

15 SEPTIEMBRE 2019



# AutoInsuranceChain

## EQUIPO DE TRABAJO:

Min Yuan Chen  
Nicolás Quintero  
Virginia Saco  
Emma Gutiérrez  
Claudia Jara  
Loreto Heres

# ÍNDICE

<b>ESTADO DEL ARTE. MERCADO ACTUAL.....</b>	<b>1</b>
<b>RESUMEN EJECUTIVO DEL PROYECTO .....</b>	<b>4</b>
<i>PROPÓSITOS POR CUMPLIR.....</i>	<i>5</i>
<b>BUSINESS MODEL .....</b>	<b>6</b>
<i>PROBLEMAS.....</i>	<i>8</i>
<i>CLIENTES OBJETIVO.....</i>	<i>8</i>
<i>PROPUESTA DE VALOR.....</i>	<i>8</i>
<i>SOLUCIÓN.....</i>	<i>8</i>
<i>CANALES .....</i>	<i>9</i>
<i>FUENTES DE INGRESOS.....</i>	<i>9</i>
<i>ESTRUCTURA DE COSTES .....</i>	<i>9</i>
<i>RECURSOS CLAVE .....</i>	<i>10</i>
<i>ELEMENTOS DIFERENCIADORES.....</i>	<i>10</i>
<b>DESCRIPCIÓN DEL PRODUCTO.....</b>	<b>11</b>
<i>FLUJO AUTOINSURANCE CHAIN .....</i>	<i>11</i>
EVALUACIÓN DEL CLIENTE .....	12
ADQUISICIÓN DEL SEGURO .....	12
INSTALACIÓN DEL DISPOSITIVO .....	13
RECOPILACIÓN DE LA INFORMACIÓN.....	13
ANÁLISIS DE LA INFORMACIÓN.....	14
COBRO DE PRIMA .....	15
<i>ARQUITECTURA DEL SISTEMA .....</i>	<i>15</i>
<b>FUNDAMENTOS TECNICOS.....</b>	<b>18</b>
<i>HYPERLEDGER .....</i>	<i>18</i>
<i>RED PERMISIONADA .....</i>	<i>18</i>
<i>HYPERLEDGER FABRIC .....</i>	<i>19</i>
ROLES.....	19
FLUJO DE TRANSACCIONES .....	20
CANALES.....	20
BASE DE DATOS DE ESTADO.....	20
<i>HYPERLEDGER COMPOSER .....</i>	<i>20</i>
ARQUITECTURA DE RED EMPRESARIAL .....	21

<i>HYPERLEDGER COMPOSER PLAYGROUND</i> .....	22
<b>DAPP – AUTOINSURANCE CHAIN</b> .....	<b>23</b>
<i>DEFINICIÓN DE LA RED</i> .....	23
ORGANIZACIONES.....	23
ESTRUCTURA INTERNA .....	24
<i>APLICACIÓN DESCENTRALIZADA PARA SEGUROS</i> .....	24
ARCHIVO DE MODELO DE RED (.CTO) .....	25
ARCHIVO DE PERMISOS (.ACL) .....	29
ARCHIVO DE JAVASCRIPT (.JS).....	30
OPERATIVA DE DESPLIEGUE Y USO DE LA APLICACIÓN .....	35
DESPLIEGUE DE LA RED DE NEGOCIO .....	36
VISUALIZACION DE LA RED DE NEGOCIO.....	38
<i>TECHNICAL ROADMAP</i> .....	44
FASE CERO.....	45
FASE MACHINE LEARNING .....	45
FASE DESPLIEGUE .....	46
FASE DE ESTUDIO DE PARÁMETROS ADICIONALES .....	46
<b>FINANCIACIÓN</b> .....	<b>47</b>
<b>EQUIPO FUNDADOR</b> .....	<b>48</b>
<b>PLAN MARKETING</b> .....	<b>49</b>
PÁGINA WEB .....	49
ASISTENCIA A FERIAS .....	50
VISITAS PRESENCIALES A LOS PRINCIPALES CLIETES POTENCIALES.....	50
<b>ANEXO I. COSTES SERVIDOR</b> .....	<b>51</b>
<b>ANEXO II. ARCHIVOS DESPLIEGUE RED</b> .....	<b>52</b>

## ILUSTRACIONES

ILUSTRACIÓN 1. CERTIFICACIÓN AENOR DE LOS SEGUROS YCAR DE MAPFRE .....	3
ILUSTRACIÓN 2 SEGURO DE VEHÍCULOS PERSONALIZADO.....	11
ILUSTRACIÓN 3 FLUJO AUTOINSURANCE CHAIN .....	11
ILUSTRACIÓN 4. ARQUITECTURA DEL SISTEMA .....	15
ILUSTRACIÓN 5. DIAGRAMA HYPERLEDGER COMPOSER .....	21
ILUSTRACIÓN 6. INFRAESTRUCTURA DE RED HYPERLEDGER FABRIC.....	23
ILUSTRACIÓN 7 BUSINESS NETWORK ARCHIVE .....	25
ILUSTRACIÓN 8 BLOCKCHAIN-EXPLORER. CONFIG.JSON.....	38
ILUSTRACIÓN 9 HYPERLEDGER COMPOSER REST SERVER. LISTADO .....	39
ILUSTRACIÓN 10 HYPERLEDGER COMPOSER REST SERVER. TRANSACCIÓN INSUREDREGISTER .....	40
ILUSTRACIÓN 11 HYPERLEDGER EXPLORER. TRANSACCIONES .....	41
ILUSTRACIÓN 12 HYPERLEDGER COMPOSER REST SERVER. TRANSACCIÓN VEHICLEREGISTER .....	42
ILUSTRACIÓN 13 HYPERLEDGER EXPLORER. TRANSACCIÓN .....	43
ILUSTRACIÓN 14. ROADMAP .....	44



## ESTADO DEL ARTE. MERCADO ACTUAL

El sector de los seguros supuso en el año 2018, y según la UNESPA (Asociación Empresarial del Seguro), el 5.3% del Producto Interior Bruto (PIB) español. Pero su importancia va más allá de una cifra, su importancia radica en la generación de flujos monetarios, centrados sobre todo en sus clientes y proveedores, que son de una gran importancia para la economía nacional.

Uno de los sectores de los seguros más potente es el de automóviles, ya que la demanda de pólizas de automóviles es mucho mayor que la de cualquier otro seguro. El seguro del coche es uno de los gastos fijos que tiene cualquier propietario de un vehículo, ya que, por ley, es obligatorio para cubrir unas coberturas mínimas en caso de accidente.

En el año 2017, la media de lecturas mensuales del Fichero Informativo de Vehículos Asegurados (FIVA) sitúa el parque asegurado en dicho año en los 30.295.290 vehículos en España (según UNESPA)

En España, el seguro del automóvil está regulado por el Real Decreto Legislativo 8/2004, de 29 de octubre, por el que se aprueba el texto refundido de la Ley sobre responsabilidad civil y seguro en la circulación de vehículos a motor. Esta regulación indica que todos los vehículos a motor del estado español deben contar con un seguro que cubra, al menos, La Responsabilidad Civil Obligatoria. Esta obligación afecta a todos los vehículos a motor inscritos en la Dirección General de Tráfico.

El seguro de Responsabilidad Civil Obligatoria tiene como objetivo principal cubrir cualquier daño a terceros que podamos causar mientras conducimos nuestro coche. Los daños causados a terceros, tal y como recoge el Reglamento del Seguro Obligatorio de Responsabilidad Civil en la Circulación de Vehículos a Motor, pueden ser de dos tipos:

- Físicos o personales: Es el daño causado a personas, ya sea a cualquiera de los ocupantes de otro vehículo como a un peatón.
- Materiales: Cualquier daño causado en otro coche, propiedad o en el mobiliario urbano.

Tener un vehículo supone muchas ventajas a nivel de movilidad, pero también muchos quebraderos de cabeza cuando tenemos problemas.

Para ello, hay muchos mecanismos en los que trabajar y que las compañías se marcan como objetivos: crear productos que respondan a sus necesidades concretas, facilitar los procesos de contratación y tramitación, simplificar el lenguaje, aprovechar al máximo las nuevas tecnologías y ofrecer una atención personalizada y diferenciada.

Actualmente unas 36 empresas aseguradoras en España comercializan seguros para coche. Las condiciones varían sustancialmente en función de cada aseguradora, el tipo de seguro elegido y el modelo de vehículo a asegurar. Ya que dependiendo de lo que quiera el usuario asegurar y el tipo de automóvil que conduzca, los gastos que debe asumir la aseguradora en caso de indemnización varían.

Otro de los factores que determina el precio de la póliza de seguros, es el conductor del vehículo, de forma que cuando un usuario pide presupuesto de un seguro de coche, la aseguradora hace un perfil de dicho usuario: Edad, Sexo, Profesión, Años de carné, Ámbito geográfico, Estado Civil, etc.



A todo esto, hay que sumarle el historial que tienen las aseguradoras de sus propios conductores asegurados, de forma que priman o penalizan según dicho historial.

Este tipo de baremación para el cálculo de la póliza de seguros es bastante injusto, sobre todo a la hora de catalogar a los propios conductores. Las compañías de seguros discriminan a los conductores por edad, sexo, años de carné y no tienen en cuenta la forma real de conducción de los asegurados. No por tener cierta edad, sexo, o más o menos experiencia, tu conducción es más o menos imprudente, y, por tanto, no tienes que pagar en base a esos datos. Además, estos seguros no discriminan por cantidad de tiempo que utilizas el coche, pagando lo mismo, conduzcas todos los días, o eventualmente.

La realidad de hoy en día es que la variedad a la hora de conducir un coche por parte de los usuarios es tan amplia como la cantidad de usuarios existentes, y las generalidades por características demográficas no reflejan dicha realidad. Los colectivos más desfavorecidos por los seguros clásicos son:

- ◆ Jóvenes menores de 23 años, sean o no el conductor principal.
- ◆ Padres con hijos jóvenes que aparezcan como segundos conductores en la póliza.
- ◆ Conductores que realicen menos de 7.000 kilómetros al año.
- ◆ Conductores que no tengan historial siniestral o sólo con 1 o 2 siniestros de Responsabilidad Civil con culpa.
- ◆ Propietarios de vehículos de alta gama.

Todo esto está llevando a las propias compañías de seguros a plantear nuevos modelos para el cálculo de las primas de las pólizas de seguros, sacando al mercado soluciones personalizadas denominadas “Pay As You Drive – Paga según tú conduces”. Este tipo de soluciones implican la monitorización de tus hábitos de conducción, de forma que dependiendo de una serie de parámetros objetivos: km recorridos, horario de conducción, velocidad media, aceleraciones, tiempo en marcha, tipo de vía, etc., la prima a pagar sufre un descuento, según las aseguradoras, considerable (de hasta un 40%).

La compañía de seguros Mapfre comercializa desde 2009 su producto “YCar”, destinado a conductores jóvenes, y que instala un dispositivo electrónico en el vehículo asegurado. Los datos obtenidos de dicho dispositivo permiten ajustar la prima de la póliza correspondiente, premiando los “buenos” hábitos de conducción mediante una reducción de dicha prima en **la renovación del seguro**.

La compañía de seguros Verti sin embargo, comercializa su seguro “CuentaKMS” donde lo único que tiene en cuenta en tu conducción es los km que recorres, existiendo tres packs de km disponibles: 1.000, 2.000 y 3.000 km, y la monitorización se realiza a través del teléfono móvil.

Algunos de estos seguros premian una conducción responsable y segura, favoreciendo al usuario con una mejora sustancial del precio de su póliza de seguros y favoreciendo a las compañías, que pueden ver reducida la siniestralidad de sus asegurados e incrementando la fidelidad del cliente a la compañía.

Estos seguros comercializados actualmente no han sufrido un gran boom entre el público, ni se han convertido en una preferencia de los usuarios. En redes sociales son muchas las críticas a este tipo de seguros, siendo uno de los principales problemas que el usuario ve en estos sistemas, la utilización que realizan las aseguradoras de todos los datos recogidos, así como la fiabilidad que tiene el usuario de que esos datos están bien recogidos y tratados de manera adecuada. Y se añade la queja generalizada del poco ahorro que se consigue efectivamente.



Para conseguir la confianza por parte de los usuarios, las aseguradoras anuncian la privacidad de ciertos datos mediante entidades certificadoras que aumentan los gastos en tiempo y dinero:

### CERTIFICACIÓN AENOR



Recuerda que la información referida a velocidades y tiempos medios, kilómetros recorridos, franjas horarias y tipo de vías recogida en el terminal telemático es gestionada por un proveedor externo, bajo las máximas medidas de seguridad. Por tanto, MAPFRE no tiene la posibilidad de identificar al conductor asegurado ni de conocer la posición exacta del vehículo, salvo por accidente o robo.

Esta garantía de privacidad  
está certificada por AENOR

**AENOR**

Certifica que:

"La información relativa a los hábitos de conducción que recoge MAPFRE en el seguro YCAR no contiene referencias alguna a la posición exacta del vehículo, y que esta posición se es proporcionada únicamente en caso de accidente o robo del vehículo."



*Ilustración 1. Certificación AENOR de los seguros YCAR de Mapfre.*



## RESUMEN EJECUTIVO DEL PROYECTO

Las nuevas tecnologías están transformando rápidamente todas las industrias, y la industria de los seguros no es distinta. Existen nuevos modelos de negocios y tecnologías disruptivas y revolucionarias que pueden ser cruciales para la transformación de la industria de los seguros: **Blockchain**.

Con los volúmenes de datos personales en juego, la industria de los seguros está actualmente atascada por innumerables comprobaciones, verificaciones y, sobre todo, está impedida para la obtención de gran cantidad de información de una forma fiable.

Toda esa información puede ser registrada y almacenada de manera inmutable en una cadena de bloques, de forma que todas las comprobaciones necesarias se realicen de manera automática. La información sería fiable y podría fluir directamente de una parte a otra. Los usuarios podrían comprobar que todos sus datos de conducción son reales y no han sufrido modificación, a la vez que se puede proteger su privacidad.

Los ahorros en tiempo y dinero serían muy considerables, y eso es lo que transformará definitivamente la industria.

Con el manejo de datos sensibles en el sector de los seguros, y teniendo en cuenta que dichos datos solo se querrán revelar a partes interesadas de la red, y como queremos definir la lectura de dichos datos por los participantes en dicha red, necesitaremos una Blockchain permissionada.

Una Blockchain permissionada es un ecosistema cerrado donde cada participante está definido. Este tipo de Blockchain nos permite crear organizaciones o consorcios de organizaciones para intercambiar eficientemente información y un historial de transacciones.

Junto a la blockchain, otras tecnologías en auge en este momento, como las funcionalidades del IOT y el análisis Big data, darán un valor añadido a nuestro producto. Recogiendo tanto información en tiempo real como asociada al historial de los conductores, las primas se pueden calcular de forma más rápida y precisa que nunca, y dicha información quedar inmutable de forma confidencial para futuros estudios. La información relativa a las condiciones de mantenimiento del coche, la forma de conducción del asegurado (acelerones, cambios bruscos de carril, etc.), el tipo de carreteras que utiliza, el tiempo al volante, conducción diurna o nocturna... pueden contribuir a una evaluación de riesgos increíblemente eficiente. De esta forma, los riesgos que corre la aseguradora a la hora de cubrir una póliza de seguros se reducirán de forma drástica basándose en datos reales de conducción.

De esta forma, la información recogida sobre la conducción de los usuarios, una vez tratada para conseguir privacidad, será transaccionada sobre la cadena de bloques asociada al número de identificación privada del usuario, de forma, que dichos datos estarán inmutables y consultables por el usuario para comprobaciones posteriores. La compañía aseguradora, utilizará dichos datos para ajustar las primas de las pólizas de seguros de sus clientes, además de estudios posteriores que le permitirán realizar modelos de Machine Learning, que le permitirá reajustar todavía más dichas primas, siendo más competitiva, con seguros con un riesgo totalmente medido.

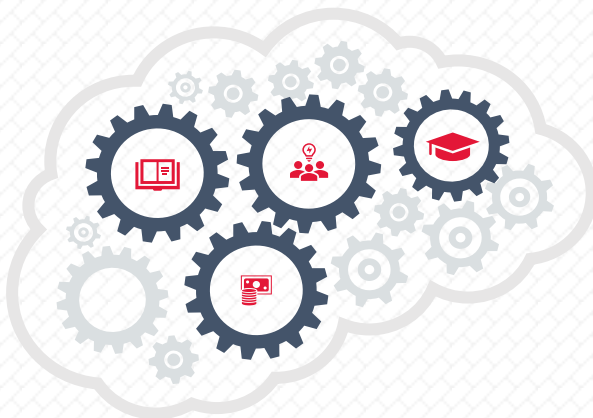
La cadena de bloques hace posible nuevas formas de pensar acerca de los seguros. Permite una cotización totalmente orientada al cliente, ya que los clientes pueden personalizar sus hábitos de conducción en un grado hasta ahora desconocido.





Y como beneficio añadido, las mejoras en las primas de los seguros, puede conseguir que los conductores adecuen su forma de conducir para la consecución de esa mejora, lo que conlleva una reducción de la peligrosidad al volante, así como una disminución en las emisiones, tan importante hoy en día.

## PROPÓSITOS POR CUMPLIR



### INMUTABILIDAD

Habrà un libro común e inmutable donde quede registrada toda la información sin posibilidad de modificación.



### SIMPLIFICAR LAS VERIFICACIONES

Tener una base de datos fiable conlleva que podrá ser utilizada por todas las partes implicadas para verificaciones y comprobaciones.



### OBTENCIÓN DE INFORMACIÓN FIABLE

Conseguir información fiable y su utilización mediante técnicas Big Data para un reajuste en el cálculo de las primas de las pólizas de seguros.



### AHORRO DE TIEMPO Y DINERO

Los anteriores propósitos, suponen un considerable ahorro de tiempo y dinero.



## BUSINESS MODEL

- Modelos de cálculo de riesgo de las aseguradoras con baja fiabilidad con los datos que se manejan actualmente. Poco ajustados a la realidad.
- Primas de seguros de coche excesivamente altas para el uso y forma de conducción del asegurado.
- Falta de confianza de los asegurados con las aseguradoras.

• **Alternativas existentes:** No existen alternativas actualmente que de una forma fiable establezcan una prima de seguro de coche realmente adaptada al conductor y que satisfaga el riesgo de las aseguradoras

### PROBLEMAS



- Se dispone de datos totalmente fiables de la conducción de los asegurados lo que permite realizar modelos de riesgo muy ajustados con la realidad
- Las primas de los seguros se pueden ajustar a la conducción del asegurado, con el beneficio claro para el asegurado.
- Dejar grabado en una blockchain los datos da confianza a los asegurados con respecto al trato recibido de sus datos.

### SOLUCIÓN



- Capital humano
- Desarrolladores y analistas con grandes conocimientos en Big Data y Blockchain
- Recursos financieros e intelectuales
- Infraestructura informática

### RECURSOS CLAVE



- Se crea una base de datos real sobre conducción y partes de siniestros, generando modelos predictivos con una alta fiabilidad.
- Mejora para el cliente del precio obtenido para asegurar su vehículo, y mejora para la aseguradora, ajustando sus precios al riesgo real, consiguiendo una ventaja competitiva importante
- Los datos guardados en Blockchain es el elemento diferenciador del resto de compañías, implementando una tecnología disruptiva. La Blockchain asegura y certifica que los datos de conducción son reales e inmutables.
- Añadimos otro elemento diferenciador en la inclusión del mantenimiento del coche en el pago del seguro. Un conductor responsable y cuidadoso, conseguirá ventajas sustanciales

### PROPUESTA ÚNICA DE VALOR





- *Blockchain: Tecnología puntera solo dominada por unos pocos profesionales. Datos inmutables.*
- *Revisión de la prima mes a mes, totalmente flexible y dependiente del asegurado.*
- *Mejora sustancial de las propuestas que hay hoy en día en el mercado*

#### ELEMENTOS DIFERENCIADORES



- *Página WEB*
- *Entrevistas individualizadas con aseguradoras*
- *Presentaciones en eventos del sector.*

#### CANALES



- *Aseguradoras dispuestas a ofrecer un producto nuevo y actual basado en nuevas tecnologías a sus clientes.*
- *Empresas de otros sectores que necesiten una solución para grandes flotas de vehículos*

#### CLIENTES OBJETIVO



- *Venta del producto.*
- *Adecuación del producto para otros sectores como negocios con gran flota de vehículos que requieran un control de sus conductores.*

#### FUENTE DE INGRESOS



- *Costes del montaje de la infraestructura necesaria para sostener la Blockchain.*
- *Costes de servidores y equipación para la gestión y tratamiento de los datos.*
- *Equipo de desarrolladores y mantenimiento de la Blockchain y Big Data, incluyendo formación para el personal propio.*
- *Costes de marketing.*

#### ESTRUCTURA DE COSTES





## PROBLEMAS

Actualmente las aseguradoras no cuentan con un modelo fiable para el cálculo del riesgo real que corren al asegurar un vehículo con los datos recogidos del asegurado. Por este motivo, los precios que pueden ofrecer son elevados en muchos de los casos, ya que, a falta de datos fiables, tienen que sobreestimar las primas para cubrir los partes de siniestralidad que aparecen. Pierden competitividad en el momento en el que aparezca un producto que solvante ese problema.

Los seguros ofertados actualmente son genéricos en cuanto a sexo, edad, años de carné, y no se ajustan a la forma de conducir real de los usuarios, ni al uso que hacen éstos de sus vehículos.

Por otro lado, el asegurado desconfía de las aseguradoras, ya que no consideran que reciban un trato justo, de forma que la pillería se convierte en una conducta habitual para tratar con los seguros.

Con las alternativas que existen hoy en día, el asegurado cree que se vulnera su privacidad, lo que fomenta aún más la desconfianza en las aseguradoras.

## CLIENTES OBJETIVO

La solución va dirigida a aseguradoras que quieran dar respuesta a un gran número de clientes desconformes con los paquetes ofrecidos actualmente. Con este producto conseguirán ajustar al modo de conducción del asegurado la prima, a la vez que minimizar los riesgos por ese mismo seguro. Ganarán competitividad frente a otras aseguradoras, disminuyendo costes y aumentando las ganancias.

## PROPUESTA DE VALOR

Se crea una base de datos totalmente fiable sobre los hábitos de conducción y uso de su vehículo de los asegurados, que permite la creación de modelos de predicción con tecnología Machine Learning de gran robustez, que marcará con alta probabilidad el riesgo asumido por las aseguradoras a la hora de firmar una póliza de seguros.

A su vez, se ofrece un producto al asegurado que le garantiza una prima ajustada a su manera de conducir mes a mes, premiando a aquellos que conduzcan y se conduzcan de una forma adecuada.

Los datos se grabarán en una Blockchain de forma pseudo anónima, elemento diferenciador por el uso de una tecnología disruptiva, que ofrece al asegurado una inmutabilidad y certificación de sus datos y mejorando la confianza de éstos en las aseguradoras.

Como añadido, la red de talleres certifica el estado de mantenimiento en el que se encuentra el vehículo asegurado, siendo esto una ventaja significativa del cliente con respecto a la aseguradora, a la vez, que ese buen mantenimiento de la flota de vehículos es una garantía para la aseguradora para accidentes producidos por un mal mantenimiento de los vehículos.

## SOLUCIÓN

La aseguradora contará con una herramienta de gran fiabilidad para ofertar un producto personalizado a sus clientes, ofreciendo ventajas económicas a aquellos conductores con buenos hábitos, a la vez que le permite ajustar



el riesgo corrido por asegurar vehículos. La disminución de ese riesgo asumido por la aseguradora, que conlleva un beneficio económico para el asegurado, también conlleva un beneficio para la aseguradora, económico y de competitividad frente a la competencia.

## CANALES

Los interesados en esta herramienta son las aseguradoras directamente, por lo que los canales serán muy direccionados y personalizados, ya que el número de éstas no es elevado en nuestro país.

De cualquier forma, se diseñará una página WEB, con videos informativos y demostraciones de la herramienta. Se mostrará las ventajas de la herramienta, para aseguradoras y asegurados.

Se realizarán presentaciones individualizadas, teniendo como contacto las distintas aseguradoras de nuestro país.

## FUENTES DE INGRESOS

La principal fuente de ingreso será la venta de la herramienta a aseguradoras, que permitirá el completo desarrollo de la aplicación personalizado según las exigencias del comprador.

La idea de monitorizar y guardar en una blockchain los datos de conducción puede ser extrapolable a otros sectores que no sea el de aseguradoras. De esta forma, se puede adecuar la herramienta para empresas o sectores con una gran flota de vehículos y que necesiten tener asegurada las buenas prácticas de los conductores.

## ESTRUCTURA DE COSTES

El grosor de los costes está compuesto por el montaje de la infraestructura, servidores y demás elementos necesarios para el procesamiento de los datos, su tratamiento y posterior grabación en la blockchain. También existen costes del equipo de desarrolladores y de mantenimiento de la infraestructura, así como su formación.

De los costes arriba mencionados, los costes imputados por la red de Hyperledger dedicada a la aseguradora, serán soportados por la propia aseguradora, estos son:

Los costes por montaje de infraestructura, servidores, etc. son costes dedicados a la red Hyperledger, los cuales serán trasladados al cliente, una vez adquiera la solución. Se incluye:

- ◆ Cuotas a IBM Blockchain Platform para poder desarrollar en Hyperledger. Estos costes dependen de la necesidad de la aseguradora en cuanto a número de peers y flujo de transacciones necesarias para su operativa diaria.
- ◆ Servidores dedicados a soportar la red Hyperledger y recepción de datos de los dispositivos IoT. Pueden ser servidores físicos o cloud.

Adicionalmente los servicios de mantenimiento, soporte, capacitación, se proporcionarán por parte del equipo bajo contratación

Los gastos de Marketing se encuentran detallados en el 'Plan de Marketing'.



## RECURSOS CLAVE

El recurso principal necesario son desarrolladores y analistas con gran experiencia en Big Data y Blockchain, así como en dispositivos IoT.

Es necesario además una infraestructura importante informática con servidores para albergar, tanto la blockchain, como para poder realizar el análisis y transmisión de los datos de conducción obtenidos.

## ELEMENTOS DIFERENCIADORES

La utilización de Blockchain, tecnología puntera solo dominada por unos pocos profesionales, unida al tratamiento y análisis de datos, hace de esta herramienta un producto novedoso, único y diferenciador en el mundo de los seguros de coches, de difícil imitación a corto plazo.

El producto ofrecido a los usuarios finales de primas adecuadas totalmente a su forma de conducir mes a mes, con la garantía de la blockchain, marca una gran diferencia con los productos ofertados en el mercado en este momento, por lo que la aseguradora que oferte este producto en primer lugar tendrá una gran ventaja competitiva.





## DESCRIPCIÓN DEL PRODUCTO

AutoInsuranceChain es una plataforma que permite la gestión de seguros de automóviles bajo el esquema “Paga como conduces”. La plataforma está montada sobre la infraestructura de HyperLedger Fabric, aprovechando las ventajas propias de esta plataforma y las consiguientes de la utilización de una blockchain.

Nuestro producto consiste en un seguro de vehículos personalizado a la manera de conducción del asegurado y el mantenimiento de su coche, que deja plasmada la información imprescindible de dichos datos en una blockchain permissionada, de forma que la manipulación de datos del asegurado sea imposible por parte de la aseguradora. De esta forma, el usuario tendrá un producto totalmente adaptado a él, con garantías de inmutabilidad y donde se podrá hacer un seguimiento seguro de su historial de conducción.



*Ilustración 2 Seguro de vehículos personalizado*

## FLUJO AUTOINSURANCE CHAIN

En el siguiente diagrama se presenta el flujo general que se seguiría en AutoInsurance Chain desde la obtención de seguro hasta el pago de la Prima.



*Ilustración 3 Flujo Autoinsurance Chain*



---

## EVALUACIÓN DEL CLIENTE

En la etapa “Evaluación del cliente” se empiezan siguiendo los procedimientos usuales de una aseguradora tradicional, ya que se necesita un punto de partida, y hay partes de la evaluación clásica que son básicos para este proyecto.

La información básica sobre el vehículo a asegurar es esencial para la aseguradora, ya que los gastos que tiene que sufragar la aseguradora en caso de siniestro depende directamente de esos datos. Entre los datos del vehículo necesarios están el modelo y la marca, el número de matrícula, aditamentos de seguridad de serie y elementos extras del modelo básico, si el coche pernoctará en garaje o al aire libre, etc.

Estos datos de partida iniciales y necesarios para la evaluación del riesgo que corre la aseguradora a la hora de suscribir una póliza con el asegurado serán los que marcarán la parte fija de la prima que tendrá que pagar el asegurado por el seguro contratado.

Por otro lado, no existe una evaluación del cliente a nivel personal, no interesa edad, sexo, años de carné de conducir, etc. salvo a modo informativo, ya que no son datos relevantes para determinar el modo de conducción del asegurado. Los datos recogidos mientras el asegurado conduce, serán los que marquen cuál es su modo de conducción. Estos datos marcarán la parte variable de la prima que tendrá que pagar el asegurado por el seguro contratado.

Un último parámetro a tener en cuenta a la hora de evaluar al cliente será el historial que éste posea en los últimos cinco años, según información recogida en el Fichero Histórico del Seguro del Automóvil (SINCO) que pone a disposición de las aseguradoras la sociedad gestora de Tecnologías de la Información y Redes para las Entidades Aseguradoras (TIREA). Estos datos obtenidos de SINCO se utilizarán para penalizar o premiar al asegurado.

Todos estos datos no son necesarios plasmarlos en la blockchain. Solo aquellos totalmente imprescindibles para garantizar la certificación de la conducción.

Se plasmará en la blockchain el usuario con un único número de identificación, de forma que los datos personales del asegurado no sean públicos, y no se pueda infringir la ley de Protección de datos.

El vehículo quedará identificado en la blockchain mediante un número de identificación, modelo, marca, color, potencia, matrícula, año de fabricación, y número de identificación del propietario.

---

## ADQUISICIÓN DEL SEGURO

Hay algunas características de los contratos de seguros que deben cumplirse para que una póliza tenga validez y sea de acuerdo con la normativa vigente.

Cuando un usuario adquiere un seguro, tanto la aseguradora como el cliente, deben conocer y aceptar expresamente las condiciones del contrato, de forma que la información debe ser comprensible y adecuada sobre los productos aseguradores a los asegurados.

El contrato establece obligaciones para cada una de las partes, tales como el pago de la prima por parte del cliente y la prestación de coberturas de la aseguradora.

Una de las características del contrato de Seguros es que ninguna de las partes sabe con seguridad si ocurrirán o no aquellos sucesos que están asegurados, ni cuándo se producirán.





Este elemento de azar es fundamental en los Seguros, al igual que la confianza: de la misma forma que esperamos que nuestro Seguro asuma la cobertura de un siniestro, la compañía parte de que el asegurado no provocará circunstancias que den lugar a que se produzca el hecho asegurado.

La póliza de seguros se confecciona con los datos del asegurado y los datos del coche, aunque no se perfecciona por el momento, ya que es necesario la instalación de los dispositivos correspondientes para que esto ocurra.

En la blockchain se identificará la póliza de seguros con un número, el número de identificación del asegurado y el número de identificación del vehículo correspondiente.

De esta forma, en caso de que el usuario cambie de vehículo, la póliza se podrá invalidar y dar de alta una nueva póliza con el mismo número de identificación del asegurado y el nuevo vehículo.

Con la confección de esta póliza de seguros, la aseguradora tomará del asegurado los datos financieros necesarios para el pago de las primas, cobrando en primera estancia una primera prima basada en los datos obtenidos en la evaluación de cliente. Una vez empiecen las mediciones de la forma de conducción, esta primera prima se adecuará a los datos reales del conductor mes a mes.

---

#### INSTALACIÓN DEL DISPOSITIVO

Una vez que se ha formalizado la póliza del seguro entre la aseguradora y el asegurado, se le hace entrega al cliente de tres dispositivos diferentes de medición. Estos dispositivos serán de fabricación dispar de forma que se minimice el error en la toma de medidas por deterioro aleatorio de fabricación.

Los dispositivos se colocarán en el coche con fácil instalación, mediante unos soportes y será el usuario el que una vez finalizada su colocación dará de alta dichos dispositivos mediante una interface cliente que escaneará el código QR de los dispositivos. En ese momento empieza la medición de la conducción para el seguro y se perfecciona el contrato, entrando en vigor el mismo y marcando los plazos de renovación y periodos de vigencia de dicha póliza.

La perfección del contrato es el momento en que el mismo comienza a obligar a cumplir lo pactado.

En la blockchain se activará la póliza de seguros que se ha creado anteriormente, asociando dichos dispositivos a esa póliza.

Los datos recibidos desde los tres dispositivos serán tratados y modificados adecuadamente, de forma que, si en algún momento se detectara algún fallo en alguno de los dispositivos, éste será sustituido en el menor tiempo posible y sus datos desechados.

Que un dispositivo esté deshabilitado temporalmente, no conlleva la anulación de la póliza de seguros, pero la aseguradora se encargará de hacer llegar al usuario un dispositivo de sustitución y de recoger el defectuoso, de forma que el usuario deberá volver a escanear el código QR del dispositivo defectuoso y del nuevo, así se añadirá el nuevo dispositivo a la póliza de seguros vigente, eliminando el dispositivo defectuoso, y quedando plasmado en la blockchain.

---

#### RECOPILACIÓN DE LA INFORMACIÓN

El proceso “Recopilar Información” es el encargado de reunir la información de conducción del asegurado y del coche correspondiente. La información se recopila de dos fuentes diferentes:



- Dispositivo de medición IOT: Recopila toda la información referente a las costumbres de conducción del asegurado. Los datos serán: aceleración lineal, aceleración lateral, posición, km recorridos, hora conducción (diurna o nocturna), tiempo conduciendo continuo por trayecto (si el vehículo se para más de media hora, se considera un nuevo trayecto).

Todos estos datos serán tratados. Las aceleraciones de los tres dispositivos se compararán, comprobarán que no difieren más de un 3% y se calculará la media, que será el dato que quede grabado en la blockchain. La posición definida por los dispositivos se tratará y transformará en tipo de carretera por la que se circula, de forma que en la blockchain solo quedará grabado si se transita por carretera tipo 1, 2 o 3. Esta clasificación del tipo de carretera vendrá marcado por las características de la carretera y su definición por la DGT. De esta forma no se tendrá constancia de la ubicación del asegurado.

La velocidad de conducción no quedará registrada en ningún sitio para no criminalizar al asegurado, ya que con las aceleraciones y el tiempo de aceleración determinaremos el riesgo en la conducción del asegurado.

El resto de los parámetros quedarán registrados en la blockchain.

- Taller: El cliente puede ir voluntariamente a los talleres asociados a la compañía para la realización de los mantenimientos respectivos del coche. Se motivará ir a los talleres ya que el mantener el coche en un buen estado redundará en el coste del seguro.

Una vez finalizada la visita de mantenimiento, el taller y el cliente firmarán una transacción mediante una interface en la que subirán información del estado del vehículo en la blockchain. Se comprobarán distintos aspectos del estado del vehículo como neumáticos, estado amortiguación, aceite, estado general, y se clasificará dicho estado en Bueno, Regular, Malo, que será lo que quedé grabado en la blockchain, a la vez que se emita una notificación.

- Partes de siniestro: Cada vez que el usuario de un parte de accidentes, la aseguradora dejará esa información reflejada en la blockchain, y pasará al historial del cliente.

---

## ANÁLISIS DE LA INFORMACIÓN

El procedimiento de análisis de la información consta de dos partes:

1. El envío de datos de los dispositivos a una base de datos, asociada con los datos del asegurado.

Cada 5 segundos los dispositivos IoT envían los datos recogidos a una base de datos de la aseguradora. Estos datos serán tratados de acuerdo con las condiciones estipuladas en la póliza de seguros. Se comprobará el buen funcionamiento de los dispositivos comparando los tres aparatos asociados con la póliza, se hará la media de las aceleraciones recibidas, se estipulará la categoría a la que pertenece la carretera recorrida, se marcará si la conducción es diurna o nocturna, etc. Con todos estos datos y los datos recibidos por la red de talleres, se generará un mapa de conducción del asegurado y al finalizar el mes se calculará la prima correspondiente según ese mapa. Los datos tratados, se grabarán en la blockchain quedando inmutables y certificando la calidad de la conducción del asegurado.

En una etapa posterior, la aseguradora contará con datos totalmente fiables sobre la incidencia que hay sobre los partes de siniestralidad, dependiendo de la forma de conducción de los asegurados, pudiendo realizar mediante análisis de Machine Learning modelos predictivos que permitan ajustar las primas de riesgo de los asegurados, aumentando la competitividad frente a otras aseguradoras ya que reducirá significativamente los riesgos.

2. La transferencia de los datos concretos anteriormente mencionados a la blockchain.



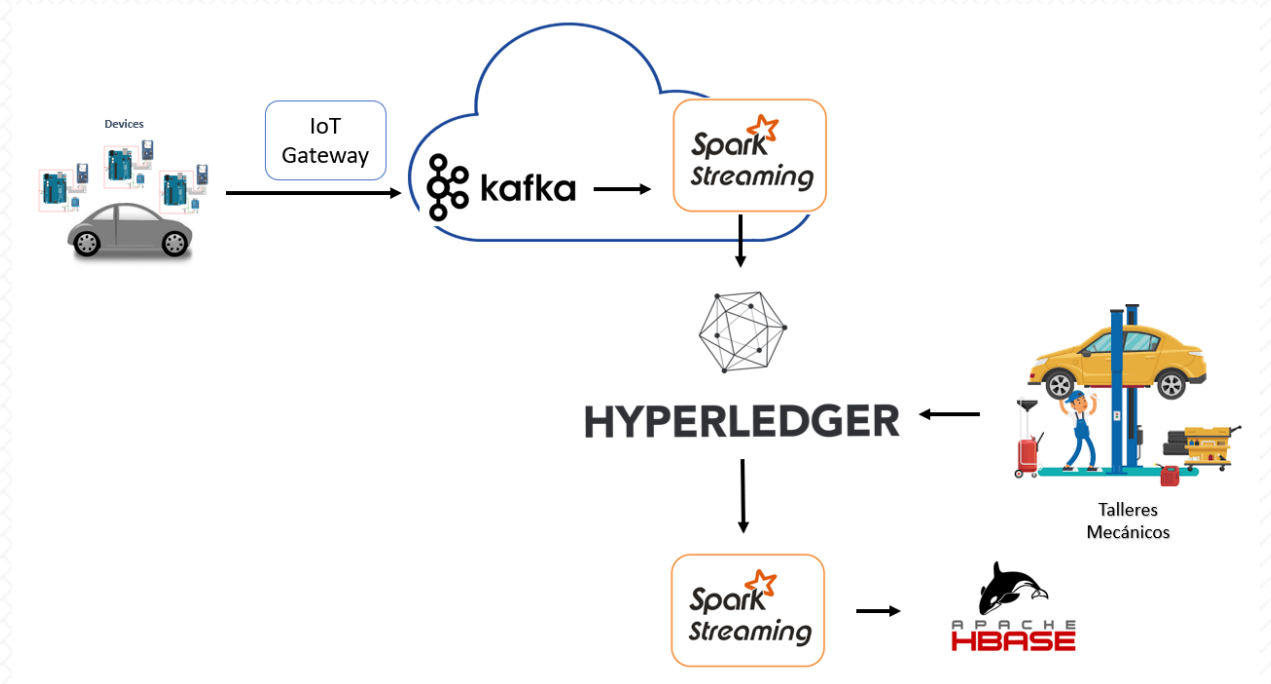
Teniendo en cuenta las características de una blockchain y la ley de protección de datos, los datos que se registrarán en dicha blockchain solo se identificarán con un número de identificación del asegurado, de forma que no se vulnere su privacidad.

### COBRO DE PRIMA

Como se ha comentado anteriormente, en un primer momento y al perfeccionamiento de la póliza de seguros, se estipula el pago de la primera cuota del seguro correspondiente al primer mes, y sin datos de conducción.

Una vez analizada la información recibida mes a mes, las siguientes cuotas mensuales se ajustarán al comportamiento del asegurado, de forma que unos datos de conducción adecuados redundarán en un coste menor de la prima del seguro del vehículo a pagar.

### ARQUITECTURA DEL SISTEMA



*Ilustración 4. Arquitectura del sistema*

El ciclo de nuestra arquitectura comienza con el flujo de datos procedentes del vehículo del asegurado que son recogidos por medio de los dispositivos IOT para terminar siendo enviados a la red de Hyperledger a través de Apache Kafka. Entre Apache Kafka y Hyperledger se hace una transformación de los datos en Spark Streaming: se comprueba que los dispositivos de un mismo asegurado no tienen diferencias significativas entre ellos y se genera el tipo de carretera por el que circula el asegurado a partir de la posición GPS recogida por los dispositivos.

Apache Kafka es un proyecto de intermediación de mensajes de código abierto desarrollado por la Apache Software Foundation escrito en Java y Scala. El proyecto proporciona una plataforma unificada, de alto rendimiento y de baja latencia para la manipulación en tiempo real de fuentes de datos. Puede verse como una cola de mensajes,



bajo el patrón publicación-suscripción, masivamente escalable concebida como un registro de transacciones distribuidas, lo que la vuelve atractiva para nuestra infraestructura de aplicación empresarial.

En nuestro caso, el uso de Apache Kafka es fundamental ya que sirve de transporte de la gran cantidad de datos real-time recogidos por los dispositivos IoT, de manera eficiente y escalable. Esto es, múltiples dispositivos IoT recogen a tiempo real (o muy alta frecuencia) datos sensoriales de los vehículos. Toda esta información es “empujada” por un gateway al Apache Kafka cluster. En Kafka los datos sensoriales se procesan en diferentes 'topics' que correspondería a cada métrica del vehículo.

A continuación, se pasan a detallar los componentes del sistema y su funcionamiento:

Como se menciona anteriormente "Kafka" se puede definir como un producto complejo, que, en primera instancia, se podría considerar como una cola distribuida de mensajes.

Kafka funciona con un sistema basado en topics, donde cada tema recoge todos los mensajes que han sido asociados con él, actuando como una especie de sistema nervioso, cuya misión es distribuir la información dentro de la plataforma, haciendo llegar a cada proceso, aquello que realmente le interesa.

Profundizando, podemos destacar las siguientes características:

- ⌘ Kafka se ejecuta como un clúster en uno o varios servidores que pueden estar en distintos centros de datos.
- ⌘ Los registros se almacenan en categorías llamadas “topics”.
- ⌘ Cada registro se compone de una clave un valor y un timestamp.

El rendimiento de Kafka es constante con respecto al tamaño de los datos, por lo que almacenar datos durante mucho tiempo no es un problema.

Las particiones que realiza Kafka se distribuyen a través de los servidores, donde cada partición se replica en un número determinado de servidores para tolerancia a fallos.

En el flujo entre Apache Kafka e Hyperledger realizaremos las transformaciones de los datos de los dispositivos con Spark Streaming como hemos descrito anteriormente

Apache Spark Streaming es un sistema de procesamiento de transmisión escalable y tolerante a fallas que admite tanto las cargas de trabajo por lotes como las de transmisión. Spark Streaming es una extensión de la API principal de Spark permitiendo procesar datos en tiempo real de varias fuentes, incluida (entre otras) Kafka. Estos datos procesados serán volcados sobre la red de Hyperledger.

Las ventajas de Spark Streaming con respecto a otros sistemas son:

- ❖ Recuperación rápida de fallas y rezagados
- ❖ Mejor equilibrio de carga y uso de recursos.
- ❖ Combinación de transmisión de datos con conjuntos de datos estáticos y consultas interactivas.
- ❖ Integración nativa con bibliotecas de procesamiento avanzadas (SQL, aprendizaje automático, procesamiento de gráficos)

Los datos contenidos en la red de Hyperledger van a ser explotados posteriormente mediante las siguientes herramientas:



Spark Streaming, nos permite trabajar con flujos de datos, o MLlib, con los que se elaboran a través de HBase con el que conecta perfectamente, ya que es una base de datos distribuida no relacional, que se ejecuta sobre HDFS, algoritmos de Machine Learning.

En nuestro caso particular, para potenciar la arquitectura Big Data vamos a utilizar Hbase para guardar los datos en bruto, de cara a poder explotarlos en estudios posteriores con Machine Learning.

Finalmente, vamos a explotar los datos de Hyperledger con las queries que se programan en los smart contracts. Esto es, destinaremos algunos peers de la aseguradora que no sean de anclaje o de ejecución de chaincodes (encoders), específicamente para ejecutar queries.



## FUNDAMENTOS TECNICOS

### HYPERLEDGER

La tecnología blockchain/DLT está ganando aceptación en el mundo empresarial por su enorme potencial y ventajas frente a otras tecnologías tradicionales. En muchos casos, las soluciones blockchain públicas no cumplen con algunos requisitos de escalabilidad y privacidad necesarias. Hyperledger nació con el objetivo de cubrir estas necesidades y se ha convertido en una de las soluciones más maduras y aceptadas en este ámbito.

Hyperledger es una plataforma blockchain de código libre gestionado por The Linux Foundation y creado con el objetivo de crear una solución blockchain para el sector empresarial que permita reducir costes y complejidad a la hora de hacer negocios.

Hyperledger consiste en ocho proyectos, cinco de los cuales son frameworks DLT y otros tres son módulos que dan soporte a dichos frameworks.

FRAMEWORKS	MÓDULOS
Hyperledger <b>Fabric</b>	Hyperledger <b>Composer</b>
Hyperledger <b>Iroha</b>	Hyperledger <b>Explorer</b>
Hyperledger <b>Sawtooth</b>	Hyperledger <b>Cello</b>
Hyperledger <b>Burrow</b>	
Hyperledger <b>Indy</b>	

Todos los frameworks de Hyperledger incluyen:

- 🔗 Un registro distribuido inmutable.
- 🔗 Un algoritmo de consenso para decidir el estado del registro.
- 🔗 Privacidad de transacciones y acceso permissionado.
- 🔗 Chaincode para la lógica de negocio.

### RED PERMISSIONADA

Hyperledger es una blockchain permissionada.

En nuestro caso, no es necesario la forma de incentivos o la existencia de criptomonedas o token como si necesitan las redes públicas.

Una red permissionada como Hyperledger, reduce el riesgo dejando unirse a la red solo a participantes conocidos que deben ser aprobados por el resto de los participantes. Si los participantes de una blockchain permissionada descubren que uno de los participantes está actuando malintencionadamente siempre pueden denegarle el acceso por medio de un consenso del resto de participantes. En nuestro caso, el principal interesado será una aseguradora que tendrá el control sobre el acceso a la red del resto de participantes.





Hyperledger está pensado para redes blockchain con un reducido número de participantes conocidos, lo que le permite usar algoritmos de consenso mucho más eficientes que los algoritmos en redes públicas alcanzando un rendimiento mucho mayor que los alcanzados en dichas redes. Dicho esto, estas tecnologías no son comparables con las redes públicas que cuentan con miles de nodos y permiten transaccionar entre participantes completamente desconocidos y por tanto con consensos mucho más exigentes.

Hyperledger implementa la privacidad de transacciones a través de lo que se conocen como canales, una especie de subredes de participantes dentro de la propia red. La privacidad es necesaria para ciertos casos de uso como en los sectores financieros y de la salud.

En definitiva, Hyperledger provee características clave de una blockchain como registro único, inmutabilidad y robustez a la vez que incluye características necesarias en un entorno empresarial como escalabilidad y privacidad.

## HYPERLEDGER FABRIC




Hyperledger Fabric es una implementación de un framework blockchain y uno de los proyectos dentro de Hyperledger. Se caracteriza por tener una arquitectura modular y altos niveles de confidencialidad, resiliencia, flexibilidad y escalabilidad. Tiene, además, un registro compartido, transacciones y Smart contracts, que en este caso se denominan chaincode. Es una blockchain permissionada y los participantes de la red deben unirse a través del Membership Service Provider (MSP) o Proveedor de Servicio de Membresía.

El MSP es un componente que define las reglas para validez, autenticar y permitir el acceso a la red a una identidad o participante. Usa Certificate Authority (CA) y el interfaz por defecto es Fabric-CA API. Este componente es fácilmente reemplazable lo que hace a Hyperledger Fabric muy flexible a la hora de usar un mecanismo de identificación u otro.

---

## ROLES

Existen tres tipos de roles en una red de Fabric:

-  **Cientes:** son aplicaciones que actúan en nombre de una persona a la hora de proponer transacciones, es decir, permite a los usuarios finales la comunicación con la blockchain.
-  **Peers:** mantienen el estado de la red y una copia del registro. Dentro de este rol existen tres tipos:
  - **Endorsers:** simulan y avalan transacciones propuestas, es decir, verifican los detalles del certificado y las funciones del solicitante.
  - **Committers:** verifican las propuestas de transacciones y validan el resultado de las transacciones antes de grabarlas en la blockchain. Dentro de este tipo de peers, existe la figura del **Anchor** que sirve de conexión entre el Ordering Service y el resto de peers.
-  **Ordering Service:** recibe las transacciones propuestas, las ordena dentro de un bloque y lo transmite al anchor que lo reparte entre el resto de peers.



---

## FLUJO DE TRANSACCIONES

El flujo de las transacciones, como se llega a consenso y en que orden se graban en la blockchain se define a continuación:

### **Propuesta de transacciones:**

Una transacción se inicia con una aplicación Cliente que envía una propuesta de transacción a una serie de nodos Endorsers.

### **Simulación y respaldo de transacciones:**

Cada uno de los endorsers que ha recibido la propuesta de transacción simula la transacción con el estado actual del registro, pero sin hacer ningún cambio sobre éste, y genera un paquete denominado RW Set que contiene una lista de lecturas y escrituras generados por la transacción simulada. Este RW Set es firmado por el Endorser y devuelto a la aplicación cliente.

### **Ordenación de transacciones:**

La aplicación cliente envía entonces la transacción firmada por el Endorser y el RW Set al Ordering Service, el cual puede ser común a toda la red, o se conecta con el resto de Ordering Services.

El Ordering Service ordena las transacciones en un bloque que envía a todos los Committers a través del Anchor. Hyperledger Fabric incluye los mecanismos de ordenación.

Hyperledger Fabric incluye hoy en día varios mecanismos de ordenación (consenso) lo que permite que las transacciones se confirmen en un tiempo inferior a 0.5s.

### **Validación y grabado:**

Los Committers comprueban entonces que los RW Sets recibidos aún son válidos y general la misma lista de lecturas y escrituras. Si una transacción resulta inválida durante este proceso será incluida en el bloque, pero marcada como inválida y no modifica el estado del registro.

Por último, los Committers informan a los Clientes de si la transacción ha sido ejecutada con éxito o no.

---

## CANALES

Los canales es uno de los mecanismos de privacidad en Hyperledger Fabric y permite tener diferentes blockchain en la misma red de forma que solo los participantes de un canal pueden conocer los detalles de las transacciones que ocurren en dicho canal.

---

## BASE DE DATOS DE ESTADO

Hyperledger Fabric guarda el estado actual en una base de datos que puede ser recreada en cualquier momento a partir de la cadena de transacciones almacenadas en la cadena de bloques. Es una forma eficiente de acceder al estado del registro (world state) a través de una tabla de clave-valor. Actualmente Hyperledger usa por defecto LevelDB como base de datos que puede ser reemplazado por CouchDB. Mientras LevelDB almacena una lista de clave-valor como decíamos, CouchDB almacena objetos JSON y presenta una interfaz mucho más potente.

## HYPERLEDGER COMPOSER

Hyperledger Composer es un conjunto de herramientas basadas en JavaScript que simplifican y aceleran la creación de aplicaciones de blockchain de Hyperledger Fabric. Con esta tecnología, los propietarios de empresas y los desarrolladores pueden crear rápidamente chaincode (lógica empresarial) y aplicaciones de blockchain.





## ARQUITECTURA DE RED EMPRESARIAL

Hyperledger Composer ayuda a modelar rápidamente una red empresarial, lo que incluye sus activos existentes y las transacciones que están relacionadas con ellos. Para general dicha red empresarial, con Hyperledger Composer, solo se requiere de un archivos de modelo de red (.cto), un archivo de JavaScript (.js), un archivo de control de acceso (.acl) y un archivo de consultas (.qry)



### **Archivo de modelo de red**

Definirá los activos, las transacciones y los participantes que pueden interactuar con esos activos como parte del modelo de red empresarial. El archivo del modelo contiene definiciones de activos, participantes y transacciones.



### **Archivo de JavaScript**

Define las funciones de las transacciones.



### **Archivo ACL**

Contiene las reglas de control de acceso que definen los derechos de los diferentes participantes en la red empresarial.



### **Archivo de consultas**

Define la consulta que se puede ejecutar en una red.

Hyperledger Composer utiliza estos archivos para crear una definición de red empresarial que se puede empaquetar y exportar como un archivo. El archivo exportado es un Business Network Archive (.bna), que se puede implementar en una red existente de Hyperledger Fabric. El archivo BNA contiene funciones ejecutables del procesador de transacciones y se puede considerar como un contacto inteligente escrito en JavaScript, siendo posible escribir aplicaciones cliente utilizando APIs de Hyperledger Composer para acceder a opciones de BNA.

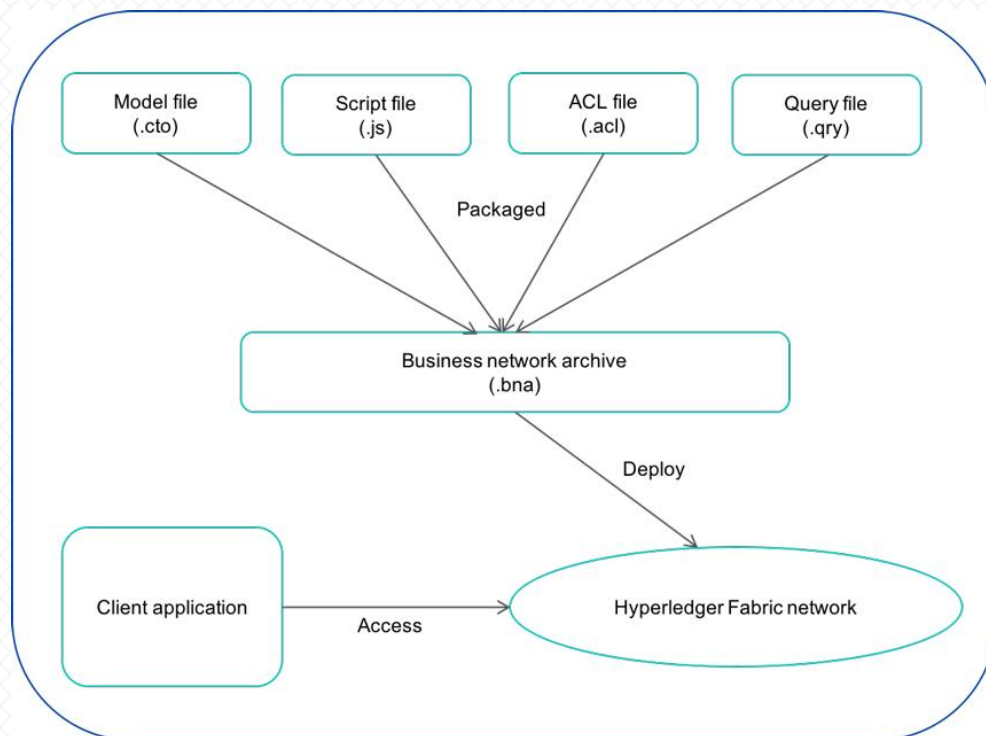


Ilustración 5. Diagrama Hyperledger Composer



## HYPERLEDGER COMPOSER PLAYGROUND

Hyperledger Composer también proporciona una interfaz de usuario web llamada “Playground” que se puede utilizar para crear modelos y probar la red empresarial. Playground utiliza el almacenamiento local del navegador para simular el almacenamiento de estado de la red de blockchain. Es un excelente punto de partida para las pruebas de concepto porque no necesita ejecutar una red de blockchain.

En resumen, Hyperledger Composer es la infraestructura de desarrollo de aplicaciones para construir aplicaciones de blockchain basándose en Hyperledger Fabric. En otras palabras, Hyperledger Composer ayuda a crear definiciones de red empresariales, que se implementan en Hyperledger Fabric, donde se ejecutan.

La belleza de Hyperledger Composer reside en que proporciona una capa de abstracción de alto nivel que se puede utilizar para crear el modelo de red empresarial, escribir funciones de JavaScript para las transacciones, y exponer esas funciones como REST APIs para el desarrollo de aplicaciones cliente. Debido a que el nivel de programación necesario es mínimo, los propietarios empresariales que no son necesariamente desarrolladores de aplicaciones pueden crear modelos y escribir fácilmente funciones de transacciones para las redes de blockchain.



## DAPP – AUTOINSURANCE CHAIN

### DEFINICIÓN DE LA RED

Para la creación de nuestra solución mediante Hyperledger Fabric se debería desplegar la siguiente infraestructura creando una red con los siguientes elementos:

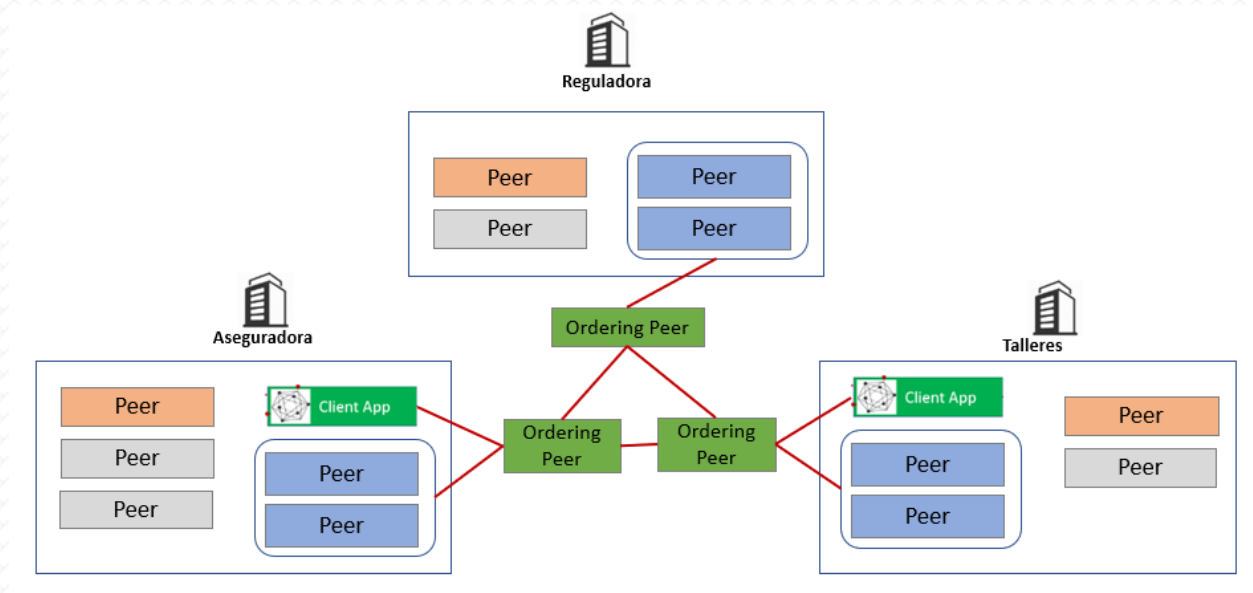


Ilustración 6. Infraestructura de red Hyperledger Fabric

### ORGANIZACIONES.

La red Hyperledger Fabric se construye a partir de los peers de las diferentes organizaciones que forman parte de la red y que contribuyen a ésta con los mismos. La red existe porque las organizaciones contribuyen a la misma con sus propios recursos individuales. Los peers tienen una identidad (*certificado digital*) asignada por un Proveedor de Servicios de Membresía (MSP) de la organización de la que son propiedad. Aunque los peers pueden pertenecer a distintos canales, en nuestro caso, no es necesario en un primer estado del proyecto la existencia de diferentes canales.

Contamos con las siguientes organizaciones:



#### **Aseguradora**

Es la organización principal y la mayor interesada en esta red ya que es la forma de conseguir un producto novedoso y puntero para ofrecer a sus clientes, consiguiendo una gran ventaja competitiva frente a sus competidores y llegando a liderar su sector.



#### **Talleres**

Otra de las organizaciones será una red de talleres asociados a la aseguradora, encargada del control del estado de los vehículos de los asegurados.



### **Regulador**

La última organización será la figura del regulador. Se contará con una entidad independiente y que forma parte de la red para asegurar que la organización principal, la aseguradora, no realiza modificaciones sobre la blockchain, ya que ésta es permissionada. Esta figura dará confianza y seguridad al asegurado.

---

## ESTRUCTURA INTERNA

Para cada una de estas organizaciones se debe establecer la siguiente estructura interna:

### **Aseguradora**

La aseguradora contará con un peer endorser encargado de ejecutar el chaincode y realizar autorizaciones, además de dos peers de anclaje (Anchor), que trabajan en paralelo, y adicionalmente nos permiten continuar con las transacciones en caso de que alguno de los dos tenga algún problema técnico. Estos peers de anclaje son los encargados de recibir información del orderer service y compartirla con todos los demás nodos de la organización. Los nodos generales simplemente cuentan con una copia de la blockchain.

### **Talleres**

La red de talleres contará también con un peer endorser encargado de ejecutar el chaincode y realizar autorizaciones, además de dos peers de anclaje (Anchor), que trabajan en paralelo, y adicionalmente nos permiten continuar con las transacciones en caso de que alguno de los dos tenga algún problema técnico. Estos peers de anclaje son los encargados de recibir información del orderer service y compartirla con el resto de los nodos de la organización. Esta organización solo tiene un nodo general que cuenta con una copia de la blockchain.

### **Regulador**

El regulador contará igual que el resto de las organizaciones con un peer endorser encargado de realizar autorizaciones, además de dos peers de anclaje (Anchor), que trabajan en paralelo, y adicionalmente nos permiten continuar con las transacciones en caso de que alguno de los dos tenga algún problema técnico. Estos peers de anclaje son los encargados de recibir información del orderer service y compartirla con el resto de los nodos de la organización. Esta organización solo tiene un nodo general que cuenta con copia de la blockchain.





Cada una de estas organizaciones tiene su propio orderer cuya objetivo principal es empaquetar las transacciones que se envían a los peers mediante un canal, pero no es capaz de validarlas, estableciendo así una comunicación entre los peers y los peers endorsers .

## APLICACIÓN DESCENTRALIZADA PARA SEGUROS

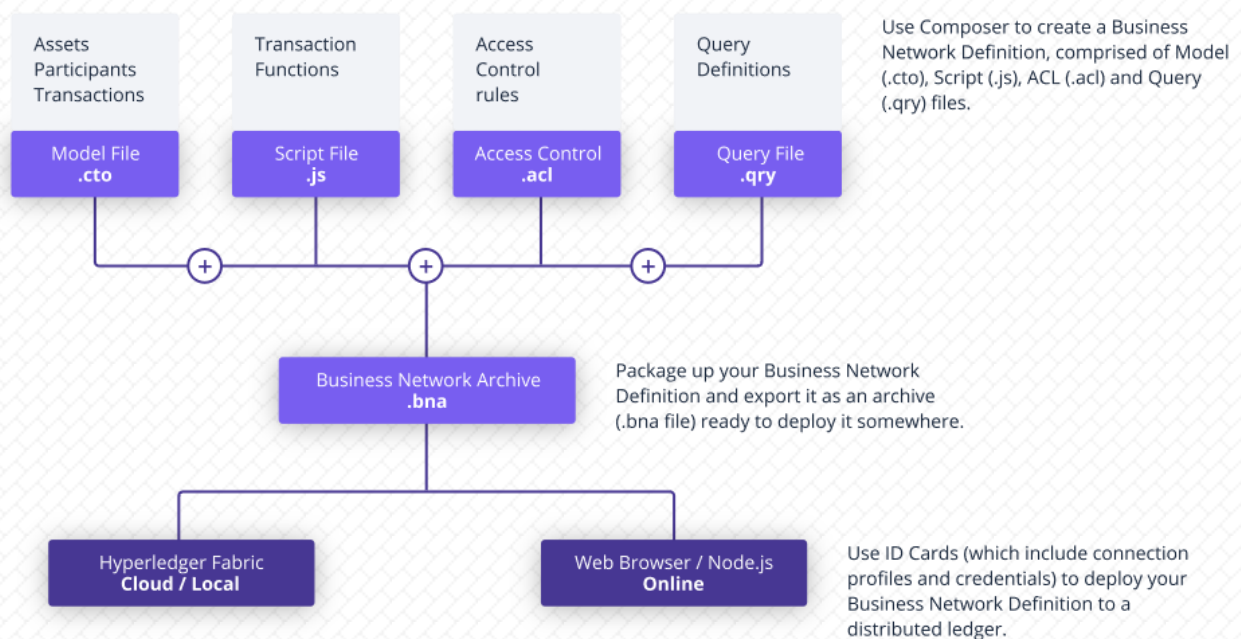
Para la creación de nuestra aplicación hemos utilizado Hyperledger Composer, que te permite realizar pruebas de concepto sobre una red de Hyperledger Fabric.



En Hyperledger Composer necesitas crear una serie de archivos que nos permite modelar rápidamente una red empresarial:

-  Archivo de modelo de red (.cto)
-  Archivo de JavaScript (.js)
-  Archivo ACL (.acl)
-  Archivo de consultas (.qry)

Con estos archivos se crea un archivo Business Network Archive (.bna) que ya podemos desplegar en una red existente de Hyperledger Fabric. Este archivo BNA contiene funciones ejecutables del procesador de transacciones y se puede considerar como un contrato inteligente escrito en JavaScript.



*Ilustración 7 Business Network Archive*

### ARCHIVO DE MODELO DE RED (.CTO)

En este archivo se definen los assets, las transacciones y los participantes que pueden interactuar con estos assets como parte del modelo de red empresarial.

En primer lugar, nombramos nuestra red:

`namespace org.autoinsurancechain`

Definimos objetos “enum” para definir distintos atributos de un tipo.

RoadType: donde clasificamos las carreteras en distintos tipos:



```
enum RoadType {
  o RoadOne
  o RoadTwo
  o RoadThree
  o RoadFour
}
```

StateVehicle: donde clasificamos el estado en el que puede estar el vehículo cuando son evaluados por los talleres:

```
enum StateVehicle {
  o StateGood
  o StateRegular
  o StateBad
}
```

State: donde clasificamos el estado en el que puede estar la póliza de seguros:

```
enum State {
  o StateInactive
  o StateActive
}
```

Definimos los participantes que son:

↔ Insurance (Aseguradora): definida por un número identificativo único y su nombre.

```
participant Insurance identified by idInsurance{
  o String idInsurance
  o String name
}
```

↔ Garage (Taller): definido por un número identificativo único, nombre comercial, ciudad en la que se encuentra y el código postal al que pertenece.

```
participant Garage identified by idGarage{
  o String idGarage
  o String commercialname
  o String city
  o String postcode
}
```

↔ Regulator (Regulator): definido por un número identificativo único y su nombre.

```
participant Regulator identified by idRegulator{
  o String idRegulator
  o String name
}
```

Definimos los assets:



- ↔ Insured (Asegurado): definido por un número identificativo único y anónimo, lo que nos garantiza el cumplimiento de la ley de protección de datos. Además, tenemos la variable owner de tipo Insurance, ya que el asegurado pertenece a la aseguradora.

```
asset Insured identified by idInsured {
  o String idInsured
  --> Insurance owner
}
```

- ↔ Vehicle (Vehículo): definido por un número identificativo único (nº del Bastidor), color, potencia, modelo, marca, año, matrícula, estado de vehículo (variable opcional) y una variable insured de tipo Insured, ya que el vehículo pertenece al asegurado.

```
asset Vehicle identified by idVehicle {
  o String idVehicle
  o String color
  o String horsepower
  o String model
  o String brand
  o String year
  o String carlicense
  o StateVehicle statevehicle optional
  --> Insured insured
}
```

- ↔ InsurancePolicy (Póliza de Seguros): definido por un número identificativo único, el estado de la póliza (enumerado anteriormente), una variable insuranceVehicle de tipo Vehicle, una variable insured de tipo Insured y una variable devicevehicle de tipo Device opcional.

```
asset InsurancePolicy identified by idInsurancePolicy {
  o String idInsurancePolicy
  o State statepolicy
  --> Vehicle insuranceVehicle
  --> Insured insured
  --> Device devicevehicle optional
}
```

- ↔ Device (dispositivos): definido por un número identificativo único, un array de números identificativos de los tres dispositivos, la hora (timestamp), la aceleración lineal, la aceleración lateral y el tipo de vía por la que se circula, opcional.

```
asset Device identified by idDevice {
  o String idDevice
  o String[] ids
  o String timestamp
  o Double linearacceleration optional
  o Double lateralacceleration optional
}
```



```
    o RoadType roadtype optional
}
```

Definimos las transacciones:

En primer lugar definimos una transacción setup que será la encargada de desplegar los participantes iniciales: la aseguradora, el regulador y un taller mínimo.

```
transaction setup {
}
```

Después tenemos la transacción del registro del asegurado, donde simplemente necesitamos el número identificativo de dicho asegurado:

```
transaction InsuredRegister {
  o String idinsurance
}
```

La transacción registro del vehículo necesita como parámetros todos los datos del vehículo, incluido el número identificativo del asegurado dueño de dicho vehículo:

```
transaction VehicleRegister {
  o String idVehicle
  o String color
  o String horsepower
  o String model
  o String brand
  o String year
  o String carlicense
  --> Insured insured
}
```

La transacción registro de la póliza necesita como parámetros el número identificado de la póliza, el vehículo y el asegurado del que será la póliza.

```
transaction PolicyRegister {
  o String idInsurancePolicy
  --> Vehicle insuranceVehicle
  --> Insured insured
}
```

La transacción identificación de los dispositivos con la póliza correspondiente, necesita en primer lugar el número identificativo de la póliza, y los números identificativos de los dispositivos.

```
transaction SignUpDevicePolicy{
  --> InsurancePolicy insurancepolicy
  o String idDevice
  o String[] ids
}
```





La transacción envío de información de los dispositivos necesita los datos que obtenemos de los dispositivos, aceleración lineal, aceleración lateral, tipo de carretera por la que circulamos, hora, y dispositivo que lo envía.

```
transaction DeviceSendInfo{
  o Double linearacceleration
  o Double lateralacceleration
  o RoadType roadtype
  o String Timestamp
  --> Device device
}
```

La transacción envío de información del taller necesita la póliza de seguro, y el estado del vehículo que evalúa:

```
transaction GarageSendInfo{
  --> InsurancePolicy insurancepolicy
  o StateVehicle statevehicle
}
```

La transacción actualización de los dispositivos de una póliza necesita los nuevos dispositivos asociados a la póliza, y el número identificativo de la póliza

```
transaction UpdateDevicePolicy{
  o String[] ids
  --> InsurancePolicy insurancepolicy
}
```

La transacción añadir taller, necesaria para poder incorporar talleres nuevos a nuestra red, necesita los datos del nuevo taller, desde su número de identificación, nombre comercial, ciudad y código postal de su ubicación.

```
transaction AddGarage{
  o String idGarage
  o String commercialname
  o String city
  o String postcode
}
```

---

#### ARCHIVO DE PERMISOS (.ACL)

Teniendo en cuenta que nuestro proyecto en esta etapa es una prueba de concepto ejecutada en Hyperledger Composer, el archivo de permisos es básico, concediendo permiso total a todos los usuarios, ya que al no estar totalmente desplegada la red, no compete la restricción de accesos a alguno de los participantes.

```
rule NetworkAdminInsurance {
  description: "Grant business network administrators full access to user
resources"
  participant: "ANY"
  operation: ALL
}
```



```
resource: "*"
action: ALLOW
}
```

---

#### ARCHIVO DE JAVASCRIPT (.JS)

En este archivo creamos las funciones de las transacciones, de forma que tenemos una función por cada una de las transacciones definidas en el archivo .cto

La función setup, asociada a la transacción del mismo nombre sería:

```
/**
 *
 * @param {org.autoinsurancechain.setup} setup - Setupinstance
 * @transaction
 */
async function setup(setup) { // eslint-disable-line no-unused-vars
  const factory = getFactory();
  const NS = 'org.autoinsurancechain';

  const insurance = factory.newResource(NS, 'Insurance', '00001');
  insurance.idInsurance = '00001';
  insurance.name = 'InsuranceBusiness';

  const insuranceRegistry = await getParticipantRegistry(NS + '.Insurance');
  await insuranceRegistry.addAll([insurance]);
  const garage = factory.newResource(NS, 'Garage', '00001');
  garage.idGarage = '00001';
  garage.commercialname = 'Taller Pepito';
  garage.city = 'Madrid';
  garage.postcode = '28009';
  const garageRegistry = await getParticipantRegistry(NS + '.Garage');
  await garageRegistry.addAll([garage]);

  const regulator = factory.newResource(NS, 'Regulator', '00001');
  regulator.idRegulator = '00001';
  regulator.name = 'REGULATOR';
  const regulatorRegistry = await getParticipantRegistry(NS + '.Regulator');
  await regulatorRegistry.addAll([regulator]);
}
```

La función insuredregistry, asociada a la transacción registro del asegurado sería:

```
/**
 *
```



```

* @param {org.autoinsurancechain.InsuredRegister} InsuredRegister -
InsuredRegisterinstance
* @transaction
*/
async function insuredregistry(InsuredRegister){
    // Get the vehicle asset registry.
    const factory = getFactory();
    const NS = 'org.autoinsurancechain';
    const aseguradora = 'INSURANCEBUSINESS';

    var insured = factory.newResource(NS, 'Insured', InsuredRegister.idinsurance)
    insured.idInsured = InsuredRegister.idinsurance;
    insured.owner = factory.newRelationship(NS, 'Insurance', aseguradora);
    const insuredRegistry = await getAssetRegistry(NS + '.Insured');
    await insuredRegistry.add(insured)
}

```

La función vehicleregister, asociada a la transacción registro del vehículo sería:

```

/**
 *
 * @param {org.autoinsurancechain.VehicleRegister} VehicleRegister -
InsuredRegisterinstance
* @transaction
*/
async function vehicleregister(VehicleRegister){
    // Get the vehicle asset registry.
    const factory = getFactory();
    const NS = 'org.autoinsurancechain';

    var vehicle = factory.newResource(NS, 'Vehicle', VehicleRegister.idVehicle);
    vehicle.idVehicle = VehicleRegister.idVehicle;
    vehicle.color = VehicleRegister.color;
    vehicle.horsepower = VehicleRegister.horsepower;
    vehicle.model = VehicleRegister.model;
    vehicle.brand = VehicleRegister.brand;
    vehicle.year = VehicleRegister.year;
    vehicle.carlicense = VehicleRegister.carlicense;
    vehicle.insured = factory.newRelationship(NS, 'Insured',
VehicleRegister.insured.idInsurance);
    vehicle.insured = VehicleRegister.insured;
    const vehicleRegistry = await getAssetRegistry(NS + '.Vehicle');
    await vehicleRegistry.add(vehicle)
}

```



La función policyregister asociada a la transacción registro de la póliza de seguro sería:

```
/**
 *
 * @param {org.autoinsurancechain.PolicyRegister} PolicyRegister -
PolicyRegisterinstance
 * @transaction
 */
async function policyregister(PolicyRegister){
    // Get the vehicle asset registry.
    const factory = getFactory();
    const NS = 'org.autoinsurancechain';

    var insurancepolicy = factory.newResource(NS, 'InsurancePolicy',
PolicyRegister.idInsurancePolicy);

    insurancepolicy.idInsurancePolicy = PolicyRegister.idInsurancePolicy;
    insurancepolicy.statepolicy = 'StateInactive';
    insurancepolicy.insuranceVehicle = factory.newRelationship(NS, 'Vehicle',
PolicyRegister.insuranceVehicle.idVehicle);
    insurancepolicy.insured = factory.newRelationship(NS, 'Insured',
PolicyRegister.insured.idInsured);
    const policyRegistry = await getAssetRegistry(NS + '.InsurancePolicy');
    await policyRegistry.add(insurancepolicy)
}
}
```

La función signupdevicepolicy asociada a la transacción identificación de los dispositivos con la póliza será:

```
/**
 *
 * @param {org.autoinsurancechain.SignUpDevicePolicy} SignUpDevicePolicy -
SignUpDevicePolicyinstance
 * @transaction
 */
async function signupdevicepolicy(SignUpDevicePolicy){

    const factory = getFactory();
    const NS = 'org.autoinsurancechain';

    const assetRegistry1 = await getAssetRegistry(NS + '.Device');
    e1=await assetRegistry1.exists(SignUpDevicePolicy.idDevice);

    if(e1==false) { //
```



```
//registrar los 3 devices
console.log('registro de device')

const device = factory.newResource(NS, 'Device',
SignUpDevicePolicy.idDevice)
device.timestamp = Date.now().toString();

device.ids=SignUpDevicePolicy.ids

const deviceRegistry = await getAssetRegistry(NS + '.Device');
await deviceRegistry.add(device);

SignUpDevicePolicy.insurancepolicy.statepolicy = 'StateActive';
SignUpDevicePolicy.insurancepolicy.devicevehicle = device;

const br = await getAssetRegistry(NS + '.InsurancePolicy');
await br.update(SignUpDevicePolicy.insurancepolicy);

}
else {
  console.log('Existen elementos');
}
}
```

La función DeviceSendInfo, asociada a la transacción envío de información de los dispositivos será:

```
/**
 *
 * @param {org.autoinsurancechain.DeviceSendInfo} DeviceSendInfo - DeviceSendInfo
 * @transaction
 */
async function DeviceSendInfo(DeviceSendInfo){
  const factory = getFactory();
  const NS = 'org.autoinsurancechain';
  if (DeviceSendInfo.device.statedevice != 'StateActive'){
    throw new Error('Device does not work');
  }
  DeviceSendInfo.device.timestamp=DeviceSendInfo.Timestamp;
  DeviceSendInfo.device.linearacceleration=DeviceSendInfo.linearacceleration;
  DeviceSendInfo.device.lateralacceleration=DeviceSendInfo.lateralacceleration;
  DeviceSendInfo.device.roadtype=DeviceSendInfo.roadtype;
  const ds = await getAssetRegistry(NS+'.Device');
  await ds.update(DeviceSendInfo.device);
}
```

La función GarageSEndInfo, asociada a la transacción envío de la información por parte de los talleres será:



```
/**
 *
 * @param {org.autoinsurancechain.GarageSendInfo} GarageSendInfo - GarageSendInfo
 * @transaction
 */
async function GarageSendInfo(GarageSendInfo){
    const factory = getFactory();
    const NS = 'org.autoinsurancechain';

    GarageSendInfo.insurancepolicy.insuranceVehicle.statevehicle=GarageSendInfo.statevehicle;
    const ip = await getAssetRegistry(NS+'.Vehicle');
    await ip.update(GarageSendInfo.insurancepolicy.insuranceVehicle);
}
```

La función UpdateDevicePolicy, asociada a la transacción actualización de los dispositivos en la póliza será:

```
/**
 *
 * @param {org.autoinsurancechain.UpdateDevicePolicy} UpdateDevicePolicy - UpdateDevicePolicyinsance
 * @transaction
 */
async function UpdateDevicePolicy(UpdateDevicePolicy){
    // Get the vehicle asset registry.
    const factory = getFactory();
    const NS = 'org.autoinsurancechain';

    console.log("Actualizando dispositivo en poliza")
    UpdateDevicePolicy.insurancepolicy.devicevehicle.ids[0]=UpdateDevicePolicy.ids[0];
    UpdateDevicePolicy.insurancepolicy.devicevehicle.ids[1]=UpdateDevicePolicy.ids[1];
    UpdateDevicePolicy.insurancepolicy.devicevehicle.ids[2]=UpdateDevicePolicy.ids[2];

    const dsdevice = await getAssetRegistry(NS+'.Device');
    await dsdevice.update(UpdateDevicePolicy.insurancepolicy.devicevehicle)

    const ds = await getAssetRegistry(NS+'.InsurancePolicy');
    await ds.update(UpdateDevicePolicy.insurancepolicy);
}
```

La función AddGarage, asociada a la transacción añadir un taller será:

```
/**
 *
```



```

* @param {org.autoinsurancechain.AddGarage} AddGarage - AddGarageinstance
* @transaction
*/
async function AddGarage(AddGarage){
    const factory = getFactory();
    const NS = 'org.autoinsurancechain';
    const garage = factory.newResource(NS, 'Garage', AddGarage.idGarage);
    garage.idGarage = AddGarage.idGarage;
    garage.commercialname = AddGarage.commercialname;
    garage.city = AddGarage.city;
    garage.postcode = AddGarage.postcode;
    const garageRegistry = await getParticipantRegistry(NS + '.Garage');
    await garageRegistry.addAll([garage]);
}

```

---

## OPERATIVA DE DESPLIEGUE Y USO DE LA APLICACIÓN

Una vez desplegada nuestra infraestructura de red inicial para empezar a operar debemos seguir los siguientes pasos:

- 🌐 En primer lugar, la aseguradora, como organización principal debe ejecutar la transacción “setup” que a través de su función correspondiente crea todos los participantes iniciales:
  - Aseguradora: asignando nombre y número de identificación.
  - Regulador: asignando nombre y número de identificación
  - Taller: asignando los datos del taller correspondiente.

A partir de ahí, ya podemos operar, añadiendo asegurados, vehículos, pólizas y demás assets necesarios.

- 🌐 El siguiente paso es dar de alta un asegurado. La aseguradora realizará la transacción registro del asegurado que mediante su función InsuredRegistry, asigna un número identificativo único al asegurado. No se añaden más datos del asegurado en la transacción para no violar la ley de protección de datos. La aseguradora, en su propia base de datos, y con su política de protección de datos debe tener el resto de los datos del asegurado, pero no en la blockchain.
- 🌐 A continuación, también la aseguradora, realiza la transacción registro del vehículo que, a través de la función correspondiente, crea el asset vehículo con los datos correspondiente del vehículo, y le asigna un propietario, creado anteriormente como asset asegurado.
- 🌐 Después de tener asegurado y vehículo, la asegurado crea una póliza de seguros a través de la transacción correspondiente. Le asigna un número identificativo y se asocia al asegurado y su vehículo.
- 🌐 La aseguradora hace llegar al asegurado los tres dispositivos correspondientes para poder activar la póliza de seguros. El asegurado se encarga de colocarlos en el vehículo según las instrucciones de la aseguradora y mediante lectura QR de la etiqueta identificadora de los dispositivos, realizar una transacción de registro de los dispositivos asociados a su póliza de seguros, identificado por un número único a través de la nube, así la nube tiene identificados los dispositivos de esa póliza, a la vez que quedan registrados en la blockchain. En ese momento empieza la lectura de datos de los dispositivos y empieza realmente el “paga según conduces”. La póliza está perfeccionada.





- Una vez que el asegurado ya tiene los tres dispositivos colocados en el vehículo, y están registrados en la blockchain, la información que estos emiten se transmite a la nube, de forma que los datos de los dispositivos se puedan tratar automáticamente. Así, las aceleraciones que se reciben de los tres dispositivos se comparan y se comprobará que no difieren unos de otros en más de un 3% y se calculará la media. La ubicación recibida por los dispositivos, después de comprobar que sitúan al vehículo en la misma carretera, se identificará el tipo de carretera que es, y ese dato será el que se envíe a la blockchain junto con la media de la aceleración y el momento de su recogida.

El envío de esta información a la blockchain se realizará cada 10s.

En caso de que uno de los dispositivos nos dé un error, ya sea de aceleraciones o de posición, se notificará al usuario y se le enviará un dispositivo nuevo, teniendo este que volver a identificar los dispositivos (nuevo + antiguos) con la póliza de seguros.

Que un dispositivo se considere “averiado”, no implica que el seguro se dé por finalizado, simplemente, la nube rechaza los datos erróneos y sigue operando con los otros dos dispositivos hasta que el dispositivo deteriorado pueda ser sustituido.

- Los talleres por su parte, y de forma paralela, y cuando los asegurados consideren oportuno pasar por sus instalaciones, certificarán el estado de los vehículos y mediante la transacción envío de información de los talleres, registran en la blockchain el estado del vehículo.
- La aseguradora por su parte, tiene la posibilidad de añadir talleres nuevos a su red mediante la transacción correspondiente.

Para la ejecución de todas estas transacciones, la aseguradora, el asegurado y los talleres, cuentan con sus aplicaciones correspondientes que les permiten acceder solo a las transacciones a las que están autorizados a ejecutar.

---

## DESPLIEGUE DE LA RED DE NEGOCIO

Para el despliegue de nuestra red de negocio contamos con una máquina virtual donde tenemos previamente instaladas las siguientes herramientas:

-  Hyperledger Composer CLI
-  Hyperledger Composer REST Server
-  Hyperledger Composer Playground
-  Hyperledger Fabric Server
-  Yeoman generator

Para el despliegue de la solución en Hyperledger Composer, se siguen los siguientes pasos:

1. Se debe trabajar con un usuario de la aplicación que tenga configurados los mismos privilegios de root.
2. Se debe trabajar en el directorio home del usuario de la aplicación.
3. Para la creación de la estructura de negocio, ejecutar la siguiente sentencia en la terminal

```
yo hyperledger-composer:businessnetwork
```

Se ingresan los siguientes valores como datos de nuestra solución

```
autoinsurancechain
```

```
dapp autoinsurancechain
```

```
turingsdream
```





*admin@autoinsurance*  
 (pulsar intro)  
*org.autoinsurancechain*  
 (seleccionar NO: generatea p...)

4. Ingresar a la carpeta creada para la estructura de negocio de “autoinsurancechain”. Si se revisa el contenido de la carpeta, se observarán las siguientes carpetas y archivos.

*features lib models networkadmin.card package.json permissions.acl README.md test*

5. Se deberán editar los archivos, de acuerdo lo indicado en la sección Aplicación Descentralizada para Seguros, de forma que actualizamos con la solución propuesta los siguientes archivos:

*permissions.acl*  
*/models/org.autoinsurancechain.cto*  
*/lib/logic.js*

6. Se deberá genera el archivo .bna, para ello, en la ruta de la estructura de negocio de “autoinsurancechain”, se ejecutará la siguiente sentencia:

*composer archive create -t dir -n .*

7. Para el despliegue de la red de negocio se ejecuta la siguiente sentencia:

*composer network install --card PeerAdmin@hlfv1 --archiveFile autoinsurancechain@0.0.1.bna*

8. Para arrancar la red de negocio:

*composer network start --networkName autoinsurancechain --networkVersion 0.0.1 --networkAdmin admin --networkAdminEnrollSecret adminpw --card PeerAdmin@hlfv1 --file networkadmin.card*

9. Para importar a la red la tarjeta de negocio:

*composer card import --file networkadmin.card*

10. Se inicializa el composer-rest-server:

*composer-rest-server*

De esta manera tenemos desplegada nuestra red de negocio con todos los participantes, assets y transacciones necesarias para el funcionamiento de la red. Podemos empezar a operar según se ha descrito en apartados anteriores.

Adicionalmente se instalará el Blockchain Explorer para Hyperledger, lo que nos permitirá comprobar las transacciones, los bloques que se van formando, los peers que tenemos (que para la prueba de concepto y con Hyperledger Composer solo se despliega uno) . Se deben tener instaladas las siguientes herramientas:

- ◆ nodejs 6.9.x
- ◆ mysql 5.7 o superior



Se deberán ejecutar las siguientes sentencias:

1. En la ruta del usuario de la aplicación, se deberá copiar el archivo `blockchain-explorer.tar.gz`, que fue provisto en clases de hyperledger, para después ejecutar la siguiente sentencia:

```
tar -zxvf blockchain-explorer.tar.gz
```

2. Ingresar a la ruta `blockchain-explorer` creada y ejecutar la siguiente sentencia para crear la base de datos de `fabricexplorer`:

```
mysql -u root -p < db/fabricexplorer.sql
```

3. Se debe editar el archivo `config.json`, colocando las rutas adecuadas de los certificados de seguridad, el usuario y password del usuario que accederá a la base de datos.

```
{
  "network-config": {
    "org1": {
      "name": "peerOrg1",
      "mspid": "Org1MSP",
      "peer1": {
        "requests": "grpc://127.0.0.1:7051",
        "events": "grpc://127.0.0.1:7053",
        "server-hostname": "peer0.org1.example.com",
        "tls_cacerts": "/home/hyperledger/fabric-dev-servers/fabric-scripts/hlfv1/composer/crypto-config/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt"
      },
      "admin": {
        "key": "/home/hyperledger/fabric-dev-servers/fabric-scripts/hlfv1/composer/crypto-config/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp/keystore",
        "cert": "/home/hyperledger/fabric-dev-servers/fabric-scripts/hlfv1/composer/crypto-config/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/msp/signcerts/"
      }
    }
  },
  "host": "localhost",
  "port": "8447",
  "channel": "mychannel",
  "keyValueStore": "/tmp/fabric-client-kvs",
  "eventWaitTime": "30000",
  "mysql": {
    "host": "127.0.0.1",
    "port": "3306",
    "database": "fabricexplorer",
    "username": "hyperledger",
    "passwd": "123456"
  }
}
```

*Ilustración 8 Blockchain-explorer. Config.json*

4. Para levantar el blockchain explorer para hyperledger ejecutar las siguientes sentencias:

```
npm install
./start.sh
```

---

## VISUALIZACION DE LA RED DE NEGOCIO

Con el `composer-rest-server` (red desplegada) y el `blockchain-explorer` levantados, se abren las ventanas de navegación para cada herramienta respectivamente.

Para `composer-rest-server` se deberá colocar la siguiente url en el explorador:

[http://\[ipservidor\]:\[puertoconfigurado\]/explorer](http://[ipservidor]:[puertoconfigurado]/explorer).



En el puerto configurado se ha colocado el 8545 en nuestro caso. Seguidamente se visualiza una pantalla como la siguiente, que contiene la lista de transacciones definidas, así como los assets y participantes.

Hyperledger Composer REST server		
org_autoinsurancechain_AddGarage : A transaction named AddGarage	Show/Hide	List Operations Expand Operations
org_autoinsurancechain_Device : An asset named Device	Show/Hide	List Operations Expand Operations
org_autoinsurancechain_DeviceSendInfo : A transaction named DeviceSendInfo	Show/Hide	List Operations Expand Operations
org_autoinsurancechain_Garage : A participant named Garage	Show/Hide	List Operations Expand Operations
org_autoinsurancechain_GarageSendInfo : A transaction named GarageSendInfo	Show/Hide	List Operations Expand Operations
org_autoinsurancechain_Insurance : A participant named Insurance	Show/Hide	List Operations Expand Operations
org_autoinsurancechain_InsurancePolicy : An asset named InsurancePolicy	Show/Hide	List Operations Expand Operations
org_autoinsurancechain_Insured : An asset named Insured	Show/Hide	List Operations Expand Operations
org_autoinsurancechain_InsuredRegister : A transaction named InsuredRegister	Show/Hide	List Operations Expand Operations
org_autoinsurancechain_PolicyRegister : A transaction named PolicyRegister	Show/Hide	List Operations Expand Operations
org_autoinsurancechain_Regulator : A participant named Regulator	Show/Hide	List Operations Expand Operations
org_autoinsurancechain_setup : A transaction named setup	Show/Hide	List Operations Expand Operations
org_autoinsurancechain_SignUpDevicePolicy : A transaction named SignUpDevicePolicy	Show/Hide	List Operations Expand Operations
org_autoinsurancechain_UpdateDevicePolicy : A transaction named UpdateDevicePolicy	Show/Hide	List Operations Expand Operations
org_autoinsurancechain_Vehicle : An asset named Vehicle	Show/Hide	List Operations Expand Operations
org_autoinsurancechain_VehicleRegister : A transaction named VehicleRegister	Show/Hide	List Operations Expand Operations
System : General business network methods	Show/Hide	List Operations Expand Operations

*Ilustración 9 Hyperledger Composer REST server. Listado*



En la siguiente pantalla se pone como ejemplo el registro de un asegurado, mediante la transacción correspondiente:

Hyperledger Composer REST server

**org\_autoinsurancechain\_InsuredRegister : A transaction named InsuredRegister** Show/Hide | List Operations | Expand Operations

**GET** /org.autoinsurancechain.InsuredRegister Find all instances of the model matched by filter from the data source.

**POST** /org.autoinsurancechain.InsuredRegister Create a new instance of the model and persist it into the data source.

Response Class (Status 200)  
Request was successful

Model | Example Value

```
{
  "$class": "org.autoinsurancechain.InsuredRegister",
  "idinsurance": "string",
  "transactionId": "string",
  "timestamp": "2019-09-13T18:03:48.525Z"
}
```

Response Content Type: application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data	<pre>{   "\$class": "org.autoinsurancechain.InsuredRegister",   "idinsurance": "00002" }</pre>	Model instance data	body	Model   Example Value <pre>{   "\$class": "org.autoinsurancechain.InsuredRegister",   "idinsurance": "string",   "transactionId": "string",   "timestamp": "2019-09-13T18:03:48.528Z" }</pre>

Parameter content type: application/json

Ilustración 10 Hyperledger Composer REST server. Transacción InsuredRegister



Esta transacción es reflejada en el blockchain explorer para hyperledger, de forma que todo lo que se vaya ejecutando en nuestra red, puede comprobarse.

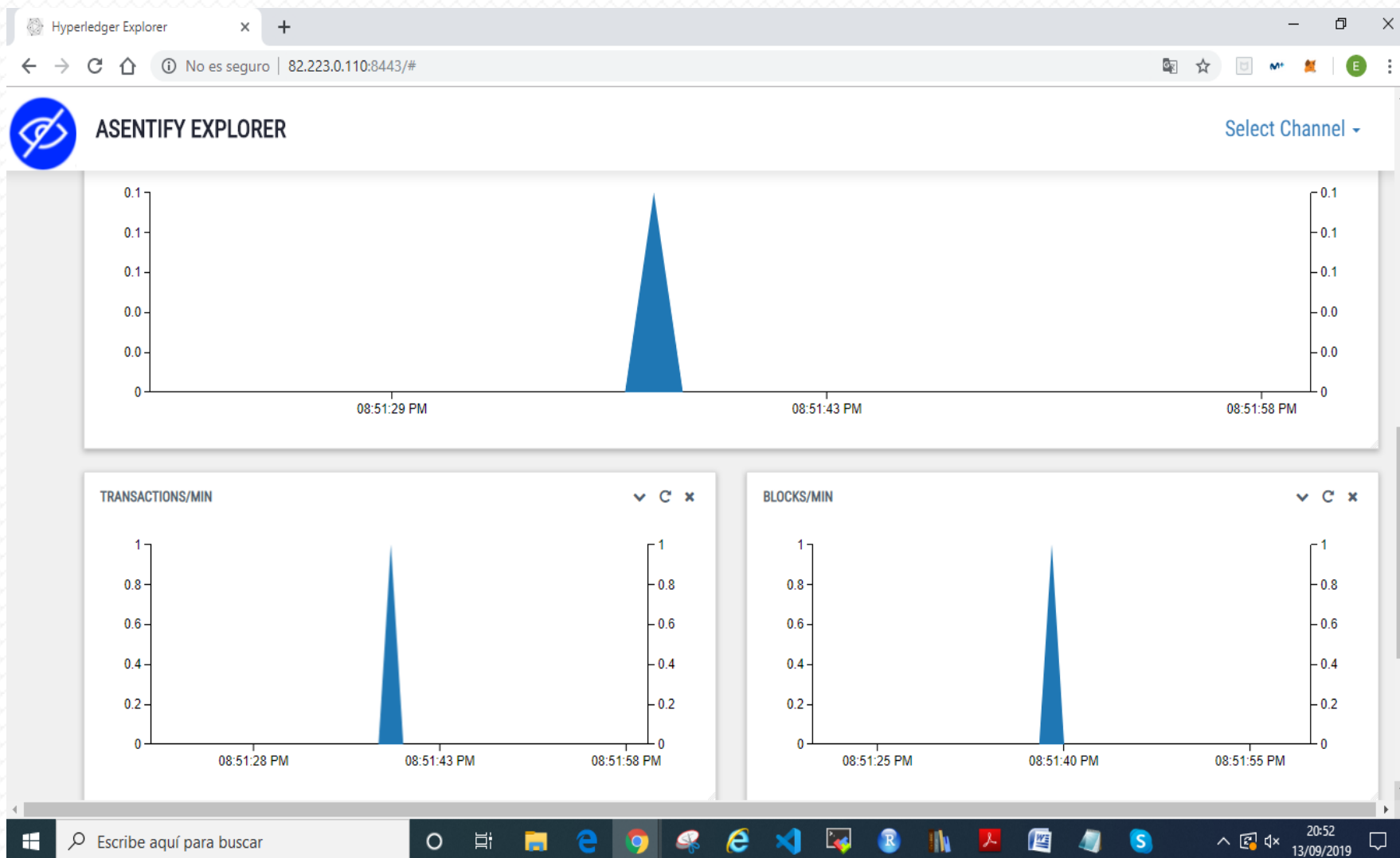


Ilustración 11 Hyperledger Explorer. Transacciones



Lo mismo se refleja para el registro de vehículos. En la siguiente pantalla se muestra el registro en el composer-rest-server

Hyperledger Composer REST server

**org\_autoinsurancechain\_VehicleRegister**: A transaction named VehicleRegister Show/Hide | List Operations | Expand Operations

**GET** /org.autoinsurancechain.VehicleRegister Find all instances of the model matched by filter from the data source.

**POST** /org.autoinsurancechain.VehicleRegister Create a new instance of the model and persist it into the data source.

**Response Class (Status 200)**  
Request was successful

Model | Example Value

```
{
  "idVehicle": "string",
  "color": "string",
  "horsepower": "string",
  "model": "string",
  "brand": "string",
  "year": "string",
  "carlicense": "string",
  "insured": {},
  "transactionId": "string",
  "timestamp": "2019-09-13T18:03:48.880Z"
}
```

Response Content Type:

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
data	<pre>{   "brand": "Renault",   "year": "2009",   "carlicense": "8661GPV",   "insured": {     "resource": "org.autoinsurancechain.Insured"   } }</pre>	Model instance data	body	Model   Example Value <pre>{   "\$class": "org.autoinsurancechain.VehicleRegister",   "idVehicle": "string",   "color": "string",   "horsepower": "string",   "model": "string",   "brand": "string",   "year": "string",   "carlicense": "string",   "insured": {},   "transactionId": "string",   "timestamp": "2019-09-13T18:03:48.880Z" }</pre>

Parameter content type:

[Try it out!](#) [Hide Response](#)

**Curl**

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{ \
  "$class": "org.autoinsurancechain.VehicleRegister", \
  "idVehicle": "string", \
  "color": "string", \
  "horsepower": "string", \
  "model": "string", \
  "brand": "string", \
  "year": "string", \
  "carlicense": "string", \
  "insured": {}, \
  "transactionId": "string", \
  "timestamp": "2019-09-13T18:03:48.880Z" \
}
```

Ilustración 12 Hyperledger Composer REST server. Transacción VehicleRegister



Y la transacción reflejada en el blockchain explorer para hyperledger:

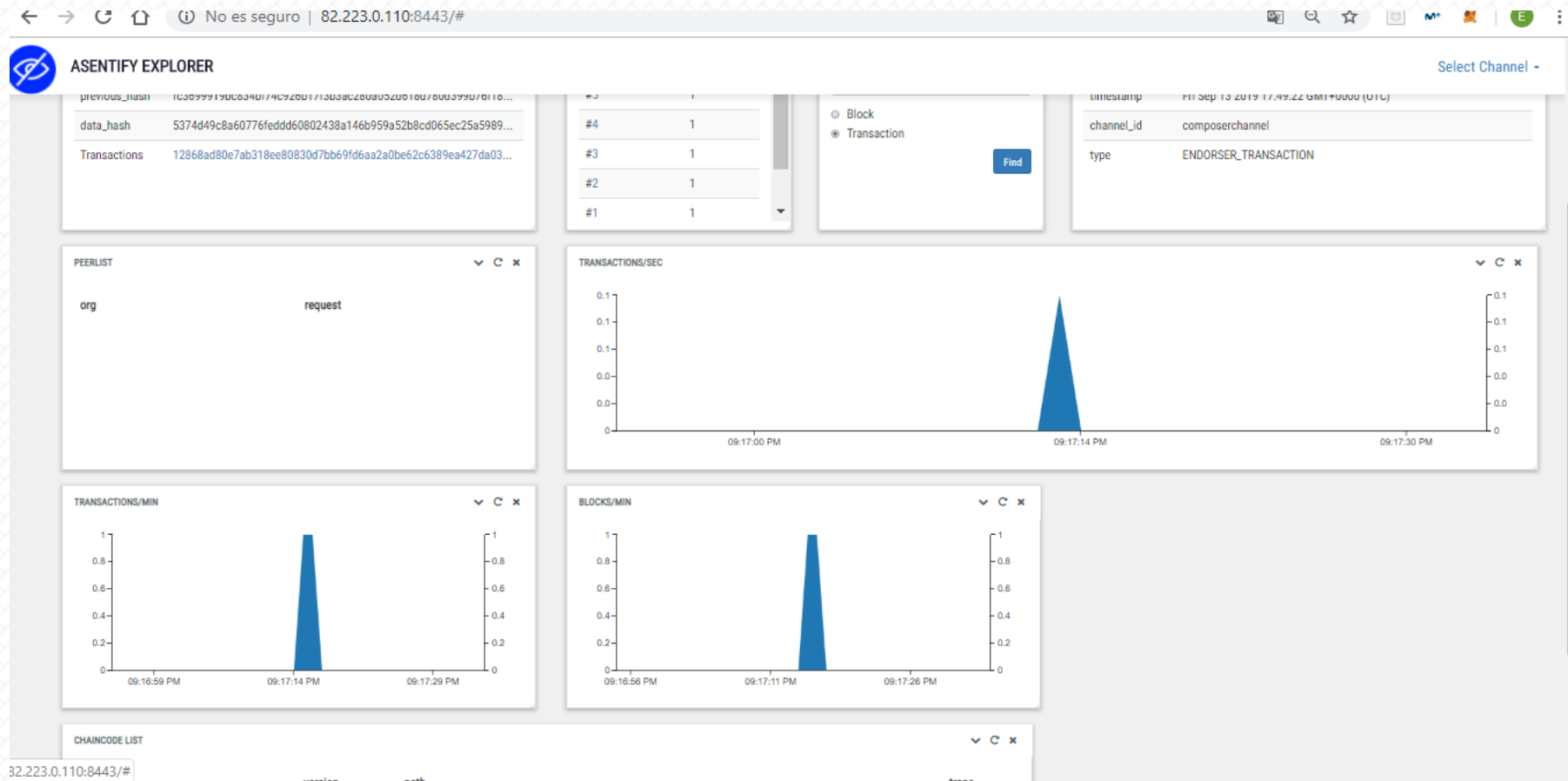


Ilustración 13 Hyperledger Explorer. Transacción





## TECHNICAL ROADMAP

Desarrollamos el siguiente roadmap técnico que explica las diferentes fases por las que pasará el proyecto en su evolución hasta alcanzar su estado pleno.

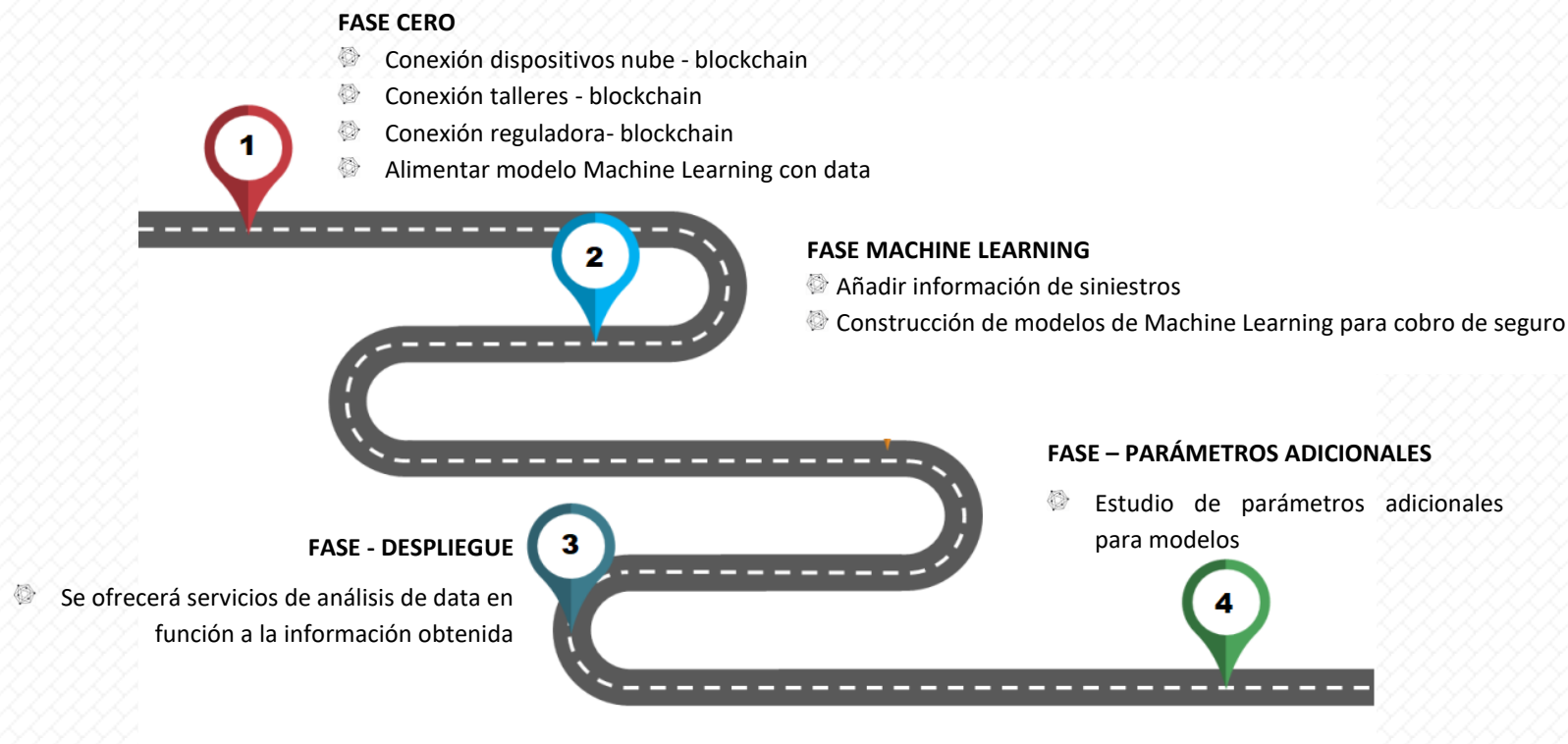


Ilustración 14. RoadMap



---

## FASE CERO

Esta primera etapa será una prueba piloto en el que se evaluará la viabilidad del proyecto y se obtendrá una primera aproximación de la conducción real de los usuarios. Se contemplará una cantidad de vehículos limitada concertada con la aseguradora, una serie de talleres delimitados y una organización reguladora. Obtendremos una cantidad considerable de datos de la forma real de conducción de los asegurados y de los siniestros que se producen.

Las funcionalidades en esta etapa cero serán:

### *Conexión aseguradora – Nube – Blockchain*

En primer lugar, debemos tener una interfaz operativa para la aseguradora, de forma que podamos desplegar la red y la aseguradora pueda operar, transaccionando asegurados, vehículos y operar en la nube configurando la transformación de datos necesaria para la grabación de dichos datos en la blockchain.

### *Conexión dispositivos – Nube – Blockchain*

También se tendrá en cuenta la conexión entre los dispositivos de los vehículos, la nube y la blockchain, incluyendo la interfaz móvil necesaria para que el asegurado pueda conectar los dispositivos del coche con la nube y la blockchain, además de la interfaz necesaria para que la aseguradora pueda empezar a operar.

### *Conexión taller – Blockchain*

Se tendrá en cuenta la conexión de los talleres con la red blockchain, incluyendo la interfaz que permitirá enviar la información de los talleres a la blockchain a través de la transacción correspondiente donde se evalúa el estado del vehículo asegurado.

### *Recolección de datos de conducción*

Se contempla la conexión de la blockchain con la infraestructura de big data para almacenar toda la información recogida en la blockchain.

En toda esta etapa, el precio de la prima del seguro para estos asegurados en la prueba piloto será un precio fijo determinado por la aseguradora, ya que no contamos con datos suficientes para poder poner precio según los datos de conducción, es decir, no podemos determinar si las aceleraciones de unos u otros valores, son más o menos decisivos a la hora de producir siniestros. El precio se ajustará una vez elaborados los modelos de Machine Learning en la próxima etapa.

---

## FASE MACHINE LEARNING

En esta etapa conseguiremos llegar mediante modelos Machine Learning a la obtención de resultados significativos en los que podamos evaluar la incidencia de los datos recogidos con el número de siniestros que acontecen.

Las funcionalidades en esta fase Machine Learning serán:



### *Información de Siniestros*

Se recopilará toda la información de siniestros de los asegurados para incluirla en la blockchain mediante su asset correspondiente de forma que estarán incluidos en los datos obtenidos de la blockchain mediante una actualización del BNA. De esta forma, dentro de los datos obtenidos desde la blockchain tenemos tanto las variables de conducción como los resultados.

### *Construcción de modelos de Machine Learning*

En este punto se cuenta con suficiente información para poder generar modelos de Machine Learning que permitan generar cluster de cliente en base a sus hábitos de conducción que nos deje ajustar el precio de la prima del seguro con los resultados obtenidos, de forma que los asegurados puedan disfrutar del precio justo por su conducción y la aseguradora, aumentar sus beneficios obteniendo ventaja competitiva.

---

#### FASE DESPLIEGUE

Una vez obtenidos los modelos de Machine Learning con los datos de la prueba piloto con un número limitado de asegurados, la aseguradora tendrá un modelo final con precios ajustados a la conducción real de los conductores y con un riesgo medido, de forma que se podrá realizar el despliegue a nivel comercial de la DApp para el mercado completo.

---

#### FASE DE ESTUDIO DE PARÁMETROS ADICIONALES

Cuando la infraestructura esté a pleno rendimiento, se contemplarán nuevas fuentes de información relevante para la evaluación del modo de conducir de los asegurados. De esta forma se estudiará si la adicción de nuevas variables a los modelos de Machine Learning, mejora la eficacia de éstos, lo que redundaría en un mejor ajuste de las primas y por tanto de una mejora competitiva para la aseguradora.

Se evaluará entre otros, la forma de medir, si los conductores mantienen o no las distancias de seguridad con los demás vehículos, y si es relevante a la hora de contabilizar el número de siniestros registrados.

Adicionalmente, y por la infraestructura que se contempla de BigData, se explotarán datos generales y gratuitos obtenidos de otras fuentes como AEMET, comunidades, etc. que se obtienen en tiempo real.



## FINANCIACIÓN



Para la implementación de nuestro proyecto empresarial vamos a valernos de una vía de financiación novedosa y poco conocida en España que crecerá en los próximos años: los business angels.

Un business angel es una persona o grupo de personas que invierten una parte de su patrimonio en el capital una empresa start-up con gran capacidad de crecimiento, normalmente a cambio de participar en el accionariado de esta. Además de aportar capital a la empresa, los business angels o inversores privados también aportan capital inteligente, es decir, su experiencia empresarial, sus conocimientos en el sector, confianza, y lo más importante, su red de contactos empresariales y personales.

El método de selección de empresas que realizan los business angels se basa en la valoración del plan de negocio que presentan los emprendedores, y analizando las tasas de retorno sobre la inversión (ROI), de al menos 10 o más veces la inversión inicial en el período de 5 años. Esta selección de proyectos empresariales donde invertir se debe hacer de un modo riguroso, ya que no debemos olvidar que este tipo de inversiones tiene un riesgo extremadamente alto, y un alto porcentaje de ellas se pierde completamente cuando las empresas fracasan.

El volumen medio de las inversiones que realizan ronda entre los 25.000€ y los 250.000 € por operación, siendo inferior al 25% del capital líquido del inversor.

Asimismo, el arranque y el mantenimiento del proyecto hasta que consigamos una cartera de clientes tendrá unos costes de:

Gastos de marketing que estimaremos en 25.000 euros.

Servidores cloud para albergar nuestra blockchain de prueba (AWS / IBM Cloud) de alrededor de 1355,40 \$ al año según el documento anexo.

Suscripción a IBM Blockchain Platform (aproximadamente 230€ - 400€ mes) para realizar pruebas de concepto.

Adicionalmente y con un propósito mayoritariamente de marketing realizamos una campaña de crowdfunding en Kickstarter. De manera que conseguiremos atraer la atención de los potenciales consumidores finales, financiar parte de los gastos emitiendo “vales descuentos” que pueden canjear con las aseguradoras una vez que se lance con éxito el proyecto. Esta idea es atractiva a nivel de marketing ya que identificamos los early-adopters y creamos una base de potenciales clientes finales para nuestro cliente (la aseguradora).



## EQUIPO FUNDADOR

- 🌐 Min Yuan Chen. CFO. Directora administrativa y financiera.
- 🌐 Nicolás Quintero. CTO. Analista desarrollador.
- 🌐 Virginia Saco. CIO. Directora de Información.
- 🌐 Emma Gutiérrez. CDO. Analista de Datos.
- 🌐 Claudia Jara. COO. Directora operativa.
- 🌐 Loreto Heres. CMO. Coordinadora de control interno e Ingeniera de marketing.



## PLAN MARKETING

En primer lugar, hay que tener en cuenta que nuestro producto es *AUTOINSURANCECHAIN*, aplicación descentralizada en blockchain para seguros de vehículos basados en la conducción real de los asegurados. Teniendo en cuenta las características peculiares de nuestro producto, el plan de marketing va dirigido a las aseguradoras de vehículos existentes que puedan estar interesadas en la implementación de esta tecnología como uno de sus productos estrella.

El plan de Marketing que vamos a aplicar a este proyecto se orienta sobre todo a presentar el modelo de negocio a dichas aseguradoras, ya que son éstas nuestros potenciales clientes. La estrategia de popularidad será por un lado crear una página web que las aseguradoras puedan consultar, y por otro dar a conocer el producto que se ofrece de forma directa a estas empresas que puedan estar interesadas.

Partiremos de la creación de una página web segura, de aspecto profesional, donde el Whitepaper del proyecto tenga un lugar visible y fácilmente accesible, que siga estándares SEO y que sea un punto de información central donde poder ver las características del producto, así como la evolución del proyecto, mejoras y actualizaciones.

Una parte muy importante del plan será la organización de eventos y presencia en ferias especializadas en el sector del seguro. Se contactará con la UNESPA, asociación empresarial del seguro en España, para darse a conocer y tener la máxima presencia en el sector. Paralelamente se contactará con el mayor número de aseguradoras posibles para poder presentarles el producto y concertar reuniones o entrevistas con los departamentos correspondientes

Así pues, vamos a repartir nuestro presupuesto de 25.000 euros en 3 puntos principales:

### PÁGINA WEB

Externamente se subcontratará:

DEFINICIÓN	CANTIDAD
Diseño de página web con administrador de contenidos para su actualización constante con una apariencia profesional, con el Whitepaper del proyecto en un lugar visible y fácilmente accesible.	1.000
Inserción de un idioma extranjero (inglés).	500
Sección de noticias/blog actualizable por el administrador de contenidos.	500
Instalación y configuración de componentes para la gestión de eventos. Calendario.	50
Creación de galería fotográfica, incluidos videos explicativos.	500



Formularios personalizados para solicitar información.	100
Diseño del logo en digital para la página web y útil para el papel.	150
Servidor profesional 2GB + database + servicio mail.	200
Subcontratación de SEO de la página web creada destinado especialmente a potenciar las palabras Blockchain, Seguros y Coches.	2.100

#### ASISTENCIA A FERIAS

Por otro lado, se potenciará la asistencia a eventos especializados de la industria. En este sentido:

DEFINICIÓN	CANTIDAD
Se instalará un Stand para presentación, en las ferias del sector, de la plataforma bien calendarizados, con presencia en la página web del proyecto.	6.000

#### VISITAS PRESENCIALES A LOS PRINCIPALES CLIETES POTENCIALES

Existen alrededor de 40 empresas aseguradoras que pueden funcionar como posibles clientes. Suponiendo que les interese y acepten una reunión la mitad de ellos, esto conllevaría unos costes asociados de:

DEFINICIÓN	CANTIDAD
Transporte	10.000
Dietas	1.600
Móvil	300
Material informático	2.000





## ANEXO I. COSTES SERVIDOR

## Microsoft Azure Estimate

Su presupuesto

Service type	Custom name	Region	Description	Estimated Cost
App Service		West Europe	Nivel Premium V2; 1 P1V2 (1 núcleos, 3.5 GB de RAM, 250 GB de almacenamiento) x 730 Hours; SO Linux	\$83,95
Support			Support	\$29,00
			Licensing Program	Microsoft Online Services Agreement
			Monthly Total	\$112,95
			Annual Total	\$1.355,40

## Disclaimer

All prices shown are in US Dollar (\$). This is a summary estimate, not a quote. For up to date pricing information please visit <https://azure.microsoft.com/pricing/calculator/>  
 This estimate was created at 9/10/2019 4:35:18 PM UTC.



## ANEXO II. ARCHIVOS DESPLIEGUE RED

Anexamos los archivos:

Logic.js

Org.autoinsurancechain.cto

Permissions.acl

Query.qry

## BIBLIOGRAFÍA

- ◆ <https://www.rastreator.com/seguros-de-coche/articulos-destacados/seguros-de-coche-pay-as-you-drive.aspx>
- ◆ <https://www.mapfre.es/portal/movilidad/careward/app/bases-ubi.html>
- ◆ <https://www.xataka.com/automovil/algun-dia-todos-los-seguros-de-coches-seran-asi-un-maridaje-entre-la-tecnologia-y-un-gran-hermano>
- ◆ <http://www.unespa.es/que-hacemos/publicaciones/>
- ◆ <https://www.turboseguros.com/blog/seguros/cuando-efecto-contrato-seguro/>
- ◆ <https://aprendeblockchain.wordpress.com/hyperledger/>
- ◆ [https://hyperledger-fabric.readthedocs.io/en/latest/build\\_network.html](https://hyperledger-fabric.readthedocs.io/en/latest/build_network.html)
- ◆ <https://hyperledger.github.io/composer/latest/installing/installing-index.html>
- ◆ <https://www.ibm.com/developerworks/ssa/library/cl-blockchain-hyperledger-fabric-hyperledger-composer-compared/index.html>
- ◆ <https://cloud.ibm.com/docs/services/blockchain/howto?topic=blockchain-ibp-pricing&locale=en>
- ◆ <https://cloud.ibm.com/docs/services/blockchain?topic=blockchain-ibp-saas-pricing&locale=en>
- ◆ <https://databricks.com/glossary/what-is-spark-streaming>
- ◆ <https://kafka.apache.org/>