende del Gas

Smart Contract

Los Smart Contract, como se ha referido anteriormente, son el principal punto de atracción de Ethereum, y la posibilidad que abrió a su programación.

Un Smart Contract no es otra cosa que la redacción de un contrato “normal” físico bajo lenguaje y código de programación, de tal manera que trabaje de manera independiente y por tanto se ejecute cuando se cumplan las condiciones establecidas y pactadas en el código.

Es decir, a través del uso de Smart Contract se eliminan intermediarios, quedando como agentes del contrato los propios intervenientes e interesados.

Las principales características de los smart contracts serían:

- Precisión
- Rendición de cuentas (las partes implicadas saben en todo momento en que estado se encuentra el contrato)
- Velocidad
- Seguridad
- Consistencia

No obstante, también presenta una serie de limitaciones, entre las que destacan:

- **Errores de código.** Es decir, al ser programados los Smart Contracts, se incluyan errores accidentales que puedan alterar el funcionamiento y el resultado del contrato.
- **Lógica incorrecta.** Deliberadamente o sin querer, pero muy análogo al anterior no existe una comprobación de que el programador pueda manipular intencionadamente el contrato. Por ejemplo, que en caso que se cumpla una condición para hacer una transferencia, esta se dirija al wallet del programador en vez de a la del beneficiario.
- **Regulación e impuestos.** Esto es prácticamente para todas las plataformas Blockchain, y casi todas las tecnologías innovadoras. En el caso anterior, ¿Cómo se puede reclamar el fraude? ¿Quién lo regula y sanciona? Y en el caso de las transferencias, ¿qué impuestos habría que pagar por una transacción económica, como por ejemplo el traspaso de un activo?

Renting/Leasing

Un producto de renting o leasing son productos de financiación especializada que nacieron como apoyo a las empresas para financiar las adquisiciones y uso de bienes, especialmente en cuanto a maquinaria, vehículos, inmobiliarios, etc...

Básicamente, son contratos de arrendamiento de bienes a largo plazo entre dos partes y en las que se pacta, entre otros, el pago de cuota periódica (mensual por lo general), el pago o no de una cuota inicial y el pago de una cuota final al final del contrato.

¿Qué diferencias hay? Por un lado, la principal es que uno se trata de un arrendamiento operativo (renting) mientras que el otro es un arrendamiento financiero (leasing). Esto en la práctica [10], consiste en que para las operaciones de renting, la formalización contractual se basa en un consentimiento entre ambas partes siendo la empresa de renting la propietaria del producto arrendado.

Por su parte, un leasing al tratarse de un arrendamiento financiero regulado por ley, se formaliza a través de una entidad de crédito, transfiriendo los riesgos y beneficios derivados de la propiedad. Además, es obligatorio la inclusión de un derecho de compra, siendo opcional en los contratos de renting.

Otra diferencia clave, es que mientras que los contratos y servicios de leasing están disponibles únicamente para empresas, las operaciones de renting también se encuentran disponibles y son operativas para particulares.

Blockchain y Renting/Leasing

Blockchain es una tecnología con un previsible gran crecimiento dentro de los sectores logístico, financiero y de seguros [11] [12]. Las ventajas fundamentales que puede ofrecer a un servicio de renting/leasing de coches (en donde se pueden integrar los 3 sectores) se pueden resumir en las siguientes [13]:

- **Trazabilidad de todo el proceso:** Tanto las terceras partes (compañías de financiación, aseguradora, recambios, etc.) como el cliente final pueden comprobar en tiempo real los cambios de estado del bien.
- **Confianza:** Como consecuencia del punto anterior y debido a la inmutabilidad de la cadena de bloques, las partes interesadas pueden tener la certeza de que el proceso se ejecuta de forma correcta y legal. Uno de los beneficios fundamentales que puede aportar este aspecto es la fidelización del cliente final (si además el servicio prestado es bueno).
- **Precios justos:** El cliente final sabe el precio exacto de los bienes por los que está consumiendo un servicio (por ejemplo: al poder tener acceso a la información de la cadena de bloques, puede saber a qué precio el taller adquirió los repuestos y qué beneficio obtiene éste por su servicio).
- **Interconectividad:** Mediante la utilización de dispositivos IoT que envíen datos de geolocalización y del movimiento de los vehículos a la Blockchain, se puede crear un sistema de seguro que repercute únicamente en el cliente final en función de la utilización del vehículo y del tipo de vía por la que transita.

Por otra parte, en el mercado existen aplicaciones basadas en tecnología Blockchain que intentan dar solución a la problemática de la creación de una Dapp que dé soporte a los servicios de leasing en el entorno automovilístico. Cabe destacar entre ellas, la desarrollada por IBM en julio del 2016 [15]. Dicha aplicación hace uso de una plataforma Blockchain integrada en un entorno cloud. A diferencia de otras plataformas, en **IBM Blockchain Platform** [15]:

1. Estamos ante una red permissionada.
2. No requiere el uso de criptodivisas.
3. Las transacciones son confidenciales y visibles a partes seleccionadas.

The Car Dossier Approach

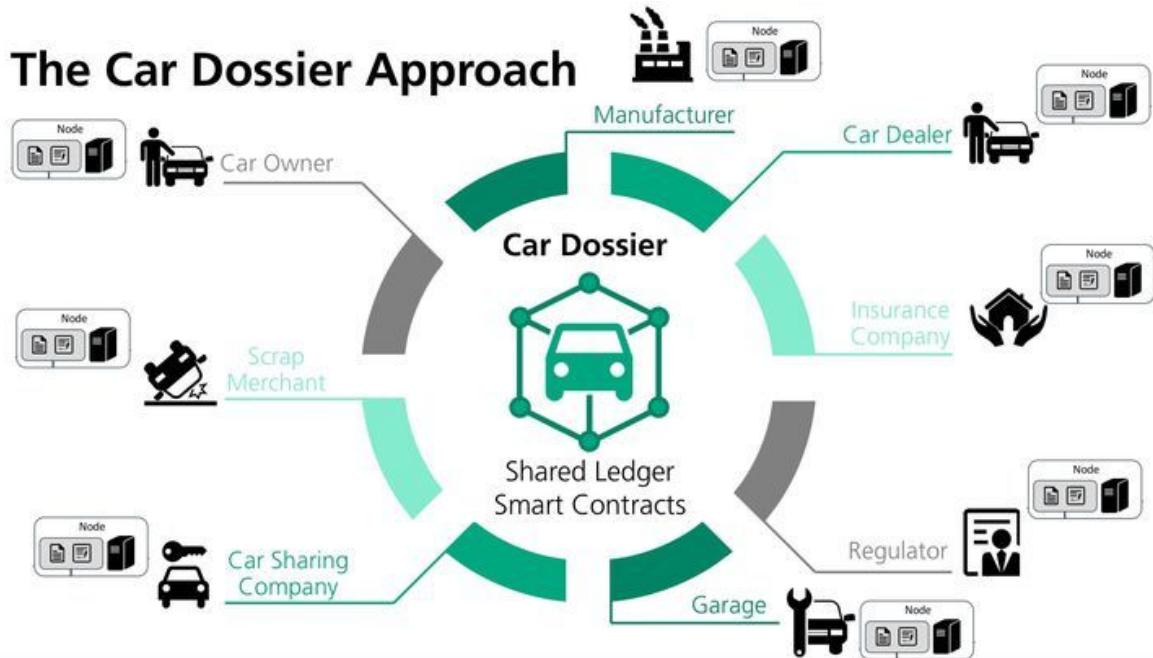


Imagen: Modelo Relación Actores Ciclo Vida Renting Coche.

Fuente: https://d5cplpsrt2s33.cloudfront.net/m/7a688abb11b5a7de/ASPECT_768-blog_comms_car_dossier_web.jpg

III. Objetivos y requisitos funcionales

De cara al desarrollo de esta plataforma, y en base a los plazos de ejecución, se ha planteado un desarrollo faseado, siendo el objetivo principal de esta primera fase correspondiente al TFM la elaboración de una Dapp que permita gestionar a través de smart contracts los contratos que se firman entre cliente, entidad financiera y entidad aseguradora. Esto incluye no sólo la programación de estos, si no el desarrollo del front de acceso y gestión.

Es decir, en esta primera fase se ha planteado los siguientes objetivos de desarrollo:

- **Creación de una Dapp** que permita interconectar a todos los distintos usuarios y agentes que participan en la gestión del ciclo de vida de una operación de renting de automóviles. En esta primera fase, el objetivo es la construcción de los elementos considerados básicos (y ampliable en futuras fases con el desarrollos de nuevos módulos, de cara a futuro de poder integrar todo el proceso en una misma plataforma) y que incluye especialmente la posibilidad de firma y automatización de la ejecución de la ejecución de los contratos entre usuarios. La Dapp cumplirá con los siguientes requisitos:
 - **Funcionará en servidor local.**
 - **Acceso a la aplicación a través de la URL** correspondiente.
 - **Las transacciones se firmarán con Metamask.**
 - **Se utilizarán eventos** para guiar al usuario final.
 - Se realizará la **implementación de librerías** que se ajusten a las necesidades del desarrollo.
 - Se implementarán **medidas de seguridad**. (Ej. emergency stop).
- **Creación de fronts para acceso y gestión** (amigables para el usuario final).
- **Desarrollo de smart contracts** con las siguientes relaciones:
 - Smart contract entre usuario y empresa de renting
 - Smart contract entre empresa de renting y entidad financiera
 - Smart contract entre usuario y entidad aseguradora
 - Los Smart Contracts serán sometidos a tests.

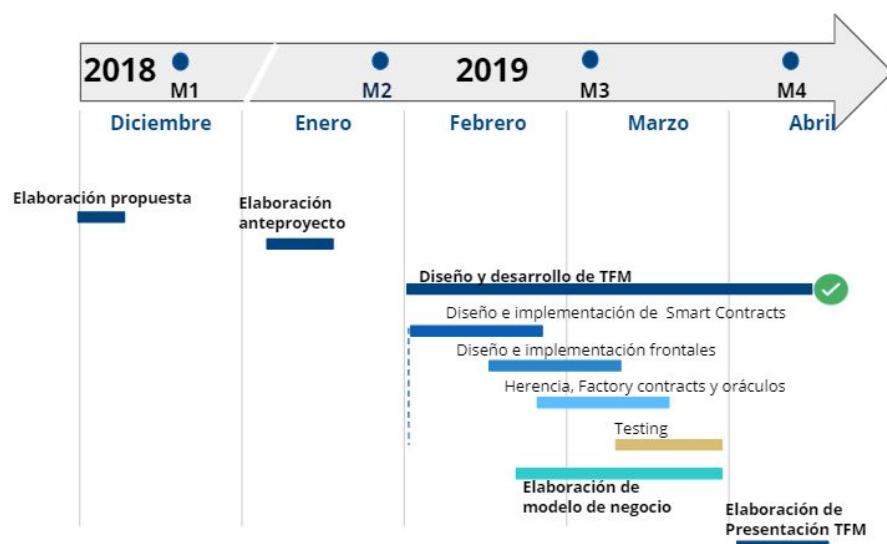
El desarrollo y consecución de los objetivos marcados, permite aportar el siguiente valor añadido:

- Creación de una DAPP novedosa (y con pocos ejemplos de implementación en el mercado actual).
- Análisis de la viabilidad de negocio de la solución.
- Implementación de medidas de seguridad (en el código) contra:
 - Race Conditions

- Orden de las transacciones
- Overflow's
- DoS
- Forcibly sending
- Creación de un modelo de básico de riesgo del usuario.

Calendario Ejecución proyecto

A continuación se presenta la planificación que se estableció y se ha seguido para la ejecución del proyecto:



Reparto de tareas y asignación de roles durante el proyecto

Respecto a la distribución de tareas entre los miembros del equipo, la estimación quedaría de la siguiente forma:

- **Rafael Pérez Arias**

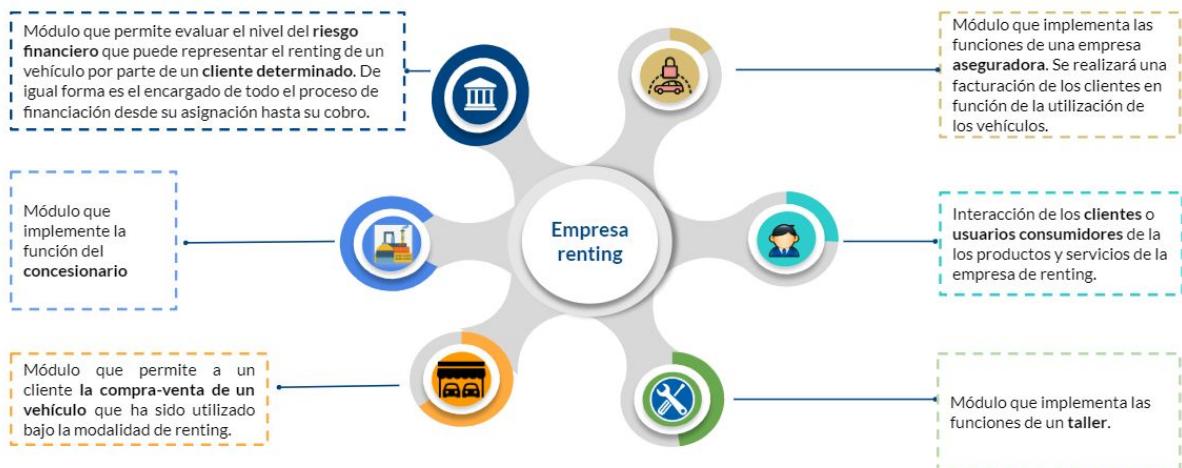
- Elaboración del modelo de negocio.
- Desarrollo de la memoria del proyecto.
- Elaboración de la presentación del TFM.
- Soporte en las pruebas unitarias de funcionalidad de los frontales.

- **Pedro Cerón Colás**

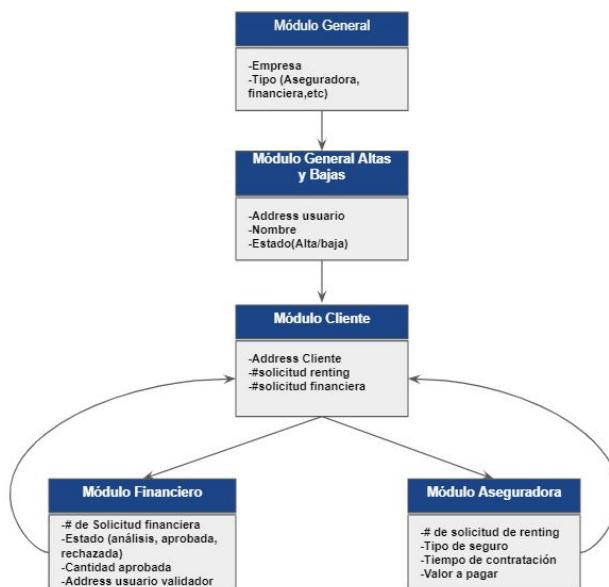
- Frontal del módulo general y frontal del módulo financiero.

- JavaScript de módulos general y financiero.
 - Desarrollo de las funciones que se integrarán dentro de los .sol para los módulos general y financiero, así como desarrollo en JavaScript de los testing de funcionalidad para los módulos general y financieros.
 - Implementación de mecanismo de circuit breaker y Factory Contract.
 - Análisis de seguridad en los contratos, integrando medidas contra:
 - Ataques de Overflow's
 - Ataques DoS
 - Soporte en el desarrollo de la memoria.
- **Omar Orlando Lozano:**
 - Frontal del módulo asegurador y frontal del módulo cliente.
 - JavaScript de módulos asegurador y cliente.
 - Desarrollo de las funciones que se integrarán dentro de los .sol para los módulos asegurador y cliente., así como desarrollo en JavaScript de los testing de funcionalidad para los módulos asegurador y cliente.
 - Integración de librerías de OpenZeppelin en el proyecto.
 - Análisis de seguridad sobre los contratos, integrando medidas de seguridad contra:
 - Race Conditions
 - Orden de las transacciones
 - Forcibly sending
 - Soporte en el desarrollo de la memoria.

Primera aproximación de la solución tentativa



Aproximación inicial: Modelo Inicial Entidad Relación



Aproximación inicial: Smart Contracts

Empresas	Empresa Financiera
function altaEmpresa function actualizarEmpresa function altaUsuario function actualizarUsuario	function pagoMensual function pagoFinal function estadoFinanciación
Cliente	Empresa Aseguradora
function altaCliente function actualizarCliente function altaFinanciacion function actualizarFinanciacion function altaSeguro function actualizarSeguro function altaRecord function actualizarRecord	function pagoFinal function límiteSeguro function estadoSeguro

Módulos incluidos en el alcance del proyecto

Módulo General

- Este módulo permite a los administradores de la Dapp realizar el alta de usuarios, de las empresas de seguros y empresas financieras.
- Es decir, es el módulo de administración donde los usuarios administradores pueden realizar todas las tareas de administración. En esta, primera fase:
 - Para cada uno de los módulos, se ha implementado la gestión de perfiles. Esto permite garantizar que sólo aquellos usuarios pertenecientes a la empresa, puedan validar las operaciones de su respectiva empresa y no de cualquiera de las demás que participan en el flujo del proceso. Para cumplir con este objetivo se ha realizado una asignación de privilegios vinculado con el address de cada uno de usuarios.
 - Además, desde el módulo de administración, los usuarios con permisos de administración de la aplicación pueden realizar tareas de mantenimiento de contratos, vehículos, empresas, etc...

Módulo Financiero

- Para permitir a los usuarios y empresas financieras desarrollar sus funciones dentro del flujo de renting, se ha desarrollado una pantalla o frontal, la cual facilita registrar las solicitudes de financiación de los clientes. De esta forma de acuerdo con la información suministrada por el cliente y su histórico, se puede proceder con el respectivo análisis de riesgo (mediante un modelo básico que tiene en cuenta el histórico del cliente, la información que se suministra por parte del cliente por pantalla, dando como resultado el scoring asociado a ese cliente) que devuelve el número máximo de tokens a solicitar prestados por parte del cliente. Una vez se ha definido el análisis de riesgo se realiza a través del mismo frontal la validación/aprobación de las solicitudes.

Módulo Cliente

- Este módulo es de vital importancia para el funcionamiento del flujo del sistema. Para tener un control de los usuarios y un registro de sus acciones, se ha desarrollado una pantalla o frontal para realizar el alta del cliente.
- De igual forma una vez el cliente ya se ha registrado, este puede acceder al portfolio de productos de renting, así como la posibilidad de utilizar financiación y seleccionar el tipo de seguros que más le convenga. Para ello se ha desarrollado un frontal en el que se reúnan dichos productos y que permite al cliente realizar la solicitud de un coche, seleccionar el tipo de seguro y las condiciones del renting (se valida a través de smart contracts, que el usuario se encuentre registrado, cuente con un saldo suficiente y no tenga un histórico negativo).

IV. Parte Técnica

Repository Aplicación

En el siguiente enlace se encuentra el código fuente de la aplicación:

[Dapp Leasing-Renting](#)

Explicación técnica

El detalle del análisis de la parte técnica se ha dividido en 5 secciones distintas que se encuentran en el [anexo II](#) de este documento. Dichas secciones son las siguientes:

- **Mappings:** Conjunto de mappings utilizados en los contratos para almacenar la información de usuarios, empresas, coches, owner de empresa de Leasing y precios por empresa aseguradora.
- **Funciones:** Conjunto de funciones utilizadas en los contratos para desarrollar funcionalidades de:
 - Gestión de usuarios y modificaciones de sus datos de perfil.
 - Transacciones, envío de tokens y verificaciones de balances.
 - Limitaciones en el número de tokens asignados.
 - Carga y recuperación de datos de los mappings.
 - Seguridad y control de contratos (por ejemplo: desactivación de emergencia de contratos y controles de desbordamientos por operaciones aritméticas).
 - Modificaciones de los precios de las aseguradoras.
 - Recuperación de datos de los mappings.
 - Seguridad y control en la ejecución de los contratos.
- **Frontales:** Desarrollado a partir de código en html y javascript. Son el conjunto de pantallas que se encargan de gestionar la interacción con el usuario final. Desde el punto de vista del diseño se han buscado patrones de flat design. Desde el punto de vista de visualización los elementos mostrados en las distintas pantallas dependen del tipo de usuario que accede a la aplicación (cliente, empresa financiera, empresa aseguradora, administrador etc). Esto se consigue mediante un filtrado por tipo de usuario en el código del "app.js".
- **Seguridad:** Conjunto de funcionalidades en el código que permiten mitigar los riesgos de algunos ataques como:
 - Ataques de desbordamiento.
 - Ataques DDoS
 - Privilegios no adecuados

- **Testeo:** Para el testing se han realizado pruebas de aserciones para las funciones de los contratos (CompaToken.sol y Token.sol) que evaluan (entre otras):
 - La generación del evento.
 - La creación de los valores correspondientes.
 - El almacenamiento en los mappings.
 - La casuística de ejecución de las funciones desde el punto de vista de varios tipos de usuarios (con distintas cuentas).

V. Modelo Económico

Para el desarrollo y ejecución del siguiente proyecto, se ha considerado el siguiente escenario económico para poder llevarlo a cabo. Al igual que el alcance del propio proyecto, se ha llevado a cabo un modelo económico faseado. Es por ello:

Modelo Económico Fase I

Este modelo incluye las funcionalidades incluidas en la fase I y detallado en el apartado “*Módulos incluidos en el alcance del proyecto*” del presente documento:

Presupuesto de la fase de desarrollo

Esta fase contempla el desarrollo y construcción de la plataforma incluyendo la programación y codificación del programa. Para ello, se estima un equipo 2,5 FTEs y un plazo de estimado de un mes y medio para llevar a cabo todas las tareas de esta fase del proyecto, detallado de la siguiente manera¹:

Tareas Incluidas dentro del presupuesto

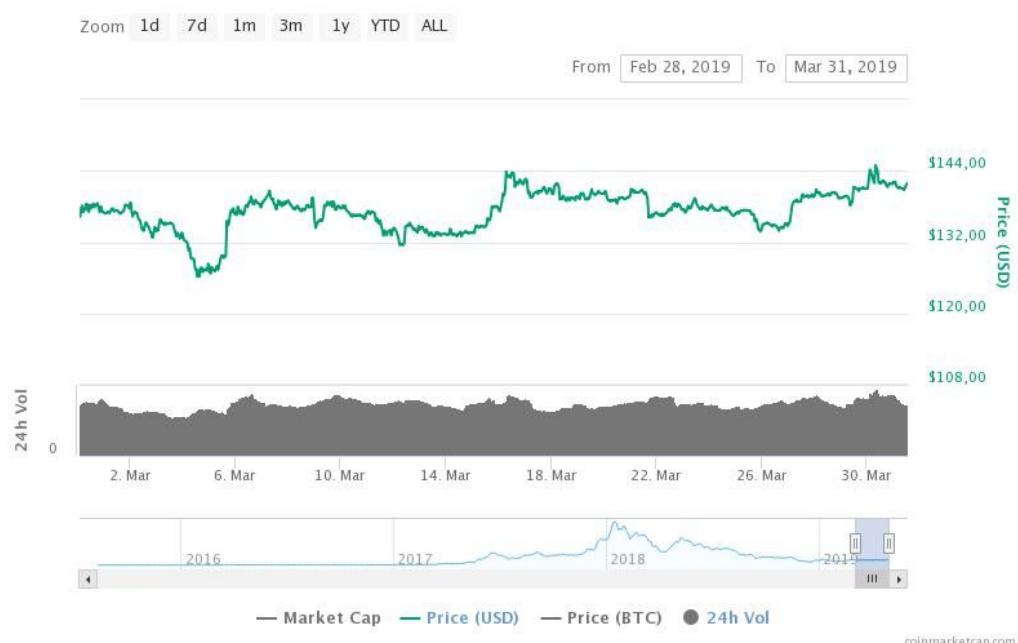
- Estudio y análisis de la situación actual de mercado. Benchmarking de otros productos de mercado y análisis de tendencias y necesidades de los usuarios
- Diseño funcional de la aplicación, incluyendo módulos necesarios y opciones a incluir en cada uno de ellos, así como la obtención de la definición de requisitos funcionales.
- Análisis y desarrollo del documento de requisitos técnicos.
- Priorización y asignación de esfuerzos.
- Codificación de la aplicación, tanto creación de la blockchain, smart contracts, frontales web,...
- Desarrollo fase de testeo, incluyendo pruebas para asegurar la corrección de las transacciones realizadas, testing de elementos del frontal, funcionamiento de Factory Contract y rendimiento o performance de la aplicación.
- Recodificación para solución de errores.
- Implantación en Producción, y soporte durante el proceso de post-implantación.

¹ Costes estimados en base a propuestas de servicios ofertados con costes típicos de una consultora de servicios tecnológicos

Recurso	Funciones	Horas estimadas	Coste hora	Importe total
Consultor Junior	<ul style="list-style-type: none"> • Programación • Análisis y resolución incidencia • Ejecución pruebas técnicas 	210	54	11.340 €
Consultor Senior	<ul style="list-style-type: none"> • Programación • Definición plan de pruebas funcionales y técnicas • Ejecución pruebas técnicas • Análisis y resolución incidencia 	210	81	17.010 €
Jefe Proyecto	<ul style="list-style-type: none"> • Gestión del proyecto • Ejecución pruebas funcionales 	105	100	10.500 €
				38.850 € ²

A estos costes de desarrollos, habría que añadir los **costes de implantación y uso**, los cuales son variables al depender del precio dinámico del coste del gas en Ethereum (la cual depende del uso de la red) y del valor de mercado de la criptomonedra Ether, la cual a modo ejemplo, durante el mes de Marzo de 2019 ha variado entre los 125 y 145 USD como se aprecia en el siguiente gráfico: ³

Ethereum Charts



Pero al ser tan volátil como el resto de las criptomonedas, esta moneda también se ha visto expuesta a grandes picos y valles de valoración. Si vemos la evolución de la criptomonedra ⁴:

² Coste IVA (21%) no incluido

³ <https://coinmarketcap.com/currencies/ethereum/>

⁴ <https://coinmarketcap.com/currencies/ethereum/>. Información obtenida a 31/03/2019

Ethereum Statistics

Ethereum Price	€126,51 EUR
Ethereum ROI ?	4.918,07%
Market Rank	#2
Market Cap	€13.342.303.249 EUR
24 Hour Volume	€3.855.536.685 EUR
Circulating Supply	105.462.576 ETH
Total Supply	105.462.576 ETH
Max Supply	No Data
All Time High	€1.275,76 EUR
All Time Low	€0,374746 EUR

Por ello, se ha planteado los costes en base a 3 escenarios de la criptomoneda:

- Escenario mínimo: precio actual de 126 euros/ether
- Escenario máximo: precio máximo obtenido por el Ether, correspondiente a 1.275 euros/ether
- Escenario medio: precio medio, obtenido del promedio de los precios anteriormente citados. Es decir, 700 euros/ether.

Presupuesto de la fase de implantación y uso

De cara a estimar el presupuesto de estas fases, se ha tenido en cuenta las siguientes condiciones:

- Se ha realizado un traspaso de contratos actuales a smart contracts. Además, todas las nuevas operaciones ya se realizarán a través de la DAPP.
- Se ha creado una empresa, que será la empresa de renting y contará con un usuario administrador.
- La empresa de renting trabaja con una empresa aseguradora única y una entidad financiera única, por lo tanto sólo se creará una entidad por cada una de ellas.
- Se ha calculado en base a la creación de 10 usuarios en la empresa de renting, y 5 usuarios para la entidad financiera y la entidad aseguradora.
- Se ha estimado una operativa de 100 smart contracts durante un año. Aunque si bien es cierto que un mismo usuario podría tener más de un contrato, se ha estimado un usuario por cada smart contract. Es decir, 100 usuarios.

Escenario	Coste Eth
Mínimos	126 €
Promedio	700 €
Máximos	1.275 €

PRESUPUESTO IMPLANTACIÓN				
TAREA	COSTE UNITARIO	ESCENARIO MÍNIMO	ESCENARIO PROMEDIO	ESCENARIO MÁXIMO
Desplegar contratos actuales en Smart Contract	0,16	20 €	112 €	204 €
Creación usuarios empresa renting (10)	0,01	13 €	70 €	128 €
Creación usuarios entidad financiera y aseguradora (10)	0,01	13 €	70 €	128 €

PRESUPUESTO USO ANUAL				
Creación usuarios (100)	0,005	63 €	350 €	638 €
Creación Smart Contracts (100)	0,48	6.048 €	33.600 €	61.200 €
Ejecución pago cuota mensual (12 cuotas anuales por 100 contratos)	0,001	151 €	840 €	1.530 €
TOTAL USO ANUAL		6.262 €	34.790 €	63.368 €
TOTAL DESARROLLO + IMPLANTACIÓN +USO ANUAL		45.125 €	73.710 €	102.345 €

VI. Conclusiones

Blockchain y el resto de tecnologías disruptivas más actuales, están permitiendo transformar el mundo tal y como se conoce, haciendo que los distintos sectores, especialmente el financiero, estén obligados a adaptarse y modificar sus procesos y operativas, así como modelos de negocio. La descentralización que ofrece la tecnología Blockchain (unido al hecho de activar la omnicanalidad de la aplicación) tiene un impacto directo en el incremento de la confianza del cliente (así como la mejora de la experiencia del mismo).

Además, permite al usuario ser parte más activa en la toma de decisiones del ecosistema creado. Para el ámbito de la empresa automovilística, el uso de las aplicaciones descentralizadas permite integrar en el mismo ecosistema a empresas cuyas finalidades son muy dispares (como empresas financieras, de seguros o de leasing), permitiendo al usuario final tener una mayor visibilidad en todo el proceso (desde la tarificación de su seguro hasta la financiación del mismo o el intercambio de vehículos).

Gracias a la aplicación de la tecnología Blockchain dota de una mayor seguridad a los datos registrados de los automóviles y sus operaciones dentro del ciclo completo, lo que ayuda a combatir el riesgo de fraude.

Y es por ello que, si bien la aplicación aquí presentada muestra la primera fase y el piloto, el desarrollo de la misma se ha realizado pensando en una futura expansión con la incorporación de nuevos módulos apoyada fuertemente en las ventajas y características que aportan las nuevas tecnologías, siendo los futuros “módulos clave o diferenciales”:

- Por un lado, la posibilidad de ofrecer servicios “Insurance as a Service” combinado con la implantación de mecanismos o dispositivos IoT en los automóviles asegurados, de tal manera que se podría adaptar las condiciones contractuales de los seguros al uso o características de los asegurados.
- Por otro lado, la posibilidad de realizar traza confiable de todo el ciclo de vida, de modo que se pueda aportar seguridad adicional a futuros terceros de los automóviles una vez finalizado el periodo contractual.

VII. Trabajos futuros

Los módulos descritos a continuación, serían tenidos en cuenta para futuras fases del proyecto, considerando su importancia en el flujo de información y en el servicio de renting.

Módulo Aseguradora

A futuro, en un escenario completo, el objetivo sería poder recabar información real (a partir de dispositivos IoT independientes) y que estos datos fueran leídos por el oráculo.

- Un modo de aseguramiento para el usuario, una solución Insutech llamada “Insurance as Service”, consistente en un modelo de renting donde se incluye un tipo de seguro que se adapte al uso de los vehículos por parte de los asegurados en el cual el asegurado recibe coberturas y paga en consonancia al uso que le da al vehículo adquirido. De manera más detallada, a futuro se plantea conectar la información de una base de datos externa con el smart contract (el oráculo leería de esa base de datos), de cara a simular la utilización de los datos provenientes de un dispositivo IoT (kilómetros recorridos, velocidad, carreteras transitadas, tiempo de estacionamiento, etc..).
- Para cumplir con este objetivo, se utilizarían datos de la trazabilidad de vehículos, con el objetivo de simular un sistema de cobro más preciso (de acuerdo al desplazamiento y el modo de conducción del usuario).
- Se diferenciará dos tipos de información: dinámica (o información ficticia que simula el comportamiento en tiempo real de los vehículos) e información estática.
 - Del primer caso tendríamos: los kilómetros recorridos, la velocidad máxima, media, las carreteras transitadas, horas de conducción, tiempo de conducción seguida, etc
 - Respecto a información estática: se puede plantear la clasificación de riesgo de las carreteras, índice de accidentes por personas, sexo y edad, índice de siniestralidad en función de marca y tipo de automóvil...
- En base a toda la información, sería necesario así mismo definir las reglas de coberturas contratables. Es decir, en base a una serie de parámetros establecidos a partir de la información y datos que se disponen, ofrecer unas coberturas u otras, y adaptar el pago de los asegurados.
 - Por ejemplo: un conductor que conduce 10.000 kms al año, por autovías y siempre respetando los límites de velocidad, no debería pagar lo mismo que otro conductor que recorre más kilómetros, o que circula por carreteras convencionales o conduce de manera inapropiada.

Módulo taller

- El módulo de taller permitirá a empresas proveedoras de servicio de reparación, participar en los procesos de mantenimiento de los vehículos que han sufrido algún desperfecto o accidente. De esta forma se podrá registrar cualquier modificación o reparación en el coche. Para garantizar este proceso, se realizará, el desarrollo de un frontal, para la gestión del servicio de taller (para la reparación de los vehículos). En definitiva, se trata de simular el libro de mantenimiento de cualquier coche que se realiza a día de hoy de manera manual y en formato papel por lo general, asegurando la transparencia y la inmutabilidad de la información.

- La información del coste de los recambios quedará registrada y será transparente para el usuario final (de tal forma que el cliente pueda ver el coste del servicio aplicado por el taller).
- Adicionalmente, quedaría registrado componentes utilizados por cada vehículo permitiendo no poder alterar la información registrada y de este modo minimizar y asegurar la lucha contra el riesgo de fraude.

Módulo Compraventa

- Este módulo permitirá a la empresa de renting trabajar asociado a concesionarios dedicados a la compra/venta de vehículos. Así se pueden vender aquellos vehículos que los usuarios no deseen adquirir directamente.
- Se plantearía la posibilidad de registrar toda la información del ciclo de vida del vehículo, en combinación con el resto de módulos especialmente taller y proveedor, para asegurar que la información de cada vehículo sea 100% veraz y prevenir posibles fraudes.

Módulo Proveedor

- El módulo de proveedor gestionará las flotas de vehículos ofrecidas a la empresa de renting, registrando todo el proceso de fabricación del automóvil, desde la petición del mismo hasta su envío, y garantizando la transparencia de cara al usuario final, muy similar en su concepto a otras soluciones blockchain que se están realizando para asegurar todo el ciclo de vida de los procesos de supply chain de mercancías.

VIII. Bibliografía

1. https://elpais.com/economia/2018/02/01/actualidad/1517515718_449357.html, enero 2019.
2. Álex Preukschat. Blokchain: La revolución industrial de Internet (<https://libroblockchain.com/revolucion/>), (Capítulo 1, pág 28 formato digital)
3. <https://retos-directivos.eae.es/blockchain-y-la-ficcion-de-la-desintermediacion-financiera/>
4. https://elpais.com/tecnologia/2018/02/26/actualidad/1519642606_449102.html
5. <http://www.expansion.com/economia-digital/innovacion/2018/07/11/5b43a10b22601dff438b463c.html>
6. Mastering Ethereum: Building Smart Contracts and DApps. Andreas Antonopoulos, Gavin Wood Ph.D. <https://github.com/ethereumbook/ethereumbook/blob/develop/01what-is.asciidoc>
7. <https://medium.com/blockmatics-blog/top-10-differences-between-bitcoin-and-ethereum-d2d3dd62101>
8. <https://blockonomi.com/ethereum-vs-bitcoin/>
9. <https://miethereum.com/ether/bitcoin-vs-ethereum/>
10. <https://www.leaseplango.es/blog/renting/renting-leasing-diferencias/>
11. How Blockchain may impact logistics, supply chain and transportation: A conversation with the blockchain in the transport area.
<https://www.forbes.com/sites/insights-penske/2018/09/04/how-blockchain-may-impact-logistics-supply-chain-and-transportation-a-conversation-with-the-blockchain-in-transport-alliance/#2a8e2f95f2b3>, enero del 2019.
12. Blockchain in insurance: Application and pursuing a path to adoption:
[https://www.ey.com/Publication/vwLUAssets/EY-blockchain-in-insurance/\\$FILE/EY-blockchain-in-insurance.pdf](https://www.ey.com/Publication/vwLUAssets/EY-blockchain-in-insurance/$FILE/EY-blockchain-in-insurance.pdf), enero del 2019.
13. [https://www.ey.com/Publication/vwLUAssets/ey-blockchain-and-the-supply-chain-three/\\$FILE/ey-blockchain-and-the-supply-chain-three.pdf](https://www.ey.com/Publication/vwLUAssets/ey-blockchain-and-the-supply-chain-three/$FILE/ey-blockchain-and-the-supply-chain-three.pdf), enero del 2019
14. IBM Blockchain car lease demo: <https://developer.ibm.com/tv/ibm-blockchain-car-lease-demo/>, enero del 2019.
15. IBM Blockchain Platform:
<https://www.ibm.com/es-es/marketplace/cloud-based-blockchain-platform/details>, enero 2019.

Anexo I - Manual del Usuario

1. Alta Usuarios

1.1. Usuarios Clientes

Para realizar el Alta de un usuario tipo Cliente es importante que haya realizado previamente el proceso de login desde la herramienta recomendada Metamask.

Una vez el usuario haya realizado el login, podrá verificar en la parte superior de la Dapp su Address en Metamask.

Para acceder al formulario de Alta, se deben seguir los siguientes pasos:

1. Pinchar sobre el botón Alta, que redirigirá al usuario al formulario de Alta.

The screenshot shows a user interface titled "ZONA DE ACCESO". At the top, there is a yellow bar with the text "Your Account: 0x11027aed76e811acc8f5965fcb3909699851c7d0". Below it is a blue bar with the text "Seleccione el módulo correspondiente". At the bottom, there is a row of buttons: "Alta" (highlighted with a red box), "Zona Admin Clientes", "Alta de coches", and "Entrega de coches".

2. Desde el Formulario de Alta el Usuario deberá seleccionar en el desplegable Tipo de Usuario la opción Cliente.

The screenshot shows a registration form titled "Registro de datos de usuario". At the top, there is a blue bar with the text "Introduzca sus datos de registro". Below it is a table of fields:

Nombre de Usuario	<input type="text"/>
Tipo Usuario	<input type="button" value="Elija opción"/>
Nombre	<input type="text"/>
Apellidos	<input type="text"/>
Edad	<input type="text"/>
ID Licencia de Conducción	<input type="text"/>
Antiguedad Licencia de Conducción	<input type="text"/>
Puntos Licencia de Conducción	<input type="text"/> (with a dropdown arrow icon)
Teléfono	<input type="text"/>
Email	<input type="text"/>
DNI	<input type="text"/>

Registro de datos de usuario

Introduzca sus datos de registro

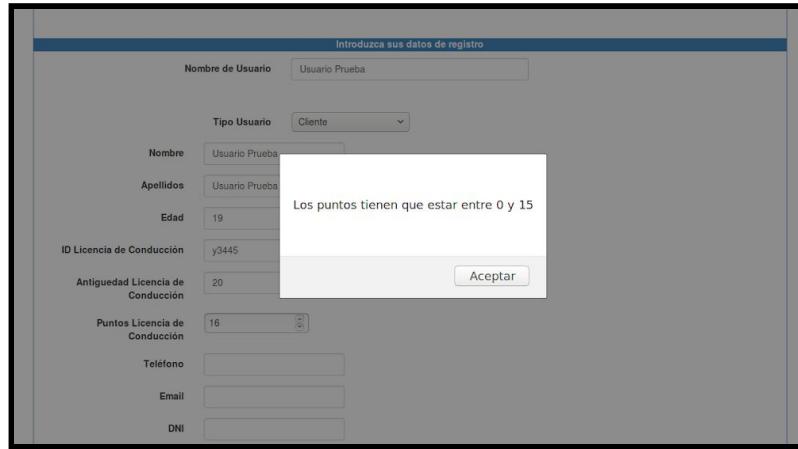
Nombre de Usuario	Usuario Prueba
Tipo Usuario	<input type="button" value="Elja opción"/>
Nombre	<input type="text" value=""/>
Apellidos	<input type="text" value=""/>
Edad	<input type="text" value=""/>
ID Licencia de Conducción	<input type="text" value=""/>
Antiguedad Licencia de Conducción	<input type="text" value=""/>
Puntos Licencia de Conducción	<input type="text" value=""/> <input type="button" value=""/>
Teléfono	<input type="text" value=""/>
Email	<input type="text" value=""/>
DNI	<input type="text" value=""/>
Nombre empresa	<input type="button" value=""/>
CIF Empresa	<input type="text" value=""/>

3. Al seleccionar la Opción cliente, el usuario podrá llenar los campos que permanezcan disponibles. Los campos que serán deshabilitados son : Nombre de empresa, CIF de Empresa, y Link Site empresa.

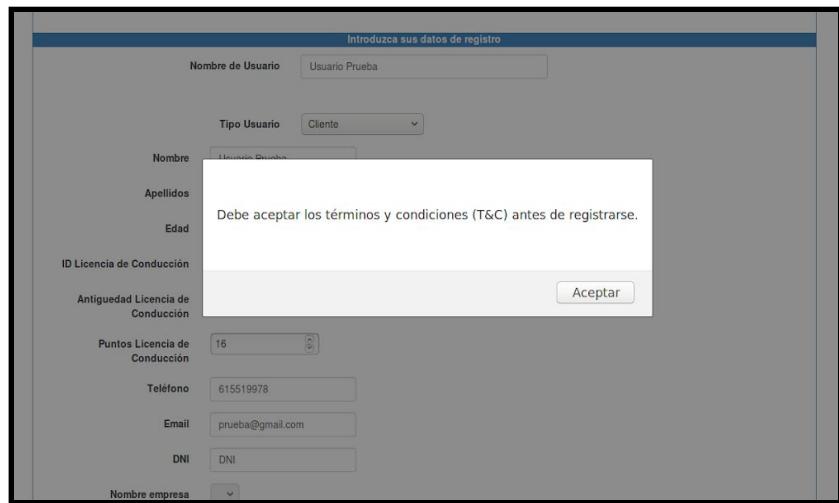
Introduzca sus datos de registro

Nombre de Usuario	Usuario Prueba
Tipo Usuario	<input type="button" value="Cliente"/>
Nombre	<input type="text" value="Usuario Prueba"/>
Apellidos	<input type="text" value="Usuario Prueba"/>
Edad	<input type="text" value="19"/>
ID Licencia de Conducción	<input type="text" value="y3445"/>
Antiguedad Licencia de Conducción	<input type="text" value="20"/>
Puntos Licencia de Conducción	<input type="text" value="16"/> <input type="button" value=""/>
Teléfono	<input type="text" value="615519978"/>
Email	<input type="text" value="prueba@gmail.com"/>
DNI	<input type="text" value="DNI"/>
Nombre empresa	<input type="button" value=""/>
CIF Empresa	<input type="text" value=""/>
Link site empresa	<input type="text" value=""/>
<input type="checkbox"/> Acepta los términos y condiciones? Consultar T&C Registro	

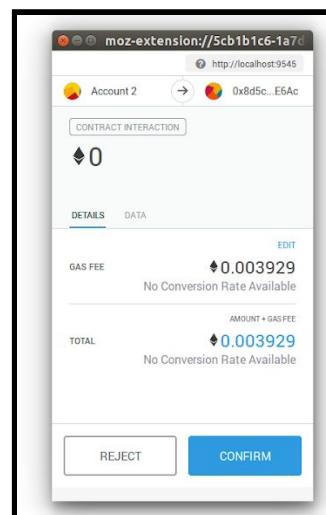
4. Una vez el usuario haya ingresado todos los campos, deberá aceptar los términos y condiciones y realizar el alta pinchando en registro.
 - a. En caso de que el usuario ingrese un valor mayor a 15 (valor máximo para España), un mensaje le indicará que debe corregir el valor para que se encuentre entre 0 y 15 puntos.



- b. En caso de que el usuario no acepte los términos y condiciones también, un mensaje le indicará que debe aceptar dichos términos y condiciones antes de poder realizar el alta.



- 5. Una vez el formulario haya sido llenado de forma correcta, el usuario deberá proceder a validar la transacción, para confirmar el alta.



1.2. Usuarios Empresa Aseguradora

Para realizar el Alta de un usuario tipo Empresa Aseguradora, es importante que haya realizado previamente el proceso de login desde la herramienta recomendada Metamask.

Una vez el usuario haya realizado el login, podrá verificar en la parte superior de la Dapp su Address en Metamask.

Para acceder al formulario de Alta, se deben seguir los siguientes pasos:

1. Pinchar sobre el botón Alta, que redirigirá al usuario al formulario de Alta.

The screenshot shows a user interface titled "ZONA DE ACCESO". At the top, a yellow bar displays the account address: "Your Account: 0x11027aed76e811acc8f5965fc3909699851c7d0". Below it, a blue bar says "Seleccione el módulo correspondiente". A navigation bar at the bottom has four items: "Alta" (highlighted with a red border), "Zona Admin Clientes", "Alta de coches", and "Entrega de coches".

2. Desde el Formulario de Alta el Usuario deberá seleccionar en el desplegable Tipo de Usuario la opción Empresa Aseguradora.

The screenshot shows a registration form titled "Introduzca sus datos de registro". It includes fields for: Nombre de Usuario (set to "Usuario Prueba"), Tipo Usuario (dropdown menu showing "Cliente", "Empresa Aseguradora" (highlighted in orange), and "Empresa Financiera"), Nombre, Apellidos, Edad, ID Licencia de Conducción, Antigüedad Licencia de Conducción, Puntos Licencia de Conducción, Teléfono, Email, DNI, Nombre empresa (dropdown menu showing "ALLIANZ" (highlighted in orange)), CIF Empresa (set to "C1140"), and Link site empresa (set to "www.prueba.com"). At the bottom, there is a checkbox labeled "Acepta los términos y condiciones?" (checked) and buttons for "Consultar T&C" and "Registro".

3. Al seleccionar la Opción Empresa Aseguradora, el usuario podrá rellenar los campos que permanezcan disponibles: Nombre empresa (Desplegable con tres empresas Mapfre, Allianz y AXA), CIF empresa y Link site empresa.

Introduzca sus datos de registro

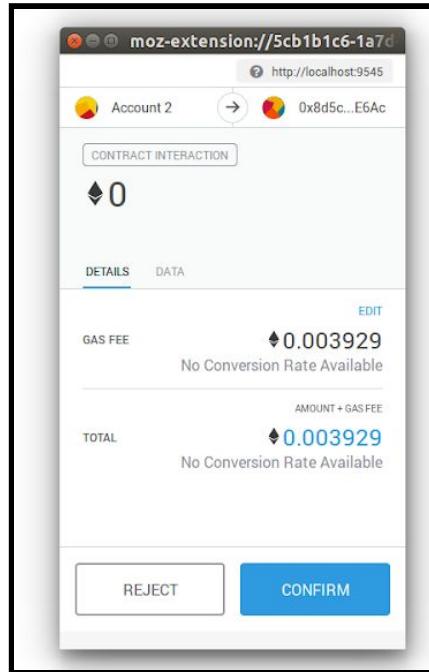
Nombre de Usuario	<input type="text" value="Usuario Prueba"/>
Tipo Usuario	<input type="button" value="Empresa Aseguradora"/>
Nombre	<input type="text"/>
Apellidos	<input type="text"/>
Edad	<input type="text"/>
ID Licencia de Conducción	<input type="text"/>
Antigüedad Licencia de Conducción	<input type="text"/>
Puntos Licencia de Conducción	<input type="text"/>
Teléfono	<input type="text"/>
Email	<input type="text"/>
DNI	<input type="text"/>
Nombre empresa	<input type="button" value="ALLIANZ"/>
CIF Empresa	<input type="text" value="C1140"/>
Link site empresa	<input type="text" value="www.prueba.com"/>
<input checked="" type="checkbox"/> ¿Acepta los términos y condiciones? <input type="button" value="Consultar T&C"/> <input type="button" value="Registro"/>	

4. Una vez el usuario haya ingresado todos los campos, deberá aceptar los términos y condiciones y realizar el alta pinchando en registro.
 - a. En caso de que el usuario no acepte los términos y condiciones también, un mensaje le indicará que debe aceptar dichos términos y condiciones antes de poder realizar el alta.

Introduzca sus datos de registro

Nombre de Usuario	<input type="text" value="Usuario Prueba"/>
Tipo Usuario	<input type="button" value="Cliente"/>
Nombre	<input type="text" value="Usuario Prueba"/>
Apellidos	<input type="text"/>
Edad	<input type="text"/>
ID Licencia de Conducción	<input type="text"/>
Antigüedad Licencia de Conducción	<input type="text"/>
Puntos Licencia de Conducción	<input type="text" value="16"/>
Teléfono	<input type="text" value="615519978"/>
Email	<input type="text" value="prueba@gmail.com"/>
DNI	<input type="text"/>
Nombre empresa	<input type="button" value=""/>
<input checked="" type="checkbox"/> Debe aceptar los términos y condiciones (T&C) antes de registrarse. <input type="button" value="Aceptar"/>	

5. Una vez el formulario haya sido rellenado de forma correcta, el usuario deberá proceder a validar la transacción, para confirmar el alta.



1.3. Usuarios Empresa financiera

Para realizar el Alta de un usuario tipo Empresa Financiera, es importante que haya realizado previamente el proceso de login desde la herramienta recomendada Metamask.

Una vez el usuario haya realizado el login, podrá verificar en la parte superior de la Dapp su Address en Metamask.

Para acceder al formulario de Alta, se deben seguir los siguientes pasos:

1. Pinchar sobre el botón Alta, que redirigirá al usuario al formulario de Alta.

2. Desde el Formulario de Alta el Usuario deberá seleccionar en el desplegable Tipo de Usuario la opción Empresa Financiera.

Introduzca sus datos de registro

Nombre de Usuario	<input type="text"/>
Tipo Usuario	<input type="button" value="Empresa Financiera"/>
Nombre	<input type="text"/>
Apellidos	<input type="text"/>
Edad	<input type="text"/>
ID Licencia de Conducción	<input type="text"/>
Antiguedad Licencia de Conducción	<input type="text"/>
Puntos Licencia de Conducción	<input type="text"/>
Teléfono	<input type="text"/>
Email	<input type="text"/>
DNI	<input type="text"/>
Nombre empresa	<input type="button" value="Mi Financiera SL"/>
CIF Empresa	<input type="text" value="c1234"/>
Link site empresa	<input type="text" value="www.prueba.com"/>
<input checked="" type="checkbox"/> ¿Acepta los términos y condiciones? Consultar T&C Registro	

3. Al seleccionar la Opción Empresa Financiera, el usuario podrá rellenar los campos que permanezcan disponibles: Nombre empresa (Desplegable con una Empresa Financiera-Mi empresa Financiera SL), CIF empresa y Link site empresa.

Introduzca sus datos de registro

Nombre de Usuario	<input type="text"/>
Tipo Usuario	<input type="button" value="Empresa Financiera"/>
Nombre	<input type="text"/>
Apellidos	<input type="text"/>
Edad	<input type="text"/>
ID Licencia de Conducción	<input type="text"/>
Antiguedad Licencia de Conducción	<input type="text"/>
Puntos Licencia de Conducción	<input type="text"/>
Teléfono	<input type="text"/>
Email	<input type="text"/>
DNI	<input type="text"/>
Nombre empresa	<input type="button" value="Mi Financiera SL"/>
CIF Empresa	<input type="text" value="c1234"/>
Link site empresa	<input type="text" value="www.prueba.com"/>
<input checked="" type="checkbox"/> ¿Acepta los términos y condiciones? Consultar T&C Registro	

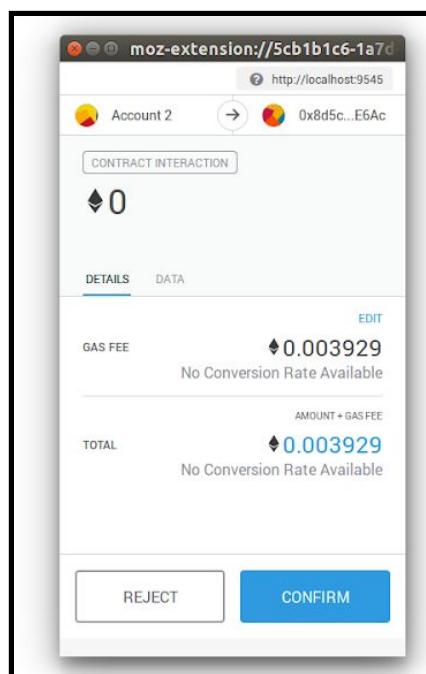
4. Una vez el usuario haya ingresado todos los campos, deberá aceptar los términos y condiciones y realizar el alta pinchando en registro.
 - a. En caso de que el usuario no acepte los términos y condiciones también, un mensaje le indicará que debe aceptar dichos términos condiciones antes de poder realizar el alta.

Introduzca sus datos de registro

Nombre de Usuario	Usuario Prueba
Tipo Usuario	Cliente
Nombre	Usuario Prueba
Apellidos	
Edad	
ID Licencia de Conducción	
Antigüedad Licencia de Conducción	<input type="button" value="Aceptar"/>
Puntos Licencia de Conducción	16
Teléfono	615519978
Email	prueba@gmail.com
DNI	DNI
Nombre empresa	

Debe aceptar los términos y condiciones (T&C) antes de registrarse.

- Una vez el formulario haya sido llenado de forma correcta, el usuario deberá proceder a validar la transacción, para confirmar el alta.



2. Usuarios

2.1. Administrador/Owner

El Usuario Admin/Owner, tendrá la opción de acceder a cualquiera de las ventanas, pero funcionalmente tendrá acceso exclusivo para:

- Administrar los usuarios.
- Al momento de hacer el despliegue del contrato será la única (address) que podrá inicializar el contrato transfiriendo los tokens iniciales
- Activar y desactivar los contratos asociados
- Realizar las gestiones de Renting, Alta Coches y Validación/Entrega de Coches.

2.1.1. Administración de usuarios

El uso y acceso de la ventana de Administración de usuarios es la siguiente:

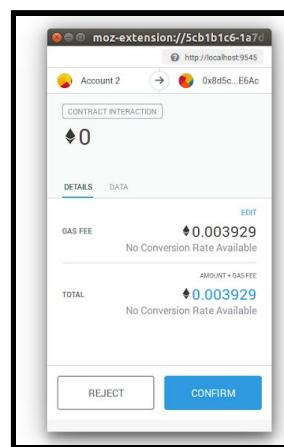
1. Pinchar sobre el botón Zona Admin.



2. Ingresar el Address del usuario que se desea Consultar. Al pinchar sobre el botón consultar, se observará en la pantalla los datos de DNI y tipo de Usuario.

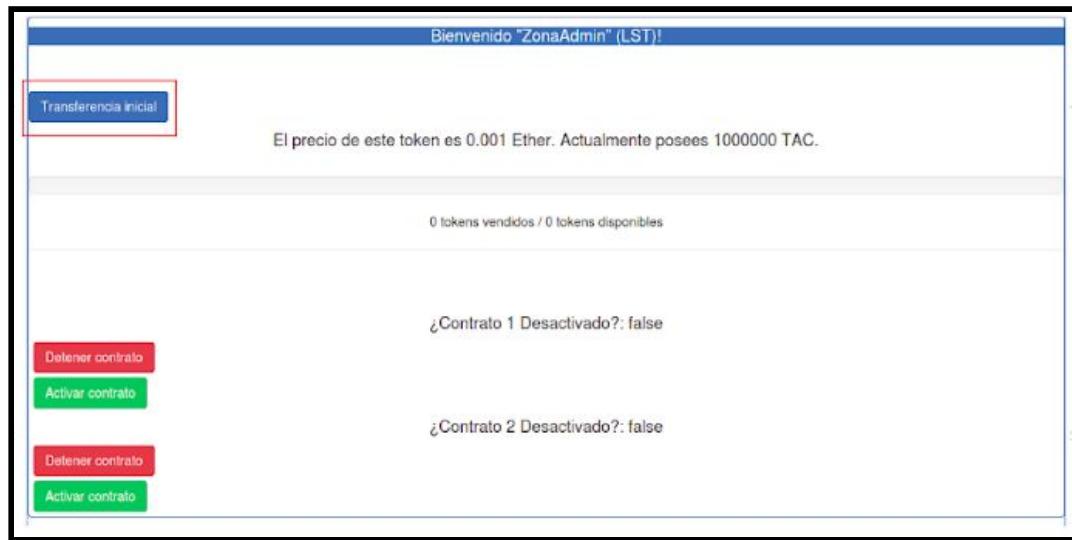


3. El Administrador puede dar de baja el usuario, pinchando sobre el botón Eliminar usuario y confirmando la transacción.

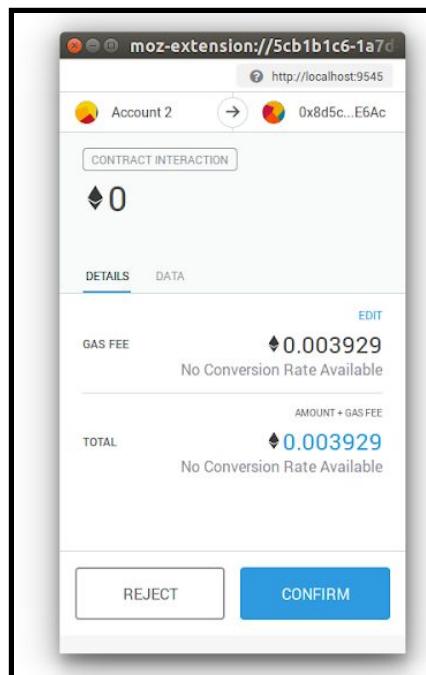


2.1.2. Inicialización de Tokens

1. El uso y acceso a esta funcionalidad se cargará en la ventana principal de la Dapp. El usuario deberá bajar hasta encontrar el módulo “Zona Admin” y pinchando sobre la opción Transferencia inicial.

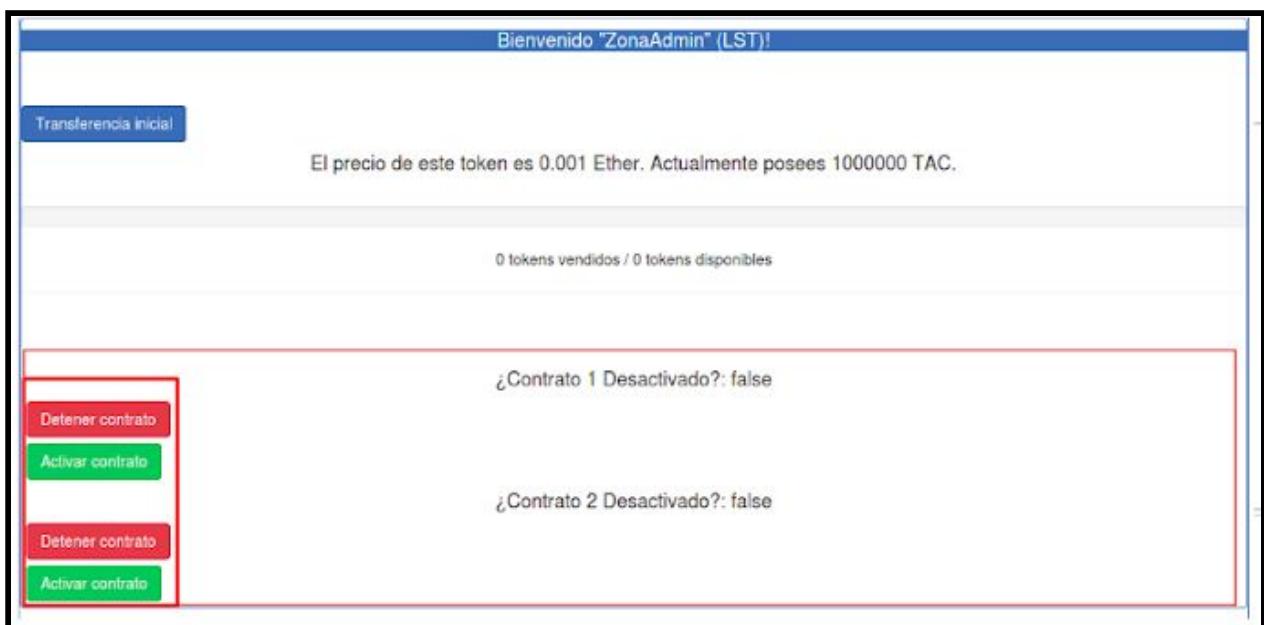


2. Se debe confirmar la transacción para que se lleve a cabo la transferencia de los Tokens al contrato.

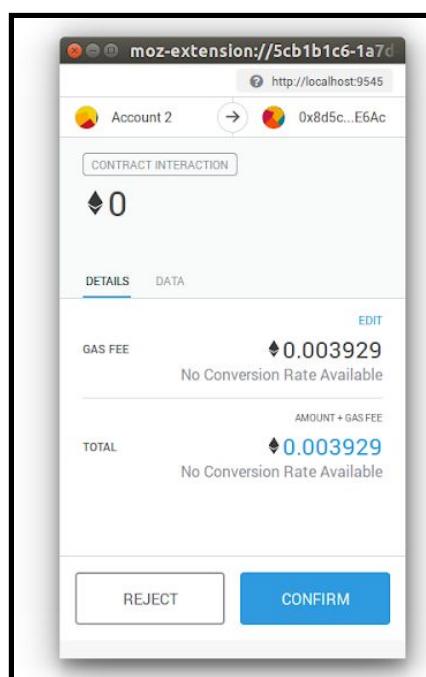


2.1.3. Panic Buttons

1. El uso y acceso a esta funcionalidad se cargará en la ventana principal de la Dapp. El usuario deberá bajar, hasta encontrar el módulo “Zona Admin” y pinchando sobre la opción “Detener” o “Activar” contrato según lo requiera (se pueden ver 4 botones dado que se pueden activar o desactivar los contratos de forma independiente).



2. Se debe confirmar la transacción para que se lleve a cabo la Activación/desactivación del contrato.



2.1.4. Renting

2.1.4.1. Alta Coches

El uso y acceso de la ventana de Administración de usuarios es la siguiente:

1. Pinchar sobre el botón Alta de Coches.

ZONA DE ACCESO

Your Account: 0x11027aed76e811acc8f5965fcb3909699851c7d0
Seleccione el módulo correspondiente

Alta Zona Admin Clientes Alta de coches Entrega de coches

2. Ingresar el ID del coche, el cual debe ser siempre un número.

Registro de datos de alta de coches

Introduzca el identificador del coche y su gama

ID coche: 1054

Gama: Elija opción ▾

Alta coche

3. Seleccionar la gama del coche.

Registro de datos de alta de coches

Introduzca el identificador del coche y su gama

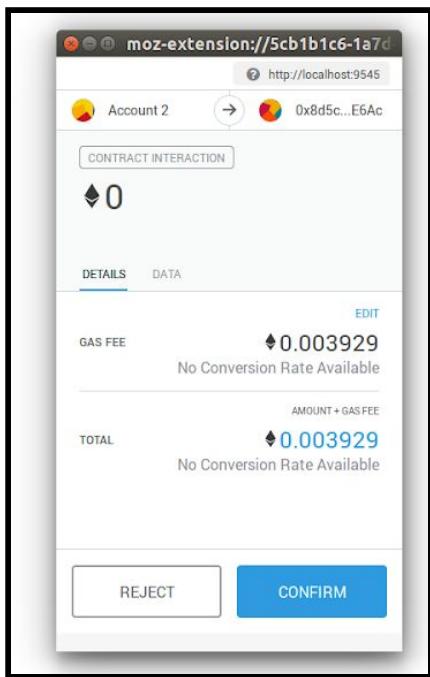
ID coche: 1054

Gama: Elija opción ▾

Premium
Luxury
Classic
Corriente
Furgoneta

Alta coche

4. Confirmar la transacción para llevar a cabo el alta del coche luego de pinchar sobre la opción Alta coche.



2.1.4.2. Validación/Entrega Coches

El uso y acceso de la ventana de Administración de usuarios es la siguiente:

1. Pinchar sobre el botón Entrega de Coches.



2. Se puede verificar si el coche ha sido entregado ingresando los datos del coche (ID del coche y Gama del coche) y pinchando sobre consultar.

Módulo de entrega de coche

Introduzca el identificador del coche y su gama

ID coche	<input type="text" value="1054"/>
Gama	<input type="button" value="Elja opción ▾"/>

Consultar Coche

Id Coche	Entregado
1054	false

Validar coche

3. En caso de que el coche haya sido entregado (true), se valida la entrega pinchando en el botón Validar Coche y confirmando la transacción.

Módulo de entrega de coche

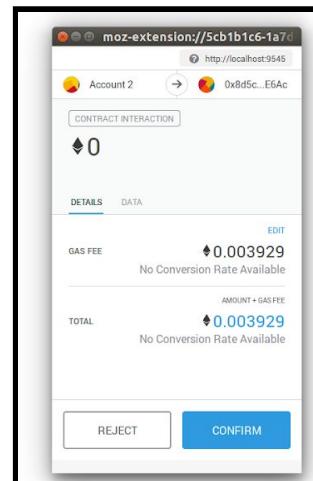
Introduzca el identificador del coche y su gama

ID coche	<input type="text" value="1054"/>
Gama	<input type="button" value="Premium ▾"/>

Consultar Coche

Id Coche	Entregado
1054	false

Validar coche



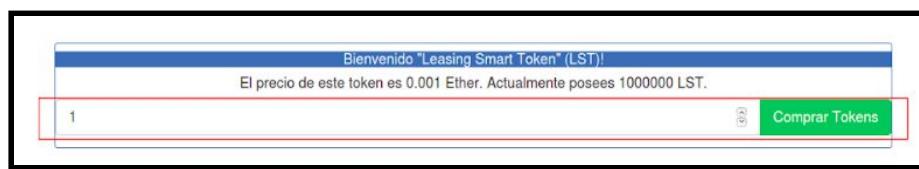
3.1. Cliente

Una vez el usuario se ha dado de alta como usuario tipo cliente, podrá tener acceso a la Zona Cliente. A continuación se explicarán las funcionalidades de éste módulo.

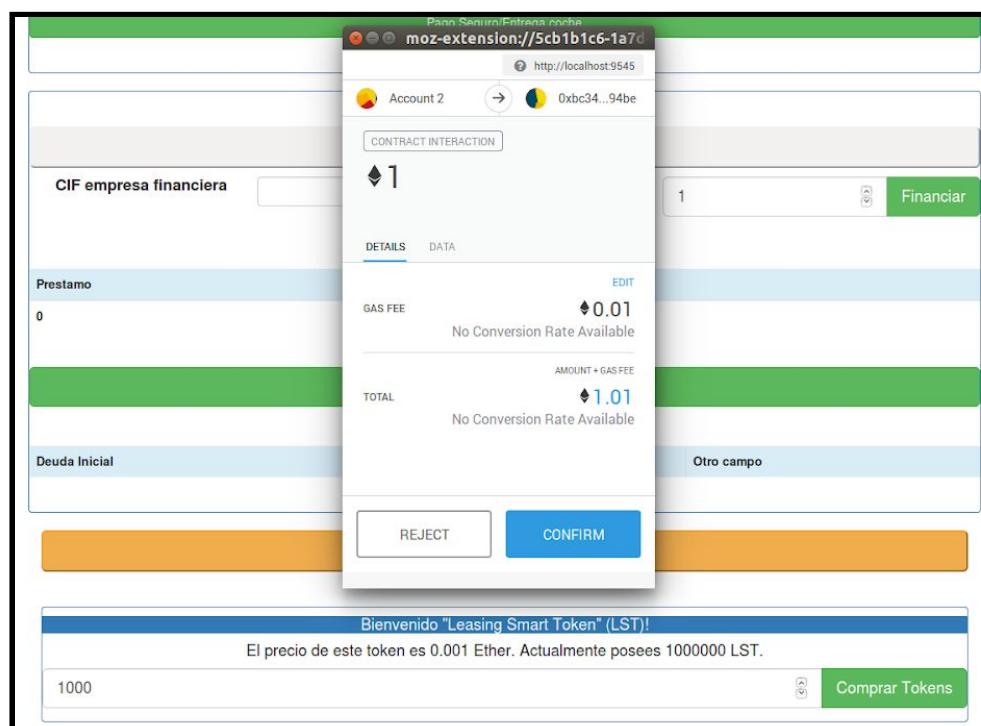
3.1.1. Comprar Tokens

Para la compra de Tokens se deben seguir los siguientes pasos:

1. Ingresas el número de Tokens y pinchar sobre la opción comprar Tokens.



2. Confirmar la compra de Tokens. De acuerdo a la cantidad de Tokens y al precio definido para cada uno, será el costo de la transacción.



3.1.2. Financiar Tokens

Para Financiar Tokens se deben seguir los siguiente pasos:

1. Pinchar sobre la opción Solicitar Financiamiento.

Solicitar Financiamiento

CIF empresa financiera

Puede financiar hasta un máximo de: 0 Ether.

1

Financiar

Prestamo	Valor a Pagar
0	0

Pago Financiación

2. Ingresar los campos requeridos, cantidad a financiar, tiempo en el que se devolverá y si es cliente o no de la empresa.

ZONA DE FINANCIACIÓN

FINANCIACIÓN

Your Account: 0x06725d3c615bf1fafb8c0111cb51de06e2764f41

Para obtener financiación ingrese los siguientes valores

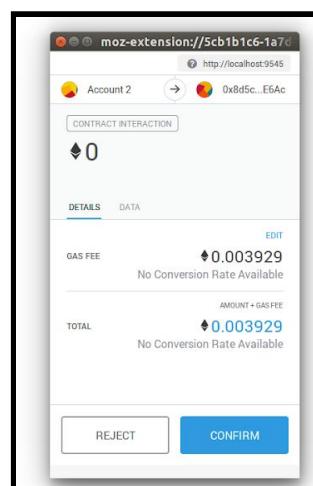
Volumen de crédito: 1000

Demora en días: 60

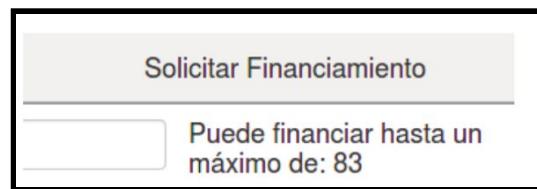
¿Es usted nuevo cliente? Si

Envío de petición

3. Se confirma la solicitud validando la transacción.



- Una vez se ha validado la solicitud, el usuario podrá verificar la cantidad máxima de Tokens que le pueden ser prestados por la empresa financiera.



- Para solicitar un préstamo y Tokens, el usuario deberá indicar el CIF de la empresa financiera, el número de Tokens y pinchar en la opción financiar.

Prestamo	Valor a Pagar
0	0

Pago Financiación

- Una vez se ha validado la transacción, se cargarán los tokens al usuario y en la parte inferior se podrá ver el préstamo realizado y el valor pagar en Tokens.

Prestamo	Valor a Pagar
70	77

3.1.2.1. *Pagar Financiación*

Para Pagar los Tokens financiado se deben seguir los siguientes pasos:

- El usuario deberá contar con la cantidad de tokens indicadas en Valor a Pagar.
- Pinchando sobre Pago Financiación, se realizará el pago total.

Prestamo	Valor a Pagar
0	0

Pago Financiación

- Una vez realizado el Pago, los valores: Préstamo y Valor a Pagar retornarán a 0.

3.1.3. Rentar Coche

Para rentar el Coche se deben seguir los siguiente pasos:

- El usuario deberá pinchar sobre la opción Ver coches.



- El usuario accede a la ventana con las gamas de coches ofrecidos, en la cual podrá escoger la gama que le interese.



- Una vez el usuario haya seleccionado la gama de coche que desea, será redirigido a una ventana en la que podrá contratar el seguro de acuerdo a las tarifas de cada empresa Aseguradora.

SEGUROS
TU SELECCIÓN

SEGURO MAPFRE

Precio Km Ciudad €	13
Precio Km Carretera €	13
Precio Aparcado € x Hora	13



SEGURO ALLIANZ

Precio Km Ciudad €	0
Precio Km Carretera €	0
Precio Aparcado € x Hora	0



SEGURO AXA

Precio Km Ciudad € x Hora	0
Precio Km Carretera € x Hora	0
Precio Aparcado € x Hora	0



4. Cuando el usuario haya escogido la Empresa Aseguradora, será redirigido a una ventana en la cual podrá confirmar su compra pinchando en Confirmar Comprar.



CONFIRMA TÚ COMPRA



GAMA PREMIUM

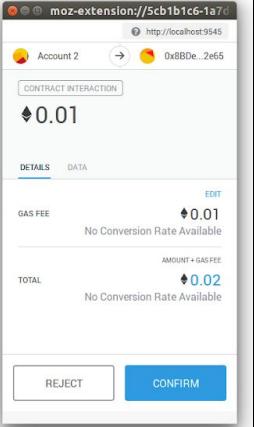
CONTRACT INTERACTION

◆ 0.01

DETAILS **DATA**

GAS FEE ◆ 0.01
No Conversion Rate Available

TOTAL ◆ 0.02
No Conversion Rate Available



- Una vez la compra sea confirmada, el usuario podrá volver a la ventana principal, donde podrá verificar el ID del coche asignado y la tarifa de seguro que ha adquirido.

Leasing			
Id Coche	Costo Seguro Km Ciudad	Costo Seguro Km Carretera	Costo Seguro Hora Aparcado
1054	13	13	13

ID Coche **ID Seguro** **Entregado**

1054 0 false

Consultar Precio Seguro

El precio de tu seguro en LST es:

CIF empresa Aseguradora

Pago Seguro/Entrega coche

3.1.4. Entregar Coche

Para Entregar el Coche se deben seguir los siguiente pasos:

- La entrega del coche implica que el usuario pague el costo del seguro, por lo cual podrá consultar el costo de su seguro de acuerdo a los Km recorridos y el tiempo aparcado.

Leasing			
Id Coche	Costo Seguro Km Ciudad	Costo Seguro Km Carretera	Costo Seguro Hora Aparcado
1054	13	13	13

ID Coche **ID Seguro** **Entregado**

1054 0 false

Consultar Precio Seguro

El precio de tu seguro en LST es: 3900

CIF empresa Aseguradora

- El usuario deberá pinchar sobre la opción Entregar Coche.

Leasing			
Id Coche	Costo Seguro Km Ciudad	Costo Seguro Km Carretera	Costo Seguro Hora Aparcado
1054	13	13	13

ID Coche **ID Seguro** **Entregado**

1054 0 false

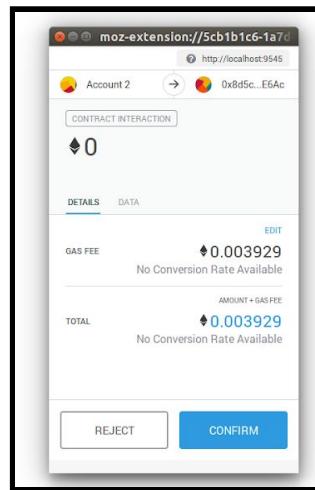
Consultar Precio Seguro

El precio de tu seguro en LST es:

CIF empresa Aseguradora

Pago Seguro/Entrega coche

- Deberá confirmar la transacción correspondiente al pago en tokens del seguro.



- Una vez haya confirmado el pago, será removido el coche de su ventana, y el estado del coche pasará a entregado para la empresa de Leasing.

3.2. Empresa Aseguradora

Una vez el usuario se ha dado de alta como usuario tipo Aseguradora, podrá tener acceso a la zona de su empresa Aseguradora. A continuación se explican las funcionalidades de éste módulo.

3.2.1. Modificar precios

El Usuario podrá modificar los precios de los Km recorridos en Ciudad, Km recorridos en carretera y el precio por el tiempo aparcado de la siguiente forma:

- Deberá seleccionar la categoría en la cual quiere realizar el cambio.

- En el desplegable encontrará tres casillas para modificar el precio y 3 botones para confirmar su cambio.

Asegurador Mapfre

Ingrese los nuevos valores en los campos. Registre el valor actual de aquellos valores que no desea cambiar.

Seguro Premium		
Precio Km Ciudad	<input type="text" value="10"/>	<input type="button" value="Actualizar"/>
Precio Km Carretera	<input type="text" value=""/>	<input type="button" value="Actualizar"/>
Precio Aparcado	<input type="text" value=""/>	<input type="button" value="Actualizar"/>
Precio Km Ciudad	<input type="text" value="0"/>	
Precio Km Carretera	<input type="text" value="0"/>	
Precio Aparcado	<input type="text" value="0"/>	
Seguro Luxure	<input type="text" value=""/>	
Seguro Classic	<input type="text" value=""/>	
Seguro Corriente	<input type="text" value=""/>	
Seguro Furgoneta	<input type="text" value=""/>	

- Una vez realice el cambio deberá validar la transacción.

Asegurador Mapfre

Ingrese los nuevos valores en los campos. Registre el valor actual de aquellos valores que no desea cambiar.

Seguro Premium		
Precio Km Ciudad	<input type="text" value="10"/>	<input type="button" value="Actualizar"/>
Precio Km Carretera	<input type="text" value=""/>	<input type="button" value="Actualizar"/>
Precio Aparcado	<input type="text" value=""/>	<input type="button" value="Actualizar"/>
Precio Km Ciudad	<input type="text" value="0"/>	
Precio Km Carretera	<input type="text" value="0"/>	
Precio Aparcado	<input type="text" value="0"/>	
Seguro Luxure	<input type="text" value=""/>	
Seguro Classic	<input type="text" value=""/>	
Seguro Corriente	<input type="text" value=""/>	
Seguro Furgoneta	<input type="text" value=""/>	

moz-extension://5cb1b1c6-1a7d

http://localhost:9545

Account 2 → 0x8BD...2e65

CONTRACT INTERACTION

◆ 0

DETAILS DATA

GAS FEE ◆ 0.001281 No Conversion Rate Available

TOTAL ◆ 0.001281 No Conversion Rate Available

REJECT CONFIRM

- En cuanto se valide la transacción podrá verificar en la parte inferior el nuevo valor.

The screenshot shows a form titled "Asegurador Mapfre" with a sub-instruction: "Ingrese los nuevos valores en los campos. Registre el valor actual de aquellos valores que no desea cambiar". It contains several input fields for "Seguro Premium" with placeholder values like "Precio Km Ciudad: 10", "Precio Km Carretera: 0", and "Precio Aparcado: 0". Each field has a green "Actualizar" button to its right.

3.3. Empresa financiera

Una vez el usuario se ha dado de alta como usuario tipo Financiera, podrá tener acceso a la zona de la empresa financiera. A continuación se explicarán las funcionalidades de éste módulo.

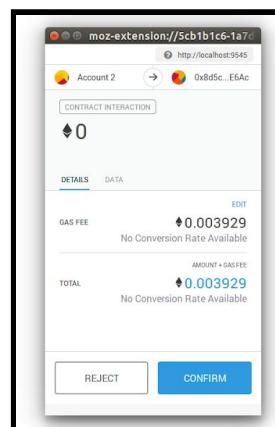
3.3.1. Comprar Tokens

Para la compra de Tokens se deben seguir lo siguiente pasos:

1. Ingresas el número de Tokens y pinchar sobre la opción comprar Tokens.

The screenshot shows a modal window titled "Bienvenido 'Leasing Smart Token' (LST)". It displays the message "El precio de este token es 0.001 Ether. Actualmente posees 1000000 LST." Below is a red-bordered input field containing the value "1" and a green "Comprar Tokens" button.

2. Confirmar la compra de Tokens. De acuerdo a la cantidad de Tokens y al precio definido para cada uno será el costo de la transacción.



Anexo II - Detalle técnico

Mappings

A continuación se van a describir los aspectos técnicos fundamentales relativos al mapping y registro de los datos:

- Los mappings que actúan como registros para guardar la información que se generan en las transacciones son los siguientes:
 - Mapping de **usuario**: Sus valores se generan en el proceso de alta del usuario. Se trata de un mapping dirigido por el address del usuario de la Dapp que se registra. Comentando adicionalmente, que este mapping posee los siguientes campos:
 - **DNI** asociado al cliente (de tipo string).
 - **Código de identificación fiscal** (de tipo string también).
 - **Tipo de usuario** (siendo este valor 1 si es de tipo cliente, 2 si es de tipo empresa aseguradora y 3 si es de tipo empresa financiera).
 - **Record**: Campo de tipo entero que se utiliza para calcular el precio por kilómetro (por gama de coche y por modalidad de uso: carretera, aparcado o ciudad).
 - Los 3 campos siguientes (que son de tipo entero sin signo) almacenan precisamente estos **precios** mencionados en el punto anterior. Al haber hasta 5 tipos de gama de coche: premium, luxury, classic, corriente y furgoneta y 3 modalidades de uso (carretera, ciudad y estacionado), se pueden establecer hasta 15 tipos distintos de precio.
 - El campo **MaxCredito** forma parte del módulo financiero y permite establecer el límite de crédito máximo que se le va a proporcionar al cliente (en función de las características de dicho crédito como el importe total o la demora en el pago).
 - El campo **IdCoche** hace referencia al vehículo que está siendo usado por el cliente (y correspondería a una clave foránea del mapping de coches).
 - Finalmente el campo **Préstamo** corresponde al valor del crédito que el cliente pediría a la empresa financiadora.

Es importante señalar que la Dapp desarrollada tiene la funcionalidad de piloto, en el que las empresas asociadas están limitadas:

- Para el caso de las empresas aseguradoras tendríamos 3 entidades: predefinidas
 - MAPFRE (IdSeguro=0).
 - ALLIANZ (IdSeguro=1).
 - AXA (IdSeguro=2).
- Para el caso de las empresas financieras se ha limitado a sólo una: Entidad Financiera S.L.

Estos valores predefinidos quedan fijados en la pantalla de alta de usuario (mediante una carga del comboBox correspondiente: nombre de la empresa).

Adicionalmente existe un **mapping de coches** que está direccionado por dos índices enteros (el tipo de gama del coche) y el identificador de coche (o IdCoche) . Este mapping almacena los siguientes campos:

- **IdCoche:** Clave primaria del mapping de coches.
- **KmCarretera:** Número de kilómetros recorridos por carretera por el vehículo. Para esta fase inicial del proyecto se ha definido un array de kilómetros de forma preestablecida, pero para fases posteriores, la idea sería integrar este registro mediante un dispositivo IoT presente en cada vehículo de la flota perteneciente a cada una de las gamas.
- **KmCiudad:** Número de kilómetros recorridos por ciudad por el vehículo. Para esta fase inicial del proyecto se ha definido un array de kilómetros de forma preestablecida, pero para fases posteriores, la idea sería integrar este registro mediante un dispositivo IoT presente en cada vehículo de la flota perteneciente a cada una de las gamas.
- **TiempoAparcado:** Tiempo en el que permanece estacionado el vehículo. Para esta fase inicial del proyecto se ha definido un array de tiempos de forma preestablecida, pero para fases posteriores, la idea sería integrar este registro mediante un dispositivo IoT presente en cada vehículo de la flota perteneciente a cada una de las gamas.
- **IdSeguro:** Correspondría al identificador de la compañía de seguros (como se ha mencionado anteriormente, existirían hasta 3 empresas).
- **Entregado:** Valor booleano que refleja si el vehículo ha sido entregado (ver sección 3.1.4 del manual de usuario). De esta forma representamos dos posibles estados:
 - **Coche disponible** (o ya entregado y validado).
 - **Coche en uso** (entregado=false).

Al respecto señalar que cuando un coche es entregado, se produce la actualización correspondiente (mediante un push) en el mapping de:

- **Coches disponibles:** Mapping que almacena por gama el IdCoche de los vehículos que se encuentran disponibles (por tanto es un mapping de entero a array de enteros).

Adicionalmente mencionar que cuando un coche es entregado y validado, se produce un evento (mediante la función: CocheValidado). En el caso de asignarse un coche a un usuario (ver sección 3.1.3), se produciría otro evento (mediante la función: CocheValidado).

Finalmente comentar que, además, existen mappings adicionales que registran la siguiente información:

- **aseguradoraPrecioAparcado:** Mapping de precios por kilómetro aparcado. El parámetro de entrada al mapping sería el id de la empresa aseguradora y la salida el precio que establece esta misma empresa si el coche está estacionado (ver sección del manual 3.2.1).
- **aseguradoraPrecioCarretera:** De forma análoga al caso anterior, este mapping sería el precio establecido en carretera por la entidad aseguradora correspondiente.

- **aseguradoraPrecioCiudad:** Finalmente, este mapping sería el precio establecido en ciudad por la entidad aseguradora.
- **maestroEmpresas:** Mapping que relaciona el hash del código de identificación fiscal con el address del usuario que dentro de la aplicación está asociada a esta empresa (un empleado o un administrador, por ejemplo). Este mapping se utilizará para localizar a la empresa que proporcionó el seguro cuando el cliente realice la entrega del coche (y así liquidar el coste del seguro) o para el financiamiento de tokens por parte del usuario final por parte de la empresa financiera.
- **ownerCuentaLeasing:** Registra el número de usuarios (clientes o empresas) asociados a la dirección de la cuenta del owner de la empresa de Leasing.

Funciones de los contratos

Existen **más de 20 funciones** distintas distribuidas en 2 contratos (CompraToken.sol y Token.col) que llevan la gestión de aspectos como:

- Gestión de usuarios y modificaciones de sus datos de perfil.
- Transacciones, envío de tokens y verificaciones de balances.
- Limitaciones en el número de tokens asignados.
- Carga y recuperación de datos de los mappings.
- Seguridad y control de contratos (por ejemplo: desactivación de emergencia de contratos y controles de desbordamientos por operaciones aritméticas).
- Modificaciones de los precios de las aseguradoras.
- Recuperación de datos de los mappings.
- Seguridad y control en la ejecución de los contratos.

A continuación se detallan los aspectos básicos concernientes a estas funciones:

- Gestión de usuarios y modificaciones de sus datos del perfil:
 - **NewUser:** NewUser(uint TypeUser, string memory DNI, bytes32 VATNumber, uint record)
 - Funcionalidad: Creación del registro correspondiente a un alta de cliente en el mapping de usuarios.
 - Parámetros de entrada: Tipo usuario, el DNI, el número de identificación fiscal y un parámetro que nos da el número de puntos del usuario.
 - Salidas: La función emite un evento con dos variables significativas.
 - **NewEmpresa:** NewEmpresa(uint TypeUser, string memory DNI, bytes32 VATNumber, uint record)

- Funcionalidad: De forma análoga al caso anterior, correspondería al alta de un usuario asociado a una de las empresas aseguradoras o financiera (y almacenamiento en el mapping correspondiente).
 - Parámetros de entrada: Tipo usuario, el DNI, el número de identificación fiscal y un parámetro que nos da el número de puntos del usuario.
 - Salidas: La función emite un evento con dos variables significativas.
- **NewCoche:** NewCoche(uint tipoCoche, uint IdCoche)
 - Funcionalidad: Dar de alta un coche determinado (perteneciente a una de las 5 gamas). Esta funcionalidad se lleva a cabo desde la pantalla de alta de coches.
 - Parámetros de entrada: Identificador de la gama del coche e Id del coche.
 - Salidas: Emisión de evento.
- **ValidarCoche:** ValidarCoche(uint tipoCoche, uint IdCoche)
 - Funcionalidad: Registrar el cambio de estado de un coche perteneciente a una de las gamas y actualizar el mapping correspondiente de coches disponibles.
 - Parámetros de entrada: Identificador de la gama del coche e Id del coche.
 - Salidas: Emisión de evento.
- **EliminarValorArray:** EliminarValorArray(uint tipoCoche)
 - Funcionalidad: Reajusta el mapping de coches disponibles de una de las gamas, quitando el id de uno de los coches (el que ha sido entregado y validado).
 - Parámetros de entrada: Identificador de la gama.
 - Salidas: Emisión de evento con la longitud del array final y devolución del array de ids de los coches disponibles en la gama.
- **deleteUser:** deleteUser(address dir)
 - Funcionalidad: Elimina el registro correspondiente en el mapping de usuarios. Dicho registro estará direccionado por el address que actúa como parámetro de entrada.
 - Parámetros de entrada: Address de direccionamiento del mapping.
 - Salidas: Emisión de evento y devolución de booleano.
- Límites en el número máximo de tokens a asignar:
 - **asignaCreditoMaximo:** asignaCreditoMaximo(uint maxCredit)
 - Funcionalidad: Asigna al usuario que realiza la llamada a esta función (el sender) un importe máximo de crédito a solicitar (en función del cálculo realizado en la función “actualizaCréditoMaximo” de app.js).

- Parámetros de entrada: Máxima cantidad a asignar.
 - Salidas: Emisión de evento y devolución de booleano.
- Modificaciones de los precios de las aseguradoras:
 - **actualizarPrecioAparcado, actualizarPrecioCarretera y actualizarPrecioCiudad:**
 - Funcionalidad: En los 3 casos estas funciones permiten modificar los precios que definidos por las aseguradoras para circulación por carretera (**actualizarPrecioCarretera**) , por ciudad (**actualizarPrecioCiudad**) o estacionado (**actualizarPrecioAparcado**).
 - Parámetros de entrada: ID de la empresa aseguradora y precio a asignar.
 - Salidas: En los 3 casos se emite un evento con el precio asignado al seguro (uint).
- Transacciones, envío de tokens y verificaciones de balances:
 - **Transfer:** transfer(address _to, uint256 _value)
 - Funcionalidad: Realiza la transacción y balance de cuentas entre el sender y la dirección destino (_to). El importe de la transacción se define en el segundo parámetro.
 - Parámetros de entrada: Dirección de destino e importe de la transacción.
 - Salidas: Evento con la dirección del sender, dirección del destinatario e importe. Si todo es correcto se devuelve un booleano (true).
 - **transferFrom:** transferFrom(address _from, address _to, uint256 _value)
 - De forma similar al caso anterior, permite realizar una transacción a la dirección destino (_to), aunque en este caso, se permite parametrizar el sender.
 - **transferInicial:** transferInicial(address _to, uint256 _value)
 - Funcionalidad: utilizamos esta función para enviar la cantidad inicial de tokens desde el owner del contrato a la dirección del contrato.
 - Parámetros de entrada: Dirección de destino del contrato e importe de la transacción.
 - Salidas: Evento.
 - **compraTokens:** compraTokens(uint256 _numeroTokens)
 - Funcionalidad: Transferimos los tokens del address del contrato al sender (quien realice la transacción). Esta función utiliza “transfers” del contrato Token.sol.
 - Parámetros de entrada: Número de tokens a transferir..
 - Salidas: Evento con la dirección del sender y el número de tokens comprados.

- **prestamoTokens:** prestamoTokens(uint256 _numeroTokens, address AddressEmpresa)
 - Funcionalidad: Transferimos el dinero de la empresa financiera al sender. Esta función hace uso de “TransferFrom” del contrato “Token.sol” y del valor máximo que se le ha permitido financiar al usuario (almacenado en el campo MaxCredito).
 - Parámetros de entrada: Número de tokens a transferir y address de la empresa financiera.
 - Salidas: Evento con la dirección del sender y el número de tokens comprados.
- **prestamoTokens:** pagarTokens(uint _pago, uint precioKmCiudad, uint precioKmCarretera, uint precioAparcado, uint tipoCoche, uint IdSeguro)
 - Funcionalidad: Se realiza la liquidación del importe del seguro. La transferencia se realiza desde el cliente a la empresa aseguradora. En este caso, el coche entregado se queda en estado disponible.
 - Parámetros de entrada: Cantidad a pagar, precio del seguro en carretera, ciudad y estacionado, gama del coche e ID de empresa aseguradora.
 - Salidas: Evento con la dirección del sender y el número de tokens pagados.
- Recuperación de datos de los mappings:
 - **fetchTypeUser:** fetchTypeUser()
 - Funcionalidad: Recupera el tipo de usuario teniendo en cuenta el address del sender.
 - Parámetros de entrada: N/A.
 - Salida: El tipo de usuario (uint).
- Seguridad y control en la ejecución de los contratos:
 - **CheckAdmin:** CheckAdmin()
 - Funcionalidad: Verifica si el que llama a esta función es el owner del contrato.
 - Parámetros de entrada: N/A.
 - Salida: Booleano.
 - **ActivarContratoCompraToken y DetenerContratoCompraToken:**
 - Funcionalidad: Activan o desactivan el SmartContract.
 - Parámetros de entrada: N/A.
 - Salida: Booleano.

Frontales

Dentro de la Dapp existen hasta 15 tipos de frontales html distintos que regulan la interacción del usuario final con la Blockchain. Estas son:

- **Admin.html:**
 - Funcionalidad: Pantalla de consulta y borrado de usuarios. Esta pantalla es accesible únicamente por el owner del contrato.
 - Funciones del contrato implicadas: deleteUser
 - Mappings implicados: users.
- **Alta.html:**
 - Funcionalidad: Pantalla con los datos de registro de los usuarios (bien sea usuarios clientes o usuarios asociados a empresas).
 - Funciones de contrato implicadas: NewUser, NewEmpresa.
 - Mappings implicados: users, maestroEmpresas.
- **AltaCoches.html:**
 - Funcionalidad: Pantalla para dar de alta un coche perteneciente a una determinada gama (seleccionable mediante un combo). Esta pantalla es únicamente accesible por el owner del contrato.
 - Funciones de contrato implicadas: NewCoche.
 - Mappings implicados: coches, CochesDisponibles
- **Frontales de confirmación** (para confirmación de adquisición de vehículo):
 - Funcionalidad: Las pantallas de confirmación de adquisición del vehículo permiten asignar el coche al usuario concreto. En el frontal se puede observar la compañía aseguradora implicada y los precios para carretera, ciudad y estacionamiento. Además, se indica la gama del vehículo asociado.
 - Funciones de contrato implicadas: pagarTokens, CompraTokens.
 - Mappings implicados: Se realiza transacciones entre direcciones (cliente y empresa aseguradora).
- **EntregaCoches.html:**
 - Funcionalidad: Frontal que permite al owner validar un coche que previamente se ha entregado.
 - Funciones de contrato implicadas: Se consulta directamente el mapping del contrato de coches.
 - Mappings implicados: coches.

- **Frontales de seguros:**

- Funcionalidad: Muestra la información concerniente a las empresas aseguradoras presentes por cada gama. Con la selección del usuario se le redireccionará a éste a las pantallas de confirmación para la adquisición.
- No hay funciones de contrato implicadas ni mappings (la ejecución es a nivel de html y javascript).

- **T_C.html:**

- Funcionalidad: Proporcionar información al usuario final respecto a los términos y condiciones para la cesión de información .
- No hay funciones de contrato implicadas ni mappings (la ejecución es a nivel de html y javascript).

- **WebCoches.html:**

- Funcionalidad: Frontal que permite acceder a la información de cada vehículo (dependiendo de la gama).
- Funciones de contrato implicadas: Se consulta directamente el mapping del contrato de coches.
- Mappings implicados: coches.

- **Financiación.html**

- Funcionalidad: Establecer la cantidad máxima de crédito a un usuario concreto (en función de las características de éste como el importe total o la demora en el pago).
- Funciones de contrato implicadas: asignaCreditoMaximo.
- Mappings implicados: users.

- **Financiación.html**

- Funcionalidad: Frontal principal en el que los distintos módulos aparecen en función del tipo de usuario que lo visualiza (cliente, empresa aseguradora o financiera).
- Funciones de contrato implicadas: todas.
- Mappings implicados: todos.