UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

DISEÑO DE LA ARQUITECTURA DE UN SISTEMA DE IDENTIDAD DIGITAL BASADA EN TECNOLOGÍA BLOCKCHAIN APLICADA AL REGISTRO ÚNICO DE CLIENTES DEL SECTOR FINANCIERO

Daniel Felipe Soto Peña

Universidad Distrital Francisco José De Caldas
Facultad de Ingeniería
Bogotá D.C, 2019

DISEÑO DE LA ARQUITECTURA DE UN SISTEMA DE IDENTIDAD DIGITAL BASADA EN TECNOLOGÍA BLOCKCHAIN APLICADA AL REGISTRO ÚNICO DE CLIENTES DEL SECTOR FINANCIERO

Autor:
DANIEL FELIPE SOTO PEÑA

Proyecto de grado para optar al título de
Especialista En Ingeniería De Software

Director:
Phd. Joaquin Javier Meza Alvarez

Universidad Distrital Francisco José De Caldas
Especialización En Ingeniería De Software
Bogotá D.C, 2019

Contenido

Índice de figuras

Índice de tablas

INTRODUCCIÓN

La digitalización de los servicios en el sector financiero ha traído consigo grandes retos en varios campos, entre ellos, la identificación digital a través de diferentes plataformas, lo cual implica para el usuario la creación de múltiples identidades en la web.

En Colombia, la ley [1] le otorga a las Fuentes Recolectoras de Información la responsabilidad de garantizar la calidad de la información obtenida, así como a conservar con las debidas seguridades los registros almacenados para impedir su deterioro, pérdida, alteración, uso no autorizado o fraudulento. A pesar de esto se han registrado múltiples casos de fuga de información, algunos de estos incluso presuntamente con consentimiento de las Fuentes Recolectoras.

La tecnología del BlockChain ha experimentado grandes avances en los ùltimos años, y su uso actualmente va más allá de las criptomonedas. Una de las principales ventajas que ofrece es la certificación descentralizada de transacciones, que, aplicada en la construcción de identidad digital, le permite a un usuario construir su identidad digital una única vez, reforzarla en el tiempo y mantener control absoluto de su propiedad.

Este documento propone una arquitectura basada en la tecnología Blockchain para la construcción de identidad digital que permita a los usuarios de ésta registrar su identidad digital, compartirla con múltiples entidades y mantener la propiedad de ella en todo momento.

PARTE I. CONTEXTUALIZACIÓN DE LA INVESTIGACIÓN

1. Descripción de la investigación

1.1. Planteamiento del Problema

En el 2017 el porcentaje de personas bancarizadas en Colombia se posicionó en un 80.1%, según la Superintendencia Financiera [2]. De estos, tan solo el 30% realizó sus transacciones por canales electrónicos [3]. La seguridad de la información es uno de los factores críticos con los que deben tratar estas entidades, pues la vulneración de sus sistemas de defensa pone en gran peligro la seguridad financiera de sus clientes.

En un ecosistema de transacciones digital heterogéneo como el de la Banca Virtual, se identifica el reto de diseñar un sistema de construcción de identidad digital, que ofrezca a las personas y empresas clientes la posibilidad de acceder a los productos financieros digitales con confianza.

Para construir un sistema con estas características se propone una arquitectura de identificación digital basada en la tecnología blockchain que permita el registro de usuarios y la certificación de una identidad digital única.

1.1.1. Formulación del problema

¿Cómo aplicar las ventajas de inmutabilidad y transparencia de la información que ofrece la tecnología blockchain para diseñar un sistema de identificación digital para usuarios del sector financiero?

1.1.2. Sistematización del problema

Con el desarrollo de la investigación se pretenden resolver los siguientes interrogantes:

- ¿Qué características de la tecnología blockchain pueden ser aplicadas a un sistema de identidad digital?
- ¿De qué manera se puede generar una identidad digital única a través de un sistema basado en blockchain?
- ¿Qué información permite generar un registro digital de usuarios dentro de un sistema basado en tecnología blockchain?

1.2. Objetivos de la investigación

1.2.1. Objetivo General

Diseñar la arquitectura de un sistema de identificación digital, haciendo uso de la tecnología blockchain, para que los usuarios de un sistema financiero puedan emplear una identidad única al acceder a los servicios que requieran.

1.2.2. Objetivos Específicos

- Aprovechar los beneficios de la tecnología blockchain, aplicando sus principios a un sistema de identificación digital, con el fin de asegurar un registro único con múltiple consulta de identidad de un usuario.
- Definir el protocolo de creación y consulta de identidad digital, mediante un algoritmo lógico, con el fin de describir la forma de operar del sistema basado en blockchain.
- Especificar el conjunto de atributos que se emplearán, creando un modelo de información pertinente, que permita lograr la identificación segura de usuarios dentro del sistema de identificación digital.

1.3. Justificación de la investigación

1.3.1. Justificación práctica

Esta investigación nace de la necesidad de generar una identidad digital única y verificable, en la que el usuario que provee los datos de identificación continúe siendo dueño de éstos y las entidades interesadas identificarle de manera confiable.

1.4. Hipótesis de trabajo

El diseño de un sistema para construcción de identidad digital de usuarios del sector financiero basado en blockchain le permitirá al titular de la información crear una identificación única y a las entidades interesadas verificar de forma confiable la autenticidad de ésta.

1.5. Marco referencial

1.5.1. Marco Teórico

Blockchain

Se entiende el Blockchain como "un libro de contabilidad público o una base de datos descentralizada; es decir, que en lugar de tener un administrador central, es distribuido y verificado por consenso a los participantes de la red" [4]. En términos generales, esta tecnología es una base red de procesamiento distribuida donde los datos y las tareas de procesamiento de las transacciones son repartidos a los participantes la red, sin embargo esta tecnología no consiste en una ´única técnica, sino que se vale de la criptografía, las matemáticas, distintos modelos económicos, combinados con redes "peer-to-peer" (punto a punto) y algoritmos de consenso distribuido para resolver el problema tradicional de sincronización de bases de datos distribuidas. Es una construcción de infraestructura que integra muchos campos. Los siguientes son los seis beneficios de Blockchain [5]:

- *Descentralización*: La característica principal de Blockchain, hace que no se necesite confiar en un ´único nodo centralizado, por el contrario cada transacción es controlada, compartida y autorizada por todos los nodos que participan de la red.
- *Transparencia*: Los datos almacenados por los sistemas de Blockchain son trasparentes para cada nodo. Como si fueran libros contables, cada transacción queda registrada y cualquier nodo puede consultarla.
- *Código abierto*: La mayoría de los sistemas de Blockchain son abiertos para cualquier individuo que desee participar en la red, es decir, cualquiera puede acceder a la información de la red. Los registros pueden ser verificados públicamente además de que las personas pueden usar esta tecnología para crear cualquier aplicación que quieran.
- *Autonomía*: Gracias a la base del consenso, cada nodo en el sistema de Blockchain puede transferir o actualizar datos de manera segura: la idea es poder confiar tanto en una sola persona, como en todo el sistema, ya que nadie puede alterarlo.
- *Inmutabilidad*: Cualquier registro se preservará para siempre ya que las operaciones que se realizan no se pueden alterar y son únicas. Las transacciones efectuadas son realizadas con base a un sistema criptográfico, lo que significa que permite que las operaciones sean casi imposibles de hackear.
- *Anonimidad*: La tecnología de Blockchain solucionó el problema de confianza entre nodos, es decir, la transferencia de datos. Inclusive, cada transacción puede ser anónima. Solo es necesario conocer la dirección de Blockchain de la persona.

Arquitectura de Blockchain

Tipos de Blockchain

Las implementaciones de Blockchain pueden ser divididas brevemente en tres tipos: [6]

- *Blockchain distribuida*: En esta red cualquier nodo puede auditar las transacciones y verificarlas, de igual manera puede participar en el proceso para obtener un consenso. Ejemplos de este tipo de blockchain son Bitcoin y Ethereum son Blockchain. En la imagen se ilustra gráficamente la estructura de esta

Imagen 1. Blockchain distribuida. Fuente: [7]

Blockchain federadas: Esto significa que existen varios nodos escogidos que tiene una autoridad superior a los demás y son quienes deciden qué nodo puede ser seleccionado para resolver una tarea en la red, usualmente estos nodos tienen asociaciones de negocio. Los datos en la Blockchain pueden ser públicos o privados, lo que la convierte en un modelo parcialmente descentralizado. Por ejemplo HyperLedger y R3CEV son ejemplos de Blockchain federadas. La imagen 2 ilustra la distribución de este tipo de red

Imagen 2. Blockchain federada. Fuente: [7]

- *Blockchain privada:* En este caso los nodos están restringidos, lo que implica que no cualquier persona puede establecerse como nodo en esta blockchain un ente central tiene la autoridad de gestión de acceso tanto de nuevos nodos como de los nuevos datos de manera estricta. La imagen 3 ilustra la arquitectura de esta blockchain.

Imagen 3. Blockchain privada. Fuente: [7]

Algoritmos de consenso en Blockchain

Un algoritmo de consenso en el entorno de Blockchain es un conjunto de reglas para determinar si una adición de un bloque nuevo a la cadena es válida o no. Este algoritmo es acordado por todos los nodos de la red. Existen varias estrategias para este algoritmo, de los cuales se presentan los dos más comunes a continuación: [7]

- PoW (Proof-of-Work) Es un algoritmo en el que se genera una pieza de datos, la cual necesita alta capacidad de procesamiento para ser construida pero que a su vez es fácil de verificar. En una cadena de bloques los mineros compiten entre sí para confirmar un bloque, es decir, registrar sus transacciones. Quien gane la competencia recibe una recompensa, de esta manera se incentiva a que los nodos de la Blockchain tengan cada vez más capacidad de cómputo.
- PoS (Proof-of-Stake) De manera similar al algoritmo PoW, es un algoritmo de consenso el cual busca conformar un bloque. Sin embargo, en este la probabilidad de recibir la recompensa es directamente proporcional a la cantidad de bloques que se poseen. De esta manera quienes están más interesados en la seguridad de la cadena son quienes aseguran las transacciones de la misma.

1.5.2. Marco Conceptual

A continuación se da una explicación breve de los conceptos relevantes dentro del marco de esta investigación, que ayudarán a contextualizar los términos empleados a lo largo del documento. La mayoría de ellos ha sido tomada de las definiciones dadas en [8]

Identidad Física: La identidad en el mundo físico está asociada con una serie de rasgos característicos de la persona, que van desde el nombre, edad, sexo hasta nivel académico, cultural, social, incluyendo sus gustos y/o preferencias. Un elemento asociativo de la identidad de la persona son sus nombres, apellidos, Número de identificación y en el caso de las empresas, la razón social y NIT.

Identidad Digital: Identidad digital o Identidad 2.0 es todo lo una persona realiza en el ciberespacio e incluye tanto sus actuaciones como la forma en que otros lo perciben en la red. La identidad se crea conforme una persona va actuando dentro del espacio digital. Todas sus acciones constituyen parte de su identidad, así como sus omisiones o todo lo que deja de hacer. Todos los datos como imágenes que sube, comentarios que escribe, clics sobre los enlaces por donde la persona navega, contexto de su interacción y el lugar donde estén accesibles sus datos le identifican y conforman el perfil digital.

Identidad Digital auto-soberana: En se define como las tecnologías que le dan a los individuos y compañías la habilidad de controlar y administrar su propia identidad digital. Es propuesta además como la octava capa en el protocolo de comunicaciones TCP. Permitiendo al usuario autenticar los productos y servicio s que usa en internet.

Firma digital: Es un valor numérico que permite verificar la autenticidad del remitente un mensaje. Se genera mediante un procedimiento matemático en el cual se calculan llaves hash de la clave pública y la clave privada del remitente. El receptor puede comprobar que el mensaje fue enviado por/desde ese remitente y que, además, no fue alterado por nadie más.

Criptografía: Es la práctica y estudio de las técnicas utilizadas para comunicar mensajes de manera segura, previniendo la interceptación de información por parte de un atacante.

Función Hash: Se le denomina función hash o de dispersión $\boldsymbol{h}(\boldsymbol{K})$, a la función que transforma una llave K en una dirección, la cual se usa como base para la búsqueda y almacenamiento de registros.

Clave privada (Private Key): es un texto alfanumérico que se utiliza en criptografía como factor de cálculo para generar un mensaje cifrado. También hay procedimientos matemáticos para generar claves privadas adecuadas, que tengan ciertas características necesarias dentro de la aritmético modular para servir su propósito criptográfico.

Clave pública (Public Key): es un texto alfanumérico que se utiliza en criptografía como factor de cálculo para obtener un mensaje a partir de su valor cifrado.

Sistema Distribuido: Una red de computadoras intercomunicadas para compartir sus recursos informáticos. La computación distribuida une miles de terminales individuales con el fin de crear un gran sistema con poder de computación masivo.

Red Peer-to-Peer (P2P): Red descentralizada que no tiene clientes ni servidores fijos, sino que tiene una serie de nodos que se comportan simultáneamente como clientes y servidores de los demás nodos de la red.

Prueba de trabajo (Proof of Work PoW): es el sistema mediante el cual se verifican transacciones en una red blockchain. La prueba consiste en la resolución de problemas matemáticos (un hash) que tiene una variable que lo dificulta. Resolver la prueba con éxito suele requerir tiempo y es por esto que en última instancia, este sistema condiciona la capacidad de contribución al poder computacional del usuario.

1.5.3. Marco Espacial

El desarrollo de esta investigación se realizará definiendo como referente espacial los productos digitales creados por el laboratorio digital de las entidades bancarias del grupo AVAL (Banco de Bogotá, Banco AV Villas, Banco de Occidente y Banco Popular), que prestan sus servicios a nivel nacional dentro del territorio colombiano, sin tener relación alguna con éstos más allá que la definida en esta sección.

1.5.4. Marco Legal

La ley 1581 de 2012 define las disposiciones generales de Habeas Data y regula el manejo de la información contenida en bases de datos personales, en especial la financiera, crediticia y comercial, de servicios y la proveniente de terceros países [1]. Esta ley, en su artículo 3° define algunos conceptos muy importantes para el marco de esta investigación. Estos son:

a) Titular de la información. Es la persona natural o jurídica a quien se refiere la información que reposa en un banco de datos y sujeto del derecho de hábeas data y demás derechos y garantías a que se refiere la presente ley;

b) Fuente de información. Es la persona, entidad u organización que recibe o conoce datos personales de los titulares de la información, en virtud de una relación comercial o de servicio o de cualquier otra índole y que, en razón de autorización legal o del titular, suministra esos datos a un operador de información, el que a su vez los entregará al usuario final. Si la fuente entrega la información directamente a los usuarios y no, a través de un operador, aquella tendrá la doble condición de fuente y operador y asumirá los deberes y responsabilidades de ambos. La fuente de la información responde por la calidad de los datos suministrados al operador la cual, en cuanto tiene acceso y suministra información personal de terceros, se sujeta al cumplimiento de los deberes y responsabilidades previstas para garantizar la protección de los derechos del titular de los datos;

c) Operador de información. Se denomina operador de información a la persona, entidad u organización que recibe de la fuente datos personales sobre varios titulares de la información, los administra y los pone en conocimiento de los usuarios bajo los parámetros de la presente ley. Por tanto el operador, en cuanto tiene acceso a información personal de terceros, se sujeta al cumplimiento de los deberes y responsabilidades previstos para garantizar la protección de los derechos del titular de los datos. Salvo que el operador sea la misma fuente de la información, este no tiene relación comercial o de servicio con el titular y por ende no es responsable por la calidad de los datos que le sean suministrados por la fuente.

d) Usuario. El usuario es la persona natural o jurídica que, en los términos y circunstancias previstos en la presente ley, puede acceder a información personal de uno o varios titulares de la información suministrada por el operador o por la fuente, o directamente por el titular de la información. El usuario, en cuanto tiene acceso a información personal de terceros, se sujeta al cumplimiento de los deberes y responsabilidades previstos para garantizar la protección de los derechos del titular de los datos. En el caso en que el usuario a su vez entregue la información directamente a un operador, aquella tendrá la doble condición de usuario y fuente, y asumirá los deberes y responsabilidades de ambos;

e) Dato personal. Es cualquier pieza de información vinculada a una o varias personas determinadas o determinables o que puedan asociarse con una persona natural o jurídica. Los datos impersonales no se sujetan al régimen de protección de datos de la presente ley. Cuando en la presente ley se haga referencia a un dato, se presume que se trata de uso personal. Los datos personales pueden ser públicos, semiprivados o privados;

f) Dato público. Es el dato calificado como tal según los mandatos de la ley o de la Constitución Política y todos aquellos que no sean semiprivados o privados, de conformidad con la presente ley. Son públicos, entre otros, los datos contenidos en documentos públicos, sentencias judiciales debidamente ejecutoriadas que no estén sometidos a reserva y los relativos al estado civil de las personas;

g) Dato semiprivado. Es semiprivado el dato que no tiene naturaleza íntima, reservada, ni pública y cuyo conocimiento o divulgación puede interesar no sólo a su titular sino a cierto sector o grupo de personas o a la sociedad en general, como el dato financiero y crediticio de actividad comercial o de servicios a que se refiere el Título IV de la presente ley.

h) Dato privado. Es el dato que por su naturaleza íntima o reservada sólo es relevante para el titular.

i) Agencia de Información Comercial. Es toda empresa legalmente constituida que tenga como actividad principal la recolección, validación y procesamiento de información comercial sobre las empresas y comerciantes específicamente solicitadas por sus clientes, entendiéndose por información comercial aquella información histórica y actual relativa a la situación financiera, patrimonial, de mercado, administrativa, operativa, sobre el cumplimiento de obligaciones y demás información relevante para analizar la situación integral de una empresa. Para los efectos de la presente ley, las agencias de información comercial son operadores de información y fuentes de información.

De acuerdo con [9] la política de tratamiento de datos personales del grupo Aval, se entiende que una entidad financiera puede obtener el rol tanto de Fuente de información como de Operador de información, por lo tanto está obligado, entre otras a conservar la información bajo condiciones de seguridad que propendan por evitar su adulteración, pérdida, consulta, uso o acceso no autorizado o fraudulento.

Este marco referencial establece las bases que hacen posible que un sistema de identidad digital basado en blockchain sea una alternativa viable para la resolución del problema planteado en la sección 2.

1.6. Metodología de la investigación

1.6.1. Tipo de estudio

Dado que la investigación propone realizar un diseño inicial de un sistema, partiendo de las ventajas que ofrece la tecnología blockchain, el tipo de ejercicio que se pretende desarrollar es de tipo Estudio Proyectivo.

1.6.2. Método de investigación

Esta investigación sigue el método de investigación inductivo – deductivo, como quiera que, parte de la generalidad de la tecnología blockchain y la aplica a una situación particular, como lo es la construcción de la identidad digital de los clientes de los productos digitales de las entidades financieras en Colombia.

1.6.3. Fuentes y técnicas para la recolección de la información

A continuación se indican las principales fuentes de información que soportarán la investigación:

- Documentación teórica sobre la tecnología Blockchain y su aplicabilidad en Sistemas de Identidad Digital.
- Información legal sobre la implementación de sistemas de Identidad Digital en Colombia y demás leyes y normas que expida el Ministerio de Tecnología, Información y Comunicaciones (MinTIC) y otros a que haya lugar.
- Informes, artículos y demás fuentes de información de dominio público sobre la situación actual de la construcción de identidad digital en Colombia.

1.6.4. Tratamiento de la información

La información obtenida como resultado de la presente investigación será tratada exclusivamente con los fines de la misma y en el desarrollo de ésta se velará en todo momento porque la propiedad intelectual de todas las fuentes de información citadas (cuando aplique) sea protegida.

1.7. Estudio de sistemas previos

En esta sección se mencionarán algunos sistemas relacionados con la identificación descentralizada de usuarios en una red digital.

IRMA: IRMA, acrónimo de *"I Reveal My Attributtes"* en español, "Yo revelo mis atributos" es una solución de Credencial basada en Atributos desarrollada por la universidad de Radbound [10]. En este sistema, el dueño de los atributos es quien decide compartir con otros miembros un subconjunto específico de los atributos que ha definido, lo que lo hace, muy confiable para la privacidad de los participantes.

La solución se basa en los siguientes requerimientos:

- No transferible: Es decir que nadie puede tomar la titularidad de un atributo por alguien más.
- Desvinculación del emisor: El dueño del sistema no debe ser capaz de conocer las transacciones de los participantes.
- Desvinculación de otros participantes: Un participante al cual le haya sido compartido un atributo por parte de otro, no puede utilizar éste para un fin diferente de la transacción que se contrata entre los dos.
- Revocación: Un token perdido o robado deberá poder bloquearse dentro del sistema para evitar su uso indebido.

En este sistema un teléfono inteligente contiene una llave secreta, la cual es usada para crear credenciales no transferibles. En una transacción, tras verificar la autenticidad de esta llave, los emisores pueden seleccionar qué atributos compartirán, los cuales serán usados para realizar únicamente dicha transacción.

iDIN: iDin, también conocida como "BankID" es una iniciativa privada desarrollada por los bancos holandeses bajo supervisión del gobierno estatal [11], la cual le permite a clientes de estos autenticarse en múltiples comercios utilizando su identificación bancaria a través de la plataforma.

En la imagen 4 se ilustra el modelo de funcionamiento del sistema, que desde la perspectiva del cliente funciona de la siguiente manera:

- El cliente se autentica en la plataforma iDIN con su banco.
- El banco del cliente verifica la autenticación y autorización del cliente.
- El banco del cliente envía la identidad al banco del comercio.

- El comercio es informado por su banco de una autenticación exitosa por parte del cliente.

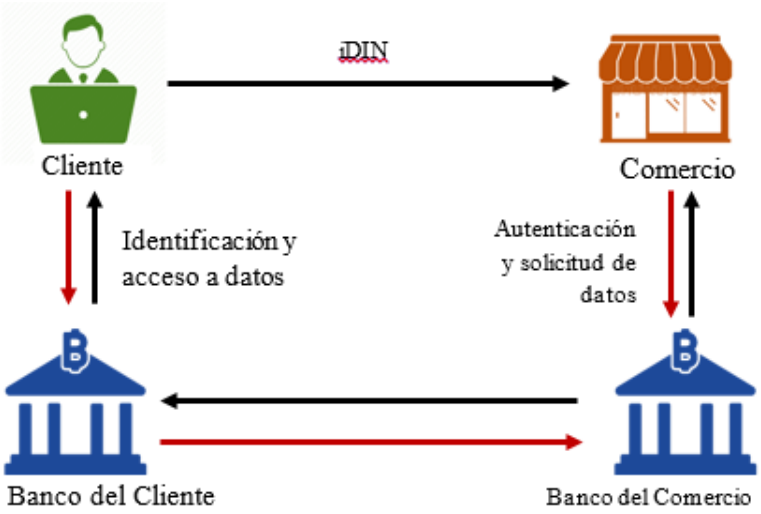

Imagen 4. Interacciones del sistema iDIN. Fuente: Autor

Onename.io: OneName es una plataforma en la que cualquier persona puede crear un ID que se guarda en un sistema blockchain y puede funcionar como una contraseña en la web. [12] La verificación de identidad es realizada usando múltiples proveedores de identidad.

Aunque en la actualidad el ID generado en esta plataforma sólo puede ser utilizado en algunas redes sociales se espera que en el futuro pueda enlazarse también a credenciales legales como números de seguro social y el número de la licencia de conducción. De cualquier manera no se espera que esta integración se logre realizar en el futuro cercano.

BIDaaS: Blockchain Based ID As A Service [13] es una propuesta de servicio de autenticación basado en blockchain en el cual existen tres actores:

- El usuario.
- El proveedor
- La parte asociada.

En este sistema existe una relación de confianza entre el proveedor y la parte asociada y entre el proveedor y el usuario. En primer lugar, el usuario se registra en el sistema del proveedor con sus datos y éste le asigna una llave privada que le dará acceso únicamente al usuario a su información. Cuando el usuario quiere acceder a la parte asociada ésta, lanza una solicitud de autenticación a la blockchain, quien mediante la técnica de certificación descentralizada certifica la identidad de la persona. Si la parte asociada requiere conocer los datos del usuario, solicita al proveedor, únicamente para esa transacción la identidad de la persona.

PARTE II. DESARROLLO DE LA INVESTIGACIÓN

2. Sistema de Identificación Digital Basado en Blockchain. SIDiB

2.1. Participantes

En el sistema intervienen tres participantes:

- Proveedor de Identidad: o simplemente proveedor, cumple las funciones de registrador dentro de SIDiB. Actúa como custodio de la información de identidad de los titulares.
- Asociado: Entidad (financiera) que ofrece un servicio al titular y requiere para ello autenticar al cliente. Cada asociado aporta dos tipos de participantes internos a la blockchain:
  - Nodo de autoridad. Encargado de certificar las transacciones en la blockchain.
  - Nodo de lectura: Con acceso de lectura para consultar los certificados de identidad en la blockchain.
- Usuario: Es el Titular de la información, quien registra una identidad en el proveedor. El usuario consume los servicios ofrecidos por el asociado, usando la red blockchain para avalar su identidad sin tener que crear una nueva identidad en el sistema de asociado.

La imagen 5 muestra las relaciones entre estos tres actores y la blockchain.

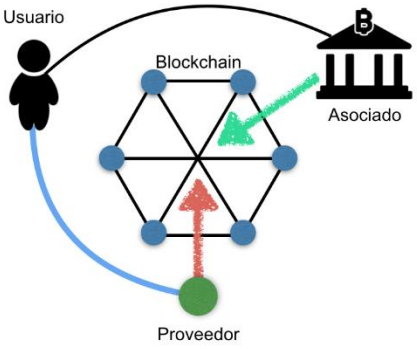

Imagen 5. Relaciones entre los usuarios de SIDiB. Fuente: Autor

2.2. Tipo de Blockchain

La arquitectura del SIDiB funciona sobre una blockchain federada, en la que se designan nodos de autoridad en representación de cada entidad financiera. Estos nodos serán los encargados de validar cada bloque y agregarlo a la cadena cuando un usuario solicite la autenticación en el sistema. La imagen 6 ilustra esta arquitectura.

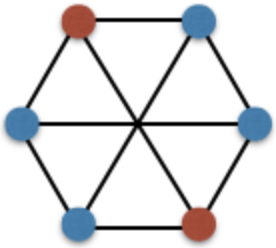

Imagen 6. Arquitectura de distribución de SIDiB. Fuente: Autor

2.3. Almacenamiento de Identidad Digital

El proveedor de identidad, como custodio de la información sensible de los usuarios cuenta con un sistema de persistencia, responsable de almacenar tanto los datos del titular como el Hash que los identifica y la llave privada de autoridad necesaria para la desencripción. Para esto contará con una base de datos distribuida, con una réplica esclava en cada una de las entidades y administrada por un sistema independiente centralizado. La imagen 7 ilustra la arquitectura de ésta forma de persistencia.

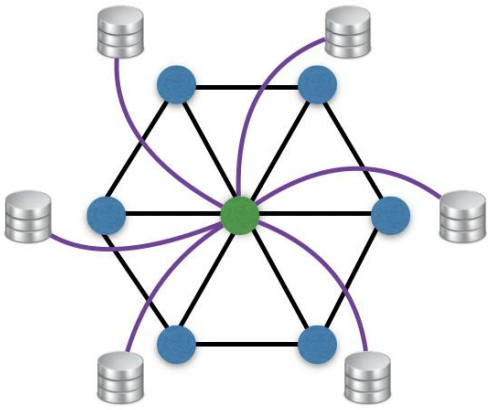

Imagen 7. Almacenamiento distribuido en SIDiB. Fuente: Autor

2.4. Interacción dentro del SIDiB

El proveedor administra las identidades dentro de la blockchain. El asociado tiene acceso únicamente de lectura a la blockchain, de tal manera que por sí sólo no puede hacer cambios en la información que por ella circula. El proveedor es responsable de generar el ID Digital de un usuario e insertarlo dentro de la blockchain. Estos datos son insertados bajo una firma digital hecha con la llave privada del proveedor, de tal manera que únicamente él puede desencriptar esta información.

El asociado lee el ID Digital y la llave pública de la blockchain cuando el usuario solicita acceso a su servicio con el ID Digital. El asociado está en capacidad de confirmar si este ID coincide con el registrado en la blockchain. Si la confirmación es exitosa el asociado procede a realizar la autenticación de doble vía con los datos obtenidos. De esta forma el asociado puede autenticar a un cliente sin tener un previo registro de éste en sus sistemas.

Tras la autenticación exitosa el asociado puede distinguir al usuario por medio del ID Digital. Si se requiere información adicional del usuario (como nombre, dirección y número de teléfono se comunica con el proveedor por medio de un canal seguro previamente establecido.

La tabla define las convenciones que se usarán para definir en detalle las diferentes etapas del proceso de interacción.

Tabla 1. Convenciones. Fuente: Autor

Convención	Definición
Y usr	ID Digital del usuario
K(usr)	Llave privada del usuario
Kusr	Llave pública del usuario
K(prv)	Llave privada del proveedor
Kprv	Llave pública del proveedor
K(asc)	Llave privada del asociado
Kasc	Llave pública del asociado
SigK(usr)	Firma hecha con la llave privada del usuario
SigK(prv)	Firma hecha con la llave privada del proveedor
EcKusr	Encripción hecha con la llave pública del usuario
EcKprv	Encripción hecha con la llave pública del proveedor
R	Nonce

2.5. Creación de ID Digital

En el primer paso de la interacción. En esta etapa el usuario crea la pareja de llaves de encripción *Kusr* y *K(usr)* usando el algoritmo de encripción RSA. y almacena de forma segura *Kusr*. A continuación se aplica el algoritmo SHA256 sobre *K(usr)*, para obtener Yusr, como se evidencia en el diagrama de estados de la imagen 8.

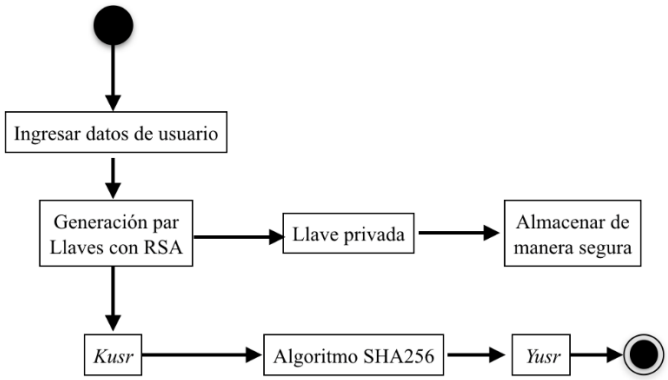

Imagen 8. Diagrama de estados. Creación de ID Digital. Fuente: Autor

2.6. Registro en la blockchain

Kusr y el *Yusr* generado en la etapa anterior son luego transferidos del usuario al proveedor mediante un canal seguro previamente establecido. El proveedor crea una firma digital a partir de de *Kusr* y *Yusr*, usando su propia llave privada *Kprv*. A continuación el proveedor registra *Kusr* y *Yusr* con la firma creada SigKprv*(Kusr, Yusr)* en la blockchain. Este registro es completado mediante una transacción blockchain que es transmitida a los nodos. Esta operación es finalmente almacenada en la blockchain, como se indica en el diagrama de estados de la imagen.

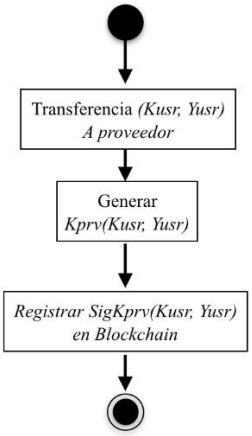

Imagen 9. Diagrama de estados. Registroen SIDiB. Fuente: Autor

2.7. Autenticación de doble vía

Cuando el usuario necesita acceso a un servicio ofrecido por el asociado, sólo debe presentar *Yusr* con un número Nonce *r* al asociado, en un mensaje *M1 = (Yusr, r,* SigK(usr)*(Yusr, r))*. Cuando el asociado recibe la solicitud de acceso al servicio del usuario, en primer lugar accede a la blockchain para verificar la existencia de *Yusr* en los registros de la blockchain. Si existe, entonces el asociado solicita al usuario *Kusr* y con ella valida *M1*. Tras el resultado exitoso de esta validación, el asociado envía un mensaje *M2 = (Yusr, r+1, EcKusr(Yusr, r+1, Kasc))*. Al recibir M2, el usuario desencripta el mensaje con *Kusr* y valida con *r+1*. Como un resultado de esta operación el usuario obtiene *Kasc* de *M2*. Posteriormente envía un mensaje *M3=(Yusr, r+2,EncKusr(Yusr,r+2))* al asociado, quien desencripta el mensaje con *Kasc y valida el mensaje con r +2*. De esta manera se establece la autenticación de doble vía entre el usuario y el asociado en el SIDiB.

La arquitectura propuesta y el modelo de comunicación planteados han sido desarrollados con base en la solución expuesta en [13], aprovechando las ventajas que allí se indican, realizando las adaptaciones de distribución de bases de datos y modelo de información definidos en este capítulo.

2.8. Modelo de información

Como parte de la actividad económica en la que actúa SIDiB se debe establecer un conjunto de datos que sea de interés para los asociados (entidades financieras) del sistema.

Un cliente de cualquiera de estas entidades provee tres tipos de conjuntos de datos para acceder a la mayoría de servicio que estos prestan.

- Datos de identificación *(DI)*: Datos de identificación de un ciudadano colombiano.
- Datos de contacto *(DC)*: Datos que le permiten al asociado del SIDiB ponerse en contacto con un cliente directa o indirectamente en un momento dado
- Datos de historial financiero *(DF)*: Datos de carácter financiero que el usuario ha suministrado a los asociados en algún momento como parte de una negociación de servicios y hacen parte de un histórico, a partir del cual se puede inferir un comportamiento financiero del usuario. Estos datos pueden consultarse, por ejemplo con el fin de verificar su perfil de riesgo como sujeto de crédito.
- Datos de Identificación SIDiB (BID): Datos que identifican al usuario dentro del SIDiB

En la tabla 2 se definen los datos que hacen parte del modelo de información propuesto agrupadas por los tipos mencionados anteriormente:

Tabla 2. Datos del modelo de información. Fuente: Autor

Grupo	Identificador	Dato	Descripción
DI	Tipo_nid	Tipo de identificación	Tipo de identificación
DI	Num_nid	Número de identificación	Número de identificación
DI	nombre	Apellidos y Nombres	Apellidos y Nombres
DI	Fecha_n	Fecha de Nacimiento	Fecha de Nacimiento
DI	Lugar_n	Lugar de nacimiento	Lugar de nacimiento
DI	Fecha_e	Fecha de expedición del documento	Fecha de expedición del documento
DI	Lugar_e	Lugar de expedición del documento	Lugar de expedición del documento
DI	Genero	Género	Género
DC	Teléfono	Teléfono	Teléfono
DC	Ciudad_r	Ciudad de residencia	Ciudad de residencia
DC	Dirección_r	Dirección de residencia	Dirección de residencia
DC	Referencia_f	Referencia familiar	Referencia familiar
DC	Referencia_p	Referencia personal	Referencia personal
DF	Reputación	Reputación	Puntaje asignado al usuario de acuerdo a su comportamiento dentro de la blockchain.
DF	Ocupación	Ocupación	Actividad en la que se emplea el usuario.
DF	Ingresos	Ingresos mensuales	Ingresos de un cliente, verificados por una fuente confiable
DF	Egresos	Egresos mensuales	Egresos que declara el cliente
DF	Activos	Activos	Activos verificados del cliente
DF	Pasivos	Pasivos	Pasivos verificados del cliente
DF	Autenticaciones	Autenticaciones exitosas Bc	Histórico de identificaciones del usuario dentro de la Blockchain
DF	Servicios_consumidos	Servicios de asociados consumidos	Histórico de servicios que el usuario ha consumido de los asociados dentro de la blockchain
DF	Asociados	Asociados con los que tiene relación	Asociados del SIDiB con los que el usuario tiene relación.
DF	Saldo_asociados	Saldo con Asociado	Estado de saldo que el usuario tiene con el asociado en un momento determinado
DF	Débitos_asociados	Débitos con Asociado	Transacciones de tipo débito que el usuario ha hecho con el asociado
DF	Créditos_asociados	Créditos con Asociado	Transacciones de tipo débito que el usuario ha hecho con el asociado

DF	Tiempo_asociados	Tiempo de relación con Asociado	Antigüedad de la relación entre el usuario y el asociado
BID	DID	ID Digital	Hash que identifica particularmente al usuario dentro de la blockchain
BID	Kusr	Llave pública	Llave que se utiliza para cifrar el modelo de información dentro de la blockchain.

Las relaciones entre estos datos se diagraman en el modelo de información ilustrado en la imagen 10.

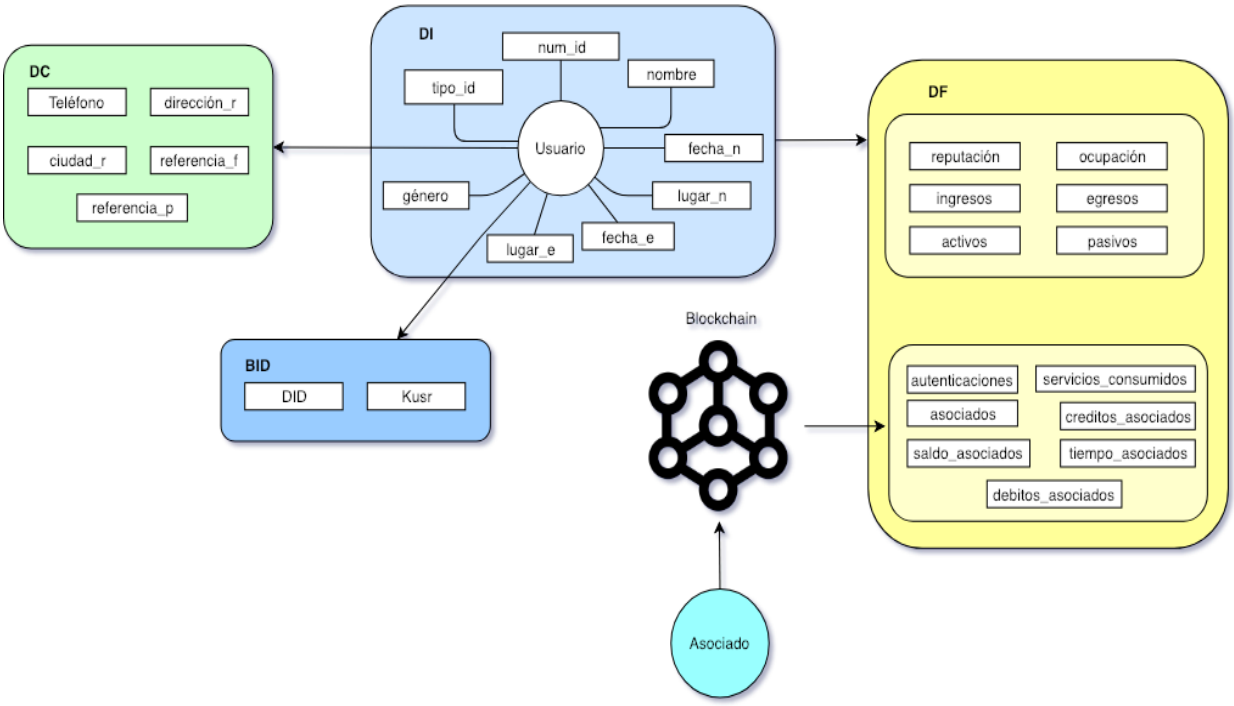

Imagen 10. Modelo de información. Fuente: Autor

3. Arquitectura del Sistema SIDiB

3.1. Negocio

Los diagramas a continuación muestran el punto de vista del negocio. En el desarrollo de éstos se evidencian las diferentes relaciones e interacciones entre los actores y los productos de negocio.

3.1.1. Punto de Vista de la Organización

Este punto de vista permite ilustrar las relaciones entre los diferentes participantes del sistema. En él se evidencia que entre el usuario y el asociado se presenta una conexión indirecta, dada a través del certificador. La imagen 10 ilustra este punto de vista.

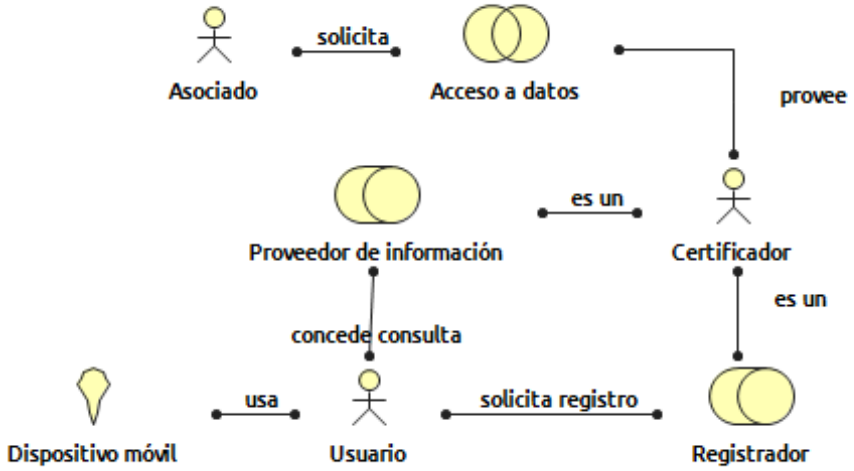

Imagen 11. Punto de vista de la Organización. Fuente: Autor

3.1.2. Punto de Vista del Producto

En este punto de vista se pone en evidencia las relaciones de los diferentes servicios del producto y los roles que interactúan en ellos. Tanto los procesos de negocio como los servicios que prestan permiten construir el producto Identidad digital, en el cual, desde diferentes acercamientos, son consumidores tanto el asociado como el usuario. La imagen 11 ilustra este punto de vista.

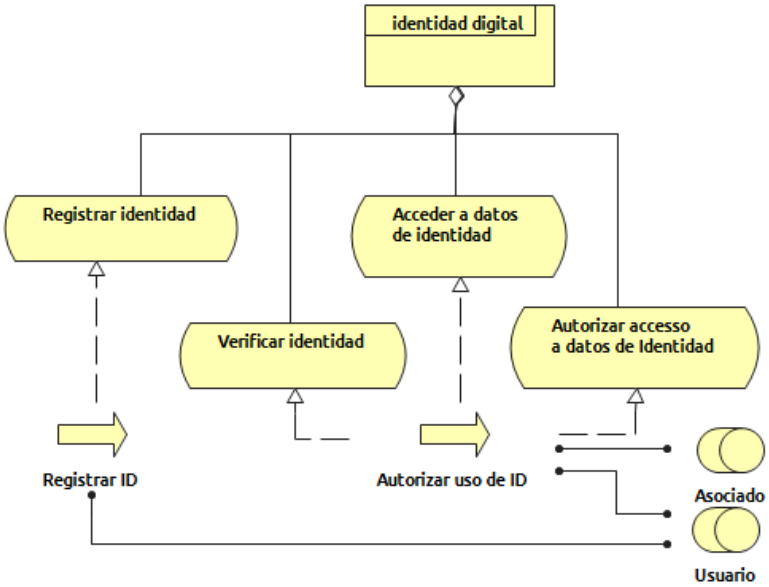

Imagen 12. Punto de vista del producto. Fuente: Autor

3.1.3. Punto de vista de Proceso de Negocio

En el punto de vista de proceso de negocio se diagraman los procesos y las interacciones que permiten realizar la verificación de Identidad dentro del sistema. Esto se logra a partir de la colaboración de cuatro procesos de negocio fundamentales: Registrar ID, autorizar ID, Generación de par llaves y autenticación de usuario. La imagen 12 ilustra este punto de vista.

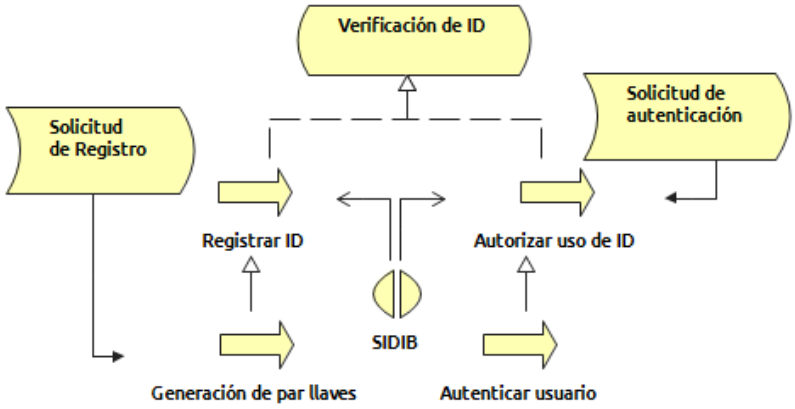

Imagen 13. Punto de vista de proceso de negocio. Fuente: Autor

3.2. Aplicación

En los siguientes diagramas se describe la arquitectura de la solución desde distintos puntos de vista, que logran cumplir con la visión de negocio descrita anteriormente.

3.2.1. Punto de Vista de Comportamiento de la Aplicación

En este punto de vista los componentes del Sistema BdID, registro, consulta y autorización permiten a un sistema externo validar el ID de una persona teniendo confianza y delegando esta tarea con confianza sobre la blockchain. La imagen 13 ilustra este punto de vista.

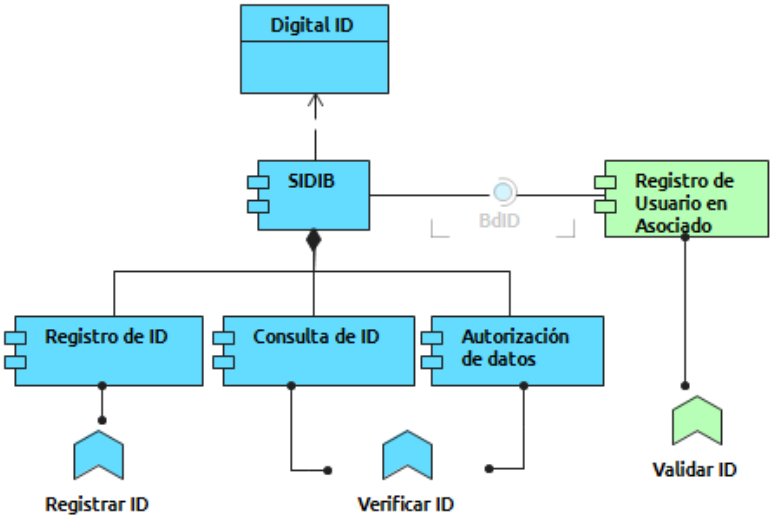

Imagen 14. Punto de Vista de Comportamiento de la Aplicación. Fuente: Autor

3.2.2. Punto de Vista de Uso de la Aplicación

En el punto de vista de uso de la aplicación se extiende la visibilidad del producto desde la exposición de servicios por parte de los componentes de registro y consulta hacia la creación de dos productos de negocio como lo son, la identidad digital como tal y la verificación de identidad, que no es más que el fortalecimiento de la identidad dentro de la blockchain. La imagen 14 ilustra este punto de vista.

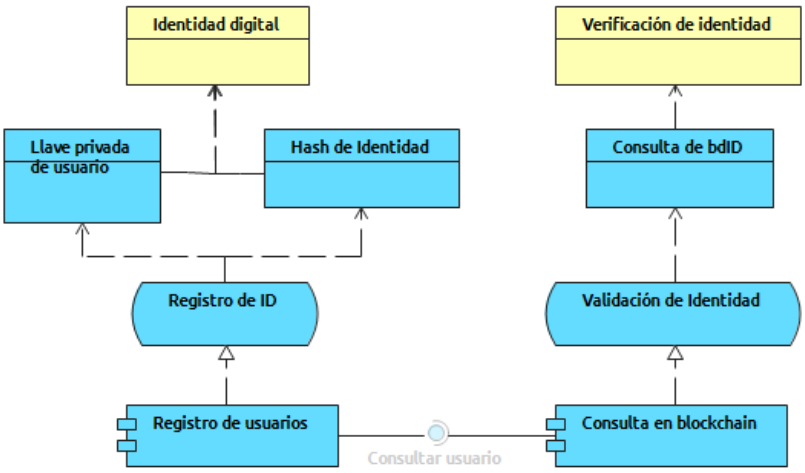

Imagen 15. Punto de Vista de Uso de la Aplicación. Fuente: Autor

3.2.3. Punto de Vista de la Estructura de la Aplicación

En el punto de vista de la estructura de la aplicación se evidencian las interfaces que ofrecen los distintos componentes del sistema, que cuenta como eje central con la blockchain, sobre la cual se pueden hacer operaciones de registro y consulta de ID Digital. La imagen 15 ilustra este punto de vista.

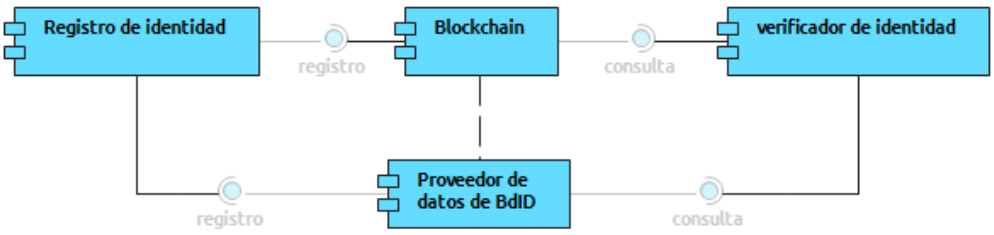

Imagen 16. Punto de Vista de la Estructura de la Aplicación. Fuente: Autor

4. Prototipo de implementación

Dentro de esta investigación se desarrolló un prototipo de implementación que implementa de manera básica la arquitectura planteada en los capítulos 2 y 3 de este documento.

La funcionalidad del prototipo desarrollado permite encriptar y registrar en a blockchain el número de identificación y el nombre de un usuario, para ello utiliza el cifrado asimétrico del algoritmo RSA. Con este cifrado se genera una cadena que contiene los datos del usuario designada con el nombre de *Digital ID*. Así mismo, el prototipo cumple con las funciones de validación de la blockchain basada en encadenamiento de hashes y validación de dígitos de dificultad igual a 2. Esto con el objetivo de demostrar el funcionamiento sin afectar la capacidad de procesamiento de una máquina.

Así mismo el prototipo permite consultar un *Digital ID* dentro de la blockchain y desencriptar asimétricamente la información del usuario contenida en él.

El código fuente con el cual se desarrolló la aplicación se detalla en el Anexo 1 de este documento.

A continuación se describen las principales características del prototipo.

4.1. Bloque

El bloque es la unidad fundamental de la blockchain. Es concebido como un objeto de representación de datos (DTO). Sus atributos son:

1) *Id*: Valor numérico que indica la posición del bloque dentro de la blockchain
2) *Nonce*: Número usado en el cálculo del valor hash del bloque.
3) *Pubic Key (Llave pública):* Uno de los valores producidos en el proceso de encriptamiento asimétrico de la información del usuario.
4) *Digital Id:* Variable alfanumérica que contiene los datos del usuario de manera encriptada.
5) *Previous hash(Hash predecesor):* valor del hash del bloque anterior, con el cual se hace la validación de inmutabilidad del bloque dentro de la blockchain
6) *Hash:* valor calculado a partir de los datos anteriores, con el cual se firma un bloque y se incluye en la blockchain para hacer validaciones de inmutabilidad dentro de la blockchain.

Un bloque también cuenta con la funcionalidad de minar, que cumple con el objetivo de firmar el bloque para hacerlo válido dentro de la blockchain. En el prototipo desarrollado un bloque válido es aquel en el que el valor del hash comienza con los dos primeros caracteres siendo 0. En la red bitcoin, por ejemplo, el último bloque creado (a 30 de enero de 2019) tiene una dificultad (cantidad de ceros con los que comienza el hash) de 5,814,661,935,891.8.([14])

La imagen 17 muestra un bloque válido, con los atributos anteriormente definidos:

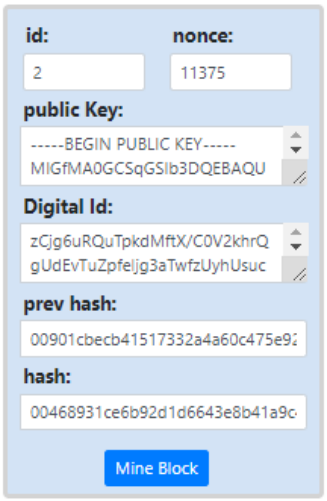

Imagen 17. Bloque. Prototipo desarrollado. Fuente: Autor

4.2. Blockchain

La blockchain es la unión lógica de los bloques. Para el desarrollo del prototipo se creó un objeto de tipo lista a la cual se van agregando los bloques.

Cuando un bloque se va a agregar a la blockchain, se le asignan, el número de ID que tendrá y el hash predecesor (que es el hash del último bloque puesto en la cadena), después se mina. De esta manera se garantiza que se está insertando un bloque válido. La imagen18 ilustra una blockchain válida dentro del sistema.

Al cambiar algún valor en un bloque dentro de la blockchain, éste último vuelve a calcular el hash y realiza una publicación al sistema. Así mismo, existe un suscriptor que escucha estas publicaciones y de inmediato realiza una validación de toda la blockchain. Si el valor del hash predecesor de un bloque es diferente al valor del hash del bloque anterior se invalida toda la cadena.

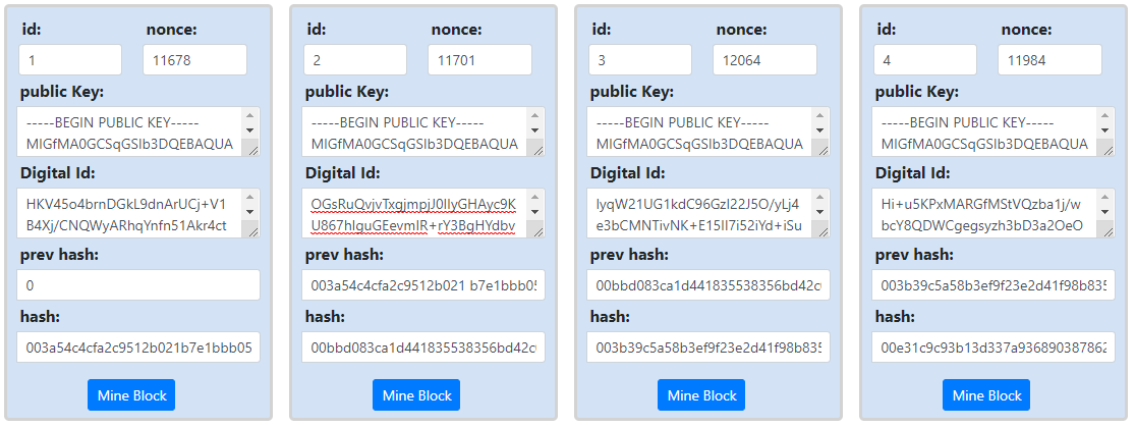

Imagen 18.Blockchain válida. Prototipo desarrollado. Fuente: Autor

La imagen 19 muestra la invalidación de la blockchain ilustrada en la imagen 18, después de haber modificado el valor del Digital Id del bloque 2. En éste escenario, el bloque 1 sigue siendo válido, pero a partir del bloque modificado la cadena queda invalidada.

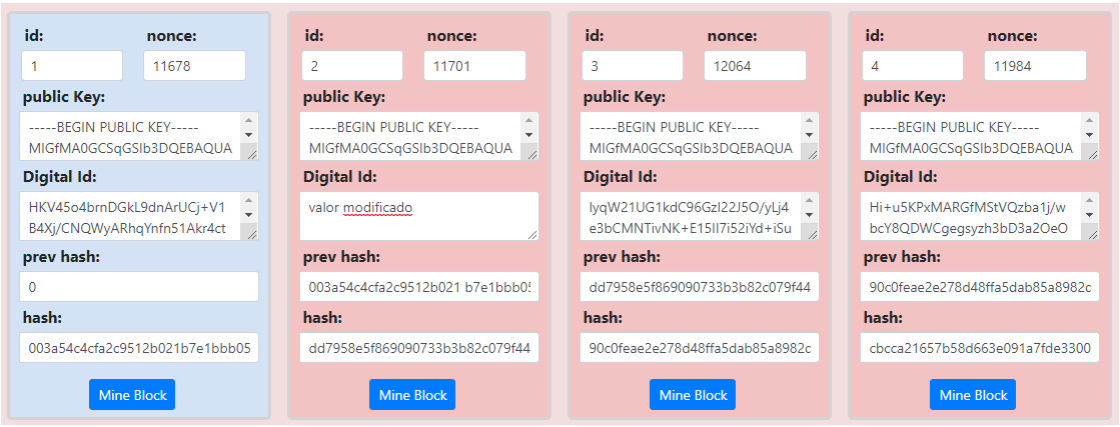

Imagen 19. Blockchain inválida. Prototipo desarrollado. Fuente: Autor

Adicionalmente, se incluyó una restricción en el sistema que evita agregar un nuevo bloque a la blockchain si ésta se encuentra en estado inválido. Esto garantiza que la información a registrar sólo se pueda ingresar en una cadena válida.

4.3. Registro de nuevos usuarios

Se desarrolló un módulo de registro de usuarios, en el cual se ingresan los datos que se cifrarán en incluirán en la blockchain. La imagen 20 ilustra la interfaz de este módulo. Se

incluyó una restricción para evitar ingresar valores nulos al sistema. La imagen 21 ilustra el cambio del módulo como modo de advertencia al incluir datos errados en la blockchain.

Imagen 20. Módulo de registro de usuario. Prototipo desarrollado. Fuente: Autor

Imagen 21. Módulo invalido por datos nulos. Prototipo desarrollado. Fuente: Autor

Al hacer clic sobre el botón de registrar con datos válidos, el módulo aplica el algoritmo RSA para crear las llaves *Kusr* y *K(usr)* definidas en el capítulo 2.4. Y las utiliza para cifrar el id y nombre suministrados. Como producto de este procedimiento se crea el ID Digital, *Yusr*, (también definido en el capítulo 2.4.).

A continuación el módulo registrador crea un nuevo bloque, con los siguientes valores:

- Id: no asignado.
- Nonce: 1111 (Por defecto)
- Public Key: Llave pública creada para este registro.
- Digital ID: Digital Id creado para este registro.
- Previous hash: No asignado.
- Hash: 0 por defecto

Por último el módulo registrador le entrega a la blockchain el nuevo bloque.

Al recibir el nuevo bloque, la blockchain realiza tres procedimientos sobre éste para agregarlo a la cadena:

- asigna los valores de id (secuencia que identifica al bloque dentro de la cadena) y Previous hash (hash del último bloque en la cadena) al bloque.
- Mina el bloque para hacerlo válido.
- Agrega el bloque a la lista

De esta manera queda agregado un nuevo usuario al sistema de identidad digital basado en blockchain.

4.4. Datos privados de un usuario

En el prototipo se incluyó una sección para visualizar los datos privados del usuario, los cuales le permiten identificarse en la blockchain. En un sistema real estos datos son enviados por un canal seguro al usuario, quien los debe custodiar de manera que estén a salvo de hurto. La visualización de estos datos se ilustra en la imagen 22 y se incluye únicamente para efectos demostración del prototipo.

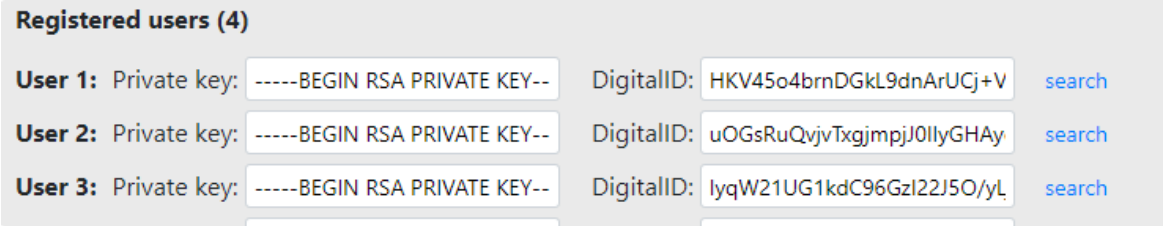

Imagen 22. Visualización de usuarios registrados. Prototipo desarrollado. Fuente: Autor

4.5. Consulta de usuarios

En la arquitectura propuesta, los asociados consultan la blockchain para validar si un usuario existe en el sistema y sus credenciales son válidas. En el prototipo desarrollado se incluyó un módulo de búsqueda de usuarios, de manera análoga a como lo harían los asociados del sistema.

En este módulo, ilustrado en la imagen 23, se realiza la búsqueda de un usuario por su Digital Id, como lo haría un representante del asociado, pero se agrega el dato de la llave privada, para desencriptar los datos de usuario, lo cual en un escenario real se haría en el dispositivo del usuario de manera segura, siguiendo el protocolo descrito en el capítulo 2.7.

Imagen 23. Módulo de consulta de usuarios. Prototipo desarrollado. Fuente: Autor

En la sección de visualización de usuarios, descrita anteriormente, se incluyó la opción *search* para copiar fácilmente los valores de la llave privada y el Digital Id en las cajas de texto del módulo de búsqueda, con el fin de evitar errores en el copiado manual y facilitar la búsqueda ágil de usuarios.

En el proceso de consulta desarrollado para el prototipo, la blockchain busca dentro de los bloques aquel que contenga el Digital Id solicitado. Si lo encuentra, procede a emplear la

llave privada del usuario para desencriptar los datos sensibles de éste y mostrarlos en pantalla. Aquí se unificaron los procedimientos descritos en los capítulos 2.7 y 2.8 para efectos de demostrar el funcionamiento. En un escenario real, para identificar a un usuario, debería bastar con encontrar el Digital Id dentro del sistema y emplear la llave pública almacenada en la blockchain para cifrar los mensajes entre el asociado y el usuario.

Como producto de esta búsqueda se pueden generar tres escenarios:

- Con los valores de la llave privada y el Digital Id suministrados se puede encontrar un usuario y desencriptar sus datos personales. En este caso, el prototipo mostrará una alerta indicando los datos sensibles, como se ilustra en la imagen 24.
- El Digital Id suministrado existe en la blockchain, pero la llave privada suministrada no sirve para desencriptar los datos sensibles. En este caso, el prototipo mostrará una alerta indicando que no se pudo desencriptar la información del usuario, como se ilustra en la imagen 25.
- El Digital Id suministrado no puede ser encontrado en la blockchain. En este caso, el prototipo mostrará una alerta indicando que no se pudo encontrar el usuario, como se ilustra en la imagen 26.

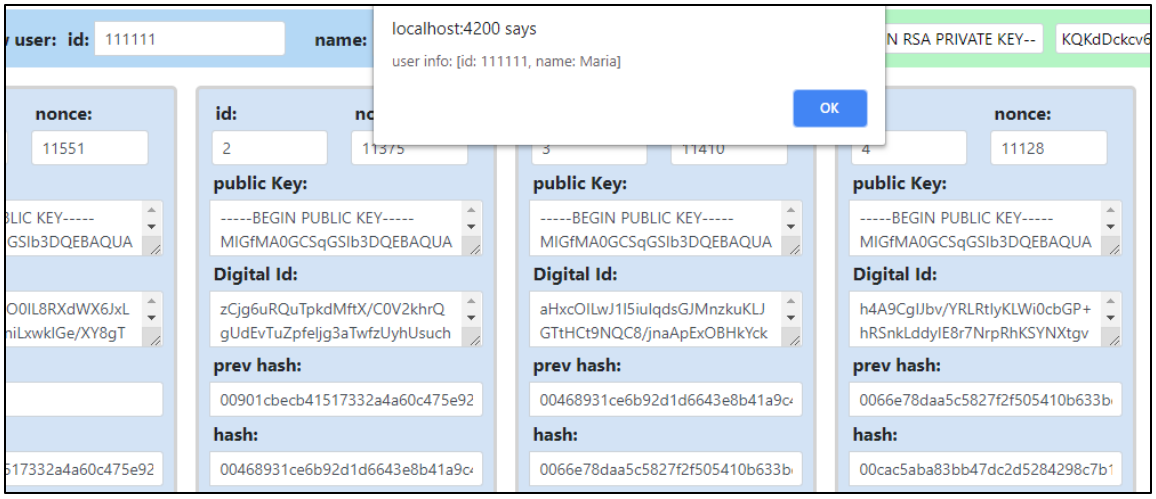

Imagen 24. Consulta exitosa de usuario. Prototipo desarrollado. Fuente: Autor

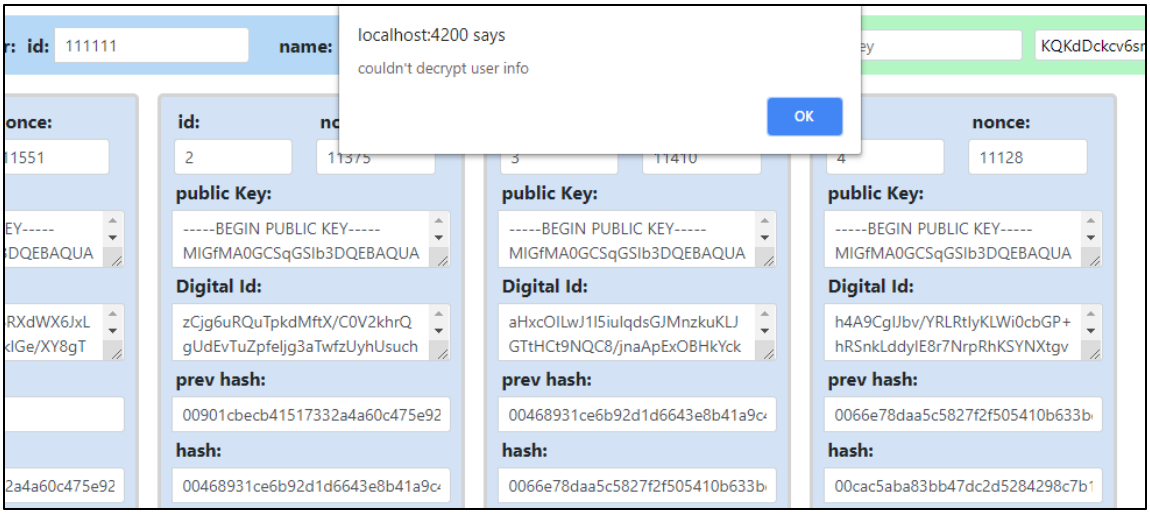

Imagen 25. Llave inválida para desencripción de datos sensibles. Prototipo desarrollado. Fuente: Autor

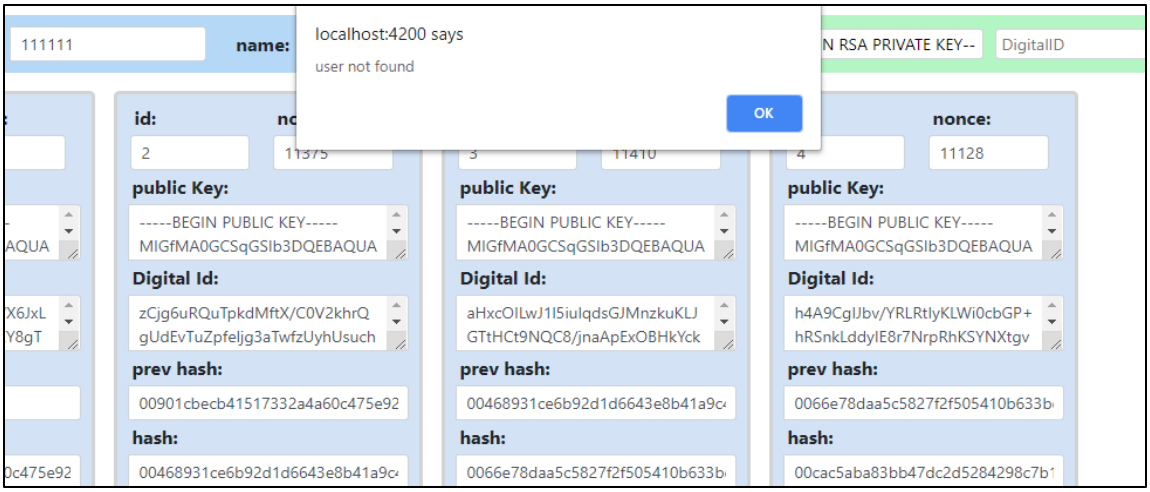

Imagen 26. Digital Id no encontrado. Prototipo desarrollado. Fuente: Autor

PARTE III. CIERRE DE LA INVESTIGACIÓN

5. Análisis final

5.1. Resultados

Como producto de esta investigación se realizó el diseño de la arquitectura de un sistema basado en blockchain que permite a los usuarios el registro de una identidad única en un sistema dentro del cual puede consumir múltiples servicios sin que para esto tenga que perder el dominio de sus propios datos.

La arquitectura propuesta también permite la creación de una comunidad de valor basada en la ganancia multilateral, dado, que los asociados cuentan con la confianza de una fuente de información única, además de liberarse de la responsabilidad de respaldar una base de datos que contiene información sensible de clientes.

Dado que la blockchain que soporta el sistema es de tipo federada, no es necesario adquirir un gran poder de procesamiento para realizar tareas de certificación de bloques, contrario a lo que sucede en una blockchain pública, que emplea algoritmos de consenso, como el llamado "Prueba de Trabajo" (PoW).

5.2. Consideraciones de Seguridad

Una vez que la información del usuario es suministrada por el proveedor al asociado, con la autorización del usuario, esta podría ser almacenada por el asociado en bases de datos propietarias. Esto significa que la propuesta de arquitectura garantice que la información del usuario será usada una única vez. Es necesario entonces proveer un mecanismo para detectar o prevenir el uso no autorizado de esta información.

De igual modo la arquitectura propuesta no provee garantía de reconocimiento de identidad de un usuario. La protección contra suplantación de identidad debe implementarse como una capa adicional al sistema, que se pondrá a prueba en el momento en que una persona se enrole en el SIDiB.

La llave privada de un usuario es el habilitador de acceso a toda la información sensible que pretende custodiar la blockchain, esta debe ser almacenada por el usuario en un lugar seguro. La implementación de este mecanismo se deja a consideración de trabajos futuros.

6. Verificación, contraste y evaluación de los objetivos

Objetivo general: Diseñar la arquitectura de un sistema de identificación digital, haciendo uso de la tecnología blockchain, para que los usuarios de un sistema financiero puedan emplear una identidad única al acceder a los servicios que requieran.

Contraste: La arquitectura del sistema propuesto permite crear una identidad digital única para un usuario dentro de un sistema que cuenta con entidades financieras como asociadas, aunque puede ser adaptada para incluir asociados de naturaleza no necesariamente financiera.

Objetivos específico 1: Aprovechar los beneficios de la tecnología blockchain, aplicando sus principios a un sistema de identificación digital, con el fin de asegurar un registro único con múltiple consulta de identidad de un usuario.

Contraste: En la arquitectura se propuso un sistema de bloques encadenados que almacenan la identidad digital del usuario, tomando como base el diseño de blockchain, logrando con esto que los asociados puedan identificar al usuario sin tener que registrarlo en sus bases de datos propietarias.

Objetivos específico 2: Definir el protocolo de creación y consulta de identidad digital, mediante un algoritmo lógico, con el fin de describir la forma de operar del sistema basado en blockchain.

Contraste: Se definió el protocolo de operación del sistema, describiendo las diferentes interacciones entre los participantes, así como la transformación de la información que circula por el sistema.

Objetivos específico 3: Especificar el conjunto de atributos que se emplearán, creando un modelo de información pertinente, que permita lograr la identificación segura de usuarios dentro del sistema de identificación digital.

Contraste: Se ha definido un modelo de información que agrega valor frente a otros medios de autenticación, pues provee datos obtenidos a partir de la participación del usuario dentro del SIDiB. Estos datos pueden ser utilizados como insumo por los asociados para estudios de riesgo y comportamiento financiero.

7. Prospectiva del proyecto

7.1. Líneas de investigación futuras

Como parte de continuación del proyecto se proponen investigaciones en las siguientes líneas:

- Seguridad sobre blockchain
- Identidad de personas en múltiples servicios.
- Modelos de comunicación entre diferentes blockchain

7.2. Trabajos de Investigación futuros

Como parte de la ampliación del horizonte del trabajo se propone continuar con proyectos de la siguiente naturaleza:

- Implementación de seguridad para el enrolamiento de usuarios en una blockchain federada.
- Implementación de mecanismos de seguridad para la prevención de fuga de información en un sistema de identidad digital basado en blockchain.
- Casos de estudio en la implementación de SIDiB en entidades financieras y de otras naturalezas
- Articulación de comunicación entre SIDiB y otros sistemas basados en Blockchain.

8. Conclusiones

- Al aplicar la tecnología blockchain en el sistema de identidad digital, surgió un conjunto datos emergentes, producto de la interacción del usuario con los diferentes asociados, lo cual agrega valor al modelo de información, pues construye un patrón de comportamiento de la persona dentro del sistema. Estos datos sirven como fuente de información de confianza para los diferentes estudios que hagan las entidades asociadas.

- El sistema de identidad digital propuesto en este documento otorga al titular de la información verdadero control sobre el acceso a sus datos sensibles, dado que son almacenados de manera cifrada y únicamente es él quien puede autorizar la lectura de los mismos. Esto diferencia a la arquitectura propuesta sobre otros medios de autenticación centralizados, pues no existe un custodio de la información diferente al titular.

- Un sistema de identificación digital basado en blockchain, como el propuesto en este documento, le permite a las entidades asociadas lograr una autenticación de usuarios más ligera y ágil, sin tener que acudir a fuentes externas más allá de la blockchain para verificar la legitimidad de una identidad, teniendo así una mayor confianza en el usuario con un menor esfuerzo de autenticación.

- En la investigación se logró desarrollar un prototipo basado en la arquitectura propuesta. Aunque éste cumple con las funcionalidades básicas de la blockchain es un buen punto de partida para la implementación de la propuesta en ambientes productivos. Lo anterior demuestra la viabilidad de un sistema basado en esta tecnología para la construcción de una identidad digital.

Bibliografía

[1] Congreso de la República de Colombia, «Ley 1581 de 2012,» 2012.

[2] Superintendencia Financiera Colombiana - Banca de las Oportunidades, «Reporte de Inclusión Financiera 2017,» Superintendencia Financiera Colombiana, 2018.

[3] C. A. G. M., «Diario El Tiempo,» 19 julio 2018. [En línea]. Available: https://m.eltiempo.com/economia/sector-financiero/crece-numero-de-clientes-digitales-de-la-banca-colombiana-245188. [Último acceso: 2018].

[4] P. V. &. K. Crosby, «BlockChain Technology: Beyond Bitcoin,» *Applied Innovation Review*, vol. 1, pp. 6-19, 2016.

[5] D. Birch, «Blockchain revolution,» de *Dutch National Bitcoin Conference*, Alemania, 2015.

[6] L. a. M. C. Butgereit, «A comparison of two blockchain architectures for inspiring corporate excellence in south africa,» *Information Communication Technology and*, p. 1–6, 2017.

[7] D. Baars, Towards Self-Sovereign Identity using Blockchain Technology, Melbourne: Universidad de Twente, 2018, p. 90.

[8] B. community, «bitcoinwiki,» 2018. [En línea]. Available: https://en.bitcoinwiki.org/wiki/Blockchain. [Último acceso: 12 2018].

[9] Grupo Aval Acciones Y Valores S.A., «M-DE-04. POLITICA DE TRATAMIENTO DE DATOS,» 2015.

[10] Universidad de Standford, «I Reveal My Attributes | IRMA,» Universidad de Standford, 2016.

[11] connective, «Introducing iDIN: a Bank ID for the Netherlands,» connective, 02 02 2016. [En línea]. Available: https://connective.eu/introducing-idin-a-bank-id-for-the-netherlands/. [Último acceso: 10 01 2019].

[12] R. Walker, «OneName: The Bridge Between Physical & DigitalIdentit,» 13 02 2015. [En línea]. Available: https://rywalk.wordpress.com/2015/02/13/onename-the-bridge-between-physical-digital-identity/. [Último acceso: 13 01 2019].

[13] J.-H. Lee, «BIDaaS: Blockchain Based ID As a Service,» *IEEE Access*, vol. 6, pp. 2274 - 2278, 2017.

[14] Blockchain Luxemburg, «blockchain.com,» 2019. [En línea]. Available: https://www.blockchain.com/btc/block/000000000000000000e1826d81b3697ab3c5404f4d2f392b7791aee48bca23a. [Último acceso: 30 01 2019].

Anexo 1 Código fuente del prototipo creado

En este apartado se evidenciará el código fuente de la aplicación desarrollada como prototipo de la investigación

El software se desarrolló en el lenguaje de programación Angular en su versión 4.0, con las especificaciones de máquina indicadas en la imagen:

Imagen 27. Especificaciones de máquina. Fuente: Autor

Se creó una aplicación basada en módulos orquestados, empleando para los estilos CSS el framework Boostrap CSS.

En la interfaz de usuario la aplicación se distribuye en los componentes indicados en la imagen 28. A nivel de código fuente, esta distribución está organizada en los siguientes paquetes:

- Block
- Model

- Register
- Utils
- App.component

Imagen 28. Distribución visual de la aplicación.

Objetos del Paquete *"Model"*

Objeto *"Block"*:

El objeto de información básico de la blockchain. Cuenta con un método para calcular el valor del Hash a partir de las variables que lo componen y un método encargado de minar el bloque.

```
export class Block {
  id: number;
  nonce: number;
  publicKey: string;
  digitalId: string;
  previous: string;
  hash: string;
  timestamp: Date;
  difficulty: number;
  isInvalid: boolean;
  constructor(id: number, nonce: number, publicKey: string, digitalId:
string, previous: string, difficulty: number) {
```

```
    this.id = id;
    this.nonce = nonce;
    this.publicKey = publicKey;
    this.digitalId = digitalId;
    this.previous = previous;
    this.timestamp = new Date();
    this.difficulty = difficulty;
    this.hash = this.calculateHash();
  }
  calculateHash(): string {
    return SHA256(this.id +
      this.nonce +
      this.previous +
      this.timestamp +
      JSON.stringify(this.publicKey) +
      JSON.stringify(this.digitalId)
    ).toString();
  };

  mineBlock() {
    let initialHash = this.hash.valueOf();
    while (!this.isValidValue(initialHash)) {
      this.nonce++;
      initialHash = this.calculateHash();
    }
    this.hash = initialHash;
    console.log('Block mined!: ' + this.hash);
  }

  private isValidValue(hash: string): boolean {
    return hash.substring(0, this.difficulty) === Array(this.difficulty +
1).join('0');
  }
  public isValid(): boolean {
    return this.isValidValue(this.hash);
  }
}
```

Objeto *"Registry"*

Un simple DTO para transferir el bloque creado, así como la llave privada y el Digital Id del usuario.

```
import { Block } from './Block';

export class Registry {
```

```
    block: Block;
    privateKey: string;
    did: string;
    constructor(block: Block, privateKey: string, did: string) {
        this.block = block;
        this.privateKey = privateKey;
        this.did = did;
    }
}
```

Componente *"BlockComponent"*

Componente de Angular encargado de brindar al usuario la interacción con un bloque y notificar los cambios en este.

Lógica del componente:

```
import { Component, OnInit, Input, Output, EventEmitter } from '@angular/core';
import { Block } from '../model/Block';

@Component({
  selector: 'app-block',
  templateUrl: './block.component.html',
  styleUrls: ['./block.component.css'],
})
export class BlockComponent implements OnInit {
  @Input()
  block: Block;
  @Output()
  _blockResult = new EventEmitter<Block>();

  constructor() { }

  ngOnInit() {
    this.block = this.block;
  }

  notify() {
    this.block.hash = this.block.calculateHash();
    this._blockResult.emit(this.block);
  }

  mineBlock() {
    this.block.mineBlock();
    this._blockResult.emit(this.block);
  }
```

```
  getClass() : string {
    let blockClass = 'app-block';
    if(!this.block.isValid())
      blockClass += ' invalid';
    return blockClass;
  }
}
```

Vista del componente:

```
<div [class]=getClass()>
  <form class="form">
    <div class="form-row">
      <div class="multiline input-group-sm">
        <div class="">
          <p>id:</p>
        </div>
        <input type="text" name="id" [(ngModel)]="block.id"
(ngModelChange)="notify()" class="form-control">
      </div>
      <div class="multiline input-group-sm">
        <div class="">
          <p>nonce:</p>
        </div>
        <input type="number" name="nonce" [(ngModel)]="block.nonce"
(ngModelChange)="notify()" class="form-control"
          placeholder="nonce">
      </div>
    </div>

    <div class="form-row">
      <div class="input-group input-group-sm">
        <div class="input-group-prepend">
          <p>public Key:</p>
        </div>
        <textarea name="publicKey" [(ngModel)]="block.publicKey"
(ngModelChange)="notify()" type="text" class="form-control"></textarea>
      </div>
    </div>
    <div class="form-row">
      <div class="input-group input-group-sm">
        <div class="input-group-prepend">
          <p>Digital Id:</p>
        </div>
        <textarea name="digitalId" [(ngModel)]="block.digitalId"
(ngModelChange)="notify()" type="text" class="form-control"></textarea>
      </div>
    </div>
    <div class="form-row">
      <div class="input-group input-group-sm">
        <div class="input-group-prepend">
```

```
          <p>prev hash:</p>
        </div>
        <input name="previous" [(ngModel)]="block.previous" type="text"
(ngModelChange)="notify()" class="form-control" placeholder="previous">
      </div>
    </div>
    <div class="form-row">
      <div class="input-group input-group-sm">
        <div class="input-group-prepend">
          <p>hash:</p>
        </div>
        <input name="hash" [(ngModel)]="block.hash" type="text" class="form-
control" placeholder="hash">
      </div>
    </div>
    <div class="form-row">
      <div class="input-group input-group-sm">
        <div class="input-group-prepend">
          <p></p>
        </div>
        <button (click)="mineBlock()" class="float-right btn btn-sm btn-
primary">Mine Block</button>
      </div>
    </div>
  </form>
</div>
```

Servicio *"RsaService"*

Servicio encargado de las tareas de encripción. Este componente utiliza la librería de javascript *KEYPAIR* para crear los pares de llaves RSA, y la librería *JSENCRYPT* para encriptar y desencriptar información asimétricamente a partir de llaves RSA.

```
import * as keypair from 'keypair';
import * as JSEncryptModule from 'jsencrypt';
export class RsaProvider {
    private _privateKey: string;
    private _publicKey: string;
    private crypt = new JSEncryptModule.JSEncrypt();
    constructor() {
        const pair = keypair(1024);
        this._privateKey = pair.private;
        this._publicKey = pair.public;
        this.crypt.setPrivateKey(this._privateKey);
        this._publicKey = this.crypt.getPublicKey();
    }

    public get publicKey(): string {
```

```
        return this._publicKey;
    }

    public get privateKey(): string {
        return this._privateKey;
    }

    encrypt(plaintext: string): string {
        return this.crypt.encrypt(plaintext);
    }

    decrypt(cypher: string): string {
        return this.crypt.decrypt(cypher);
    }
}
```

Componente *"RegistryComponent"*

Componente de Angular encargado de registrar un bloque en la blockchain. Utiliza el servicio *RSAService*, definido anteriormente, para generar el par de llaves privada y pública del usuario y encriptar los datos sensibles de éste, produciendo así un Digital Id. Con estas variables crea un nuevo objeto de tipo *Block* y lo notifica a la aplicación para ser insertado en la blockchain.

Lógica del componente:

```
import { Component, OnInit, Output, EventEmitter } from '@angular/core';
import { Block } from '../model/Block';
import { RsaProvider } from '../utils/RsaProvider';
import { Registry } from '../model/Registry';
@Component({
  selector: 'app-register',
  templateUrl: './register.component.html',
  styleUrls: ['./register.component.css']
})
export class RegisterComponent implements OnInit {

  private idNumber: number;
  private name: string;
  private isValid = true;
  @Output()
  _newRegistry = new EventEmitter<Registry>();
  constructor() {
    this.idNumber = 111111;
    this.name = "Maria"
  }
  ngOnInit() {
```

```
  }
  create() {
    if (!this.idNumber || this.name === '') {
      this.isValid = false;
    }
    else {
      this.isValid = true;
      const rsa = new RsaProvider();
      let crpt = rsa.encrypt('id: ' + this.idNumber + ', name: ' + this.name);
      console.log('crypt: ', crpt);
      console.log('decrypt: ', rsa.decrypt(crpt));
      this._newRegistry.emit(new Registry(new Block(0, 11111, rsa.publicKey,
crpt, '', 0), rsa.privateKey, crpt));
      // this.name = '';
      // this.idNumber = null;
    }
  }

  getClass(): string {
    if (!this.isValid) {
      return 'container register invalid';
    } else
      return 'container register';
  }
}
```

Vista del componente:

```
<div [class]= getClass()>
  <div class="form-inline">
      <label>Register new user:</label>
    <div class="register-input input-group input-group-sm">
      <label for="id">id: </label>
      <input type="number" name="id" [(ngModel)]="idNumber" class="form-control">
    </div>
    <div class="register-input input-group input-group-sm">
      <label for="name">name: </label>
      <input type="text" name="name" [(ngModel)]="name" class="form-control">
    </div>
    <button class="btn btn-success btn-sm" (click)="create()">register</button>
  </div>
</div>
```

Componente *"AppComponent"*

Componente principal de la aplicación. Éste orquesta los comportamientos de los componentes anteriormente identificados y agrega los módulos de visualización de usuarios y búsqueda de usuarios, para completar la funcionalidad de la aplicación.

Lógica del componente:

```
import { Component, OnInit } from '@angular/core';
import { Block } from './model/Block';
import { Registry } from './model/Registry';
import { RsaProvider } from './utils/RsaProvider';
import * as JSEncryptModule from 'jsencrypt';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent implements OnInit {
  private blocks: Array<Block> = [];
  private registries = new Array<Registry>();
  private difficulty = 2;
  private blocksCounter = 1;
  private _isBlockchainValid = true;
  private qPrivateKey: string;
  private qDid: string;

  constructor() {
  }

  ngOnInit(): void {
  }
  validateBlockchain(event: Block) {
    for (let i = event.id - 1; i <= this.blocks.length - 2; i++) {
      if (i === event.id - 1) { //Evaluates the modified block
        this.blocks[i + 1].previous = event.hash;
        this.blocks[i + 1].hash = this.blocks[i + 1].calculateHash();
      } else { //evaluate next blocks
        this.blocks[i + 1].previous = this.blocks[i].hash;
        this.blocks[i + 1].hash = this.blocks[i + 1].calculateHash();
      }
    }
    this.isBlockchainValid();
  }
  registerBlock(event: Registry) {
    if (this.isBlockchainValid()) {
      event.block.id = this.blocksCounter++;
      if (this.blocks.length > 0)
```

```
        event.block.previous = this.blocks[this.blocks.length - 1].hash;
      else
        event.block.previous = '0';
      event.block.difficulty = this.difficulty;
      event.block.mineBlock();
      this.blocks.push(event.block);
      this.registries.push(event);
    }
  }

  searchValues(id: number) {
    this.qPrivateKey = this.registries[id].privateKey;
    this.qDid = this.registries[id].did;
  }
  find() {
    let targetBlock = this.blocks.filter(b => b.digitalId == this.qDid)[0];
    if (targetBlock !== undefined) {
      console.log(targetBlock.digitalId);
      let crypt = new JSEncryptModule.JSEncrypt();
      crypt.setPrivateKey(this.qPrivateKey);
      let decrypted = crypt.decrypt(targetBlock.digitalId);
      alert(decrypted === false ? "couldn't decrypt user info" : 'user info: [' +
decrypted + ']');
    }else {
      alert('user not found');
    }

  }
  getClass(): String {
    if (this._isBlockchainValid)
      return 'form-inline app-blocks';
    return 'form-inline app-blocks invalid';
  }

  isBlockchainValid(): boolean {
    if (this.blocks.length !== 0 && this.blocks.filter(b => !b.isValid())[0] !==
undefined)
      this._isBlockchainValid = false;
    else
      this._isBlockchainValid = true;
    return this._isBlockchainValid;
  }
}
```

Vista del componente:

```
<div class="container-fluid">
  <div class ="header-section">
    <div class="app-register">
      <app-register (_newRegistry)="registerBlock($event)"></app-register>
    </div>
    <div class=" form-inline app-query">
      <label>Search user: </label>
      <div class="registry-part">
          <input name="q_privateKey" [(ngModel)]="qPrivateKey"
placeholder="Private key"/>
        </div >
        <div class="registry-part">
          <input name="Did" [(ngModel)]="qDid" placeholder="DigitalID"/>
        </div>
        <button class="btn btn-info btn-sm" (click)="find()">find</button>
    </div>
  </div>
  <div [class]= getClass()>
    <div class="app-block" *ngFor="let block of blocks">
      <app-block [block]="block"
(_blockResult)="validateBlockchain($event)"></app-block>
    </div>
  </div>
  <div class="keys-registry">
    <label>Registered users ({{registries.length}})</label>

    <div *ngFor="let key of registries; let i = index">
      <div class="form-inline registry-group">
        <label>User {{i+1}}:</label>
        <div class="registry-part">
          <label class="input-group-addon">Private key:</label>
          <input readonly value={{key.privateKey}} />
        </div>
        <div class="registry-part">
          <label class="input-group-addon">DigitalID:</label>
          <input readonly value={{key.did}} />
        </div>
        <button class="btn btn-sm btn-link"
(click)=searchValues(i)>search</button>
      </div>
    </div>
  </div>
</div>
```