Escuela Superior
de Ingeniería y Tecnología
Universidad de La Laguna

Trabajo de Fin de Grado

Grado en Ingeniería Informática

Seguro Paramétrico SPC-19 con Blockchain

Parametric Insurance SPC-19 with Blockchain

Eduardo Suárez Ojeda

La Laguna, *10* de *junio* de 2021

D. **Julio Antonio Brito Santana**, con N.I.F. 42.812.143-Q profesor Titular de Universidad adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutor

D. **Benito Cuesta Viera**, con N.I.F. 43.818.241-K adscrito a la Escuela Superior de Ingeniería y Tecnología de la Universidad de La Laguna, como cotutor

C E R T I F I C A (N)

Que la presente memoria titulada:

“Seguro Paramétrico SPC-19 con Blockchain”

ha sido realizada bajo su dirección por D. **Eduardo Suárez Ojeda**,

con N.I.F. 43.379.934-W.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 10 de junio de 2021.

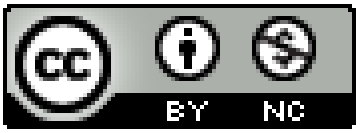
Agradecimientos

En especial agradecer a Benito Cuesta y Carlos Domínguez, con los que he formado equipo para la programación de la aplicación y me han ayudado mucho en mi primer proyecto real.

También a Julio Antonio Brito por la tutorización del proyecto y a José Luis Roda por apoyar como director de la cátedra.

Por último agradecer a la Mutua Tinerfeña por interesarse en el proyecto y ayudar a crear un caso de uso realista gracias a su experiencia.

Licencia



Resumen

El coronavirus COVID-19 ha impactado de lleno en la economía y en las personas. Nos ha puesto en alerta, buscando soluciones en muchos ámbitos de protección y seguridad. La entrada de turistas en destinos como Canarias, el control de pasajeros, la verificación de estar libre de la enfermedad, los seguros asociados a los viajes y las pruebas PCR son algunos retos para los que se buscan soluciones.

La tecnología Blockchain es una referencia para la creación de modelos de negocio y servicios transparentes y eficientes que puede servir de marco de solución a algunos de estos retos. Este trabajo desarrolla una prueba de concepto basada en tecnología Blockchain que permite la gestión de las condiciones para la indemnización de un nuevo seguro paramétrico. Este seguro indemniza al asegurado con un importe a determinar por la aseguradora, en el caso de que contraiga el virus SARS-COV-2 durante la estancia en el destino turístico.

El empleo de esta tecnología permite el registro inmutable de los datos del cliente y de la póliza contratada, garantizando la privacidad de la información registrada. Además, el registro fehaciente tanto de las solicitudes de pruebas PCR como de sus resultados, asegura la autenticidad, integridad y confidencialidad de la información. La validación de las condiciones necesarias para la indemnización a los asegurados se realizará automáticamente mediante smart contracts. Los contratos inteligentes garantizan a la aseguradora que terceros no han alterado ni los datos ni la lógica empresarial utilizada.Y por último aumenta la confiabilidad de los clientes de la aseguradora, así como la calidad y comodidad del servicio, poniendo a su disposición un sistema completamente transparente y verificable para la gestión del riesgo asegurado.

PALABRAS CLAVE: **COVID-19, Seguros, Identidad Digital, Blockchain, Contrato Inteligente**

Abstract

The COVID-19 coronavirus has fully impacted the economy and people. It has put us on alert, looking for solutions in many areas of protection and security. The entry of tourists in destinations such as the Canary Islands, the control of passengers, the verification of being free of the disease, the insurance associated with trips and PCR tests are some challenges for which solutions are sought. Blockchain technology is a reference for creating transparent and efficient business models and services that can serve as a solution framework for some of these challenges.

This project develops a proof of concept based on Blockchain technology that allows the management of the conditions for the indemnification of new parametric insurance. This insurance compensates the insured with an amount to be determined by the insurer in contracting the COV-19 during their stay at the tourist destination.

The use of these technologies allows the immutable registration of the client's data and the contracted policy, guaranteeing the privacy of the registered information. In addition, the reliable record of both PCR test requests and their results ensures the authenticity, integrity and confidentiality of the data. The validation of the necessary conditions for compensation to the insured will be carried out automatically through smart contracts. The smart contracts guarantee the insurer that third parties have not altered either the data or the business logic used. And finally, it increases the reliability of the insurer’s clients and the quality and comfort of the service, making available a completely transparent and verifiable system for managing the insured risk.

Keywords**: Insurance, COVID-19, Blockchain, Smart Contract, Digital Identity**

Índice general

Índice de figuras

Capítulo 1 Introducción

El turismo es un sector económico que ha crecido ampliamente en los últimos años. Existen países como España, cuya economía está basada principalmente en el turismo. Factores como su clima, cultura, gastronomía… hacen que destaque como destino ideal para las vacaciones. Canarias se encuentra entre las regiones españolas donde su economía es más dependiente del desarrollo turístico.

Debido al coronavirus o COVID-19 la economía global ha sufrido una crisis muy importante. Entre los sectores más afectados se encuentra el turismo, ya que en un intento por controlar la pandemia se ha restringido la movilidad. Los gobiernos han optado por medidas de control dentro de los propios países como el toque de queda, en casos extremos, se ha llegado a un confinamiento domiciliario para evitar contagios y se han cerrado las fronteras.

Países como Alemania o Reino Unido, desde los cuales llegan la mayor cantidad de turistas a nuestras islas, o bien no permiten directamente el viaje, o establecen una cuarentena (generalmente de dos semanas) que deben cumplir los viajeros a la vuelta a su país.

Actualmente se atisba en los países más desarrollados el fin de las medidas extraordinarias que se han tomado para controlar la pandemia. El factor de mayor optimismo actualmente es la alta tasa de vacunación en estos países, lograda por la capacidad de producción de vacunas junto con los dispositivos sanitarios puestos en marcha para la vacunación. Ejemplo de ello es el Gobierno de España, que ha previsto que al final del verano el 70% de la población está vacunada.

La actual situación nos permite ser optimistas en cuanto a la posibilidad de recuperar la conocida como “antigua normalidad”, pero a partir de ahora tendremos que convivir con el virus, ya que no será erradicado totalmente, aunque la inmunidad obtenida por las vacunas permitirá que baje drásticamente su mortalidad y los casos graves sean muy escasos.

Todo ello hace prever una recuperación del sector turístico y de la economía a finales de año o inicios de 2022, pero siempre teniendo en cuenta la nueva situación.

Nuestro proyecto trata de paliar esta crisis y relanzar el sector turístico, aprovechando lo máximo posible la gran cantidad de infraestructuras que poseen las Islas Canarias destinadas a él.

La naturaleza altamente infecciosa del COVID-19 hace de la detección temprana de los casos una necesidad de primer nivel. La tecnología blockchain puede ser muy útil para el manejo de las epidemias [1]. Puede proveer un marco robusto y transparente para la toma de decisiones, aumentando su velocidad y precisión. Es ideal tanto para el seguimiento de los casos como incluso para realizar seguros enfocados en dicha enfermedad.

Las aplicaciones blockchain pueden monitorizar los casos a lo largo del tiempo al establecer un registro seguro e inmutable. Incluso pueden llegar a mejorar la precisión del diagnóstico y tratamiento efectivo de la enfermedad mediante la información que guarda. Las autoridades sanitarias podrían usar sistemas blockchain permisionados para tratar el problema de la interoperabilidad de los datos sanitarios, ya que esta tecnología es especialmente útil para tratar la privacidad del paciente.

Otro caso interesante sería el de rastreo de casos. Mediante una aplicación sencilla llamada IReport-Covid [2] en Singapur, las personas tanto sintomáticas como asintomáticas pueden emitir datos útiles acerca del virus mediante la realización de una simple encuesta anónima.

En este proyecto aprovechamos la tecnología Blockchain para desarrollar una prueba de concepto que se encargue de la gestión de un nuevo seguro paramétrico. Este seguro está diseñado para indemnizar al asegurado en caso de que contraiga el virus SARS-COV-2 durante la estancia en el destino turístico. Dicha indemnización cubrirá los gastos de estancia requeridos debido a la cuarentena obligatoria que deberá cubrir el paciente, de entre 10 y 14 días, en el propio hotel.

La prueba de concepto realizada,sentará las bases de la aplicación de la tecnología Blockchain en el sector de los seguros. Blockchain es una herramienta que tiene mucho futuro en este ámbito, debido a su transparencia, integridad, y confidencialidad . A ambas partes, usuario y aseguradora supone una gran comodidad y ahorros, ya que la propia tecnología es la que actúa como intermediario. Derivado de estas características destaca su capacidad para detectar fraudes [3].

Para la realización de este proyecto han colaborado la Universidad de La Laguna, la Cátedra Cajasiete de Big Data, Open Data y Blockchain y la Mutua Tinerfeña, como empresa interesada en esta clase de proyectos orientados a seguros. Fruto de dicha colaboración es este Trabajo de Fin de Grado, al que corresponde este documento.

La iniciativa se toma desde la Universidad, con el apoyo y colaboración de las entidades descritas anteriormente se propone el proyecto y el TFG. Para el desarrollo del proyecto se constituye un equipo de trabajo con participantes de la propia Universidad (Julio Brito como Tutor, Benito Cuesta como cotutor y jefe de proyecto y Eduardo Suárez como alumno realizando su TFG) y desde de la Cátedra Cajasiete (José Luis Roda como director de la cátedra y Carlo Domínguez como ingeniero de la Cátedra). Este equipo se ha encargado de desarrollar el prototipo de seguro basado en blockchain. Los miembros de la Mutua Tinerfeña participantes en el proyecto han proporcionado unas bases para generar un caso de uso lo más realista posible, gracias a su experiencia en el sector.

A continuación, el contenido de este documento se estructura en los siguiente capítulos. Un capítulo segundo introduce los fundamentos y característas de la tecnología principal del proyecto (blockchain). Seguidamente en el capítulo tercero se estudia el estado del arte de su uso en el campo de los seguros de todo tipo pero especialmente centrado en los seguros de salud. En dicho estudio se expone las características que la hacen encajar en el sector y el impacto de cara a su futuro, basándonos en la bibliografía existente. Una vez expuesto el marco conceptual en el capítulo cuarto, se describe la aplicación desarrollada y el caso de uso para el que ha sido diseñada e implementada. Para finalizar se detallan las conclusiones y los trabajos futuros del proyecto, añadiendo un presupuesto estimado del mismo.

Capítulo 2 Tecnología Blockchain

2.1 Características generales

Como su nombre indica se trata de una estructura de datos cuya información se agrupa formando una cadena de bloques. Cada eslabón de la cadena posee metainformación sobre el bloque que le precede en forma de hash, cuyo código se genera aplicando una función hash criptográfica al contenido del bloque. De esta manera se establece el orden total de la cadena.

Estos hashes tienen la peculiaridad de que el mínimo cambio en el contenido del bloque tiene un efecto sobre el hash generado, lo cambia completamente. Por medio de estas codificaciones podemos estar seguro que la información de cada bloque permanece segura sin riesgo de modificarse, ya que entonces el hash cambiará y no coincidirá con el contenido del hash previo del siguiente bloque y la cadena quedará invalidada.

Los usuarios interactúan con la blockchain por medio de las cuentas registradas en ellas. Dichas cuentas están protegidas mediante criptografía para establecer la propiedad mediante un par de claves pública y privada. Estas claves se generan a partir de un número aleatorio muy grande al que se le aplica una función con la que se obtienen las claves. La dirección de la cuenta en la blockchain se extrae de la propia clave pública. Este tipo de criptografía es especialmente útil para manejar la identidad de forma segura y confiable.

El atributo más representativo de la blockchain es su almacenamiento distribuido. La cadena se replica totalmente en cada nodo de la red. Los nodos son los puntos donde se replica la cadena. Estos nodos son responsables de comprobar si los bloques que les comparten otros nodos son válidos, guardar los bloques de transacciones realizadas en la blockchain y compartir dichos bloques con los nodos de su misma blockchain, con los que se pueda comunicar directamente. El proceso de incorporación de nuevos bloques se denomina “minería” y lo realizan los “mineros”, los cuales obtienen un beneficio en las blockchains públicas debido al coste que supone conseguir un bloque válido.

La manera de obtener un bloque válido varía según el algoritmo de consenso usado en dicha blockchain. El más famoso y que se ha usado generalmente en las principales redes públicas es el algoritmo de prueba de trabajo (Proof of Work, “PoW”). Este algoritmo se basa en la capacidad computacional, ya que para obtener un bloque válido se requiere

realizar un cálculo complejo para obtener un hash determinado. Generalmente este tipo de hash depende de la “dificultad”, una variable de la red que indica cómo debe ser un hash válido. En concreto, esta variable indica el número de ceros a la izquierda consecutivos desde el final del hash (en hexadecimal). Es una variable debido a que se va regulando automáticamente según el poder computacional que tenga la red, para mantener un tiempo de minado estable. Es decir, que a mayor poder computacional tenga la red de mineros, mayor incremento sufrirá la dificultad. Para modificar el hash generado sin modificar el conjunto de transacciones que contiene el bloque existe una parte de dicho bloque que se denomina “nonce”. Básicamente es un número que se va cambiando hasta llegar a un hash que supere la prueba de dificultad.

Este es uno de los puntos más controvertidos de la blockchain, ya que el primer minero que consigue un bloque válido en una red pública obtiene una compensación económica en la criptomoneda de dicha red. Dicha compensación varía entre las diferentes redes, por ejemplo, en Bitcoin cada 4 años desde su creación se divide a la mitad la recompensa genérica al minar un bloque, mientras que en Ethereum se mantiene constante. La compensación no sólo comprende la recompensa genérica, sino que también se suman las tarifas asociadas a cada transacción. Dichas tarifas las establecen los usuarios que realizan las transacciones para asegurarse que los mineros añaden sus transacciones en los bloques que minan. Por lo tanto, cuanto mayor sea la tarifa de la transacción más posibilidades habrá de que a los mineros les interese añadirlas en el nuevo bloque y la transacción generalmente se llevará a cabo más rápido.

La controversia viene debido a que cuanto mayor premio económico se obtenga minando mayor número de mineros tendrá la red, lo que a la larga conlleva que aumente la dificultad de la red, por lo que hará falta un mayor poder computacional para minar un bloque. Como el primer bloque válido es el que se añade a la cadena, el esfuerzo computacional que han realizado el resto de mineros no ha valido de nada y deben volver a empezar desde cero, debido a que al añadir un nuevo bloque se modifica el hash del bloque anterior. Toda esta cadena de acontecimientos conlleva que una alta cantidad de energía se “desperdicie”. En éste factor se está trabajando para subsanarlo.

Cambios en este sentido, están ocurriendo, por ejemplo en la red Ethereum (una de las blockchains públicas más famosas) se está pasando del algoritmo de consenso PoW a un algoritmo PoS (Power of Stake o prueba de participación) que consiste en que los propios validadores de bloques sean usuarios de dicha red que posean una alta cantidad de la criptodivisa de la misma. La idea es que los que más tienen que perder si se corrompe la red sean los encargados de validar las transacciones. Con ello se ahorrará una gran cantidad de energía que se usaba anteriormente para superar el algoritmo PoW, pero a cambio puede suponer a la larga que el poder se establezca en unos pocos, debido que como el que valida las transacciones se escoge porque tiene un alto porcentaje de la divisa de la red, al validarlas obtiene un beneficio en forma de más criptomonedas.

Este tipo de algoritmos tienen como principal objetivo proteger la red de intentos maliciosos de falsear información de la misma. Actualmente las redes PoW se dice que son seguras cuando no hay ninguna entidad que posea el 51% o más del poder de computación de la misma. Así , el que posea dicha cantidad podrá elegir los bloques que se añadan a la cadena y por lo tanto modificarla a su gusto. Como anteriormente se ha descrito el primer bloque que se mina es el que se añade a la cadena y se puede dar un escenario en el que se minen dos bloques casi al mismo tiempo sin que ninguno tenga consciencia del otro. Esto puede suceder cuando la latencia de comunicación entre los nodos es alta. La solución a este conflicto es que la cadena más larga es la que prevalece, es decir, que se sigue minando hasta que una de las dos cadenas consigue minar un bloque más que la anterior y da tiempo a al comunicarse con los demás nodos para que la den como válida y la otra cadena quede invalidada. Aunque hay muy pocas posibilidades de que se dé el caso, no es un suceso imposible y conlleva que algunas transacciones que un usuario ha dado por hechas y el bloque añadido pueden acabar siendo invalidados si se da esta situación Por ello se suele esperar un tiempo prudencial una vez minada la transacción para confirmar que no será invalidada. También comentar que los mineros que minan este tipo de bloques que se acaban descartando, llamados “bloques tío” (uncle blocks) también se llevan un premio económico.

Otro concepto importante de la blockchain es el de bifurcación, o en inglés “fork”, que se da cuando se produce un ajuste de las reglas de consenso en una red blockchain.

Existen dos tipos, duras y blandas. Las duras requieren que todos los nodos de la red se actualicen para poder ser compatibles entre sí, mientras que en las blandas se mantiene la retrocompatibilidad y los nodos pueden elegir actualizarse para disfrutar de las nuevas características.Esto puede llegar a producir una división de la misma, que acaba generando un conjunto de cadenas a partir de una anterior. El ejemplo más famoso es el de Ethereum y Ethereum Classic, que fragmentó la red original debido a que un fallo de software acabó con un hackeo en el que se llegó a “robar” una gran cantidad de dinero en la criptomoneda de la red. La división llegó cuando un grupo decidió seguir adelante a pesar del problema anterior y otro grupo decidió revertir el error y “deshacer” el hackeo producido en la red.

2.2 Smart Contracts

Los Smart Contracts son programas asociados a la blockchain y que se ejecutan en la propia cadena mediante transacciones de los usuarios. Básicamente es una colección de código en forma de funciones y datos que forman su estado, los cuales residen en una dirección en la blockchain.

La primera blockchain que permitió crear smart contracts y almacenarlos en ella fue Ethereum. Dicha red se centra en ellos para distinguirse de las demás, ya que al contrario

que Bitcoin, que es una red específicamente centrada en la divisa, Ethereum se autodenomina una red de cómputo. Esta blockchain va más allá de la propia criptodivisa de la red y se centra en las aplicaciones descentralizadas que se pueden lograr mediante la programación de smart contracts.

Los smart contracts se programan mediante diferentes lenguajes de programación. En este trabajo se ha utilizado Solidity, un lenguaje orientado específicamente para generar contratos que se ejecuten en la máquina virtual de Ethereum (EVM). Esta es una pieza fundamental de las blockchains basadas en Ethereum, que permite que se ejecuten los contratos en la propia cadena. Solidity tiene una sintaxis similar a Javascript y es un lenguaje orientado a contratos.

En Ethereum los contratos inteligentes son un tipo de cuenta. Poseen una dirección, un balance de la criptodivisa (Ether) y pueden recibir y lanzar transacciones en la red. Los usuarios que interactúan con ellos lo hacen llamando a funciones especificadas como públicas o externas del propio contrato. Dichas funciones pueden obtener valores de entrada y emitir valores de salida.

El despliegue de un smart contract en la red se considera una transacción. Los usuarios pueden desplegarlos pero también los contratos pueden a su vez desplegar otros contratos. Como los smart contracts se encuentran dentro de la propia red, las transacciones se llevan a cabo dentro de la blockchain (en la EVM) y todos los nodos deben ejecutar el mismo código. Los smart contracts poseen muchas limitaciones.

En Solidity, en comparación con otros lenguajes de programación de alto nivel no orientados a programación blockchain, destacan varias limitaciones:

- Tipos de datos reducidos y limitados a la hora de agruparlos (no se pueden crear arrays de strings).
- La cantidad de variables que puede manejar una función es de un máximo de 16. Usar un número mayor conlleva que el contrato no se pueda compilar.
- A la hora de devolver datos desde funciones, existe el tipo Estructura pero no se puede devolver desde una función.

Además de tener en cuenta estas limitaciones, los smart contracts (sobre todo si se van a usar en una red pública) se deben programar pensando exhaustivamente en el gas. El gas es una unidad de medida que cuantifica el esfuerzo realizado por Ethereum a la hora de realizar transacciones en la red. El gas se paga en la divisa de la red (en Ethereum con el Ether del usuario que lanza la transacción).

Todas las acciones tienen un coste, desde sumar a modificar el valor de una variable. El sistema está diseñado así para evitar que el lanzamiento de una transacción con un bucle infinito colapse la red eternamente, ya que se ejecuta en la propia

Blockchain, en la EVM. Para lograr esto las transacciones tienen un límite de gas, y si llegan a dicho límite se consideran inválidas. Las criptomonedas gastadas en gas no se recuperan, por lo que hay que ser muy cuidadoso cuando se lancen las transacciones. Como alternativa, se puede especificar un máximo de gas al realizar la transacción para evitar gastar más de lo esperado, pero de la misma manera cuando se alcanza dicho máximo se invalida la transacción y los recursos usados se pierden. Otra cuestión interesante es que el propio usuario puede establecer el precio del gas. Esto sirve para que al lanzar una transacción con un precio de gas alto, los mineros la seleccionen como prioritaria para minar los bloques debido a que obtienen un mayor beneficio con ella, lo cual consigue que la transacción se lleve a cabo en menor tiempo. Dos contratos pueden tener una funcionalidad similar pero la forma en la que estén programados puede suponer que la misma transacción en uno sea más cara que en el otro. Los altos costes de la red, actualmente hacen contraproducente el desarrollo de aplicaciones distribuidas dentro de una red pública.

En Solidity existen funciones que no consumen gas cuando se las llama. Esto es así porque no modifican la cadena, sino que se limitan, como mucho, a consultar datos en ella. Están las tipo Pure, que son funciones que que ni modifican la blockchain ni usan ninguna variable externa del smart contract en el que está definida. Mientras que las funciones View sí usan variables del smart contract pero no modifican su estado, se podría decir que actúan como una consulta.

Los smart contracts también pueden obtener información del exterior de la cadena para adaptarse mejor a los casos de uso del mundo real, donde los datos precisos son cruciales. Esto se consigue mediante los denominados Oráculos, que pueden tener diferente forma, desde fuentes de datos a APIs web. Entre otros tipos de Oráculos tenemos: hardware, software, consensus, inbound y outbound. Los más comunes son los de software, que representan las APIs de las que se obtiene información directamente. Para obtener la información a partir de ellos se les debe llamar en el propio código de Solidity, es decir, no se actualiza automáticamente.

2.3 Blockchain Pública

Como se ha descrito anteriormente, las redes públicas son las más conocidas y las que reflejan mejor la idea principal de las blockchains, un entorno distribuido seguro y confiable.

Como indica su propio nombre cualquiera se puede unir a las redes públicas creando una cuenta en ellas (incluso varias cuentas) y realizar transacciones libremente en la red. Todos los usuarios tienen acceso a todos los demás usuarios y contratos de la cadena (aunque cada contrato interiormente puede establecer medidas de seguridad para restringir el acceso a según qué usuarios).

Los costes derivados de las transacciones hacen inviable el desarrollo de aplicaciones distribuidas en este tipo de redes. Esto ocurre en Ethereum, a pesar de que inicialmente la red estaba pensada para soportar esta clase de aplicaciones mediante los smart contracts.

Con carácter público, existen un conjunto de redes llamadas de prueba. Estas redes se usan para probar aplicaciones en una futura producción en la red principal. Ethereum tiene varias redes de prueba como Görli, Kovan, Rinkeby, las tres con algoritmo de consenso “Proof of Authority” y Ropsten con “Proof of Work”. Ya que son de prueba existe un mecanismo llamado “grifo” que permite recibir criptodivisa de la propia red para llevar a cabo las transacciones de manera gratuita.

En general tienen las mismas características y el hecho de ser gratuitas facilita el lanzamiento de aplicaciones distribuidas a producción ahorrando costes. En la práctica ocurre que al ser gratuitas dependen completamente de los administradores ya que no produce un beneficio para los propios mineros. Esta dependencia en ocasiones conlleva que la propia red no se encuentre en un estado correcto. No es de extrañar que en muchas ocasiones estas redes estén caídas o tengan otro tipo de problemas técnicos que las hacen inoperativas.

2.4 Blockchain Privada

Como su nombre indica este tipo de redes están diseñadas para limitar la participación en las mismas a individuos totalmente confiables, generalmente participantes de una misma empresa o de varias empresas con objetivos de negocio comunes. La principal desventaja sobre las redes públicas es que los participantes deben coordinarse y ponerse de acuerdo para formar una red a modo de consorcio. Pero en cambio poseen un gran número de ventajas:

- No es necesaria una criptodivisa, debido a que no se necesita incentivar el minado de bloques, ya que los propios administradores de la red se encargan de ello.

- Al no existir minería no se debe tener tanto en cuenta el gas que gastan las transacciones, con lo que se tiene más libertad para programar los Smart Contracts.

- Las transacciones en general son más rápidas debido a que no es necesario un algoritmo de consenso como tal, sino un mecanismo para ordenar las transacciones. Aunque las que parten de una base pública suelen usar el algoritmo “Proof of Authority”, que se basa en que un grupo de administradores sean los que validen los nuevos bloques y su seguridad se basa en la confianza que se tenga con dichos administradores, es decir, su reputación.

- En las redes privadas se pueden lanzar transacciones privadas que solo sean

visibles para los participantes designados por el nodo que lanza la transacción. Existen diferentes formas de establecer esta privacidad. Todos los nodos validan la lista de transacciones, mientras que sólo se revela el detalle de las transacciones y contratos privados a las partes interesadas. El estado privado de los smart contracts es validado sólo por las partes que tienen acceso a ellos. La privacidad de los datos es a nivel de nodo.

- En general los nodos son limitados y la cadena es de menor tamaño que las de redes públicas. Por ello es más sencilla la transmisión de nuevos bloques a todos los nodos participantes y suelen ser más ligeras debido a que se usan para un único fin generalmente.

Todos estos factores las hacen idóneas para casos de uso de negocios en los que la privacidad, seguridad y validación de los datos sea especialmente importante. Todos estos factores tienen un coste de forma que mantener este tipo de redes es más caro que mantener una base de datos centralizada, por lo que se deberá estudiar su viabilidad previamente.

Capítulo 3 Estado del Arte

3.1 Blockchain en seguros

Blockchain es una tecnología que se adapta especialmente a los casos de seguros [4], debido a la fiabilidad, seguridad y confianza que aporta. Estas redes pueden actuar como intermediarios para revisar los posibles casos de siniestro e indemnización de forma objetiva mediante los smart contracts. Tiene el potencial para convertir los actuales seguros en modelos peer to peer de economía compartida.

Actualmente es vista como una tecnología emergente con gran potencial, pero las compañías en su mayoría aún permanecen escépticas para su aprovechamiento. Una adopción mayor se dará cuando estas compañías analicen el potencial de la tecnología, que tiene la capacidad incluso para abrir nuevos mercados [5].

Estos nuevos tipos de seguros corresponden a seguros de pago por uso y especialmente microseguros, que no tienen tanto uso actualmente debido a los costes administrativos en cuanto a lo que supone contratarlos. Pero el hecho de que dichos seguros se puedan acceder de una forma sencilla mediante la identificación en la blockchain de la aseguradora hace que contratarlos sea muy sencillo, llegando incluso a no necesitar la aprobación de la aseguradora de manera directa, ya que se pueden medir las condiciones de contrato en los smart contracts. Ésto hace que sólo se necesite la confianza de la aseguradora en un inicio para obtener las claves y el nodo necesarios para participar en su blockchain privada. Ésto no solo afecta a este tipo de contratos aumentando su viabilidad, sino que también se reducen los costes del proceso de contratación y se convierte en instantáneo [6].

Una de sus características se centra en mejorar la experiencia del cliente a la vez que reduce los costes operacionales. Ésto se consigue gracias a que la red blockchain junto con los smart contracts aumentan la velocidad de las reclamaciones a la vez que reducen los errores asociadas a las mismas debido a la automatización del sistema.

También permite el seguimiento de las reglas de la póliza, por ejemplo, en un caso de seguro de coche poder confiar en que el asegurado repare el automóvil en uno de los mecánicos certificados por la compañía aseguradora mediante una transacción al smart contract por parte de dicho mecánico que pruebe su identidad, con lo que se reducirían ampliamente los casos de fraude.

Otros casos de uso más complejos podrían requerir del uso de oráculos que obtengan información del mundo real [7]. Por ejemplo, el caso de un seguro turístico que obtenga datos sobre el tiempo atmosférico e indemnice al asegurado en caso de que durante su estancia turística no ha podido disfrutar de una cantidad de horas de sol establecidas previamente. De este mismo tipo serían los seguros orientados a las catástrofes naturales [8].

En concreto es especialmente útil para los casos de seguros paramétricos, ya que al tener un evento desencadenante que se ha especificado previamente es más sencillo comprobar los casos de posible pago mediante un comprobante de que se ha dado dicho evento al añadirlo al smart contract correspondiente a la póliza.

Por ejemplo, la compañía francesa AXA [9] desarrolló un seguro paramétrico de viajes basado en blockchain en el que la condición de siniestro se daba cuando un cliente sufría un retraso en alguno de sus vuelos asegurados que superase las dos horas. El proceso es totalmente automatizado y seguro, sin necesidad de intervención manual. Los detalles de la póliza se codifican en smart contracts, los cuales integran los datos del tráfico de vuelos en su sistema. Esto hace que sea mucho más transparente debido a que el propio smart contract se encarga de emitir la compensación en base a la información que posee, notificando adecuadamente al cliente. Esto hace que los propios clientes confíen aún más en el seguro.

Blockchain se usa como un registro público y como un protocolo de ciberseguridad, con lo que ha tenido un amplio impacto tanto en la industria de seguros como en la financiera [10].

Una de las cualidades de los seguros de propiedades es la falta de transparencia. Muchas indemnizaciones se resuelven por estimaciones subjetivas, como la cuantificación de los daños. Mediante blockchain se pueden codificar las reglas del seguro para hacerlo más transparente de cara a los clientes, incluso llegando a realizar las indemnizaciones en tiempo real sin necesidad de terceras partes. Esto se puede conseguir mediante la implantación de sensores IoT [11] que automáticamente alerten cuando se produce un accidente. El propio smart contract que recibe dicha información podría encargarse de estudiar la indemnización e informar a los servicios médicos si fueran necesarios.

El hecho de que la póliza como tal esté programada en un smart contract hace que las condiciones contractuales queden muy claras, eliminando casi por completo la parte subjetiva de las mismas. A su vez esto permite facilitar la comparación entre las pólizas similares que ofrecen diferentes aseguradoras.

En la la automatización de los pagos sería necesario combinar tanto blockchains privadas como públicas. En la privada se establecería el ecosistema de la póliza en la que se estudian los casos de los clientes y se detectan los casos de indemnización y en la blockchain pública se establecería el pago de dicha indemnización al desencadenarse desde el evento obtenido en la red privada. Ésta necesidad es debida a que dentro de las redes privadas las criptomonedas (si las hubiera) carecen de valor de cara al exterior. La ventaja de estos pagos sería la inmediatez de los mismos al contrario que en la situación actual de muchos seguros.

La disponibilidad de mecanismos criptográficos de blockchain facilita la identificación y verificación de los clientes. Los clientes se identifican con una dirección única en la red y dicha dirección se puede conectar a sus datos. En el caso de perder o que le roben sus claves de acceso a la blockchain le podrían suplantar la identidad y la única forma de arreglar la situación sería establecer unas nuevas credenciales y dar por perdidos los datos de las anteriores, debido a que no se trata de un sistema centralizado en el que un administrador pueda devolver el poder sobre su antigua cuenta al cliente, sino uno descentralizado cuyo perfecto funcionamiento depende altamente de la seguridad de las claves de identificación de la red.

En este mismo aspecto existen compañías que ofrecen servicios de seguridad externa basados en blockchains públicas que generan una red de validadores que se encargan de validar las transacciones de los clientes proveyendo una capa de seguridad extra, aunque dicho servicio tendría un coste por cada transacción validada. Ejemplos de estos servicios son el caso de Civic [12], basado en la blockchain de Bitcoin y KYC Legal [13], que reside en la blockchain de Ethereum.

3.2 Blockchain en el sector salud

En los últimos tiempos la tecnología blockchain ha visto incrementado su papel en los servicios de salud. Combinando su uso junto a la inteligencia artificial [14] se puede crear un sistema predictivo que ayude a contener el impacto de posibles pandemias.

La propia Organización Mundial de la Salud ha recomendado a todos los países que se establezca un “plan de pandemias” para nuevos casos como el vivido con el Covid-19. La tecnología blockchain es un candidato idóneo tanto para compartir como para manejar de mejor manera los datos de los pacientes mediante los intercambios de datos criptográficamente seguros que provee.

Las aplicaciones de “machine learning” necesitan una gran cantidad de datos para generar un modelo con cierta precisión. La obtención de datos médicos directamente desde una blockchain, destinada a ellos, podría generar modelos predictivos muy precisos y tratar mejor enfermedades nuevas y contagiosas.

Esta tecnología es idónea para este tipo de casos donde la privacidad de los datos alcanza una gran importancia, ya que permite que sólo se muestren los datos relevantes para cada caso. Además la posibilidad de reunir todos los datos del paciente en una única base de datos permite que se le pueda realizar un tratamiento más eficaz, debido a que si se cambia de médico u hospital no tiene por qué tener el historial médico completo [15].

La aplicación de blockchain en este sector permitiría mejorar la prevención y el control de las patologías y por tanto una mejor gestión del riesgo clínico en contextos de emergencia como la pandemia global actual.

La rápida aparición y difusión del Coronavirus por todo el mundo ha demostrado la ineficacia de los sistemas de vigilancia sanitaria vigentes a la hora de gestionar la salud pública durante emergencias. A su vez nos ha permitido comprobar la falta de sistemas predictivos avanzados basados en el intercambio de datos clínicos a gran escala, que habrían permitido atenuar las consecuencias de dicha crisis [16].

La inmutabilidad de blockchain permite llevar un mejor registro de la medicación de los pacientes y podría ser un elemento clave a la hora de eliminar el fraude de recetas médicas [17].

Es necesario el uso de la tecnología actual para el intercambio de información médica entre diferentes registros electrónicos de salud. El manejo de estos registros mediante blockchain puede reducir el sesgo clínico y mejorar en general el ámbito sanitario. Esto se puede conseguir guardando los datos médicos en blockchains y usando una blockchain que actúe como puente donde se produzcan los intercambios de la información contenidas en las blockchains médicas. Lo cual aseguraría intercambios criptográficamente seguros de datos entre dos o más usuarios [18].

La descentralización de los datos médicos fomenta el desarrollo de medicina de precisión y la personalización de la prevención, el diagnóstico y el tratamiento de un solo paciente. La información compartida de esta manera puede contribuir a la difusión de diferentes resultados científicos de estudios e investigaciones con lo que se fomentaría la difusión de las mejores prácticas clínicas y la medicina basada en evidencias [19].

Capítulo 4 Desarrollo del SPC-19

4.1 SPC-19

Este trabajo desarrolla una aplicación para la gestión de un seguro paramétrico aplicado al Covid-19 en la que participan: aseguradora, hoteles y laboratorios. La principal ventaja respecto a los seguros tradicionales es la automatización del sistema.Los hoteles se registran en la red de la aplicación contactando con la aseguradora. Los hoteles son los encargados de contratar los seguros, sin necesidad de aprobación por parte de la aseguradora. Todas las transacciones de la red se realizan contra una blockchain que actúa como conector global. Los smart contracts que se generan al contratar el seguro son los encargados de informar si se dan casos de siniestro. En el siguiente esquema [Figura 1] se muestran las diferentes transacciones que realizan los participantes durante una póliza de este seguro.

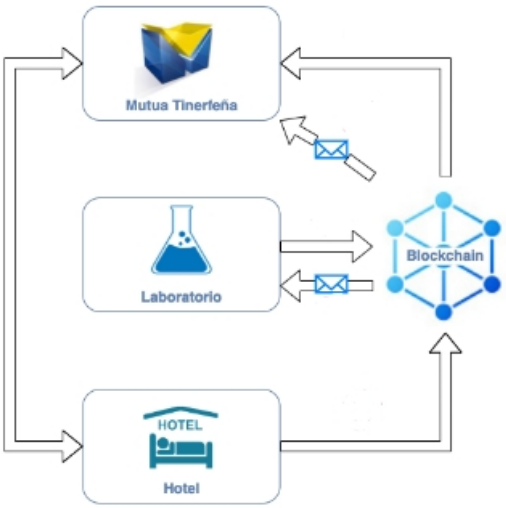


Figura 1. Esquema de transacciones.

Para guardar todos los datos de manera segura y fiable se ha usado una blockchain privada, por lo que los participantes deben ser admitidos para poder modificar la cadena de bloques de tal forma que posean un nodo y un par de claves.

Para la implementación se ha optado por una blockchain privada debido a que en ellas se pueden realizar transacciones privadas en las que el emisor selecciona los nodos que tienen acceso a dichas transacciones, con lo que se establece la privacidad de los datos y cada participante sólo tiene acceso a la información que necesita. Por otro lado,

actualmente este tipo de aplicaciones descentralizadas son inviables en las redes públicas, principalmente por los costes que conlleva cada transacción en estas redes. Además las restricciones de inclusión en las redes privadas permiten a la aseguradora tener el control de quién puede contratar sus pólizas, ya que previamente le ha debido permitir formar parte de la red.

La aplicación consta de 3 partes: la interfaz de usuario, la API Rest y los smart contracts. Como parte del equipo de proyecto he trabajado por completo en el desarrollo de los smart contracts y en la implementación de las transacciones con ellos en la API.

Cada uno de los participantes en la aplicación (aseguradora, laboratorio y hotel) tienen diferentes funciones dentro de la red que se explicarán detalladamente más adelante.

Por otro lado comentar que la aplicación que se ha programado está pensada para realizar una prueba con un único hotel y un único laboratorio, ya que debido al tiempo limitado del proyecto (se propuso acabarlo a finales de abril). Lo ideal era conseguir un producto funcional para analizar su rendimiento y estudiar su viabilidad de cara a presentarlo ante los interesados de la Mutua Tinerfeña.

El cliente ethereum en el que se basa la blockchain privada es Hyperledger Besu. Ésto es importante debido a que el estilo de la privacidad es diferente según la tecnología que se use. En Besu existen los denominados “grupos de privacidad”, formados por grupos de nodos. Todos los nodos observan el mismo estado público de la blockchain mientras que cada nodo sólo puede observar y modificar los estados privados correspondientes a grupos de privacidad a los que pertenece. Esto hace que al crear un nuevo contrato mediante una transacción privada se deba especificar su grupo de privacidad y que todas las posteriores transacciones que se realicen en él deben usar el mismo grupo. Con lo cual el estado del contrato es el mismo para todos los participantes que pueden observarlo, a diferencia de otras tecnologías que promueven que dependiendo del usuario que observe el contrato vea unos datos u otros, permitiendo establecer la privacidad a nivel de nodo. Debido a esto, en Besu la privacidad se establece a nivel de contrato. Por ello es necesario que si se quiere ocultar información a algunos participantes se creen diferentes contratos que pueden ser del mismo tipo o diferente.

4.2 Smart Contracts

La tecnología escogida para establecer la red privada (Besu) ha determinado la creación de tres tipos diferentes de contratos. Con ello se consigue dividir los datos de tal manera que diversos datos privados solo sean accesibles para los participantes que deben conocerlos para realizar su función adecuadamente. En el caso de programar un único contrato que englobe toda la información de la red, todos los participantes podrían tener acceso a todos los datos del contrato. En este caso se deberían establecer roles en el propio contrato para tratar la seguridad, regulando las funciones que puede usar cada participante. Por ello, para adaptarse mejor a Besu y aplicando una arquitectura orientada

a objetos se ha dividido el contenido en tres contratos, PCR, Póliza y General.

4.2.1 Contrato PCR

Este contrato equivale a una PCR enlazada a un cliente y a la póliza a la que pertenece. Posee datos como su propia ID, la ID del cliente, la ID de la póliza, el resultado, fechas de petición y resultado y metadatos sobre la misma. Sus funciones principalmente se enfocan en obtener la información sobre ella y añadir el resultado una vez la prueba se ha completado. Engloba desde la petición de la prueba hasta que se le añade el resultado. Lanza eventos cuando se crea y cuando se actualiza su resultado.

En un inicio se pensó en introducirlos directamente en los contratos Póliza, pero al ha sido inviable por la forma de privacidad de Besu, ya que requeriría que el laboratorio entrase en el grupo de privacidad del contrato póliza, lo cual se evita para limitar la información a los participantes necesarios y mantener la privacidad de la misma. Con lo cual la forma de representar PCRs en el contrato póliza es mediante una estructura y no con un contrato PCR. Los contratos PCR se comparten entre el hotel que ha contratado la póliza cuyo cliente se va a realizar la prueba y el laboratorio.

```
/// @notice Initialize the PCR.
constructor(
    bytes32 _id,
    bytes32 _insuranceId,
    bytes32 _insuredId,
    uint256 _requestDate,
    address _insuranceAddress
) {
    pcrData.insuranceAddress = _insuranceAddress;
    pcrData.requestDate = _requestDate;
    pcrData.insuredId = _insuredId;
    pcrData.insuranceId = _insuranceId;
    pcrData.id = _id;
    owner = payable(msg.sender);
    completed = false;
    pcrData.result = "UNDEFINED";
    deleted = false;
    emit pcrCreated(_id, _insuranceId, _insuredId, _requestDate, address(this), _insuranceAddress);
}
```

Figura 2. Ejemplo de constructor de contrato PCR.

En esta imagen [Figura 2] observamos como los tipos de dato bytes32 se usan para guardar los identificadores en los smart contracts. A su vez podemos observar que se guarda la dirección del contrato Póliza al que pertenece la prueba PCR y se emite un evento en su creación que a la postre es el encargado de notificar al laboratorio que debe realizar dicha prueba. El laboratorio se encarga de realizar la cita al cliente, el contrato inteligente no entra en detalles sobre la fecha o lugar de la realización de la prueba.

4.2.2 Contrato Póliza

En este contrato se guarda la información tanto de la propia póliza como del hotel que la contrata. El contrato deriva del contrato Seriality [20], que se usa para serializar grandes conjuntos de datos en arrays de bytes para evitar las limitaciones de Solidity en cuanto al número máximo de variables que se puede devolver desde una función. El propio contrato emite eventos de PCR positiva y de caso de pago (siniestro). El contrato Póliza se comparte entre la aseguradora y el hotel que contrata esa póliza. Es importante mencionar que para tratar con RGPD (Reglamento General de Protección de Datos) no se han incluido datos personales de los clientes dentro del contrato Póliza. Los datos del hotel en cambio si se añaden, ya que al ser datos de empresa y contando con que se ha firmado un acuerdo de cooperación para pertenecer a la red blockchain cumplen con el RGPD.

Como se comentó en el punto anterior guarda los datos de las PCRs de los clientes como estructuras de datos y no como contratos PCR. A la hora de emitir el evento de pago se comprueba que se ha actualizado una PCR con resultado positivo dentro del tiempo correspondiente a la póliza. Las funciones se enfocan a tratar con las PCRs de los usuarios, crearlas y actualizarlas y a devolver los datos de la póliza. En el contrato se encuentra una función que devuelve toda la información de la póliza convertida en un array de bytes, por lo que en la API he diseñado una función en Javascript que se encarga de obtener el array de bytes y convertirlo en un objeto Póliza.

```
/// @notice Convert all the information of the insurance into an array of bytes.
/// @dev Returns all data of the insurance in an array of bytes.
function serializeInsurance() external view returns (bytes memory _serializedInsurance, uint256 _size) {
    uint16 nameSize = uint16(sizeOfString(takerData.takerName));
    uint16 addressSize = uint16(sizeOfString(takerData.takerAddress));
    uint16 emailSize = uint16(sizeOfString(takerData.takerEmail));

    uint256 offset = (
        4 + // 2 uint16
        128 + // 2 uint256 and 1 bytes32 Insurance Info
        (32 * insuranceData.insuredNumber) +  // ids of insureds
        (32 * insuranceData.insuredNumber * 2) + // negative previous PCR hash and date
        (32 * 9) + //bytes32 of takerData
        nameSize + addressSize + emailSize +
        10 + // uints16 to check string sizes (3) and to check insuredNumber (1) and to check number of PCRs (1)
        2 + // 2 bools
        (32 * 5 * insuranceData.pcrNumber) + // Info about all PCRs of insureds
        30 // Extra size to prevent null return when no PCRs added. Used in pcrNumber
    );
    _serializedInsurance = new bytes(offset);
    _size = offset;

    // Insurance Info
    // serialize Insurance ID
    bytes32ToBytes(offset, insuranceData.id, _serializedInsurance);
    offset -= 32;

    // serialize sinister compensation
    uintToBytes(offset, insuranceData.sinisterCompensation, _serializedInsurance);
    offset -= 32;

    // serialize Insurance Start Date
    uintToBytes(offset, insuranceData.insuranceStartDate, _serializedInsurance);
    offset -= 32;
```

Figura 3. Ejemplo de función de serialización de contrato Póliza (recortada).

En este ejemplo [Figura 3] podemos observar cómo se van añadiendo los diferentes datos de la póliza al array de bytes que previamente se ha generado con el espacio suficiente para poder guardar todos los datos necesarios. Es importante resaltar que de cara a descifrar el array de bytes resultante es necesario tener en cuenta la posición de cada uno de los datos, así como el tamaño que ocupa cada uno en número de bytes.

4.2.3 Contrato General

Este último contrato se encarga de guardar todas las pólizas presentes y pasadas que le han contratado a la aseguradora y enlazarlas a los hoteles que las han contratado. En un inicio estaba pensado para que existiera un único contrato General y que la aseguradora pudiera obtener de él todas las pólizas que se le han contratado y en cambio cada hotel solo pudiera obtener las pólizas que han contratado ellos mismos. Este enfoque no se corresponde con la privacidad de Besu, ya que todos los participantes de un contrato son capaces de observar los mismos datos sobre el mismo. Si se quisiera mantener un único contrato y tratar la privacidad a nivel de contrato con roles, el contrato no escalaría adecuadamente debido a que se deberían añadir todos los hoteles que pueden ser capaces de contratar pólizas desde el principio al crear el contrato y no permitiría ir añadiendo hoteles en el futuro. Por ello se crea un contrato General cada vez que un hotel se une a la red y se comparte con dicho hotel y con la aseguradora.

Este contrato guarda los contratos Póliza referentes a las pólizas que contrata el hotel correspondiente de dicho contrato general. Las funciones del mismo principalmente son las de añadir pólizas y recuperar los datos de las pólizas, ya sea sus datos en forma de array de bytes, un array de bytes de todas las pólizas combinadas (para el cual también existe una función en Javascript que convierte el array de pólizas en un array de objetos póliza de Javascript) o añadiendo la ID de la póliza obtener la dirección de su contrato Póliza.

```
/// @notice Adds a new insurance to the hotel which is the taker of it.
/// @dev Modify the mappings that link hotels and insurances.
function addInsurance(Insurance _newInsurance) external {
    _newInsurance.addSpcAddress(address(this));
    require(idInsuranceToHotel[_newInsurance.getId()] == 0, "Insurance ID has already been registered");
    if (!hotelHasInsurances[_newInsurance.getTakerId()]) {
        hotels.push(_newInsurance.getTakerId());
        hotelHasInsurances[_newInsurance.getTakerId()] = true;
    }
    hotelInsurances[_newInsurance.getTakerId()].push(_newInsurance);
    InsuranceLocation[_newInsurance.getId()] = hotelInsurances[_newInsurance.getTakerId()].length - 1;
    idInsuranceToHotel[_newInsurance.getId()] = _newInsurance.getTakerId();
    idInsuranceToAddress[_newInsurance.getId()] = address(_newInsurance);
}
```

Figura 4. Ejemplo de función de añadir póliza del contrato general.

En esta función [Figura 4] se puede observar un tratamiento de errores que no permite añadir dos pólizas con la misma identificación. También se observa cómo se rellenan los diferentes mappings presentes en el mismo al añadir una nueva póliza.

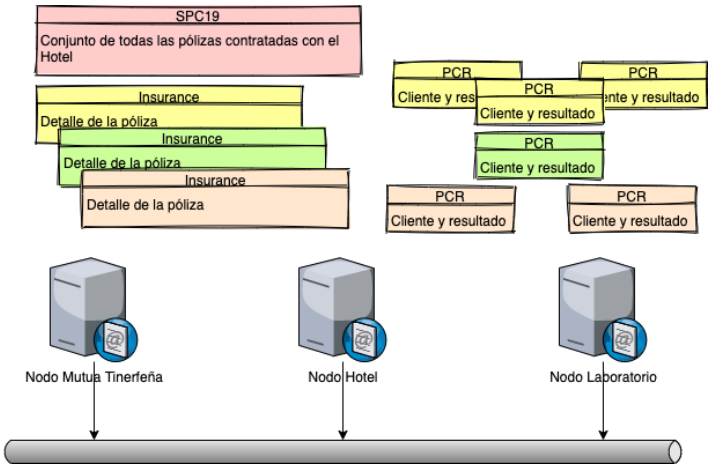


Figura 5. Distribución de smart contracts.

En esta figura [Figura 5] se observa la distribución de los smart contracts en cuanto a los nodos de los participantes. Los dos tipos de grupos de privacidad corresponden a Hotel/Aseguradora y Hotel/Laboratorio. En el primer grupo se comparten los contratos General y Póliza y en el segundo los contratos PCR.

4.2.4 Eventos

Los eventos se manejan mediante un componente que se encarga de escuchar en la blockchain y recibir los eventos que emite. Para ello se debe especificar el grupo de privacidad si se trata de eventos emitidos por transacciones privadas. Este diseño por parte de Besu no cuida la privacidad en cuanto a eventos privados, ya que el ID del grupo de privacidad se puede obtener mediante las claves públicas de los integrantes de dicho grupo, por lo que cualquiera podría estar escuchando eventos privados aunque no fueran dirigidos hacia él, por lo que sería adecuado encriptar los datos que se envían junto a los eventos para aumentar la privacidad y seguridad de los datos. Esta solución no se ha tomado en la prueba, ya que la demostración se ha centrado en establecer la funcionalidad del seguro, aunque no se envían datos privados en los eventos para cumplir con el RGPD.

Se lanza un evento cuando se crea una nueva solicitud de PCR por parte del hotel. Dicho evento se utiliza para enviar los datos de la solicitud al laboratorio por medio de un correo electrónico. Estos correos contienen datos personales de los turistas la primera vez que se lanza por cada turista de la póliza. Datos como el nombre, teléfono, correo electrónico… para que el laboratorio pueda citar al paciente para realizarle la prueba. Si se requieren varias pruebas los siguientes correos sólo incluirán los datos relativos a las IDs del turista, póliza y la PCR ya que, como se indicó en su momento, no se guardan datos personales de los clientes en la blockchain, y se asume que el laboratorio puede llegar a los datos personales al guardarlos del primer correo recibido y mapearlos con las IDs del turista y la póliza.

Un evento de PCR actualizada se emite cuando el laboratorio obtiene el resultado de alguna de sus pruebas y actualiza el contrato correspondiente. Dicho evento incluye los

datos de ID de póliza, turista y PCR así como el resultado de la prueba. Es crucial para actualizar los contratos Póliza debido a que, como se ha comentado anteriormente, el laboratorio actualiza el contrato PCR pero éste no está relacionado directamente con el contrato Póliza, por lo que el manejador de eventos debe actualizar la estructura PCR determinada en el contrato Póliza en base a los datos de este evento.

Por último cuando se actualiza un resultado a positivo en una PCR en el contrato Póliza se lanza un evento de PCR positiva donde se indica la ID del turista que ha contraído la enfermedad, la ID de la póliza a la que pertenece y la fecha del resultado. En el caso de que sea la primera vez en la póliza que se actualice una PCR a positiva y dicha actualización se ha realizado durante la vigencia del seguro se emitirá un evento de pago que el manejador de eventos usará para enviar en correo electrónico a la aseguradora indicando la ID del hotel y la ID de la póliza de dicho hotel que se ha establecido como siniestro. La aseguradora puede usar dicho evento para efectuar, e incluso automatizar, el pago de la indemnización.

```
/// @notice Fires when the contract get a positive PCR.
/// @dev Emit the sinister event.
function compensation() public {
    if (positivePcrTest && !paymentEmitted) {
        emit checkPayment(takerData.takerId, insuranceData.id);
        paymentEmitted = true;
        timePaymentEmitted = block.timestamp;
    }
}
```

Figura 6. Ejemplo de emisión de evento de pago de indemnización.

En la función de pago [Figura 6] primero se comprueba que se ha dado un resultado positivo en una prueba PCR y además se comprueba que es el primer caso de la póliza, para no enviar varias veces la indemnización en caso de varios contagios, ya que se ha establecido una indemnización general por grupo en caso de cualquier número de positivos. A su vez se guarda en el contrato la fecha en la que se produce el siniestro.

4.3 Participantes

En este apartado se explicarán los roles de los diferentes participantes de la aplicación. El ciclo de vida de una póliza se dividirá en los diferentes pasos que supone y se explicarán detalladamente cada uno de ellos y el participante encargado de realizarlo.

Se ha establecido una única API Rest que actúa de manera distinta según la identidad del usuario que interactúa con ella de entre los participantes. A su vez en el caso de hotel y de aseguradora se ha usado un manejador de eventos de la blockchain para realizar transacciones en base a los eventos que se obtienen.

La blockchain actúa como un conector software[21] entre todos los participantes y sus APIs. Es un conector especialmente centrado en la seguridad, privacidad, confiabilidad y escalabilidad. Es similar a una base de datos descentralizada pero con especial énfasis en la seguridad. Permite la comunicación entre los distintos participantes de la red sin necesidad de una conexión directa entre ellos.

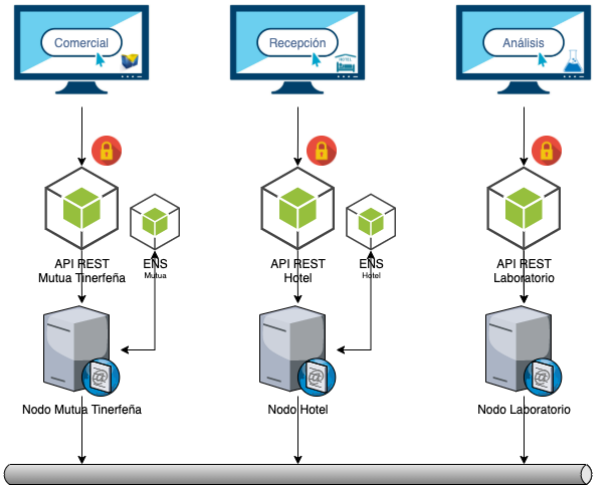


Figura 7. APIs y nodos.

Esta imagen [Figura 7] representa las diferencias respecto a la API, interfaz de usuario y nodo que usa cada tipo de participante de la red y si posee o no un manejador de eventos.

4.3.1 Aseguradora

Su principal cometido es registrar los nuevos hoteles dentro de la red blockchain otorgándoles un nodo y su par de claves y realizar los pagos relativos a la indemnización cuando se reciben los eventos de siniestro por parte de los contratos Póliza. A su vez pueden consultar el estado de todas las pólizas de todos los hoteles, tanto las actuales como las finalizadas.

El evento desencadenante del siniestro consiste en una prueba PCR positiva realizada a uno de los clientes asegurados en uno de los laboratorios que pertenecen a la red blockchain de la aplicación. La alta fiabilidad de estos test, así como su registro fehaciente dentro de la cadena de bloques mediante una representación de dicha prueba en forma de hash permite automatizar el proceso de forma sencilla y muy transparente de cara a los usuarios finales.

4.3.2 Hotel

Los hoteles se encargan de contratar las pólizas (generando los contratos Póliza) cuando reciben un nuevo grupo de clientes. Los hoteles de la red se encargan de crear y

eliminar las solicitudes de PCR para sus clientes con pólizas activas. Cada hotel, como la aseguradora, es capaz de consultar la información de las pólizas, pero únicamente las que ha contratado él.

4.3.3 Laboratorio

El laboratorio se encarga de consultar las solicitudes PCR y actualizarlas una vez se dispone del resultado.

4.3.4 Flujo de la aplicación

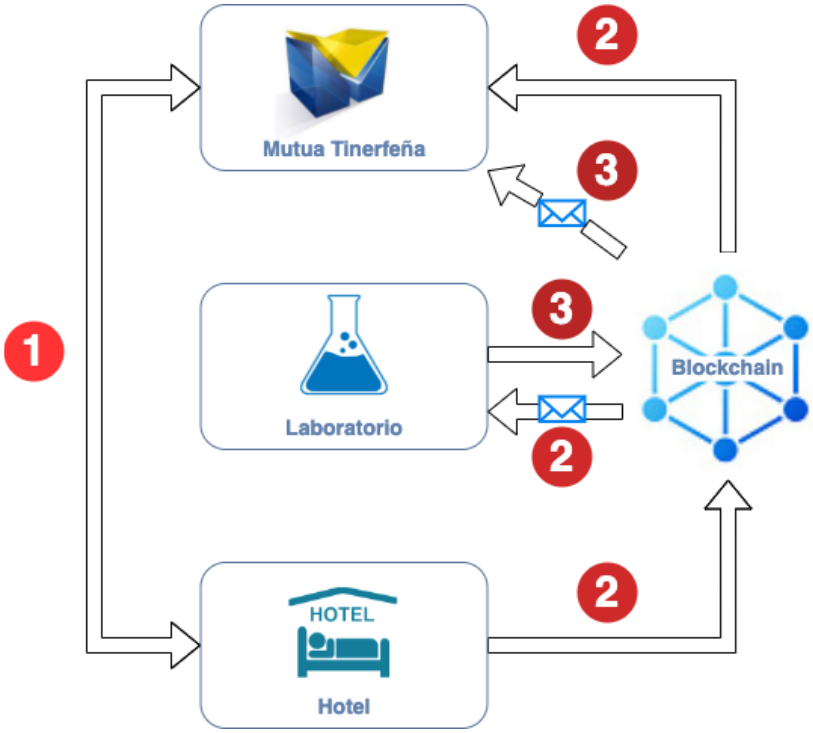


Figura 8. Diagrama de transacciones.

Este diagrama [Figura 8] establece las comunicaciones que se realizan entre los distintos participantes de la aplicación incluyendo la propia blockchain. Las flechas con sobre corresponden a los correos que se envían cuando el servicio de eventos detecta los casos de nueva PCR o los casos de siniestro y se envían los correos necesarios.

El paso ‘1’ del anterior diagrama [Figura 8] representa la relación previa que se debe establecer por parte del hotel para negociar con la mutua su inclusión a la red blockchain y la obtención de un nodo y el par de claves pública/privada (así como usuario y contraseña para acceder a la interfaz de usuario). Una vez registrado podrá empezar a contratar las pólizas SPC-19 por su cuenta.

El segundo paso corresponde a la contratación de dicha póliza por parte del hotel.

En consecuencia provoca que se genere el nuevo contrato póliza que la aseguradora podrá observar la siguiente vez que pida los datos de todas las pólizas, y a su vez la petición de una prueba PCR por turista añadido a la póliza. Ésto se representa enviando un correo al laboratorio con la información de dichas PCR por medio del servicio de eventos cuando se recibe el evento de nueva PCR que se emite en el constructor de los contratos PCR.

Por último el tercer punto corresponde a la actualización de las pruebas PCR por parte del laboratorio en el contrato PCR que se extenderán al contrato Póliza por medio del manejador de eventos. En el caso de que una o varias pruebas hayan dado un resultado positivo dentro de la duración del seguro se enviará un correo electrónico a la aseguradora. Este correo informa de que dicha póliza ha tenido un caso de siniestro y que deberá proceder al pago.

4.4 API

4.4.1 Conexión con Blockchain

Las funciones de la API Rest diseñada se encargan principalmente de lanzar transacciones privadas a la blockchain tanto para modificar el estado de ésta como para consultarlo. Para acceder a dichas funciones se necesita una cuenta en la aplicación que a su vez establece el rol de dicho usuario. Dependiendo del rol se podrá llamar a unas funciones u otras, e incluso podrá variar el comportamiento de las funciones, por ejemplo, al pedir la información de todas las pólizas en el caso de ser la aseguradora se le mostrarán las pólizas de todos los hoteles del sistema, en cambio si se trata de un hotel el que llama a la función únicamente se le mostrarán sus propias pólizas.

Para comunicarse con la blockchain desde el código de la API se usa la librería Web3. Es una librería Javascript que permite interactuar con nodos ethereum. Para poder usarla en una red basada en Hyperledger Besu se debe extender Web3 con la librería Web3-eea, perteneciente a Besu. Ésto permite que se puedan lanzar las transacciones privadas en las redes Besu.

En dichas transacciones se debe especificar el emisor de la transacción, su clave privada, el grupo de claves públicas a los que se comparte dicha transacción y que junto con el emisor de la misma forman el grupo de privacidad, los datos que incluyen la firma de la función y sus argumentos si los hubiera y por último la dirección del contrato en la red. Es importante mencionar que para que funcione correctamente la transacción, la clave privada debe coincidir con la del nodo desde el que se lanza la transacción, cuya dirección de red se requiere a la hora de construir el objeto derivado de Web3-eea que permite lanzar las transacciones.

4.4.2 Endpoints

A continuación se explicarán los diferentes endpoints. No se han incluido funciones de añadir nuevos hoteles debido a que la demostración realizada se ha acotado a la existencia de un único hotel y un único laboratorio. Extenderlo al caso general donde puedan coexistir varios hoteles se explicará en las conclusiones y posibles mejoras.

- Añadir póliza. POST (Hotel): Se encarga de crear un nuevo contrato póliza cuya información se ha enviado junto a la petición. Una vez creado se añade al contrato general correspondiente al hotel que ha enviado la petición. Además se encarga de crear un contrato PCR por cada turista que incluya la póliza para realizar la prueba al final de la estancia y comprobar que no se ha dado un caso de siniestro. En este caso se aprovechan los datos personales de los turistas que se encuentran en la petición para añadirlos en el correo al laboratorio y a continuación se descartan.

- Obtener datos de pólizas. GET (Hotel, Aseguradora): Se encarga de llamar a la función del contrato general (que guarda los contratos póliza) que genera un array de bytes acoplando los datos de todas las pólizas de dicho smart contract. Estos datos se transforman en un conjunto de objetos póliza que es lo que se devuelve en la función. En el caso de que la llame el hotel se obtendrán sólo sus pólizas y en el caso de la aseguradora todas las del sistema.

- Pedir prueba PCR. POST (Hotel): Crea un nuevo contrato PCR y a su vez una estructura PCR dentro del contrato póliza. Es necesario incluir en la petición la identificación de la póliza, del turista al que se va a realizar la prueba y la propia ID de la PCR.

- Obtener datos PCR. GET (Laboratorio): Se obtienen los datos de un contrato PCR al especificar la ID de la pcr, la ID de la póliza y la dirección del contrato PCR dentro de la blockchain. Tanto los hoteles como la aseguradora ya pueden observar los datos de todas las PCRs de cada póliza al usar la función de obtención de datos de pólizas, por lo que no es necesario que puedan usar esta función también, ya que además los contratos PCR no se comparten con la aseguradora, solo entre hotel y laboratorio.

- Actualizar PCR. PATCH (Laboratorio): Una vez se ha realizado la prueba y se ha obtenido un resultado el laboratorio llama a esta función indicando la identificación de la PCR y de la póliza a la que pertenece (además de la dirección del contrato en la blockchain) para actualizar el resultado de la misma. La fecha de resultado se actualizará automáticamente a la hora en la que se lanza la transacción. Ésto es importante debido a que los pagos por siniestro se producen únicamente cuando se ha actualizado a un caso positivo dentro del tiempo del seguro.

- Borrar PCR. DELETE (Hotel): Sirve para cancelar una prueba PCR.Elimina tanto el contrato PCR como el mapeo a la estructura que contiene sus datos en el contrato póliza. Su uso se destina a casos muy concretos donde el paciente posea síntomas en medio de la estancia y se le pida una nueva PCR a realizar lo antes posible, con

lo que en caso de que el resultado sea positivo se podrá cancelar mediante esta función la prueba programada para el fin de su estancia y generar una nueva para el fin de la cuarentena que deba asumir.

```
/**
 * Crea un contrato con el bytecode elegido con las opciones elegidas
 * @param {String} bytecode
 * @param {String} privFrom
 * @param {String} privKey
 * @param {String} privFor
 * @param {Web3} web3
 * @returns {String} Hash de la transacción
 */
async function createContract(bytecode, privFrom, privKey, privFor) {
  return new Promise(async function (resolve, reject) {
    // Creando contrato en nodo Mutua
    const contractOptions = {
      data: '0x' + bytecode,
      privateFrom: privFrom, // orion.member1.publicKey,
      privateFor: privFor, // [orion.member3.publicKey],
      privateKey: privKey, // besu.member1.privateKey
    };
    logger.info('Creating contract...');
    const c = await web3.eea.sendRawTransaction(contractOptions);
    let hash = await web3.priv.getTransactionReceipt(
      c,
      config.orion.taker.publicKey
    );
    if (hash.revertReason) {
      let error = Web3Utils.toAscii('0x' + hash.revertReason.slice(138));
      logger.error(error);
      reject({ code: '400', message: error });
    }
    resolve(c);
  });
}
```

Figura 9. Ejemplo de creación de contrato mediante Web3.eea.

En este ejemplo podemos observar el esqueleto del objeto que se debe incluir para lanzar las transacciones privadas. En caso de dirigirse a un contrato en particular se le añade un atributo “to” en el que se plasma la dirección del contrato en la red. En este caso, al crear un nuevo contrato en la red no hace falta dicho atributo.

Al contrario que en el propio ethereum, que distingue entre transacciones de consulta (View o Pure), que no gastan gas debido a que no modifican el estado de la cadena, sino que simplemente devuelven información sobre la misma. En la librería web3-eea no se diferencia entre consulta o transacción y todas se lanzan con la función “sendRawTransaction”. Esto sin duda es una limitación de la implementación actual de la librería web3-eea y que introduce un retardo innecesario en la ejecución de las operaciones de consulta, que esperamos se resuelva en un futuro debido a que Hyperledger Besu soporta la consulta de datos sin consumo de gas.

4.5 Seguridad

Para evitar la inclusión de datos erróneos dentro de los contratos inteligentes en la propia API se realiza una validación previa sobre los datos introducidos para comprobar que el formato de los mismos es el correcto. Una vez que se ha constatado que el formato es correcto dentro de los smart contracts existen mecanismos de seguridad para evitar transacciones que puedan dañar la integridad del sistema.

En el contrato PCR se establecen sistemas de seguridad para comprobar que la PCR que se quiere actualizar es la adecuada comparando las IDs de pcr, póliza y turista, que realmente no harían falta para actualizar el estado de la misma en dicho contrato, bastaría con su dirección en la blockchain. También en esa misma función se comprueba que la PCR no se ha actualizado anteriormente, ya que no tiene sentido cambiar el resultado una vez se ha introducido.

En el contrato póliza se establecen restricciones para asegurarse que las IDs de turistas y PCR coinciden como el caso anterior. A su vez, en el propio constructor del contrato se comprueba que los datos referentes a las PCRs previas que deben entregar los turistas para poder viajar no pueden superar los tres días de antigüedad. También, por supuesto, se comprueba que se han introducido todos los datos necesarios y que la fecha de inicio es anterior a la de finalización. A la hora de enviar o no un evento de siniestro e indemnización se comprueba que la fecha de la transacción de actualización a PCR positiva se encuentra dentro del tiempo estipulado del seguro, en caso contrario no se emite dicho evento, ya que se da por hecho que el seguro ha finalizado. La fecha de estas actualizaciones se establece en el código del endpoint de la API, ya que al necesitar el mismo dato en dos contratos diferentes (PCR y Póliza) usar el block.timestamp (que sirve para obtener la hora de la transacción en la blockchain) daría resultados diferentes para los dos contratos, con lo que el dato temporal se les pasa a ambos contratos por igual.

En el contrato general se comprueba que las IDs de los hoteles o las pólizas existen cuando son requeridas para sus funciones.

Capítulo 5 Conclusiones y líneas futuras

5.1 Conclusiones

En este proyecto he tenido la suerte de poder trabajar en una tecnología muy disruptiva que se ha popularizado por las criptomonedas pero que con trabajos como este demuestra que tiene un potencial mucho mayor. En los casos donde la confianza tiene una importancia superior, como en los escenarios de los seguros, alcanza una alta utilidad llegando incluso a sustituir a los intermediarios que se encargan de aportar esta confianza. Gracias a su capacidad criptográfica se puede estar seguro de la identidad de los usuarios que lanzan las transacciones y por otro lado se puede confiar en la propia cadena si se da el ecosistema adecuado, ya que debe detectar cualquier tipo de manipulación sobre la misma mediante sus enlaces de hash criptográficos.

También me gustaría comentar las facilidades que otorga en este tipo de casos de uso donde son los propios smart contracts los que se encargan de dilucidar si se debe llevar a cabo una indemnización, por lo que los clientes pueden saber con certeza las situaciones que desencadenan los pagos. Incluso se puede llegar a automatizarlos debido a que se confía en las decisiones de los contratos inteligentes.

En nuestro caso el proceso se automatiza de tal manera que la aseguradora apenas tiene incidencia en él y se limita a registrar nuevos hoteles en la red y a emitir los pagos cuando le llegan las notificaciones de siniestro (procesos que se podrían automatizar también). Por tanto no sólo el cliente sale beneficiado, sino que todos los participantes de la misma ven reducido el trabajo que deben realizar.

Obviamente tiene también sus partes negativas. Por la propia arquitectura blockchain, las transacciones requieren un tiempo mucho mayor para llevarse a cabo que el mismo proceso en un sistema de base de datos centralizado, ya que tiene que llegar al consenso con los bloques y distribuirlos a todos los nodos de la red. También el espacio de memoria necesario puede ser mayor, pero esto no es un problema tan grave, ya que al tratarse de blockchains privadas están muy acotadas al ámbito al que se dirigen y el espacio de almacenamiento ha dejado de ser un problema grave en la época actual. Incluso se puede integrar nuestra aplicación en otras redes que ya contengan otras aplicaciones y un conjunto de empresas y clientes de las mismas para aumentar la confianza sobre la propia red, ya que al depender más empresas de ella los administradores deben cuidarla más. También el hecho de poseer una mayor antigüedad y un gran conjunto de transacciones pasadas mejora la confianza en la red.

Por otro lado me gustaría destacar la capacidad que tiene esta tecnología para tratar con la identidad soberana y que los datos pertenezcan a su dueño por medio de la seguridad que ofrece la criptografía. Creo que es un paso que se debe dar hacia el futuro, reducir las dependencias con servicios externos y generar una identidad digital propia. Especialmente para que se adecúe de manera sencilla a las leyes que establecen los gobiernos, que hoy en día parece que van a un ritmo distinto al que crece la tecnología. Incluso existen blockchains especializadas en la identidad, como Hyperledger Indy. He tenido la oportunidad de documentarme sobre nuevas tecnologías muy inspiradas por blockchain, que se centran especialmente en la identidad y establecer la capa de seguridad e identidad que no existe en el internet actual eliminando características propias de blockchain que no son necesarias para tratar con estos temas, ya que en sus inicios fue pensada como libro de cuentas destinado a intercambio de divisas. Por tanto puedo afirmar que el potencial de este campo es inmenso y aún queda mucho por descubrir.

Por último se realizó una reunión final de cierre de proyecto entre todos los interesados: miembros del proyecto, miembros de la cátedra Cajasiete y miembros de la Mutua Tinerfeña. Dicha reunión se realizó en la sede de la Mutua en Santa cruz de Tenerife. En ella se presentó el proyecto y se realizó una demostración del prototipo programado para la ocasión. Los participantes de la Mutua, los cuales valoraron positivamente el proyecto, consideraron que se había conseguido llevar a la práctica un caso de uso de seguro paramétrico con blockchain con clara aplicabilidad de la tecnología.

5.2 Líneas futuras

El siguiente paso lógico en el proyecto sería la inclusión de la posibilidad de aumentar el número de hoteles involucrados en la red, ya que en la demostración se ha limitado a uno. Esto se ha debido principalmente a la forma de tratamiento de la privacidad que propone Hyperledger Besu comparada con otras tecnologías. En el caso de seguir usando Besu, habría que crear un nuevo tipo de contrato, que podríamos llamar global, que únicamente fuera privado para el nodo de la aseguradora y en el que se mapean todas las direcciones de los contratos generales por cada hotel que pertenece a la red. No se pueden añadir los contratos generales directamente a este nuevo contrato debido a que poseen grupos de privacidad diferentes (Aseguradora y Aseguradora y Hotel), con lo que mapear a las direcciones bastaría para que la aseguradora itere por todos los contratos generales recuperando todos los datos de las pólizas usando una única función.

Para poder registrar nuevos hoteles se debe añadir un nuevo endpoint en la API Rest que use la propia mutua para registrar nuevos hoteles en la red. Dicha función se encargaría de generar un nuevo nodo en la red y el par de claves para el nuevo hotel, desplegar un contrato general para el nuevo hotel, generar un usuario y contraseña para que dicho hotel pueda usar la interfaz de usuario de la aplicación y añadir su dirección en el mapping del contrato global. La llave de los mappings en el contrato global sería el ID

del Hotel, con lo cual quedaría “Hotel ID -> Dirección contrato general”.

En cambio si se usase otra tecnología de blockchain privada como Quorum, en la que los contratos contienen diferentes datos según quien lo observe se resolvería de manera más sencilla usando un único contrato general y sin necesidad del nuevo contrato global. A su vez se podrían eliminar utilidades como la necesidad del servicio de eventos para transmitir las actualizaciones de PCRs de su contrato a un contrato Póliza, ya que en este caso se podrían integrar directamente los contratos PCR dentro de los contratos Póliza sin depender de que deban tener el mismo grupo de privacidad, lo cual facilitaría enormemente el diseño. Y también a la hora de tratar con los eventos privados, que al usar Besu cualquiera puede interceptarlos y leer sus datos.

Por otro lado, se podría mejorar el rendimiento en los escenarios en los que se realizan transacciones múltiples, actualmente realizadas en serie para garantizar el correcto funcionamiento. Para que se incluyan varias transacciones privadas en el mismo bloque en Besu se deben enviar con distinto “nonce”. Para ello se debe obtener tanto el nonce público y el privado mediante la llave privada de una cuenta para poder usar un nonce diferente (cambiando el privado) para lanzar cada transacción. Ésto reduciría el delay que se produce en los casos en los que se lanzan varias transacciones a la vez, cuyo máximo exponente es cuando se registra una póliza con seis clientes (actualmente el número máximo de turistas por póliza), ya que al crear los contratos PCRs para cada cliente de forma serializada hace que dicha transacción pueda llegar a durar casi 1 minuto, un tiempo que se reduciría al usar este tipo de transacciones en paralelo.

Otro trabajo a futuro interesante sería la integración del servicio realizado dentro de las aplicaciones de la aseguradora que opte por trabajar con nuestro proyecto a la hora de ofrecer este tipo de seguros paramétricos basados en blockchain, pero que lógicamente ha quedado fuera del alcance de este trabajo de fin de grado debido a que la intención era realizar una prueba de concepto para comprobar que se podía llegar a comercializar este tipo de seguros basados en blockchain.

Para finalizar sería interesante la integración dentro de una blockchain de consorcio en la que participen numerosas empresas ofreciendo distintos tipos de productos para aumentar la confiabilidad en la red y mejorar la seguridad de la misma.

Capítulo 6 Summary and Conclusions

6.1 Conclusions

In this project I have been fortunate to be able to work on a very disruptive technology that has become popular with cryptocurrencies but that with projects like this one, shows that it has much greater potential. In cases where trust is of higher importance, such as in insurance scenarios, it achieves a high utility, even replacing the intermediaries, who are responsible for providing the trust, because due to their cryptographic capacity it is possible to be sure of the identity of the users who send the transactions and, on the other hand, the chain itself can be trusted if the appropriate ecosystem is given, since it must detect any type of manipulation on it through its cryptographical links.

I would also like to comment on the facilities it provides in this type of use case where the smart contracts themselves are in charge of elucidating whether compensation should be carried out, so that clients can know with certainty the situations that would trigger the payments and this payment can even be automated because smart contract decisions are trusted and therefore the risk of fraud is reduced.

In our case, the process is automated in such a way that the insurer has little effect on it and is limited to registering new hotels on the network (a process that could also be automated) and issuing payments when claim notifications arrive. Therefore, not only the client benefits, but all the participants see the work they have to do reduced.

Obviously it also has its negative parts. Due to the blockchain architecture itself, transactions require a much longer time to be carried out than the same process in a centralized database system, since it has to reach consensus with the blocks and distribute them to all the nodes of the network. Also the memory space required may be greater, but this is not such a serious problem, since as they are private blockchains they are very limited to the area to which they are directed and although our application is combined in other networks that already contain other applications and a set of companies and their clients to increase confidence in the network itself, as more companies depend on it, administrators must take care of it to a greater extent, storage space is no longer a serious problem in the current era.

On the other hand, I would like to highlight the ability of this technology to deal with sovereign identity and that the data belongs to its owner through the security offered by cryptography. I believe that it is a step that must be taken towards the future, reducing dependencies with external services and generating a digital identity of its own. Especially so that it adapts in a simple way to the laws established by governments, which today

seem to be going at a different rate than technology grows. There are even blockchains specialized in identity, like Hyperledger Indy. I have had the opportunity to document myself on new technologies very inspired by blockchain, which are especially focused on identity and establish the layer of security and identity that does not exist in the current internet, eliminating characteristics of blockchain that are not necessary to deal with these issues , since in its beginnings it was thought as an account book for currency exchange. Therefore I can say that the potential of this field is immense and there is still much to discover.

Finally, a final project closing meeting was held among all stakeholders: project members, members of the Cajasiete cathedra and members of the Mutua Tinerfeña. This meeting was held at the Mutua headquarters in Santa Cruz de Tenerife. In it the project was presented and a demonstration of the prototype programmed for the occasion was carried out. The participants of Mutual Tienerfeña, who positively valued the project, considered that a case of parametric insurance with blockchain with clear applicability of the technology had been put into practice.

6.2 Future works

The next logical step in the project would be the inclusion of the possibility of increasing the number of hotels involved in the network, since in the demonstration it has been limited to one. This has been mainly due to the way of treatment of privacy that Hyperledger Besu proposes compared to other technologies. In the case of continuing to use Besu, a new type of contract would have to be created, which we could call global, which would only be private for the insurer's node and in which all the addresses of the general contracts are mapped for each hotel that it belongs to. to network. General contracts cannot be added directly to this new contract because they have different privacy groups (Insurer and Insurer and Hotel), so mapping the addresses would be enough for the insurer to iterate through all the general contracts recovering all the data of policies using a single function.

In order to register new hotels, a new endpoint must be added to the Rest API that uses the mutual itself to register new hotels on the network. This function would be in charge of generating a new node in the network and the key pair for the new hotel, deploying a general contract for the new hotel, generating a username and password so that said hotel can use the application's user interface, and adding its address in the global contract mapping. The key to the mappings in the global contract would be the Hotel ID, which would result in “Hotel ID -> General contract address”.

On the other hand, if another private blockchain technology such as Quorum were used, in which the contracts contain different data depending on who observes it, it would be resolved more easily using a single general contract and without the need for the new global contract. At the same time, utilities could be eliminated, such as the need for the event service to transmit the updates of PCRs from their contract to a Policy contract, since in this case the PCR contracts could be directly integrated into the Policy contracts without depending on whether they must have the same privacy group, which would

greatly facilitate the design. And also when dealing with private events, by using Besu anyone can intercept them and read their data.

On the other hand, the speed of transactions could be improved in scenarios in which multiple transactions are carried out, currently carried out in series to guarantee correct operation. In order for several private transactions to be included in the same block, Besu must send them with a different “nonce”. To do this, both the public and private nonce must be obtained through the private key of an account to be able to use a different nonce (changing the private one) to launch each transaction. This would reduce the delay that occurs in cases in which several transactions are launched at the same time, whose maximum exponent is when a policy is registered with six clients (currently the maximum number of tourists per policy), since when creating the contracts PCRs for each client in a serialized way means that said transaction can last almost 1 minute, a time that would be reduced when using this type of transactions in parallel.

Another interesting future work would be the integration of the service carried out within the applications of the insurer that chooses to work with our project when offering this type of parametric insurance based on blockchain, but which has logically been outside the scope of this work end-of-degree because the intention was to carry out a proof of concept to verify that this type of blockchain-based insurance could be marketed.

Finally, it would be interesting to integrate within a consortium blockchain in which numerous companies participate offering different types of products to increase network reliability and improve network security.

Capítulo 7 Presupuesto

7.1 Tabla de Tipos

Tipos	Descripción
Producción de la Aplicación	Coste derivado de la programación de la aplicación por parte del equipo de programadores que la han llevado a cabo. Entre ellos un programador Junior, un programador semi senior y un programador senior y jefe de proyecto. Duración de 3 meses.
Mantenimiento de la Red. Aseguradora.	Coste derivado del mantenimiento de la red blockchain que usa la aplicación durante 1 año. Presupuestado para usar el servicio Amazon Web Service, que cobra por horas el mantenimiento de los nodos y una pequeña parte por transacciones. Presupuesto para 2 nodos de la aseguradora (redundancia), los nodos de hotel y laboratorios entrarían por separado. En caso de que se uniera a una blockchain de consorcio serían los gastos derivados de la pertenencia a dicha red.
Mantenimiento de la Red. Participantes.	Debido a que en el prototipo contamos únicamente con un Hotel y un Laboratorio añadiremos el coste de contar con estos dos nodos en la misma plataforma (AWS).
Integración en la Aseguradora	Coste derivado de integrar la aplicación en el ecosistema de la aseguradora interesada en comercializar este seguro.
Formación de	Incluye el coste destinado a la

programadores	formación del equipo de empleados de la aseguradora que opte por comercializar el seguro. Está presupuestado para dos grupos de entre 15-20 participantes durante 2 semanas.

Figura 10. Tabla de tipos.

7.2 Presupuesto Estimado

Tipos	Unidad de Medida	Coste/Unidad	Nº Unidades	Coste Estimado
1. Personal Propio en el Proyecto				
Programador Senior y jefe de proyecto	Coste hora	60.00 €	240	14.400 €
Programador semi Senior	Coste hora	40.00 €	240	9.600 €
Programador Junior	Coste hora	20.00 €	240	4.800 €
Total 1				28.800 €
2. Mantenimientos de Red Blockchain en la nube				
Mantenimiento de la red. Aseguradora (1 año)	Coste hora	0.676 €	8760	5.922 €
Mantenimiento de la red. Hotel y Laboratorio. (1 año)	Coste hora	0.676 €	8760	5.922 €
Total 2				11.844 €
3. Integración en la Aseguradora				
Integrador de aplicaciones	Coste hora	60	100	6.000 €
Total 3				6.000 €
4. Formación de Personal de Aseguradora				

Formador para SPC-19	Coste hora	40	120	4.800 €
Total 4				4.800 €
Imprevistos (10%)				5.144 €
TOTAL PRESUPUESTO				56.588 €

Figura 11. Presupuesto detallado del proyecto.

El precio de mantenimiento de los años posteriores [Figura 11] aumentaría debido a la cantidad de nodos Hotel y Laboratorio incluidos en la red (el precio del presupuesto de AWS equivale a dos nodos por año) y a las posibles modificaciones requeridas en el código para adaptarse a nuevos casos de uso.

Bibliografía

[1]. Anshuman Kalla et al. *"The Role of Blockchain to Fight Against COVID-19",* IEEE Engineering management review, Vol. 48, NO. 3, Third Quarter. September. 2020.

[2]. Mihalis Kristikos, *"Ten technologies to fight coronavirus",* European Parliamentary Research Service, April. 2020.

[3]. Park Daehyeon, Doojin Ryu. *"Blockchain in Health Insurance: Sharing Medical Information and Preventing Insurance Fraud",* Korean Journal of Financial Studies, August. 2019.

[4]. Maganahalli, Samarth. *"Blockchain: The Future of Insurance"*. International Journal for Research in Applied Science and Engineering Technology. 8. 1450-1454. 10.22214/ijraset.2020.5234, 2020.

[5]. Valentina Gatteschi et al. *"Blockchain and Smart Contracts for Insurance: Is the Technology Mature Enough?",* Future Internet. February. 2018. [online] Available: https://www.researchgate.net/publication/323298791_Blockchain_and_Smart_Contracts_for_Insurance_Is_the_Technology_Mature_Enough

[6]. Lamberti, F., Gatteschi, V., Demartini, C., Pranteda, C., Santamaria, V. "*Blockchains can work for car insurance: using smart contracts and sensors to provide on-demand coverage".* IEEE Consum. Electron. Mag. 7 (4), 72-81. 2018.

[7]. Bertani, T.; Butkute, K.; Canessa, F. "Smart Flight Insurance—InsurETH". December. 2017. [online] Available: http://mkvd.s3.amazonaws.com/apps/InsurEth.pdf

[8]. Pagano, A.J.; Romagnoli, F.; Vannucci, E. "*Implementation of Blockchain Technology in Insurance Contracts Against Natural Hazards: A Methodological Multi-Disciplinary Approach*". Environ. Clim. Technol. 2019, 23, 211–229.

[9]. Arpan Kumar Kar, L. Navin, "*Diffusion of blockchain in insurance industry: An analysis through the review of academic and trade literature",* ELSEVIER Telematics and Informatic, November. 2020

[10]. Dogan Kaleli. *"How #Blockchain is Helping the Insurance Industry to Transform".* The Future of Insurance - Linkedin's Newsletter Series. April. 2021. [online] Available: https://www.linkedin.com/pulse/how-blockchain-helping-insurance-industry-transform-dogan-kaleli/

[11]. Davies, S. Bitcoin: *"Possible bane of the diamond thief"*. June. 2016. [online] Available: http://www.ft.com/cms/s/0/

f2b0b2ee-9012-11e4-a0e5-00144feabdc0.html#axzz4DAQsiRry

[12]. Civic. [online] Available: https://tokensale.civic.com/CivicTokenSaleWhitePaper.pdf

[13]. KYC legal. [online] Available: https://icoholder.com/es/kyc-legal-ico-16010

[14]. A. Fusco, G. Dicuonzo, V. Dell'Atti, M. Tatullo. *"Blockchain in Healthcare: insights on COVID-10".* International Journal of Environmental Research and Public Health. September. 2020.

[15]. Kuo, T.T.; Kim, H.E.; Ohno-Machado, L. *"Blockchain distributed ledger technologies for biomedical and health care applications"*. J. Am. Med. Inform. Assoc. 2017, 24, 1211–1220.

[16]. Mackey, T.K.; Kuo, T.T.; Gummadi, B.; Clauson, K.A.; Grishin, G.C.D.; Obbad, K.; Barkovich, R.; Palombini, M. *"'Fit-for-purpose?'—Challenges and opportunities for applications of blockchain technology in the future of healthcare"*. BMC Med. 2019, 17, 68.

[17]. Clauson, K.A.; Breeden, E.A.; Davidson, C.; Mackey, T.K. *"Leveraging Blockchain Technology to Enhance Supply Chain Management in Healthcare: An Exploration of Challenges and Opportunities in the Health Supply Chain"*. Blockchain Healthcare Today 2018, 1, 1–12.

[18]. Angeles, R. *"Blockchain-Based Healthcare: Three Successful Proof-of-Concept Pilots Worth Considering"*. J. Inf. Technol. Manag. 2018, 27, 4.

[19]. Siyal, A.A.; Junejo, A.Z.; Zawish, M.; Ahmed, K.; Khalil, A.; Soursou, G. *"Applications of Blockchain Technology in Medicine and Healthcare: Challenges and Future Perspectives"*. Cryptography 2019, 3, 3. [online] Available: https://www.mdpi.com/2410-387X/3/1/3

[20]. https://github.com/pouladzade/Seriality

[21]. Xiwei Xu et al. *"The Blockchain as a Software Connector",* Working IEEE/IFIP Conference of Software Architecture. 2016.