

MÉTODO IMPLEMENTADO EN COMPUTADOR PARA CONFIRMACIÓN DEL ORIGEN, CARACTERÍSTICAS Y ELEMENTOS RELEVANTES DE DOCUMENTOS ELECTRÓNICOS

5 CAMPO TÉCNICO

La presente divulgación está relacionada con sistemas y métodos implementados en computador que permiten la identificación, el seguimiento y la confirmación del origen de documentos electrónicos. También, se relaciona con métodos implementados por computador y sistemas de verificación de características y elementos de documentos digitales mediante la generación de datos de certificación.

DESCRIPCIÓN DEL ESTADO DE LA TÉCNICA

15 En la actualidad, diversos procesos incorporan la transmisión de información y documentos electrónicos, como garantías bancarias, contratos de seguro de cumplimiento y documentos de patrimonios autónomos utilizados como garantía, los cuales son emitidos a través de documentos electrónicos.

20 Estos procesos y las transmisiones asociadas plantean varias preocupaciones. Por ejemplo, es relevante llevar un seguimiento detallado de las acciones realizadas con la información consignada en los documentos, especificando cuándo y por quién se llevaron a cabo. Asimismo, se busca que esta información de seguimiento esté resguardada contra el acceso no autorizado, preservando la integridad de los datos.

25 Por consiguiente, es deseable que los procedimientos y las transferencias vinculadas con documentos electrónicos permitan a beneficiarios e interesados obtener la generación de datos de certificación que faciliten verificar el origen, las características y los elementos relevantes de dichos documentos electrónicos mediante la generación de datos de certificación.

35 El estado de la técnica divulga en el documento US11488269B2 un sistema basado en blockchain para la transformación y responsabilidad de documentos en el sector inmobiliario. El sistema utiliza un modelo de datos para identificar reglas específicas para una transacción y seleccionar documentos para un paquete de documentos. El sistema interactúa con terceros, como instituciones financieras y reguladoras, para verificar la información y los documentos de la transacción. Todo el proceso se registra en la cadena de bloques (blockchain) para garantizar la transparencia y la seguridad en el proceso de gestión de documentos.

Por otro lado, el documento WO2022002526 divulga un método implementado por computadora para vincular criptográficamente múltiples documentos a través de una secuencia de transacciones de cadena de bloques (blockchain). Este método implica calcular datos de firma para documentos que cumplen con requisitos específicos de firma, lo que se realiza en transacciones de blockchain asociadas con los documentos. Estos datos de firma se utilizan para firmar una transacción de vinculación que conecta criptográficamente documentos existentes y complementarios. La firma garantiza la integridad y la relación entre los documentos, y todo el proceso se registra en la cadena de bloques (blockchain) para asegurar transparencia y seguridad en la gestión de documentos.

El documento US11063744B2 divulga un sistema informático y un método para gestionar documentos digitales mediante el uso de blockchain. Este sistema recibe documentos digitales cifrados de un primer sistema informático de cliente, registra la recepción y envío de estos documentos en transacciones de blockchain, y utiliza identificadores únicos generados a partir del contenido de los documentos. Además, el sistema puede realizar consultas en la cadena de bloques para recuperar el historial de transacciones de un documento electrónico, mostrar resultados a usuarios autorizados y gestionar la modificación del contenido del documento.

De acuerdo con lo anterior, los métodos y sistemas anteriormente descritos no permiten abordar de manera integral y eficiente la generación segura de datos de certificación para documentos electrónicos, la identificación precisa y el seguimiento completo de documentos en flujos de trabajo digitales, así como la garantía de la integridad y autenticidad de los documentos mediante enfoques criptográficos sólidos y sigue siendo necesario métodos implementados en computador que permiten la identificación, el seguimiento, la confirmación del origen de documentos electrónicos mediante la generación de datos de certificación y existe, por lo tanto, la necesidad de alternativas. Adicionalmente, se observa un margen de mejora en el ámbito del seguimiento de documentos y flujos de trabajo.

BREVE DESCRIPCIÓN

La presente divulgación corresponde a un método implementado en computador para generación de datos de certificación de un documento electrónico, que comprende ejecutar en una unidad de cómputo las etapas de: a) autenticar un usuario mediante un proceso de verificación de identidad, donde el proceso de verificación de identidad incluye ejecutar al menos una primera autenticación y una segunda autenticación; donde el usuario accede desde una terminal a una aplicación que conecta la terminal con la unidad de cómputo; b) recibir, desde una terminal del usuario, una solicitud de certificación de estado asociada a un archivo, donde el archivo se almacena en una base de datos, y donde el archivo incluye al menos un documento electrónico; c) transmitir, a un servidor validador, una solicitud de verificación

que incluye el archivo, un dato de estampa de tiempo y un dato de identificación asociado al usuario y, d) recibir, desde el servidor validador, un dato de certificación asociado a la solicitud de verificación, donde el servidor validador obtiene el dato de certificación mediante un proceso basado en blockchain

5

En una modalidad del método de la presente divulgación, en el proceso basado en blockchain para obtener el dato de certificación, el servidor validador obtiene el dato de certificación mediante un proceso basado en blockchain que comprende: validar si existe una billetera digital asociada al usuario en una red blockchain; si la validación es negativa entonces se crea una billetera digital asociada al usuario. Dicha billetera digital incluye un dato de clave privada, un dato de clave pública, y un dato de identificación asociado al usuario. Con la red blockchain, asociada a la billetera digital, se genera, un dato hash asociado al archivo del dato de solicitud de verificación, y se obtiene el dato de certificación a partir del dato hash. El dato de certificación incluye el dato hash, un dato de estado del documento electrónico del archivo y un dato de estampa de tiempo asociado al momento en que se genera el dato hash.

10

15

En algunas modalidades del método de la presente divulgación el servidor validador y la red blockchain se conectan mediante una puerta de enlace de API, donde La API puede integrar endpoints que ofrecen servicios de autenticación, firma digital, notificaciones, estampado cronológico en blockchain, y combinaciones de los mencionados, además de realizar la conexión con un servicio de backend.

20

BREVE DESCRIPCIÓN DE LAS FIGURAS

25

La FIG. 1 ilustra un diagrama de flujo del método implementado en computador para generación de datos de certificación (12) de un documento electrónico.

La FIG. 2 ilustra un diagrama de flujo de un proceso basado en blockchain (13) para obtener el dato de certificación (12).

30

La FIG. 3 ilustra una modalidad de arquitectura de una puerta de enlace de API (17).

La FIG. 4A, ilustra la primera parte del diagrama esquemático para un ejemplo del método implementado en computador para generación de datos de certificación (12).

35

La FIG.4B, ilustra la segunda parte del diagrama esquemático para un ejemplo del método implementado en computador para generación de datos de certificación (12).

La FIG. 4C ilustran la tercera parte del diagrama esquemático para un ejemplo del método implementado en computador para generación de datos de certificación (12).

La FIG. 5. ilustra un ejemplo de documento que contiene un dato hash (15) y un código QR que permite la consulta de los datos de datos de certificación (12).

DESCRIPCIÓN DETALLADA

Haciendo referencia a la FIG. 1 la presente divulgación se relaciona con un método implementado en computador para generación de datos de certificación (12) de un documento electrónico, que comprende ejecutar en una unidad de cómputo (1) las etapas de:

- a) autenticar un usuario (2) mediante un proceso de verificación de identidad, donde el proceso de verificación de identidad incluye ejecutar al menos una primera autenticación y una segunda autenticación; donde el usuario (2) accede desde una terminal (3) a una aplicación (4) que conecta la terminal (3) con la unidad de cómputo (1);
- b) recibir, desde una terminal (3) del usuario (2), una solicitud de certificación de estado (5) asociada a un archivo (6), donde el archivo (6) se almacena en una base de datos (7), y donde el archivo (6) incluye al menos un documento electrónico;
- c) transmitir, a un servidor validador (9), una solicitud de verificación (8) que incluye el archivo (6), un dato de estampa de tiempo (10) y un dato de identificación (11) asociado al usuario (2), y
- d) recibir, desde el servidor validador (9), un dato de certificación (12) asociado a la solicitud de verificación (8), donde el servidor validador (9) obtiene el dato de certificación (12) mediante un proceso basado en blockchain (13).

El método implementado en computador para generación de datos de certificación (12) de la presente divulgación ofrece varias ventajas técnicas distintivas: La autenticación del usuario, que puede incluir múltiples capas de verificación de identidad, atribuye un nivel de seguridad en el proceso. La eficiente comunicación entre la terminal del usuario (2) y la unidad de cómputo (1) ayuda a la integridad y protección de los datos transmitidos, mejorando así la seguridad informática entre las conexiones del sistema.

Además, la integración de un proceso basado en blockchain (13) para obtener el dato de certificación (12) brinda una seguridad adicional y una verificación. Esto se traduce en la inmutabilidad de los datos certificados y una trazabilidad del proceso de certificación, lo que eleva la transparencia y la confiabilidad del sistema en su conjunto. Lo anterior hace que el

método sea una solución segura para la generación de datos de certificación de documentos electrónicos.

En algunas modalidades del método aquí divulgado, en la etapa a) es posible que
5 opcionalmente se ejecute una etapa de autenticación con el fin de fortalecer la seguridad del acceso del usuario (2) a la aplicación (4), estableciendo una conexión segura con la unidad de cómputo (1). Así, por ejemplo, el usuario (2) inicia sesión proporcionando sus credenciales de autenticación, lo que desencadena el proceso de autenticación para verificar su identidad.

10 Una vez autenticado, se puede almacenar de manera segura información relacionada con el acceso en una base de datos, como por ejemplo en una base de datos no relacional. Esto sirve en la gestión y trazabilidad de las interacciones de los usuarios (2). Se puede entonces registrar de manera específica la información relevante, asegurando un seguimiento detallado
15 de las actividades de acceso.

Adicionalmente, como parte de la etapa a), en una modalidad del método aquí divulgado se puede asignar un estado a la acción de inicio de sesión, el cual puede ser clasificado como “solicitud”, “aprobación” o “rechazo”. Este estado es determinado por la interacción del
20 usuario (2) durante la etapa de ingreso de credenciales, proporcionando un marco de referencia claro para la gestión de documentos digitales asociados al usuario (2).

Igualmente, en algunas modalidades del método de la presente descripción es posible asignar roles a un usuario (2) y crear categorías como "comercial", "oficial", "agente" y "empresa",
25 se puede utilizar por ejemplo un sistema de gestión de identidades y accesos (IAM). En primer lugar, se crea un usuario (2) en el sistema IAM, proporcionando detalles como nombre y credenciales. Luego, se definen roles o perfiles, como "comercial", "oficial", "agente" y "empresa", cada uno con permisos específicos, determinando así sus responsabilidades y privilegios. Para cada rol, se configuran políticas de acceso que especifican las acciones y
30 recursos permitidos. Si es necesario, se establecen jerarquías de acceso.

Como medida opcional en la etapa a), se puede por ejemplo ejecutar un servicio de seguridad destinado a salvaguardar los datos, cuentas y cargas de trabajo contra accesos no autorizados. Ejemplos de medidas de seguridad adicionales pueden implementadas por ejemplo mediante
35 protocolos como HTTPS (HyperText Transfer Protocol Secure), respaldados por soluciones de seguridad proporcionadas por servicios, como un servicio de seguridad en la nube de Amazon Web Services (AWS). Esta capa de seguridad actúa como un escudo contra posibles amenazas y ayuda a la integridad y confidencialidad de la información sensible gestionada.

El servicio de seguridad en la nube de Amazon Web Services (AWS) implementa una infraestructura para proteger las aplicaciones mediante el uso de HTTPS (HyperText Transfer Protocol Secure). Este proceso se inicia con la utilización de un AWS Firewall, que actúa como una primera línea de defensa, filtrando y monitoreando el tráfico entrante y saliente.

5

La conexión del servicio se puede establecer, por ejemplo, a través de un “Domain”, permitiendo la gestión centralizada de los recursos asociados con la aplicación (4) web. El Domain sirve como un punto de entrada organizado y seguro para el tráfico, facilitando la administración de la seguridad y el monitoreo de la actividad.

10

Para el entendimiento de la presente divulgación "Domain" se refiere a un servicio de Amazon Web Services (AWS) llamado Amazon Route 53. Este es un servicio de Sistema de Nombres de Dominio (DNS) y registro de dominios que permite al usuario (2) gestionar la resolución de nombres de dominio y la configuración de rutas de tráfico en la nube de AWS. Este servicio permite asignar nombres de dominio a los recursos de AWS, como las instancias de servidores web o los puntos de conexión de servicios. Además de gestionar la resolución de nombres, Amazon Route 53 facilita la configuración de registros DNS, como registros de alias, registros de enrutamiento de tráfico y otros, para optimizar y asegurar el flujo de tráfico hacia los recursos relevantes.

20

En este entorno, se puede implementar, por ejemplo, un AWS Web Application Firewall (WAF), el cual proporciona una capa adicional de protección contra amenazas específicas de aplicaciones. Este componente permite definir y aplicar reglas de seguridad personalizadas, identificando y mitigando potenciales ataques, como por ejemplo inyecciones SQL o intentos de cross-site scripting (XSS).

25

Adicionalmente, se puede integrar por ejemplo un servicio de SSL (Secure Sockets Layer) para cifrar la comunicación entre el usuario (2) y la aplicación (4). Esta capa de encriptación asegura la confidencialidad de la información transmitida, previniendo la interceptación no autorizada de datos sensibles. Para el entendimiento de la presente divulgación el servicio de SSL (Secure Sockets Layer) se refiere a un protocolo de seguridad estándar utilizado para establecer conexiones cifradas y seguras en internet. En otro ejemplo de la presente descripción el servicio SSL es el protocolo TLS (Transport Layer Security).

30

En una realización particular la segunda autenticación de la etapa a) del método implementado en computador para generación de datos de certificación (12) de la presente invención incluye las siguientes subetapas:

35

40

a4) enviar una solicitud de autenticación (21) a un servidor de generación de OTP (20) (Por las siglas en inglés de One-Time Password);

5 a5) recibir desde la terminal (3) del usuario (2) un comando OTP (22) que incluye un código de único uso (26) generado por el servidor de generación OTP (20) y que es enviado electrónicamente a un dispositivo electrónico (27) del usuario (2).

10 Es decir que se inicia con la solicitud de autenticación (21) enviada al servidor de generación de OTP (20), "One-Time Password o contraseña de único uso, en español" y el servidor de generación de OTP (20) responde al usuario (2) proporcionando una contraseña de único uso "OTP". Este código de único uso (26) se genera internamente en el servidor de generación de OTP (20) y se incluye en un comando OTP (22), que se envía electrónicamente al dispositivo electrónico (27) del usuario (2).

15 El código de único uso (26) se refiere a un código de seguridad de un solo uso que se genera dinámicamente y se utiliza durante un breve período de tiempo. Su propósito principal es proporcionar una capa adicional de seguridad en el proceso de autenticación, reduciendo la vulnerabilidad asociada con contraseñas estáticas. Cada vez que un usuario (2) solicita autenticación, se genera un nuevo código de único uso (26), lo que significa que incluso si el código es interceptado, no será útil para futuros intentos de acceso.

25 El servidor de generación OTP (20) es un componente en este proceso. Este servidor tiene la capacidad de generar, gestionar y validar código de único uso (26), que también se puede llamar códigos OTP, de manera segura. Cuando un usuario (2) solicita autenticación, el servidor de generación OTP (20) recibe la solicitud y genera un código de único uso (26), que luego se envía electrónicamente al dispositivo electrónico (27) del usuario (2). La generación de código de único uso (26) generalmente se basa en algoritmos criptográficos que garantizan la aleatoriedad y la imprevisibilidad de los códigos, fortaleciendo así la seguridad del proceso.

30 En otra realización particular la segunda autenticación de la etapa a) del método implementado en computador para generación de datos de certificación (12) de la presente invención incluye las siguientes subetapas:

35

a6) enviar una solicitud de autenticación (23) a un servidor validador (9), donde la solicitud de autenticación (23) incluye el dato de identificación (11) del usuario (2);

- a7) recibir desde el servidor validador (9), un dato de respuesta (19) con un valor seleccionador entre “aprobado” o “rechazado”, donde el servidor validador (9) genera el valor “aprobado” si el valor del dato de identificación (11) del usuario (2) coincide con unas credenciales (24) asociadas a una billetera digital (14) previamente creada por el usuario (2); y
- a8) pasar a la etapa b) si el valor del dato de respuesta (19) es “aprobado”.

Es decir que el criterio para determinar el valor de "aprobado" se basa en la comparación del dato de identificación (11) del usuario (2) con las credenciales (24) asociadas a una billetera digital (14) previamente creada por el usuario (2). En caso de que el dato de identificación (11) del usuario (2) (11) coincida con las credenciales asociadas a la billetera digital (14) creada por el usuario (2), el servidor validador (9) emitirá un valor "aprobado", indicando que la autenticación ha sido exitosa y pasar a la etapa b) del método implementado en computador para generación de datos de certificación (12) de la presente descripción. Por el contrario, si no hay coincidencia, se emitirá un valor "rechazado", indicando que la autenticación ha fallado.

La creación de una billetera digital (14) por parte del usuario (2), se lleva a cabo por ejemplo en un proceso que implica la generación y configuración de un entorno seguro para almacenar y gestionar información de manera digital.

En primera instancia, el usuario (2) inicia el proceso mediante la selección de la opción de creación de una billetera digital (14) en la aplicación (4) correspondiente. En este punto, se le solicitan datos de identificación y autenticación para establecer su identidad en el sistema.

Posteriormente, el sistema procede a generar y asignar de manera única una billetera digital (14) al usuario (2). Esta billetera digital (14) actúa como un contenedor seguro para almacenar información relacionada con transacciones financieras, como saldos, historiales y credenciales asociadas.

Durante la creación de la billetera digital (14), se implementan medidas de seguridad, como la encriptación de datos y la autenticación de dos factores, para salvaguardar la integridad y confidencialidad de la información almacenada. Además, se establecen protocolos para garantizar la interoperabilidad y la accesibilidad del usuario (2) a su billetera digital (14) desde distintos dispositivos.

Una vez finalizado el proceso de creación, la billetera digital (14) queda disponible para que el usuario (2) realice transacciones seguras, acceda a su historial financiero y gestione de manera eficiente sus activos digitales. Este enfoque técnico en la creación de billeteras

digitales contribuye a ofrecer una experiencia segura y funcional en el ámbito de las finanzas digitales.

De manera opcional la billetera digital (14) es una billetera tipo custodia donde por ejemplo el usuario (2) confía la custodia de sus claves privadas a un servicio o entidad de confianza, como una plataforma de intercambio o una empresa especializada en custodia de activos digitales. Este enfoque proporciona un nivel adicional de seguridad y gestión de los activos digitales, especialmente en entornos institucionales o empresariales.

El concepto de custodia de claves privadas podría aplicarse a otros usos no financieros. Una billetera tipo custodia podría, como en el caso del método aquí divulgado, realizarse para gestionar claves privadas relacionadas con la autenticación de usuarios en sistemas informáticos, control de acceso a dispositivos, o incluso para la gestión segura de información sensible en entornos no financieros.

En algunas realizaciones y ejemplos del método de la presente descripción, la creación de una billetera digital (14) se logra asociando al usuario (2) en una red blockchain (13). Para esto se establece una conexión entre el usuario (2) y una dirección específica en la cadena de bloques. En este proceso, el usuario (2) es autenticado en la red blockchain (13), y se le asigna una dirección de billetera digital (14) única. Esta dirección sirve como un identificador en la red.

En la etapa b) método aquí divulgado, se recibe desde una terminal (3) del usuario (2), una solicitud de certificación de estado (5) asociada a un archivo (6), donde el archivo (6) se almacena en una base de datos (7), y donde el archivo (6) incluye al menos un documento electrónico. En algunas modalidades del método aquí divulgado, la base de datos (7) puede ser una base de datos no relacional.

Cuando se recibe la solicitud de certificación de estado (5), el sistema identifica y accede al archivo (6) correspondiente en la base de datos (7). Este archivo (6) contiene al menos un documento electrónico que requiere certificación de estado (5). A partir de este punto, se desencadena una de las etapas siguientes en proceso de verificación y preparación para la certificación de estado (5).

Para el entendimiento del método de la presente descripción se entenderá por certificación de estado (5) la confirmación del origen, características y elementos relevantes de documentos electrónicos. Esta certificación de estado (5) se convierte en una declaración oficial que respalda la autenticidad e integridad de un documento electrónico, proporcionando información sobre su origen y otros atributos clave, los cuales pueden, abarcar, pero no se limita al formato del archivo, la fecha de creación, así como la presencia

de firmas digitales u otros atributos distintivos. Igualmente, puede hacer referencia a elementos clave dentro del documento que se consideran relevantes para su propósito. Esta referencia puede comprender información específica, como datos críticos o metadatos asociados con el contenido del documento electrónico. Así, el certificado de estado no solo
5 valida la autenticidad y las características del documento, sino que también destaca información relevante para su propósito específico. La inclusión de estas características en el certificado refuerza la identidad única del documento electrónico, proporcionando una capa adicional de certeza sobre su naturaleza y origen. Permitiendo la trazabilidad y autenticidad de documentos electrónicos en entornos digitales.

10

En la etapa c) del método aquí divulgado se transmite a un servidor validador (9), una solicitud de verificación (8) que incluye el archivo (6), un dato de estampa de tiempo (10) y un dato de identificación (11) asociado al usuario (2).

15 La etapa c) del método de la presente descripción se inicia mediante la transmisión de una solicitud de verificación (8) hacia un servidor validador (9). Esta solicitud está diseñada para asegurar la autenticidad y la integridad del documento electrónico, incorpora varios elementos. En primer lugar, se incluye el archivo (6), el cual se puede encontrar almacenado en la base de datos (7) que puede ser una base de datos no relacional.

20

Adicionalmente, la solicitud de verificación (8) incorpora un dato de estampa de tiempo (10). El cual, por ejemplo, en algunas realizaciones del método de la presente divulgación puede ser obtenido mediante una red blockchain (13) como blockchain Algorand, que proporciona un registro inmutable de la transacción en la cadena de bloques. Esta elección de tecnología
25 blockchain puede reforzar la seguridad y la confiabilidad del proceso al garantizar la inalterabilidad del dato de estampa de tiempo (10) asociada al documento electrónico.

Asimismo, la solicitud de verificación (8) incluye un dato de identificación (11) asociado al usuario (2), contribuyendo a la trazabilidad del proceso. El dato de identificación (11) sirve
30 como un elemento para verificar la autoría y/o la legitimidad de la solicitud.

Para el entendimiento de la presente descripción, blockchain Algorand es cualquier implementación de cadena de bloques basada en el protocolo Algorand Consensus Algorithm.

35

En la etapa d) del método aquí divulgado se recibe desde el servidor validador (9), un dato de certificación (12) asociado a la solicitud de verificación (8), donde el servidor validador (9) obtiene el dato de certificación (12) mediante un proceso basado en blockchain (13).

Es decir que el servidor validador (9) transmite de vuelta un dato de certificación (12) vinculado a la solicitud de verificación (8). Este dato de certificación (12) es adquirido por el servidor validador (9) a través de un proceso basado en tecnología blockchain (13).

5 Haciendo referencia a la FIG. 2 en una realización del método de la presente divulgación el proceso basado en blockchain (13) para obtener el dato de certificación (12) comprende las subetapas de:

- 10 i. validar si existe una billetera digital (14) asociada al usuario (2) en una red blockchain (13);
- ii. si la validación de la subetapa anterior i es negativa entonces crear una billetera digital (14) asociada al usuario (2); donde la billetera digital (14) incluye un dato de clave privada, un dato de clave pública, y un dato de identificación (11) asociado al usuario (2);
- 15 iii. generar, con una red blockchain (13) asociada a la billetera digital (14), un dato hash (15) asociado al archivo (6) del dato de solicitud de verificación (8), y
- iv. obtener el dato de certificación (12) a partir del dato hash (15), donde el dato de certificación (12) incluye el dato hash (15), un dato de estado del documento electrónico del archivo (6) y un dato de estampa de tiempo (10), asociado al momento en que se genera el dato hash (15), que se obtiene mediante el estampado cronológico blockchain.

25 En estas subetapas se implementa un proceso técnico que involucra la generación de un dato hash (15) asociado a un archivo (6) que incluye al menos un documento electrónico. Este dato hash (15) actúa como una representación única y criptográficamente segura del contenido del archivo (6), proporcionando un identificador único para su posterior verificación de integridad.

30 Así, por ejemplo, el archivo (6) se puede someter a una función de dato hash (15) dentro de una red blockchain (13) que incluye al menos un documento electrónico o un dato de solicitud de verificación (8). Esta función del dato hash (15) es tomar el contenido del archivo (6) como entrada y genera un valor hash único, que es una cadena alfanumérica de longitud fija.

35 Este proceso es determinista, lo que significa que el mismo contenido siempre generará el mismo dato hash (15).

El dato hash (15) resultante se asocia directamente al archivo (6) y se almacena en la blockchain (13) como parte de la transacción o registro correspondiente. Cualquier modificación en el archivo (6), por mínima que sea, generará un dato hash (15)

completamente diferente, lo que permite detectar cualquier cambio en el contenido original del documento electrónico asociado al archivo (6).

Cuando se realiza una solicitud de verificación (8), el archivo (6) en cuestión se somete nuevamente a la función de hash, generando un nuevo dato hash (15). Este nuevo dato hash (15) se compara con el valor del dato hash (15) almacenado previamente en la blockchain (13). Si ambos valores coinciden, se verifica la integridad del archivo (6), indicando que el contenido no ha sido alterado desde su registro inicial.

Así las cosas, la función del dato hash (15) en la red blockchain (13) proporciona un método para verificar la integridad del archivo (6) y datos asociados a la billetera digital (14), permitiendo un seguimiento fiable de la validez de la información almacenada en la cadena de bloques.

En la subetapa iv) para obtener el dato de certificación (12), el proceso técnico se extiende a la obtención de un dato de certificación (12) a partir del dato hash (15) asociado a un archivo (6) que incluye al menos un documento electrónico. Este dato de certificación (12) engloba tres componentes fundamentales. En primer lugar, incorpora el propio dato hash (15), que sirve como identificador único y criptográfico del contenido del archivo (6). En segundo lugar, el dato de certificación (12) incluye un dato de estado del documento electrónico incluido en el archivo (6). Este componente proporciona información sobre el estado actual del documento electrónico, indicando si ha sido modificado o permanece inalterado desde su última certificación. Este aspecto permite mantener un registro confiable de la evolución del documento electrónico a lo largo del tiempo.

Por último, el tercer componente del dato de certificación (12) es un dato de estampa de tiempo (10). Este dato está asociado al momento específico en que se genera el dato hash (15) original del archivo (6). La estampa de tiempo (10) se obtiene mediante un estampado cronológico blockchain, lo que garantiza que la certificación se asocie a un instante específico e inmutable en la línea temporal de la cadena de bloques.

En una modalidad del método implementado en computador para generación de datos de certificación (12) de la presente divulgación, el dato de estampa de tiempo (10) se obtiene mediante las subetapas de:

- l. preparar los datos a estampar en la blockchain (13) proporcionado estructura y formato adecuados de la información que se va a estampar;
- m. generar una transacción con los datos de la subetapa anterior;
- n. introducir la información específica a inmutabilizar;

- o. calcular el dato hash (15) de la información incluida en la transacción para garantizar su integridad;
- p. firmar la transacción con una clave privada para autenticar y asegurar la propiedad de la información;
- 5 q. transmitir la transacción a la red blockchain (13) para su procesamiento y validación;

Opcionalmente puede existir etapas adicionales de confirmar transacción por la red blockchain (13) y etapas adicionales para verificar el estampado consultando la cadena de bloques para confirmar la marca de tiempo y la integridad de los datos.

En la subetapa l, se procede a preparar los datos que serán estampados en la red blockchain (13). Esto implica estructurar y formatear la información de manera adecuada, asegurando que cumpla con los requisitos para su inclusión en la cadena de bloques. Los requisitos son por ejemplo identificar y completar campos obligatorios, verificar que los tipos de datos sean apropiados y respetar las longitudes máximas permitidas, validar la información, incluida la verificación de la existencia de entidades asociadas y la conformidad con estándares. Además, se deben aplicar codificaciones y encriptaciones según sea necesario para preservar la seguridad de los datos durante su tránsito en la red blockchain (13). En algunos ejemplos y realizaciones del método de la presente divulgación, los documentos electrónicos y los archivos (6) también puede ser firmados digitalmente por el usuario (2), por ejemplo, mediante el empleo de una billetera digital (14) tipo custodia, asegurando así su autenticidad y preparación para la siguiente etapa del proceso.

En la subetapa b, se genera una nueva transacción que contendrá los datos preparados en la etapa anterior. Esta transacción se convierte en el vehículo a través del cual la información será inmutable en la red blockchain (13), es decir que la información pasa a ser información inmutable. La transacción lleva consigo la carga útil de datos estructurados de manera específica para su inclusión en la cadena de bloques. Esta transacción se configura de acuerdo con los requisitos de la red blockchain (13), por ejemplo, una blockchain Algorand, asegurando la coherencia con la estructura de datos aceptada. Una vez creada, esta transacción se encuentra lista para ser procesada por la red blockchain (13), que verificará y validará la información antes de incorporarla de manera inmutable en el registro distribuido.

En la subetapa c, se introduce la información específica que se busca inmutabilizar en la transacción recién generada, por ejemplo, origen, características y elementos relevantes de un documento electrónico.

Posteriormente, en la subetapa d, se calcula el dato hash (15) de la información incluida en la transacción. En algunas realizaciones del método de la presente divulgación el dato hash

(15) resultante puede actuar como una firma digital única de la información contenida en la transacción.

La subetapa e, implica la firma de la transacción, utilizando por ejemplo una billetera digital (14) tipo custodia. Esta firma no solo autentica la transacción, sino que también asegura la propiedad del documento electrónico y el origen.

Finalmente, en la subetapa f, la transacción firmada se transmite a la red blockchain (13) para su procesamiento y validación. La red blockchain (13), por ejemplo, una blockchain Algorand, se encargará de verificar la autenticidad de la transacción, incorporarla en el registro distribuido y generar la estampa de tiempo (10) asociada al dato de certificación (12) del documento electrónico.

En una modalidad del método implementado en computador para generación de datos de certificación (12) de la presente divulgación la unidad de cómputo (1), el servidor validador (9) y la red blockchain (13) se conectan mediante una puerta de enlace de API (17), y donde, en la etapa a), la primera autenticación comprende las subetapas:

- a1) transmitir, mediante la puerta de enlace de API (17), una solicitud de autenticación (18) que incluye el dato de identificación (11) del usuario (2) y un dato de credenciales de acceso asociadas al usuario (2), y
- a2) enrutar la solicitud de autenticación (18), mediante la puerta de enlace de API (17), hacia un servicio de backend configurado para validar el dato de credenciales de acceso asociadas al usuario (2) de acuerdo con un protocolo de seguridad predeterminado;
- a3) recibir, a través de la puerta de enlace de API (17), un dato de respuesta (19) con un valor seleccionador entre “aprobado” o “rechazado” generado por el servicio de backend; donde la puerta de enlace de API (17) está configurada para limitar el acceso de la unidad de cómputo (1) a la billetera digital (14) en la red blockchain (13), de acuerdo con el protocolo de seguridad predeterminado.

Para el entendimiento de la presente divulgación API significa "Interfaz de Programación de Aplicaciones" (Application Programming Interface, en inglés). La API emplea un conjunto de reglas y protocolos que permite la interacción y la comunicación entre diferentes aplicaciones y servicios de software. Sirve como un puente que facilita que la aplicación (4) se comunique y comparta datos con otros servicios.

La puerta de enlace de API (17) actúa como un intermediario para transmitir y enrutar la solicitud de autenticación (18) desde la unidad de cómputo (1) hacia el servicio de backend encargado de validar las credenciales de acceso asociadas al usuario (2). La puerta de enlace de API (17) proporciona un conjunto de reglas y endpoint que especifican cómo se deben estructurar y enviar las solicitudes de autenticación (18), así como el formato de las respuestas que se pueden esperar por ejemplo el valor seleccionado entre “aprobado” o “rechazado”.

Para el entendimiento de la presente divulgación un "endpoint" se refiere a un punto final o una URL específica a la que se puede acceder para realizar operaciones o intercambiar información. Los endpoints son puntos de acceso a un servicio o una API (Interfaz de Programación de Aplicaciones) y son usados en arquitecturas orientadas a servicios para facilitar la comunicación entre diferentes sistemas y aplicaciones, permitiendo el intercambio de datos de manera segura y estructurada.

Cada endpoint está asociado con una funcionalidad específica o una operación que se puede realizar a través de la puerta de enlace de API (17). Por ejemplo, en un servicio de autenticación, puede haber un endpoint para iniciar sesión, otro para registrar un nuevo usuario (2) y otro para cerrar sesión. Cada uno de estos endpoints representa una acción específica que el servicio de autenticación puede realizar.

En un ejemplo particular de la presente divulgación se tienen los siguientes grupos de endpoints: endpoints de usuario, endpoints de compañías, endpoints para gestionar documentos electrónicos de garantías, endpoints para gestionar documentos digitales, endpoints para gestionar billeteras digitales, endpoints para proporcionar información analítica, endpoints para autenticación, y endpoints para redes sociales. A continuación, se presenta una descripción de estos grupos de endpoints:

Los Endpoints de usuario, llamados para efectos de la presente divulgación, se agrupan con el nombre usersRouter y proporcionan una serie de funciones para gestionar usuarios en una aplicación. Estas funciones incluyen la generación y verificación de códigos OTP (One Time Password) para autenticación, la obtención de listas de usuarios basadas en diferentes criterios como el rol o la pertenencia a ciertos grupos, así como la obtención del perfil del usuario autenticado actualmente. Además, permiten la creación, edición y eliminación de usuarios, así como la gestión de sesiones de usuario, como iniciar sesión y refrescar tokens de autenticación. También ofrece funcionalidades para habilitar o deshabilitar usuarios, editar roles de usuario, y gestionar notificaciones y firmas para usuarios específicos. Todas estas funciones están respaldadas por una base de datos global, por ejemplo, Mongoose.

Los endpoints de compañías, llamados para efectos de la presente divulgación, se agrupan con el nombre `companiesRouter`. Estos endpoints permiten realizar diversas acciones relacionadas con la administración de compañías. Por ejemplo, permiten obtener una lista de compañías según su tipo, así como añadir documentos electrónicos a compañías específicas o aceptar solicitudes de cuotas tanto individuales como globales. Además, permite la creación de nuevas compañías y la obtención de información detallada sobre compañías existentes mediante el uso de identificadores únicos. Para mejorar la interacción con las compañías, también proporciona funciones para buscar compañías según roles específicos y IDs asociados. De igual manera ofrece funcionalidades para añadir usuarios a compañías, habilitar o deshabilitar compañías según sea necesario, y editar la información de compañías existentes. Además de estas acciones básicas, este grupo endpoints permite realizar operaciones específicas como la aprobación de desembolsos para compañías particulares.

Los Endpoints para gestionar documentos electrónicos de garantías, llamados para efectos de la presente divulgación, se agrupan con el nombre `guaranteesRouter`. Bajo este nombre se ofrece una variedad de endpoints para gestionar diferentes aspectos relacionados con las garantías en un sistema. Estos endpoints permiten realizar operaciones como la obtención de información sobre garantías específicas, la creación de nuevas garantías y solicitudes de garantías, así como la gestión de firmas y actualizaciones relacionadas con las garantías. Además de estas acciones, también proporciona funcionalidades para obtener informes sobre las garantías, así como para realizar operaciones automáticas como la firma automática de garantías. Entre las funciones asociadas con estos endpoints se incluyen la creación y actualización de solicitudes y tokens de documentos electrónicos, la recuperación de garantías expiradas y activas, así como la gestión de acciones específicas realizadas por operadores en relación con las garantías.

Todas estas interacciones están diseñadas para proporcionar una gestión eficiente y completa de las garantías dentro del sistema, permitiendo a los usuarios realizar las acciones necesarias para administrar y supervisar el estado y la operación de las garantías de manera efectiva.

Los endpoints para gestionar documentos electrónicos, llamados para efectos de la presente divulgación, se agrupan con el nombre `documentsRouter`. Este grupo proporciona endpoints específicos para la gestión de documentos electrónicos asociados a compañías. Estos endpoints permiten editar y eliminar documentos electrónicos. El endpoint de edición de documentos permite realizar cambios en un documento electrónico existente de una compañía. La lógica de este proceso está implementada en el controlador correspondiente, donde se manejan las operaciones de edición de documentos electrónicos. Por otro lado, el endpoint de eliminación de documentos electrónicos permite eliminar un documento específico de una compañía. Para ello, se utiliza el ID del documento electrónico como parámetro en la URL. La lógica de eliminación está gestionada en el controlador respectivo,

donde se llevan a cabo las operaciones necesarias para eliminar el documento electrónico de manera segura.

Los endpoints para gestionar billeteras digitales, llamados para efectos de la presente divulgación con el nombre walletsRouter, ofrecen una serie de puntos para gestionar las billeteras digitales de los usuarios en el sistema. Estos endpoints permiten realizar diversas operaciones relacionadas con la creación, recuperación, habilitación y deshabilitación de billeteras digitales, así como la actualización de contraseñas asociadas. Hay endpoints para agregar nuevas billeteras para usuarios específicos, donde el usuario se identifica mediante su ID. Además, existen puntos finales para recuperar todas las billeteras disponibles tanto de manera general como para un usuario específico. Estas interacciones están diseñadas para facilitar la gestión eficiente de las billeteras digitales de los usuarios, proporcionando acceso a las funcionalidades necesarias para administrarlas y garantizar su uso seguro.

Los endpoints para proporcionar información analítica, llamados para efectos de la presente divulgación con el nombre de analyticsRouter, son una serie de endpoints diseñados para proporcionar información analítica sobre el rendimiento y las métricas clave de diferentes entidades dentro del sistema. Estos puntos finales permiten obtener una visión general de las analíticas, así como métricas específicas para agentes y operadores. Estos endpoints proporcionan indicadores clave de rendimiento (KPIs) parametrizados, lo que permite a los usuarios obtener métricas según sus necesidades y proporcionar a los usuarios una visión del rendimiento y las métricas relevantes, lo que les permite tomar decisiones informadas.

Las métricas clave de rendimiento de analyticsRouter pueden incluir la tasa de uso, que indica la frecuencia de solicitudes realizadas; el tiempo de respuesta que mide la eficiencia en la entrega de resultados; los errores que señalan problemas en el procesamiento de solicitudes; la satisfacción del usuario que refleja la opinión de los usuarios sobre la calidad; y el uso de recursos que evalúa la eficacia en el uso de recursos. Estas métricas ayudan a evaluar la eficiencia y la calidad del servicio proporcionado.

Los endpoints para autenticación, llamados para efectos de la presente divulgación con el nombre de indexRouter ofrecen varios puntos finales para gestionar el registro y el acceso de usuarios en la aplicación. Esto incluye la posibilidad de registrarse en la aplicación, así como opciones para iniciar sesión con credenciales estándar o mediante una contraseña de un solo uso (OTP). Además, se proporciona funcionalidad para obtener perfiles de usuario y generar códigos OTP específicos.

Para la gestión de contraseñas, se puede permitir, por ejemplo, la recuperación y el restablecimiento de contraseñas. También se pueden incluir opciones para habilitar, deshabilitar y eliminar usuarios específicos.

Los endpoints para redes sociales, llamados para efectos de la presente divulgación con el nombre de socialRouter, proporcionan funcionalidades para que los usuarios inicien sesión utilizando diversas plataformas de redes sociales. Esto incluye opciones para iniciar sesión con Google, Facebook y RUS. De esta forma, en un ejemplo, se puede simplificar el proceso de inicio de sesión para los usuarios, permitiéndoles utilizar sus credenciales de cuentas en las plataformas de redes sociales mencionadas. Al hacer solicitudes a estos puntos finales, se debe especificar la versión de la red social correspondiente según sea necesario.

Para el entendimiento de la presente divulgación "servicio de backend" se refiere al procesamiento y gestión de datos de manera interna, detrás de la interfaz de usuario visible. Los servicios de backend constituyen la lógica de negocio, la gestión de bases de datos, y otras operaciones como por ejemplo una operación que va desde gestión de usuario hasta finalización del documento: incluyendo Autenticación de usuarios; Firma de documentos; Aprobación de documentos; Envío de documentos; Administración de documentos electrónicos, permitiendo el funcionamiento de una realización del método de la presente divulgación.

El servicio de backend es responsable, entre otras funciones y actividades, de validar las credenciales de acceso asociadas a un usuario (2) durante el proceso de autenticación. Este servicio realiza tareas como la verificación de la identidad del usuario (2), la consulta de la base de datos para comparar las credenciales proporcionadas con las almacenadas, y la generación de respuestas que indican el resultado de la autenticación de acuerdo con un protocolo de seguridad predeterminado.

En algunos ejemplos y realizaciones del método de la presente divulgación los servicios de backend se encuentran virtualizados en contenedores, es decir, se utiliza la tecnología de contenedores para ejecutar y gestionar dichos servicios. Los contenedores ofrecen ventajas, como portabilidad, ya que pueden ejecutarse en cualquier entorno compatible con el sistema de contenedores utilizado (por ejemplo, Docker). También facilitan la escalabilidad, ya que los contenedores son rápidos de iniciar y detener, permitiendo ajustar la cantidad de instancias de servicios según sea necesario. Además, ofrecen aislamiento, lo que significa que las aplicaciones pueden ejecutarse de manera segura sin interferencias entre sí.

La eficiencia es otra ventaja, ya que los contenedores comparten recursos del sistema operativo host, lo que resulta en un uso más eficiente de los recursos en comparación con las máquinas virtuales. En resumen, la virtualización en contenedores es una estrategia eficaz para implementar y gestionar servicios de backend, proporcionando eficiencia, flexibilidad y aislamiento.

En una modalidad del método implementado en computador para la generación de datos de certificación (12) de la presente divulgación en la subetapa a2) el servicio de backend es configurado para autenticación y se realiza mediante las subetapas de:

- 5 u. iniciar sesión por parte del usuario (2) en una aplicación (4) solicitando la autenticación (18) del usuario (2);
- v. redirigir la solicitud de autenticación (18) de usuario (2) al servidor donde se aloja el servicio de backend de autenticación (18);
- 10 w. aplicar un protocolo de autorización de las credenciales de acceso del usuario (2), si la autenticación (18) es exitosa continuar a la etapa siguiente de lo contrario no se genera autorización;
- x. generar un código de autorización mediante el servicio de backend alojado en un servidor;
- 15 y. intercambiar el código de autorización por un token de acceso:
- z. utilizar el token de acceso para acceder a recursos protegidos en nombre del usuario (2).

En la subetapa d, el usuario (2), tras ser autenticado correctamente, solicita al servicio de backend alojado en un servidor que genere un código de autorización. Este código de autorización es una credencial temporal que representa el consentimiento del usuario (2) para acceder a recursos protegidos.

Luego, en la subetapa e, el usuario (2) intercambia ese código de autorización por un token de acceso. Este intercambio se realiza mediante una comunicación segura con el servicio de backend. El token de acceso es un elemento de seguridad que otorga al usuario (2) permisos específicos y proporciona una ventana de tiempo para acceder a recursos protegidos en el sistema.

Finalmente, en la subetapa f, el usuario (2) utiliza el token de acceso para acceder a los recursos protegidos en nombre propio. El token actúa como una credencial de autorización al ser presentado a los recursos protegidos, permitiendo al usuario (2) realizar acciones específicas o acceder a información restringida según los permisos asociados al token.

Este flujo de autorización y autenticación puede ser implementado en protocolos como por ejemplo OAuth, OpenID Connect (OIDC), SAML (Security Assertion Markup Language), JWT (JSON Web Token), Kerberos, CAS (Central Authentication Service), LDAP (Lightweight Directory Access Protocol), combinación de los anteriores protocolos u otros protocolos conocidos por una persona medianamente versada en la materia.

En una modalidad del método implementado en computador para generación de datos de certificación (12) de la presente divulgación, en la subetapa a2) el servicio de backend configurado para validar el dato de credenciales de acceso asociadas al usuario (2) se hace una firma digital mediante las subetapas de:

- r. solicitar una billetera digital (14) para firmar transacciones, mediante un servicio Signer;
- s. crear una billetera digital (14) mediante un servicio Signer;
- t. almacenar la billetera digital (14).

Es decir que, en esta modalidad del método de la presente divulgación, inicialmente se puede solicitar una billetera digital (14), como, por ejemplo, una billetera tipo custodia, que permita la firma de transacciones mediante un servicio Signer. El servicio de Signer es una herramienta o componente para gestionar operaciones de firma digital. Su función principal es manejar las claves privadas asociadas a la billetera digital (14), que es, por ejemplo, una billetera tipo custodia, garantizando la seguridad de las transacciones. Además de la generación de firmas digitales y la verificación de transacciones, el servicio Signer puede administrar claves privadas de manera cifrada, asegurando su confidencialidad y protección contra accesos no autorizados.

Cuando un usuario (2) solicita una billetera digital (14) para firmar transacciones, el servicio Signer procesa esta solicitud, verificando la identidad del usuario (2) y validando la solicitud según las reglas de negocio establecidas. Posteriormente, genera una billetera digital (14) única para el usuario (2), lo cual incluye la generación de claves públicas y privadas utilizando algoritmos criptográficos las cuales son almacenadas en esta de manera segura para protegerlas de accesos no autorizados.

Para terminar este proceso, una vez obtenida o creada la billetera digital (14), se puede almacenar en una base de datos, asegurando su acceso y disponibilidad cuando sea necesario realizar firmas digitales.

La firma digital proporciona integridad, autenticidad y no repudio del documento en un ejemplo del método de la presente divulgación la firma digital comprende las siguientes características:

- 1 Autenticidad: identifica el autor de un documento electrónico sin lugar a duda. Para ello, en un ejemplo del método de la presente divulgación, la firma digital tiene los siguientes elementos de autenticidad:

- a. Address de firmante: El formato del Address firmante que es un identificador hexadecimal, seguido de 58 dígitos hexadecimales.
Un ejemplo de Address es:
FODSTL3L7AN36I546H6LXSKSS3RA2OJDBAJMVT6ND67MPQ4TJV
SHPQPCI.
- b. Mensaje firmado: Será el dato hash (15) resultante del documento electrónico, que puede ser por ejemplo de una garantía emitida por una entidad. El algoritmo para generar el dato hash (15) del documento será SHA3-512.
- c. Versión: Versión del método de firma digital empleado mediante el consenso de las partes.
- d. Firma digital resultante: Valor alfanumérico hexadecimal resultante del método empleado para firma digital. Un ejemplo de firma digital es:
KUCGVQYAGZVUYBHMMUIEVV2FAIJL77PXBOWNEEPBFJ6W5CG
JHLMQ
- 2 Integridad: indica que el mensaje no ha sido adulterado o modificado una vez comunicado. Para ello, por ejemplo, el mensaje a firmar digitalmente será siempre el dato hash (15) SHA3-512 del documento electrónico que puede ser por ejemplo una garantía de una entidad. En los eventos en que el documento electrónico sea modificado, el dato hash (15) resultante del documento electrónico modificado será siempre diferente al dato hash (15) del documento electrónico original, y el dato hash (15) del documento electrónico modificado a su vez, no corresponderá a la firma digital resultante, ni al Address firmante, pues este Address nunca emitió la firma digital con dicho documento electrónico.
- 3 No repudio: La firma digital cuyo uso es reconocido en el documento electrónico garantiza que las partes, por ejemplo, un acuerdo, contrato o contrato inteligente no podrán rechazar el documento electrónico con firmas digitales emitidas de conformidad con lo estipulado en el documento electrónico. Por lo anterior, las partes reconocen como auténtica e íntegra la firma digital validada por una red blockchain (13).
- 4 Seguridad complementaria: De forma complementaria la firma digital se garantiza la seguridad asociada al documento electrónico con firma digital.

En una realización del método implementado en computador para generación de datos de certificación (12) de la presente divulgación, posterior a la etapa a3), se realiza una subetapa a10) de conectar la unidad de cómputo (1), través de un DashBoard en una aplicación (4),

con la puerta de enlace de API (17) de conexión, que se encarga de gestionar la autenticación mediante un protocolo de autorización y permite el acceso a la billetera digital (14).

Para el entendimiento de la presente divulgación, la puerta de enlace de API (17) de conexión hace parte del conjunto de reglas y endpoints que permiten la comunicación entre la unidad de cómputo (1) y el servicio de backend. Esta puerta de enlace API (17) facilita la conexión, permitiendo gestionar la autenticación mediante un protocolo de autorización y el acceso a la billetera digital (14).

En un ejemplo del método de la presente divulgación, la creación de la billetera digital (14) se realiza mediante la puerta de enlace de API (17) que ejecuta los siguientes pasos:

A. Solicitar mediante un correo electrónico la creación de la billetera digital (14): en este primer paso, se solicita al usuario (2) que proporcione su dirección de correo electrónico. Luego, se utiliza el servicio de conexión a una red blockchain (13), como por ejemplo el servicio blockchain Algorand de "Algorand Connector", y en conjunto con una máquina de estados (State Machine) se crea billetera digital (14) en la red blockchain (13), por ejemplo, en una red blockchain (13) Algorand.

B. Crear un correo electrónico al generar un usuario (2): Cuando se realiza la creación de un usuario (2), se generará automáticamente una dirección de correo electrónico asociada a ese usuario (2) o se toma la dirección de correo electrónico proporcionada en el paso A y se vincula con las credenciales del usuario (2).

C. Solicitar la creación de un contrato de generación de usuario (2): La plataforma utiliza la puerta de enlace de API (17) y un servicio de autenticación para solicitar un contrato, por ejemplo, un contrato inteligente, que genere un usuario (2) en servidor validador (9). El contrato inteligente se encarga de crear un registro de usuario (2) en el servidor validador (9) con la información relevante del usuario (2).

D. Almacenar la asociación de usuario (2) y contraseña en una base de datos: La información de usuario (2), que incluye la dirección de correo electrónico y credenciales de la billetera digital (14) se almacena de manera segura en una base de datos.

El servicio de conexión a una red blockchain (13) es una herramienta que facilita la conexión y la interoperabilidad entre la puerta de enlace de API (17) y red blockchain (13). En un ejemplo particular, el servicio blockchain Algorand de "Algorand Connector" es una herramienta o interfaz que ayuda al usuario (2) a conectarse con la red blockchain Algorand. Esto podría incluir, entre otras funciones, la gestión de transacciones, la

interacción con contratos inteligentes en la red blockchain Algorand, o la obtención de información específica de la blockchain.

Para el entendimiento de la presente divulgación la máquina de estados (State Machine), es una descripción de cómo se comporta un sistema en distintos estados y cómo se mueve entre ellos, lo cual resulta beneficiosa por diversas razones. En primer lugar, la billetera digital (14) atraviesa distintos estados durante su operación, como por ejemplo "sin conexión", "esperando confirmación" o "transacción completada" entre otros. La máquina de estados organiza y gestiona estos estados de manera estructurada.

Además, la máquina de estados proporciona un control claro sobre las transiciones entre los diferentes estados de la billetera digital (14), asegurando que siga un flujo lógico y seguro en respuesta a las acciones del usuario (2) u otros eventos. Cada evento, como por ejemplo una solicitud de transacción o una actualización, puede desencadenar una transición de estado, y la máquina de estados facilita el manejo ordenado de estos eventos.

La máquina de estados también contribuye a la seguridad y validación de la billetera digital (14). Al tener estados claramente definidos se pueden implementar validaciones y controles de seguridad específicos para cada estado. Por ejemplo, ciertas validaciones pueden aplicarse solo cuando la billetera digital (14) está en el estado de realizar una transacción.

Opcionalmente la máquina de estados puede ofrecer una representación visual y lógica del comportamiento de la billetera digital (14), facilitando la comprensión del código y el desarrollo de nuevas funcionalidades o la resolución de problemas. En términos de mantenimiento y escalabilidad, la máquina de estados puede simplificar la incorporación ordenada de nuevas funciones o actualizaciones, sin perturbar la estructura general del sistema.

Para el entendimiento de la presente divulgación, un contrato inteligente también conocido como "smart contract" en inglés, es una secuencia de pasos ejecutados en un programa informático, es decir que se trata de un conjunto de reglas y condiciones programadas en código que se ejecutan automáticamente cuando se cumplen ciertas condiciones predefinidas. Dichas reglas y condiciones se ejecutan en una red blockchain (13) y tiene la capacidad de automatizar, verificar o hacer cumplir la ejecución de acuerdos o contratos de manera autónoma, es decir que una de las características del contrato inteligente es su capacidad de autoejecución cuando se cumplen las condiciones programadas en el contrato, este se ejecuta automáticamente sin intervención humana, eliminando la necesidad de intermediarios. Otras características pueden ser que, por ejemplo, los contratos inteligentes son inmutables, ya que, una vez desplegados en la blockchain, no pueden ser modificados ni eliminados. Esta característica proporciona seguridad y confianza en la integridad del contrato. La

transparencia es otro aspecto destacado, ya que todos los detalles del contrato, incluidas las condiciones y ejecuciones, son visibles en la red blockchain (13), lo que mejora la visibilidad y la confianza en el proceso.

5 La creación y gestión de la billetera digital (14) mediante la puerta de enlace de API (17) proporciona una mayor seguridad en la billetera digital (14). En este sentido, la puerta de enlace de API (17) puede funcionar como un punto de entrada único, centralizando el acceso a los servicios. Este enfoque reduce la exposición al compartir únicamente la puerta de enlace de API (17) públicamente, mejorando la seguridad de acceso y manteniendo ocultos los
10 detalles internos de implementación.

La puerta de enlace de API (17) puede tener la capacidad de configurar políticas específicas que aseguran que únicamente usuarios (2) autenticados y autorizados puedan acceder a los servicios. Esta medida refuerza la seguridad de acceso y proporciona un mejor control sobre
15 quiénes pueden interactuar con la billetera digital (14).

Luego, la puerta de enlace de API (17) cumple también la función de enrutamiento de la solicitud al servicio correspondiente. Este paso asegura que cada solicitud sea dirigida al servicio específico necesario, evitando accesos no autorizados y mejorando la custodia de
20 datos al canalizarlos hacia la lógica adecuada.

Por otro lado, el servicio backend puede, por ejemplo, realizar el procesamiento seguro de la solicitud, aplicando medidas de seguridad y cumpliendo con las políticas de autorización establecidas. Lo que ayuda a garantizar que los datos sean gestionados de manera adecuada,
25 contribuyendo a mejorar la custodia de la información de la billetera digital (14)

Finalmente, la puerta de enlace de API (17) se encarga de devolver la respuesta al usuario (2), asegurando que solo los resultados autorizados y seguros sean compartidos. Por ejemplo, la puerta de enlace de API (17) se puede comunicar con un sistema de
30 almacenamiento de objetos a través de una interfaz de servicio web como Amazon S3, Google Cloud Storage, y Azure Blob Storage.

En una realización del método implementado en computador para generación de datos de certificación (12) de la presente divulgación posterior a la subetapa a2), se realiza una
35 subetapa a9) de balancear la carga de trabajo mediante las siguientes subetapas:

- f. recibir solicitudes de acceso mediante la dirección IP de un balanceador de carga (25);
- 40 g. distribuir las solicitudes de acceso entre instancias del grupo de destino;

- h. seleccionar la instancia del grupo de destino que esté disponible;
- i. comprobar que cada instancia del grupo de destino esté operativa y sea capaz de manejar el tráfico;
- j. redireccionar respuesta de solicitud de acceso;
- 5 k. registra información sobre las solicitudes y el rendimiento de los recursos;

Para el entendimiento de la presente divulgación un balanceador de carga (25) es un componente que se encarga de distribuir de manera equitativa las solicitudes de usuarios entre múltiples servidores o recursos, con el objetivo de optimizar el rendimiento, mejorar la
10 disponibilidad y evitar la sobrecarga de un servidor.

En una realización del método de la presente divulgación en el proceso de enrutar la solicitud de autenticación (18) se emplea un balanceador de carga (25) para gestionar eficientemente la distribución de las solicitudes de autenticación (18) entre múltiples instancias del servicio.
15 El balanceador de carga (25) opera como un intermediario, recibiendo las solicitudes de acceso mediante la dirección IP asociada al balanceador de carga (25).

La función principal del balanceador de carga (25) es distribuir de manera equitativa las solicitudes de acceso entre las diferentes instancias del grupo de destino. Utilizando
20 algoritmos de equilibrio de carga, selecciona la instancia del grupo de destino que esté disponible y sea capaz de manejar el tráfico en ese momento. Dentro de estos algoritmos se pueden utilizar, por ejemplo, Round Robin, que distribuye las solicitudes de manera secuencial en orden de llegada; Least Connections, que envía las solicitudes nuevas a instancias con menor cantidad de solicitudes activas; IP Hashing, que asigna solicitudes a
25 instancias basado en la dirección IP del cliente; o Weighted Round Robin, que asigna pesos a cada instancia, y las instancias con mayor capacidad reciben mayores solicitudes. Este enfoque permite una distribución uniforme de la carga de trabajo, ayudando a la eficiencia y el rendimiento del sistema.

30 El balanceador de carga (25) realiza constantes verificaciones para asegurarse que cada instancia del grupo de destino esté operativa y sea capaz de manejar el tráfico entrante. Si una instancia presenta problemas o no está disponible, el balanceador de carga (25) redirige la solicitud hacia una instancia alternativa, contribuyendo así a la disponibilidad y confiabilidad del sistema.

35 Además de su función principal de equilibrio de carga, el balanceador de carga (25) también puede registrar información sobre las solicitudes de autenticación (18) y el rendimiento de los recursos. Esta información registrada proporciona datos valiosos para el monitoreo y la optimización continua del sistema, permitiendo ajustes según las demandas del tráfico y
40 garantizando un rendimiento eficiente en el manejo de las solicitudes de acceso.

En un ejemplo la infraestructura del método de la presente divulgación es AWS, es decir que se aplica una plataforma de servicios en la nube proporcionada por Amazon Web Services (AWS) y el balanceador de carga (25) es el servicio de ELB (Elastic Load Balancing), más específicamente, el ALB (Application Load Balancing), que son balanceadores de aplicaciones. Estos ALB están asociados al Ingress del clúster, dependiendo, por supuesto, del servicio y el espacio de nombres (namespace). Cada uno de ellos tiene su propia regla o balanceador dedicado a recibir las solicitudes de las instancias. Asimismo, en el interior de cada uno de los servicios o pods, se configura un HPA (Horizontal Pod Autoscaler) específico por servicio. Esto permite que, cuando el tráfico aumenta, se puedan escalar horizontalmente los pods y distribuir la carga de trabajo de manera adecuada. El balanceador de carga (25) puede realizar por ejemplo las siguientes funciones:

E. Recibir solicitudes: el balanceador de carga (25) recibe solicitudes de los usuarios (2) que desean acceder a un servicio. Estas solicitudes llegan al punto de entrada del balanceador de carga (25), que es la URL o dirección IP pública del balanceador de carga (25).

F. Distribuir equitativamente: el balanceador de carga (25) distribuye automáticamente las solicitudes entrantes entre los recursos o instancias del grupo de destino que se han configurado previamente. Esto se hace de manera equitativa para evitar que una instancia se sobrecargue mientras otras instancias están disponibles.

G. Seleccionar el recurso destino: el balanceador de carga (25) selecciona el recurso del grupo de destino que esté disponible y responda más rápido en ese momento. Y utiliza algoritmos de balanceo de carga (ejemplo: Round Robin o Least Connections) para tomar esta decisión.

H. Verificar el estado de manera continua: el balanceador de carga (25) verifica constantemente el funcionamiento de los recursos del grupo de destino. Si una instancia falla en la verificación de funcionamiento, el balanceador de carga deja de enviar tráfico a esa instancia hasta que vuelva disponible.

I. Redireccionar respuestas: una vez que el recurso del grupo de destino maneja la solicitud, el balanceador de carga (25) redirige la respuesta al usuario (2) que hizo la solicitud inicial. Esto ocurre de manera transparente para el usuario (2), quien percibe que está interactuando directamente con el balanceador de carga (25).

J. Registrar y monitorizar: el balanceador de carga (25) registra información sobre las solicitudes y el rendimiento de los recursos. Además, permite la monitorización en tiempo real para evaluar el funcionamiento del sistema.

5 K. Elasticidad y escalabilidad: el balanceador de carga (25) se ajusta automáticamente a cambios en la carga de trabajo y la capacidad de recursos. Si es necesario, se pueden agregar o quitar instancias del grupo de destino para adaptarse a la demanda.

10 Para el entendimiento de la presente divulgación cuando se dice que el balanceador de carga (25) verifica continuamente el funcionamiento de los recursos del grupo de destino, significa que el balanceador de carga (25) se asegura que cada instancia o servidor pueda manejar el tráfico de manera confiable. Este proceso implica comprobaciones periódicas, conocidas como "health checks", que consisten en solicitudes regulares enviadas a cada instancia para
15 verificar su estado operativo. La frecuencia de estas verificaciones se establece en intervalos regulares, como por ejemplo segundos o minutos, según la configuración.

Si una instancia responde correctamente a la verificación de funcionamiento, el balanceador de carga (25) la considera operativa y apta para recibir tráfico. En caso de una respuesta
20 incorrecta o un estado no funcional, la instancia se marca como "no saludable", y el balanceador de carga (25) puede dejar de enviar tráfico a esa instancia, redirigiendo las solicitudes a instancias saludables.

Además, el balanceador de carga (25) puede, por ejemplo, realizar una recuperación
25 automática. Si una instancia inicialmente marcada como "no saludable" vuelve a un estado "saludable", el balanceador de carga (25) la reintegra en la distribución del tráfico. Este monitoreo constante garantiza que el tráfico se dirija a instancias operativas, mejorando la confiabilidad y la capacidad de recuperación del sistema.

30 Haciendo referencia a la FIG. 3 en una modalidad del método aquí divulgado, la arquitectura de la puerta de enlace de API (17) es dada por:

La unidad de cómputo (1) través de un DashBoard de aplicación (4), se conecta con la puerta de enlace de API (17), también llamada Rest API Gateway, que contiene toda la lógica de
35 negocio y se encuentra integrada en una infraestructura de servicios en la nube como puede ser Koibanx Cloud y ésta a su vez puede estar complementada con Google Cloud Storage (GCS).

En esta modalidad la puerta de enlace de API (17) se comunica con un servicio de
40 almacenamiento de objetos en la nube (35), como, por ejemplo, un servicio de S3 (Simple

Storage Service), que es un servicio de almacenamiento de objetos en la nube proporcionado por Amazon Web Services (AWS) y que puede ser utilizado para almacenar y recuperar datos desde cualquier lugar de la web. La puerta de enlace de API (17) se comunica mediante una librería AWS con los módulos (CORE) formado por los siguientes servicios:

5

- A. Auth (32) (Autenticación): un servicio encargado de gestionar el registro de usuario (2), la autenticación de credenciales de usuario (2) y la asignación de roles y permisos para controlar el acceso.
- 10 B. Notifications (33) (Notificaciones): un servicio que se encarga de notificar a los usuarios sobre eventos relevantes en la plataforma. Se comunica con servicios externos como sistema de mensajería (31) como por ejemplo Twilio a través de HTTPS para enviar notificaciones.
- C. Signer (28) (Firmante): un servicio que se utiliza para crear billeteras digitales (wallets) y permitir que estas firmen transacciones con un identificador único.
15 Es responsable de la generación y gestión de claves para firmar transacciones de manera segura.
- D. Algorand Stamper (29) (Estampador de Algorand): un servicio que se encarga de estampar datos en la red blockchain (13) de Algorand. Prepara, crea y envía transacciones a la red Algorand para que sean incluidas en la red blockchain
20 (13).
- E. Algorand- Connector (30): es un servicio que ayuda al usuario (2) a conectarse con la red blockchain (13) Algorand. Esto podría incluir, entre otras funciones, la gestión de transacciones, la interacción con contratos inteligentes en la red blockchain (13) de Algorand, o la obtención de información específica de la
25 blockchain (13).

Cualquiera de los anteriores servicios se puede conectar a una base de datos (7) que puede ser una base de datos no relacional, como por ejemplo MongoDB.

30 Para el entendimiento de la presente divulgación se debe entender que “servicios” se refiere a componentes de software, o métodos específicos que desempeñan funciones especializadas y esenciales para el funcionamiento de la puerta de enlace de API (17). Cada uno de estos servicios tiene una tarea específica y contribuye a la funcionalidad general del sistema.

35 En un ejemplo el servicio Auth (32) (Autenticación) realiza los siguientes pasos:

- F. Registro de usuario: cuando un nuevo usuario (2) se registra en la plataforma, el servicio Auth (32) (Autenticación) recopila y almacena la información del usuario (2), incluyendo credenciales de acceso.

- G. Autenticación: el servicio Auth (32) (Autenticación) verifica las credenciales de acceso del usuario (2) durante el proceso de inicio de sesión para garantizar que solo usuarios autorizados tengan acceso a la plataforma.
- H. Gestión de roles y permisos: Auth (32) (Autenticación) gestiona la asignación de roles y permisos a los usuarios (2) según sus necesidades y niveles de acceso en la plataforma.

En un ejemplo el Algorand Stamper (29) (Estampador de Algorand) realiza servicio el estampado cronológico en una red blockchain (13) por ejemplo una red blockchain Algorand de los estados del documento electrónico, que se realiza mediante las subetapas de:

- A1. Preparación de datos: el servicio Algorand Stamper (29) comienza preparando los datos que se deben estampar en la red blockchain (13). Esto puede incluir la estructura y formato adecuados de la información que se va a estampar.
- B1. Creación de transacción: basándose en los datos preparados, el servicio crea una transacción de Algorand Stamper (29) que contiene la información relevante que se va a estampar en la red blockchain (13) de Algorand Stamper (29).
- C1. Envío de Transacción: La transacción preparada se envía al nodo de estampado de Algorand Stamper (29) para su procesamiento. Esto implica el envío de la transacción a la red Algorand Stamper (29) para que sea incluida en la red blockchain de Algorand Stamper (29).
- D1. Estampado en la Blockchain: El nodo de estampado de Algorand Stamper (29) procesa la transacción y la incluye en la red blockchain de Algorand Stamper (29). La transacción se confirma y se convierte en parte del registro inmutable de la red blockchain (13).

En un ejemplo el servicio Notifications (33) (Notificaciones) realiza los siguientes pasos:

- A2. Notificación de eventos: cuando ocurre un evento relevante en la plataforma (por ejemplo, una transacción exitosa), el servicio Notifications (33) recopila los detalles del evento.
- B2. Conexión con sistema de mensajería (31) (por ejemplo, Twilio): el servicio Notifications utiliza una conexión segura, por ejemplo, a través de HTTPS, para enviar notificaciones a través del sistema de mensajería (31).
- C2. Entrega de notificaciones: las notificaciones se entregan de manera eficiente al destinatario adecuado, ya sea un usuario o un sistema externo, para informar sobre el evento ocurrido.

Haciendo referencia a la FIG. 5, en un ejemplo particular donde un documento electrónico incluido en un archivo (6) y que contiene un dato hash (15) obtenido mediante proceso basado en la red blockchain (13) de la presente divulgación, se puede imprimir en un documento PDF o documento impreso físicamente. A este documento se le agrega un código visual (34) para la identificación y captura de datos que utiliza patrones visuales para representar información. Un ejemplo de código visual (34) son los códigos de barras y códigos bidimensionales, que incluyen el grupo conformado por códigos de barras tradicionales, como los códigos de barras lineales, así como los códigos bidimensionales, que incluyen los códigos QR, los códigos de matriz de datos (como, por ejemplo, los códigos de barras PDF417 y DataMatrix), entre otros.

El código visual (34) permite la consulta de los datos de certificación (12) del documento electrónico obtenido en la etapa d) del método implementado en computador de la presente divulgación. En un ejemplo particular, el código visual (34) es un código QR.

Para el entendimiento de la presente divulgación, código QR (Quick Response) es un código que consiste en una matriz de puntos (cuadrados) dispuestos en un patrón cuadrado. Puede almacenar varios tipos de datos como texto, números, direcciones web, información de contacto o incluso instrucciones para ejecutar una acción específica, por ejemplo, en una unidad de cómputo (1) o en un dispositivo electrónico (27).

Por otra parte, la presente divulgación también se relaciona con un sistema configurado para ejecutar una o más de las modalidades, realizaciones y ejemplos del método de la presente divulgación anteriormente descritos.

El sistema puede incluir una unidad de cómputo (1) configurada para autenticar un usuario (2) mediante un proceso de verificación de identidad, donde el proceso de verificación de identidad incluye ejecutar al menos una primera autenticación y una segunda autenticación. La unidad de cómputo (1) se conecta con la terminal (3), desde la cual el usuario (2) accede a una aplicación (4) que conecta la terminal (3) con la unidad de cómputo (1). Además, el sistema puede incluir un servidor validador (9), que se conecta con la unidad de cómputo (1) por ejemplo, a través de la red de comunicaciones.

El sistema puede incluir un proceso para la creación y encriptación de una billetera digital (14) en una red blockchain (13) configurada para ejecutar cualquiera de las modalidades del método de la presente divulgación.

En cualquiera de las modalidades del sistema en las que se incluya unidad de cómputo (1), el servidor validador (9) se puede seleccionar entre una máquina virtual, un servidor, un computador, una granja de servidores, y combinaciones de estos.

Opcionalmente las bases de datos o módulos de almacenamiento de datos (7) puede hacer parte de una arquitectura de servicios basada en aplicaciones tipo “bucket” o servicios web de almacenamiento de datos en la nube (por ejemplo, Amazon S3, Firebase, entre otros) con la que los servicios de la puerta de enlace de API (17) puede comunicarse en un protocolo de comunicaciones basado en servicios.

Por otra parte, la presente divulgación también describe un programa de computador y un medio legible que comprenden instrucciones, que al ejecutarse en el programa de un sistema (de acuerdo con cualquiera de las modalidades del sistema anteriormente descritas), causan que dicho sistema lleve a cabo las etapas de un método, de acuerdo con cualquiera de las modalidades de los métodos anteriormente descritos en esta divulgación.

El medio legible por computador puede seleccionarse entre archivos ejecutables, archivos instalables, discos compactos, memorias RAM (memoria caché, SRAM, DRAM, DDR), memoria ROM (Flash, Caché, discos duros, SSD, EPROM, EEPROM, memorias ROM extraíbles (v.g. SD (miniSD, microSD, etc), MMC (MultiMedia Card), Compact Flash, SMC (Smart Media Card), SDC (Secure Digital Card), MS (Memory Stick), entre otras)), CD-ROM, discos versátiles digitales (DVD por las siglas en inglés de Digital Versatile Disc) u otro almacenamiento óptico, casetes magnéticos, cintas magnéticas, almacenamiento o cualquier otro medio que pueda usarse para almacenar información y a la que se puede acceder por una unidad de procesamiento.

El medio legible por computador puede ser un conjunto de elementos legibles por computador en los que se dividen o fraccionan instrucciones que al ser ejecutadas por unidad de cómputo (1), servidor validador (9) y demás elementos de hardware de cualquiera de las modalidades anteriormente descritas del sistema, permite llevar a cabo los pasos, etapas, subetapas de un método de acuerdo con cualquiera de las modalidades de los métodos anteriormente descritos en esta divulgación.

Ejemplo

Haciendo referencia a las FIG. 4A, FIG. 4B y FIG. 4C en un ejemplo la terminal (3) se conecta al dispositivo electrónico (27), desde donde accede el usuario (2). Además, la terminal de usuario (3) está conectada a la unidad de cómputo (1) a través de la cual se accede desde una aplicación (4). A su vez, la terminal de usuario (3) también está conectada al servidor validador (9), lo que le permite acceder a la aplicación (4).

El servidor validador (9) está conectado tanto a la terminal (3) como a la base de datos (7) y a un servidor de generación de OTP (One-Time Password) (20). La base de datos (7) está

conectada a la terminal (3), al servidor validador (9) y a un sistema para la creación y encriptación de una billetera digital (14) en una red blockchain (13). En la base de datos (7) se registra un dato de estampa de tiempo (10), el dato de identificación (11) del usuario (2) y un archivo (6) que incluye al menos un documento electrónico.

5

La unidad de cómputo (1) y el servidor validador (9) están conectados al servidor de generación de OTP (One-Time Password) (20). El servidor de generación de OTP (20) está conectado al balanceador de carga (25), el balanceador de carga (25) puede opcionalmente estar conectado una VPN por ejemplo Amazon Virtual Private Cloud (AVP) en un clúster de Amazon EKS en adelante se encuentra como VPC de EKS, la cual se conectada a la red blockchain (13) mediante la puerta de enlace de API (17). La puerta de enlace de API (17), opcionalmente se conectada tanto a la VPC de EKS como a la red blockchain (13), esta última a su vez está conectada a la base de datos (7).

10

15

La puerta de enlace API (17) está conectada al servidor validador (9), por ejemplo, a través de la conexión en serie de la VPC de EKS, y con el balanceador de carga (25) y con el servidor de generación de OTP (One-Time Password) (20). Dado que el servidor validador (9) está conectado tanto a la terminal (3) de usuario (2) como a la puerta de enlace API (17), esto permite una conexión entre la terminal (3) de usuario (2) y la puerta de enlace API (17) utilizando el servidor validador (9) como puente. Esta conexión permite que la terminal (3) de usuario (2) consuma los servicios Auth (32), Notifications (33), Signer (28), Algorand Stamper (29) y Algorand Connector (30).

20

25

En un ejemplo, la conexión entre los elementos del hardware, como la terminal de usuario (3), el dispositivo electrónico (27), la unidad de cómputo (1), la aplicación (4), el servidor validador (9), la base de datos (7), un servidor de generación de OTP (One-Time Password) (20), un sistema para la creación y encriptación de una billetera digital (14), una red blockchain (13), un balanceador de carga (25) y opcionalmente con una VPC de EKS, puede realizarse a través de una red de comunicaciones estándar utilizando protocolos como HTTP, HTTPS, TCP/IP u otros compatibles. La conexión puede ser directa, a través de una red local o privada, o a través de internet, utilizando enrutadores y switches para dirigir el tráfico de manera segura y eficiente. La arquitectura es distribuida, con diferentes componentes desplegados en servidores separados que se comunican entre sí a través de la red de internet. Se basa en principios de escalabilidad, disponibilidad y seguridad, utilizando tecnologías como balanceadores de carga para distribuir el tráfico, firewalls para la seguridad de la red de internet y servicios de almacenamiento en la nube para almacenar datos.

30

35

40

Haciendo referencia a las FIG. 4A, FIG. 4B y FIG. 4C en un ejemplo del método implementado en computador para generación de datos de certificación (12) de la presente divulgación, el flujo del método se inicia cuando usuario (2) accede desde una terminal (3) a

una aplicación (4) que conecta la terminal (3) con la unidad de cómputo (1) y se realiza la etapa a) de autenticar un usuario (2) mediante un proceso de verificación de identidad, donde el proceso de verificación de identidad incluye ejecutar al menos una primera autenticación y una segunda autenticación. Posteriormente, en la etapa b) del método de la presente divulgación se recibe desde una terminal (3) del usuario (2), una solicitud de certificación de estado (5) asociada a un archivo (6), donde el archivo (6) se almacena en una base de datos (7), y el archivo (6) incluye al menos un documento electrónico. Seguidamente se ejecuta la etapa c) transmitir a un servidor validador (9), una solicitud de verificación (8) que incluye el archivo (6), un dato de estampa de tiempo (10) y un dato de identificación (11) asociado al usuario (2). Después se ejecuta la etapa d) que consiste en recibir, desde el servidor validador (9), un dato de certificación (12) asociado a la solicitud de verificación (8), donde el servidor validador (9) obtiene el dato de certificación (12) mediante un proceso basado en la blockchain (13).

Haciendo referencia a las FIG. 4A, FIG. 4B y FIG. 4C en el ejemplo se realizan las subetapas de: i) validar si existe una billetera digital (14) asociada al usuario (2) en una red blockchain (13); ii) crear una billetera digital (14) asociado al usuario (2) cuando la validación es negativa; donde la billetera digital (14) incluye un dato de clave privada, un dato de clave pública, y un dato de identificación (11) asociado al usuario (2); iv) generar, con la red blockchain (13) asociada a la billetera digital (14), un dato hash (15) asociado al archivo (6) del dato de solicitud de verificación (8), y obtener el dato de certificación (12) a partir del dato hash (15). El dato de certificación (12) incluye el dato hash (15), un dato de estado del documento electrónico del archivo (6) y un dato de estampa de tiempo (10), asociado al momento en que se genera el dato hash (15), que se obtiene mediante estampado cronológico en la red blockchain (13).

Haciendo mención a las FIG. 4A, FIG. 4B y FIG. 4C en el caso de ejemplo, la conexión entre la unidad de cómputo (1), el servidor validador (9) y la red blockchain (13) se establece a través de una puerta de enlace de API (17). En la etapa inicial (a), el proceso de autenticación comprende las siguientes subetapas: a1) la transmisión de una solicitud de autenticación (18), que incluye el dato de identificación (11) del usuario (2) y la información de credenciales de acceso asociadas al usuario (2), mediante la puerta de enlace de API (17); a2) la dirección de la solicitud de autenticación (18) a un servicio de backend configurado para validar la información de credenciales de acceso asociadas al usuario (2) de acuerdo con un protocolo de seguridad predefinido; a3) la recepción, a través de la puerta de enlace de API (17), de un dato de respuesta (19) generado por el servicio de backend, con un valor que indica si la autenticación fue “aprobada” o “rechazada”. Es importante destacar que la puerta de enlace de API (17) se configura para restringir el acceso de la unidad de cómputo (1) a la billetera digital (14) en la red blockchain (13) de conformidad con el protocolo de seguridad predefinido.

Haciendo alusión a las FIG. 4A, FIG. 4B y FIG. 4C en el ejemplo, durante la etapa a), la segunda autenticación se desglosa en las siguientes subetapas: a4) el envío de una solicitud de autenticación (18) al servidor de generación de OTP (20); a5) la recepción, desde la terminal (3) del usuario (2), de un comando OTP (22) que contiene un código de único uso (26) generado por el servidor de generación OTP (20) y se envía electrónicamente a un dispositivo electrónico (27) del usuario (2). Adicionalmente, durante la etapa a), la segunda autenticación comprende las subetapas de: a6) enviar una solicitud de autenticación (18) al servidor validador (9), donde la solicitud de autenticación (18) incluye el dato de identificación (11) del usuario (2); a7) recibir desde el servidor validador (9) un dato de respuesta (19) con un valor indicando “aprobado” o “rechazado”. El servidor validador (9) genera el valor “aprobado” si el dato de identificación (11) del usuario (2) coincide con las credenciales (24) asociadas a una billetera digital (14) previamente creada por el usuario (2); a8) avanzar a la etapa b) si el valor del dato de respuesta (19) es “aprobado”.

En el contexto de las FIG. 4A, FIG. 4B y FIG. 4C, en la etapa a2), se ejecuta una subetapa a9) para equilibrar la carga de trabajo mediante un balanceador de carga (25).

GLOSARIO:

Puerta de Enlace API: una puerta de enlace API (Interfaz de Programación de Aplicaciones) se refiere a un componente que actúa como un intermediario entre diferentes aplicaciones o servicios, permitiendo la comunicación y la transferencia de datos entre ellos. La puerta de enlace API facilita la interoperabilidad entre sistemas heterogéneos al proporcionar un conjunto de reglas y protocolos para la comunicación. Puede encontrarse entre el usuario (2) y un conjunto de servicios de backend. Funciona como un proxy inverso que acepta todas las llamadas a la interfaz de programación de la aplicación (4), agrega los servicios necesarios para cumplir con las solicitudes y devuelve el resultado adecuado. La puerta de enlace de API intercepta, por ejemplo, las solicitudes que ingresan y se encarga de varias funciones necesarias, por ejemplo la puerta de enlace API puede realizar las funciones de dirigir las solicitudes del usuario (2) hacia los microservicios correspondientes, gestionar versiones y acceso, transformar el flujo de las comunicaciones, facilitar logging, monitorear los cortafuegos, administrar la autenticación y autorización, combinar llamadas a servicios, y posibilitar la escalabilidad distribuyendo tráfico de red equitativamente.

Endpoints: Se define como un punto de acceso específico en una API o servicio web. Es una URL única que representa una operación o recurso dentro de la API. Los endpoints son utilizados para realizar diversas acciones, como leer, escribir, modificar o eliminar datos. Los endpoints están asociado por ejemplo con métodos HTTP, como GET, POST, PUT, DELETE, entre otros, que define qué tipo de acción se realizará en el recurso. La estructura

de un endpoint generalmente sigue un patrón de URL que incluye la dirección base del servicio, seguida de una ruta específica que identifica el recurso y, en algunos casos, parámetros adicionales que pueden ser utilizados para filtrar o modificar la respuesta.

- 5 **Documento electrónico:** Un documento en formato digital que puede almacenar información de manera electrónica. Puede estar incluido en archivos como textos, imágenes, videos, o cualquier tipo de contenido que se pueda representar digitalmente.

- 10 **Blockchain o red blockchain (13):** Una cadena de bloques es una estructura de datos distribuida y descentralizada que registra transacciones de manera segura y transparente. Cada bloque contiene un conjunto de transacciones y está vinculado al bloque anterior, formando una cadena inmutable.

- 15 **Información inmutable:** Se refiere a los datos que se han registrado en una blockchain y que, una vez registrados, no pueden ser modificados ni eliminados. En el contexto de una red blockchain, la inmutabilidad garantiza que los registros históricos de transacciones, contratos inteligentes, documentos electrónicos y su confirmación de origen, características y elementos relevantes del documento electrónico, así como otros tipos de datos, permanezcan intactos y sean confiables a lo largo del tiempo

- 20 **Servidor validador (9):** Es un servidor que verifica la autenticidad y validez de la información o transacciones. En el contexto de la red blockchain (13), el servidor validador (9) puede participar en el proceso de consenso para confirmar transacciones y agregar nuevos bloques a la cadena.

- 25 **Aplicación (4):** Un programa informático diseñado para realizar tareas específicas en un dispositivo electrónico. Las aplicaciones, o apps, pueden variar desde simples herramientas hasta programas complejos con múltiples funciones. La aplicación (4) tiene la función de: Creación de usuarios, aprobación de roles según reglas de negocio, solicitud de generación de documento electrónico, aprobación y firma de documentos electrónicos, generación del documento electrónico, envío del documento electrónico al beneficiario, lectura y verificación de los documentos electrónicos, según por ejemplo QR, firma asociada y estampado cronológico.

- 35 **Twilio:** Es un servicio de comunicación en la nube a través de APIs, que permite a los desarrolladores integrar fácilmente funciones como mensajería de texto, voz y video en aplicaciones y sitios web.

- 40 **Credenciales de acceso asociadas al usuario:** Información utilizada para autenticar y verificar la identidad de un usuario al acceder a sistemas o servicios en línea. Las credenciales

de acceso suelen incluir un nombre de usuario y una contraseña, aunque pueden implicar otros métodos como por ejemplo biometría o claves de seguridad.

Servicios: Se refiere a un componente de software autónomo y especializado que proporciona una funcionalidad o una unidad funcional específica dentro la solución. Estos componentes de software están diseñados para ser independientes y modulares, lo que significa que pueden ser implementados, actualizados y escalados de forma independiente sin afectar a otros servicios o a la arquitectura en su conjunto.

Cada servicio expone una interfaz claramente definida, generalmente a través de una puerta de enlace API (17), que especifica cómo otros componentes pueden interactuar con este. Esta interfaz define los tipos de datos que el servicio acepta y devuelve, así como las operaciones que puede realizar.

En el contexto de la presente divulgación, los servicios realizan tareas como la autenticación de usuarios, la verificación de identidad, la certificación de documentos y la gestión de datos. Están diseñados para ser escalables, lo que significa que pueden manejar cargas de trabajo variables y recuperarse automáticamente de fallos sin interrumpir el funcionamiento en su conjunto.

Así las cosas, un servicio es un componente independiente que ofrece una funcionalidad específica a través de una interfaz bien definida (por medio de una puerta de enlace API (17)), permitiendo la interacción con otros componentes de la solución de forma escalable y sin dificultades de acoplamiento.

Unidad de cómputo (1), unidad de procesamiento, o módulo de procesamiento: Son un dispositivo que procesa datos, por ejemplo, microcontroladores, microprocesadores , DSCs (Digital Signal Controller por sus siglas en inglés), FPGAs (Field Programmable Gate Array por sus siglas en inglés), CPLDs (Complex Programmable Logic Device por sus siglas en inglés), ASICs (Application Specific Integrated Circuit por sus siglas en inglés), SoCs (System on Chip por sus siglas en inglés), PsoCs (Programmable System on Chip por sus siglas en inglés), computadores, servidores, tabletas, celulares, generadores de señales y unidades de cómputo, unidades de procesamiento o módulos de procesamiento conocidas por una persona medianamente versada en la materia y combinaciones de estas.

Dato: Es una representación simbólica que puede ser numérica, alfabética, algorítmica, lógica, y/o vectorial que codifica información.

Un dato puede tener una estructura o trama compuesta de bloques de caracteres o de bytes que representan diferentes tipos de información. Cada bloque se conforma de cadenas de caracteres, números, símbolos lógicos, entre otros.

- 5 También un dato puede formarse sólo de bits (cadenas en lenguaje binario), formarse de caracteres formados uno a uno por una combinación de bits, formarse a partir de campos, registros o de tablas formadas de campos y registros, o formarse de archivos de intercambio de datos (formatos como csv, json, xls, entre otros). Además, un dato puede ser una matriz de “n” filas por “m” columnas. A su vez un dato puede contener varios datos.

10

Por ejemplo, cuando el dato tiene estructura de trama, la trama puede tener un bloque de caracteres de identificación, conocida generalmente como encabezado o “header”, la cual contiene información relacionada con un dispositivo de cómputo o procesador que envía el dato, y puede contener información relacionada con un dispositivo de cómputo o procesador que recibe el dato. Preferiblemente, si el dato tiene un formato de trama, la trama contiene bloques relacionados con capas de acuerdo con, por ejemplo, el modelo de referencia OSI.

15

Asimismo, la trama puede tener un bloque de caracteres de cola (o simplemente cola), o “tail” en inglés, que permite identificar a una unidad de cómputo o servidor que es el fin del dato, es decir, que después de ese bloque ya no se encuentra información contenida en el dato identificado previamente por la unidad de cómputo o servidor con el “header”. Además, el dato tiene entre el bloque “header” y el bloque “tail” uno o más bloques de caracteres que representan estadísticas, números, descriptores, palabras, letras, valores lógicos (e.g. booleanos) y combinaciones de estos.

20

25

OpenID Connect (OIDC): Es un protocolo que se basa en Oauth 2.0 y añade una capa de autenticación. Permite a las aplicaciones web autenticar usuarios a través de un proveedor de identidad y obtener información del usuario en forma de tokens de identidad.

30

SAML (Security Assertion Markup Language): Es un estándar XML para el intercambio de información de autenticación y autorización entre partes, particularmente entre proveedores de servicios y proveedores de identidad.

35

JWT (JSON Web Token): Aunque no es un protocolo en sí mismo, JWT es un estándar para representar claims (afirmaciones) entre dos partes de manera segura. Es comúnmente utilizado como token de acceso en combinación con Oauth 2.0.

Kerberos: Es un protocolo de red que autentica a los usuarios y servicios en una red de computadoras donde la comunicación se realiza a través de un canal no seguro.

40

CAS (Central Authentication Service): Un protocolo de autenticación de entrada única (Single Sign-On, SSO) utilizado para autenticar usuarios en múltiples aplicaciones a través de un solo proceso de inicio de sesión.

- 5 **LDAP (Lightweight Directory Access Protocol):** Centrado en el acceso a directorios, también se utiliza para la autenticación y autorización en sistemas distribuidos.

Garantía: Es un documento emitido por parte de un deudor, mediante el cual éste asume el compromiso irrevocable de pagar o garantizar el pago hasta por el monto máximo garantizado a un beneficiario por cuenta de obligaciones a cargo del ordenante, en caso de que el cliente las incumpla.

Garantía Bancaria: Es un documento emitido por parte de una entidad bancaria, mediante el cual esté asume el compromiso irrevocable de pagar o garantizar el pago hasta por el monto máximo garantizado a un beneficiario por cuenta de obligaciones a cargo del ordenante, en caso de que el cliente las incumpla.

VPC: Es una red virtual prácticamente idéntica a una red tradicional que podría operar en su propio centro de datos. Una vez creada una VPC, podrá agregar las subredes. Un ejemplo de VPC es la Amazon VPC.

VPC de EKS: Es una red virtual dedicada y segura que se crea al configurar un clúster de contenedores o Kubernetes en un servicio en la nube. Esta VPC está diseñada específicamente para admitir clústeres de Kubernetes y se configura con las subredes necesarias para el correcto funcionamiento del clúster. Un ejemplo de VPC de EKS es aquella que se crea al implementar un clúster de EKS en AWS, proporcionando un entorno seguro y aislado para ejecutar aplicaciones en contenedores.

Instancia: se refiere a una ejecución específica o una copia individual de un componente de software o hardware dentro de un sistema más grande. Una instancia representa una entidad única y concreta que puede ser replicada o ejecutada múltiples veces en un entorno tecnológico, como un servidor, una aplicación o un proceso en ejecución. Las instancias son utilizadas para manejar tareas específicas de manera independiente dentro de un sistema, lo que permite una mayor escalabilidad y eficiencia en el uso de recursos.

Para el entendimiento de la presente divulgación se debe diferenciar entre credenciales (24) asociadas a una billetera digital (14) y el término dato de credenciales de acceso asociadas al usuario (2)

Se debe entender que la presente invención no se halla limitada a las realizaciones descritas e ilustradas, pues como será evidente para una persona versada en el arte, existen variaciones y modificaciones posibles que no se apartan del espíritu de la presente divulgación, el cual solo se encuentra definido por las siguientes reivindicaciones.

5