UNIVERSIDAD DE LA REPÚBLICA

FACULTAD DE INGENIERÍA

Instituto de Computación - InCo

PROYECTO DE GRADO

Uso de blockchain en la administración pública

Autores:

Eduardo Mereles

Juan Ortellado

Tutor:

Javier Barreiro

Montevideo, Uruguay

2019

RESUMEN

Blockchain es una tecnología emergente que ha cobrado gran notoriedad en los últimos años en relación a su uso en criptomonedas como Bitcoin. Esto ha provocado que se analicen en profundidad sus funcionalidades y se evalúe su aplicabilidad en otros contextos.

Esta tecnología presenta una serie de características que dan indicios de aplicabilidad en escenarios que poco tienen que ver con criptomonedas, como lo son la descentralización en escenarios donde no hay confianza entre los participantes, escenarios donde es importante la inmutabilidad de la información agregada, escenarios donde la información no se pueda corromper, o en definitiva cualquier combinación de estos tres.

El objetivo de este proyecto fue investigar la tecnología, así como herramientas que permitan trabajar con blockchain y a través de ellas evaluar su aplicabilidad a un caso de uso en la administración pública. Se investigó Bitcoin, dado que es la primera implementación en producción y habiendo comprendido los conceptos expuestos por la criptomoneda, se investigó la tecnología abstrayéndose de la misma. Luego el foco se puso en estudiar frameworks de desarrollo para blockchain, analizando las prestaciones ofrecidas en relación a los distintos escenarios posibles donde aplicarlo. Por último, y en base al conocimiento adquirido, se relevaron casos de uso posibles en dónde aplicar la tecnología blockchain en el entorno de la administración pública. Sobre los casos relevados, se consultaron distintos involucrados para entender más en profundidad cada una de las realidades y de esa forma buscar el mejor escenario para el proyecto. Este proceso permitió seleccionar un caso de uso particular, que fue sometido a dos cuestionarios genéricos que asisten a quien lo usa para entender si el caso puede o no ser resuelto con blockchain, y en caso de que el resultado sea afirmativo, da indicios de qué tipo de blockchain se debería utilizar.

En base al análisis previo, se eligió uno de los casos de uso para el desarrollo de una aplicación sobre la tecnología blockchain, el cual implica una mejora en relación a la solución actual. Adicionalmente, el desarrollo fue realizado pensando en que este caso pueda ser extendido fácilmente, dado que fueron encontradas otras funcionalidades en las que también sería de gran valor el uso de blockchain, pero se optó por no implementarlo debido al alcance del proyecto.

Palabras clave: blockchain, Hyperledger, cadena de bloques.

Contenido

1. Introducción 7
1.1. Objetivos 8
1.2. Organización del documento 8
2. Marco Conceptual 11
2.1 Blockchain 11
2.1.1. Conceptos de blockchain 13
2.1.2. Tipos de blockchain 15
2.1.3. Integridad 18
2.1.4. Smart Contracts 21
2.1.5 Otros componentes 22
2.2. Bitcoin 23
3. Casos de aplicación 25
3.1. Situación actual 25
3.2. Casos de uso en ámbito público 28
4. Plataformas de blockchain 33
4.1 Plataformas investigadas 34
4.1.1 Ethereum 34
4.1.2 Corda 34
4.1.3 Credits 35
4.2 Hyperledger Fabric 35
5. Caso de uso en administración pública 45
5.1. Caso de uso: empadronamiento vehicular 45
5.2. Validación de aplicabilidad 45
5.2.1. Primera validación 46

5.2.1. Segunda validación 48
6. Arquitectura de la solución 51
6.1. Vista lógica 51
6.2. Vista física 56
6.3. Escenario 57
7. Implementación 59
7.1. Tecnologías utilizadas 59
7.2. Aspectos de la implementación 60
7.3. Dificultades y limitaciones encontradas 63
8. Gestión del proyecto 67
8.1. Organización 67
8.2. Planificación y ejecución 67
9. Conclusiones y trabajo a futuro 71
Referencias 73
GLOSARIO 79
APÉNDICE A - Instalación de Hyperledger Fabric 83
APÉNDICE B - Instalación de la aplicación 85
APÉNDICE C - Cambio versión Hyperledger 89
ANEXO 91

1. Introducción

Aproximadamente en el año 2009 surge la criptomoneda llamada Bitcoin [1], a través de la cual también se le da origen a la tecnología hoy conocida como blockchain. En los años subsiguientes, fueron creadas muchas otras criptomonedas emulando el funcionamiento de Bitcoin y a pesar de su gran uso en este contexto, blockchain es observado como una tecnología revolucionaria por sus características y posible aplicabilidad en muchos otros contextos.

Muchas plataformas o servicios que se ofrecen en la actualidad dependen de un tercero de confianza para poder funcionar con las garantías que los usuarios exigen, como por ejemplo las aplicaciones bancarias. Aquí los usuarios que desean transferir dinero entre ellos, delegan en los bancos el control sobre los saldos, confiando en que estos realizarán correctamente las transferencias. Este tipo de requerimientos, hace que se centralice el control y el punto de falla del sistema y por tanto se convierta en un componente crítico para la solución. Esta situación, se ve alentada por la intervención de actores con malas intenciones, que encuentran vulnerabilidades y sacan provecho de ellas, afectando en algunos casos a todo el sistema informático.

Esta situación ha despertado el interés de la aplicación de la tecnología blockchain por parte de múltiples sectores, tales como la salud, gestión de mercadería y propiedades, almacenamiento de información, verificación de identidad y administración pública, entre otros, por lo que han empezado a evaluar e implementar distintos casos de aplicación [2].

En el gobierno uruguayo, AGESIC es una institución que lidera el desarrollo del gobierno digital, la sociedad de la información y el conocimiento, desde Presidencia de la República. Es responsable por el desarrollo de la política digital del Estado y del Uruguay Digital. Algunos de los objetivos estratégicos son impulsar el gobierno abierto, facilitar el relacionamiento con el Estado, integrar digitalmente a los distintos organismos, fortalecer el ecosistema de ciberseguridad, simplificar los trámites mejorando los servicios, contribuir con la alfabetización digital, aportar soluciones informáticas innovadoras para mejorar los servicios, entre otros. A través del plan de gobierno digital 2020 [3] propuesto por AGESIC uno de los medios a través de los cuales se pretende alcanzar los objetivos es facilitando instrumentos normativos, institucionales y técnicos que habiliten el uso de nuevas tecnologías para la validez de registros, como blockchain.

De esta forma es que surge el interés de investigar sus características y si realmente es aplicable en otros contextos, en particular en la administración pública.

1.1. Objetivos

El objetivo general del proyecto es evaluar la aplicabilidad de la tecnología blockchain en la administración pública e implementar una prueba de concepto donde se pueda apreciar esa aplicabilidad. Para cumplir con este objetivo, es que se subdivide en los siguientes objetivos específicos:

- Entender el funcionamiento de la tecnología y los escenarios donde aporta valor aplicarlo.
- Investigar herramientas disponibles que existen para implementar los distintos tipos de blockchain.
- Evaluar casos de uso en la administración pública, en los que sea de utilidad aplicar una solución basada en la tecnología blockchain.
- Elegir un caso de uso de los evaluados anteriormente e implementar una prueba de concepto.

1.2. Organización del documento

El documento se encuentra dividido en 9 capítulos, los cuales se detallarán a continuación.

En el capítulo 2 se presenta el marco conceptual, donde se desarrollan los conceptos básicos para comprender qué es blockchain y cómo funciona, detallando componentes, su utilidad y otros conceptos fuertemente relacionados con la tecnología.

En el capítulo 3 se puede encontrar una descripción de los casos donde existe una mejor aplicabilidad y ejemplos de contextos en los que se está aplicando o se encuentra en fases de pruebas, así como también casos en la administración pública en los que se entiende es aplicable.

En el capítulo 4 se presentan algunas de las herramientas utilizadas actualmente para implementar blockchain, así como la elección de una herramienta específica para el desarrollo, profundizando en aspectos técnicos y limitantes.

En los capítulos 5 y 6 se presenta el caso de uso elegido sobre el cual se realizó la prueba de concepto y la arquitectura definida para su implementación.

El capítulo 7 hace referencia a la implementación y allí se desarrollan las tecnologías utilizadas para poder realizar dicha implementación y las decisiones tomadas.

En el capítulo 8 se desarrolla todo lo referente a la gestión del proyecto, donde se puede encontrar principalmente lo planificado para su ejecución y las situaciones que generaron desvíos.

En el capítulo 9 se expone el trabajo a futuro y las conclusiones tanto de la tecnología en sí, como de la prueba de concepto realizada.

2. Marco Conceptual

En este capítulo se profundizará en cada uno de los aspectos que se entienden relevantes para la comprensión de la tecnología blockchain, analizando su funcionamiento, su utilidad y los componentes necesarios para su creación. Esto será de utilidad para comprender el análisis e implementación realizado sobre el caso de uso. Adicionalmente se describen problemáticas y aspectos técnicos relevantes para blockchain, así como también se mostrará una categorización realizada en base a las características que se pueden encontrar en la tecnología.

2.1 Blockchain

La idea principal que se introduce con la tecnología blockchain es la de eliminar a los intermediarios en las transacciones, pero esto supone un problema mayor, ya que estos intermediarios existen en gran medida para brindar confianza a las partes involucradas y al no estar, podrían generarse hechos maliciosos entre ellos, como por ejemplo el ataque gasto doble. Se define gasto doble [4] [5] como el ataque del uso acertado de los mismos medios dos veces. Para explicar este concepto, supongamos que entregamos a otra persona o entidad un objeto como por ejemplo una manzana, la propiedad de ese objeto deja de pertenecernos y pasa a esa persona o entidad y solo podremos recuperarlo si nos lo devuelve voluntariamente o por la fuerza, pero no podremos gastar ese objeto físico de nuevo si no lo recuperamos. En este sentido, el problema con los ficheros digitales es que se pueden reproducir infinitamente, como por ejemplo un documento PDF y por este motivo, los sistemas digitales tienden a ser centralizados, de forma de garantizar que nadie más pueda generar copias digitales de los activos con los que se trabaja.

Como solución al ataque de gasto doble, se propone que el control realizado por los intermediarios se lleve a cabo de forma distribuida, pero esto introduce un nuevo problema. Para realizar este control, los datos deben de estar distribuidos, lo cual genera la necesidad de asegurar la incorruptibilidad de los mismos. Esta problemática es atacada utilizando la técnica propuesta por Habert y Stornetta en 1991, en su investigación "How to time-stamp a digital document" [6]. Esta plantea cómo se podrían patentar documentos digitales, archivos de audio, video, etc, de forma que estas patentes no puedan alterarse. La investigación menciona que no basta con agregar una fecha de creado o modificado, pues esto no es confiable y puede ser modificado fácilmente con otras intenciones y plantea que para resolverlo se debe utilizar una función hash, que a partir de la información que se quiere patentar, genera un código único llamado hash. Esto implica que, ante el cambio más mínimo en la entrada, el código generado se vea modificado, dando como resultado los primeros fundamentos de la tecnología blockchain.

En el año 2008, surge un documento de investigación cuyo nombre es “Bitcoin: A peer to peer electronic cash system” [1], el cual toma como insumo el concepto mencionado en el párrafo anterior. Un año más tarde, luego del desarrollo realizado por su autor en conjunto con otros colaboradores, se realiza la puesta en producción del sistema y es lo que hoy se conoce como Bitcoin, la primera moneda digital descentralizada. En la sección “2.2. Bitcoin” se profundizará en el concepto, ya que es fundamental en la investigación para la definición de blockchain.

Por tanto, blockchain [7] se puede definir como un registro, un libro mayor de acontecimientos digitales que está distribuido o es compartido entre muchas partes diferentes, y solo puede ser actualizado a partir del consenso de los participantes del sistema y, una vez introducida, la información nunca podrá ser borrada.

Profundizando en las características [8] de la tecnología blockchain, la primera que se encuentra es que la información no puede ser borrada fácilmente. Esto es debido a que la información se encuentra replicada en cada uno de los participantes y en caso de que uno de ellos sufra un ataque, siempre existirá otro participante que tenga una copia de la información. La única forma en que se podría alcanzar ese objetivo, sería realizando un ataque simultáneo a todos los participantes, lo cual acompañado de medidas básicas de seguridad, como por ejemplo el uso de certificados digitales, usuario y contraseña, lo vuelve una tarea que requiere un gran poder computacional para poder ser llevada a cabo. Claramente, a medida que se aumenta en número de participantes más difícil se vuelve esta tarea.

La segunda característica está relacionada con la mencionada anteriormente y hace referencia a la incorruptibilidad de la información. Si bien en parte la forma de protegerse contra este tipo de problemas viene dado por la replicación de la información, aún quedan situaciones en las que se podría llegar a vulnerar lo registrado. Para protegerse de esas situaciones, es que se adapta la propuesta de Haber y Stornetta, haciendo uso de hashes para proteger la información guardada en la blockchain, como será explicado en la sección “2.1.5. Componentes”.

La tercera característica es que la información es actualizada a partir del consenso de los participantes de la red. Esta característica juega un rol fundamental en el proceso de aprobación o rechazo de la información a agregar en la blockchain y es el mecanismo que se utiliza para sustituir al intermediario que brinda confianza en la red. La idea detrás del consenso es que la propia red se defienda de los participantes maliciosos o agentes externos que intentan agregar información falsa, bajo la hipótesis de que la mayoría de los participantes estará en contra de esos actos. Cabe aclarar que dependiendo de las necesidades del negocio, puede variar la definición de consenso para ese escenario.

En base a estas características se puede identificar que esta tecnología aplica a problemáticas de coordinación, confiabilidad y seguridad de datos entre múltiples actores y sin intermediarios, asegurando tener en cada actor una copia fiel y ordenada de los mismos. Ejemplo de estos casos son [2]: moneda digital, registro catastral, voto electrónico, manejo de identidad, trazabilidad en cadena de suministros, salud, sistema tributario, administración de títulos, etc.

2.1.1. Conceptos de blockchain

Los componentes [7] [9] de la blockchain básicamente son tres: nodos, transacciones y bloques. A modo de ejemplo, se representa la blockchain como un libro de cuentas, donde los bloques se corresponden a hojas de ese libro, las transacciones párrafos de cada hoja y los nodos cada lugar del mundo donde se encuentra una copia de ese libro. En la *figura 2.1* [10] se muestra un diagrama ilustrativo de los componentes antes mencionados.

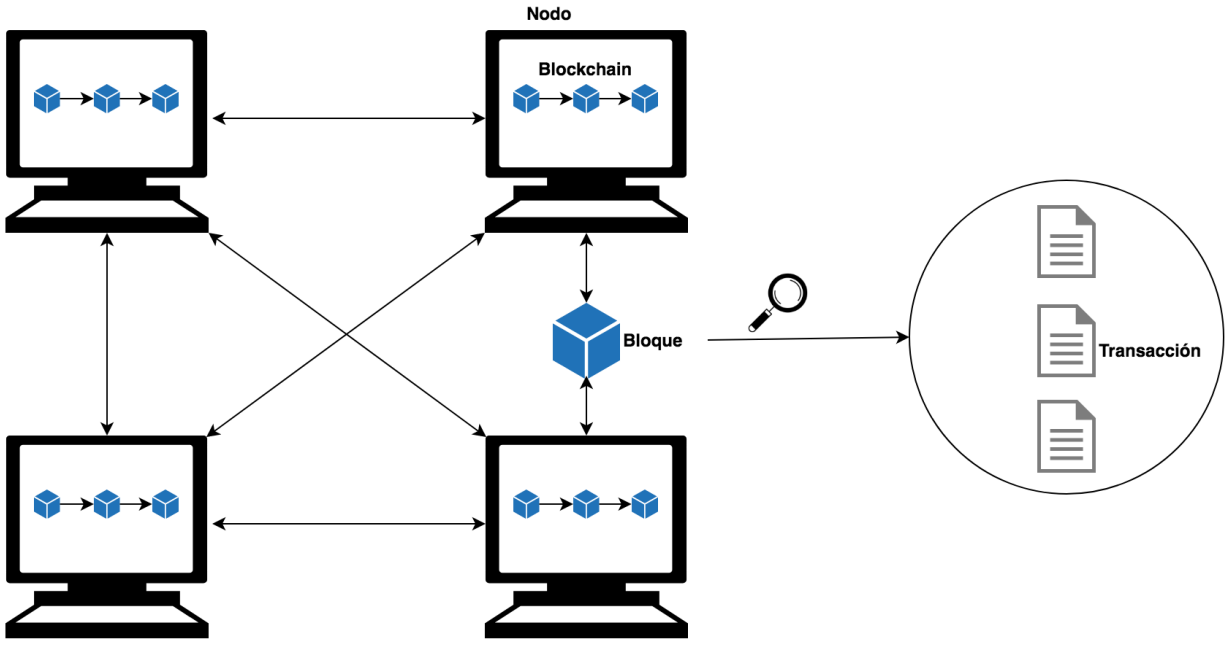

Figura 2.1 - Ilustración de componentes de blockchain

En algunos casos es posible encontrar otros componentes como lo son autoridades certificadoras u oráculos y su uso varía dependiendo del tipo de blockchain sobre el cual se esté trabajando y la herramienta que se esté utilizando para su implementación. A continuación, se define y profundiza en cada uno de los componentes antes mencionados.

Transacciones

Una transacción [9] [11] es un mensaje firmado digitalmente que autoriza alguna acción particular asociada a la red blockchain. Para el caso de las criptomonedas, el tipo de transacción dominante son las que involucran por lo menos a dos partes y están asociadas al intercambio de bienes o servicios a cambio del capital correspondiente. A modo de ejemplo, supongamos una red blockchain compuesta por bancos, una transacción podría ser representada por una transferencia de un monto de dinero de un banco a otro, o bien podría representarse como suscripción de un nuevo cliente en uno de los bancos.

De forma que los participantes de la red puedan leer y si corresponde validar la veracidad de las transacciones, las mismas no se encuentran encriptadas y los datos que conforman una transacción varían dependiendo de la plataforma sobre la cual se trabaje. De acuerdo a lo investigado se pueden identificar tres elementos que siempre se encuentran presentes, el identificador de la transacción, una marca de tiempo de cuando es ingresada la transacción a la red y un identificador de quién es el emisor de la transacción. Adicionalmente dependiendo de la implementación se pueden incluir otros elementos como: costo de realizar la transacción en la criptomoneda asociada, número de bloque al cual pertenece, estado de transacción, conjunto de datos leídos y escritos al ejecutar la transacción, firma digital del origen de la transacción y firmas digitales de los participantes validadores.

Bloque

Los bloques [11] son un conjunto de registros que representan la información de la blockchain. Un bloque está formado por un conjunto de transacciones confirmadas e información adicional que se ha incluido en la blockchain. Al igual que sucede con las transacciones, dependiendo de la plataforma sobre la que se trabaje, puede variar la información que lleva el bloque, pero en líneas generales algunos de estos campos se mantienen. Estos campos que forman parte de cualquier bloque de la cadena (excepto el bloque generatriz, que inicia la cadena) son:

1. Un hash que enlaza con el bloque anterior
2. El conjunto de transacciones que incluye (cuyo número viene determinado por diferentes factores)
3. Marca de tiempo de cuando se genero el bloque
4. Otro hash que enlazará con el siguiente bloque.

A continuación, se muestra la *figura 2.2* [7] ilustrativa de cómo se vería una cadena de bloques en construcción.

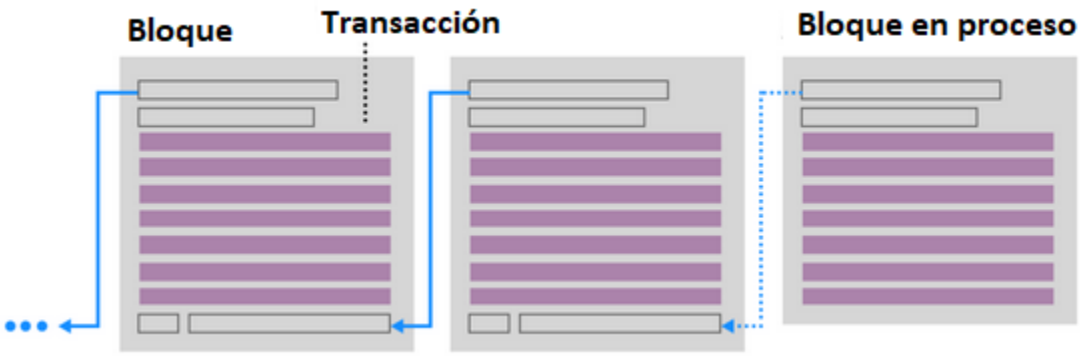

Figura 2.2 - Estructura de bloques

El bloque en proceso o en construcción, es el siguiente bloque a ser ingresado en la blockchain, en este se ingresan las nuevas transacciones a ser agregadas. Antes de agregar este bloque en la blockchain, se debe obtener mediante cálculos el cuarto punto anteriormente indicado. Este punto refiere a un hash que es calculado a partir de los datos del bloque en progreso junto con el hash del bloque anterior, provocando que si algún dato en ese bloque o el anterior es alterado, el hash generado no coincidirá con el original. De esta forma es posible detectar si un bloque fue corrompido y asegurar la detección de cambios en los bloques.

Nodo

Un nodo es un ordenador conectado a la red blockchain utilizando un software que almacena y distribuye una copia de la información de la blockchain en tiempo real. Conceptualmente este nodo puede representar una organización, como por ejemplo una empresa o un organismo del estado. Dependiendo del tipo de blockchain y los privilegios que tenga el nodo, éste podrá validar bloques y transacciones, agregar transacciones a un bloque y añadir bloques a la blockchain. Cada vez que un bloque es confirmado y se añade a la cadena, es comunicado a todos los nodos y este se añade a la copia que cada uno almacena. Cada nodo mantiene viva la plataforma blockchain, por lo que cuantos más nodos haya en la plataforma, más copias habrá y por tanto más segura estará la información en la blockchain.

A continuación, se identifican los distintos tipos de blockchain destacando sus distintas características.

2.1.2. Tipos de blockchain

Dependiendo de las necesidades del caso de uso donde se va a aplicar, se pueden presentar distintos requerimientos para los participantes de la red blockchain. A modo de ejemplo, una institución privada como un banco podría querer utilizar la tecnología restringiendo el acceso a ciertos participantes, otorgando permisos a quienes son de interés para el banco. Profundizando en el ejemplo, también podría requerir que

algunos de los que pueden acceder solo puedan leer y el resto agregar información. Este tipo de situaciones presentan la necesidad de categorizar los distintos tipos de tecnología blockchain [12]que pueden ser construidos.

Una primera categorización para las blockchain puede ser en tres tipos distintos, públicas, federadas y privadas, donde éstas refieren a los permisos de escritura y lectura que pueden tener los nodos sobre los datos, aunque además refieren a la relación que tienen los participantes. A continuación, se detalla cada una de estas.

Blockchain pública [12]

Las blockchains públicas, como lo indica su nombre se encuentran abiertas al mundo, por lo que cualquier interesado que desee, puede unirse. Además, todos pueden copiar los datos de la misma, leer o escribir nuevas transacciones y comenzar a participar del proceso de validación de transacciones. Dado que en este tipo de blockchains todos los participantes validan transacciones, se deben utilizar mecanismos para prevenir posibles alteraciones de la información. Para protegerse de ello, se recomienda que al momento de agregar nueva información a la blockchain, se haga uso de algoritmos de consenso como proof of work o proof of stake, los cuales serán mencionados en la sección “2.1.3. Integridad”. Es importante destacar que la utilización de estos algoritmos presenta problemas de performance, ya que estos se basan en realizar cálculos que implican un gran costo computacional y que terminan generando demoras a la hora de agregar nueva información, como se explicará en la sección mencionada anteriormente.

Blockchain federada [12]

Por otra parte, las blockchains federadas operan bajo el control de un grupo, estas no permiten que todos los miembros de la red blockchain participen del proceso de validación. El derecho a leer de la blockchain puede ser público o restringido a los participantes. Dado que en estas blockchains los nodos que validan las transacciones son un conjunto reducido e identificado, generalmente utilizan certificados para firmar sus transacciones, por lo que el ingreso de una transacción se realiza de forma ágil. Si bien esto reduce el costo de transacción, éste no es disruptivo.

Este tipo de blockchain se ajusta fácilmente al sector bancario, dado que en este caso los bancos son las únicas autoridades capaces de validar las transacciones generadas por los usuarios. Por ejemplo, imaginemos un conjunto federado de 15 instituciones financieras, donde cada una opera un nodo y 10 de ellas deben firmar cada bloque para que sea válido.

Existen argumentaciones que dicen que este tipo de sistemas no puede ser definido como blockchain, pero la tecnología aún se encuentra en sus primeras etapas y no es claro cómo será adoptado.

Blockchain privada [12]

Las blockchains privadas son las que pertenecen a una única organización, en donde todos los participantes están plenamente identificados. Pero a diferencia de las anteriores, el derecho a leer de la blockchain es restringido únicamente a los participantes. Al igual que en las blockchains federadas, al estar los nodos identificados se suelen firmar las transacciones con certificados, por lo que el ingreso de las mismas también se hace de forma ágil, lo que reduce el costo de transacción, aunque este no es disruptivo. Este tipo de blockchain son ideales para el manejo de los procesos internos de las organizaciones, como por ejemplo datos de stock, lista de precios por cliente y demás datos internos y privados.

Resumiendo, estos tipos de blockchains, en la *figura 2.3 s*e presenta una tabla comparativa de aspectos como son el acceso, velocidad, seguridad e identidad.

	Publica	Privada	Federada
Acceso	Lectura / Escritura libre	Lectura / Escritura libre (para los miembros de la red)	Lectura / Escritura solo para los nodos autorizados. El resto de los participantes solo lectura.
Velocidad	Lento	Rápido	Rápido
Seguridad	Proof of Work, Proof of Stake, Otros mecanismos de consenso	Participantes autorizados	Participantes autorizados
Identidad	Anonimos, Pseudoanonimos	Identidades conocidas	Identidades conocidas

Figura 2.3: Tabla comparativa de tipos de blockchain

Otro tipo de categorización utilizado generalmente es en blockchain permisionadas [13] y no permisionadas [13], donde estas solo refieren a los permisos de escritura, lectura y validación de las transacciones.

Blockchain no permisionadas

En este tipo de blockchain cualquier participante puede unirse a la red sin restricciones, además los mismos pueden leer, escribir y verificar transacciones. La validez de las transacciones es establecida mediante un algoritmo de consenso y, por otra parte, aunque las transacciones pueden ser leídas por cualquier participante, es posible ocultar información sensible, mediante un algoritmo de cifrado de información, como por ejemplo AES-256 [14].

Blockchain permisionadas

Para este tipo de blockchain hay una autoridad central encargada de otorgar permisos de lectura, escritura y verificación de transacciones a los distintos participantes. Dado que el permiso de escritura es otorgado solo a participantes de confianza, se hace posible que el algoritmo de consenso sea más simple y eficiente. Al igual que en las blockchains no permisionadas, el acceso de lectura puede ser público.

2.1.3. Integridad

Como se vio en la sección “2.1.1. Conceptos de blockchain” la tecnología blockchain utiliza bloques para agrupar la información que allí se agrega, asegurando la integridad mediante la utilización de hashes. Existen distintas implementaciones de la tecnología blockchain, las cuales implementan distintos mecanismos para garantizar la integridad de sus datos. Es importante destacar que la dificultad de violar la integridad no es solamente debido a la modificación de los datos y la generación de un nuevo hash en base a esas modificaciones, esto último es necesario por lo mencionado respecto al proceso de construcción. Para que tenga probabilidades, los subsiguientes bloques también deben ser modificados de la misma forma y esta tarea debe ser llevada a cabo en todos los nodos de la red. Es por este motivo que se dice que es incorruptible.

Algunos de los mecanismos más utilizados para esto son: proof of work [1], proof of stake [15] y proof of authority [16]. Los mismos se comentan a continuación.

Proof of work

Proof of work o prueba de trabajo, es el mecanismo utilizado por Bitcoin [1], Litecoin [17], Namecoin [18], Dogecoin [19] en la actualidad, y también fue utilizado por Ethereum [20] en su orígen. Este hace referencia a la generación de un hash que sea difícil de producir, pero fácil de verificar. Esto implica que se genera un hash de un valor, o una transacción en este caso, con un determinado algoritmo, como por ejemplo

SHA-256 [21] y que también se ajuste a ciertas condiciones. Lo que se busca es que el generador de ese hash tenga que dedicar gran parte de sus recursos en la generación del mismo, para satisfacer la prueba de trabajo y que a su vez la generación de uno de esos hashes no sea fácil de reproducir o modificar, ya que esto implicaría un costo aún mayor.

Como se comentó en el párrafo anterior, el costo computacional de generar estos hashes es alto, lo que se traduce en una gran cantidad de energía que debe de ser consumida para su generación. En 2015 se estimó que la cantidad de energía eléctrica utilizada para generar una transacción en Bitcoin equivale a 1,75 veces el consumo de energía eléctrica de un hogar promedio en los Estados Unidos por día. Un estudio más reciente realizado en 2018 [22] estima que el costo de realizar una transacción en Bitcoin es de 851 kilowatt/hora mientras que para realizar 100.000 transacciones en VISA se utilizan 169 kilowatt/hora, como se muestra en la figura 2.4.

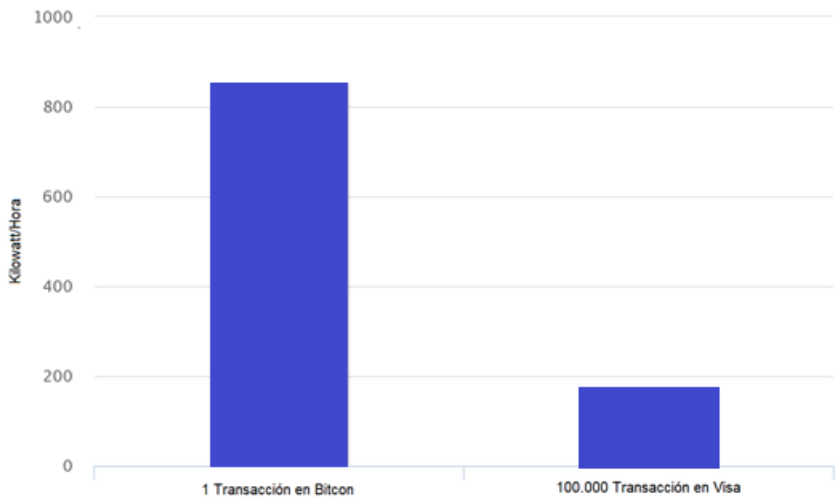

Figura 2.4: Tabla comparativa costo transacción Bitcoin contra Visa

Dado el gran costo y la baja ganancia de generar hashes para los sistemas basados en proof of work, los mismos son susceptibles a que se reduzca la cantidad de nodos generadores de hash. Con menos nodos generadores, el sistema se torna más vulnerable a Ataques del 51%, que consisten en que una organización o conjunto de nodos organizados obtengan la mayoría de los nodos generadores y de esta forma puedan ingresar fácilmente transacciones falsas que los favorezcan.

Proof of stake

Este mecanismo fue creado como alternativa a proof of work, específicamente para solucionar el problema de costos que este presenta. Proof of stake es utilizado por Ethereum, Credits y Litecoin, entre otros. A diferencia de proof of work donde los nodos generadores pueden generar la cantidad de bloques que deseen, en proof of stake es necesaria la utilización de criptomoneda, dado que cada nodo puede generar bloques en proporción a la cantidad de criptomoneda que posea el mismo. Ésto es por ejemplo si un nodo posee un 3% de la cantidad total de criptomoneda de la red blockchain, el mismo solo podrá generar el 3% de los bloques. Esto sirve como mecanismo de protección contra el Ataque del 51%, dado que los atacantes necesitan la mayoría de la moneda del sistema para realizar este tipo de ataques. Para una organización que posea esa cantidad de moneda, sería contraproducente realizar un ataque ya que producto de este, el valor de la moneda bajaría y por tanto estaría perdiendo dinero.

Proof of authority

Proof of authority se basa en tener un conjunto de nodos validadores, los cuales son encargados de validar todas las transacciones que ocurren en el sistema. Este conjunto generalmente se mantiene reducido para asegurar que la eficiencia y la seguridad sea manejable. Es utilizado por Hyperledger, Corda y VeChainThor, entre otros.

Está diseñado para utilizar menos poder computacional que modelos como proof of work que requieren un gran gasto de energía para la resolución de algoritmos, como se mencionó anteriormente en la sección “Proof of work”. Adicionalmente, proof of authority remueve una de las preocupaciones del modelo que plantea proof of stake, en el cual la cantidad de criptomoneda de dos miembros puede ser igual, pero el valor que esto representa para cada uno de los participantes puede ser totalmente distinto dependiendo de sus pertenencias fuera de la red. A modo de ejemplo, en una red dos participantes, Alicia y Bob, pueden tener la misma cantidad de criptomoneda, pero Alicia tiene $10 millones fuera de la red y Bob $10.000. Por lo tanto, es mucho más probable que Bob invierta en el éxito de la red que Alicia, ya que su cantidad de criptomoneda representa una parte sustancialmente mayor de sus finanzas generales.

Se recomienda que los nodos validadores cumplan los siguientes requerimientos básicos, para incentivar el comportamiento honesto por parte de los mismos:

- Sus identidades deben de estar formalmente identificados en la red blockchain, con la habilidad de referenciar su identidad dentro del dominio.
- Convertirse en un nodo validador debe de ser algo difícil, de forma de garantizar que la posición a largo plazo del nodo sea un incentivo, tanto financiero como de reputación, para seguir siendo un validador honesto.
- Tiene que haber uniformidad completa en el proceso de validación.

2.1.4. Smart Contracts

Para definir qué es un smart contract [23], es útil recordar el significado de un contrato: “Un contrato es un pacto o convenio, oral o escrito, entre partes que se obligan sobre materia o cosa determinada, cuyo cumplimiento pueden ser compelidas” [24]. Es decir, son las reglas de juego que permite a las partes que lo aceptan, entender en qué va a consistir su interacción. Están sujetos a leyes y jurisdicciones territoriales, y en ocasiones requiriendo de notarios, algo no accesible para cualquier persona. Un detalle no menor, es que los contenidos de los contratos pueden estar sujetos a la interpretación. Los smart contracts [23] son código escrito en un lenguaje de programación determinado, siendo los términos del contrato sentencias y comandos en el código que lo forma, evitando la interpretación al no ser verbal o escrito en el lenguaje que hablamos.

En el contexto de tecnologías blockchain, los smart contracts son utilizados para eliminar a los intermediarios, implementando los controles y acciones realizadas por estos. Los smart contracts se encuentran almacenados y distribuidos entre los participantes de la red blockchain, de forma de evitar que exista un único punto de falla y que no sean fácilmente modificados. Estas características permiten simplificar procesos que intervienen en una transacción, con el objetivo de ahorrar costos asociados a esos procesos. Un smart contract es creado por uno o más participantes de la red blockchain de acuerdo a los términos de uso y condiciones de ejecución, y una vez codificado es almacenado en la misma blockchain. Este queda disponible para ser invocado por los participantes u otros smart contracts y una vez realizada la invocación, se ejecutan las acciones acordadas, siempre que se cumplan las condiciones de ejecución.

A modo de ejemplo, un smart contract puede ser utilizado para modelar la venta de un bien físico. Supongamos dos participantes de la red blockchain, Alicia y Bob, donde Alicia tiene registrada una casa a la venta y Bob decide comprarla. Se puede generar un smart contract, de tal forma que cuando se genere una transacción indicando que Bob le transfiere el dinero correspondiente a la casa a Alicia, el smart contract puede generar una transacción indicando que la posesión de la casa pasa de Alicia hacia Bob y para su realización no se estarán usando agentes de control externos ni mediadores. Luego de la venta de Alicia a Bob, si Alicia intenta vender nuevamente su casa a otro participante, esta acción será rechazada por los participantes de la red blockchain, dado que allí mismo figura como posesión de Bob.

2.1.5 Otros componentes

Autoridad certificadora

Una autoridad certificadora [25] es la entidad que expide los certificados digitales, así como también la lista de revocación de certificados (CRL). Adicionalmente puede soportar funciones administrativas, aunque generalmente éstas son delegadas a una o varias autoridades. Permite además validar que un certificado haya sido emitido por ella. Aunque las autoridades certificadoras no necesariamente tienen que ser parte de la red blockchain, son utilizadas para emitir certificados para cada participante de forma de poder firmar con esos certificados, la emisión y aprobación de transacciones y/o bloques en la blockchain. La emisión del certificado para el participante sucede cuando éste ingresa a la red y luego lo utiliza durante su participación en la red firmando, transacciones generadas por él o para aprobar una transacción o bloque ingresado por otro participante. Adicionalmente, se pueden realizar consultas contra la autoridad certificadora, para validar que una firma de otro participante haya sido realizada con un certificado emitido por esa autoridad y utilizar esa información para confiar o no.

Oráculos

Un oráculo [26] es un agente que se encarga de obtener y verificar datos externos a la red blockchain de forma segura y confiable, para ser usados por los smart contracts. La necesidad de la utilización de un oráculo en la tecnología blockchain, surge porque no es posible garantizar que la ejecución de una consulta a datos fuera de la red blockchain en dos smart contracts diferentes, arroje el mismo resultado. Por tanto, es necesario contar con un tercero que brinde estas garantías con la mayor precisión posible. Estas características convierten al oráculo [27] en un proveedor de información externo a la red blockchain, que dispara ejecuciones en los smart contracts que son predefinidas ante determinadas situaciones. A modo de ejemplo, estas situaciones pueden ser la temperatura del clima, pagos realizados satisfactoriamente, fluctuaciones de precios, cambios en la propiedad de un bien, etc.

Adicionalmente, la bibliografía [26] presenta una categorización de los tipos de oráculo existentes. Allí se pueden encontrar los del tipo Software que brindan información que existe en línea, los del tipo Hardware que brindan información de objetos físicos, del tipo Inbound que proveen a los smart contracts de información, del tipo Outbound que permiten a los smart contracts enviar información fuera de la red blockchain y los Basados en Consenso que obtienen determinada información de distintas fuentes y la ofrecen a los smart contracts solo si se cumplen las condiciones de consenso.

2.2. Bitcoin

Si bien el foco de esta investigación no es Bitcoin, como se mencionó anteriormente es uno de los primeros casos de implementación de la tecnología blockchain, por lo que es de referencia para las distintas implementaciones. Por este motivo, es relevante para la investigación analizar el proceso de construcción de la blockchain de Bitcoin, el cual se encuentra representado en la *figura 2.5*, que ilustra los siguientes pasos:

1. Las transacciones nuevas son emitidas a todos los nodos.
2. Cada nodo recolecta nuevas transacciones en un bloque.
3. Cada nodo trabaja en encontrar una prueba de trabajo difícil para su bloque.
4. Cuando un nodo encuentra una prueba de trabajo, emite el bloque a todos los nodos.
5. Los nodos aceptan el bloque si todas las transacciones en el bloque son válidas y no se han gastado ya.
6. Los nodos expresan su aceptación del bloque al trabajar en crear el próximo bloque en la cadena, utilizando el hash del bloque aceptado como hash previo.

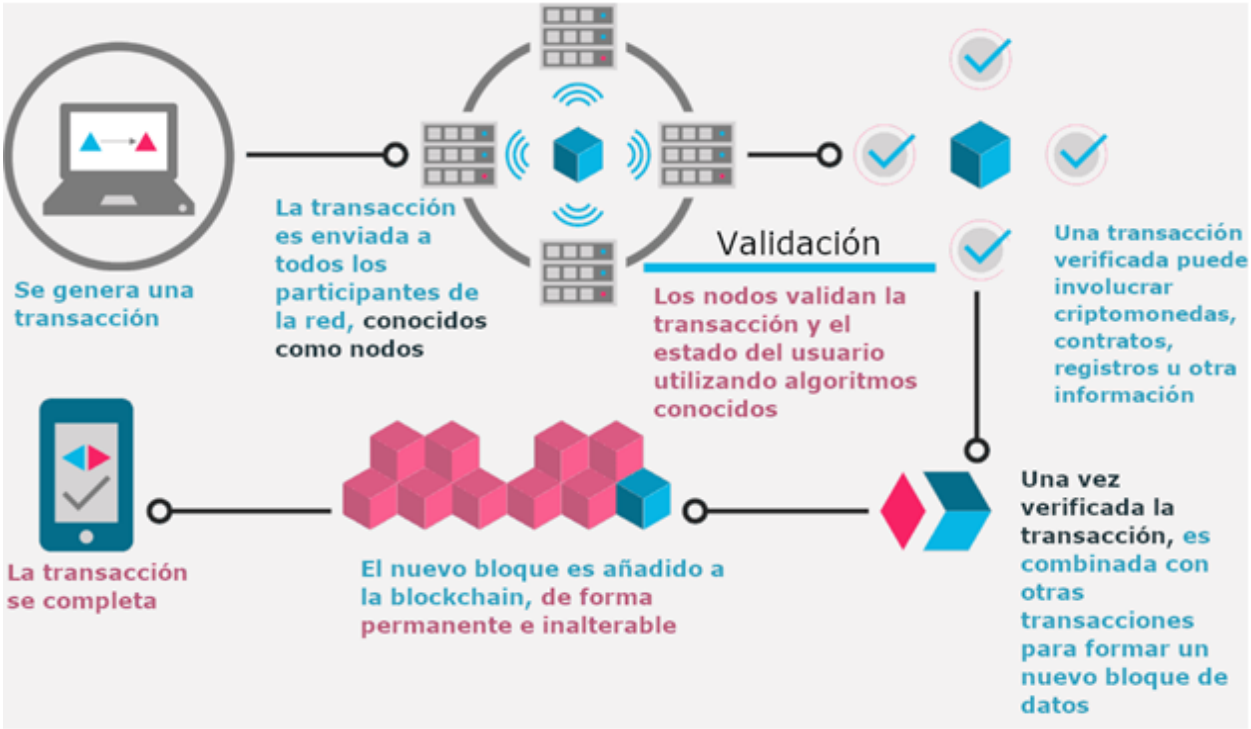

Figura 2.5 - Pasos de construcción en Bitcoin

En Bitcoin las transacciones viajan agrupadas dentro de un bloque y los interesados toman como válido un bloque cuando ya ingresaron por lo menos 6 bloques más, así se tiene plena seguridad de que ese bloque ya está aprobado. Los nodos siempre deberán considerar la cadena más larga como la correcta y deberán continuar trabajando sobre ella para continuar extendiéndola. Si dos nodos envían simultáneamente dos versiones distintas de bloques, algunos nodos recibirán una versión primero, mientras que otra parte recibirá la otra versión primero. En estos casos, cada nodo debe trabajar en la primera versión que recibe, pero igualmente deberá conservar la cadena que se forme a raíz del otro bloque, en caso de que esta cadena se torne la más larga. Estas dos cadenas (o la cantidad de ramas que se hayan generado) se mantendrán hasta que en una de ellas se logre resolver primero un problema planteado para cada una de las cadenas. Este concepto es el llamado proof of work que fue explicado en la sección “2.1.3. Integridad”. La cadena que resuelva este problema primero se convertirá en la cadena más larga y esto implica que los nodos que se encuentren trabajando sobre otra cadena, deberán entonces cambiarse a la cadena más larga.

3. Casos de aplicación

Blockchain es una tecnología muy joven que presenta aspectos revolucionarios para la industria por lo que es de gran interés su aplicación en distintos negocios. Sin embargo, en base a lo investigado, se ha detectado que no se trata de algo simple de aplicar y que, sin contar con el caso de Bitcoin y otras criptomonedas, los casos conocidos de aplicaciones en producción al dia de hoy son reducidos.

Como se mencionó anteriormente, uno de los objetivos de este trabajo es evaluar el uso de esta tecnología en la administración pública, por lo que se considera prudente comenzar la investigación por el estado de situación en estos contextos.

Para poder llevar a cabo la investigación, para cada proyecto mencionado en el capítulo, se desarrollaron tres conceptos importantes para su compresión: en qué consiste el proyecto, qué problemática intenta solucionar o mejorar y en caso de que aplique, su comparación con soluciones ya existentes no basadas en blockchain.

3.1. Situación actual

En la actualidad existen pocos desarrollos realizados sobre la tecnología blockchain que hayan tenido éxito y en su mayoría están vinculados a la criptomoneda. Alineado a los objetivos de este documento, si bien el uso de la tecnología blockchain vinculado a criptomonedas es interesante, no es el foco de estudio y por tanto se investigó su aplicación en otros contextos, de forma de entender mejor otros escenarios posibles dentro de la administración pública.

Existe un servicio ofrecido por la plataforma llamada Proof Of Existence [28] para el registro de patentes y/o propiedades, guardandolo en la blockchain de Bitcoin a un costo de 0.00025 BTC. El objetivo detrás de esto es que por ejemplo una empresa pueda probar que ha creado una tecnología en una fecha concreta, sin necesidad de hacer una aplicación formal para registrar la patente. Si se compara con los costos de registrar una patente en Uruguay, según la tabla de costos presentada en el documento “Tasas y Precios - Dirección Nacional de la Propiedad Industrial” [29], la solución brindada por Proof of Existence [28], resulta menos costosa.

Por otra parte, se pueden encontrar muchas propuestas de uso en cadenas de suministro, como son TradeLens [30], Moyee Coffee [31] u OriginTrail [32]. La primera es un proyecto impulsado por Maersk [33] e IBM y lo que propone es registrar la trazabilidad de los contenedores para poder realizar un seguimiento de ellos. Esto genera como beneficio el saber exactamente qué está pasando con la cadena de suministros, de forma de reducir el stock de seguridad y aumentar la capacidad de

recuperación. Adicionalmente, elimina la necesidad de administrar manualmente gran parte de la información relacionada a los envíos y obtener información confiable sobre el envío para validar eficientemente las tarifas y recargos.

Por su parte Moyee Coffee, es un emprendimiento que busca eliminar las empresas que procesan el café mundial que se quedan con gran parte de las ganancias, según lo mencionado en su sitio web [31]. Lo que se busca con su aplicación es la de favorecer a los productores de café, que actualmente obtienen un bajo porcentaje de la ganancia de la producción con un nivel de transparencia sin precedentes. Como prueba de concepto se implantó un piloto en Etiopía, que consiste en que cada participante (granjero, productor y consumidor) pueda acceder a toda la información a lo largo de la cadena de suministros.

En cuanto a OriginTrail, es una solución de cadena de suministros, la cual permite manejar cualquier tipo de artículo, de forma de generalizar la solución. Esta brinda un entorno en el cual, las distintas partes interesadas pueden interoperar utilizando sus sistemas de información, compartiendo información de sus cadenas de suministros y utilizando mecanismos de consenso para garantizar la integridad de los datos. Además esta solución es pública y descentralizada, para brindar mayor performance y escalabilidad y reducir costos. Su primera versión está siendo probada en la industria alimenticia, pero consideran que en las próximas versiones se podrá trabajar con cualquier producto, automovilístico, farmacéutico, etc.

Una aplicación que se encuentra en desarrollo, con un enfoque más bien hacia criptomonedas es GodZillion [34], que propone la financiación colectiva de distintos tipos de proyectos a través de criptomoneda. Esta solución está basada en smart contract, utilizando la máquina virtual de Ethereum. Su objetivo es reducir los costos inherentes a la utilización de sitios para financiación colectiva de Startups, eliminando los intermediarios y descentralizando la información.

Por otro lado, recientemente la empresa WePower [35] finalizó una prueba de concepto en Estonia, que propone la utilización de la tecnología blockchain para registrar transacciones de transferencia de energía entre consumidores y productores. Esto surge por el auge que están teniendo las distintas fuentes de energía renovable (solar, eólica, etc.) en el mundo. Dadas las distintas fuentes de energía, se podría crear a través de la tecnología blockchain un sistema que no dependa de un solo proveedor y que no solo se pueda elegir el tipo de energía que se desea consumir, sino también hasta su proveedor. En comparación con la oferta actual centralizada, esto supone un beneficio económico importante para los consumidores, pudiendo acceder a una oferta variada, que se ajuste mejor a sus necesidades, además de una mejor disponibilidad por la característica distribuida.

Un escenario encontrado que genera dudas en torno a su viabilidad, refiere al almacenamiento en la nube de forma distribuida, creando un servicio como Mega o Google Drive, pero sobre la tecnología blockchain. La plataforma se llama Storj, es de código abierto y está planificada su liberación para principios de 2019. Los casos de uso que contempla básicamente engloban el almacenamiento de grandes volúmenes de datos, backup y recuperación de datos ante eventuales fallas o ataques, entre otros. De acuerdo a lo que plantean en su paper [36], el almacenamiento descentralizado es una respuesta al desafío de proveer una solución de almacenamiento en la nube, segura, privada, performante y económica, forzando un almacenamiento más económico a una tasa mayor que la que un único proveedor puede brindar. Si bien tendría un costo mensual de U$S 0.015 por GB almacenado por mes y U$S 0.05 por GB descargado, si comparamos con servicios como Mega que por U$S5.69 al mes, permite almacenar 200 GB y 1 TB de transferencia, en el mismo periodo de tiempo por las mismas capacidades tendría un costo mayor, siendo el mismo U$S 54.2 por mes. Por otra parte, si comparamos únicamente el almacenamiento con Google Drive que en su tarifa más económica permite almacenar hasta 100 GB por U$S 1.99 al mes, Storj resultaría ser más económico a un valor de U$S 1.5 al mes. Como se puede ver en la *figura 3.1* y *3.2*, Storj resultaría más económico que otras alternativas en contextos de baja transmisión de datos y más cara en contextos de alta transmisión de datos.

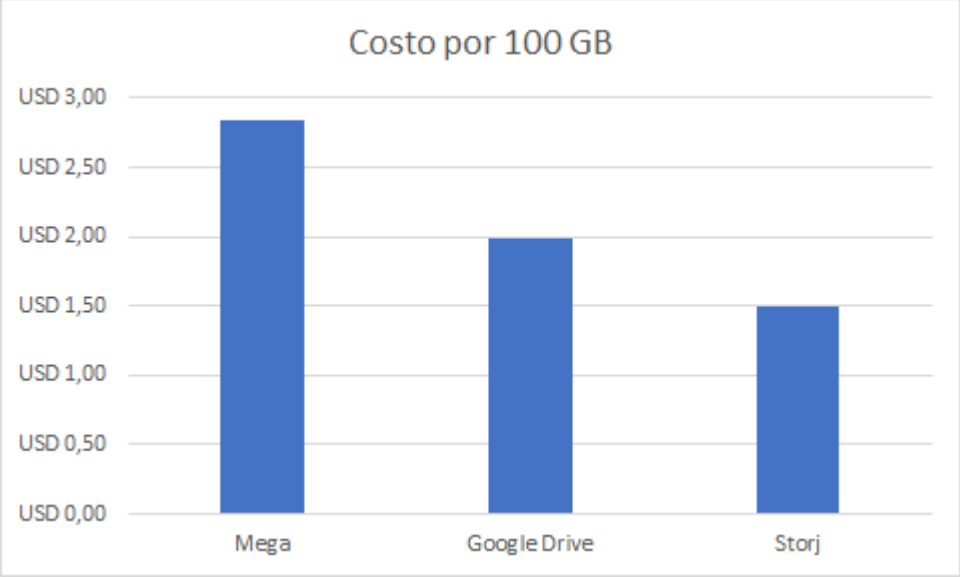

Figura 3.1 Gráfica de costo de almacenamiento por plataforma

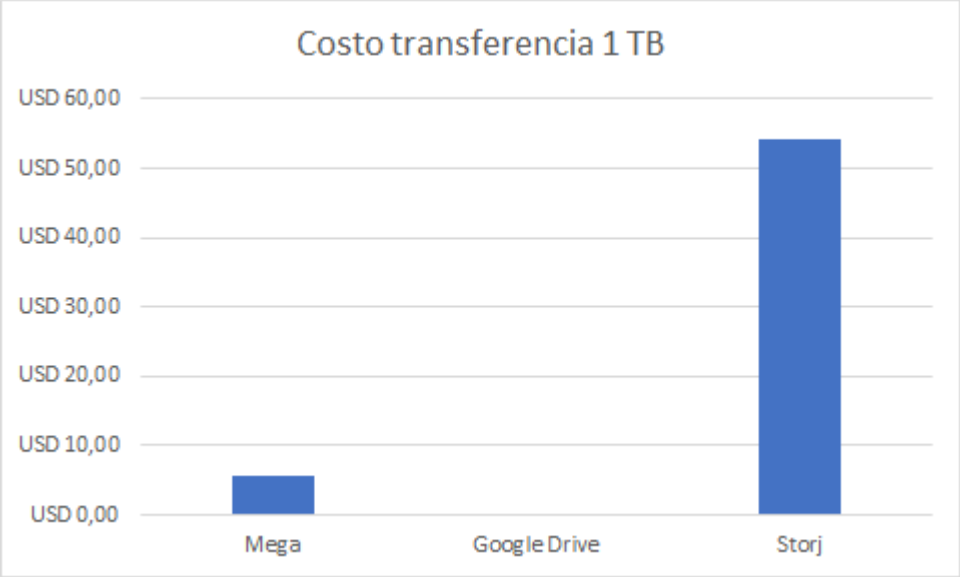

Figura 3.2 Gráfica de costo de transferencia por plataforma

Más allá del costo, suponiendo que cada nodo tiene una copia de la información de la blockchain, y que la plataforma tuviera 10000 usuarios con un promedio de 500 GB de información almacenados por usuario, esta blockchain ocuparía un promedio de 5000 TB de espacio de almacenamiento. Por tanto, la utilización de la tecnología blockchain en este caso no parece viable, dado que cada nodo de la red debería de poder almacenar esta cantidad de información.

3.2. Casos de uso en ámbito público

Dado que el foco del proyecto está centrado en el uso de la tecnología blockchain en el ámbito público, se comenzó analizando el estado de situación en el mundo. Para llevar adelante este análisis se tomó como punto de partida un documento realizado por la empresa Deloitte, llamado “Will blockchain transform the public sector?” [9] [2]. Este documento presenta los distintos casos en los que se está investigando en el mundo, como lo muestra la *figura 3.3*. El documento fue elegido por sobre otros, debido a que existe poca documentación que realice este tipo de análisis y lo poco encontrado hace referencia a ésta misma, como por ejemplo “Blockchain and its Use in the Public Sector” [37].

Como se mencionó al principio del capítulo, para realizar la investigación de los proyectos se analiza en qué consiste, qué soluciona y su comparación con soluciones existentes que no utilizan la tecnología blockchain, en caso de que aplique.

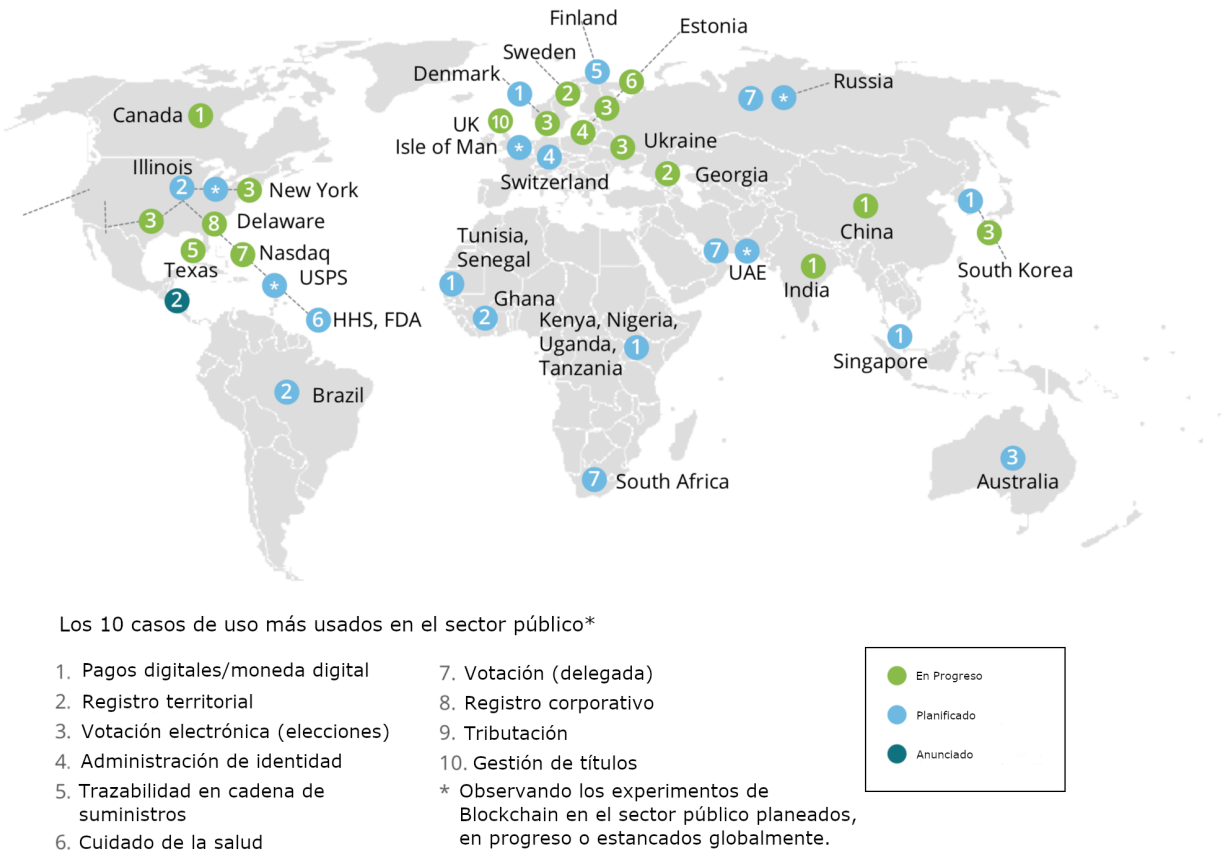

Figura 3.3. Distribución del uso de blockchain en el mundo

Como se puede apreciar, existe una gran variedad de casos y hay una mayor concentración en las zonas de Europa y Estados Unidos, teniendo una mayor importancia los casos relacionados con pagos y uso de moneda digital. Como se mencionó anteriormente, este no es el foco del proyecto, por lo que se hace énfasis en otros casos que se entiende son de gran valor. Estos son la gestión de identidad, el registro territorial o de documentos y la votación electrónica.

Según el documento de referencia, la gestión de identidad es uno de los desafíos más grandes a nivel mundial (un quinto de la población mundial no tiene una identidad legal u oficialmente reconocida) y esto es reconocido por todos los actores del sector público. La falta de estándares para establecer una identidad digital y diferentes procesos de certificación e identidad, son algunos de los puntos críticos de este tipo de casos de uso. En estos contextos, de acuerdo a lo mencionado en la bibliografía, blockchain aporta valor definiendo un entorno seguro y de gestión sobre la identidad, permitiendo el intercambio eficiente de distintos tipos de información. Además, permite y deja explícito el control sobre qué identidades y con qué propósito son compartidas las identidades. En este sentido, surgen proyectos que buscan ese objetivo, algunos

con una finalidad más bien experimental como es “World Citizenship” [38] que propone el registro de pasaportes de forma distribuida y segura. El objetivo del proyecto es diseñar un proceso simple que permita crear un servicio de pasaporte privado que pueda ser utilizado para validar y verificar la identidad de una persona, mediante la utilización de herramientas de código abierto.

Actualmente la verificación de identidad de una persona, es realizada mediante verificación ocular, corroborando que la persona que presenta el documento coincida con la fotografía que contiene el documento. El problema con este sistema es que, si una persona falsifica el documento y cambia la fotografía por la de otra persona, esta puede hacerse pasar por la dueña del documento. Este problema es resuelto por los proyectos antes nombrados, donde el documento y la fotografía son almacenados en la blockchain, por lo que permanecerán distribuidos, inalterados a lo largo del tiempo y sus cambios quedarán registrados en la misma siendo estos aprobados mediante el consenso de los participantes de la red blockchain. En comparación a una solución utilizando bases de datos distribuidas, si se vulnera uno de los nodos de la base y se modifican sus datos, se puede ver comprometido todo el sistema, mientras que mediante la aplicación de la tecnología blockchain, dependiendo del algoritmo de consenso, se tendrían que vulnerar una cantidad mayor de nodos para comprometer el sistema.

Otro punto del documento refiere al voto electrónico, el cual menciona que: es una de las funciones públicas más críticas y que tiene que brindar garantías legítimas. Los ciudadanos podrían realizar su voto de forma electrónica, de la misma forma que hacen otras transacciones hoy en día e inclusive verificar el estado de la elección en ese momento. Si bien parece ser un caso muy interesante para aplicar, aún hay aspectos en los que trabajar, como lo son la gestión de la identidad que ya fue mencionado, anonimización del voto y la verificación del mismo por parte del votante.

La empresa tecnológica Ágora [39] formada en 2015, se encuentra desarrollando una solución de votación verificable para gobiernos e instituciones. Está basada en blockchain, buscando proveer una solución más segura y rentable. Pretende ser a prueba de manipulaciones durante todo el proceso de votación, ofreciendo transparencia para los votantes, auditores y público en general. Se trata de una blockchain permisionada, que presentaría nodos con permisos de escritura operados por Ágora y reconocidos por terceros de confianza (Nodos de consenso) y nodos de solo lectura (Nodos auditores ciudadanos), que podrian ser operados por cualquiera en el mundo. Adicionalmente, utilizaría la blockchain de Bitcoin, para almacenar el hash del bloque más reciente cada cierto periodo de tiempo, de forma que fácilmente pueda verificarse que la blockchain de Ágora permanece inalterado.

La utilización de voto electrónico, como fue mencionado anteriormente, brinda mayores garantías de seguridad que la votación mediante urnas, manteniendo la información

anónima. En Uruguay las urnas deben ser custodiadas y procesadas manualmente y en algunas ocasiones como en las elecciones nacionales, requiriendo extensas jornadas de conteo de votos, demorando hasta una semana para obtener los resultados oficiales. Además, las urnas del interior del país deben de ser transportadas a la capital para realizar el último conteo, provocando retrasos, riesgo de adulteración y hasta pérdida de las urnas, sin contar el costo de todo el personal necesario para su transporte. Con la aplicación de la tecnología blockchain se propone ahorrar estos costos, agilizando los tiempos de procesamiento y publicación de la información, manteniendo la confidencialidad de los sufragios realizados por cada persona, asegurando la integridad y dificultando la alteración de los votos.

Otro punto atacado en el documento hace referencia a los registros territoriales o de documentos, para los cuales plantea que no solo sirven para demostrar la propiedad sobre lo que represente ese documento, sino que también pueden servir como inversión o crecimiento económico en países en desarrollo. Este tipo de procesos actualmente insumen una gran cantidad de papeleo, un alto costo, poco eficientes y presentan varias vulnerabilidades en su manipulación. El documento plantea que un sistema estándar y descentralizado de registros puede reducir el número de intermediarios requeridos. La utilización de la tecnología blockchain pretende facilitar la identificación y validación de los participantes de las transacciones, aumentar la eficiencia de los procesos y generar un ahorro en el tiempo y costo del procesamiento, y asegurar la integridad de los documentos almacenados.

En este sentido, el gobierno de Georgia [40] se encuentra trabajando en la implementación de un sistema para el registro de títulos de propiedad, que permitiría firmar y verificar un documento que contiene información esencial para un ciudadano y obtendría de esta forma pruebas de su propiedad. Adicionalmente, permitiría asegurar la legitimidad del documento sin exponer el contenido de este. Esto se lograría registrando en la blockchain un hash único que representa a ese documento y que por tanto no dejaría expuesta ninguna información confidencial. Se trata de una blockchain privada y permisionada, que estaría enlazada con la blockchain de Bitcoin, para registrar los hashes de los documentos, favoreciendo la consulta pública y aprovechando la seguridad de la blockchain de Bitcoin.

En el caso del gobierno de Estonia, se almacenan los registros médicos de más de un millón de pacientes para asegurar su integridad. Está basado en Keyless Signature Infrastructure (KSI) [41], que únicamente permite que se agregue nueva información y mantiene los registros seguros de cualquier alteración, además de protegerlo contra interferencias, fallas o virus. Como es mencionado en el informe *Registros médicos en Estonia y Dubai* [42], esto pretende empoderar a los usuarios, brindar mayor transparencia y mejorar el diálogo entre el paciente y el doctor.

En Dubai [43] se está testeando el uso de la tecnología blockchain para asegurar los documentos, eliminar el papeleo de las interacciones y de esta forma ahorrarse aproximadamente 1.500 millones de dólares anuales. Como prueba de concepto se comenzó con los registros médicos, buscando minimizar los problemas de interacción entre paciente y doctor, permitiéndole al paciente compartir su información con cualquier doctor en cualquier sistema.

En Uruguay se está llevando a cabo el programa HCEN [44] de Salud.uy, que tiene como principal cometido promover y mejorar la continuidad del proceso asistencial de los usuarios del sistema de salud uruguayo, mediante un mecanismo que permite unificar y hacer disponible toda la información clínica del usuario de Salud ante un evento asistencial. Comparando este con el caso de Dubai y Estonia, y evaluando las ventajas que este otorga al centralizar y disponibilizar la información de historias clínicas en todo el país, se puede apreciar que son las mismas, aunque en el caso de Dubai y Estonia, se busca asegurar la integridad e incorruptibilidad de las historias clínicas mediante el uso de la tecnología blockchain, haciendo de esta una mejor opción.

Por otra parte, a pesar que este caso está vinculado a la utilización de criptomonedas, es importante mencionar un caso encontrado en el campo de refugiados sirios en Jordania [45], el cual tiene asistencia de la ONU y que a partir del programa Building Blocks [46] en el 2017 se comenzó a aplicar la tecnología blockchain para asistir económicamente a las familias que allí viven. La utilización de blockchain supone un ahorro considerable, debido a que se dejaron de utilizar bancos locales e internacionales para poder realizar las transferencias, por las cuales estos cobran una comisión. También se pueden encontrar problemas relacionados a proveedores de servicios financieros, como la privacidad de los datos, inoperabilidad y riesgos financieros. Esta solución utiliza una blockchain basada en el protocolo de Ethereum, plataforma que será presentada en la sección “4.1.1 Ethereum”, que consiste en registrar en la plataforma blockchain todas las transacciones de los beneficiarios, quedando asociadas a sus billeteras virtuales. Esta solución implicó que no fueran necesarios fondos de proveedores de servicios financieros, se redujo en un 98% la comisión de los bancos locales y se logró proteger la información de los beneficiarios al no compartirse fuera de la ONU, sin afectar la experiencia de los mismos.

4. Plataformas de blockchain

En este capítulo se mencionan las principales plataformas o frameworks que se encuentran en el mercado para implementar aplicaciones sobre la tecnología blockchain y sus distintas características. Dado que ya se encuentran disponibles análisis comparativos de distintas plataformas, se optó por basarse en uno de ellos para la elaboración de este documento. La elección de la publicación fue realizada basada en que las plataformas analizadas, presentaran características que pudieran aplicar a la administración pública. Según lo comentado, la publicación que mejor se ajustó a esta condición fue “A blockchain platforms comparison” [47], la cual fue utilizada para la elección de las plataformas. En base a esa preselección, se investigó cada una de ellas destacando qué tipo de blockchain implementa, qué algoritmo de consenso utiliza y si puede modelar múltiples tipos de problema o es específica de uno solo. Entre las plataformas y frameworks analizados se encuentran: Ethereum, Corda, Credits y Hyperledger Fabric, esta última será detallada en mayor profundidad en su versión 1.1 [48], dado que la misma fue elegida para realizar la prueba de concepto de este proyecto, el cual es presentado en el capítulo “5. Caso de uso en administración pública”. La decisión de usar este framework en esta versión fue tomada en base a lo siguiente:

- No depende de una criptomoneda. Desde su base Hyperledger no utiliza criptomonedas para la creación de la blockchain, pero en caso de necesitarlo permite hacerlo. Esto le da versatilidad para la elaboración de cualquier caso de uso basado en blockchain, que como se comentó en la sección anterior, existen casos en los que sí aplica y otros en los que no.

- Definición de canales de datos, como será explicado más adelante, esto permite un nivel más de privacidad de datos, dado que los datos de un canal solo se encuentran en los nodos registrados a ese canal. Esto es importante porque se pueden definir canales de intercambio bajo la misma red blockchain, reutilizando la red blockchain ya definida, sin necesidad de crear una nueva red blockchain o plataforma de intercambio. Además, esta característica se encuentra alineada con necesidades que surgen dentro de la administración pública, en las que un conjunto cerrado de entidades tiene la necesidad de intercambiar información sensible únicamente dentro de ese conjunto.

- Permite gestionar quién puede leer y escribir datos dentro de un canal, esto es importante en el contexto de la administración pública, donde se requiere un control exhaustivo de quién tiene permitido publicar información y quién consumirla.

- Flexibilidad en la determinación de consenso, como se mencionó anteriormente, el conjunto de entidades que puede escribir cierta información está estipulado,

este conjunto puede variar, por lo que es importante poder modificar el consenso para cumplir esto.

4.1 Plataformas investigadas

En esta sección se presenta una breve introducción de las plataformas investigadas, como ya fue mencionado, se indicará qué tipo de blockchain implementa, qué algoritmo de consenso utiliza y si puede modelar múltiples tipos de problema o es específica de uno solo.

4.1.1 Ethereum

Ethereum [49] es una blockchain pública y descentralizada, la cual permite la creación de cualquier tipo de aplicaciones dado que es Turing completo [50]. Esta define una criptomoneda llamada Ether, la cual es consumida en la ejecución de cada transacción realizada en la plataforma blockchain. Mediante la utilización de smart contracts se pueden implementar distintos tipos de lógica de negocio, asociando un costo en Ether a la ejecución de cada smart contract. Según el sitio web de Ethereum de esta forma se permite modelar cualquier tipo de transacción, acuerdo o cualquier actividad con aspectos económicos o de gobierno.

Como característica a destacar, esta plataforma blockchain utiliza proof of work. Como se vio anteriormente, proof of work implica costos considerables por transacción, lo cual puede originar disconformidad entre los mineros y pérdida de usuarios. Se está trabajando en esta plataforma para migrar a proof of stake y como estimación inicial se pretendía comenzar la migración a finales de 2017, pero esto se retrasó dada la necesidad de escalar de la plataforma, dejando de lado la migración para realizarse entre 2019 y 2020. Como fue mencionado en la subsección “2.1.3. Integridad”, proof of stake reduce los costos a los que se enfrentan los mineros, permitiéndoles solo minar una cantidad de bloques relativa a la cantidad de la criptomoneda que poseen, aumentando de esta forma el atractivo de utilizar la red.

4.1.2 Corda

Corda [51] es una plataforma blockchain de código abierto que disponibiliza la empresa R3 [52], la cual se dedica al desarrollo de software blockchain para empresas de distintas industrias tanto del sector público como del privado. Adicionalmente ofrece Corda Enterprise, que es una versión comercial de Corda, optimizada para el uso empresarial. Corda es una blockchain permisionada, la cual fue especialmente diseñada para la industria de servicios financieros. Cuenta con el apoyo de grandes bancos internacionales, como: ABN Amro Bank, BBVA, Bangkok Bank, ING, SEB, U.S. Bank, HSBC, entre otros.

Como mecanismo de consenso utiliza proof of authority, que como fue mencionado en la subsección “2.1.3. Integridad” son un conjunto de nodos los encargados de validar

las transacciones, para luego poder ser guardadas en la blockchain. En esta plataforma, estos nodos son llamados nodos notarios.

4.1.3 Credits

Credits [53] es una plataforma para blockchain, permisionada, basada en el principio de peer-to-peer, totalmente descentralizada, para una interacción directa entre sus miembros. Los participantes tienen la posibilidad tanto de crear como de utilizar servicios financieros. El mecanismo de consenso es una combinación de proof of stake y tolerancia al problema de los generales bizantinos, llamado así para ilustrar la problemática de lograr un consenso entre un conjunto de entidades con un objetivo común cuando entre esas entidades pueden existir objetivos opuestos a los del resto.

4.2 Hyperledger Fabric

Hyperledger Fabric [54] es un framework que surge de un proyecto de código abierto, iniciado en diciembre de 2015 por la Fundación Linux. El objetivo del proyecto es aunar esfuerzos independientes para desarrollar estándares y protocolos abiertos para el trabajo sobre la tecnología blockchain, así como proporcionar un marco modular que soporte componentes diferentes para usos diferentes.

A diferencia de otros frameworks de blockchain que permiten miembros desconocidos en la red, la red definida por este framework es privada y permisionada y por tanto exige que sus miembros sean inscriptos. Para esto, Hyperledger define en su arquitectura un componente proveedor de servicios de membresía, que es denominado MSP por sus siglas en inglés (Membership Service Provider), cuyo objetivo principal es el de emitir certificados para cada uno de los miembros de la red blockchain.

Hyperledger provee algunas características destacables respecto a otros frameworks ya presentados en cuanto al manejo de la tecnología blockchain. Estos son que los mecanismos de consenso pueden ser actualizados según las necesidades de la red y soporta múltiples MSP, pudiendo utilizar los provistos por defecto por Hyperledger o implementaciones propias.

Una arquitectura genérica de lo que propone este framework, que es utilizada en los ejemplos provistos por esta, es la propuesta en la *Figura 4.1*. Existen varios detalles en cuanto a su funcionamiento interno que son importantes como son la definición de la topología de la red, la configuración de canales, las condiciones de consenso, configuración de los Orderers y definición de las chaincodes. Estos serán explicados a lo largo de esta sección, de forma de entender los ajustes que se deben hacer para poder aplicarlo en el caso de uso que se desee.

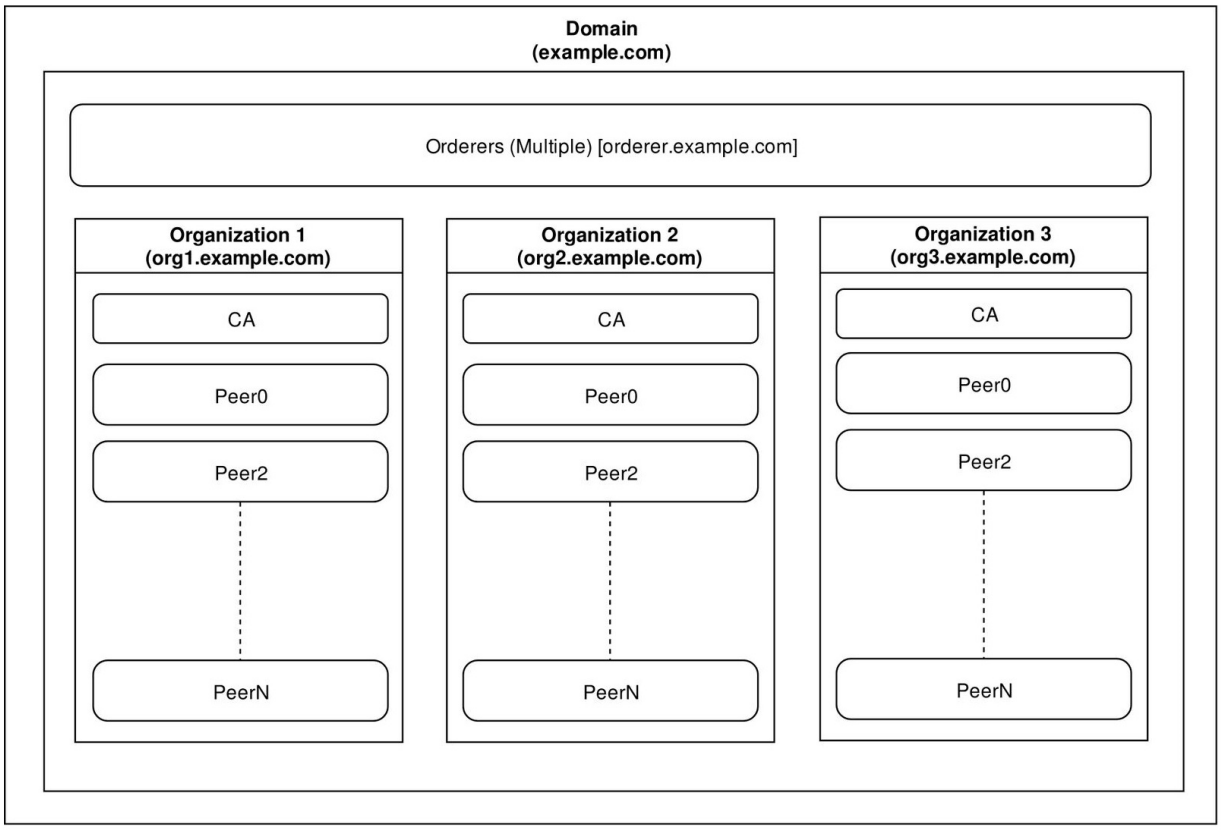

Figura 4.1. Diagrama de red compuesta por tres organizaciones

Ledger

Hyperledger Fabric tiene un subsistema llamado Ledger, el cual consta de dos componentes: el “Estado del mundo” y el “Registro de transacciones”, como se puede apreciar en la *Figura 4.2*. Cada participante tiene una copia de cada ledger [55] de cada red a la que pertenece. El componente del Estado del mundo describe el estado del ledger en un punto determinado del tiempo. Es la base de datos del ledger, siendo la misma del tipo clave-valor y pudiendo cambiar sus datos. El componente de Registro de transacciones registra todas las transacciones que dieron como resultado el valor actual del Estado del mundo, es decir, es el historial de actualizaciones del Estado del mundo y a su vez almacena todas las actualizaciones que no fueron satisfactorias. Este componente, registra en orden todas las transacciones, imponiendo de esta forma un orden específico de ejecución.

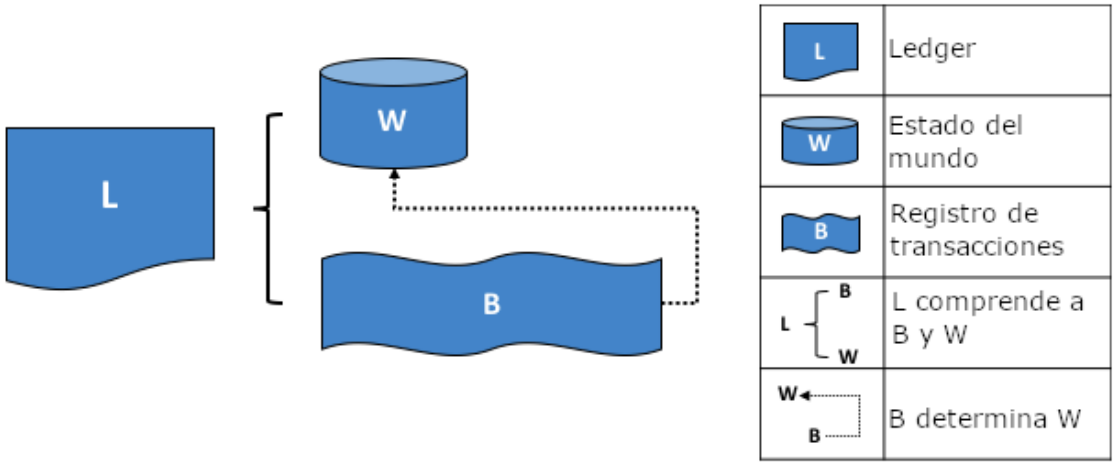

Figura 4.2. Composición del ledger

El ledger es almacenado en todos los nodos que pertenecen a la red, mientras que en el Orderer se almacena solamente el Registro de transacciones, como se puede ver en la *Figura 4.3*. Opcionalmente los nodos pueden tener un subconjunto del ledger, el cual indica cuales son las transacciones válidas e inválidas. El Registro de transacciones almacenado en los Orderers brinda disponibilidad y tolerancia a fallas, permitiendo replicar la historia de todas las transacciones a los nodos y que así construyan el estado del mundo.

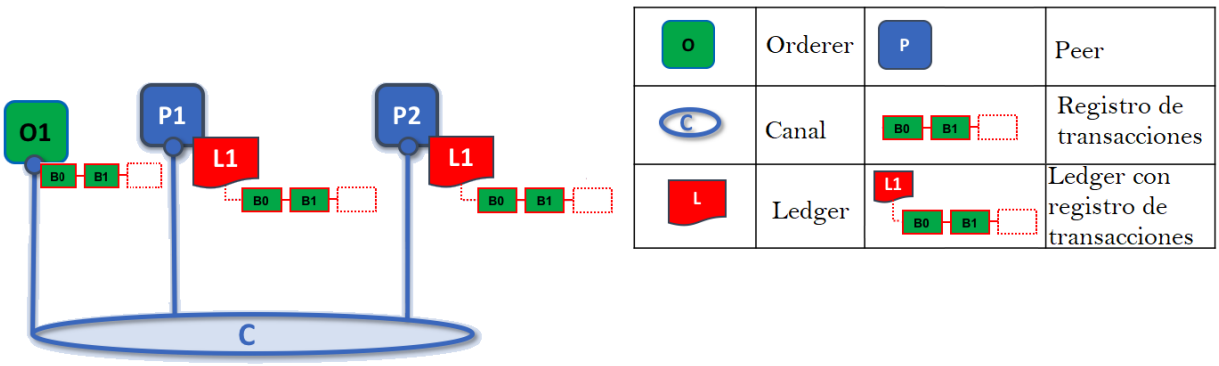

Figura 4.3. Almacenamiento de ledger y Registro de transacciones

Topología

Hyperledger Fabric es compatible con redes en las que la privacidad es un requisito operacional clave, así como redes públicas. Esto significa que se pueden realizar configuraciones para adaptarse a una u otra característica. Para el caso de redes privadas, por ejemplo, se utilizan certificados emitidos únicamente para los

participantes autorizados, los cuales serán los únicos válidos para firmar cualquier transacción. El intercambio de información se encuentra determinado por los smart contract, los cuales son instalados en los participantes de la red blockchain. En Hyperledger Fabric, los smart contracts son llamados chaincode y serán presentados más adelante.

Las topologías [56] de red que se pueden construir con Hyperledger Fabric, permiten configuraciones en las que todos sus nodos se encuentran directamente conectados, perteneciendo todos a una misma organización o una configuración descentralizada en la que los nodos se agrupan por organizaciones. Al mismo tiempo, también nos permite tener simultáneamente varias chaincode funcionando en la red, dedicadas cada una de ellas, en principio, a un negocio distinto. A modo de ejemplo, podríamos considerar una cadena de supermercados, en la cual utilizarían una chaincode para el manejo de stock de productos y otra para el registro del personal. Además, permite publicar distintos canales sobre una misma red, teniendo la posibilidad de instalar la misma chaincode en distintos canales.

Hyperledger Fabric, al igual que otras plataformas, permite el agregado de nuevas organizaciones, o nuevos peers a organizaciones ya existentes sin necesidad de afectar el funcionamiento normal de la red que ya se encuentra activa. Adicionalmente y por el propio funcionamiento de la tecnología blockchain, al dar de alta ese peer o conjunto de peers, automáticamente se sincronizan con el estado actual de la blockchain. Esto se logra obteniendo cada bloque de la blockchain y ejecutando las transacciones que allí se encuentran.

Por otra parte, además del manejo de los peers participantes, también se encuentra la posibilidad de agregar nuevas chaincode o nuevas versiones de chaincode ya existentes en pleno funcionamiento de la red, esto permite agregar nuevas funcionalidades a la red, cambiar funcionalidades ya existentes y realizar ajustes sobre chaincodes con errores.

Canales

Hyperledger Fabric provee una característica destacada con respecto a otros frameworks, los canales [57]. Hyperledger Fabric lo define como una “subred” privada de comunicación entre dos o más miembros específicos de la red, con el propósito de intercambiar transacciones privadas y confidenciales. Por tanto, los nodos no registrados en el canal, no serán notificados de las transacciones que afecten a ese canal. Como se verá más adelante, el ingreso de datos a los canales también puede ser restringido a ciertos nodos pertenecientes al canal, esto mediante las chaincodes y sus políticas de escritura. Al igual que los nodos, los canales pueden ser agregados en cualquier momento en la red.

Orderer

Este componente es el encargado de ordenar y verificar que se cumplan las políticas de escritura correspondiente al canal y chaincode donde se ejecuta la transacción, las mismas serán explicadas más adelante en la subsección “Chaincode”. Para esto, no son utilizados los datos de las transacciones, sino que son utilizadas las firmas digitales generadas por los nodos donde se enviaron a validar las transacciones, como será explicado más adelante en la subsección “Consenso”. Existen distintos tipos de configuraciones para el Orderer, “Solo”, “Kafka” y “PBFT” y cada una de ellas tiene sus características particulares y por ende cada una es recomendada para distintos escenarios.

La configuración “Solo” [58], es recomendada para el testeo del desarrollo de las chaincodes, canales y nodos participantes. Este está pensado para desplegar fácilmente y consiste de un solo proceso que atiende a todos los clientes. Esta configuración no es recomendada para ser utilizado en producción, dado que no cuenta con alta disponibilidad o escalabilidad.

Por otra parte, se encuentra la configuración “Kafka” [58]. Esta utiliza el sistema publish/subscribe para el ordenamiento de las transacciones implementado por Apache Kafka [59] pero de forma tal que el código no tenga que ser escrito específicamente para Kafka. De momento esta configuración es la opción para los despliegues en producción, que demandan alta disponibilidad y una alta tasa de transferencia, pero que no requiera tolerancia a fallas (tolerancia al problema de los generales Bizantinos).

Por último, se encuentra la configuración “PBFT” [58], la cual de momento continúa bajo desarrollo y no tiene una fecha estipulada de liberación. Este tipo de configuración tiene el cometido de ordenar el flujo de los mensajes siendo tolerante a fallas del tipo bizantino.

Chaincode

Como se mencionó anteriormente, estos son la implementación [60] de los smart contracts. Las chaincodes se tratan de código programado, versionado y desplegado en cada participante de la red donde son ejecutados y pueden ser implementados en diferentes lenguajes de programación como Java y Go, siendo este último el más utilizado. Durante el proceso de consenso, son utilizados por los participantes para verificar que las transacciones que le son enviadas sean válidas.

Como se mencionó anteriormente estas son instaladas por canal, definiendo un contexto de ejecución, pudiendo ser instaladas en más de un canal. Este par, chaincode-canal es único y toda información intercambiada bajo ese contexto, será visible únicamente para los nodos que tengan instalado este par.

A la hora de instalar una chaincode en una red es necesario definir: peer donde se instalará, nombre, canal, número de versión y política [61] de invocación a operaciones de escritura. Las políticas antes mencionadas son especificadas en un lenguaje definido por Hyperledger Fabric el cual permite indicar qué organizaciones y cuántos peers de las mismas deben de firmar la transacción para ser considerada válida.

A continuación, se mostrarán ejemplos de políticas para una red con tres organizaciones:

- *AND ('Org1MSP.member','Org2MSP.member')*: Esto quiere decir que la transacción tiene que ser firmada por lo menos por un miembro de la organización “Org1” y un miembro de la organización “Org2”.
- *OR ('Org1MSP.member','Org2MSP.member','Org3MSP.member')*: Esto quiere decir que la transacción tiene que ser firmada por lo menos por un miembro de alguna de las tres organizaciones.
- *AND ('Org1MSP.member','Org1MSP.member')*: Esto quiere decir que la transacción tiene que ser firmada por lo menos por dos miembros distintos de la organización “Org1”.
- *OR ('Org1MSP.member',AND ('Org2MSP.member','Org3MSP.member'))*: Esto quiere decir que la transacción tiene que ser firmada por lo menos por un miembro de la organización “Org1” o por un miembro de la organización “Org2” y un miembro de la organización “Org3”.

Esta forma de especificar políticas permite representar la pertenencia a conjuntos, mediante operaciones de unión e intersección, no pudiendo especificar que una transacción no puede estar firmada por cierta organización. Esto último no sería necesario, ya que con las políticas se busca que se cumpla que un mínimo aceptable de peers de ciertas organizaciones validen la transacción. Por otra parte, esta forma de indicar las condiciones de consenso, en algunos casos puede tornarse en una política compleja de implementar, como por ejemplo la mayoría de los nodos participantes. Suponiendo una situación en la que tenemos dos organizaciones con dos nodos cada una, la mayoría se podría representar de la siguiente manera:

AND(*OR ('Org1MSP.member', 'Org2MSP.member'), OR ('Org1MSP.member', 'Org2MSP.member'), OR ('Org1MSP.member', 'Org2MSP.member'))*

Si bien esta política no es compleja de realizar, si la red comienza a crecer y llegamos por ejemplo a cincuenta miembros en cada una de ellas, sí se vuelve algo complejo para administrar.

Adicionalmente, las chaincodes pueden invocar a otras chaincodes, lo cual se puede realizar utilizando una API que provee una función específica en la cual se debe

indicar, la chaincode que se quiere invocar, los parámetros, la función y el canal en que se va a realizar la invocación, de la siguiente forma:

resultado := stub.InvokeChaincode("chaincodeA", arrayParametros, "mychannel")

Tipos de transacciones

Como se mencionó en la subsección “Transacciones”, una transacción es un mensaje firmado digitalmente que autoriza alguna acción particular asociada a la blockchain. Las transacciones [62] pueden ser de dos tipos:

- De instalación. Este tipo de transacciones representan la instalación o actualización de una chaincode en un canal.
- De invocación. Este tipo de transacciones representan la invocación a una operación de una chaincode en un canal, el cual debe de haber sido previamente instalado. El resultado de esta operación puede provocar la modificación del estado del mundo o el retorno de datos de esta chaincode en el canal. Por ejemplo, en una chaincode de lista de precios de artículos, una transacción puede representar la actualización del precio de uno de ellos.

Transacciones en Hyperledger

Como ya fue mencionado, una transacción es un mensaje que autoriza alguna acción particular asociada a la blockchain. Entre los elementos a destacar de una transacción se encuentran los siguientes:

- Identificador de la chaincode a llamar.
- Canal donde ejecuta la chaincode.
- Función a ejecutar.
- Lista de argumentos a ser utilizados por la función de la chaincode.
- Conjunto de claves de elementos leídos en la ejecución de la función.
- Conjunto de elementos escritos en la ejecución de la función.
- Conjunto de firmas realizadas por los nodos que validaron la transacción.

Nodos en Hyperledger

En Hyperledger los nodos son las entidades de la red blockchain que se comunican, siendo agrupados en dominios de confianza y asociados a entidades lógicas que los controlan. Estos ejecutan las chaincodes para realizar operaciones de escritura y lectura sobre el ledger.

Como se puede apreciar en la Figura 4.4, existen tres tipos de nodos:

- Client: envian invocaciones de transacción a los endorsers y difunden propuestas de transacción al Orderer. El endorser es el encargado de devolver la transacción recibida, junto con su firma, al Orderer.
- Peer: un nodo que crea transacciones, mantiene el estado y una copia del ledger. Puede tener un rol de endorser especial. Este solo pertenece a una única organización.
- Orderer: un nodo corriendo el servicio de comunicación que garantiza la entrega.

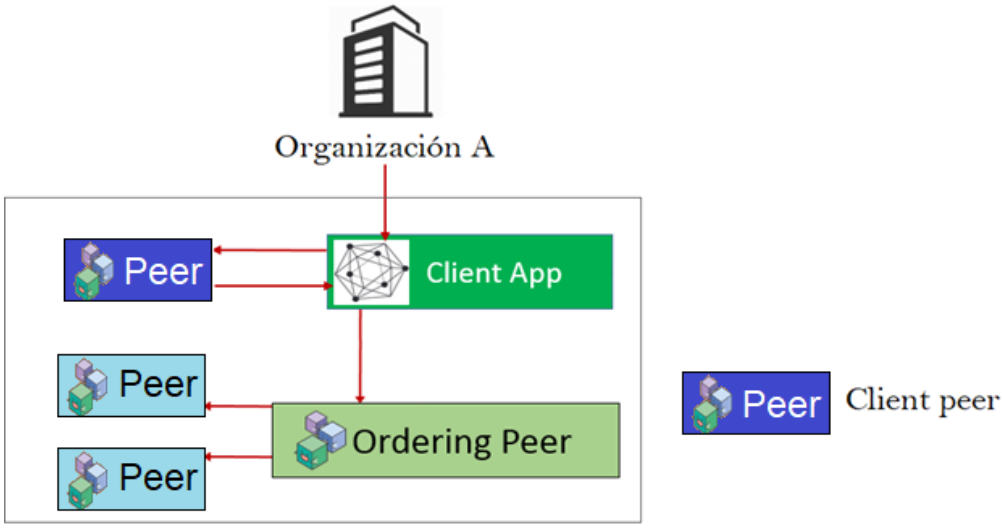

Figura 4.4. Nodos en Hyperledger

Consenso

Hyperledger Fabric en su versión 1.1, utiliza proof of authority en conjunto con políticas de validación de transacciones, por chaincode y canal, para evitar que se generen errores de gasto doble o transacciones fraudulentas. Como fue mencionado anteriormente en la subsección “Chaincode”, permite indicar qué organizaciones y cuántos peers de las mismas deben de firmar la transacción, para que esta sea válida. Además, ejecuta un algoritmo que coordina la validación de las transacciones, su postulación a ser agregada en la red blockchain y su distribución a todos los peers, el funcionamiento del mismo será descrito a continuación.

Flujo de ejecución de transacciones

Como se puede apreciar en la *Figura 4.5*, los nodos están divididos por organización y si bien se repiten en una y otra organización, lo que se comentará es cómo se da el

comportamiento no solo dentro de la misma organización, sino como fluye luego hacia otras organizaciones.

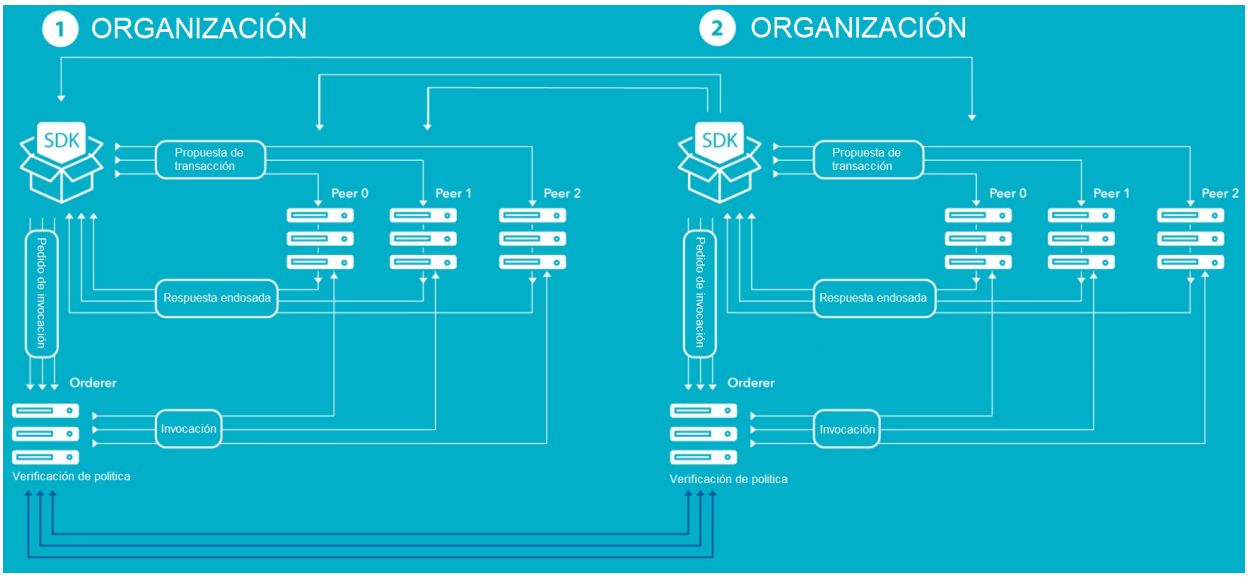

Figura 4.5 - Flujo de ejecución de transacciones [63] [64]

En esta misma figura se puede observar el componente SDK(Software Development Kit), el cual provee un entorno estructurado de librerías para que los desarrolladores puedan implementar y probar sus aplicaciones. A través de este componente es posible desarrollar software que se integre con la red blockchain. Adicionalmente cuenta con una API para el procesamiento de transacciones, membresía y manejo de eventos. Este elemento interactúa con el código del cliente recibiendo las transacciones para ser agregadas a la red blockchain. Otros elementos que podemos identificar son los peers, cuya interpretación depende de cómo se modele el caso de uso sobre el cual se va a aplicar, pero pueden ser identificados como los participantes de la red blockchain o como parte de un participante. Lo importante es que siempre van a pertenecer a una única organización. Por último, podemos identificar al Orderer.

El flujo comienza en el SDK al recibir una transacción para ser agregada en la blockchain, este componente luego envía dicha transacción a el/los peer/s para validar. Luego los peers reciben la transacción y la validan. En caso de que sea válida, envían al SDK una respuesta diciendo que la misma es válida y añadiendo una firma la cual va a ser utilizada para verificar la política. En caso de que sea inválida, envía al SDK una respuesta rechazando la transacción. El SDK al recibir las respuestas de los peers a los que envió, puede hacer validaciones a las respuestas de los peers, esto depende de la implementación que realice el cliente en base al SDK. En caso de que se cumplan las validaciones, se envía al Orderer la transacción junto con las firmas, de lo contrario la transacción es rechazada. El Orderer al recibir la transacción con las firmas, verifica que éstas cumplan la política y de ser así, ésta es almacenada en el bloque en construcción.

Luego de transcurrido un cierto tiempo configurado en la red, el Orderer cierra el bloque en proceso, generando un hash para el mismo y enviándolo a los peers de su organización y demás Orderers de la red. Estos Orderers a su vez enviarán este bloque a los peers de su organización para que actualicen la cadena. Los peers al recibir el bloque, recorren las transacciones en orden, comprobando que los conjuntos de elementos leídos provistos en la transacción, se correspondan con los del peer. En caso de que se correspondan, la transacción es marcada como válida, de lo contrario como inválida. Luego del proceso anterior, el nodo modifica el Estado del mundo ingresando o modificando los valores del conjunto de elementos escritos de las transacciones válidas.

5. Caso de uso en administración pública

En este capítulo se comentará el caso de uso elegido, a partir del cual se llevó adelante una prueba de concepto en el contexto del gobierno uruguayo, indicando los pasos realizados y el alcance para el mismo. Este caso de uso fue elegido por sobre otros, ya que ataca una problemática actual, que no necesariamente aplica para otros casos analizados, además se obtuvo asistencia de un profesional (escribano) para evacuar cualquier duda relacionada. A continuación, se detalla el caso de uso y el aporte que genera blockchain producto de su aplicación.

5.1. Caso de uso: empadronamiento vehicular

Actualmente el empadronamiento vehicular se realiza de forma independiente en cada intendencia del país, suponiendo un beneficio económico directo para cada una de ellas. Los escribanos como parte del procedimiento de validación de la documentación para la compraventa de un vehículo deben analizar que los datos de empadronamiento de ese vehículo sean consistentes en el tiempo. Para ello deben revisar el histórico de empadronamiento de ese vehículo, pero esta tarea se realiza consultando esa información en todas las intendencias en las que fue empadronado y de forma presencial.

Antes de registrar un vehículo en la intendencia, los escribanos deben verificar que los datos de anteriores empadronamientos sean consistentes con los registrados en cada intendencia, siendo estos datos: el número de chasis, motor, cambios de motor y propietarios; verificando además que no tenga multas y que no esté empadronado en otra intendencia.

Los datos mínimos que se registran para cada vehículo son el número de padrón, número de chasis, número de motor, marca, modelo, tipo, dueño y si tiene deuda correspondiente al empadronamiento o multas.

Existen varios actores que participan de este caso como son las 19 intendencias departamentales, los escribanos que realizan las consultas y la policía de tránsito (que solo existe en Montevideo) y policía caminera (a nivel nacional) para el manejo de multas.

5.2. Validación de aplicabilidad

Como se mencionó anteriormente, el control de empadronamiento de un vehículo, que no esté empadronado en múltiples intendencias al mismo tiempo, la morosidad del vehículo y la comprobación de la historia completa de cambios de motor del vehículo,

es llevado a cabo por los escribanos, lo cual deja cabida al error humano. En base al conocimiento adquirido, blockchain agregaría valor a la solución ya que permitiría hacer estas validaciones de forma automática al dar de alta o modificar un empadronamiento y además registraría en la misma las consultas realizadas por los escribanos, haciéndolo de forma inalterable para cualquiera de estas transacciones. Esto daría sustento a la documentación elaborada por los escribanos.

Por otra parte, la necesidad de que el escribano deba consultar en cada intendencia la información de los empadronamientos, es debido a que ésta se encuentra únicamente en la intendencia en que fue empadronado. La utilización de blockchain en este caso generaría que la información se encontrase distribuida entre todos los participantes, facilitando las consultas por este tipo de información y bajando sus costos asociados, por ejemplo, el trámite tiene un costo aproximado de $800 en Montevideo. Adicionalmente se podrían sumar costos de traslado a cada intendencia o contratación de gestores para la realización del trámite.

Otro aspecto importante es que la información de los empadronamientos es utilizada por los escribanos para la confección de documentos que tienen validez legal, por lo que, para evitar consecuencias de esta índole, se deberían basar en información confiable e incorruptible. A modo de ejemplo, cuando el escribano prepara la documentación puede que el vehículo no presente multas, pero que luego de unas semanas, surja un registro de una multa con una fecha anterior a la de elaboración del documento. Utilizando blockchain, cada vez que un escribano realice una consulta, se puede generar un registro que indique la fecha de realización de esa consulta, permitiendo verificar la veracidad del documento contra ese registro.

Es importante entender si para este caso, una blockchain es aplicable, agrega valor y en cuyo caso, qué tipo de blockchain utilizar. Para determinar esto se relevaron varios cuestionarios, de los cuales se eligieron dos, dado que son consistentes con el conocimiento adquirido y siguen un esquema similar de preguntas. Estos cuestionarios ayudan a entender si la tecnología blockchain es aplicable o no y en caso afirmativo, indican el tipo de blockchain que se debería aplicar. Adicionalmente, durante el proceso de ejecución de estos cuestionarios, es posible reconocer si aplicar blockchain aporta valor o no.

5.2.1. Primera validación

La primera validación fue elaborado en base al cuestionario que se plantea en “Do you need a blockchain?” [65], el cual es referenciado por otros artículos como “What Blockchain Alternative Do You Need?” [66] o sitios web como Medium [67], para la elaboración de sus propios cuestionarios. Este documento analiza blockchain como solución técnica para distintos escenarios, contrastando sus propiedades contra las de una base de datos administrada centralmente y proporcionando una metodología

estructurada para determinar la solución técnica adecuada para resolver un problema de aplicación en particular. Para ello presenta como herramienta el cuestionario presentado en la *figura 5.1*.

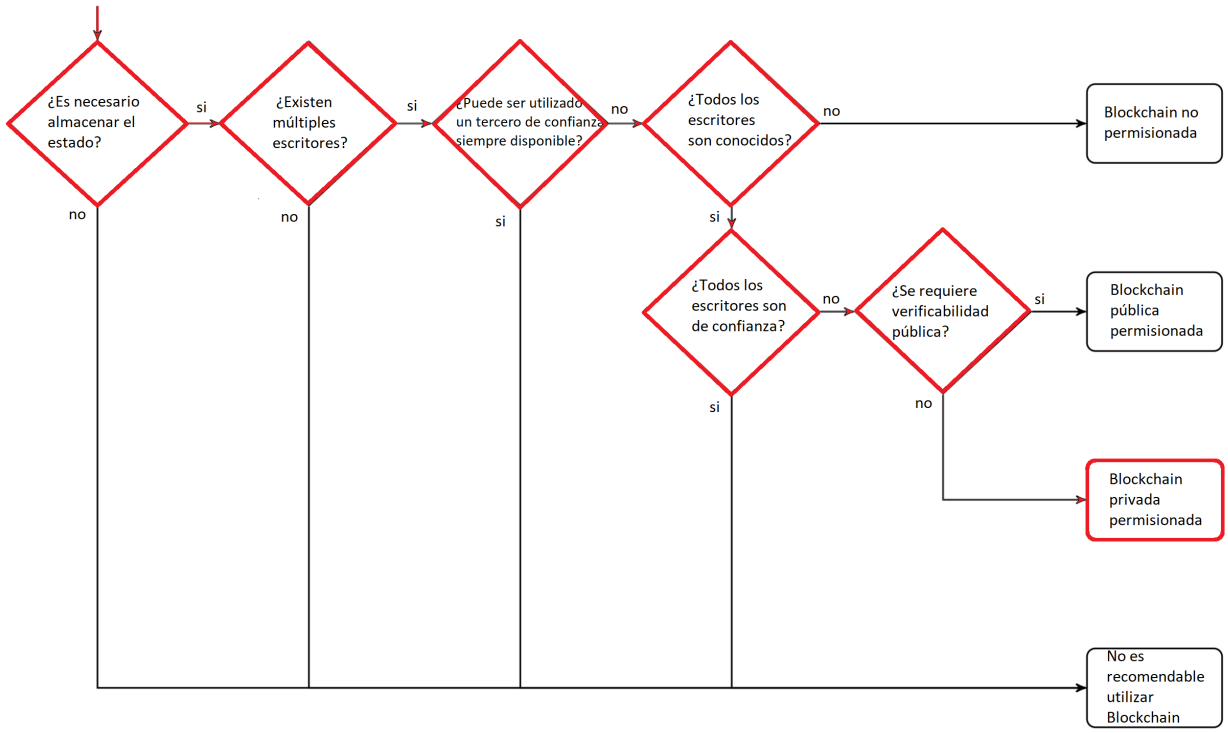

Figura 5.1 - Cuestionario "Do you need a blockchain?" [65]

En la *tabla 5.2* se presentan las preguntas junto a sus respuestas para el cuestionario.

Pregunta	Respuesta
¿Es necesario almacenar el estado?	Sí, dado que los empadronamientos pueden cambiar y se deben guardar estos cambios.
¿Existen múltiples escritores?	Sí, dado que cada intendencia puede ingresar y editar los empadronamientos.
¿Puede ser utilizado un tercero de confianza siempre disponible?	No. Aunque en la actualidad no existe una entidad central de confianza y el caso podría resolverse creando dicha entidad, no es un escenario posible dada la realidad política.
¿Todos los escritores son conocidos?	Sí, dado que en este caso los participantes que escriben son las intendencias, por lo que se encuentran plenamente identificados.
¿Todos los escritores son de confianza?	No, dado que el registro de empadronamiento supone un beneficio económico directo para cada intendencia de forma independiente. Esto genera un ecosistema de competencia entre los participantes y por tanto desconfianza entre ellos.
¿Se requiere verificabilidad	No, dado que quienes deben validar son únicamente las

pública?	intendencias.

Tabla 5.2 - Respuestas a cuestionario "Do you need a blockchain?" [65]

Al seguir el flujo del cuestionario, este indica que para este caso es posible solucionarlo aplicando blockchain, utilizando una blockchain privada permisionada.

5.2.1. Segunda validación

Como segundo cuestionario, se utilizó el provisto por el siguiente estudio "Blockchain Beyond the Hype A Practical Framework for Business Leaders" [68], el cual es referenciado por documentos como "Can Blockchain revolutionize international trade?" [69] y "Distributed Ledger Technology, Blockchains and Identity" [70], para comentar la importancia de identificar si es beneficioso el uso de la tecnología blockchain o no en distintos contextos. Al igual que el anterior es presentado como un flujo de preguntas, en donde a partir de la respuesta de cada uno se direcciona a la siguiente pregunta o al resultado del mismo, como se ve en la *Figura 5.3*.

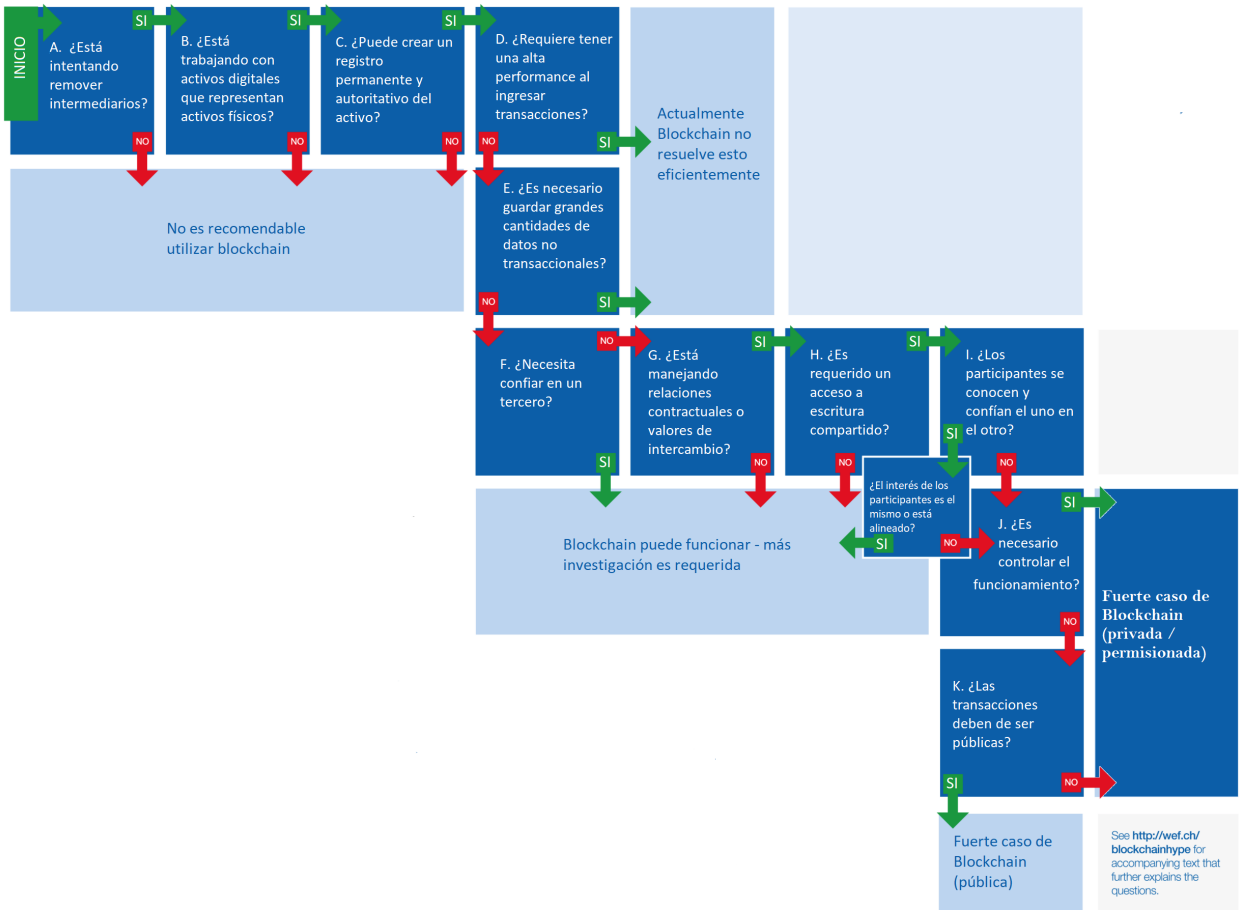

Figura 5.3 - Cuestionario "Blockchain Beyond the Hype A Practical Framework for Business Leaders" [68]

En la *tabla 5.4* se presentan las preguntas junto a sus respuestas para el cuestionario.

Pregunta	Respuesta
¿Está intentando remover intermediarios?	Sí, en el caso planteado no existe un intermediario, pero se busca evitar la creación de uno.
¿Está trabajando con activos digitales que representan activos físicos?	Sí, dado que se desea trabajar sobre la representación digital del empadronamiento de vehículos.
¿Puede crear un registro permanente y autoritativo del activo?	Sí, ya que justamente es uno de los objetivos que se buscan.
¿Requiere tener una alta performance al ingresar transacciones?	No, dado que es información que tiene pocas variaciones en el tiempo.
¿Es necesario guardar grandes cantidades de datos no transaccionales?	No.
¿Necesita confiar en un tercero?	No, si bien no queda claro a qué apunta esta pregunta, está claro que las intendencias en el caso no necesariamente confían en la información agregada por las otras, como fue explicado en el cuestionario anterior.
¿Está manejando relaciones contractuales o valores de intercambio?	Sí, dado que se están manejando empadronamientos de vehículos.
¿Es requerido un acceso a escritura compartido?	Sí, dado que cada intendencia debe poder agregar sus empadronamientos
¿Los participantes se conocen y confían el uno en el otro?	Si bien cada intendencia conoce a la otra, entendemos que este punto puede estar sujeto a más de una interpretación, ya que en cuanto a la información ingresada por cada intendencia, puede considerarse que confían o no. En cualquier caso, tanto si la respuesta es negativa como afirmativa, se terminará en la misma pregunta (J), ya que en caso de contestar afirmativamente, la siguiente pregunta sería si las intendencias tienen un interés común o están alineados y en este punto la respuesta es claramente negativa.
¿Es necesario controlar el funcionamiento?	Si bien no queda claro a qué nivel de control se está refiriendo, en principio la respuesta sería negativa. De cualquier forma, en este punto queda claro que es un escenario donde se puede aplicar blockchain, de acuerdo a los caminos alternativos que quedan.
¿Las transacciones deben de ser públicas?	No, dado que actualmente esta información es privada y solo puede ser accedida si se tiene el número de empadronamiento y la matrícula del vehículo.

Tabla 5.4 - Respuestas a cuestionario "These 11 questions will help you decide if blockchain is right for your business" [68]

Por tanto, este cuestionario también indica que es posible aplicar blockchain y adicionalmente indica que sería un tipo de blockchain privada o permisionada.

Es necesario destacar que se identificaron preguntas de los cuestionarios que no siguen un enfoque técnico, por lo que es recomendable que el cuestionario sea realizado por un equipo multidisciplinario. Relacionado con este aspecto, podemos encontrar en ambos cuestionarios preguntas relacionadas con la confianza entre los participantes, como por ejemplo “¿Todos los escritores son de confianza?” o “¿Los participantes se conocen y confían el uno en el otro?”. La respuesta para cualquiera de estas preguntas no fue simple, ya que o bien faltaba conocimiento de la realidad o bien la pregunta era ambigua y se debieron evaluar distintos enfoques para la misma. La forma cómo se resolvieron y respondieron estas preguntas, fueron comentadas en cada respuesta de la Tabla 5.2 y Tabla 5.4.

Igualmente, como se puede apreciar, ambos cuestionarios validan la aplicabilidad de utilizar la tecnología blockchain para implementar el caso de uso y además, estos indican que la blockchain debe de ser una de tipo privada. Por tanto, se prosigue con el análisis para resolver el caso de uso, utilizando Hyperledger Fabric cómo framework, ya que como se vio anteriormente en la sección “4.2 Hyperledger Fabric” este permite la implementación de redes de blockchain privadas permisionadas.

6. Arquitectura de la solución

En este capítulo se presentará la arquitectura propuesta para la solución del caso de uso planteado de acuerdo al alcance mencionado en la sección “5.1. Caso de uso: empadronamiento vehicular”. Para representar la arquitectura se tomó como referencia el modelo 4+1 [71], para el cual se desarrollarán las vistas lógica, física y un escenario.

El modelo 4+1 propone separar los múltiples aspectos de una arquitectura de un sistema, en múltiples vistas. Estas vistas describen el sistema desde el punto de vista de diferentes interesados, como usuarios, desarrolladores o directores de proyecto. Las cuatro vistas son:

- Vista lógica: enfocada en describir la estructura y funcionalidad del sistema.
- Vista física: enfocada en describir los componentes físicos, cómo se comunican y la distribución del software dentro de ellos.
- Vista de proceso: enfocada en los aspectos dinámicos del sistema, explica los procesos del sistema, como se comunican, considerando aspectos de concurrencia, distribución, rendimiento, escalabilidad, etc.
- Vista de desarrollo: enfocada en describir el sistema desde el punto de vista de un programador, describiendo la organización estática del software en el entorno de desarrollo.

Las cuatro vistas anteriores se complementan con una selección de casos de uso o escenarios que ilustran la arquitectura como una vista más.

Como se mencionó anteriormente, las vistas elegidas fueron la lógica y física, esto dado que se considera necesario definir las funcionalidades del sistema y cómo estas interactúan con la red blockchain. A su vez, también se considera necesario mostrar la distribución del sistema y en qué lugar está posicionada la blockchain. Finalmente se presenta un escenario, el cual es necesario para mostrar la distribución de la solución conteniendo un conjunto reducido de organizaciones.

6.1. Vista lógica

En esta vista, el modelo 4+1, presenta una variedad de diagramas que pueden ser utilizados, para este caso de negocio se decidió realizar un Diagrama de casos de uso, el cual presenta los distintos casos de uso que presenta la implementación del caso de negocio, un Modelo de dominio para describir la estructura de los empadronamientos y tres diagramas de secuencias, los cuales muestran cómo interactúan los distintos componentes para los casos de uso: “Alta de empadronamiento” y “Consulta de empadronamiento”.

En la *figura 6.1* se muestra, el diagrama de casos de uso, indicando los distintos tipos de actores y las funciones que cada uno puede ejecutar.

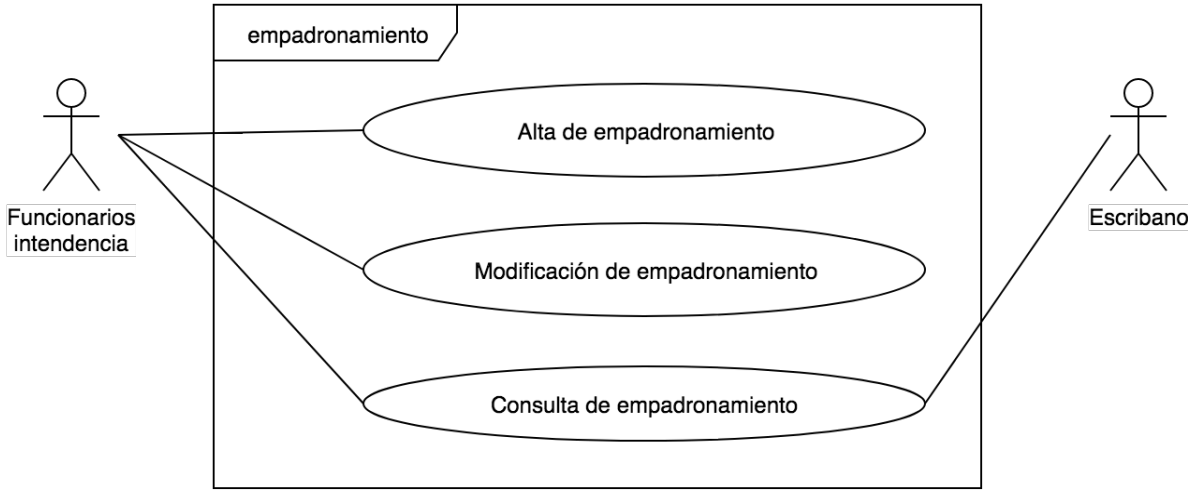

Figura 6.1. Diagrama de casos de uso

Actores:

- Funcionarios intendencia: estos corresponden a los usuarios de las intendencias, los cuales están encargados de ingresar y mantener la información sobre los empadronamientos.
- Escribanos: si bien está enfocado a escribanos, también puede ser cualquier ciudadano que desea consultar un empadronamiento.

Casos de uso:

- Alta de empadronamiento: al ejecutarse, el funcionario de una intendencia da de alta un empadronamiento para un vehículo, controlando que no encuentre en el sistema este empadronamiento.
- Modificación de empadronamiento: al ejecutarse, el funcionario de una intendencia modifica el estado de un empadronamiento ya existente en el sistema.
- Consulta de empadronamiento: al ejecutarse, al escribano se le presenta una lista con el estado actual del empadronamiento para el número de padrón e intendencia correspondiente, junto con todos los empadronamientos del vehículo correspondiente al empadronamiento.

Para la implementación de estos casos de uso, fueron desarrolladas dos chaincodes con características diferentes funcionando en simultáneo. Una de ellas(“empadronamiento”) tiene el propósito de manejar todo lo referente al alta y modificación sobre los empadronamientos, así como realizar las distintas consultas sobre ellos. La otra chaincode(“historico_empadronamiento”) es utilizada para realizar las consultas de empadronamiento, registrando en ella una marca de tiempo de cuando fue realizada dicha consulta, junto con los datos devueltos por la misma. Esta separación fue realizada de esta forma debido a que las políticas de escritura se

definen por chaincode-canal, como fue mencionado en “4.2 Hyperledger Fabric”. Por tanto, de no separarse en chaincodes distintas, se estaría permitiendo a escribanos ejecutar casos de uso cuando no deberían.

En la *figura 6.2* se presenta el modelo de dominio de los elementos que corresponden al empadronamiento.

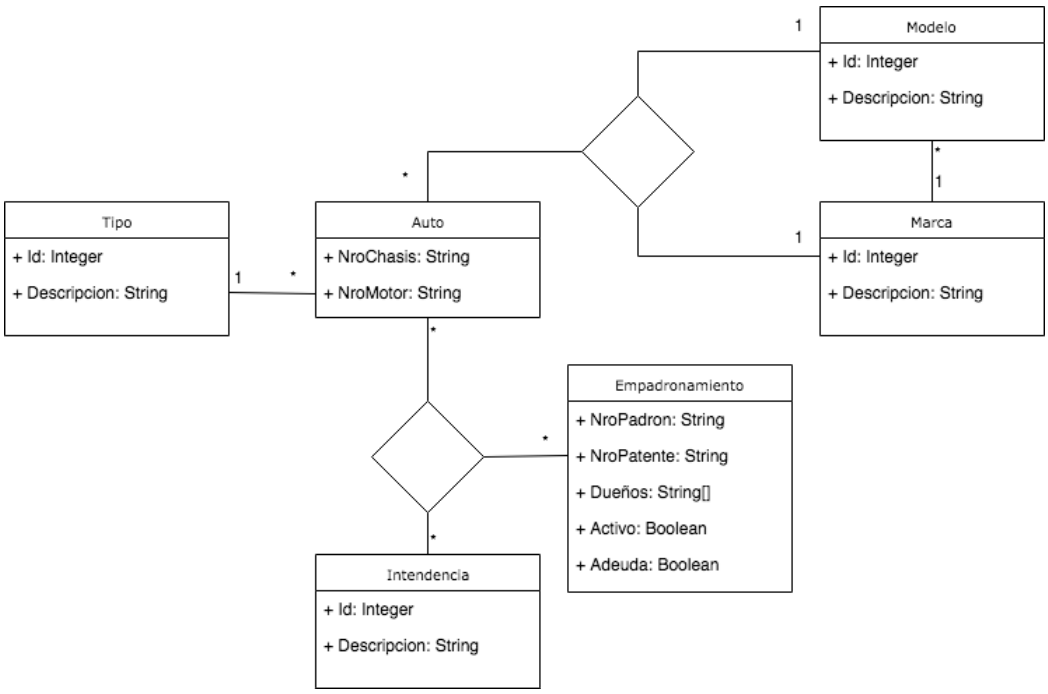

Figura 6.2. Modelo de dominio

Este modelo representa las distintas propiedades que poseen los autos, en donde se identifica que un auto tiene: número de chasis, número de motor, tipo, marca y modelo, a su vez se presentan las características que tienen los autos empadronados en una intendencia, las cuales son: número de padrón, número de patente, dueños registrados para ese empadronamiento, si está activo el empadronamiento y si tiene alguna deuda.
Restricciones no estructurales:

- Un auto solo puede estar empadronado en una intendencia a la vez

A continuación, se presentan los diagramas de secuencia para los casos de uso Consulta de empadronamiento y Alta de empadronamiento, los cuales presentan los componentes: Aplicación Web, utilizada por los usuarios para interactuar con el sistema, el SDK que recibe las peticiones de la Aplicación Web y se encarga de interactuar con la red blockchain y Blockchain la cual es una instancia de blockchain ejecutando en un nodo.

La figura 6.3 presenta el diagrama de secuencia para la consulta de empadronamiento por número de padrón y dueño. En ella se puede apreciar que el SDK luego de recibir

el llamado de la función “consultaEmpadronamiento” por parte de la Aplicación Web, realiza un llamado a un nodo de la red blockchain y queda a la espera del resultado de este, para luego enviar ese resultado a la Aplicación Web.

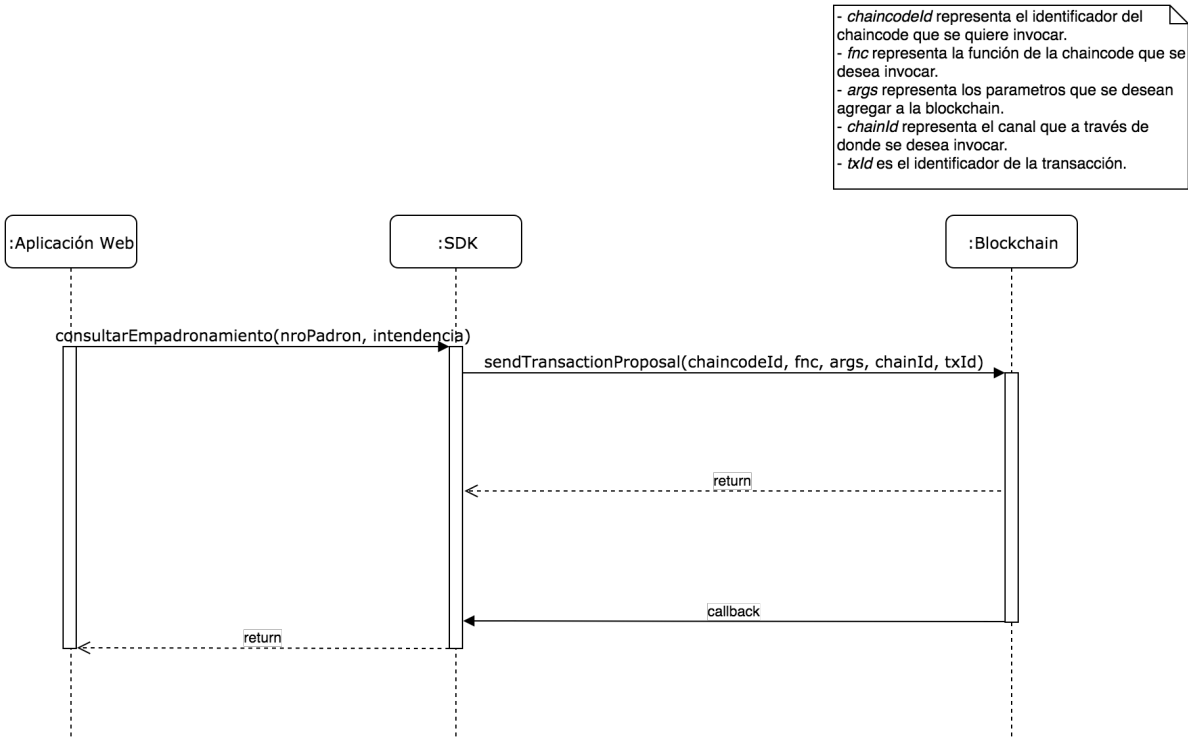

Figura 6.3. Diagrama de secuencia de Consulta de empadronamiento

Una vista más detallada sobre los llamados internos de la blockchain es presentada en la figura 6.4. En esta se puede apreciar que la blockchain luego de recibir el llamado de la función “sendTransactionProposal”(encargada de realizar propuestas de transacción para su validación en la blockchain) por parte del SDK, realiza una invocación a la chaincode “historico_empadronamiento”, mediante la función “Invoke”(función utilizada para realizar los llamados a las funciones de las chaincodes) y esta luego realiza una invocación a la chaincode “empadronamiento”, utilizando la función “InvokeChaincode”(función para realizar llamados entre chaincodes).

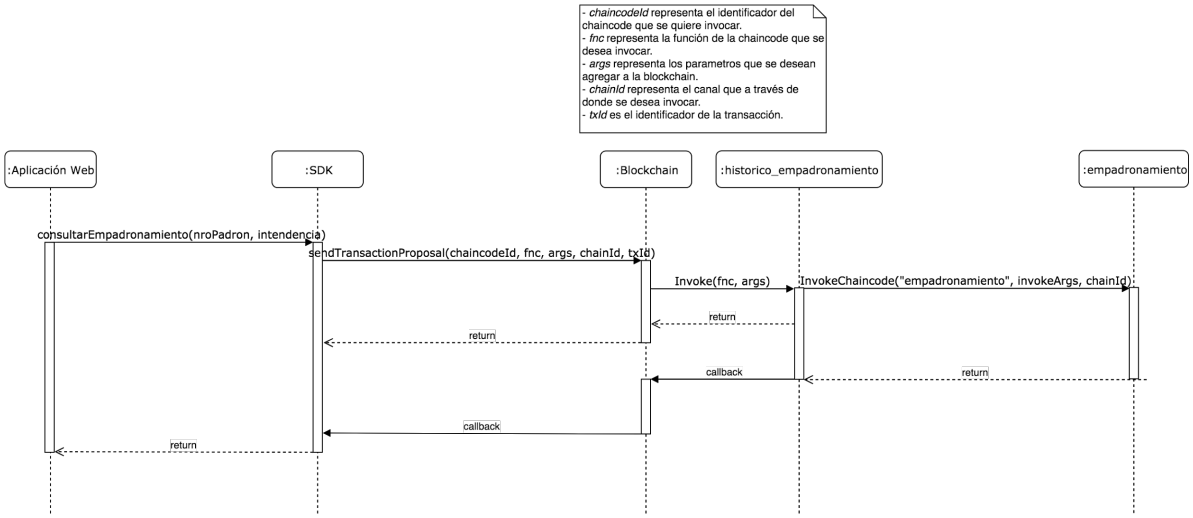

Figura 6.4. Diagrama de secuencia de Consulta de empadronamiento extendido

En la *figura 6.5* se podrá apreciar el diagrama de secuencia para el alta de un empadronamiento. Se entiende que el diagrama de secuencia para modificar un empadronamiento no aporta valor dado que es en su mayoría el mismo comportamiento que el declarado para dar de alta un empadronamiento.

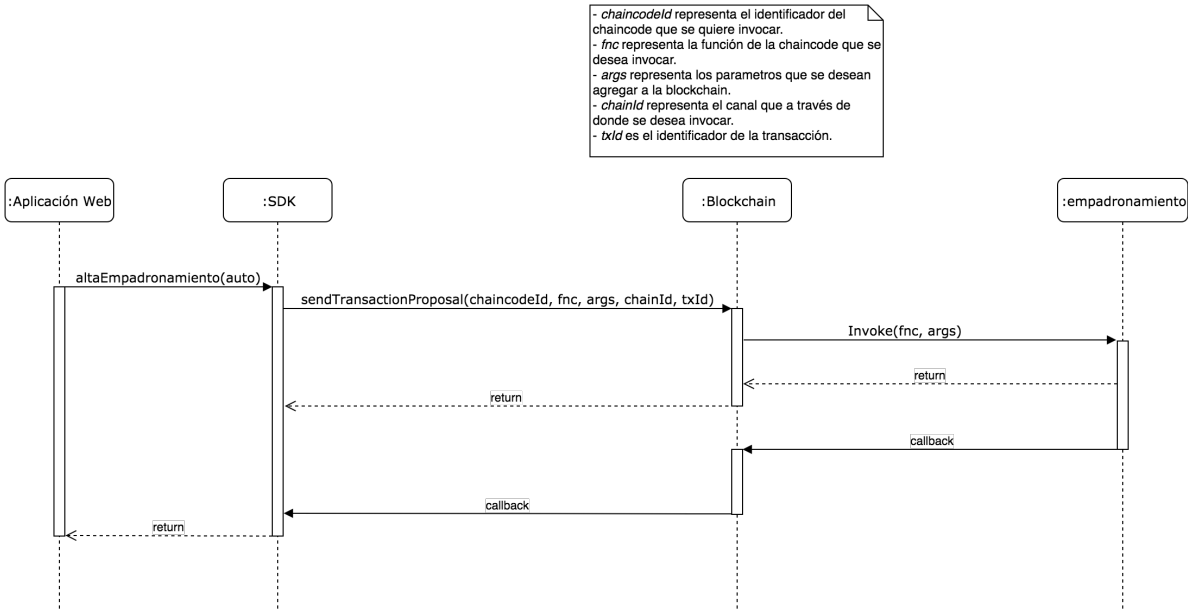

Figura 6.5. Diagrama de secuencia de Alta de empadronamiento

Luego, en la *figura 6.6* podemos ver el diagrama de comunicación asociado al alta de empadronamiento. Nuevamente podemos apreciar que el diagrama para la modificación de empadronamiento no aporta gran valor, por ser igual que el de agregar un empadronamiento.

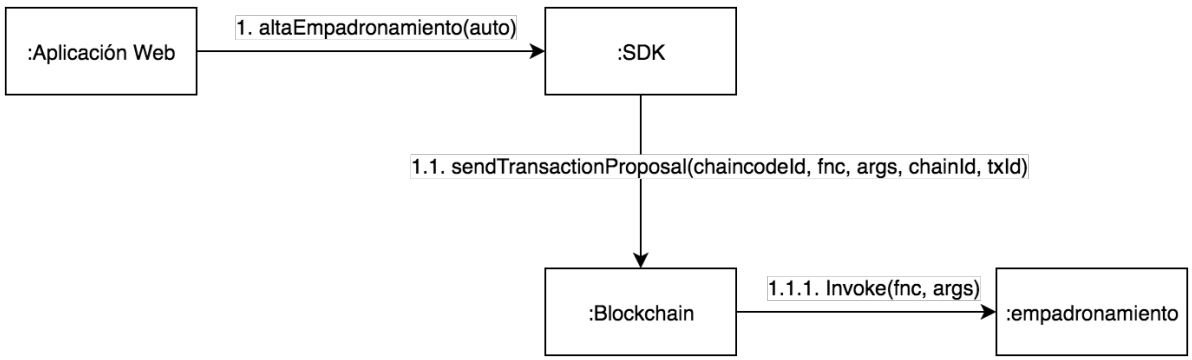

Figura 6.6. Diagrama de comunicación de Alta de empadronamiento

6.2. Vista física

En esta vista, se realizó un diagrama de despliegue asociado a la solución, el cual representa la distribución de la solución en cada intendencia. Esta vista sugiere contar con más de un nodo de la blockchain por organización, de forma de lograr alta disponibilidad. Esta decisión fue tomada ya que en las pruebas realizadas, al momento de agregar un nodo a una red ya existente, este demoraba varios minutos en quedar operacional, aunque la red contaba con pocas transacciones. Esto se da porque al momento de restablecer un nodo, se ejecutan todas las transacciones desde el origen, además de las subscripciones a los canales e instalación de las chaincodes pertinentes.

Por otra parte, para no depender de un protocolo específico de comunicación, se estandariza la comunicación hacia la red blockchain a través de un proxy, el cual implementa una API basada en REST que consume las operaciones provistas por el SDK, permitiendo acoplarse fácilmente con cualquier otro componente, dado que traduce la complejidad del llamado a operaciones en la red blockchain, por un llamado REST.

De forma que cada organización pueda contar con una interfaz de usuario, ya sea para funcionarios o escribanos, se ofrece una aplicación web que ya brinda las funcionalidades mencionadas en la sección “6.1. Vista lógica”, con la posibilidad de acceder al sistema ya sea a través de la aplicación o desde la capa de servicios definida.

En la figura 6.7 se puede ver cuáles componentes son desplegados en cada organización, donde cada una está conformada por: dos nodos, cada uno con su copia de la blockchain para dar alta disponibilidad; un servidor de aplicaciones el cual

contiene la aplicación web con la que interactúa el usuario; un servidor web el cual contiene el proxy para la interacción entre la Aplicación Web y la red blockchain.

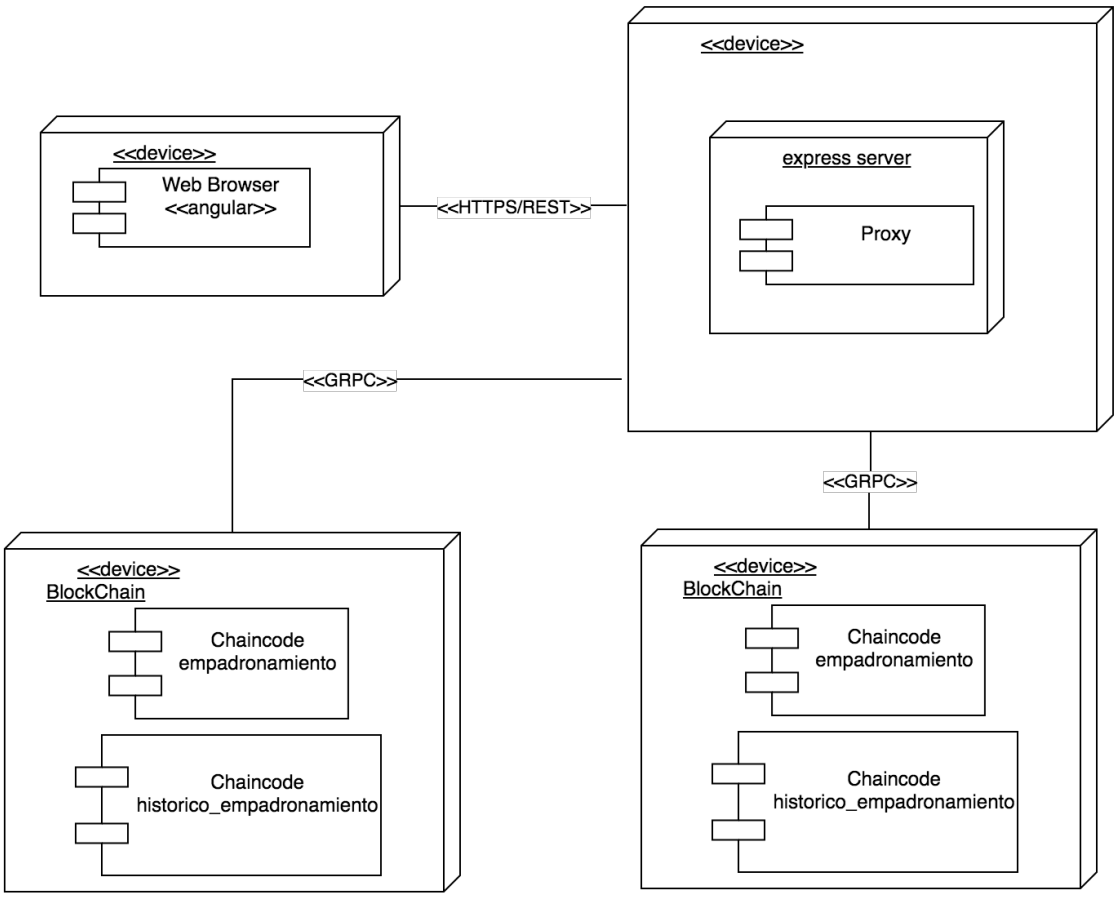

Figura 6.7. Diagrama de despliegue de una organización

6.3. Escenario

Como se muestra en la *figura 6.8*, el despliegue de la solución propuesto está conformada por:

- Tres organizaciones: dos intendencias y una para las consultas de escribanos
- Un Orderer

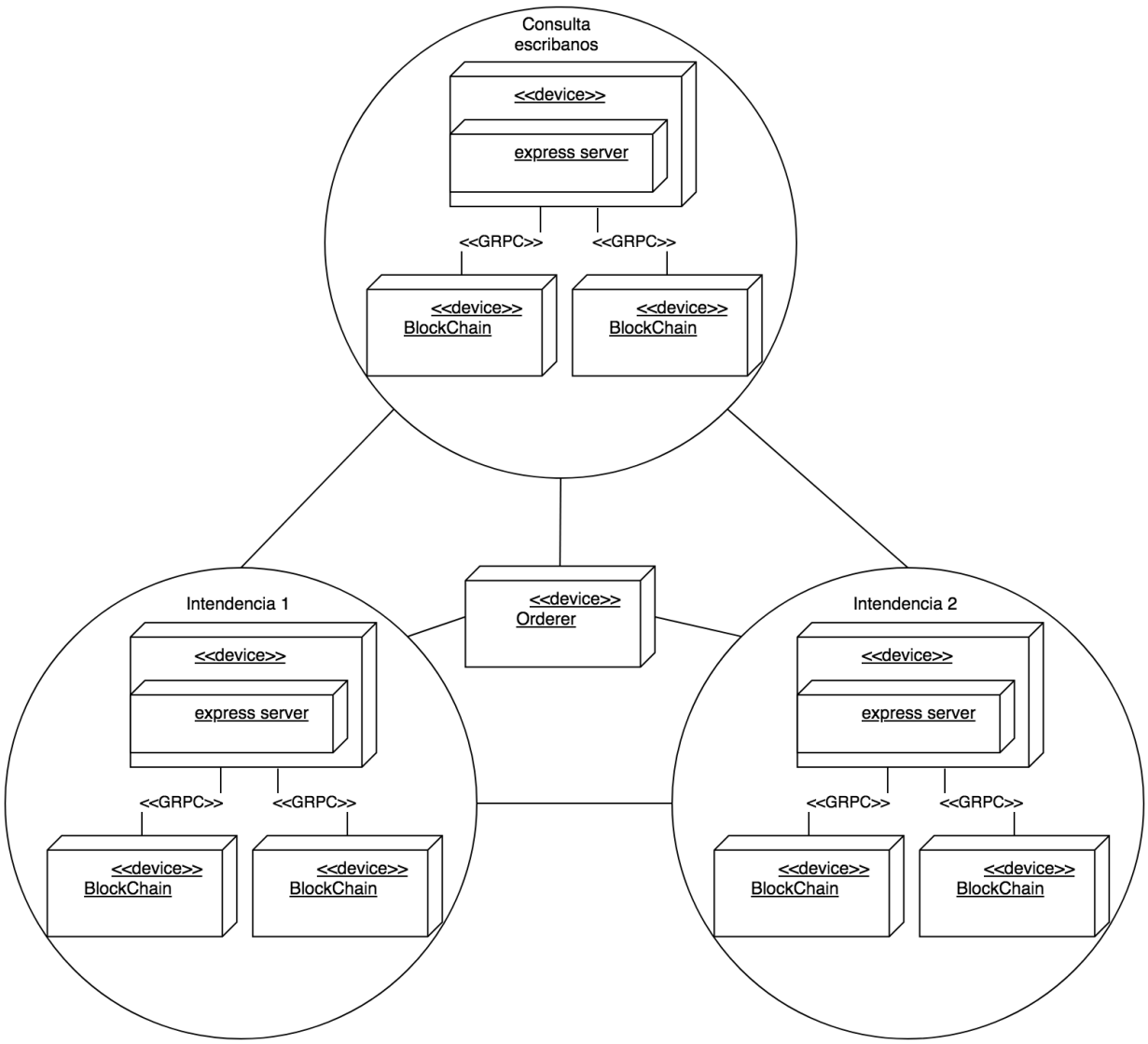

Figura 6.8. Diagrama de despliegue de la red

La red antes mencionada puede ser extendida fácilmente agregando más organizaciones, la cual simularía la adhesión de nuevas intendencias a la red. En el momento en que una nueva intendencia se registra en la red, los nodos pertenecientes a esa intendencia se conectan con el Orderer, para recibir los bloques existentes en la blockchain al momento. Luego de esto, los nodos comienzan a ejecutar las transacciones incluidas en los bloques en orden, hasta que reconstruyen el estado actual de la blockchain. Adicionalmente, el algoritmo de consenso debe de ser actualizado en el Orderer para incluir a la nueva intendencia, para así cumplir el consenso definido en la sección “7.2. Aspectos de la implementación”. Finalmente, la intendencia queda lista para comenzar a enviar transacciones a la blockchain.

7. Implementación

En este capítulo se presentan las tecnologías y las herramientas utilizadas para el desarrollo de la prueba de concepto. Además, se explican los problemas encontrados y las decisiones técnicas tomadas en base a ello, así como también son comentadas las limitantes encontradas.

7.1. Tecnologías utilizadas

En esta sección se describirán las distintas tecnologías utilizadas para la implementación del caso de uso presentado en el capítulo “5. Caso de uso en administración pública”.

Express.js

Express.js [72] es un framework para realizar aplicaciones web utilizando JavaScript dentro de Node.js [73]. Este es de código abierto bajo la Licencia MIT. Está diseñado para construir aplicaciones web y APIs, y fue utilizado para implementar el Proxy. Se utilizó esta tecnología dado que se contaba con conocimiento previo y adicionalmente es utilizada en los ejemplos provistos por Hyperledger.

Angular

Angular [74] es un framework para aplicaciones web, desarrollado en TypeScript, de código abierto y mantenido por Google. Se utiliza para crear y mantener aplicaciones web de una sola página. El mismo es ejecutado en el navegador web del cliente y se enfoca en el patrón de diseño Model View Controller (MVC o Modelo Vista Controlador), en un esfuerzo para hacer que el desarrollo y pruebas sean más fáciles. Este fue utilizado para realizar la aplicación web que se comunica con el servidor en Express.js y fue elegido dado que se contaba con conocimiento previo y experiencia en su utilización.

Docker

Docker [75] es un proyecto de código abierto el cual automatiza el despliegue de aplicaciones dentro de contenedores de software, los cuales son paquetes de elementos, tales como librerías, ejecutables, sistema operativo, servidores de aplicaciones, servidores de bases de datos, etc, que permiten ejecutar aplicaciones en cualquier sistema operativo. Esto proporciona una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos. Este fue utilizado dado que los nodos de Hyperledger Fabric se encuentran en contenedores Docker.

Blockchain Explorer

Blockchain Explorer [76] es una herramienta desarrollada por Hyperledger para poder observar la composición de la red(PeerList), y principalmente el estado de la blockchain(BlockList), como se muestra en la *Figura 7.1*.

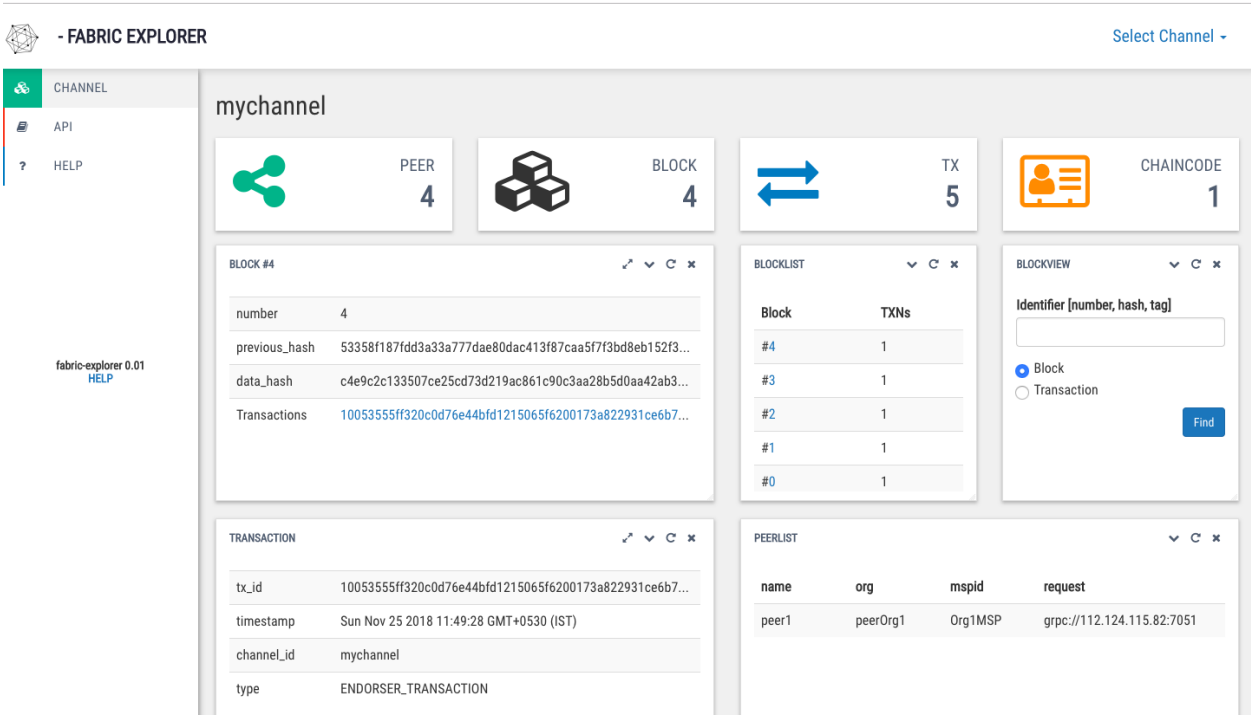

Figura 7.1. Componentes y estado de blockchain en Blockchain Explorer

Para la implementación del caso de uso fue necesario contar con una herramienta que asistiera para verificar el funcionamiento y la configuración de la red. En este caso esta herramienta fue la única encontrada, completamente integrada con Hyperledger Fabric. La utilización de la herramienta permitió analizar en profundidad la estructura de las transacciones y de los bloques, asistiendo en la comprensión de lo planteado por los manuales de Hyperledger Fabric. Adicionalmente, fue utilizada para observar qué nodos se encontraban funcionando correctamente y cuáles no.

7.2. Aspectos de la implementación

En esta sección se presentan los aspectos de implementación y configuración, tomados en cuenta para el desarrollo del caso de uso.

Repositorio de código

Se utilizó el servicio GitLab [77] provisto por la Facultad de Ingeniería para alojar el código fuente de la solución en un repositorio Git. Debido a que los lenguajes utilizados fueron JavaScript, TypeScript y Golang, se optó por utilizar Visual Studio Code [78]

como entorno de desarrollo. Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.

Sistema operativo

La etapa de investigación del framework comenzó utilizando dos sistemas operativos, macOS High Sierra 10.13.6 y Ubuntu 16.04, este último dentro de una máquina virtual. En el transcurso de esta investigación, se encontraron diferencias en versiones de tecnologías (por ejemplo Docker) o componentes de Hyperledger (por ejemplo Blockchain Explorer), generando resultados diferentes en la ejecución, por lo que para evitar diferencias al momento de la implementación de la prueba de concepto, se continuó utilizando únicamente la máquina virtual de Ubuntu.

Chaincodes

Para la implementación de las chaincodes se utilizó el lenguaje de programación Golang, el cual es el recomendado por Hyperledger y está configurado de forma estándar en las redes de ejemplo.

Orderer

Se utilizó el componente Orderer con el perfil Solo. Esto es debido a que en las pruebas realizadas en la etapa de investigación se observó que la configuración Kafka presenta el mismo comportamiento que la configuración Solo e implica un mayor gasto de recursos de la máquina virtual sobre la que se ejecuta. Si bien se menciona en la sección “4.2 Hyperledger Fabric”, que la configuración recomendada para producción es la de Kafka, se optó por la configuración Solo, ya que de acuerdo al objetivo del proyecto se entiende que un despliegue como este es suficiente.

Tolerancia ante fallas y seguridad

Como se mencionó en la sección “4.2 Hyperledger Fabric”, Hyperledger Fabric aún no posee medios de protección contra el problema de los generales bizantinos, por lo que su tolerancia a fallas posee ciertas deficiencias. Un ejemplo de esto fue encontrado mediante la realización de pruebas, cuando se modifican los datos de la clave de un empadronamiento directo en la base de datos del estado del mundo de un nodo. Si se realiza un cambio en ese empadronamiento mediante una transacción y la transacción es originada en este nodo, este la aceptará, mientras que los demás nodos no. En caso contrario, si la transacción es originada en cualquiera de los demás nodos, estos actualizarán sus datos, mientras que el nodo con las modificaciones no. En ambos casos, el nodo que fue modificado queda desfasado de los demás, no pudiendo ser actualizado por las transacciones de los demás nodos y tampoco surtiendo efecto sus transacciones. Esto representa un problema, dado que ese nodo queda inutilizable, no

pudiendo participar del consenso y por lo tanto afectando a la seguridad de la red, dado que una red de blockchain es más segura cuantos más nodos pertenezcan a ella, como fue mencionado en la sección “2.1.1. Conceptos de blockchain”. Para resolver este problema se debería detener el nodo y volver a iniciar, éste se conectaría al Orderer y ejecutaría todas las transacciones en orden, quedando operacional nuevamente.

Consenso

Para el caso de uso se configuró como política de escritura de la chaincode “empadronamiento”, que para que una transacción de escritura sea válida, la misma debe de estar validada y firmada, por lo menos, por un nodo de cada intendencia. De esta forma, de acuerdo al flujo de ejecución de transacciones visto en el capítulo “4. Plataformas de blockchain”, a la hora de dar de alta un empadronamiento, se siguen estos pasos:

1. La REST API utilizando el SDK envía a un nodo de cada intendencia una propuesta de transacción, conteniendo el llamado a la chaincode ”empadronamiento”.
2. Los nodos al recibir esta propuesta de transacción, ejecutan la misma y envían al SDK, la transacción firmada y con el resultado de la misma.
3. El SDK al recibir esto, verifica que las respuestas sean correctas. De ser así, se envía la transacción al Orderer, junto con las respuestas y sus firmas.
4. El Orderer al recibir esta transacción, verifica que las firmas enviadas cumplan la política definida.
5. Si se cumple la política, el Orderer agrega esta transacción al bloque en construcción. Una vez finalizado el bloque, este será distribuido a todos los nodos de la red.

En caso de que las respuestas de los nodos sean negativas o que las firmas no alcancen a cumplir las políticas, la transacción es descartada.

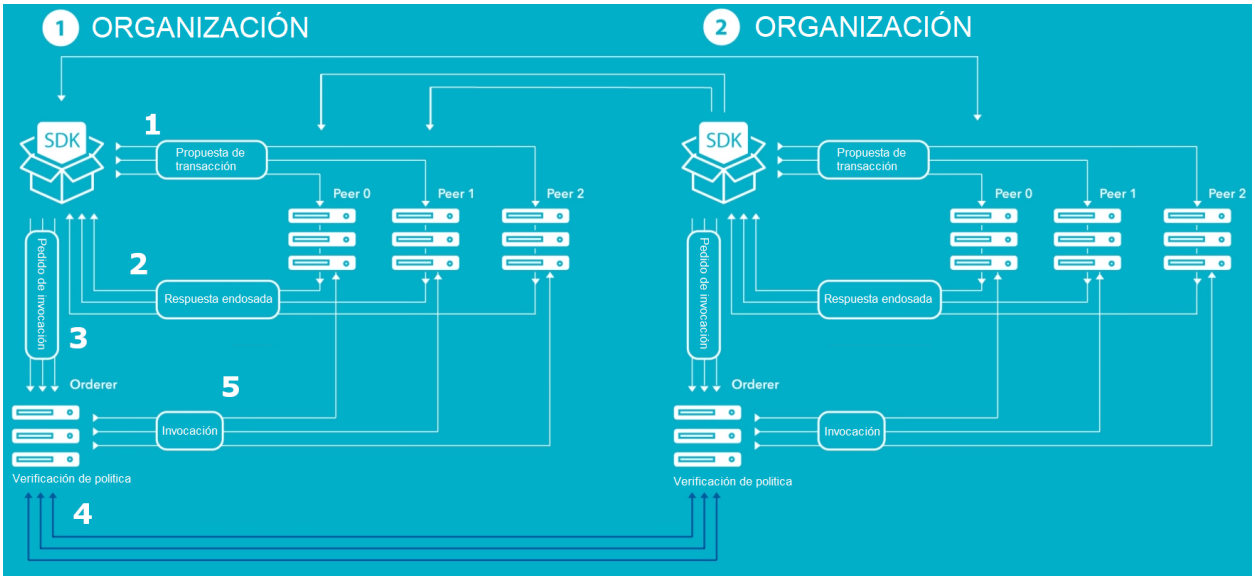

Figura 7.2. Flujo de ejecución de transacciones

Testing

Se realizó testing funcional de caja negra, debido a las condiciones de desarrollo de la prueba de concepto. Esto es debido a que no se contaba con una herramienta de depuración de código o algo más que cumpliera las mismas funciones y que los logs por defecto de Hyperledger no ofrecían suficiente información. Por tanto, para diagnosticar los problemas se debió realizar escritura de logs a salida estándar y observación de lo registrado en la base de datos de cada nodo. Respecto a las pruebas realizadas, no fueron desarrollados casos de prueba, dado que a medida que se implementaban las distintas funcionalidades se iban probando, debido a las dificultades ya mencionadas. Finalmente, para probar el funcionamiento de la aplicación implementada se realizaron pruebas funcionales basadas en los flujos de ejecución.

En su mayoría, los errores encontrados estaban relacionados con el funcionamiento de la herramienta Hyperledger, esto implica el funcionamiento de la red blockchain, sus configuraciones asociadas y errores en el código implementado. Para entender los problemas a nivel de red y su configuración, y en base a ello identificar los errores, se utilizó en gran parte la herramienta Blockchain Explorer, mencionada en la sección “7.1. Tecnologías utilizadas”.

7.3. Dificultades y limitaciones encontradas

En esta sección se presentan las distintas dificultades y limitaciones encontradas en la herramienta utilizada para la implementación del caso de uso, indicando cómo esto afectó la implementación, provocando retrasos en lo planificado.

Documentación de Hyperledger Fabric

En el estudio e implementación de las distintas configuraciones, el SDK y chaincodes, se encontraron diversas dificultades a lo largo del proyecto, dada la poca y defectuosa documentación que era presentada oficialmente. En el caso de configurar el Orderer con Kafka y definir múltiples Orderers para una red, no existe documentación oficial y fue realizado a partir de comentarios en distintos foros. Para la implementación del SDK, se analizó el código de ejemplo, realizando pruebas para comprender el funcionamiento del mismo, debido a que la documentación no lo explicaba. Con respecto a la implementación de chaincodes, ocurrió que los tutoriales oficiales no ejecutaban en las redes oficiales de ejemplo, dado que no compilaban.

Funcionamiento de Golang

Como fue mencionado anteriormente, el lenguaje de programación utilizado para la implementación de las chaincodes fue Golang, por lo cual antes de estimar el esfuerzo de desarrollo de las chaincodes, fue necesario estudiar el lenguaje. Aunque se realizó el estudio previamente comentado, la estimación realizada no fue lo suficientemente precisa, provocando en definitiva atrasos en la implementación. A modo de ejemplo, se presentó un problema a la hora de guardar y leer los datos en la blockchain, almacenandose de forma incorrecta. Esto se debe a que los nombres de las propiedades almacenadas en la blockchain tienen que comenzar con mayúsculas. Por otra parte, tanto en tiempo de compilación como en ejecución, nunca se pudo encontrar un mensaje de error o advertencia que hiciera alusión a esa problemática.

Funcionamiento de Docker

Investigando el funcionamiento de Docker, uno de los detalles encontrados es que utiliza un caché para los archivos utilizados por las máquinas virtuales, de forma de optimizar los tiempos de construcción de la máquina virtual. Esto originó problemas ya que en la etapa de pruebas e implementación de las chaincodes, al realizar modificaciones, estas no se veían reflejadas en los nodos. Hasta identificar la existencia de la caché, se realizaron varias modificaciones sobre las chaincodes, que se interpretaron como erróneas, siendo que en realidad, estas modificaciones no estaban siendo impactadas.

Pruebas sobre chaincodes

El desarrollo de las chaincodes no pudo ser probado antes de la instalación en cada uno de los nodos y la posterior puesta en funcionamiento de toda la red. No se encontró ninguna herramienta que permitiera validar previamente el comportamiento, por lo que lo desarrollado fue validado y probado mediante la instalación en la red a utilizar, reiniciando la misma cada vez que se realizaba un cambio, provocando retrasos en la implementación. Adicionalmente, los mensajes de errores en tiempo de compilación y ejecución, eran escuetos y poco descriptivos, por lo que su corrección no siempre era una tarea fácil.

Llamados entre chaincodes

En las pruebas realizadas, las invocaciones del tipo consulta entre chaincodes se realizaron satisfactoriamente, mientras que las invocaciones de alta o modificación en la chaincode destino no tuvieron efecto. Si bien la documentación indica que esto es posible, se manejan varias hipótesis de por qué no funcionó. Una de ellas es que la versión utilizada aún no soportara ese comportamiento, otra es que fuera un mecanismo defensivo para evitar loops infinitos entre chaincodes o que correspondiera a temas de permisos de escritura sobre la chaincode. No fue posible encontrar el motivo real de este comportamiento y por un tema de alcance y tiempos del proyecto se decidió no seguir indagando en el tema. El estudio de esta funcionalidad en la

herramienta, llevo un tiempo mayor al esperado, en gran parte por el alta o modificación en la chaincode destino. Esto produjo retrasos en la planificación, requiriendo una reestructuración de la solución planteada.

8. Gestión del proyecto

En este capítulo se comenta cómo fue organizado todo el proyecto, los ajustes que se debieron hacer y los motivos de éstos, así como las desviaciones que se dieron.

8.1. Organización

Dada la característica de investigación del proyecto, al comienzo se plantearon reuniones cada dos semanas con el tutor, de forma de poder compartir los avances de la investigación de la tecnología y definir en qué aspectos enfocarse. Las tareas fueron divididas entre el equipo, con excepción de algunos aspectos como el Consenso que fueron atacados en conjunto. En paralelo, la información obtenida de la investigación, era integrada en un documento. Si bien al principio la dinámica de trabajo era aceptable, existieron momentos donde no se pudo mantener, debido a diferentes factores que afectaron: estancamiento en el análisis debido a falta de información, licencias y tiempo dedicado a otras materias. Más adelante en el tiempo, cada uno de estos puntos fue solucionado y se pudo encauzar el ritmo del proyecto.

8.2. Planificación y ejecución

Al inicio del proyecto se creó una planificación a grandes rasgos sobre las tareas a realizar en el proyecto en un tiempo estimado de 12 meses, como se muestra en la *figura 8.1*.

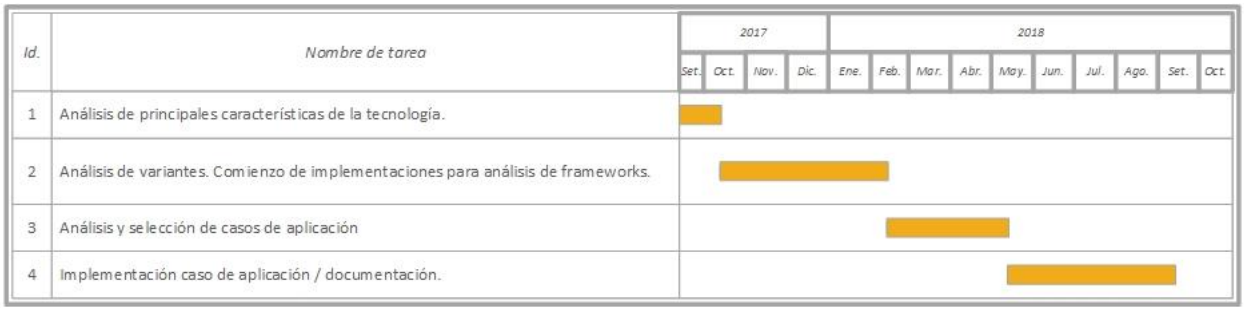

Figura 8.1. Gantt de estimación inicial

Luego de culminado el primer punto de la planificación inicial y transcurrido el primer mes del segundo, se realizó un desglose sobre las tareas pendientes del segundo punto, dado que a esta altura del proyecto ya se tenía definido qué tecnología iba a ser utilizada para el caso de uso, obteniendo como resultado lo siguiente:

1. Semana 1 y 2: Ecosistema de Hyperledger funcionando.

2. Semana 3 y 4: Hacer pruebas sobre los ejemplos que trae y hacer modificaciones en el código para evaluar el impacto (distintos tipos de consenso, distintos tipos de blockchain).
3. Semana 5 y 6: Desarrollo de una blockchain nueva en base a alguna elección.
4. Semana 7 y 8: Desarrollo de un smart contract que funcione sobre esta blockchain.

Durante el transcurso de las primeras dos semanas del segundo punto de la planificación inicial, se decidió tomar un receso por preparación de exámenes y finalización de otras materias. Una vez que se retomó el trabajo, se continuó la segunda etapa desde el punto 1, dejando el ecosistema de Hyperledger funcionando y luego se continuó con el resto de los puntos según el orden. Es necesario destacar que cada uno de estos puntos se vio afectado de desfasajes producto de problemas con el framework.

Al finalizar la segunda etapa de la planificación inicial, se comenzó la etapa de análisis y selección de casos de uso, para ello se evaluaron distintos casos de aplicación, los cuales en su mayoría no aplicaban. Posteriormente, una vez seleccionado el caso de uso se elaboró un Gantt (ver *figura 8.2*) con la planificación para el desarrollo del caso elegido y posterior documentación.

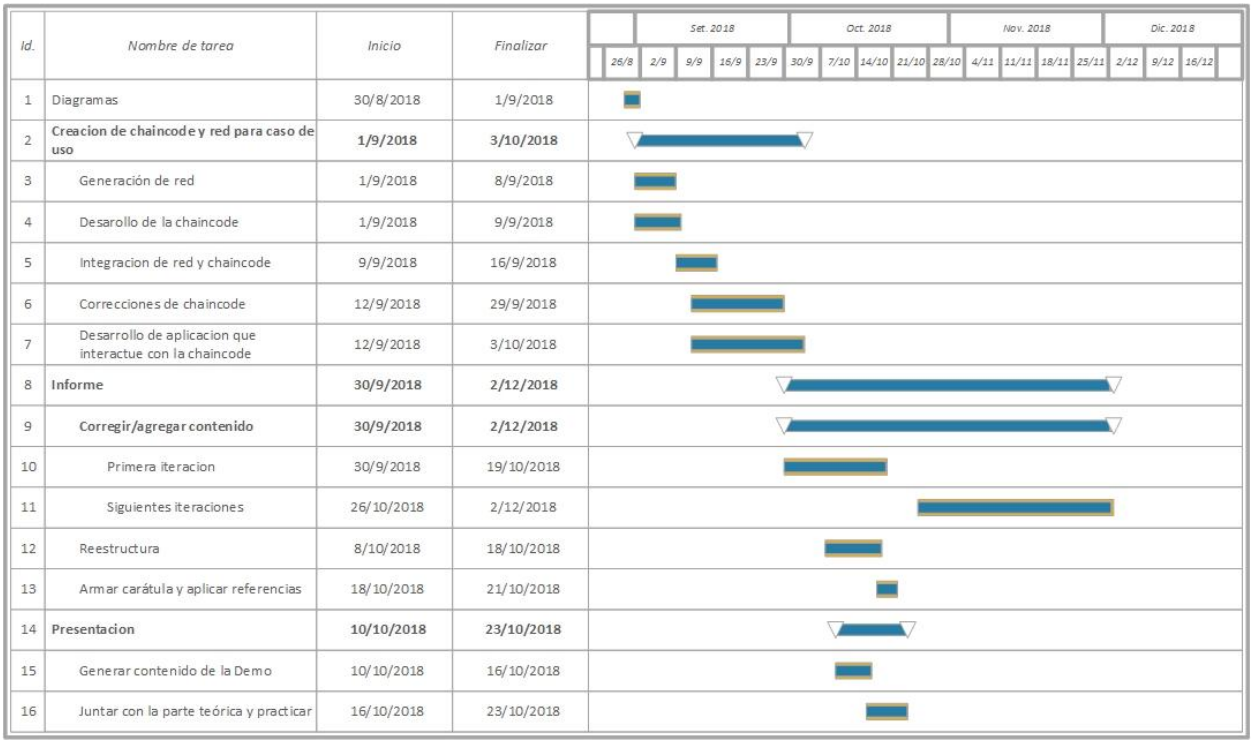

Figura 8.2. Gantt de implementación y documentación

Luego de finalizado el proyecto se pueden observar ciertas desviaciones a lo largo de las distintas etapas. En la etapa uno “Análisis de principales características de la tecnología” la desviación ocurrió debido a la gran dependencia con Bitcoin de la

información encontrada. Con respecto a la etapa dos “Análisis de variantes. Comienzo de implementaciones para análisis de frameworks”, la escasa y defectuosa documentación de Hyperledger Fabric en algunas de las configuraciones, como por ejemplo la configuración del Orderer donde el cambio a Kafka aparentaba cambiar el tipo de consenso, provocaron que se tuviera que dedicar una mayor cantidad de tiempo y esfuerzo para obtener resultados positivos. Por último, la desviación en la etapa de documentación está relacionada a la inexperiencia en la generación de este tipo de documentos. Las desviaciones mencionadas anteriormente se pueden observar en la *figura 8.3*, representando en naranja la estimación inicial y en verde la ejecución real.

Figura 8.3. Desviaciones en las etapas del proyecto.

Como se puede apreciar en la *figura 8.2*, se identificó un desvío de aproximadamente cinco meses en comparación a la planificación inicial, provocado por las distintas situaciones comentadas a lo largo del capítulo.

9. Conclusiones y trabajo a futuro

En este capítulo se presentan las conclusiones sobre el trabajo finalizado, los aspectos identificados a corregir y trabajo a futuro. Adicionalmente se presentan aspectos que se entienden pueden mejorar.

La tecnología blockchain aún se encuentra bajo estudio y evolución, lo cual es razonable para su corta vida, pero esto supone un riesgo dado el auge y velocidad de adopción que se visualiza actualmente, ya que puede llegar a ser aplicado a casos en los que no corresponda.

Un aspecto importante que se saca como conclusión y se considera se debe mejorar es lo relacionado con la aplicabilidad de blockchain. El uso de la tecnología blockchain es altamente recomendable en escenarios donde sea necesario contar con coordinación, confiabilidad y seguridad de datos entre múltiples actores sin intermediarios. Si se utiliza blockchain por una sola de estas características, es posible encontrarse con herramientas que ya resuelven este problema de forma más eficiente, como por ejemplo, bases de datos distribuidas. De esta forma, al comienzo del proyecto, se identificaron situaciones en las que por características específicas del problema se pensaba en la utilización de blockchain, cuando haciendo un análisis más profundo, resultaba no ser la mejor solución. Avanzado el proyecto, cuando se analizaron los casos de uso a aplicar, se encontraron una serie de cuestionarios que fueron de gran ayuda en este aspecto, pero que pueden presentar preguntas ambiguas, mencionadas en el capítulo “5.2. Validación de aplicabilidad” o ausencia de preguntas, tales como: “¿La información almacenada puede estar distribuida?”, “¿Los tiempos de respuesta deben de ser bajos?” y “¿La información debe de ser borrada completamente?”, dejando de todas formas dudas de su aplicabilidad. A pesar de ello se logró entender el funcionamiento de la tecnología y los escenarios donde aporta valor su aplicación.

Al estudiar los casos desarrollados en la actualidad sobre la tecnología blockchain, se identificó que, en gran medida, estos refieren a la utilización de la tecnología para asegurar la integridad de los datos y tener un registro de las modificaciones que estos pudieran sufrir. En el estudio de la tecnología, se encontraron casos donde no es recomendable su utilización debido a su forma de construcción. Estos casos involucran datos que deben de poder ser eliminados en algún momento, como por ejemplo el registro de datos personales, que deben de ser eliminados luego de cinco años desde su incorporación, como está estipulado en el artículo 9 de la ley 17.838 [79].

Luego de evaluar las distintas posibilidades de aplicación de blockchain en la administración pública, se logró implementar una prueba de concepto de un caso de uso para dar solución a la problemática que tienen los escribanos a la hora de recabar información sobre el estado del empadronamiento y deudas de un auto, como parte del

proceso de compraventa. A su vez, esta implementación ayuda a las intendencias a controlar que un auto solo esté empadronado en una única intendencia, dado que el registro de empadronamientos pasa a estar distribuido y sincronizado en cada una, a diferencia a como se encuentra actualmente. Por otra parte, al utilizar la tecnología blockchain, la información y sus cambios quedan registrados de forma incorruptible, permitiendo así, un mayor control y seguridad sobre los datos de empadronamiento almacenados.

Como fue mencionado a lo largo del documento, fueron evaluadas varias herramientas para el desarrollo de blockchain. Si bien para la herramienta utilizada en la implementación del caso de aplicación se identificaron varios puntos positivos, esta presenta una gran curva de aprendizaje y una pobre documentación, además de que algunas funcionalidades importantes, como lo es la tolerancia a fallas, aún no ha sido resuelta. Esto lleva a que al momento de trabajar con la herramienta se generen situaciones en las que se tenga incertidumbre de si es un error de la herramienta o es una falla en la configuración o implementación del usuario, generando desvíos sobre lo planificado para el proyecto. Por lo tanto, se recomienda esperar a que la herramienta madure para su utilización en la implementación de una solución en producción.

Como trabajo a futuro, se propone mejorar la implementación del caso de uso al momento de dar de alta o modificar un empadronamiento, controlando que el usuario que originó la transacción, pertenezca a la intendencia del empadronamiento. Adicionalmente, se podría integrar con el manejo de multas para los vehículos integrando nuevos actores a la red como lo son policía caminera y policía de tránsito, brindando al usuario una vista más detallada sobre el estado de endeudamiento del vehículo, indicando las multas adjudicadas al mismo, junto con otros datos relevantes, como lo son: fecha, hora, importe, concepto de multa, etc. Esta forma de brindar más información a los usuarios, permite que sea utilizada por los escribanos a la hora de realizar una compraventa de un vehículo. Como mejora arquitectónica, se propone agregar más intendencias a la red, configurando Kafka y un Orderer por cada intendencia.

Con respecto a la investigación, se propone continuar actualizando el caso de uso a versiones más recientes de Hyperledger Fabric. De forma de utilizar las nuevas funcionalidades, optimizaciones y en un futuro, cuando la configuración del Orderer PBFT sea liberada, utilizarla.

Como conclusión general del proyecto se puede destacar que el estudio realizado sobre la tecnología blockchain, brinda una base para la comprensión de la misma, ya que no es un tema fácil de abordar, dado que se encuentra fuertemente vinculado a Bitcoin y esto provoca que la primera aproximación a los conceptos sea a través de lo definido por la criptomoneda.

Referencias

[1] S. Nakamoto, «Bitcoin: A Peer-to-Peer Electronic Cash System,» 2008. [En línea]. Available: https://bitcoin.org/.

[2] J. KILLMEYER, M. WHITE y B. CHEW, «Will blockchain transform the public sector?,» Deloitte University Press, 2017. [En línea]. Available: https://www2.deloitte.com/content/dam/insights/us/articles/4185_blockchain-public-sector/DUP_will-blockchain-transform-public-sector.pdf.

[3] Agesic, «Plan de Gobierno Digital Uruguay 2020,» 2015. [En línea]. Available: https://www.agesic.gub.uy/innovaportal/file/6539/1/plan_de_gobierno_digital.pdf.

[4] « Bitcoin wiki, Double-Spending,» [En línea]. Available: https://es.bitcoinwiki.org/wiki/Double-spending.

[5] «Problema de gasto doble en Bitcoin ¿Qué es double spending o gasto doble en tecnología descentralizada?,» [En línea]. Available: https://www.oroyfinanzas.com/2014/11/que-es-problema-double-spending-doble-gasto-p2p-bitcoin/.

[6] S. Haber y W. S. Stornetta, «How to time-stamp a digital document. Journal of Cryptology,» *Journal of Cryptology,* vol. 3, nº 2, pp. 99-111, 1991.

[7] «¿Qué es la Cadena de Bloques (Blockchain)?,» [En línea]. Available: https://academy.bit2me.com/que-es-cadena-de-bloques-Blockchain/.

[8] M. Allende López, «Blockchain - Como desarrollar confianza en entornos complejos para generar valor de impacto social,» 2018. [En línea]. Available: https://publications.iadb.org/bitstream/handle/11319/8919/Blockchain-Como-desarrollar-confianza-en-entornos-complejos-para-generar-valor-de-impacto-social.pdf.

[9] «Cómo funciona Blockchain y sus partes,» [En línea]. Available: https://miethereum.com/blockchain/como-funciona/.

[10] «The Wall Street Journal - CIO Explainer: What Is Blockchain?,» [En línea]. Available: https://blogs.wsj.com/cio/2016/02/02/cio-explainer-what-is-blockchain/.

[11] «Blocks (of a blockchain) – definition and meaning,» [En línea]. Available: https://marketbusinessnews.com/blocks-blockchain/.

[12] «Types of Blockchains & DLTs (Distributed Ledger Technologies),» [En línea]. Available: https://Blockchainhub.net/Blockchains-and-distributed-ledger-technologies-in-general/.

[13] «Types of Blockchains,» [En línea]. Available: https://devopedia.org/types-of-Blockchains.

[14] «AES-256,» [En línea]. Available: https://www.boxcryptor.com/es/encryption/.

[15] «Proof of Stake (PoS) Definition,» [En línea]. Available: https://www.investopedia.com/terms/p/proof-stake-pos.asp.

[16] «What is Proof of Authority Consensus? (PoA) Staking Your Identity,» [En línea]. Available:

https://blockonomi.com/proof-of-authority/.

[17] «Litecoin,» [En línea]. Available: https://litecoin.org/es/.

[18] «Namecoin,» [En línea]. Available: https://namecoin.org/.

[19] «Dogecoin,» [En línea]. Available: https://dogecoin.com/.

[20] «Ethereum,» [En línea]. Available: https://www.ethereum.org/.

[21] «SHA-2,» [En línea]. Available: https://es.wikipedia.org/wiki/SHA-2.

[22] «Bitcoin Energy Consumption Index - Digiconomist,» [En línea]. Available: https://digiconomist.net/bitcoin-energy-consumption.

[23] «Smart Contracts: ¿Qué son, cómo funcionan y qué aportan?,» [En línea]. Available: https://academy.bit2me.com/que-son-los-smart-contracts/.

[24] «Definición contrato,» [En línea]. Available: https://dle.rae.es/?id=AdXPxYJ.

[25] S. Talens-Oliag, «Introducción a los certificados digitales,» 2003. [En línea]. Available: https://www.uv.es/~sto/articulos/BEI-2003-11/certificados_digitales.pdf.

[26] «What's a Blockchain oracle? Information oracles & external data feeds,» [En línea]. Available: https://Blockchainhub.net/Blockchain-oracles/.

[27] «What is a blockchain oracle?,» [En línea]. Available: https://blog.apla.io/what-is-a-blockchain-oracle-2ccca433c026.

[28] «Proof of Existence,» [En línea]. Available: http://www.proofofexistence.com/.

[29] «Tasas y Precios - Dirección Nacional de la Propiedad Industrial,» [En línea]. Available: http://www.uruguay.gub.uy/GuiaTramitesEstado/Archivos/1258596Tasas%20y%20Precios%20-%20Direccin%20Nacional%20de%20la%20Propiedad%20Industrial.pdf.

[30] «TradeLens,» [En línea]. Available: https://www.tradelens.com.

[31] «MoyeeCoffee,» [En línea]. Available: https://moyeecoffee.ie/blogs/moyee/world-s-first-blockchain-coffee-project.

[32] B. Rakic, T. Levak, Z. Drev, S. Savic y A. Veljkovic, «First purpose built protocol for supply chains based on blockchain,» 5 Octubre 2017. [En línea]. Available: https://origintrail.io/storage/documents/OriginTrail-White-Paper.pdf.

[33] «Maersk,» [En línea]. Available: https://www.maersk.com.

[34] R. Sainz, C. Pereira y E. Portugues, «Godzillion: A Decentralized Startup Crowdfunding System,» Agosto 2017. [En línea]. Available: https://godzillion.io/assets/documentos/godz_white_paper.pdf.

[35] «WePower,» [En línea]. Available: https://github.com/WePowerNetwork/wepower-testbed/blob/master/EleringPilot.ipynb.

[36] «Storj: A Decentralized Cloud Storage Network Framework,» [En línea]. Available: https://storj.io/storjv3.pdf.

[37] «Blockchain and its use in the public sector,» [En línea]. Available: https://oecd-opsi.org/wp-content/uploads/2018/06/Blockchains-Unchained-Slides.pdf.

[38] «World Citizenship,» [En línea]. Available: https://github.com/MrChrisJ/World-Citizenship.

[39] «Agora,» [En línea]. Available: https://www.agora.vote.

[40] R. Pipan, «Blockchain Land-Titling Project,» 7 Febrero 2016. [En línea]. Available: https://bitfury.com/. [Último acceso: 1 Noviembre 2018].

[41] «Keyless Signature Infrastructure,» [En línea]. Available: https://www.guardtime-federal.com/ksi/. [Último acceso: 11 Febrero 2019].

[42] «Registros médicos en Estonia y Dubai,» [En línea]. Available: https://news.coinify.com/estonia-dubai-medical-records-Blockchain/. [Último acceso: 18 Noviembre 2018].

[43] «Smart Dubai - Blockchain,» [En línea]. Available: https://smartdubai.ae/initiatives/blockchain. [Último acceso: 27 Octubre 2018].

[44] «HCEN,» [En línea]. Available: https://hcen.salud.uy. [Último acceso: 13 Diciembre 2018].

[45] «Programa Building Blocks,» [En línea]. Available: https://innovation.wfp.org/project/building-blocks. [Último acceso: 1 Noviembre 2018].

[46] «Building Blocks - The future of cash disbursements at the World Food Programme,» [En línea]. Available: https://innovation.wfp.org/project/building-blocks. [Último acceso: 1 Noviembre 2018].

[47] «A blockchain platforms comparison,» [En línea]. Available: https://vironit.com/a-blockchain-platforms-comparison/. [Último acceso: 27 Octubre 2018].

[48] «HyperLedger Fabric 1.1,» [En línea]. Available: https://media.readthedocs.org/pdf/hyperledger-fabric/release-1.1/hyperledger-fabric.pdf. [Último acceso: 1 Noviembre 2018].

[49] «Ethereum Introduction · ethereum/wiki Wiki · GitHub,» [En línea]. Available: https://github.com/ethereum/wiki/wiki/Ethereum-introduction. [Último acceso: 18 Noviembre 2018].

[50] S. Kepser, A Simple Proof for the Turing-Completeness of XSLT and XQuery, Montréal, Québec: SIGCOMM, 2004.

[51] «Corda,» [En línea]. Available: https://www.corda.net. [Último acceso: 27 Octubre 2018].

[52] «R3,» [En línea]. Available: https://www.r3.com. [Último acceso: 30 Noviembre 2018].

[53] «Credits blockchain platform | Credits.com,» [En línea]. Available: https://credits.com/. [Último acceso: 27 Octubre 2018].

[54] «A Blockchain Platform for the Enterprise,» [En línea]. Available: http://hyperledger-fabric.readthedocs.io/. [Último acceso: 18 Noviembre 2018].

[55] «Ledger,» [En línea]. Available: https://hyperledger-fabric.readthedocs.io/en/master/ledger/ledger.html. [Último acceso: 1 Noviembre 2018].

[56] «Blockchain network,» [En línea]. Available: https://hyperledger-fabric.readthedocs.io/en/master/network/network.html. [Último acceso: 1 Noviembre 2018].

[57] «Channels,» [En línea]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.1/channels.html. [Último acceso: 18 Noviembre 2018].

[58] «Hyperledger Fabric Components,» [En línea]. Available: https://medium.com/swlh/hyperledger-chapter-6-hyperledger-fabric-components-technical-context-767985f605dd. [Último acceso: 7 Febrero 2019].

[59] «Apache Kafka,» [En línea]. Available: https://kafka.apache.org/. [Último acceso: 29 Octubre 2018].

[60] «Chaincode for Developers,» [En línea]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.3/chaincode4ade.html. [Último acceso: 26 Diciembre 2018].

[61] «Endorsement policies,» [En línea]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.1/endorsement-policies.html. [Último acceso: 18 Noviembre 2018].

[62] «Architecture Explained,» [En línea]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.1/arch-deep-dive.html. [Último acceso: 18 Noviembre 2018].

[63] «Canal oficial de Hyperledger,» [En línea]. Available: https://www.youtube.com/watch?v=2_RgCfjunEU&t=2s. [Último acceso: 18 Noviembre 2018].

[64] «Transaction Flow,» [En línea]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.1/txflow.html. [Último acceso: 18 Noviembre 2018].

[65] A. Gervais y K. Wüst, «Do you Need a Blockchain?,» de *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, Zug, 2018.

[66] T. Koens y E. Poll, «What Blockchain Alternative Do You Need?,» Nijmegen, 2018.

[67] M. ElSeidy, «To Blockchain or To Not Blockchain,» 28 Agosto 2018. [En línea]. Available: https://medium.com/zkcapital/to-blockchain-or-to-not-blockchain-40e6a3a60f46. [Último acceso: 16 Febrero 2019].

[68] C. Mulligan, J. Zhu Scott, S. Warren y J. Rangaswami, «Blockchain Beyond the Hype A Practical Framework for Business Leaders,» Abril 2018. [En línea]. Available: http://www3.weforum.org/docs/48423_Whether_Blockchain_WP.pdf. [Último acceso: 16 Febrero 2019].

[69] E. Ganne, «Can Blockchain revolutionize international trade?,» 2018. [En línea]. Available: https://www.wto.org/english/res_e/booksp_e/blockchainrev18_e.pdf. [Último acceso: 16 Febrero 2019].

[70] «Distributed Ledger Technology, Blockchains and Identity. A Regulatory Overview,» Septiembre 2018. [En línea]. Available: https://www.gsma.com/identity/wp-content/uploads/2018/09/Distributed-Ledger-Technology-Blockchains-and-Identity-20180907ii.pdf. [Último acceso: 16 Febrero 2019].

[71] P. Kruchten, «Architectural Blueprints—The “4+1” View Model of Software Architecture,» *IEEE Software 12 (6),* pp. 42-50, Noviembre 1995.

[72] «Express - Infraestructura de aplicaciones web Node.js,» [En línea]. Available: https://expressjs.com/es/. [Último acceso: 3 Noviembre 2018].

[73] «Node.js,» [En línea]. Available: https://nodejs.org/es/. [Último acceso: 3 Noviembre 2018].

[74] «Angular,» [En línea]. Available: https://angular.io/. [Último acceso: 3 Noviembre 2018].

[75] «Enterprise Container Platform | Docker,» [En línea]. Available: https://www.docker.com/. [Último acceso: 3 Noviembre 2018].

[76] «GitHub - hyperledger/blockchain-explorer,» [En línea]. Available: https://github.com/hyperledger/blockchain-explorer. [Último acceso: 3 Noviembre 2018].

[77] «GitLab,» [En línea]. Available: https://gitlab.fing.edu.uy/. [Último acceso: 18 Noviembre 2018].

[78] «Visual Studio Code - Code Editing. Redefined,» [En línea]. Available: https://code.visualstudio.com/. [Último acceso: 18 Noviembre 2018].

[79] «Ley 17.838. PROTECCIÓN DE DATOS PERSONALES PARA SER UTILIZADOS EN INFORMES COMERCIALES Y ACCIÓN DE HABEAS DATA,» *Diario Oficial,* Octubre 2004.

[80] «Hyperledger Fabric 1.3,» [En línea]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.3/whatsnew.html. [Último acceso: 10 Noviembre 2018].

[81] C. Pérez-Solà y J. Herrera-Joancomartí, «Bitcoins y el problema de los generales bizantinos,» Alicante, 2014.

[82] L. LAMPORT, R. SHOSTAK y M. PEASE, «The Byzantine Generals Problem,» *ACMTransactions on Programming Languages and System,* vol. 4, pp. 382-401, Julio 1982.

GLOSARIO

Criptomoneda: medio digital de intercambio que utiliza criptografía para asegurar las transacciones financieras, controlar la creación de unidades adicionales y verificar la transferencia de activos.

API: interfaces de servicios ofrecidos a través de la Web, de manera pública y que sirven de puerta de entrada para el uso de los diferentes servicios que una organización posee.

REST API: interfaces entre sistemas que usen el protocolo HTTP para obtener datos o generar operaciones sobre los mismos en todos los formatos posibles, como XML y JSON.

Aplicación web: en la ingeniería de software se refiere a cualquier aplicación que los usuarios pueden utilizar a través de una red como internet o una intranet.

Servidor web: es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales o unidireccionales y síncronas o asíncronas con el cliente y generando una respuesta en cualquier lenguaje o aplicación del lado del cliente.

CPU: es el hardware dentro de una computadora, que interpreta las instrucciones de un programa informático mediante la realización de las operaciones básicas aritméticas, lógicas y de entrada/salida del sistema.

JSON: es un formato de intercambio de datos rápido.

JavaScript: lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Node.js: entorno en tiempo de ejecución multiplataforma de código abierto, utilizado comúnmente para la capa del servidor. Es asíncrono, con entrada y salida de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google.

Typescript: lenguaje de programación de código abierto, desarrollado y mantenido por Microsoft. Es un superconjunto de JavaScript, que esencialmente añade tipado estático y objetos basados en clases. Este puede ser utilizado para desarrollar aplicaciones JavaScript que se ejecutarán en el lado del cliente o del servidor.

Golang: lenguaje de programación de código abierto, el cual es compilado en código de máquina antes de ser ejecutado. Este lenguaje fue utilizado dado que los smart contracts en Hyperledger Fabric pueden ser escritos en este lenguaje, además la documentación y ejemplos son utilizando este lenguaje.

Funciones hash: una función hash H es una función computable mediante un algoritmo tal que tiene como entrada un conjunto de elementos, que suelen ser cadenas, y los convierte en un rango de salida finito, normalmente cadenas de longitud fija.

Función criptográfica: son aquellas que cifran una entrada y actúan de forma parecida a las funciones hash, ya que comprimen la entrada a una salida de menor longitud y son fáciles de calcular.

Función biyectiva: una función es biyectiva si todos los elementos del conjunto de salida tienen una imagen distinta en el conjunto de llegada, y a cada elemento del conjunto de llegada le corresponde un elemento del conjunto de salida.

peer-to-peer: es una red de computadoras en la que todos o algunos aspectos funcionan sin clientes ni servidores fijos, sino una serie de nodos que se comportan como iguales entre sí.

B2B: hace referencia a las transacciones comerciales entre empresas, es decir, a aquellas que típicamente se establecen entre un fabricante y el distribuidor de un producto, o entre un distribuidor y un comercio minorista. Las relaciones entre un comerciante y su cliente final se denominan B2C (del inglés, Business to Consumer).

Dirham: moneda de curso legal de los Emiratos Árabes Unidos.

Framework: estructura conceptual y tecnológica de asistencia definida, normalmente, con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

MVC: Modelo-vista-controlador (MVC) es un patrón de arquitectura de software, que separa los datos y la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones.

publish-subscribe: es un patrón de diseño de arquitectura de software, enfocado en mensajería, en donde los generadores de mensajes llamados publicadores, envían sus mensajes en canales agrupados por tópicos, en donde los interesados en recibir estos mensajes, se suscriben a los canales.

Trusted third party: Una entidad que facilita la interacción entre distintas partes, dado que esas partes confían en esa entidad.

Keyless Signature Infrastructure (KSI): Es un método y una red distribuida globalmente para la emisión y verificación de firmas KSI.

Software Development Kit (SDK): es un conjunto de herramientas de desarrollo de software que le permite al programador o desarrollador de software crear una aplicación informática para un sistema concreto.

Startup: es una empresa en su etapa temprana; a diferencia de una Pyme, la Startup se basa en un negocio que será escalable más rápida y fácilmente, haciendo uso de tecnologías digitales

APÉNDICE A - Instalación de Hyperledger Fabric

En este apéndice se explica cómo realizar la instalación de la red con Hyperledger Fabric para el sistema operativo Ubuntu.

Primero hay que instalar los siguientes programas:

- cURL
- Docker version 17.06.2-ce o mayor
- Docker Compose version 17.06.2-ce o mayor
- Node.js version 8.9.x o mayor
- Python 2.7 (solo para ubuntu)
- Golang version 1.9.x

Luego de lo anterior, se debe de ingresar la variable de entorno GOPATH al directorio de instalación de Golang.

Luego de culminados los pasos anteriores, se debe de continuar con la instalación de Hyperledger Fabric y sus redes de ejemplo, corriendo los siguientes comandos en consola:

1. git clone -b master https://github.com/hyperledger/fabric-samples.git
2. cd fabric-samples
3. git checkout release-1.1
4. curl -sSL https://goo.gl/6wtTN5 | bash -s <fabric> <fabric-ca> <thirdparty>
5. curl -sSL https://goo.gl/6wtTN5 | bash -s 1.1.0 1.1.0 0.4.6
6. export PATH=<ruta a carpeta donde se descargaron los archivos>/bin:$PATH

Luego de esto Hyperledger Fabric quedará completamente instalado.

APÉNDICE B - Instalación de la aplicación

En este apéndice se explica cómo realizar la instalación de la aplicación que implementa el caso de uso.

Como primer paso debe de poseer la carpeta “ProyectoGrado” con las carpetas que muestra la *figura B.1*

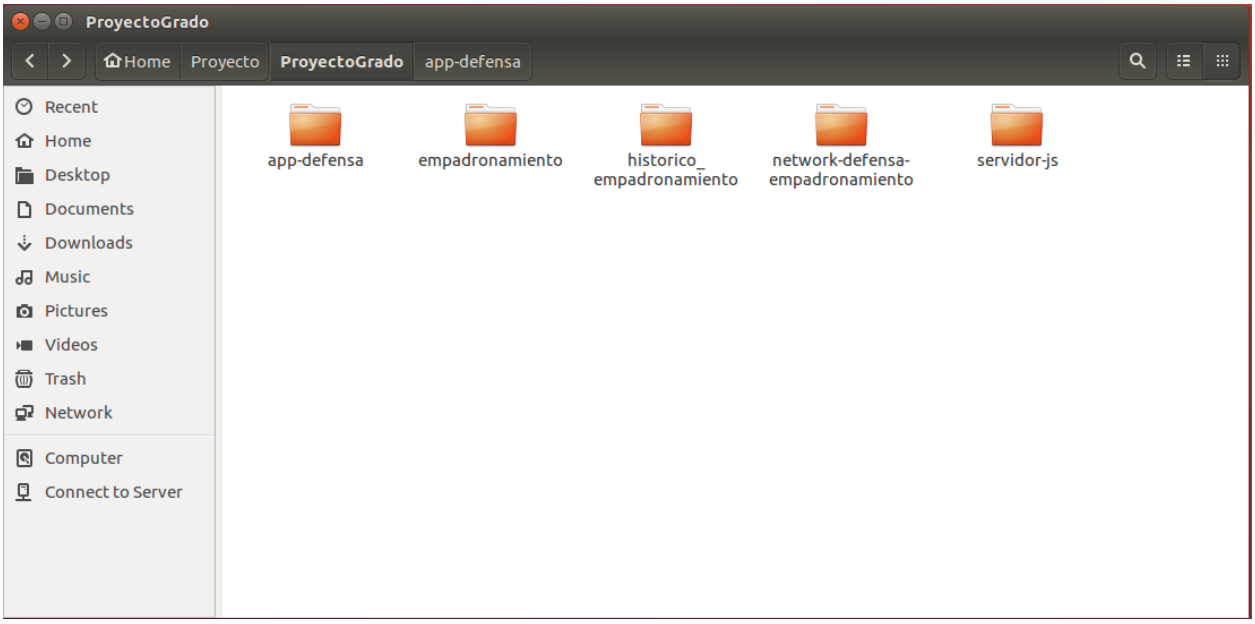

Figura B.1. Contenido de carpeta de caso de uso

A continuación, se pasará a describir qué es cada carpeta:

- app-defensa: aplicación web que presenta interfaz de usuario para los trabajadores de las intendencias y los escribanos.
- network-defensa-empadronamiento: red de caso de uso.
- servidor-js: servidor express.js.
- empadronamiento: contiene la chaincode de “empadronamiento”.
- historico_empadronamiento: contiene la chaincode de “historico_empadronamiento”.

Como primer paso se debe de copiar las carpetas “empadronamiento” y “historico_empadronamiento” a la carpeta “chaincode” que se encuentra dentro de “fabric-samples”, instalada en el apéndice “APÉNDICE A - Instalación de Hyperledger Fabric”.

Luego se debe de copiar la carpeta “network-defensa-empadronamiento” a la misma carpeta “fabric-samples” ya mencionada. Para levantar esta red, hay que ejecutar con el comando “sudo” el archivo “levantar.sh” y esperar a que este complete su ejecución.

Al finalizar se debería visualizar el mensaje mostrando en consola al igual que en la *figura B.2*.

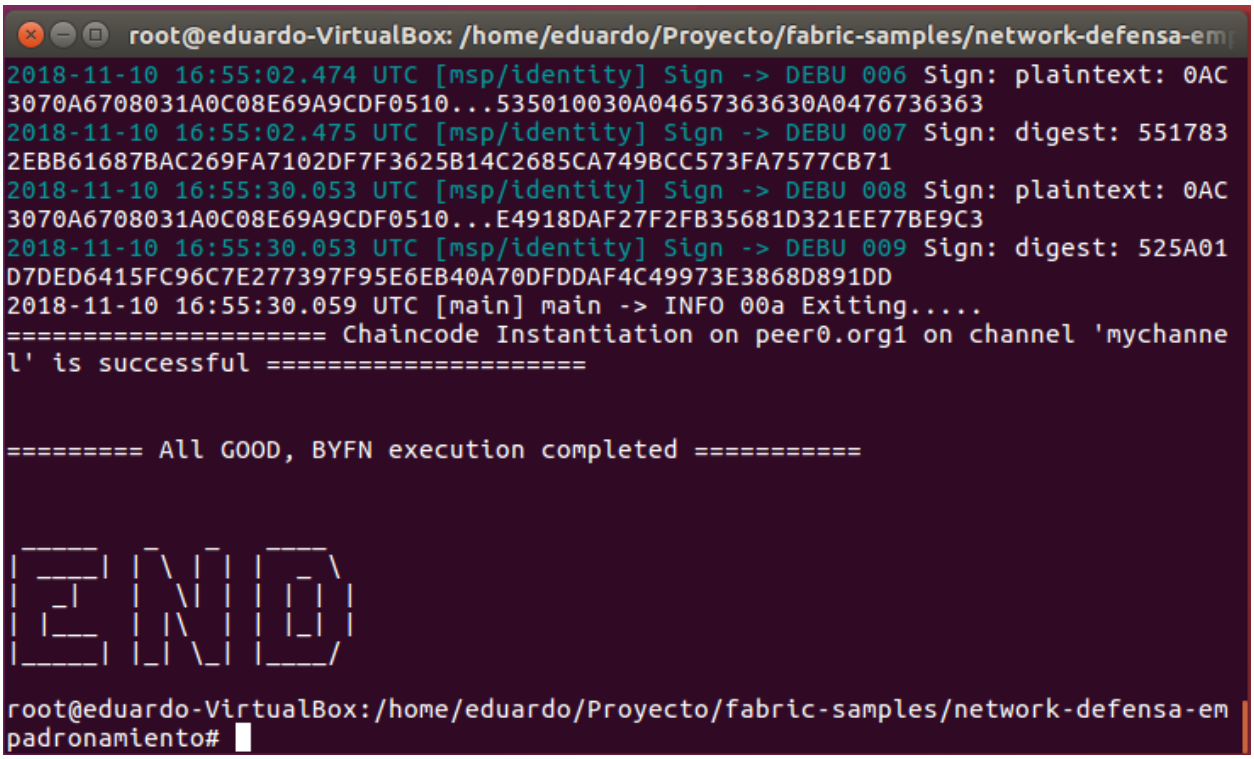

Figura B.2. Consola mostrando red levantada correctamente.

Luego de esto hay que ejecutar el comando “npm install” en las carpetas “servidor-js” y “app-defensa”, esperando a que se complete su ejecución. Luego ejecutar el comando “node app.js” dentro de la carpeta “servidor-js”, esperando a que se muestre el mensaje “app running on port. 3000”, como se muestra en la *figura B.3*.

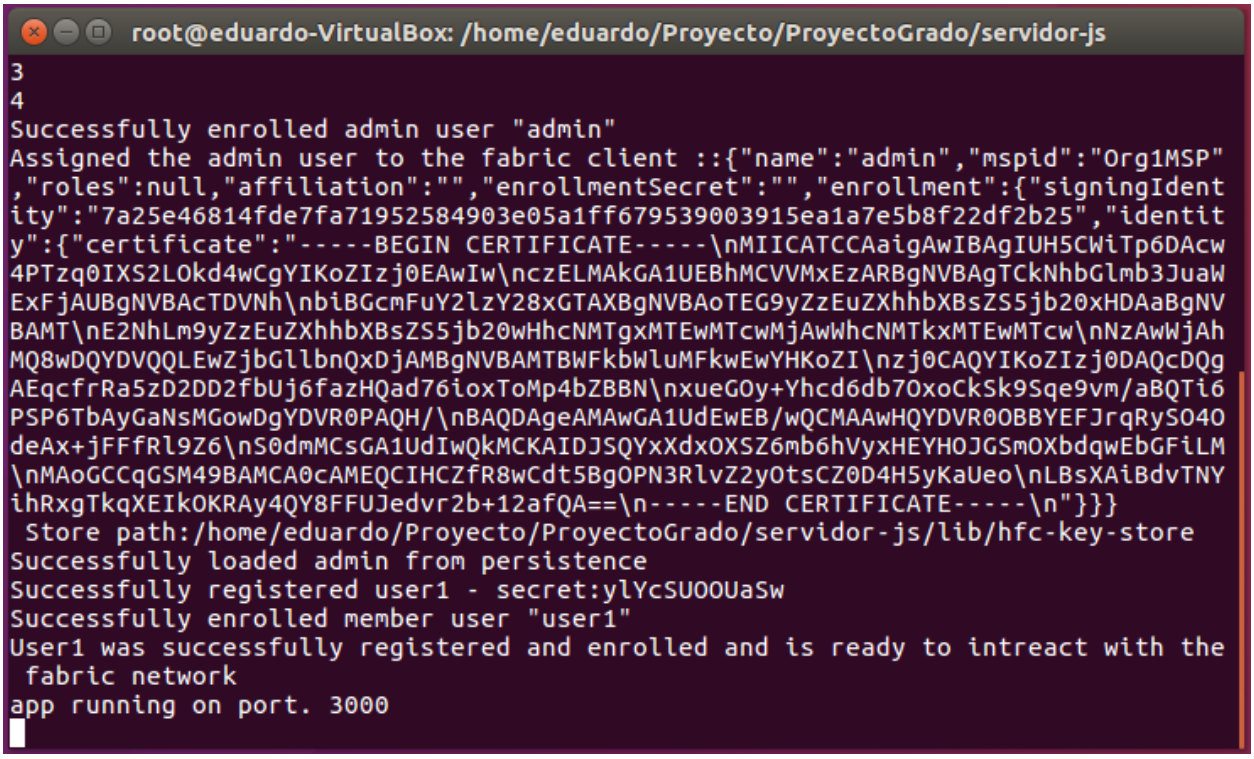

Figura B.3. Ejecución de "node app.js" dentro de servidor-js

Luego de esto ejecutar el comando "npm start" dentro de la carpeta "app-defensa", esperando el mensaje "Compiled successfuly", como se muestra en la *figura B.4*.

Figura B.4. Ejecución de "npm start" dentro de app-defensa

Luego de completado estos pasos, se puede acceder a la aplicación ingresando en el navegador web, a la dirección “localhost:4200”.

APÉNDICE C - Cambio versión Hyperledger

La última versión que se encuentra activa de Hyperledger Fabric es la 1.3 [80]. Esta presenta las siguientes funcionalidades nuevas:

- El componente MSP tiene una nueva funcionalidad conocida como Identity Mixer. Esta es una nueva forma de mantener las identidades de forma anónima e indistinguibles a través del uso de la prueba zero-knowledge. Para lograr esto se generan credenciales Identity Mixer en un entorno de prueba con una herramienta conocida como idexmigen.
- Se permite sobreescribir políticas de aprobación por defecto a nivel de chaincode, por políticas de aprobación por clave. En la práctica, esto significa que la política de aprobación a nivel de clave puede ser menos restrictiva o más restrictiva que la política de aprobación a nivel de chaincode. La política de aprobación a nivel de chaincode debe ser aprobada para que se pueda establecer una política de aprobación a nivel de clave por primera vez.
- Las consultas realizadas a la chaincode ahora están paginadas, por lo que los clientes ahora pueden experimentar una mejor performance ante grandes consultas.
- Hasta el momento Fabric únicamente soportaba para la escritura de chaincodes los lenguajes Go y Node.js, pero en esta nueva versión ahora también soporta Java.
- El servicio de eventos basado en canales de peers no es un concepto nuevo en Fabric, este aparece en la versión 1.1, pero la versión 1.3 marca el fin de este tipo de comunicación y si se quiere actualizar a la versión 1.3 se deberá actualizar a este mecanismo.

Por otra parte, también presenta nuevos tutoriales y nueva documentación:

- Se utiliza la red BYFN (Build Your First Network) para mostrar cómo debería funcionar el flujo de actualización. Este incluye dos scripts, que pueden servir como templates para la confección de sus propias actualizaciones, así como también los comandos individuales.
- Se actualiza el tutorial existente para CouchDB, agregando la nueva funcionalidad de paginación.
- Se agrega documentación conceptual que muestra las distintas partes que interactúan en la red blockchain. La versión inicial de esta documentación fue añadida en la versión 1.2.

ANEXO

Problema de los generales Bizantinos

El problema de los generales bizantinos [81] [82] es un experimento creado para ilustrar el dilema de lograr un consenso entre un conjunto de entidades con un objetivo común cuando entre ellas pueden existir traidores, es decir, entidades con objetivos opuestos a los del resto. Además, supone que las comunicaciones entre dichas entidades son limitadas e inseguras. El problema se presenta como una analogía con un escenario de guerra, donde un grupo de generales bizantinos se encuentran acampados con sus tropas alrededor de una ciudad enemiga que desean atacar. Después de observar el comportamiento del enemigo, los generales deben comunicar sus observaciones y ponerse de acuerdo en un plan de batalla común que permita atacar la ciudad y vencer. Para ello, los generales se comunican únicamente a través de mensajeros que pueden ser interceptados para cambiar el mensaje enviado o incluso no permitir que llegue a destino. Además, existe la posibilidad que algunos de los generales sean traidores y, por lo tanto, decidan enviar mensajes con información errónea con el objetivo de confundir a los generales leales. Un algoritmo que solucione el problema debe asegurar que todos los generales leales acuerdan un mismo plan de acción y que unos pocos traidores no pueden conseguir que el plan adoptado por los generales leales sea equivocado.

Dejando de lado el enfoque bélico y reformulando para los sistemas distribuidos, un componente como un servidor, se puede encontrar en un estado inconsistente y aparentar estar fallando o funcionando correctamente para sistemas de detección de fallas, presentando diferentes síntomas a diferentes observadores. Ante esta situación es difícil declarar que ese componente está fallando y expulsarlo de la red, dado que el primer objetivo que se tiene en estos sistemas es continuar funcionando correctamente a pesar de las fallas de estos componentes.