



Universidad Nacional de Misiones

Facultad de Ciencias Exactas, Química y Naturales

Tesis de Grado

**Propuesta de un Sistema de Validación de
Documentos Digitales con Tecnología Blockchain
de Actividades de Extensión en la Universidad
Nacional de Misiones.**

Para la obtención del grado de
Licenciatura en Sistemas de Información

Autor:

Ezequiel Agustín Britez

Director de Tesis:

Carlos Roberto Brys

Codirector:

Sergio Rubén Acosta

Octubre de 2021

Resumen

El área de Secretaría de Extensión de la Facultad de Ciencias Económicas (FCE) de la Universidad Nacional de Misiones (UNaM) emite certificados académicos de eventos o actividades efectuados por el sector. Los certificados académicos constatan que una persona participó de un curso o evento, también si lo aprobó. En este trabajo de tesis se diseña y desarrolla una propuesta de sistema de validación de documentos digitales con tecnología Blockchain para validar los certificados académicos emitidos por la entidad y respaldar que no fueron alterados.

Palabras Claves: blockchain, dirección, validación, contrato inteligente, documentos digitales.

Abstract

The Extension Secretariat area of the School of Economic Sciences of the National University of Misiones (UNaM) issues academic certificates of events or activities carried out by them. The academic certificates confirm that a person participated in the course or event, also if approved. This thesis is to design and develop a proposal of a digital document validation system with Blockchain technology to validate the academic certificates issues for the entity and backup that it hasn't been altered.

Keywords: blockchain, address, validation, smart contract, digital documents.

Dedicatorias

A mi familia por sus sacrificios, alientos y calidad de sus acciones.

A mis directores que me prestaron parte de su tiempo para finalizar este trabajo.

A mis amigos por motivarme a seguir.

Índice general

1. Introducción	17
1.1. Motivación	17
1.2. Problema	19
1.3. Objetivos	19
1.3.1. Objetivo General	19
1.3.2. Objetivos Particulares:	20
1.4. Hipótesis	20
2. Marco Teórico	21
2.1. Certificado Digital y Firma Digital	21
2.2. Conceptos de Interés	24
2.2.1. Integridad	24

2.2.2.	Documentos	24
2.2.3.	Redes entre pares	24
2.2.4.	Protocolo de Consenso	25
2.2.5.	Función Hash	25
2.2.6.	Nodos	25
2.2.7.	Prueba de Trabajo	26
2.2.8.	Prueba de Participación	26
2.2.9.	Prueba de Autoridad	27
2.2.10.	Dirección	27
2.3.	Blockchain	27
2.3.1.	Contratos Inteligentes	27
2.3.2.	Billetera	28
2.3.3.	Funcionamiento de la Blockchain	29
2.3.4.	Características de una Blockchain	33
2.3.5.	Clases de Blockchain	33
2.3.6.	Solidity	35
2.3.7.	Ethereum	35
2.4.	Aplicación Descentralizada	35
2.4.1.	OpenCerts	36
2.4.2.	BlockCerts	37

3. Antecedentes y Tendencias	41
3.1. Nivel Internacional	41
3.1.1. Antecedentes	41
3.1.2. Tendencias	43
3.2. Casos en Argentina	44
3.2.1. Antecedentes	44
3.2.2. Tendencias	45
3.3. Casos en Misiones	46
3.3.1. Antecedentes	46
3.3.2. Tendencias	47
4. Presentación del Caso	49
4.1. Contexto	49
4.2. Certificaciones de Actividades de Extensión en la FCE	51
4.2.1. Proceso para Generar y Emitir Certificados	53
4.2.2. Problemas Actuales	55
5. Modelado de la Solución	57
5.1. Procesos para la Construcción del Sistema	58
5.1.1. Análisis de Tecnologías y Métodos para Validaciones	59
5.1.2. Análisis de Tecnologías para Desarrollar en Blockchain	60
5.2. Análisis de la Propuesta de Sistema de Validación de Documentos Digitales	64
5.2.1. Bases del Sistema	65
5.2.2. Solución Propuesta	66

6. Desarrollo del Sistema Propuesto	69
6.1. Tecnologías y Herramientas Seleccionadas	69
6.2. Diseño de la Lógica	70
6.3. Diseño de Sistema	72
6.3.1. Estructura del Smart Contract	73
6.3.2. Roles y Permisos	78
6.4. Desarrollo del Smart Contract	79
6.4.1. Instalación de Metamask	80
6.4.2. Desarrollo del Smart Contract	83
6.5. Desarrollo de la Interfaz de Usuario	87
6.5.1. Desarrollo de la Vista del Sistema	89
7. Ensayos de Validaciones	95
7.1. Preparación Inicial	96
7.2. Ensayo A	97
7.3. Ensayo B	101
7.4. Ensayo C	105
7.5. Consideraciones Detectadas	107
8. Conclusiones	109
Bibliografía	113

A. Anexos del Desarrollo	121
A.1. Enlaces de Interés	122
A.2. Métodos de relevamientos	122
A.3. Comunicaciones Personales y Observaciones	122
A.3.1. Entrevista a Responsable de la Secretaría de Extensión de la Facultad de Ciencias Económicas	123
A.4. Código Smart Contract	128

Índice de figuras

2.1. Estructura de una cadena de bloque genérica, Fuente: imagen extraída del whitepaper de Nakamoto (Nakamoto, 2008)	31
2.2. Figura de una transacción, Fuente: imagen extraída del whitepaper de Nakamoto (Nakamoto, 2008)	32
2.3. Fuente: Página Web de OpenCerts (OpenCerts, 2018)	37
2.4. Circuito de validación de certificados con BlockCerts, Fuente: Imagen extraída de la página oficial de BlockCerts (BlockCerts, s.f.-b).	38
3.1. Comparación de búsquedas en google ,Fuente: Captura de pantalla de la página Google Trends el día 11/08/2021	43
3.2. Modelo del Proyecto Colmena, Fuente: extraída de la página noticiasdel6.com (Jimenez, 2020; Noticiasdel6, 2020).	47
5.1. MetaMask Plugin, Fuente: Captura de pantalla (producción propia)	62

6.1. Diagrama de Flujos de datos, Fuente: producción propia.	71
6.2. Diagrama de domino, Fuente: producción propia.	72
6.3. Smart Contract estructura, Fuente: producción propia.	74
6.4. Página para descargar Metamask, Fuente: captura de pantalla (página oficial de metamask).	80
6.5. Agregando el plugin, Fuente: captura de pantalla (página plu- gins chrome web store).	81
6.6. Inicio para crear o importar una wallet, Fuente: captura de pan- talla.	81
6.7. Menú crear wallet, Fuente: captura de pantalla.	82
6.8. Frases semillas, Fuente: captura de pantalla.	82
6.9. Creación de la wallet finalizada, Fuente: captura de pantalla. . .	83
6.10. Vista como compilar el contrato, Fuente: captura de pantalla (pá- gina de remix).	84
6.11. Seleccionando la Red de Ropsten, Fuente: captura de pantalla. .	85
6.12. Grifo de MetaMask, Fuente: captura de pantalla (página de me- tamask faucet).	85
6.13. Menú deploy Remix, Fuente: captura de pantalla (página remix).	86
6.14. Deploy Contrato Inteligente, Fuente: captura de pantalla (pági- na remix).	86
6.15. Contrato en la Blockchain de Ropsten, Fuente: captura de pan- talla (página remix).	87
6.16. Comando para instalar VUE CLI, Fuente: captura de pantalla. .	88

6.17. Configuración de proyecto, Fuente: captura de pantalla.	89
6.18. Instalación de librerías y componente, Fuente: captura de pantalla.	89
6.19. Archivos para conexión con la Blockchain, Fuente: captura de pantalla.	90
6.20. Copiar el ABI del código, Fuente: captura de pantalla.	91
6.21. Vista de Áreas, Fuente: captura de pantalla (propuesta de sistema).	92
6.22. Vista de Eventos, Fuente: captura de pantalla (propuesta de sistema).	92
6.23. Vista de Documentos como propietario, Fuente: captura de pantalla (propuesta de sistema).	93
6.24. Vista de Documentos como usuario público, Fuente: captura de pantalla (propuesta de sistema).	94
6.25. Vista de Organización, Fuente: captura de pantalla (propuesta de sistema).	94
7.1. Cambio de nombre de la organización, Fuente: captura de pantalla (propuesta de sistema).	96
7.2. Creación de nueva área en el sistema, Fuente: captura de pantalla (propuesta de sistema).	97
7.3. Carga de certificado Fondos Buitres, Fuente: captura de pantalla (propuesta de sistema).	98
7.4. Modelo certificado original de fondos buitres, Fuente: Brindada por la Secretaría de Extensión de la FCE.	99
7.5. Confirmación que el certificado se encuentra en la blockchain, Fuente: captura de pantalla.	99

7.6. Certificado Fondos Buitres Modificado, Fuente: captura de pantalla.	100
7.7. Validación de certificado modificado, Fuente: captura de pantalla.	101
7.8. Modelo de certificado de participación curso Geogebra, Fuente: Brindada por la Secretaría de Extensión de la FCE.	102
7.9. Certificado de Geogebra Validado, Fuente: captura de pantalla.	102
7.10. Cambio de nombre del Certificado, Fuente: captura de pantalla.	103
7.11. Certificado de Geogebra No Validado, Fuente: captura de pantalla.	104
7.12. Certificado de Geogebra Validado, Fuente: captura de pantalla.	104
7.13. Modelo de certificado original del Curso de Liquidación e Ingresos Brutos, Fuente: Brindada por la Secretaría de Extensión de la FCE.	105
7.14. Prueba de validación del certificado original, Fuente: captura de pantalla.	106
7.15. Certificado alterado del curso de Liquidación e Ingresos Brutos, Fuente: captura de pantalla.	106

Índice de tablas

7.1. Comparación de los hash del certificado original 7.4 y el modificado 7.6	101
7.2. Comparación de los hash del certificado original 7.8 y el modificado 7.11	105
7.3. Comparación de los hash del certificado original 7.13 y el modificado 7.15	107

CAPÍTULO *1*

Introducción

1.1. Motivación

Con la incorporación de las Tecnologías de la Información y la Comunicación (TIC) a la vida cotidiana cambió las prácticas de gestión, transacción e interacción de la información, por lo que las actividades que se desarrollan, tanto en el sector privado como sector público, requieren de una constante incorporación de nuevas herramientas para su gestión documental (Gauchi Riso, 2012).

La gestión documental implica un conjunto de operaciones y técnicas relativas a la concepción, desarrollo, implantación y evaluación de los sistemas

administrativos necesarios desde la creación de los documentos hasta su destrucción o transferencia a los depósitos de archivos. Para ello, a partir del desarrollo de la tecnología, se incorporó a los sistemas administrativos herramientas innovadoras para la gestión documental digital, lo cual requiere también métodos de validación (Drescher, 2017; Retamal, Roig, y Muños Tapia, 2017).

A los fines de dar validez a los documentos digitales, se han desarrollado métodos que permiten mantener la integridad de los mismos y así evitar problemas tales como duplicaciones o alteraciones de datos, y así tener valor de autenticidad y certeza de que los datos no hayan sido alterados desde su rúbrica (Drescher, 2017; Palomeque, 2015; Retamal y cols., 2017).

La tecnología Blockchain permite brindar beneficios tales como disminuir la probabilidad de modificar o realizar duplicación de datos sin autorización, así como también disminuir la probabilidad de recibir ataques informáticos (Badreddin, Rivera, y Malik, 2018; Choo, Dehghantanha, y Parizi, 2020; Drescher, 2017; Retamal y cols., 2017); la idea detrás de la tecnología blockchain se remonta a 1991 cuando Stuart Haber y W. Scott Stornetta describieron el primer trabajo en una cadena de bloques criptográficamente segura. En 1992, incorporó en el diseño los árboles Merkle, lo que permitió recopilar varios documentos en un bloque. Sin embargo, la tecnología blockchain como la conocemos hoy ganó importancia a partir de 2008, cuando se publicó el informe técnico “Bitcoin: un sistema de dinero en efectivo electrónico entre pares” bajo el seudónimo Satoshi Nakamoto (Rambo, Jara, Estigarribia, Syniuk, y Brys, 2021), y la misma puede ser utilizada de diferentes maneras, por lo que en la actualidad existen varias implementaciones de ella, manteniendo los beneficios de su utilización (BFA, s.f.-b; Dannen, 2017).

En cuanto a su aplicación en la gestión documental, los datos almacenados en la Blockchain, pueden ser representados por documentos digitales o por una identificación única que simbolice ese documento digital (BFA, s.f.-b).

La Universidad Nacional de Misiones (UNaM) genera a través de sus procedimientos administrativos distintos tipos de documentos, tales como resoluciones, ordenanzas, convenios, reglamentos, solicitudes, certificaciones y otros;

los cuales son de suma importancia, por lo que la utilización de la tecnología Blockchain permitiría validar la integridad de los datos pertenecientes a la documentación digital que se generen en la institución.

1.2. Problema

Las Secretarías de Extensión de la UNaM emiten diversas documentaciones que avalan la realización de actividades de extensión por parte de estudiantes, docentes, profesionales y público en general. Sin embargo, es dable destacar que, aquellas documentaciones digitales carecen de métodos de validación y que es necesario indagar acerca de tecnologías que brinden mayor seguridad y aseguren que no hayan sido alteradas desde su emisión (ver anexo A.3).

Esta necesidad impulsa al presente desarrollo a hallar respuestas a los siguientes interrogantes: ¿qué tecnologías permiten validar documentos digitales?, ¿qué antecedentes y tendencias existen para la validación de documentos digitales?, ¿cuáles son las prácticas generales que se realizan respecto a las certificaciones de actividades de extensión en la UNaM? y ¿de qué manera se podrían validar los documentos digitales emitidos por actividades de extensión en la UNaM que permitan garantizar la inmutabilidad de los certificados?

1.3. Objetivos

1.3.1. Objetivo General

Proponer un sistema de validación de documentos digitales con tecnología Blockchain de actividades de extensión en la Universidad Nacional de Misiones

1.3.2. Objetivos Particulares:

1. Investigar acerca de herramientas tecnológicas referentes a la validación de documentos digitales.
2. Exponer antecedentes y tendencias de validaciones con la tecnología Blockchain.
3. Releva prácticas y necesidades en el ámbito de la Universidad Nacional de Misiones respecto de la validación de documentos digitales de actividades extensión.
4. Diseñar la propuesta de sistema para establecer la validez de un documento digital con tecnología Blockchain.
5. Desarrollar la propuesta del sistema.
6. Ensayar y validar los resultados con el sistema propuesto en el ámbito de extensión correspondiente a una unidad académica de la UNaM.

1.4. Hipótesis

Un sistema de validación de documentos digitales con tecnología Blockchain relacionadas a actividades de extensión en la Universidad Nacional de Misiones, garantizaría la inmutabilidad de los certificados emitidos.

CAPÍTULO 2

Marco Teórico

Sin perjuicio de que existan diversas herramientas tecnológicas referentes o para validar documentos digitales, a efectos del presente trabajo se expondrán el certificado digital y la Blockchain.

2.1. Certificado Digital y Firma Digital

Por un lado, el Certificado Digital es un documento emitido por una organización que presta un servicio (denominada autoridad de certificación), el documento o archivo digital contiene datos relacionados a un usuario en particular, una persona física o una organización. El certificado puede ser emitido

en diversos formatos de archivos, la autoridad encargada reconoce y confirma los datos expuestos en el certificado digital, con utilización de técnicas criptográficas de clave pública y privada o de infraestructura de clave pública (Public key infrastructure) (Palomeque, 2015).

La infraestructura de clave pública permite a las entidades ser autenticadas y reconocidas por otras entidades mediante los certificados digitales que además permiten cifrar y descifrar mensajes, firmar digitalmente y garantizar su integridad (Palomeque, 2015).

Las partes o componentes más importante de una Infraestructura de Clave Pública o Public Key Infrastructure (PKI) son las que se exponen a continuación (Palomeque, 2015):

- La Autoridad de Certificación o Certificate Authority (AC): encargada de generar, emitir y revocar los certificados; es la responsable que da legitimidad a la clave pública con respecto a la entidad.
- La Autoridad de Registración o Registration Authority (RA): es la responsable de verificar si los enlace de las claves públicas corresponden a las entidades titulares.
- Los repositorios: se encargan de almacenar la información relativa a la PKI, como los repositorios que guardan las claves públicas y otras que resguardan las listas de claves revocadas.
- La Autoridad de Validación o Validation Authority (VA): se encarga de comprobar que los certificados digitales sean válidos.
- La Autoridad de Sellado de Tiempo o TimeStamp Authority (TSA): encargada de firmar los documentos para crear un registro específico en el momento exacto que existió.
- Los usuarios finales: obtienen una clave privada y una clave pública, también un certificado asociado a su clave pública. Mediante el uso de diversas aplicaciones hacen uso de la tecnología PKI para validar, cifrar y descifrar documentos.

Por otro lado, la Firma Digital es el mecanismo de criptografía de clave pública que permite dar seguridad al receptor del documento firmado digitalmente; la firma digital se realiza con la clave privada del remitente para cifrar la información a enviar y el receptor emplea la clave pública del remitente para descifrarla, para así garantizar la autenticidad del origen de la información y verificar que ésta no fue modificada desde su creación. Este es un método para validar si mensajes, documentos o archivos han sido alterados. Al método de clave pública y clave privada también se lo denomina método asimétrico (Palomeque, 2015).

La aplicación del método de cifrado asimétrico puede encontrarse en el caso de que, por ejemplo, una institución envía por correo electrónico un título digital a un estudiante. Es decir, la institución tiene en su poder una clave pública y una privada, la clave privada solo es conocida por la institución, ella no debe revelarla. Mientras que la clave pública es un código que cualquier individuo puede conocerla, la institución puede mostrar su clave pública. Para que el título tenga validez la institución utiliza su clave privada para cifrar el título digital. Una vez hecho esto, la institución envía el documento cifrado al estudiante y éste utiliza la llave pública para descifrar el documento y tener acceso al título digital. El método valida inequívocamente a la institución ya que cualquier mensaje cifrado con la clave privada solamente puede ser descifrado con la clave pública, por lo que ambas se complementan. Por lo tanto si el mensaje es interceptado y cambian el contenido, al momento de cifrarlo no podrá ser descifrado con la clave pública de la institución y de esta manera se verifica que la institución no envió el mensaje (AvanzaExportador, 2009; Garcia Rojas, 2008; Palomeque, 2015).

Las autoridades certificadoras juegan un rol importante ya que ellas son la que emiten estas claves, y validan la identidad de las organizaciones o personas. Por lo tanto si una entidad "A" quiere enviar un mensaje privado a una entidad "B", la entidad "A" debe consultar a la autoridad de certificación es la clave pública de la entidad "B" y confiar que esa es la correcta (Garcia Rojas, 2008).

El cifrado asimétrico no es el único método pero es uno de los más comúnmente utilizados para validar documentaciones digitales (Sheinix, 2020).

Utilizando este método de criptografía se puede asegurar que una entidad es quién dice ser y también que un mensaje fue emitido por ella. Por lo tanto el método de criptografía se utiliza para validar que una entidad es la que emitió un documento digital, pudiendo ser este algún certificado académico, título académico (Garcia Rojas, 2008; Palomeque, 2015; Sheinix, 2020).

2.2. Conceptos de Interés

Para mayor entendimiento sobre la temática del presente trabajo, se exponen a continuación determinados términos y conceptos que facilitarán la comprensión sobre las tecnologías utilizadas para la validación de documentos digitales.

2.2.1. Integridad

Cuando utilizamos la palabra Integridad se hace referencia a que los datos no sufrieron ningún tipo de cambio o que los datos se mantuvieron constantes (Badreddin y cols., 2018; Retamal y cols., 2017).

2.2.2. Documentos

Los documentos son los archivos o piezas documentales que necesitan de una validación para confirmar si sus datos son íntegros (Palomeque, 2015).

2.2.3. Redes entre pares

Las redes con arquitectura entre iguales o Peer to Peer permite la conexión entre iguales o pares, donde cada usuario cumple el rol de servidor o de cliente,

sin la necesidad de servidores centrales. Los datos están distribuidos dentro de la red de pares permite que los usuarios puedan acceder directamente a ella (Schollmeier, 2002).

2.2.4. Protocolo de Consenso

Son los protocolos que ejecutan los nodos de la red descentralizada para decidir qué bloques se van almacenar en la Blockchain y qué nodo es responsable de hacerlo (Badreddin y cols., 2018; Retamal y cols., 2017).

2.2.5. Función Hash

Función de resumen o función hash, es un algoritmo matemático que usa como entrada una cantidad de datos variables de longitud no definida y de salida genera una cantidad de datos variables de longitud fija. Sin importar los datos de entrada la función siempre mantiene la misma longitud para los datos de salida. Una de sus características importantes es la dificultad de encontrar la entrada de datos a partir de la salida, ya que el mínimo cambio en la entrada altera la salida (Drescher, 2017; Joaquín López Lérica, 2016).

2.2.6. Nodos

Los nodos de una Blockchain, son los participantes de la red de una Blockchain específica, donde cada uno almacenan los bloques y compiten por agregar nuevos y validar transacciones, los nodos contienen toda la información, si en algún momento todos los nodos de la red desaparecen, no existiría la Blockchain en la que los nodos formaban parte. A medida que el número de nodos de la red aumenta, también lo hará la descentralización y seguridad (Drescher, 2017; Nakamoto, 2008; Vázquez, 2020).

2.2.7. Prueba de Trabajo

La Prueba de Trabajo o Proof of Work (PoW) es un protocolo en el cual se le pide a un usuario o cliente que resuelva un problema matemático que requiere de poder computacional, la complejidad de él determina el tiempo aproximado que se tarda en resolverlo. En el caso de algunas Blockchain usan este protocolo mediante el cálculo de hashes, en el cual deben iniciar con una serie de ceros, por cada cero que inicie el hash el problema aumenta de manera exponencial (Back, 2002; Nakamoto, 2008; Vázquez, 2020).

2.2.8. Prueba de Participación

La Prueba de Participación o Proof of Stake (PoS) es un protocolo de consenso como PoW pero, la idea de PoS es reducir los costos de consumo energético producido por el poder computacional que se necesita con el protocolo de PoW (Brys, 2019; Vázquez, 2020).

En PoS cuenta con la analogía en el que los nodos de la red deben tener una cantidad de la moneda nativa de la Blockchain bloqueada, como consecuencia, los nodos con más cantidad de monedas bloqueadas tienen más posibilidad de ser elegidos para validar el nuevo bloque (Brys, 2019; Vázquez, 2020).

Cuanta más Criptomonedas tiene bloqueada el nodo, el incentivo en realizar actividades maliciosas en la red disminuye, porque algún ataque que perjudique a la red puede impactar en el valor de la moneda bajando su precio, por lo tanto la cantidad de dinero bloqueado se traducen en pérdidas, por ende el incentivo que querer controlar la red completa se reduce, en cambio teniendo más cantidad de la criptomoneda nativa, tiene más posibilidades de ser elegido como el nodo que forja el bloque y recibir la recompensa (Academia, 2018; King y Nadal, 2012; Vázquez, 2020).

2.2.9. Prueba de Autoridad

La Prueba de Autoridad o Proof of Authority (PoA) es el protocolo de consenso utilizado en redes Blockchain como la Blockchain Federal Argentina (BFA, s.f.-c).

La PoA funciona en una red entre pares donde se conoce la identidad de los nodos, que en comparación con el protocolo PoW no es necesario. Este protocolo brinda la ventaja de ejecutar un mayor número de transacciones en menos tiempo en comparación a los protocolos PoW y PoS (BFA, s.f.-c; Retamal y cols., 2017).

2.2.10. Dirección

Una dirección o address es la clave pública de un usuario de la red o elemento de la red Blockchain, el cual se pueden utilizar como dirección de destinatario en las transacciones, sirve para identificar a un usuario permitiendo enviar la transacción, es usada para encriptar la transacción y que pueda ser descifrada solamente por el poseedor de la clave privada correspondiente a la clave pública (Drescher, 2017).

2.3. Blockchain

2.3.1. Contratos Inteligentes

También llamados Smarts Contracts, son códigos de programas almacenados en la Blockchain donde asegura que no podrá ser modificado, contiene funciones y variables para resolver un problema específico. El contrato inteligente puede recibir y devolver datos. Para que los nodos verifiquen que un contrato inteligente fue correctamente ejecutado comprueban si los datos de entrada devuelven los datos de salidas propuestos por el primer nodo que

resolvió el código. Por ende se ejecutarán de la misma manera (BFA, s.f.-e; Raskin, 2017). Esto brinda la posibilidad de realizar cualquier tipo de aplicaciones y los consumidores no necesitan confiar en el programa, ya que tienen la libertad de visualizar las acciones del contrato antes de ejecutarlo (BFA, s.f.-e).

2.3.2. Billetera

Una Billetera o Wallet es una herramienta de software que permite a los usuarios enviar y recibir criptoactivos, algunas también permiten interactuar con los smart contract, se usan estas billeteras para que la comunicación con Blockchain sea mas sencilla. Un usuario con su misma clave privada puede usarla en cualquier billetera o wallet que soporte el software. Existen diferentes tipos de billeteras (Ambito, 2021; Dannen, 2017; Rezaeighaleh y Zou, 2019):

1. Hardware wallet: Son billeteras que almacenen la clave privada y pública de manera física, similar a un pendrive se consideran una de las formas más seguras, porque no están conectadas directamente a internet.
2. Wallet Online: Las claves están almacenadas en un servidor.
3. Wallet Escritorio: Aplicaciones descargadas y ejecutadas desde el computador, donde se puede acceder a los criptoactivos que se gestionan.
4. Paper Wallet: Las claves publicas y privadas son almacenadas en un papel físico.

Algunas de las Wallet que permiten gestionar sus criptosactivos son metamask, trezor (física), coinbase, entre otros (Ambito, 2021; Coinsenda, 2019; Dannen, 2017). Blockchain o en español cadenas de bloques es una base de datos que puede ser utilizada de forma compartida con usuarios que interactúan entre ellos de la manera peer-to-peer. La Blockchain almacena información y brinda la característica de ser inmutable y de estructurar los datos en un orden secuencial (Retamal y cols., 2017).

En ocasiones se confunde esta tecnología con el Bitcoin. Pero el Bitcoin es una moneda digital que utiliza la Blockchain, programada de una manera para obtener mayor grado de seguridad y seudonimato en comparación a otras Blockchain (Choo y cols., 2020).

En el caso del Bitcoin los datos son públicos y pueden ser consultados en cualquier momento por cualquier usuario. Entonces la Blockchain es una base de datos compartida y descentralizada donde la modificación de las transacciones sea casi imposible, los datos (registros o transacciones) son almacenados dentro de un bloque y puede contener una o más transacciones. Para que los bloques sean agregados a la Blockchain la mayoría de los nodos deben estar de acuerdo mediante algún tipo consenso o protocolo para validar la integridad de los bloques (Choo y cols., 2020; Retamal y cols., 2017).

Los nodos deben ejecutar algún tipo de consenso o protocolo para obtener un acuerdo de que transacciones se almacenan en la Blockchain, para esto existen diferentes consensos, como la Prueba de Trabajo o Proof of Work (PoW) (Retamal y cols., 2017), Prueba de Participación o Proof of Stake (PoS) (Drescher, 2017), Prueba de Autoridad o Proof of Authority (PoA), según el algoritmo utilizado se requiere un nivel mayor o menor de recursos y se obtiene más o menos seguridad. Estas bases de datos compartidas permiten que los registros almacenados no sean alterados. Para identificar quien es el propietario de los datos almacenados se le asigna una clave pública (identificadores visibles por todos los nodos de la red) y una clave privada al usuario (utilizado para firmar las transacciones) (Retamal y cols., 2017).

2.3.3. Funcionamiento de la Blockchain

La Blockchain como se describió anteriormente, es una base de datos compartida por todos los usuarios de una red peer to peer. En el caso de las Blockchain públicas los datos pueden ser accedidos en cualquier momento y por cualquier usuario, la información añadida a la cadena de bloques pasa por un protocolo de consenso, o proceso para confirmar que la información que

se pretende almacenar es correcta y cumple con las reglas definidas en la red (Dannen, 2017; Retamal y cols., 2017).

Los nodos de la Blockchain son pares iguales, en cada nodo se contiene la misma información con todas las transacciones realizadas dentro de un bloque que contiene un número de transacciones (en el caso de Bitcoin cuenta con alrededor de unas dos mil transacciones aproximadamente) (Retamal y cols., 2017).

Las llamadas transacciones, son los mensajes enviados desde una dirección a otra; en ella se almacenan datos como cantidades de una cierta moneda a gastar u otros, los que dependen de cómo se hayan definido las reglas de la Blockchain. Pero por lo general, las transacciones transmiten una cantidad de una criptomoneda y adjuntan datos extras. Ellas son enviadas a todos los nodos de la red en donde existe una competencia por cuál de los nodos va almacenar primero el bloque con la mayoría de las transacciones nuevas. La competencia se debe a que el nodo que resuelva un problema para poder agregar el bloque recibe una recompensa con la criptomoneda nativa de su Blockchain, como en el caso de la prueba de trabajo, consiste en que los nodos buscan obtener un hash para identificar al nuevo bloque con ciertas características como ser iniciar el hash con una cantidad de ceros. De la manera que trabaja Bitcoin, existen también otros tipos de competencias, ellas son denominadas algoritmos de consensos o protocolos de consensos. Esto sirve para incentivar a los usuarios ser parte de la red, a mayor cantidad de nodos la red se vuelve más segura (Brys, 2019; Nakamoto, 2008; Retamal y cols., 2017).

El bloque nuevo se agrega a la base de datos apuntando al hash que representa el bloque anterior, de esta manera cada bloque “apunta” hacia el bloque que le precede pudiendo ir hasta la raíz y obtener todo el historial de las transacciones. Los bloques almacenan información extra como el tiempo en el cual el bloque fue agregado a la cadena, la referencia hacia el bloque anterior, y otros datos dependiendo de las reglas que se han programado. En la figura 2.1 se muestra la estructura de un bloque, donde TX representa las transacciones almacenadas en el bloque y el nonce (un numero que solo puede usarse una vez)

es el número que hace que el hash del bloque cumpla con las reglas definidas para la competencia, por ejemplo, si se necesita que el hash del bloque inicie con tres ceros, los competidores (mineros) prueban diferentes valores, en el nonce hasta hallar el hash que inicie con los tres ceros, esto no ocurre en todas las Blockchain, sino en las que cuentan con el algoritmo de consenso prueba de trabajo (Drescher, 2017; Ethereum, 2021; Kelly, Lauer, Prinster, y Zhang, 2018; Nakamoto, 2008).

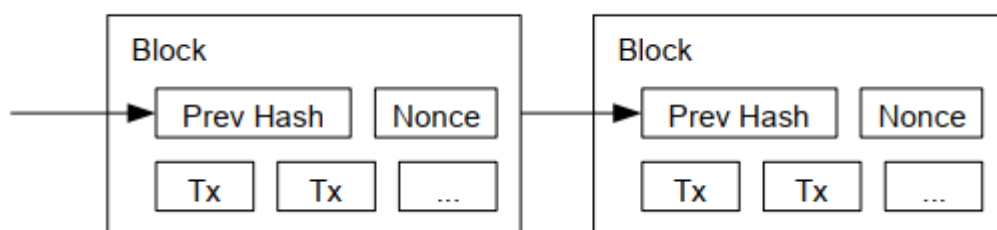


Figura 2.1: Estructura de una cadena de bloque genérica, Fuente: imagen extraída del whitepaper de Nakamoto (Nakamoto, 2008)

Las criptomonedas en la Blockchain son definidas como una cadena de firmas digitales, donde cada dueño transfiere la moneda al próximo, firmando digitalmente el hash de la transacción previa y la clave pública del destinatario. Entonces en la transacción se almacena el hash de la transacción anterior más el mismo hash firmada por el emisor de la transacción, con lo cual se puede comprobar quien fue el emisor y también quien es el receptor. Esta explicación se puede observar gráficamente en la figura 2.2 (Nakamoto, 2008).

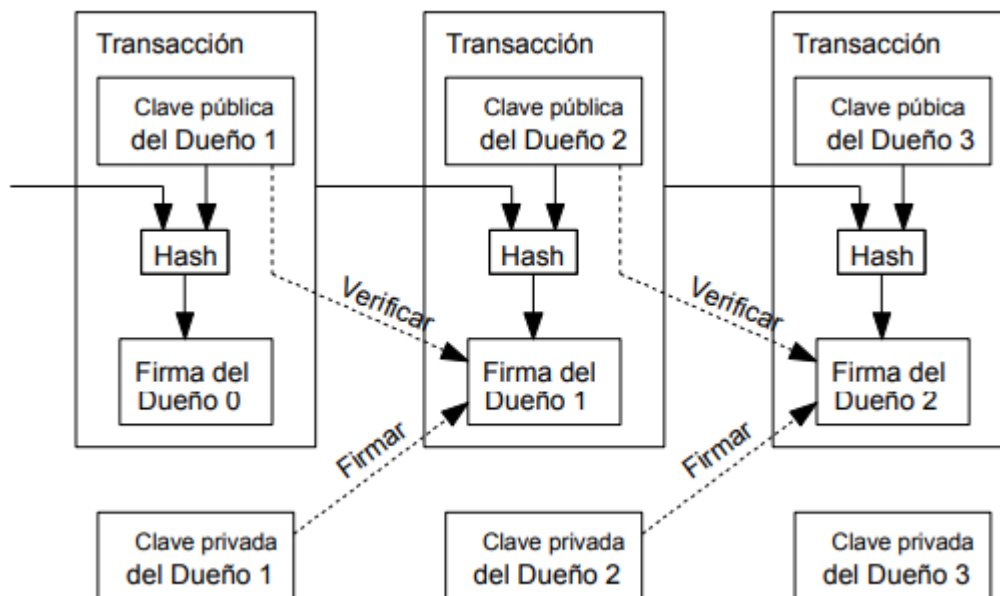


Figura 2.2: Figura de una transacción, Fuente: imagen extraída del whitepaper de Nakamoto (Nakamoto, 2008)

Con este protocolo cuando más grande es la red, más difícil es poder modificar sus estados actuales, porque un nodo que modifique algún estado interno ya almacenado, tiene que ser verificado para que todos los nodos los agreguen, si ven que no cumple con las reglas o los datos no son los mismos que los demás nodos, se rechaza. Por ende, los datos almacenados en la Blockchain no pueden ser modificados, sin que todos los nodos lo modifiquen. El autor de un ataque malicioso deberá tener el control de la mitad más un nodo para poder realizar sus acciones, pero esto resulta muy costoso y a medida que la red crece, cada vez es más difícil realizarlo ya que el poder de cómputo de la red se vuelve más compleja (Joaquín López Lérída, 2016; Nakamoto, 2008).

Un minero es un nodo en la red que recopila transacciones y trabaja para organizarlas en bloques, cada vez que reciben una transacción verifican si es correcta, y por el trabajo que realizan se les da una recompensa, que son una pequeña comisión que es cobrada al emisor de la transacción y dependiendo del algoritmo de consenso, también se minan nuevas criptomoneda por bloque almacenado (Joaquín López Lérída, 2016; Nakamoto, 2008; Preukschat y cols., 2018).

2.3.4. Características de una Blockchain

En cuanto a las características más destacadas de la tecnología Blockchain se pueden mencionar a las siguientes (Torres, 2020):

- Seguridad y Privacidad: Los usuarios de la red pueden realizar transacciones libremente sin brindar ningún tipo de información personal simplemente firmando con su clave privada para verificar que le corresponde la clave pública. Pero la seguridad de las cadenas de bloques se da en la manera que se interconectan los bloques, donde cada uno de ellos referencia el bloque anteriormente generando un identificador único a partir del algoritmo de hash por lo tanto si algún dato del bloque anterior es modificado el hash del siguiente cambiará y la cadena almacenada en el nodo no concordaría con los demás, por consecuencia es rechazada.
- Trazabilidad: Es posible realizar un rastreo de todas las operaciones que fueron realizadas en una dirección específica, y también conocer los recorridos de las transacciones que fueron ejecutadas, ya que todas ellas se encuentran enlazadas y almacenadas en los nodos de la Blockchain.
- Transparencia: En las Blockchain públicas se conoce su código fuente y el funcionamiento del sistema, esto permite a los usuarios entender qué acciones están permitidas o no en el sistema y ver su funcionamiento real.
- Confianza: No es necesaria la confianza entre dos usuarios para realizar una transacción ya que el protocolo se encarga de realizar el proceso.

2.3.5. Clases de Blockchain

Existen las Blockchain privadas, públicas e híbridas; las públicas son las nombradas a lo largo del documento y son las Blockchain de Ethereum, EOS, Bitcoin. Las primeras Blockchain fueron diseñadas para ser públicas. Cualquier individuo puede acceder sin ser usuario y leer las transacciones, cualquier persona o máquina puede convertirse en usuario y participar del protocolo, todos

los nodos son iguales, y seudónimas porque los dueños de las transacciones no necesitan brindar datos personales, por lo cual se desconoce quién es dueño de una dirección pero sí podrían ser rastreables dado su carácter público (Preukschat y cols., 2018).

Las Blockchains privadas también denominadas redes de permisos se caracterizan porque pueden ser accedidas y utilizadas en algunas o todas las transacciones por participantes de la red. También por lo general cuentan con distintos niveles de accesos según cómo se halla definido la Blockchain, por ejemplo, establecer diferentes tipos de roles. Las redes privadas están compuestas en la mayoría de los casos de un número inferior de nodos, ya que la cantidad de participantes es limitada, pero la seguridad de la Blockchain en general se caracteriza porque a mayor números de nodos mayor seguridad (Torres, 2020).

Al combinar las características de las Blockchains públicas y privadas surgen las Blockchains híbridas. La diferencia con las públicas y privadas es en el protocolo de consenso. En lugar de ser un sistema abierto donde todos pueden validar los bloques o uno cerrado donde una única entidad ejerce todo el control, nos encontramos con una red donde el proceso de consenso es controlado por varias entidades, un grupo de nodos funcionan como validadores de las transacciones (Torres, 2020).

La ventaja de usar la Blockchain es que todo los datos almacenados en ella son totalmente accesibles, visibles, y permiten trazabilidad por lo tanto es auditable (Drescher, 2017). Absolutamente todo dato almacenado es válido, porque el protocolo asocia una dirección o address, con algún dato específico, el mismo puede ser manipulado (pero los cambios siguen almacenados) como lo permita el smart contract, el poseedor de algún dato puede manejarlo según su voluntad cambiando estados del programas pero sin poder eliminar todo el historial de cambios realizados (Drescher, 2017).

2.3.6. Solidity

Solidity es un lenguaje de programación orientado a objetos influenciado por C++, Python y JavaScript utilizado para crear los contratos inteligentes en las máquinas virtuales de Ethereum, esto permite realizar aplicaciones que se almacenan en la Blockchain, aplicar estructuras de controles, creación de clases y comunicaciones entre smart contracts (Dannen, 2017; Ethereum, s.f.).

2.3.7. Ethereum

Ethereum es una plataforma open source, que sirve para programar contratos inteligentes, su diferencia con las primeras Blockchain como Bitcoin es que Ethereum permite resolver cualquier problema computacional. Esta Blockchain brinda la posibilidad de crear cualquier programa, almacenar y ejecutarlos de manera descentralizada, facilita la creación de nuevos activos digitales y aplicaciones. Existen Blockchains de pruebas, en la cual el funcionamiento es similar a la red principal, donde los desarrolladores pueden probar sus programas y conseguir las criptomonedas de manera gratuita, para hacer sus controles en esas testnet. Las redes de pruebas de Ethereum más conocidas son Ropsten Rinkeby, Kovan, etc (Dannen, 2017; Sadouskaya, 2017; Torres, 2020; Vázquez, 2020).

2.4. Aplicación Descentralizada

Las Aplicaciones Descentralizadas o Distributed Application (DApp) se ejecutan en una Blockchain, un usuario con una dirección puede ejecutar métodos definidos en un programa almacenado en la Blockchain, la lógica del programa dependerá del código que diseñó el programador. En Ethereum se permite diseñar programas con los smart contract, en el lenguaje de programación Solidity, el cual es ejecutado por una máquina virtual que permanece

en el nodo. Algunas DApps conocidas son UniSwap ¹ usada para intercambiar criptomonedas dentro de la Blockchain, Opensea mercado para compra y venta de criptoactivos (Dannen, 2017). Existen diferentes sitios para visualizar las diferentes aplicaciones de las Blockchain como [Dapp.com](https://dapp.com) , [DappRadar.com](https://dappradar.com) , entre otros.

2.4.1. OpenCerts

OpenCerts es una aplicación que permite a las entidades como escuelas, universidades, gestionen los certificados de sus alumnos de manera segura y sin intermediarios utilizando la tecnología Blockchain. La aplicación no almacena los datos privados de los alumnos, sino que utilizan las claves públicas referenciadas a los usuarios. La función de las organizaciones o entidades es gestionar los certificados de los alumnos asociándolos a su clave pública. Los estudiantes pueden consultar, descargar y compartir sus certificados. El administrador valida que las entidades dadas de alta sean correctas (OpenCerts, 2018).

Funciona generando un código único a partir del certificado e información extra considerada como necesaria para validar el documento en el futuro, y luego se crea un archivo con extensión “.opencerts”; dicho archivo se carga al sitio web de OpenCerts y compara el contenido con el almacenado en la Blockchain para verificar si existió el certificado en cuestión. Para dar de alta los certificados usan el smart contract donde también crean los métodos para emitir o revocar un documento (OpenCerts, s.f.).

¹<https://app.uniswap.org/>

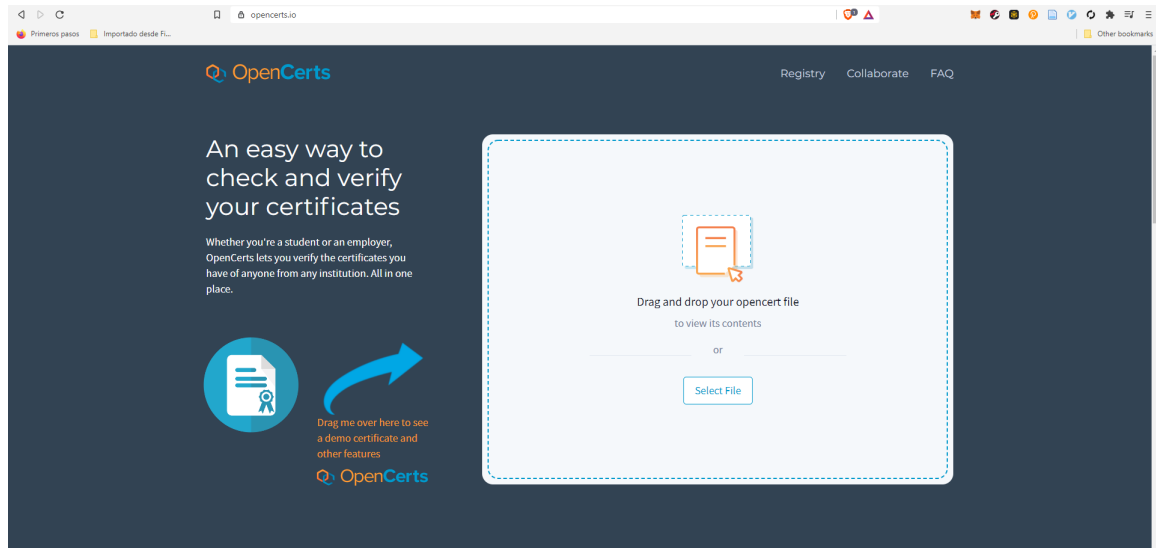


Figura 2.3: Fuente: Página Web de OpenCerts (OpenCerts, 2018)

En la imagen 2.3 se puede observar lo sencillo que es verificar un documento. Simplemente se arrastra el archivo de extensión“.opencerts” y se buscará en la Blockchain si realmente fue emitido por alguna entidad. OpenCerts utiliza los smart contract en la Blockchain de Ethereum. También utiliza tecnologías como Ract.js, Metamask, Web3.js, entre otros. Lo que permite desarrollar un sistema de certificados totalmente descentralizado (OpenCerts, 2018).

2.4.2. BlockCerts

Se define a si mismo como un estándar abierto desarrollado por el MIT Media Lab y Learning Machine, para la construcción de aplicaciones que emiten y verifican registros oficiales basados en la Blockchain. Pueden incluir certificados de registros civiles, académicos, licencias profesionales y más. Consiste en una librería con herramientas y apps móviles habilitando un ecosistema descentralizado, basado en estándar y habilitando verificación sin necesidad de la confianza mediante la tecnología Blockchain (BlockCerts, s.f.-b). Algunas universidades como el Instituto Tecnológico de Massachusetts (MIT), Tecnológico de Monterrey, la Universidad Harvard, la Universidad de California en Berkeley lo aplican. El estándar plantea que los certificados puedan ser com-

patibles a un nivel global, sin importar la Blockchain que se utilice pudiendo ser Bitcoin, Ethereum u otra (CriptoMonedasTV, 2018; Edublocs, 2019).

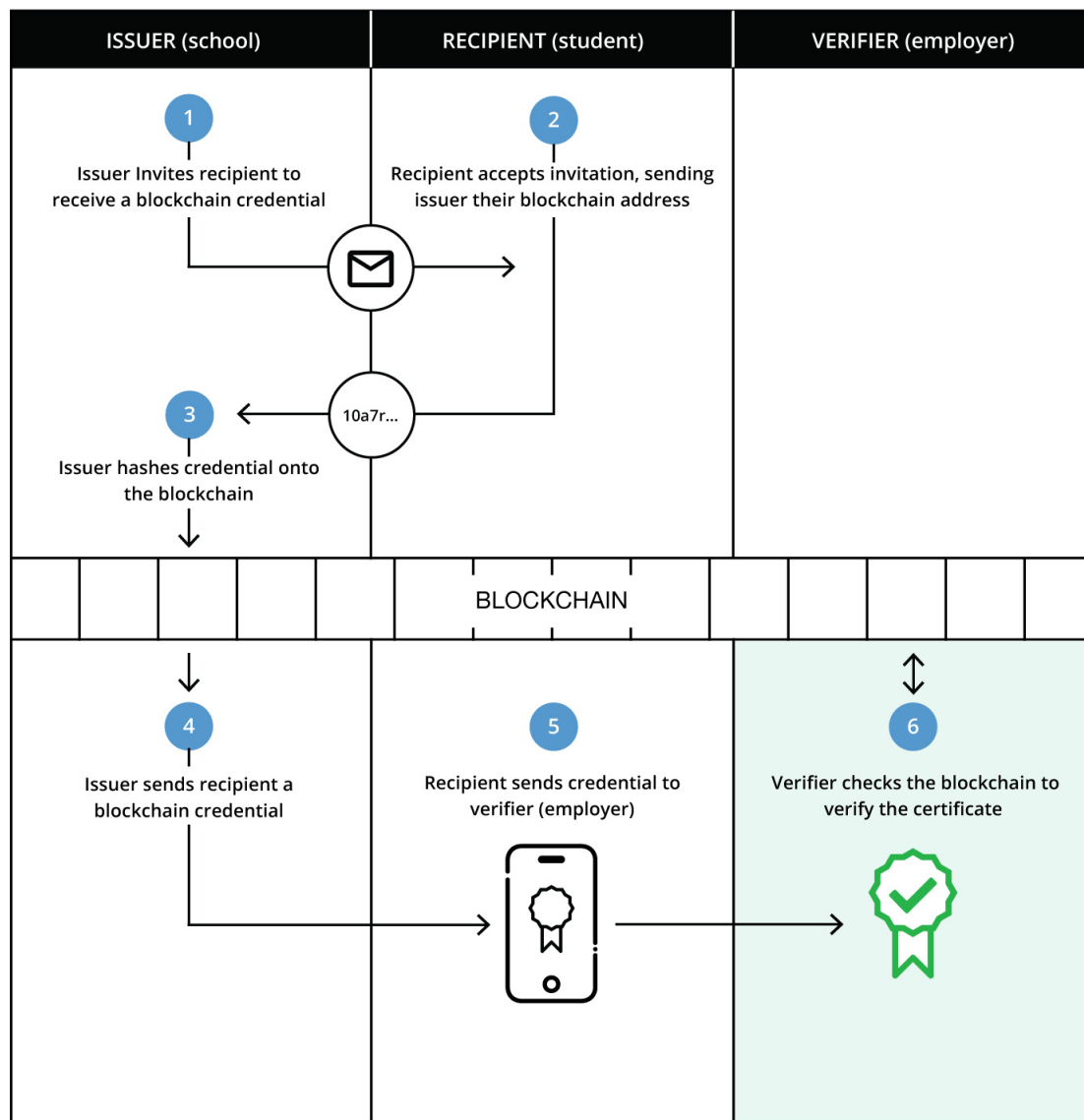


Figura 2.4: Circuito de validación de certificados con BlockCerts, Fuente: Imagen extraída de la página oficial de BlockCerts (BlockCerts, s.f.-b).

El flujo básico que se visualiza en la figura 2.4 para comprobar que un certificado se encuentra almacenado en la Blockchain y es validado por un instituto se explica a continuación:

En el paso 1, el emisor o institución invita a un usuario a que brinde su dirección de cuenta o su clave pública creada descargando, la aplicación móvil

que provee BlockCerts. En el paso 2, el usuario envía al emisor su clave pública. El paso 3 y 4, el emisor crea el hash a partir del certificado y lo almacena en la Blockchain para luego enviar un archivo de tipo JSON ², que contiene la información sobre el documento del estudiante o dueño del certificado. En el paso 5, puede enviar este documento a cualquier empresa o individuo que desee. En el paso 6 el usuario que posee el archivo puede verificarlo en el sitio web de BlockCerts (BlockCerts, s.f.-b).

²JavaScript Object Notation es un formato basado en texto estándar para representar datos estructurados (Mozilla, s.f.).

CAPÍTULO 3

Antecedentes y Tendencias

En el capítulo actual se exponen los antecedentes de la tecnología Blockchain utilizándola como soporte para la validación, así como las tendencias del uso de la tecnología en distintos niveles como el internacional, nacional y local.

3.1. Nivel Internacional

3.1.1. Antecedentes

Tras la presentación de Bitcoin a partir del año 2008 (Rambo y cols., 2021), como una solución al problema del doble gasto y un medio de pago pura-

mente electrónico (Nakamoto, 2008), surgen copias de esta red con mejoras en cuanto a la escalabilidad y velocidad de transacciones. Cuando Ethereum hizo presencia con un enfoque diferente a Bitcoin, expandió el uso de la Blockchain con la incorporación de los smart contract (Ethereum, 2020). No tardaron en surgir otras Blockchain similares a la de Ethereum, como EOS, NEO, cada una de ellas tiene su particularidad en sus objetivos, la manera de utilizarlos y sus protocolos.

Desde el surgimiento de Ethereum, se desarrollaron aplicaciones que permiten representar activos del mundo real, siendo estos: títulos de automotores, terrenos o propiedades inmuebles, títulos académicos, u otros instrumentos que representen valores para las personas. Desde el despliegue de Ethereum 1.0 en el año 2015 se crearon aplicaciones de video juegos, financieras, redes sociales, arte digital, entre otros. El mundo del Blockchain es un mar de proyectos e innovaciones, por ende se hará foco en el área de educación académica, especialmente en la validación de certificados, títulos o documentaciones relacionadas a esta área (Cheng, Lee, Chi, y Chen, 2018; Drescher, 2017).

La tecnología Blockchain es muy utilizada en la administración pública en países como Estonia, donde tienen un modelo de gobierno electrónico que es la identidad digital, con ella los estonios tienen sus datos que lo identifican electrónicamente, permite acceder a servicios del país y viajar por la Unión Europea. Tienen un sistema de ciudadanía e-Residency (Brys, 2019) que permite a los extranjeros viajar a distintos países con su registro digital. Usan sus documentos de identidad electrónica para editar y revisar documentos fiscales, solicitar beneficios de seguridad social y obtener servicios bancarios (Brys, 2019).

Estonia no es el único país que se beneficia de la tecnología Blockchain. Ucrania usa un sistema eAuction 3.0, usado para el alquiler o ventas de bienes del Estado para combatir la corrupción y disminuir la burocracia. En Suecia hay un proyecto que permite almacenar transacciones inmobiliarias de forma que todas las contrapartes: los bancos, los agentes, los compradores y vendedores pueden tener la oportunidad de seguir el proceso de la implementación

son muy consumidas en las últimas fechas por el retorno de inversión que obtienen los usuarios a depositar sus criptoactivos. Y por el otro lado los NFTs son unos criptoactivos que permiten representar la posesión de un elemento en particular, por ejemplo, una obra de arte digital se puede comprar en el ecosistema de Blockchain, pero solo unos pocos usuarios serían dueños de ellos por en la escasez de ellos hacen que los NFT se han vendido a precios elevados (Brys, 2019; Cripto247, 2021).

3.2. Casos en Argentina

3.2.1. Antecedentes

La Blockchain Federal Argentina (BFA) ¹ se describe a sí misma como una plataforma multiservicios, abierta y participativa, pensada para integrar servicios y aplicaciones sobre Blockchain. Una iniciativa confiable y auditable que permita optimizar procesos y funcione como herramienta de empoderamiento para toda la comunidad (BFA, s.f.-d). Sobre esta plataforma se han creado diversas aplicaciones de distintos ámbitos como la publicación de documentos como el Boletín Oficial de la República Argentina, de empresas privadas, también aplicaciones de trazabilidad, y otras que se pueden relacionar a:

1. El Ministerio de Educación, Cultura, Ciencia y Tecnología cuenta con el Registro Público de Graduados Universitarios que proporciona datos de egresados universitarios certificados por el Ministerio de Educación de toda la República Argentina. Gracias a la digitalización del trámite de certificación de diplomas y analíticos, y a la incorporación de Blockchain en el proceso, es posible autenticar la veracidad de la información contenida en el registro y que ésta sea accesible a la comunidad (BFA, s.f.-a).
2. Sistema de Información Universitario (SIU). El SIU-Diaguíta como el Módulo de Compras y el de Contrataciones y Patrimonio del SIU, son utilizados por más de 50 Universidades Nacionales y Organismos Públicos.

¹bfa.ar

Se incorporó el uso de la BFA a través de la funcionalidad de recepción de ofertas. De esta manera, las fechas y horas de las ofertas de los proveedores se registran en el sistema y queda certificada en la Blockchain (BFA, s.f.-a).

3. En la Universidad Nacional de Córdoba gracias a la digitalización de los sistemas de gestión de alumnos utilizados en universidades para la carga y archivo de actas de examen, promoción y equivalencia, entre otros datos, es posible verificar dicha información por medio de Blockchain, y garantizar al alumnado, personal administrativo y autoridades de las unidades académicas, que el sistema no puede presentar alteraciones en los registros sin que esas modificaciones sean detectadas (BFA, s.f.-a).

3.2.2. Tendencias

Por otro lado el gobierno Argentino impulsa un proyecto de ley (La Honorable Cámara de Diputados y el Senado de la Nación, 2020) relacionada a las criptomonedas y activos digitales, el objetivo es crear un marco regulatorio integral, aplicable a las transacciones y operaciones civiles y comerciales de criptoactivos y permitir el crecimiento del ecosistema local (D'Agostino, 2020).

El artículo describe que la promulgación de esta ley permitirá:

1. El Estado pueda determinar qué Proyectos Criptos autoriza y cuáles no en base criterios legales que hoy no existen.
2. Las empresas tengan modelos de negocios que cumplan con ciertos criterios, como una Blockchain pública o casos de uso.
3. Se puedan representar tenencias de acciones, una propiedad, etcétera.

La definición de un criptoactivo posibilitará que las capitales de riesgos internacionales que pretenden invertir tengan seguridad jurídica y la certeza destinado los fondos (D'Agostino, 2020).

3.3. Casos en Misiones

3.3.1. Antecedentes

El proyecto Colmena está basado en la participación ciudadana para recuperar residuos y generar un nuevo modelo económico que recompensa a los ciudadanos con la criptomoneda JellyCoin por sus aportes a partir del aprovechamiento de los residuos. Iván Zubilewicz, director del Proyecto Colmena, explicó que es un modelo para recuperar los residuos a través de una economía colaborativa con la tecnología Blockchain. El modelo integra al usuario que quiere recuperar los materiales residuales por medio de la plataforma y el recolector puede visualizar los residuos que la comunidad acumula para luego transportarlo a los emprendimientos industriales. El modelo permite incentivar al ciudadano a cuidar el medio ambiente y a su vez obtener JellyCoin, la cual las empresas deberán utilizar para comprar los residuos recolectados y formar un circuito de economía con la criptomoneda como muestra la figura 3.2 extraída de la página noticiasdel6.com Proyecto Colmena: un modelo de recuperación de residuos (Jimenez, 2020; Noticiasdel6, 2020).



Figura 3.2: Modelo del Proyecto Colmena, Fuente: extraída de la página noticiasdel6.com (Jimenez, 2020; Noticiasdel6, 2020).

3.3.2. Tendencias

La Cámara de Diputados de la Provincia de Misiones de Argentina aprobó el proyecto denominado como Programa Misionero de Innovación Financiera con Tecnología Blockchain y Criptomoneda (Cámara de Representantes de la Provincia de Misiones, 2020). A partir del proyecto la Provincia podrá emitir su propia Criptomoneda y almacenar datos de la administración pública en la Blockchain. Los objetivos principales son emitir una propia StableCoin (Moneda Estable) que usará la Provincia como herramienta de financiación entre el sector público y privado, otro uso es la gestión de datos para la administración pública y por último la emisión de certificados verdes con el uso de la tecnología usarlo para validar títulos o certificados (Clementín, 2021).

CAPÍTULO 4

Presentación del Caso

4.1. Contexto

El estatuto describe a la UNaM como una institución Universitaria de Derecho Público, autónoma en lo académico e institucional y autárquica en el sector económico y financiero. La institución tiene asiento en la provincia de Misiones de la República Argentina (UNaM, 2012).

Impulsa la relación e integración con las instituciones afines, gubernamentales y no gubernamentales de la provincia, regional, nacional e internacional, que compartan o coincidan con sus fines y objetivos (UNaM, 2012).

Los fines de la UNaM son descritos en el estatuto lo cual se cita textualmente a continuación:

1. “La preservación promoción y difusión de la cultura universal con énfasis en lo nacional y regional.”
2. “El resguardo acrecentamiento y difusión del conocimiento universal y del generado en su propio ámbito.”
3. “La organización instrumentación y evaluación de la enseñanza-aprendizaje en los niveles de su competencia y su articulación con los otros sectores del sistema educativo.”
4. “La aplicación del conocimiento a la solución de problemas del desarrollo humano en la provincia , la región y el país.”
5. “El compromiso con la conservación y preservación del medio ambiente y los recursos naturales.”
6. “El de constituirse en un ámbito de formación ciudadana y ejercicio democrático.” (UNaM, 2012)

La UNaM esta integrada por la Facultad de Humanidades y Ciencias Sociales (FHyCS); Facultad de Ciencias Económicas (FCE); Facultad de Ciencias Exactas, Químicas y Naturales (FCEQyN); Facultad de Ingeniería (FI) ; Facultad de Arte y Diseño (FAyD); Facultad de Ciencias Forestales (FCF) (UNaM, 2012). Cada facultad de la UNaM cuenta con las diferentes Secretarías, Direcciones y Departamentos que necesiten gestionar información sobre los alumnos.

Prestando atención a las secretarías las cuales varían sus nombres según la facultad, ellas son la Secretaría Administrativa, Secretaría Académica, Secretaría de Investigación, Secretaría de Bienestar Estudiantil y Secretaría de Extensión.

Todas tienen en común la gestión de documentos y de información por ende deben generar nuevos documentos a partir de los hechos ocurridos, como

ejemplo, la creación de un certificado de título académico por consecuencia de que un alumno de la facultad finalizó el cien por ciento (100 %) de su carrera. O la generación y gestión de cualquier otro documento que sea utilizado por la entidad como los planes de estudios, historia académica de los estudiantes, actas de exámenes, informes, entre otros (ver anexo A.3) (UNaM, 2012).

La problemática que inicialmente impulsó la investigación es la validación de documentos relacionados a los estudiantes, sean certificados o registros que puedan ser alterados. Dada la gran variedad de documentación que utiliza la universidad, se delimitó el alcance de la investigación en un área específica, aún especificando un sector, los documentos pueden ser variados, por lo tanto es necesario enfocarse en documentos relacionados a lo académico, con respecto a este tipo de documentación son considerados los que manejan datos de estudiantes, o participantes de eventos, excluyendo todo tipo de documentación relacionada al sector financiero o asuntos administrativos (ver anexo A.3).

El área seleccionada para realizar la presente investigación es la Secretarías de Extensión de la FCE. Esta área tiene que gestionar documentaciones de estudiantes y participantes de eventos por lo tanto, es un área adecuada para desarrollar el presente proyecto. Se decide por este sector FCE de la UNaM por el acceso a la información y acotar el alcance de la investigación.

4.2. Certificaciones de Actividades de Extensión en la FCE

Las actividades universitarias de la UNaM pretenden promover la interacción con el medio en la cual integra, aportando al crecimiento social (UNaM, 2012).

El estatuto (UNaM, 2012) de la UNaM describe que las actividades de extensión pueden implicar transferencia científico-tecnológica, educación perma-

nente, difusión de actividades y producciones de la UNaM, el desarrollo de las expresiones culturales y vinculaciones institucionales.

A un nivel jerárquico general está la Secretaría General de Extensión Universitaria (SGEU), como lo define su nombre es la secretaría general encargada de gestionar y coordinar con las secretarías de extensión de las distintas facultades (UNaM, 2012).

Por otra parte, la FCE cuenta con su propia Secretaría de Extensión donde llevan a la práctica tareas como la generación de cursos, congresos, charlas, capacitaciones, eventos de cualquier características, que permita cumplir con sus objetivos de expansión de conocimientos y crecimiento (ver anexo A.3).

El área administrativa de la Secretaría de Extensión es la encargada de realizar la gestión de las actividades que se planea efectuar. Los eventos pueden ser iniciados por docentes, no docentes, funcionarios, secretarios o estudiantes pero el evento o actividad debe estar aprobada por algún instrumento. Estos instrumentos pueden ser las disposiciones o resoluciones que representan documentaciones que validan e impulsan la creación y ejecución de las actividades (ver anexo A.3).

Por otro lado, para que una actividad sea aprobada es necesario que la propuesta tenga relación con los proyectos o programas planeados previamente. Esto quiere decir que las actividades deben estar justificadas y sus temas deben tener una relación estricta con los objetivos que la facultad propuso en sus proyectos o programas. En el caso que se presente una propuesta de actividad que no se encuentra relacionado directamente a los proyectos o programas, existe la posibilidad de presentarlo a las autoridades de la Secretaría de Extensión para evaluarlos y dictaminar su aprobación o rechazo (ver anexo A.3).

Una vez aprobada la actividad o el evento, el personal administrativo inicia con las preparaciones para relanzarlo. Se gestionan y reservan las aulas necesarias o en caso que los participantes deban asistir físicamente una fecha y hora determinada, se prepara el registro para quienes van dirigidos, ya que los eventos se realizan para un grupo selecto o también para todo el público

interesado. Se inicia toda la logística necesaria para el evento en particular (ver anexo A.3).

Para que los participantes de los eventos tengan un documento que constate su presencia en las actividades, se les entrega un certificado que dependiendo de la situación son de asistencia al evento o exámenes aprobados (ver anexo A.3).

Depende de como se organizó el evento y su magnitud, puede ocurrir que un evento cuente con más de una charla y a su vez más de un curso. Por lo tanto al planificar las actividades se opta cómo se entregarán los certificados: uno por charla asistida o por asistencia total del evento sin detallar la actividad que realizó (ver anexo A.3).

Los tipos de certificados académicos que se expiden en la secretaría de extensión de la FCE, son en formato papel, hojas tamaño A4 u oficio, también certificados digitales como PDF e imágenes (ver anexo A.3).

4.2.1. Proceso para Generar y Emitir Certificados

A continuación se describe el proceso que se utiliza en la generación y emisión de los certificados relacionados a las actividades de la Secretaría de Extensión de la FCE; cabe aclarar que se explica el proceso para la certificación y no toda la gestión que se lleva a cabo para que una actividad se realice.

El proceso para la emisión y generación de los certificados para un evento está explicado en la entrevista de realizada al encargado de la Secretaría de Extensión de la FCE de la UNaM (ver anexo A.3):

1. Días antes del evento se realiza el modelado de los certificados correspondiente a las actividades que se planea realizar, dependiendo si los certificados serán solo por asistir al evento, se crea un modelo. Pero en el caso que se realice certificación por algún examen aprobado o realizado, se crean la cantidad de modelos de certificados según los exámenes a evaluar.

2. Una vez diseñados los modelos ideales para el evento se realiza el escaneo de las firmas de las autoridades de la facultad que representan a la institución académica, en este caso el secretario de extensión y otros responsables de ser necesarios.
3. La firma escaneada de las autoridades se integra en los modelos de certificados, generando la documentación con la firma de las autoridades.
4. El día del evento se registran los participantes que asisten, tomando los datos necesarios para los certificados.
5. Al finalizar el evento se verifican los datos de las personas que se registraron, y las que asistieron a ellas. En este punto se determina quiénes de los participantes, recibirán los certificados de asistencia.

Con los datos de los participantes del evento se completan los certificados de asistencia de manera manual. Se agregaron los datos necesarios como Apellido y Nombre, en otros casos otros datos extras como DNI, correo electrónico, los datos del certificado dependerá de como se modeló el certificado.

6. Cuando se finalizó con la carga de datos en los certificados de asistencias se entregan a los participantes. Depende de la planificación de la actividad, se entrega el certificado impreso en papel. O se envía el certificado digital por correo electrónico a cada participante.
7. Si la actividad contempla la certificación de aprobación de exámen, o algún otro requerimiento específico. Se comprueba cuántos y quiénes de los participantes realizaron y aprobaron los requerimientos o el exámen.
8. Una vez filtrados los participantes que aprobaron los requerimientos o exámenes, se les crean manualmente (se ingresa el nombre y otros datos necesarios) el certificado perteneciente a cada participante que haya cumplido con las exigencias.
9. Finalizada la creación de los certificados de aprobación de exámen o el requerimiento que se solicita para el certificado en particular se realiza la

entrega del mismo modo que los certificados de asistencia. Mediante correo electrónico y en caso de ser físico, se contactan con los participantes para enviarles los certificados.

Este último proceso de certificación se separa del certificado de asistencia. Porque de acuerdo al volumen de participantes y el criterio para evaluarlos, puede consumir más tiempo.

4.2.2. Problemas Actuales

Cuando los participantes pretenden demostrar que asistieron o fueron parte de un evento de actualización, capacitación o curso, ellos no tienen la manera de validar que sus documentos digitales son los originales o si realmente fue emitido por la UNaM, la única manera es con la impresión del certificado y llevarlo a la Secretaría de Extensión de la FCE para sellar dando confirmación de que es un certificado emitido por la institución. Pero la validación de los mismos de manera digital no cuenta con un método definido para realizarlo (ver anexo A.3).

La Secretaría de Extensión de la FCE genera los certificados de manera física, los sellan y entregan físicamente; en los casos que el participante de los eventos, charlas o cursos necesite demostrar que asistió a las actividades el proceso para validar sus certificados es lento y con probabilidades de ser alterados. En comparación con un proceso de validez digital que permite ser automatizado, por lo tanto los medios de envíos son mas rápidos como por ejemplo correos electrónicos y brinda mayor velocidad en la validación de los certificados. El método que utilizan para que los certificados digitales sean considerados válidos es incluir las firmas ológrafas de los responsables como el Secretario de Extensión o también los profesionales que son parte de la organización del evento, pero estas firmas pueden ser copiadas y adaptadas a otros certificados de esta manera crear documentos que no fueron emitidos por la Secretaría de Extensión de la FCE. Las entidades externas que desean averiguar si una persona realmente estuvo en el evento de interés, no tiene

manera de comprobar la veracidad o autenticidad del certificado, por ende, el problema puede extenderse a dudar de la validez de los certificados (ver anexo A.3).

CAPÍTULO 5

Modelado de la Solución

Como se mencionó en el capítulo anterior, el problema principal surge de la no utilización de métodos que permitan validar los documentos digitales, emitidos por la Secretaría de Extensión de la Facultad de Ciencias Económicas (FCE) de la Universidad Nacional de Misiones (UNaM). En este capítulo se explica cómo abordar la solución a este problema.

Se propone un sistema que permita al usuario consultar la validez de sus documentos digitales por la entidad que los emitió y cuando se publicó, asimismo que permita a los gestores de la organizaciones subir nuevos documentos o algún identificador para comprobar de manera inequívoca si fue emitido por ellos, aun cómo mostrar datos relacionados al evento de la institución u

organización. Con esta propuesta de sistema, se atenderían los aspectos mencionados en cuanto a las necesidades de brindar un medio por el cual se pueda verificar que la documentación de las actividades desarrolladas por la Secretaría de Extensión de la FCE no fue adulterada, modificada o cambiada.

En cuestiones generales para solucionar los problemas se propone realizar un sistema que permita a la Secretaría de Extensión de la FCE de la UNaM :

1. Dar validez a un documento por tiempo determinado y gestionar sus estados.
2. Definir qué entidades externas puedan validar la integridad de documentos digitales emitido por la Universidad.
3. Permitir acceso a todo público al sistema para la validación.
4. Proteger la privacidad de los datos que pertenecen a los documentos.

A tales efectos los ítems recientemente expuestos se pueden lograr con la utilización de la tecnología Blockchain, otorgando inmutabilidad a los datos almacenados y que ellos sean accesibles a todo público. En cuanto a protección de la privacidad es necesario utilizar, algún método criptográfico para que los datos sean leídos, solo por las partes autorizadas. Y por último, para que las entidades o cualquier individuo pueda validar el documento digital, si es el mismo que emitió la Secretaría de Extensión de la FCE.

5.1. Procesos para la Construcción del Sistema

Serán necesarios para el diseño, modelado y desarrollo de la solución definir las herramientas a utilizar, y los pasos.

1. Definición de métodos y tecnologías a utilizar.
2. Diseño de la solución.

3. Desarrollo del diseño propuesto.
4. Ensayos y validaciones del funcionamiento.

5.1.1. Análisis de Tecnologías y Métodos para Validaciones

Para poder realizar la propuesta de sistema, hay que determinar métodos y tecnologías necesarias para llevarla a cabo. Existen distintos tipos de métodos para validar los certificados con la Blockchain como el estándar BlockCerts, OpenCerts y la BFA.

La BFA, la cual es una Blockchain que utiliza el protocolo de consenso llamado prueba de autoridad. El sitio web cuenta con una herramienta llamada sello de tiempo 2.0, que permite verificar cuándo se selló un archivo y en qué bloque, permite confirmar que (desde esa fecha) el archivo en cuestión no sufrió modificaciones. Utiliza el mismo método que las anteriores tecnologías, crear un hash a partir de un archivo y almacenar el hash en la Blockchain (BFA, s.f.-b).

Estas tecnologías, permiten demostrar que una secuencia de bits o cualquier tipo de archivo (puede ser un documento o certificado) se mantuvo inalterable a partir del día que se almacenó su hash en la Blockchain o identificador único. Algunas de ellas cuentan con un flujo y trabajo más elaborado, pero al fin todas cumplen con el objetivo de asegurar a los interesados si existió algún cambio en datos del documento o archivo en cuestión. Luego de analizar el funcionamiento de éstas, se pretende realizar una propuesta de sistema, que permita a una institución principalmente a la FCE emitir sus certificados validados y los participantes de los eventos demostrar que sus documentos digitales son totalmente auténticos y correspondiente a su persona (BFA, s.f.-b; BlockCerts, s.f.-a; OpenCerts, 2018).

5.1.2. Análisis de Tecnologías para Desarrollar en Blockchain

Existen diferentes tecnologías para elaborar un sistema que necesita comunicarse con una red de peer to peer y leer o escribir en la Blockchain. Primero hay que definir cómo se almacenará la información en la Blockchain, si hacerlo en una transacción de cualquier Blockchain (como lo hace BlockCerts) o realizar una lógica y almacenarla en una Blockchain que soporte contratos inteligentes. La ventaja de los smart contract es que permiten almacenar información específica como ser el nombre de una institución, sus participantes o cualquier dato que se desee en una sola transacción o invocación a un método del smart contract (mientras que el primer método se necesitará almacenar diferentes datos en diferentes transacciones, o realizar un hash que representará que un documento sigue inalterable).

Por conveniencia de esta investigación se almacenarán datos en los smart contract, permite obtener información directamente de la Blockchain, como ser el nombre, áreas, eventos de la institución. Esta diferencia da lugar a la creación de aplicaciones, por la conveniencia de almacenar datos que pueden ser recuperados y leídos, permitirá tener seguridad par evitar cambios no autorizados y asegurar que la información este disponible, , en comparación con el estándar de BlockCerts que permite hacer lo mismo, pero se necesita el archivo que almacena toda la información del emisor (BlockCerts, s.f.-b; OpenCerts, 2018). A partir de este punto se enumeran las diferentes herramientas necesarias, para el desarrollo de un sistema de validación de certificados usando Blockchain.

Lo más importante es elegir la Blockchain que se utilizará para las pruebas y desarrollo. Existen las Blockchain locales y online, también Blockchains principales y de pruebas. Están las Blockchain de Ethereum, Binance Smart Chain, EOS, como redes principales y online; redes de pruebas como Ropsten, Rinkeby, entre otros. Una de las herramientas es Ganache ¹, que permite levantar una Blockchain local con configuraciones personalizadas. La diferencia entre las Blockchain serán los algoritmos de consenso propios de la red, la

¹<https://www.trufflesuite.com/ganache>

cantidad de nodos, pero para el desarrollo de la propuesta de sistema esas características no son necesarias, mientras que la red se comporte similar a la de una red principal, la cual estas últimas son más seguras. Dependerá seleccionar la Blockchain según las necesidades de la propuesta, si se utiliza los smart contract o otra manera (Truffle, s.f.-a).

Comunicación con Blockchain

Es necesario un método o herramienta para comunicar las transacciones que se deben realizarse en la Blockchain, para ello se precisa de una wallet o billetera que permita enviar las transacciones y comunicarse con los smart contract, en este caso existen diferentes wallet, pero una de las usadas por desarrolladores es Metamask, permite gestionar las cuentas de los usuarios de manera local, conectarse con diferentes Blockchain, recibir y enviar dinero e interactuar con los Smart Contract. Se instala como una extensión del navegador en Google Chrome o Firefox, es usado como puente entre una aplicación y la Blockchain (Dannen, 2017; MetaMask, 2021). La herramienta se puede observar en la figura 5.1.

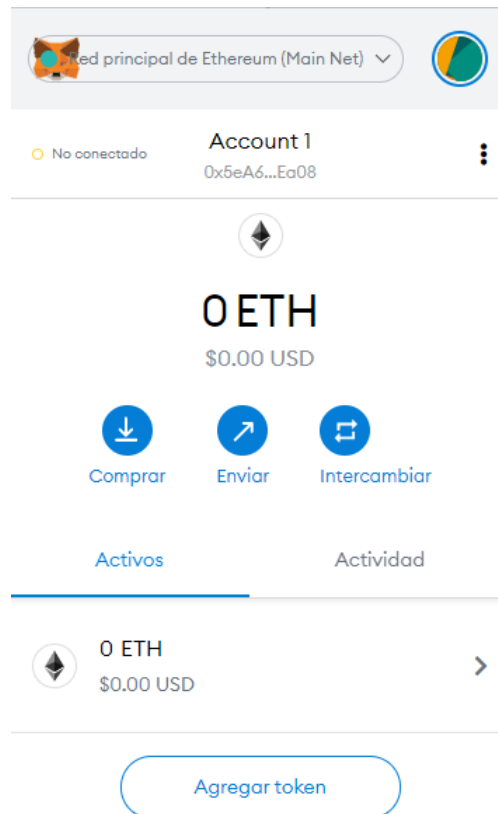


Figura 5.1: MetaMask Plugin, Fuente: Captura de pantalla (producción propia)

Por otro lado para la invocación de los métodos del contrato inteligente, se usa web3.js, que aporta las funcionalidades necesarias para realizar las llamadas de procedimientos remotos a los nodos de la Blockchain, por la cual se pueden leer y escribir datos abstrayendo la comunicación directamente y la creación de las transacciones, mientras la gestión de la cuenta de los usuarios se realiza con MetaMask (Dannen, 2017).

Programación en Blockchain

Se deberá decidir en qué Blockchain se trabajará, porque los smart contract se escriben en un lenguaje específico, en el caso de Ethereum es Solidity, pero existen un gran número de Blockchain que utilizan la misma estructura de Ethereum para generar su propia Blockchain con algunas diferencias, pero en cuanto al lenguaje de programación para el desarrollo de los smart contract es el mismo. En este caso se utilizará Solidity porque existen documentaciones

que describen su uso y es la más usada para el desarrollo de smart contract (Dannen, 2017; Ethereum, s.f.)

Publicación de Smart Contract

Es necesario utilizar una herramienta que permita enviar el código escrito con el lenguaje Solidity a la Blockchain, para ello existen herramientas tales como Remix o Truffle, las que permiten compilar y almacenar el smart contract en una red de prueba local como online. En el caso de Remix es una herramienta que puede usarse de manera online o local. Permite desplegar el código en una Blockchain de prueba local o de prueba online como Ropsten o una Mainnet como Ethereum (Remix, s.f.). Truffle es un entorno de desarrollo, marco de pruebas y canalizados de activos para cadenas de bloques que utilizan la máquina virtual Ethereum (EVM), con el objetivo de facilitar el desarrollo de proyectos con Blockchain, permite desplegar también de manera pública o privada, así mismo provee otras facilidades como gestión de paquetes, automatización de pruebas, entre otras (Truffle, s.f.-b).

Desarrollo del font-end

Se deben seleccionar las herramientas que se necesitan para la creación de la interfaz de los usuarios que pretenden gestionar la información relacionada a la documentación digital y a la institución. Al utilizar las tecnologías mencionadas, es necesario un desarrollo web, donde se puede obtener un servidor central o descentralizado para almacenar datos de usuario, y conectando el front-end con la Blockchain directamente. Esta última es conveniente para evitar manejar datos sensibles porque serían visibles a todo público, puede verificar la existencia de los datos y el servidor central no es necesario para la validación de documentos. Para el desarrollo de un front-end es necesario la utilización de tecnologías como HTML, javascript y CSS o utilizar un framework como ReactJS, VueJS o AngularJS, para desarrollar el sistema de manera

organizada incluyendo todo lo requerido para la interfaz de usuario (Dannen, 2017; Educacionit, s.f.).

Para la creación de un sistema de validación de documentos digitales es conveniente utilizar los puntos importantes del estándar BlockCert y la utilización de los smart contract como lo hace OpenCert, también existen códigos fuentes que pueden ser reutilizados. Uno de ellos son las herramientas que provee la BFA, por ejemplo la interfaz para que un usuario valide o selle un documento, este código se encuentra escrito en VueJS, por lo que se optará su uso para el desarrollo de la vista del usuario. Cabe aclarar que se puede realizar con cualquier otro framework (esto se puede demostrar encontrando los diferentes proyectos existentes con diferentes tecnologías usadas o en cursos de desarrollo de Blockchain con ReactJS o AngularJs (BFA, s.f.-b).

Criptografía

La función hash es necesaria como parte de cualquier proyecto de documentos digitales visto, como lo usan BlockCert y OpenCert, también las propias Blockchain la usan para crear las direcciones de los bloques. Asimismo la BFA usa esta función hash. Existen diferentes tipos como MD5, SHA y sus variaciones por lo general en las Blockchains y en los sistemas o aplicaciones utilizan SHA-256, por su uso extendido se elige esta última (Satoh y Inoue, 2007).

5.2. Análisis de la Propuesta de Sistema de Validación de Documentos Digitales

Para que el sistema permita validar documentos digitales sin almacenar la información contenida en ella y evitar acceder a datos sensibles de un individuo, se utilizarán los métodos de los estándares o sistemas (BlockCerts y OpenCerts). En otras palabras, genera un hash a partir de una secuencias de

caracteres como valores de entrada y se obtiene una salida de longitud fija, de este modo se aumenta la dificultad para conocer el contenido del documento y se genera un identificador único, el hash asegura que el documento no fue modificado y puede ser verificado usando nuevamente el mismo archivo original (BlockCerts, s.f.-b; Jirgensons y Kapenieks, 2018). Luego se necesitará información adicional, como ser el día en el cual el documento se creó, o si esta relacionado a algún evento de la organización que la emitió, nombre de la organización, el nombre del Área o Departamento encargado de generar el documento y los datos extras que se necesiten. Por otro lado permitir a los encargados de emitir los certificados poder cambiar los estados de los documentos, por ejemplo, si pasado un tiempo un documento ya no tiene validez, se puede cambiar el estado del documento a no válido, o darle una fecha de expiración si se necesitara. Con esta información se creará un smart contract que permita almacenar estos datos en la Blockchain, manteniéndolos inmutable a menos que se permita roles a usuarios específicos, con permisos para modificarlos.

5.2.1. Bases del Sistema

La base del sistema es mantener los documentos digitales o el hashes inalterables, para asegurar que el contenido no cambió.

En primer lugar es gestionar los tipos de documentos que se manejan, definir cómo se almacenarán y asignarán el/los responsable/s de hacerlo, con la posibilidad de cambiar algunas características de los documentos digitales (OpenCerts, 2018). Por ejemplo, manejar estados de los documentos, que permita a los interesados de validar si el documento fue emitido por la universidad o una institución. Los estados sirven para que el interesado pueda observar si la documentación venció o si está en algún estado específico. Los datos usados por los estándares como BlockCert y OpenCert son el hash del documento, que una vez almacenado no se permite cambiarlo, cambiar estados, definir una fecha de creación y una fecha de expiración.

Los estándares almacenan información relacionada con el lugar donde se emitió el documento (OpenCerts, 2018), esto se logra agregando información como el evento que lanzó el certificado, por ejemplo, un cursado, un examen, etc, cualquier situación que pueda generar una documentación digital (Lab, 2016). El evento puede suceder en un lapso de tiempo o puede ser indeterminado, por ejemplo, un evento de un curso que su duración son 3 días, o un evento que inicia una fecha pero no se conoce cuando finalizará.

Los eventos suceden en algún sitio o área, al observar la Facultad de Ciencias Económicas (FCE) de la Universidad Nacional de Misiones (UNaM) la Secretaría de Extensión es la encargada de realizar los eventos o actividades como cursos, o actividades externas, por ende, el área encargada es la Secretaría de Extensión (ver anexo A.3).

Se podrá almacenar información como el nombre y las áreas de la institución, con estos datos se obtiene qué área almacenó el documento digital, conocer el evento que se generó en una fecha determinada o indeterminada, si el documento puede caducar, y por último, si el documento fue generado por la institución, dando una validez de que existe en la Blockchain.

Estas informaciones, se obtuvieron de relevamientos que explican el funcionamiento de las aplicaciones para certificados digitales y las informaciones que son usadas frecuentemente. Por ejemplo, el estándar BlockCerts que almacena el hash del documento e información relacionada al emisor (BlockCerts, s.f.-a).

5.2.2. Solución Propuesta

Para validar los documentos digitales de la Secretaría de Extensión de la FCE de la UNaM, se propone crear un sistema, basado en los protocolos definidos. Para la validación de documentos se utilizará tecnología Blockchain, para ello se realizará un smart contract que tendrá la lógica necesaria para almacenar datos relacionados al documento, sin exponer datos sensibles de los intervinientes, y permitir la validación del documento que permaneció íntegro.

También gestionar información de los documentos y cambiar estados futuros de ellos, almacenar el documento en la Blockchain permitiría crear una relación única entre una cuenta y la información guardada, habilita seguir consultando los datos en el futuro aunque la facultad no exista.

CAPÍTULO 6

Desarrollo del Sistema Propuesto

En este capítulo se desarrolla la solución al problema de la falta de validación de documentos digitales emitidos por la Secretaría de Extensión de la FCE de la UNaM. Se realiza el diseño teórico que se pretende aplicar y se define su funcionamiento y las relaciones entre las herramientas y tecnologías en la implementación.

6.1. Tecnologías y Herramientas Seleccionadas

Las tecnologías y herramientas seleccionadas para el desarrollo de la propuesta son:

1. La red de prueba llamada Ropsten, esta Blockchain de prueba permite publicar los smart contract y la emisión de su token de prueba es de forma gratuita, cuenta con pocos nodos pero el objetivo de su uso es hacer pruebas en esta red, su comportamiento es similar a otra Blockchain con maquina virtual de Ethereum (Dannen, 2017).
2. Solidity como lenguaje de programación para los contratos inteligentes (Bragagnolo, Rocha, Denker, y Ducasse, 2018; Dannen, 2017).
3. Node JS con su gestor de paquetes para instalar las librerías necesarias.
4. Web3.js una librería de javascript que permite conectarse a la Blockchain de una manera más rápida y abstrayendo la complejidad de la comunicación con la Blockchain (Dannen, 2017).
5. SHA-256 un algoritmo de hash, pero también existen diferentes librerías de javascript que permite generar un hash a partir de caracteres.
6. Vue.js un framework de javascript, para facilitar el desarrollo del front-end, reutilizando partes de otros proyectos open sources.(VUEJS, s.f.)
7. Para la publicación de los contratos inteligente se usará Remix, es una de las tecnologías ya mencionadas y su uso es intuitivo (Remix, s.f.).

Estas tecnologías permitirán crear el sistema para validar los documentos de la Secretaría de Extensión de la FCE de la UNaM, dando el beneficio a terceros de verificar que el contenido de un documento es correcto y no fue alterado (Brys, 2019).

6.2. Diseño de la Lógica

A continuación se muestra un diagrama para comprender los datos esenciales para el diseño del sistema.

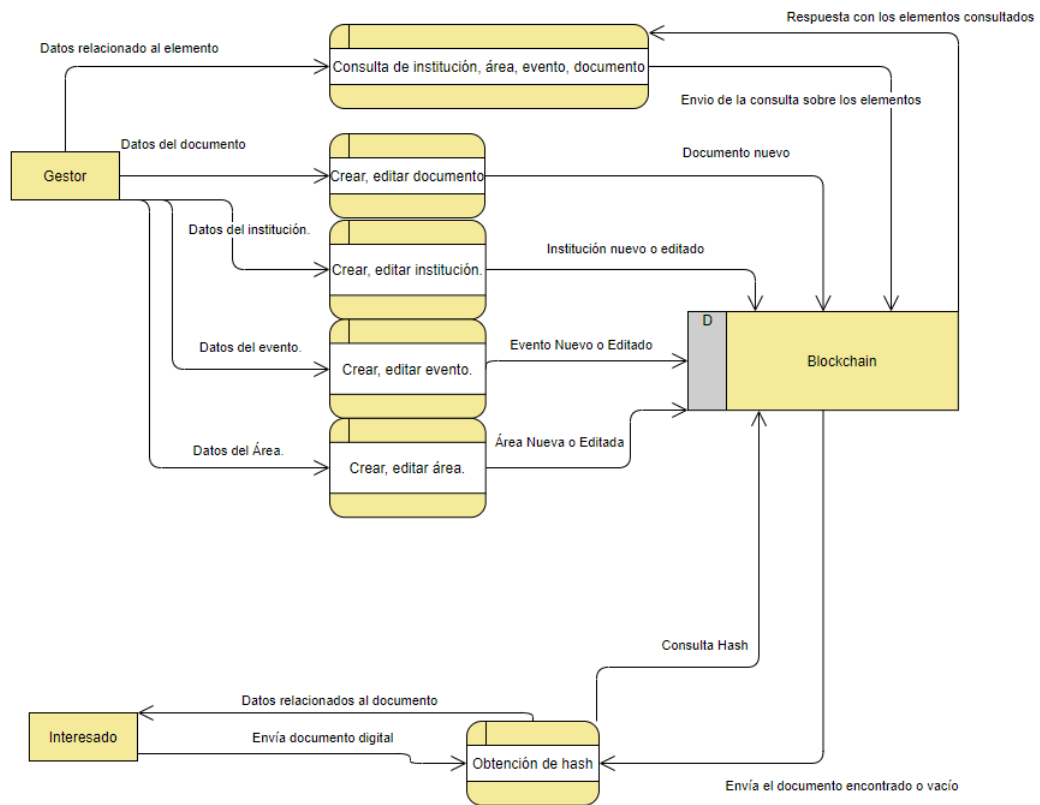


Figura 6.1: Diagrama de Flujos de datos, Fuente: producción propia.

En la figura 6.1 se observa el flujo de los datos que se manejan en una validación de certificados, principalmente se almacenan los datos en la Blockchain y son leídos por las personas interesadas. Los responsables de sus áreas deben almacenar los datos en la Blockchain, esto permitirá validar que los documentos almacenados solamente serán cargados por personal autorizado.

Los interesados en el diagrama representan a cualquier individuo (como el propio dueño del documento digital, o una autoridad de la organización o persona externa de ella), se puede obtener datos relacionados con el documento quién fue el emisor o si el documento fue realmente subido por la organización. Pero para poder realizar tal verificación debe tener el documento digital en su posesión; de esta forma comprobar si el contenido del documento no fue modificado y que fue emitido por la organización.

Los gestores son los encargados de almacenar la información y serían las personas responsables de los documentos emitidos.

6.3. Diseño de Sistema

En la sección 5.2.1 se explica de manera genérica, los datos necesarios para validar un documento digital en base a los estándares BlockCert y OpenCert. Para ello se diseña un diagrama de la figura 6.2 para la propuesta del sistema para validación de documentos:

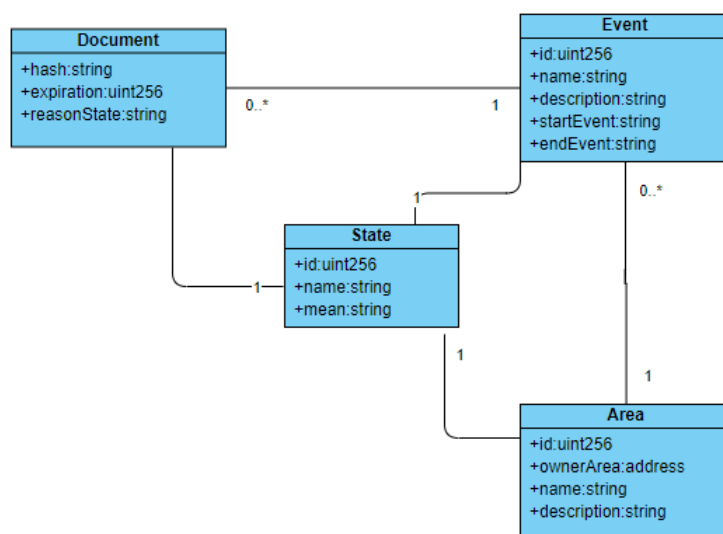


Figura 6.2: Diagrama de domino, Fuente: producción propia.

En la figura 6.2 se puede observar la propuesta de construcción del sistema basándose en el estándar de BlockCerts donde se almacenan los hashes del documento e información relacionada a ella (BlockCerts, s.f.-a, s.f.-b; CriptoMonedasTV, 2018), el diagrama muestra un área que es la responsable de emitir los certificados, sin revelar datos de personas o contenido del documento. Por lo tanto el objetivo de no incluir datos sensibles quedan cubiertos.

Se analizaron los requerimientos de la universidad y en conjunto a la entrevista realizada (ver anexo A.3) para poder diseñar la lógica del sistema. Se incluyeron administradores o los propietarios (owner) de las áreas, serían los responsables de almacenar los hashes de documentos que se generen en los Eventos de las Áreas, cuando se habla de los eventos, se los trata como genéricos, un evento puede ser una clase, una charla, un congreso o inclusive

una determinada hora del día, de esta manera los documentos pueden estar relacionados a los eventos que ocurren.

La construcción, el almacenamiento y lectura de datos de una aplicación en la Blockchain es diferente a una aplicación centralizada, por ejemplo, los datos almacenados no se consultan mediante lenguaje SQL, tampoco se puede agregar nuevas variables una vez almacenado en la Blockchain, en cambio en una base de datos centralizada si se pueden agregar nuevos atributos (Dannen, 2017; Mayor, 2017).

En resumen el dominio comprende áreas, eventos, estados y los documentos cada uno de ellos almacena datos representativos y básicos para validar la existencia de un documento digital en la Blockchain.

6.3.1. Estructura del Smart Contract

Definir los datos necesarios para la propuesta del sistema, permite dar forma a la estructura del smart contract, ya que éste no cambiará una vez que sea publicado en la Blockchain, pero no significa que se pueda publicar otras versiones. La propuesta es utilizar una Blockchain de prueba para realizar todos los intentos necesarios hasta obtener el funcionamiento del sistema.

La estructura del smart contract será como se refleja en la figura 6.3, a continuación se explican cada parte de ella:

Los atributos (de la figura 6.3) que almacenarán los datos necesarios serán los siguientes :

- **organisation: string**, se almacenará el nombre de la organización responsable de las áreas, eventos y todos los documento digitales.
- **ownerOrg: address**, la dirección de la cuenta del único responsable para manejar todos los datos del smart contract.



Figura 6.3: Smart Contract estructura, Fuente: producción propia.

- **states: State[]**, los posibles estados que maneja el sistema, como ser estado revocado, el cual el estándar BlockCert y OpenCert lo usan (Block-Certs, s.f.-a; OpenCerts, 2018).
- **ownerArea:mapping(address =>uint256[])**, son el conjunto de direcciones que están encargadas de manejar una o varias áreas específicas. Estos owner deberían ser agregados por el owner de la organización para evitar que un individuo desconocido tenga el poder de modificar datos.
- **areas: Area[]**, el conjunto de áreas de la organización.
- **events: Event[]**, los eventos que pueden surgir en un área específica y encargados de generar los documentos digitales.
- **documents: mapping(string =>Document)**, los hashes de documentos asociado a la información del documento.

La notación que se utilizó para el diagrama 6.3 para los tipos de datos son de la misma manera en la que se definen en Solidity, como por ejemplo la definición de `mapping(string =>Document)`. Esto sirve para definir una variable que recibe como índice un string por ejemplo “A” referencia un documento almacenado en la Blockchain. El string se creó como índice para que el usuario

pueda usar cualquier tipo de algoritmo de hash y el sistema permita almacenar en la Blockchain sin tomar en cuenta la función hash , esta idea es usada por los estándares ya mencionados (BlockCerts, s.f.-a, s.f.-b; OpenCerts, 2018).

Los métodos para realizar cambios en los datos del sistema:

- *setOrganisation(org:string)*, el método permite cambiar el nombre de la organización, sólo debe ser ejecutado por el propietario de la organización.
- *editOwnerOrg(newOwner:address)*, permite cambiar al único usuario que podrá ejecutar todos los métodos.
- *addState(name:string, mean:string)*, name es el nombre del estado y mean el significado o para qué se usaría. El método agrega un nuevo estado que podrá ser usado por las áreas, eventos y documentos.
- *editState(id:uint256, name:string, mean:string)*, edita el nombre y el significado del estado.
- *addArea(ownerArea:address, name:string, description:string)*, crea un nuevo estado y pasa por parámetros la dirección del dueño o responsable del área a crear, el nombre, y una descripción como dato extra. El único que puede ejecutar este método es la dirección que concuerde con la ownerOrg
- *editArea(id_area:uint256, name:string description:string, id_state:uint256)*, el id que representa el área a editar, los datos a modificar como el nombre, la descripción y el estado actual.
- *changeOwnerArea(id_area:uint256, newOwner:address)*, permite cambiar el propietario de un área específica.
- *addEventFull(area_id:uint256, name:string, description:string, startDate:string, endDate:string)*, agrega un evento a la Blockchain relacionándolo, con un área específica.

- *editEventFull(event_id:uint256, name:string, description:string, startDate:string, endDate:string, area_id:uint256, state_id:uint256)*, edita la mayoría de los atributos de un evento.
- *addDocumentEvent(idHash:string, event_id:uint256, state_id:uint256, reasonState:string, expiration:uint256)*, crea un documento nuevo con relación a un evento particular y una fecha de vencimiento para el documento digital. La fecha es representada por un valor numérico o Marca de tiempo (TimeStamp).
- *addAllDocumentsEvent(hashid:string[], event_id:uint256, state_id:uint256, reasonState:string, expiration:uint256)*, permite almacenar muchos documentos enviando un array de hashes y los atributos iguales que tendrán los documentos, por ejemplo todos los documentos serán del mismo evento, tendrán el mismo estado y fecha de vencimiento.
- *editAllDocumentsEvent(hashid:string[], event_id:uint256, state_id:uint256, reasonState:string, expiration:uint256)*, edita todos los documentos que se encuentran en el array de hashes.
- *changeStateDocument(idHash:string, state_id:uint256, reasonState:string)*, cambia el estado de un solo documento.
- *newVersionDocument(idHash_old:string, idHash_new:string)*, se le asigna una nueva versión a un documento antiguo, idHash_old es el documento antiguo o idHash_new es el hash del nuevo documento, esto sirve para mantener versiones de un solo documento.

Y por último, los métodos para lectura de los datos son:

- *getOrganisation():(string)*, retorna un string que es el nombre de la organización.
- *getOwnerOrg(): (address)*, devuelve la dirección del dueño o propietario, el cual podrá ejecutar todos los métodos.

- *getState(id:uint256) : (id: uint256, name:string, mean:string)*, retorno los atributos del estado que tenga el id pasado por un parámetro.
- *getLengthStates() : (uint256)*, obtiene la cantidad de estados almacenados.
- *getAreaOfOwner(id:address, area_index:uint256) : (uint256)*, obtiene el id de un área de un propietario de área.
- *getLengthAreaOfOwner(id:address) : (uint256)*, obtiene la longitud o la cantidad de areas de un propietario de área.
- *getArea(id: uint256) : (id:uint256,owner : address,name : string,description : string,state_id : uint256,cantEvents : uint256)*, obtiene un área a partir de un id.
- *getLengthAreas() : (uint256)*, obtiene la cantidad de áreas que hay almacenado en la Blockchain.
- *getLengthEventsOfArea(id_area: uint256) : (uint256)*, la cantidad de eventos que están relacionado a un área.
- *getEventOfArea(id_area : uint256, id_event_index : uint256) : (uint256)*, obtiene el id un evento relacionado a un área.
- *getAllEventsOfArea(id_area : uint256) : (uint256[])*, trae todos los id de eventos de un área.
- *getEvent(id : uint256) : (id : uint256,name : string ,description : string,state_id : uint256,area_id : uint256,startEvent : string,endEvent : string)*, obtiene los atributos de un evento.
- *getCantDocumentEvent(id: uint256) : (uint256)*, la cantidad de documentos relacionado a un evento.
- *getLengthEvents() : (uint256)*, cantidad de eventos que existen almacenados.

- *getDocument(idHash : string) : (idHash : string, state_id : uint256, event_id : uint256, reasonState : string, expiration : uint256, newDocument: string)*, obtiene todo los atributos de un documento a partir de su hash.
- *getDocumentsOfEvent(id_event: uint256) : (string[])*, obtiene todos los hashes de los documentos relacionado a un evento.
- *checkDocument(idHash: string) : (bool)*, devuelve true si el hash del documento esta almacenado en la Blockchain.
- *checkDocuments(idHashes : string[]) : (bool[])*, a partir de un array de hashes de documentos devuelve en el mismo orden true o false dependiendo si existe o no en la Blockchain respectivamente.

6.3.2. Roles y Permisos

En la propuesta de sistema se utiliza un smart contract para el código almacenado en la Blockchain. Para ello también es necesario definir los responsables de gestionar los datos. Se establecen dos niveles, el primero es un usuario que pueda gestionar todo los datos, como un administrador. Por ende, se debe definir el rol de un usuario o propietario del smart contract, que sea el único que tenga el poder de crear nuevas áreas, cambiar en nombre a la organización, editar los datos de las áreas, agregar o quitar responsables de cada área. El otro nivel son los usuarios encargados de una o muchas áreas, los cuales tienen la responsabilidad de gestionar los documentos y eventos de un área específica, para ello el administrador o el propietario del smart contract decide quiénes son los operarios o encargados de crear los documentos digitales y almacenarlos en la Blockchain mediante su hash. Esto es importante para evitar que usuarios externos a la organización controlen los documentos emitidos por la entidad. Para ello se crean los roles o niveles de seguridad. También habrá un usuario que no tiene el poder de modificar los datos almacenados, simplemente podrá consultar acerca de los documentos y validaciones, es el individuo que requiere conocer la validez del documento y comprobar que es emitido por la organización. En resumen se definen dos niveles: nivel de

administrador es el encargado y responsable de todo los datos almacenados, por otro lado un nivel de encargados de áreas.

Estructura de permisos

Todos los usuarios tendrán acceso a leer sus datos, por eso desde el inicio se evita almacenar datos sensibles relacionados a los dueños de los documentos, mientras que los datos de la organización son públicos y pueden ser almacenados. En esta sección se definirán qué comportamiento tendrán los roles y qué permisos según los niveles de usuarios. Al analizar los datos sobre los métodos y las necesidades a satisfacer, se parte de tres niveles:

1. Nivel público: el usuario podrá consultar datos almacenados, (cómo verificar si un documento existe en la Blockchain).
2. Nivel de propietario de área: son los encargados de almacenar y crear los eventos relacionado a sus áreas, también de gestionar los documentos digitales, cambiando sus estados, agregando nuevas versiones.
3. Nivel de propietario del smart contract: es la única dirección que tiene permiso de ejecutar todos los métodos. Puede cambiar el nombre de la organización, asignar nuevos propietarios de áreas, crear áreas, y hacer lo mismo que los demás niveles.

6.4. Desarrollo del Smart Contract

A continuación se detallan las herramientas y las manera de utilizarlas para el desarrollo del Smart Contract.

6.4.1. Instalación de Metamask

Como primer paso, hay que instalar el plugin de Metamask de la página web oficial el navegador que se usará de prueba (en este proceso, es Google Chrome) tal como se muestra en la figura 6.4.

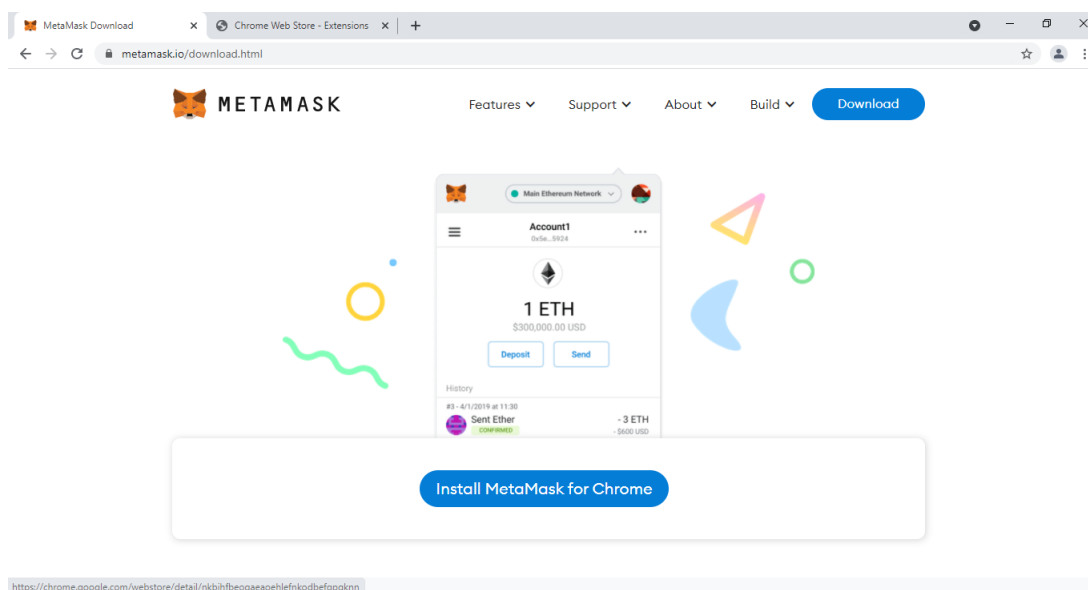


Figura 6.4: Página para descargar Metamask, Fuente: captura de pantalla (página oficial de metamask).

Se agrega el plugin al navegador como se ve en la figura 6.5, luego se abrirá una nueva pestaña o en caso que no aparezca deberá buscar Metamask en sus extensiones del navegador para hacer clic donde este le redirigirá a una nueva pestaña para crear o importar su wallet. Los pasos son hacer clic en Get Stated visto en la figura 6.6.

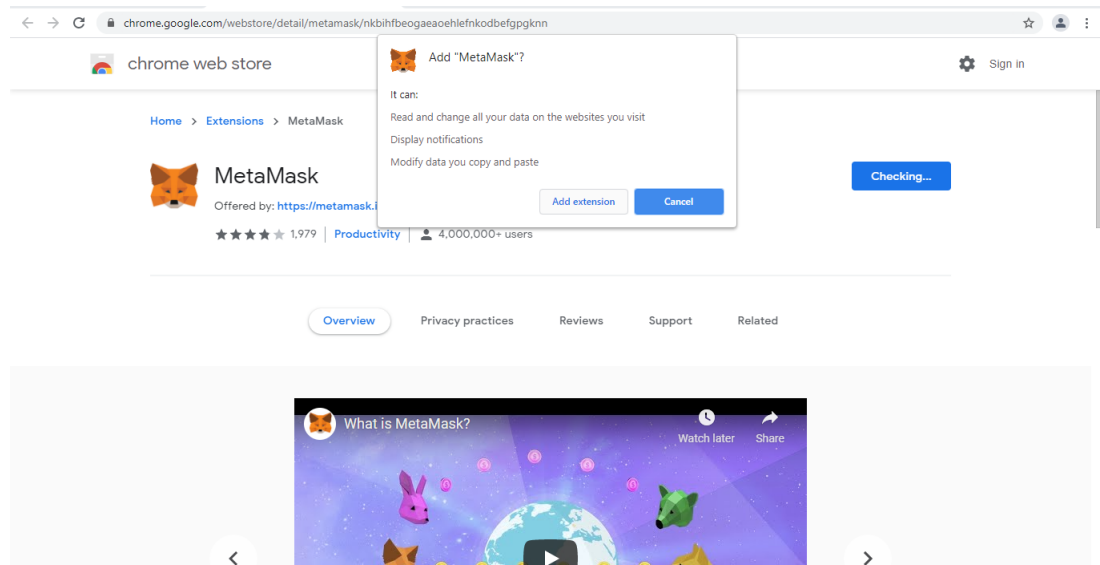


Figura 6.5: Agregando el plugin, Fuente: captura de pantalla (página plugins chrome web store).

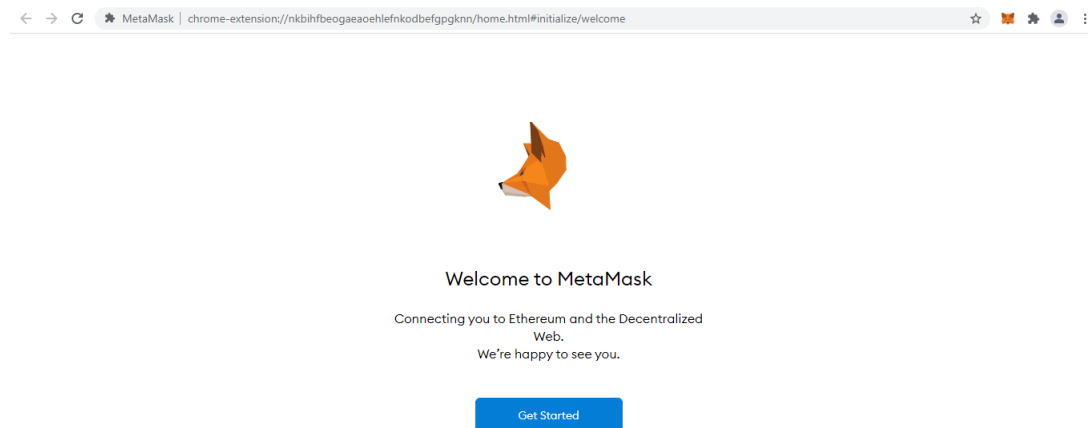


Figura 6.6: Inicio para crear o importar una wallet, Fuente: captura de pantalla.

Como siguiente paso hay que crear una wallet. Esto es importante para poder interactuar con el sistema (específicamente con los smart contract). Leer los términos y condiciones si están de acuerdo aceptarlos, el siguiente paso es crear una contraseña, esta sirve para acceder a Metamask en el computador local, evita que otro usuario pueda acceder si no conoce la contraseña y agrega un nivel más de seguridad, ya que se pueden realizar diferentes operaciones.

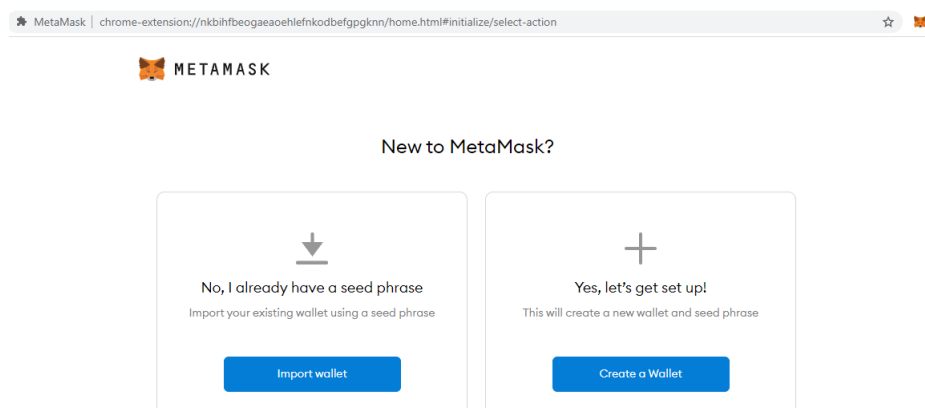


Figura 6.7: Menú crear wallet, Fuente: captura de pantalla.

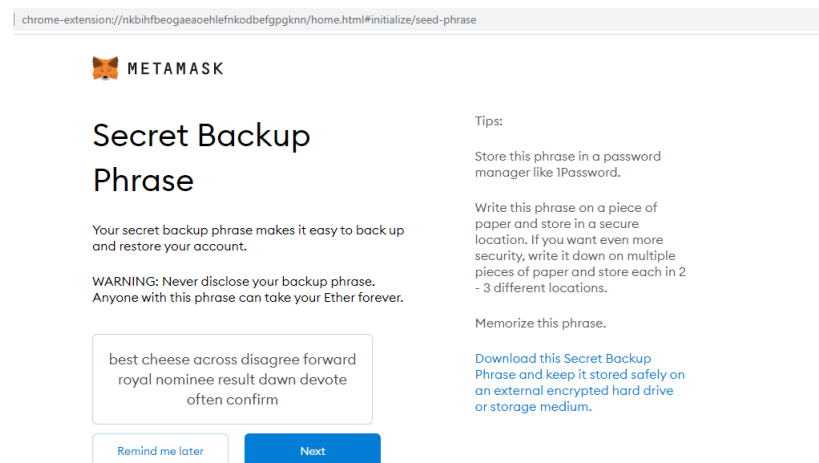


Figura 6.8: Frases semillas, Fuente: captura de pantalla.

Un vez ingresada la contraseña se mostrará su frase semilla de su wallet, Tal como se observa en la figura 6.8, la frase se muestra de modo didáctico pero la misma nunca debe ser compartida bajo ningún término, ya que el usuario que conozca la frase semilla podrá abrir la wallet desde otro metamask y hacer lo que desee con ella, la única manera que otro usuario no pueda intervenir o manejar su wallet es que no conozca la frase semilla, por ello es muy importante guardarla en un sitio que solo la personas dueña de la frase semilla conozca, a partir de esta frase se genera la llave privada. La frase semilla generada para esta investigación será rechazada y se usarán otras. Finalizada la creación de la wallet, se podrá fijar en la parte derecha como se muestra en la figura 6.9

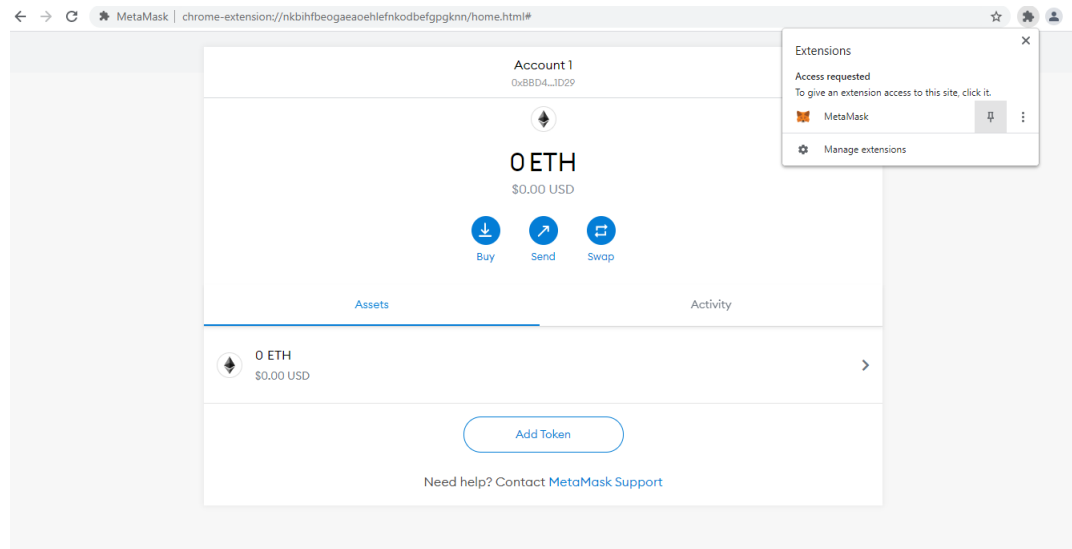


Figura 6.9: Creación de la wallet finalizada, Fuente: captura de pantalla.

6.4.2. Desarrollo del Smart Contract

Se utilizó Remix para la edición del código fuente escrito en Solidity, este programa permite compilar el código para probarlo de manera local y también es una herramienta para publicarlo de manera online en otras Blockchain. En este caso se publicó en Ropsten.

El código del smart contract se desarrolló siguiendo las definiciones de la sección 6.3, se encuentra en el anexo A.4 En él se definieron los métodos ya mencionados y se agregaron reglas para que métodos en particulares puedan ser ejecutados solo por el propietario del smart contract y otros solo para los propietarios de las áreas.

El código desarrollado, cumple con los principios descritos en otras secciones, y sigue algunas pautas de los estándares de certificados ya mencionados como BlockCerts y OpenCerts. Algunas pautas son: almacenar información del emisor, almacenar el hash del documento, no relacionar datos personales del propietario del documento digital.

Se desarrolla el código dentro del archivo “validationssystem.sol”, que tiene toda la lógica del sistema con los comportamientos que se pueden realizar. Una

vez que el código está finalizado, hay que compilarlo, la figura 6.10 muestra un ejemplo.

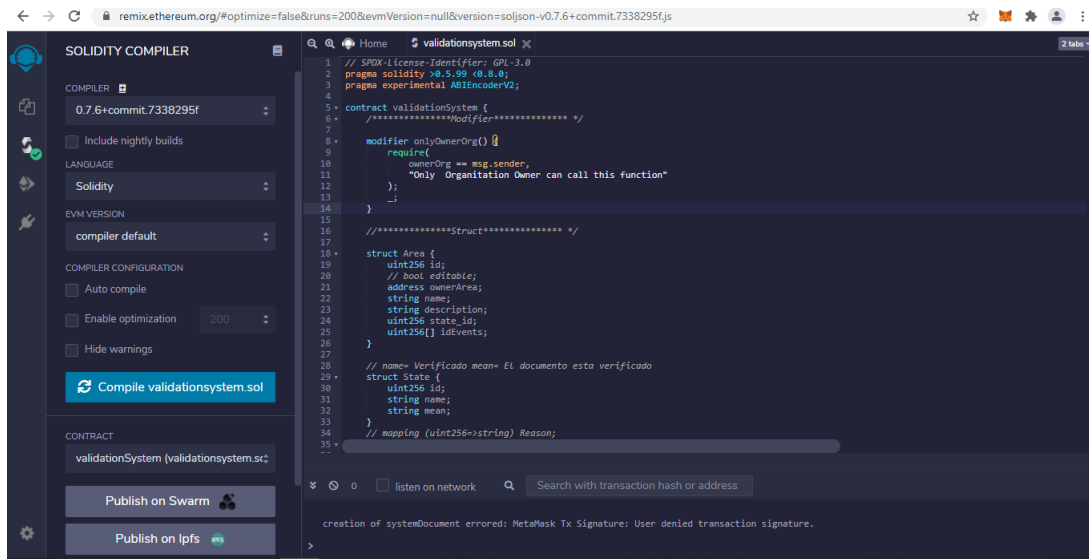


Figura 6.10: Vista como compilar el contrato, Fuente: captura de pantalla (página de remix).

Por otra parte es un requisito publicar el smart contract en alguna Blockchain, para ello hay que ejecutar Metamask y seleccionar la red de Ropsten como se muestra en la figura 6.11. Luego es necesario conseguir la criptomoneda nativa de la Blockchain, para ello hay que acceder a los sitios web denominados como Grifo o Faucet, estos sitios web facilitan obtener las criptomonedas nativas. Para Ropsten se puede acceder al sitio web proporcionado por [Metamask](#) o proporcionado por [Ropsten](#) ambos sirven para obtener las criptomonedas para la red de prueba.

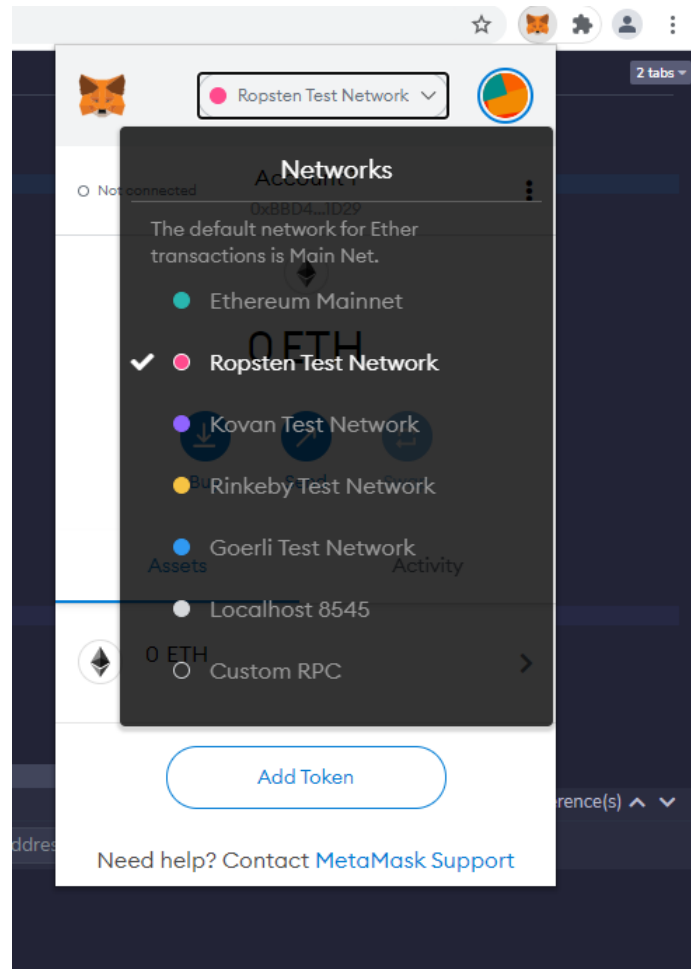


Figura 6.11: Seleccionando la Red de Ropsten, Fuente: captura de pantalla.

En este caso se utilizó el sitio web de la figura 6.12 haciendo clic en el botón verde, suma 1 ETH a la cuenta.

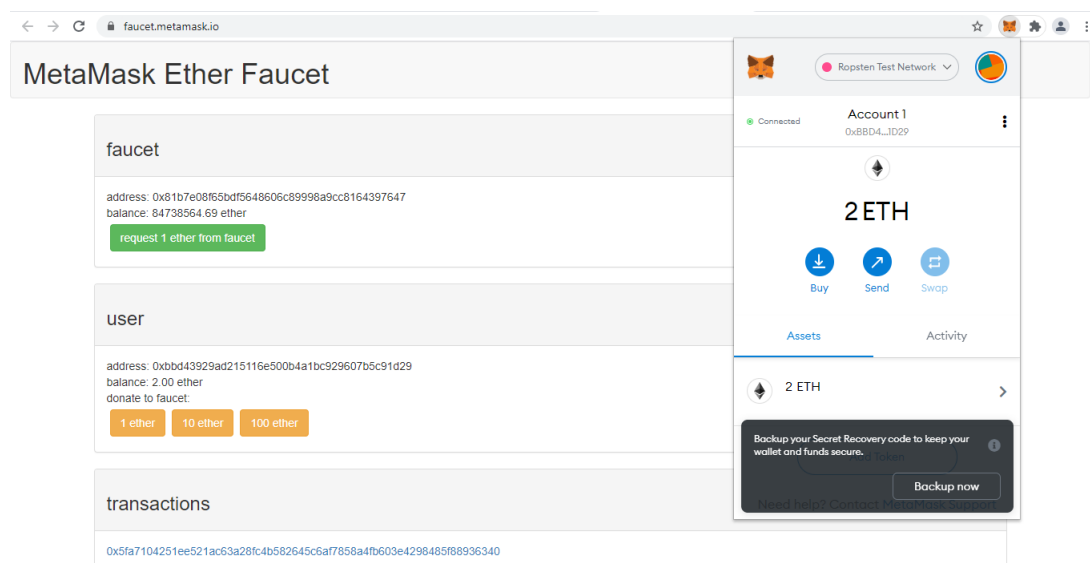


Figura 6.12: Grifo de MetaMask, Fuente: captura de pantalla (página de metamask faucet).

A partir de ahora, se cuentan con todas las condiciones para publicar el código en la Blockchain de Ropsten. Para ello en el menú izquierdo de Remix seleccionar la opción de deploy, dentro de ella ir en la opción de “environment” y seleccionar “inject web3”, lo que abrirá Metamask requiriendo conectar la wallet con el sistema de Remix. Luego hacer clic al botón de color naranja (deploy) que se observa en la Figura 6.13

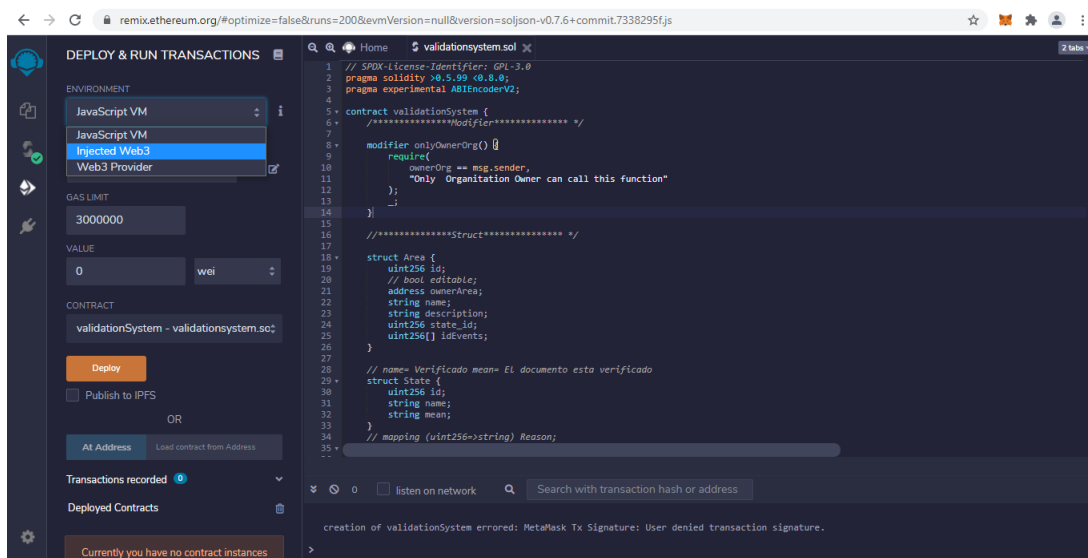


Figura 6.13: Menú deploy Remix, Fuente: captura de pantalla (página remix).

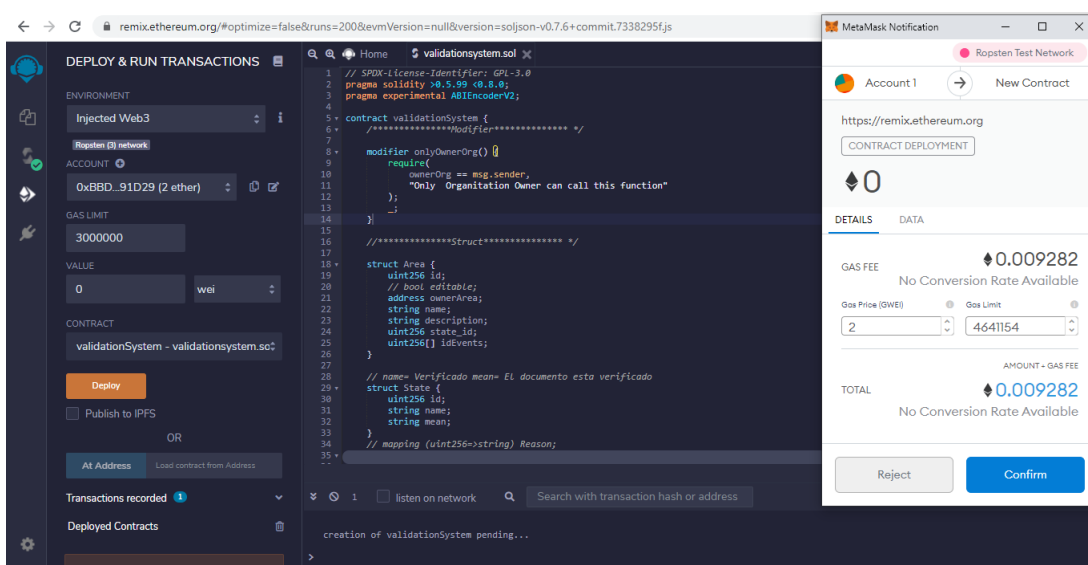


Figura 6.14: Deploy Contrato Inteligente, Fuente: captura de pantalla (página remix).

Se abrirá una ventana de Metamask (Figura 6.14), que pedirá la confirmación para gastar una cantidad de ETH, que es la criptomoneda que se obtuvo

mediante la pagina web de grifo o faucet de Ropsten, la cual se utiliza para pagar las transacciones en la Blockchain. Se confirma la transacción, y quedará en pendientes hasta ejecutarse, una vez finalizada el smart contract estará en la Blockchain lista para usarla. En la figura 6.15 se muestran los métodos del contrato que se pueden ejecutar en la Blockchain, la interfaz Remix también sirve para llamar a los diferentes métodos creados.

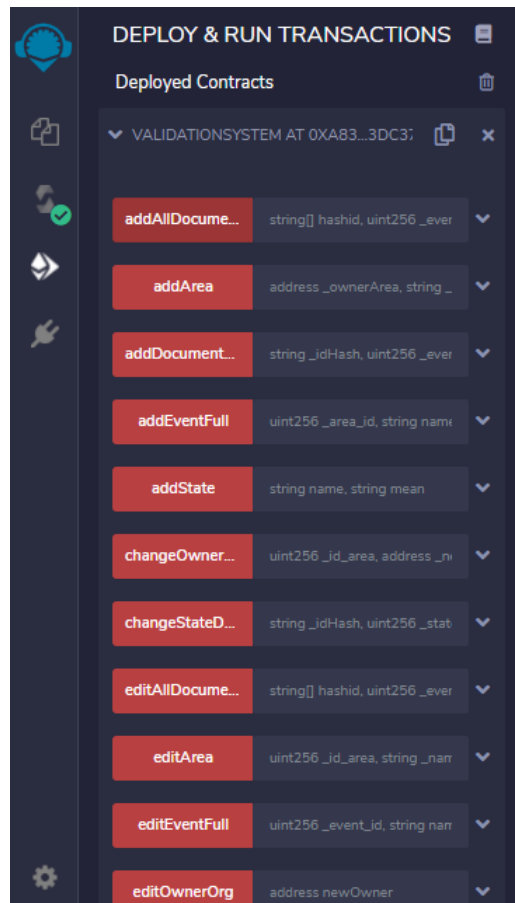


Figura 6.15: Contrato en la Blockchain de Ropsten, Fuente: captura de pantalla (página remix).

6.5. Desarrollo de la Interfaz de Usuario

El desarrollo se realizó con el sistema operativo Windows 10 Home, pero las herramientas pueden usarse en distribuciones de GNU/Linux. Como primer paso se requiere instalar NODE JS desde su sitio oficial; también se utilizará Visual Studio Code como editor de texto.

Otro punto, es instalar VUE JS CLI para facilitar la instalación de todos los paquetes y tener una estructura más organizada (VUEJS, 2019), para ello previamente se requiere instalar NODE JS, en cmd o la consola de comandos del Visual Studio Code se ejecuta el comando “npm install -g @vue/cli”, para instalarlo de manera global, en la figura 6.16 se realiza utilizando Visual Studio Code.

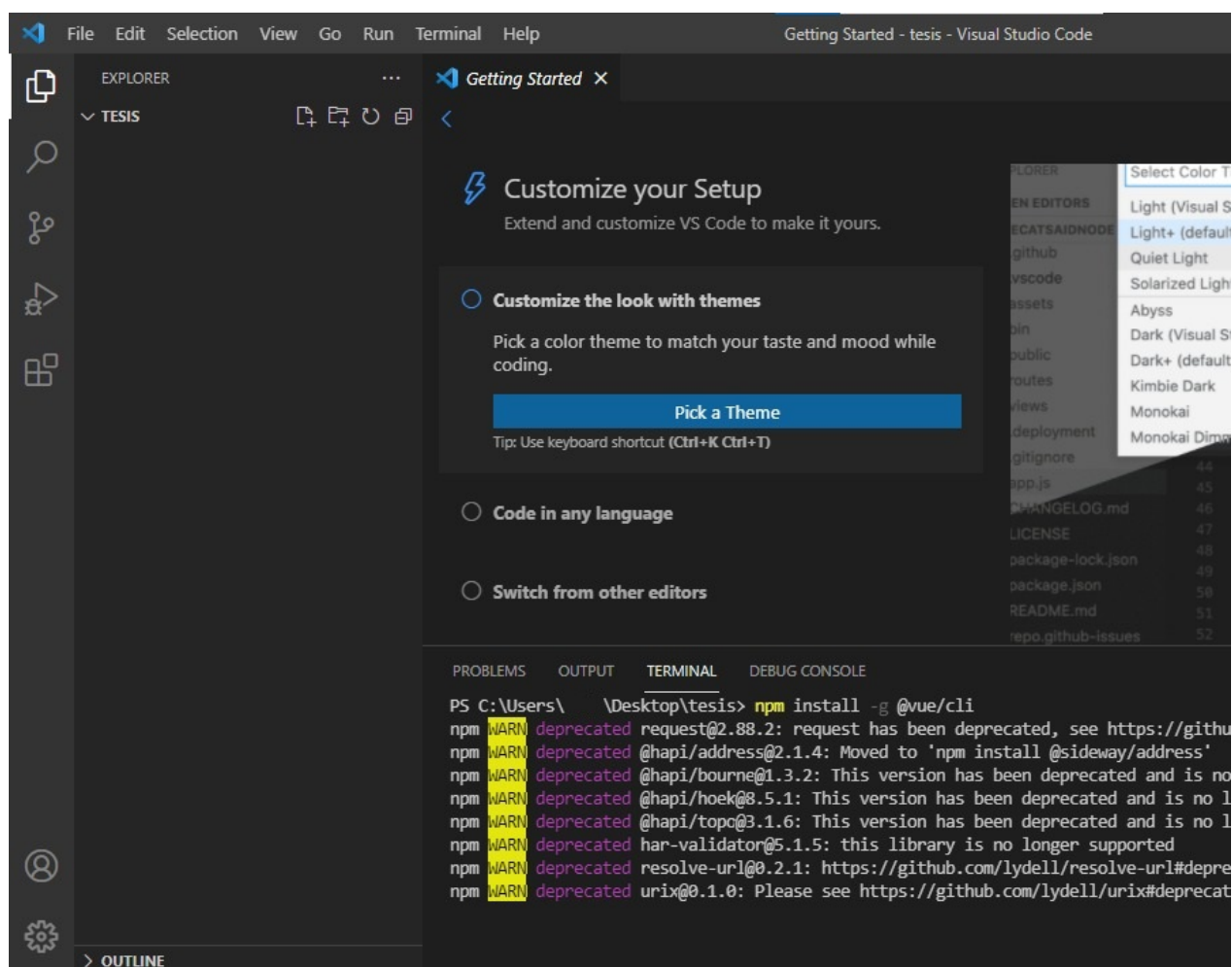


Figura 6.16: Comando para instalar VUE CLI, Fuente: captura de pantalla.

Luego en la consola de comandos hay que ubicarse en el directorio que se desea instalar el proyecto. Una vez hecho, ejecutar el comando “vue create nombre_proyecto”, en este caso se usó “vue create validation_system” esto abre unas opciones de configuraciones en consola, para el proyecto se utiliza router, vuex, babel; seleccionados estos ítems continuar con la instalación, en la figura 6.17 se muestra las opciones necesarias para la creación del proyecto.

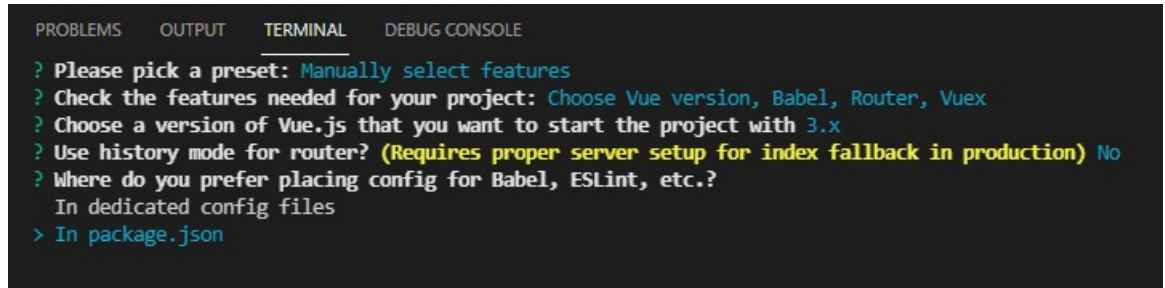


Figura 6.17: Configuración de proyecto, Fuente: captura de pantalla.

Como siguiente paso se debe ingresar dentro del directorio del proyecto creado en el cual se usarán dos librerías ya mencionadas (web3 (Web3js, 2016), y sha256 (Sato y Inoue, 2007)), también se usará un componente que facilita la creación y carga de los select múltiples, mediante VUE. La instalación de estos se realiza con los comandos mostrados en la figura 6.18

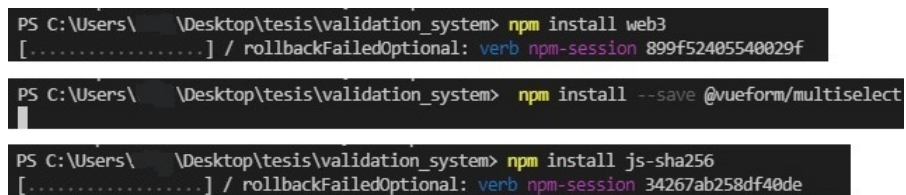


Figura 6.18: Instalación de librerías y componente, Fuente: captura de pantalla.

A partir de este punto comienza el desarrollo de la interfaz gráfica con las librerías y herramientas instaladas.

6.5.1. Desarrollo de la Vista del Sistema

Inicialmente, con la carpeta del proyecto generada se crean unos archivos extras, en este caso dentro de la ruta del proyecto “nombre_proyecto/src/” se crea manualmente una carpeta con el nombre app donde se almacenarán archivos que permitirán conectar con la Blockchain.

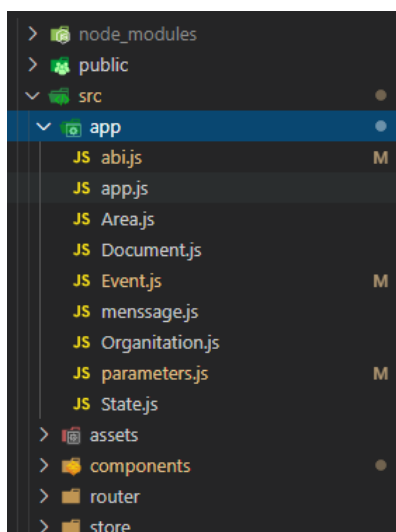


Figura 6.19: Archivos para conexión con la Blockchain, Fuente: captura de pantalla.

El archivo `abi.js` define todos los métodos o funciones que se pueden usar en el smart contract creado, y para eso hay que dirigirse a Remix, compilar nuevamente el código del smart contract y copiar el ABI creado. Por ejemplo, en la figura 6.20 se resalta con un círculo rojo, el botón para copiar el ABI; este código se almacena dentro del archivo `abi.js`, ya que es utilizada para las llamadas a los métodos.

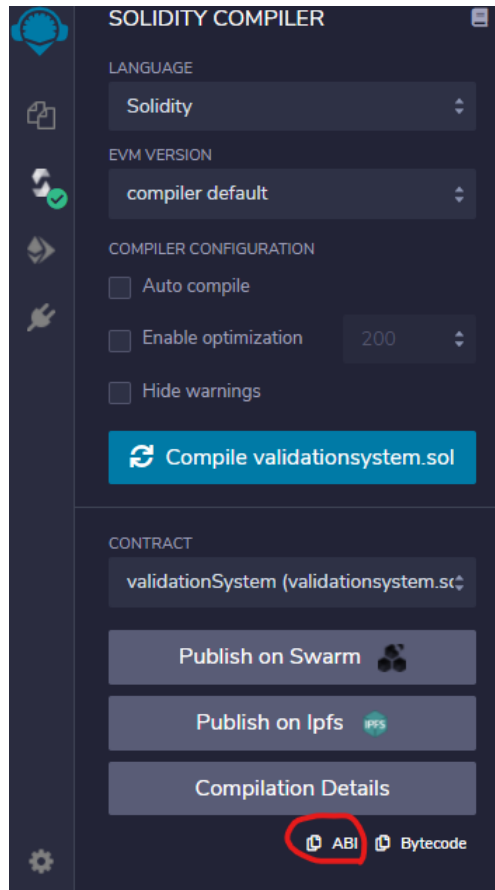


Figura 6.20: Copiar el ABI del código, Fuente: captura de pantalla.

Posteriormente se crean una serie de archivos como `app.js`, `document.js`, `parameters.js` que contendrán la lógica para conectar la vista frontal con la Blockchain. El último archivo `parameters.js` almacena la dirección del smart contract, la dirección se crea en el momento que se publicó el código con Remix, la dirección generada es `“0x7006882779C21D8246 C82989F813237f78A781b1”`.

Las vistas desarrolladas con VUE JS:

1. *Gestión de Área*: permite crear Áreas, asignar un propietario mediante una dirección o clave pública, un nombre, descripción, a partir de él se pueden crear eventos.

Áreas					
NUEVO (+)					
ID	Nombre	Descripción	Cant Eventos	Estado	Acciones
1	Secretaría de Extensión		2	actived	  
4	Area Desarrollo		0	actived	  
5	Secretaría Académica		1	actived	  
6	Secretaría de Ciencia y Tecnología		0	actived	  

Figura 6.21: Vista de Áreas, Fuente: captura de pantalla (propuesta de sistema).

2. *Gestión de Evento*: se crean los Eventos encargados de generar los documentos, los datos para crear un evento son el nombre, área relacionada, descripción, fecha de cuando inicio el evento, fecha cuando finaliza el evento y un estado. Las fechas sirven para saber en que período se crearon los documentos, ya que cada documento está relacionado a un evento.

Eventos							
NUEVO (+)							
ID	Nombre	Descripción	ID Área	Inicio de Evento	Fin del Evento	Cantidad Docs.	Acción
1	Curso de Programación3		1	25/5/2021 14:44	26/5/2021 11:44	4	  
4	Congreso 3		1	29/4/2021 10:17	6/5/2021 10:17	1	  
3	Taller de Inglés		5	1/3/2021 12:0	1/3/2022 12:0	0	  

Figura 6.22: Vista de Eventos, Fuente: captura de pantalla (propuesta de sistema).

3. *Gestión de Documentos*: Se verifican o validan los documentos, si el usuario tiene el rol para crear documentos también se utiliza como carga del mismo, si no tiene un rol solamente puede verificar si el documento fue registrada por la organización. Un usuario propietario del área ve como se muestra en la figura 6.23.

The screenshot shows a web interface titled 'Validación'. At the top, there are three dropdown menus: 'Secretaría de Extensión', 'Congreso 3', and 'actived'. Below these are two input fields: 'Fecha de Vencimiento' (mm/dd/yyyy) and 'Razón del Estado'. A section labeled 'DOCUMENTOS' shows a file named 'certificado_modelo.pdf' with a progress bar at 1/1. Below this is a table with two tabs: 'VERIFICADOS' and 'NO VERIFICADOS'. The 'VERIFICADOS' tab is active. The table has columns: Hash, Nombre Archivo, Estado, Evento ID, Nueva Versión, and Acciones. There is one row of data.

Hash	Nombre Archivo	Estado	Evento ID	Nueva Versión	Acciones
bdb8...b203	certificado_modelo.pdf	actived	Congreso 3 (4)	No	

Figura 6.23: Vista de Documentos como propietario, Fuente: captura de pantalla (propuesta de sistema).

Y en el caso de una dirección pública que no está registrada como propietario de un área o de la organización visualiza los documentos como en la figura 6.24.



Figura 6.24: Vista de Documentos como usuario público, Fuente: captura de pantalla (propuesta de sistema).

4. *Gestión de Organización*: Mantiene los datos de la Organización, permite cambiar el nombre, cargar nuevos estados y cambiar al propietario de la organización, que en este caso es el usuario que tiene permitido todas las acciones creadas. Los estados cargados por defecto son “deleted”, “actived”, “expired” y no pueden ser eliminados en el sistemas ni modificados excepto “expired”.

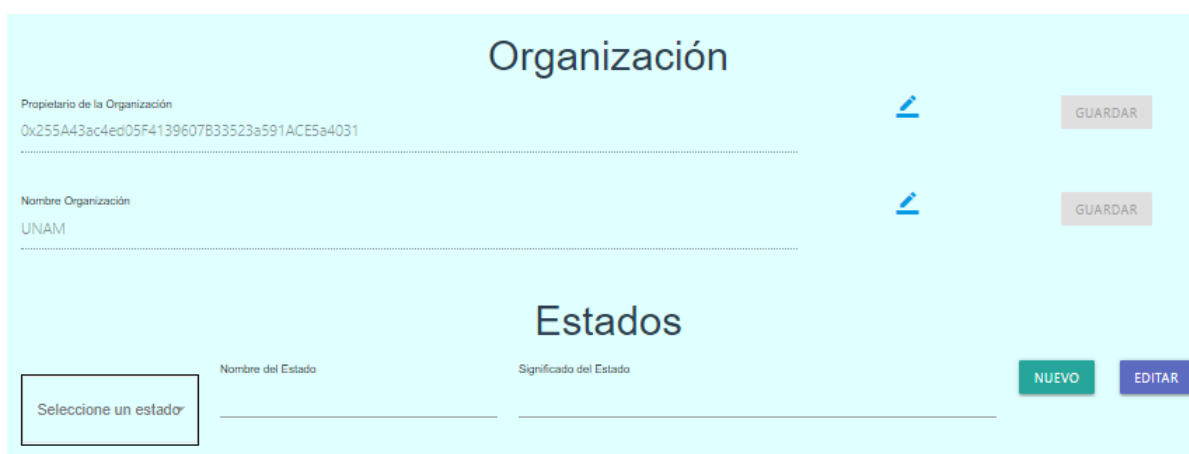


Figura 6.25: Vista de Organización, Fuente: captura de pantalla (propuesta de sistema).

CAPÍTULO 7

Ensayos de Validaciones

Los ensayos se realizarón con certificados emitidos por la Secretaría de Extensión de la FCE de la UNaM, por una cuestión de conveniencia en el acceso de los documentos digitales de la Universidad, pero los ensayos se podrían realizar sin importancia del área.

El ensayo inicia con la cuenta o dirección del propietario del smart contract, quien carga los datos base, como el nombre de la Organización, las áreas, los eventos en donde se emitirán los certificados. Partiendo de un smart contract desplegado como se muestra en otras secciones, el administrador o propietario de la organización es la misma dirección que se encargó de publicar el contrato inteligente, pero el usuario administrador puede cambiar su dirección por otra.

7.1. Preparación Inicial

Los siguientes pasos se realizan con la cuenta del propietario del Smart Contract. Primero se ingresa el nombre de la organización a “Universidad Nacional de Misiones Facultad de Ciencias Económicas”, esto sirve para identificar el nombre de la organización que es dueño del Smart Contract, se crean estados nuevos como “en espera”, “on hold” 7.1. Se crean las Áreas de la Organización, es este caso “Secretaría de Extensión”, con el propietario del área con dirección “0x255A43ac4ed05F41396 07B33523a591ACE5a4031” y el estado “activated”, conforme a la figura 7.2. El propietario del área puede crear las actividades o eventos y realizar la validación de los documentos.

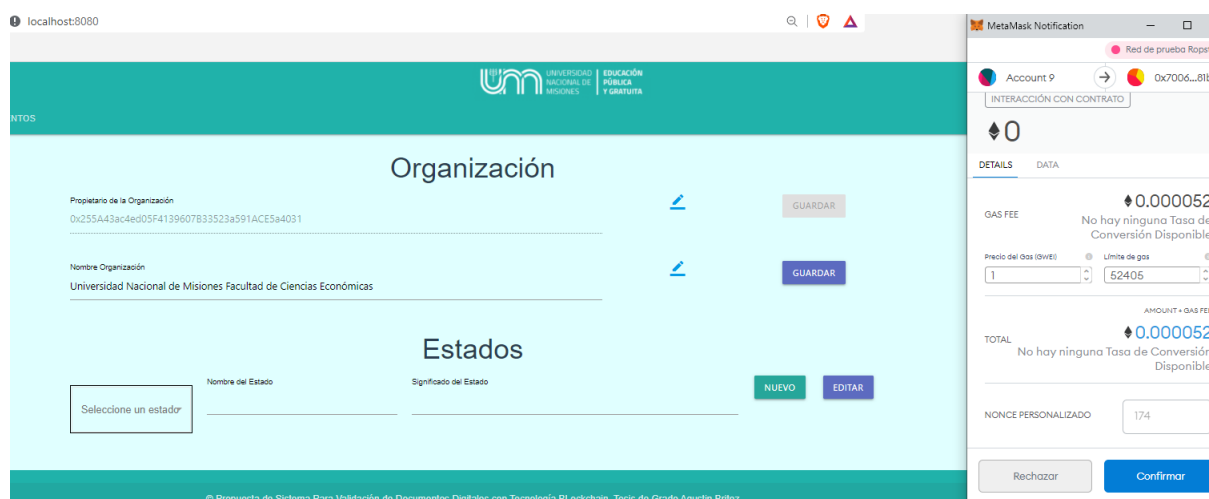


Figura 7.1: Cambio de nombre de la organización, Fuente: captura de pantalla (propuesta de sistema).

Figura 7.2: Creación de nueva área en el sistema, Fuente: captura de pantalla (propuesta de sistema).

Se crean los eventos “Fondos Buitres”, “Geogebra” y “Taller Liquidación”, que se relacionan al área recién mencionada que es “Secretaría de Extensión”.

Los ensayos serán realizados con la cuenta del propietario del Smart Contract.

7.2. Ensayo A

Se ensayó en un flujo habitual de la Secretaría de Extensión de la Facultad de Ciencias Económicas (FCE), siguiendo los circuitos normales de una actividad realizada por esta área. En base a la información recaudada en la entrevista (ver anexo A.3), comenzando el flujo desde el momento que se crea un documento digital para entregar al participante de una actividad.

Una vez que el evento haya finalizado y se deben entregar los certificados, estos tiene que ser subidos al sistema de validación de la siguiente manera.

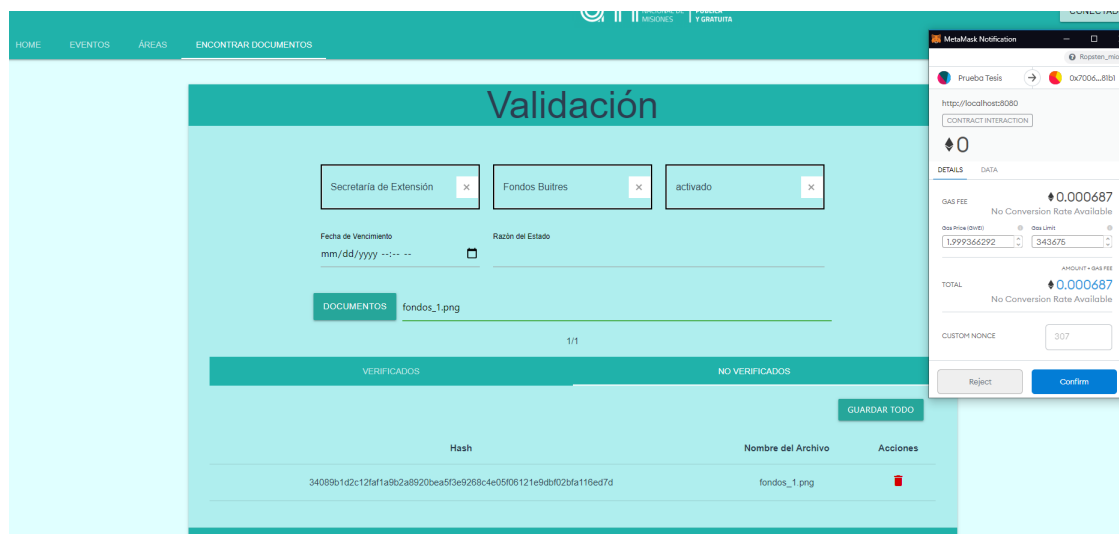


Figura 7.3: Carga de certificado Fondos Buitres, Fuente: captura de pantalla (propuesta de sistema).

1. Ingresar al sistema con la cuenta de administrador del área que organiza el evento y dirigirse a la vista “encontrar documento”, en la parte superior del sistema se puede observar en la figura 7.3.
2. Seleccionar el área que en este caso es la Secretaría de Extensión.
3. Seleccionar el evento, en el caso actual es Fondos Buitres y seleccionar el estado que se requiere para los documentos a subir, por ejemplo el estado “activado”.
4. Hacer clic en el botón “documentos” y se busca el certificado a subir en este caso es llamado fondos_1.png, el certificado se muestra en la figura 7.4.
5. En la figura 7.3 muestra el momento antes de generar el hash y almacenarlo en la Blockchain, cuando el propietario del área presiona el botón “GUARDAR TODO”, el sistema genera el hash de los documentos seleccionados que están en la parte de “NO VERIFICADOS” y lo almacena en la Blockchain.
6. A partir del punto cualquier individuo que consulte la Blockchain podrá verificar que el certificado actual fue emitido por la Secretaría de Extensión de la FCE.



Figura 7.4: Modelo certificado original de fondos buitres, Fuente: Brindada por la Secretaría de Extensión de la FCE.

En el caso que un interesado desea comprobar que el PDF o documento que esta tiene en su poder fue emitido por la Secretaría de Extensión de la FCE de la UNaM. Debe ingresar al sistema en el caso de no ser propietario del área o del smart contract la vista aparece como la figura 6.24, luego seleccionar el botón "DOCUMENTOS" y subir el certificado que se quiere validar que es fondos_1.png



Figura 7.5: Confirmación que el certificado se encuentra en la blockchain, Fuente: captura de pantalla.

Sí una persona intenta modificar el certificado para sus propios beneficios o para ocasionar algún tipo de daño de reputación u otro caso, el sistema se encargará de separar los certificados que no fueron emitidos por una cuenta de propietario de área de la FCE o del propietario del smart contract. A continuación el certificado de la figura 7.4 fue modificado como se muestra en la figura 7.6 se cambió el nombre del participante del evento a “Dr Ezequiel Britez DNI 00.134.321” (el nombre y el DNI del nuevo participante son arbitrarios y ficticios). Este es un posible caso el cual una persona necesite un certificado con características esenciales para ella y modifique el documento a su voluntad.



Figura 7.6: Certificado Fondos Buitres Modificado, Fuente: captura de pantalla.

En el caso que el certificado sea presentado a una entidad podrá verificar que los datos del certificado son correctos. Para ello la entidad debe acceder al sistema y realizar los mismo pasos ya explicados, el cual es subir el documento con el botón “DOCUMENTOS” de la vista. Para saber que el certificado o documento mantiene su información inalterable y que es emitida por la FCE de la UNaM, el hash del certificado aparece en la sección “VERIFICADOS” como se muestra en la figura 7.7.



Figura 7.7: Validación de certificado modificado, Fuente: captura de pantalla.

Como se puede observar la tabla 7.1 el hash del documento modificado es distinto al documento original.

Hash Original	Hash Modificado	Cambia
34089b1d2c12faf1a9b2a8920 bea5f3e9268c4e05f06121e9d bf02bfa116ed7d	e95eacaecc442decbe397af 4533cbc3bcb027926f9c6fc eb3d7f5c8c54141d	SI

Tabla 7.1: Comparación de los hash del certificado original 7.4 y el modificado 7.6

El resultado son distintos hashes porque los bits de cada documento son distintos (Back, 2002; BlockCerts, s.f.-a; Nakamoto, 2008).

7.3. Ensayo B

Se sometió a prueba el certificado (Certificado_Geogebra.png de la figura 7.8), donde se cargan los datos Área: Secretaría de Extensión, Evento: Geogebra y Estado: Activado. Y de la misma manera que el Ensayo A se carga el documento y confirma la transacción.



Figura 7.8: Modelo de certificado de participación curso Geogebra, Fuente: Brindada por la Secretaría de Extensión de la FCE.

Es observable en la figura 7.9 que el certificado se subió correctamente y el hash se genero.

Validación

Secretaría de Extensión ×

Geogebra ×

activado ×

Fecha de Vencimiento

mm/dd/yyyy --:-- --

Razón del Estado

DOCUMENTOS

Certificado_Geogebra.png

1/1

VERIFICADOS

NO VERIFICADOS

EDITAR TODO

Hash	Nombre Archivo	Estado	Evento ID	Nueva Versión	Acciones
256f...0a96	Certificado_Geogebra.png	activado	Geogebra (9)	No	<div style="display: flex; justify-content: center; gap: 10px;"> <div style="color: blue;">🔍</div> <div style="color: green;">✏️</div> <div style="color: red;">🗑️</div> </div>

Figura 7.9: Certificado de Geogebra Validado, Fuente: captura de pantalla.

Como prueba se cambió el nombre del archivo de Certificado_Geogebra.png a Bartolomeo_Certificado_Geo.png (ver Figura 7.10) y se sometió al sistema para conocer si la integridad del certificado cambió.

El certificado aparece en la sección “VERIFICADOS” tras cambiar el nombre del archivo y probarlo. El sistema verifica que el hash del documento no se modificó porque internamente no sufrió cambios, esto permite realizar modificaciones en el nombre del documento y no alterar a la validación.

DOCUMENTOS	Bartolomeo_Certificado_Geo.png	1/1			
VERIFICADOS			NO VERIFICADOS		
EDITAR TODO					
Hash	Nombre Archivo	Estado	Evento ID	Nueva Versión	Acciones
256f...0a96	Bartolomeo_Certificado_Geo.png	activado	Geogebra (9)	No	

Figura 7.10: Cambio de nombre del Certificado, Fuente: captura de pantalla.

A continuación se realiza una modificaciones internas del Certificado_Geogebra.png y el resultado se muestra en la figura 7.11



Figura 7.11: Certificado de Geogebra No Validado, Fuente: captura de pantalla.

En la figura 7.11 se modificó el año del evento, en algunas ocasiones las personas requieren demostrar que realizó una capacitación o que tienen las habilidades que el certificado demuestra. O cuando necesitan tener años de experiencias podría ocurrir un cambio de fechas.

Se probó el certificado modificado con el mismo nombre archivo pero este último no afecta al cálculo del hash. La prueba se muestra en la figura 7.12

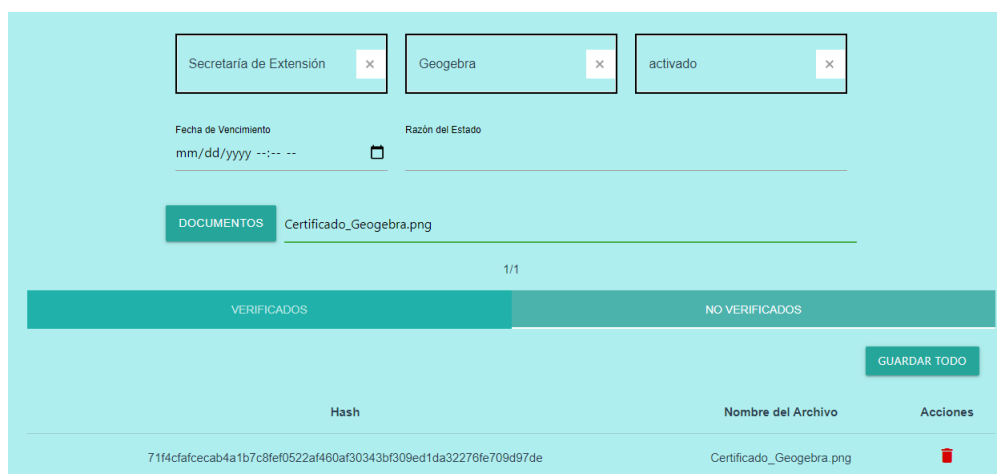


Figura 7.12: Certificado de Geogebra Validado, Fuente: captura de pantalla.

Como resultado se detectó que el HASH cambió por ende el certificado aparece en el sector “NO VERIFICADOS”. Con la tabla 7.2 se puede observar la diferencias entre los hashes.

Hash Original	Hash Modificado	Cambia
256f5c768d464033868cede3a e62dffbe3c506ceed9079b951 f08b4b3bd00a96	71f4cfafcecab4a1b7c8fef0 522af460af30343bf309ed1d a32276fe709d97de	SI

Tabla 7.2: Comparación de los hash del certificado original 7.8 y el modificado 7.11

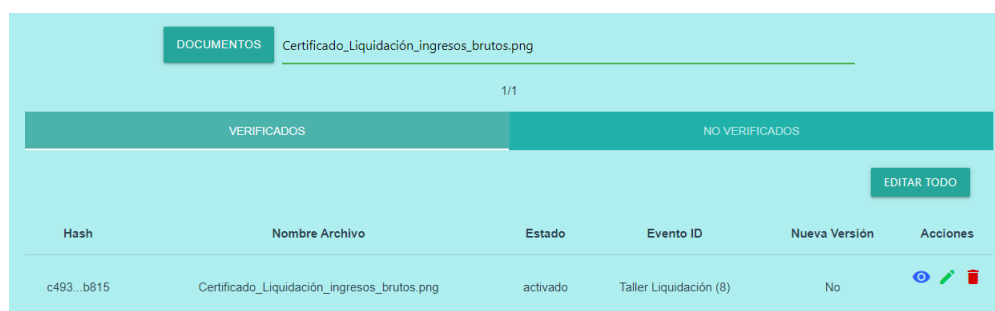
7.4. Ensayo C

Se sometió a prueba el certificado (Certificado_Liquidación_ingresos_brutos.png de la figura 7.13), donde se cargan los datos Área: Secretaría de Extensión, Evento: Taller Liquidación y Estado: Activado. De la misma manera que el Ensayo A y B se carga el documento y confirma la transacción.



Figura 7.13: Modelo de certificado original del Curso de Liquidación e Ingresos Brutos, Fuente: Brindada por la Secretaría de Extensión de la FCE.

Se realiza el mismo proceso donde se valida el certificado (ver figura 7.14).






Hash	Nombre Archivo	Estado	Evento ID	Nueva Versión	Acciones
c493...b815	Certificado_Liquidación_ingresos_brutos.png	activado	Taller Liquidación (8)	No	  

Figura 7.14: Prueba de validación del certificado original, Fuente: captura de pantalla.

Se realizó un cambio al nombre y DNI en el certificado como resultado se obtuvo el siguiente certificado (ver figura 7.15)



Figura 7.15: Certificado alterado del curso de Liquidación e Ingresos Brutos, Fuente: captura de pantalla.

Al probar el certificado en la propuesta de sistema de validación, se obtiene que fue modificado porque el hash cambió, observar la tabla 7.3 de los hashes las diferencias.

Hash Original	Hash Modificado	Cambia
c493819261bd3dc6e88b99637 cdf843f84c1dc7080854b83ec bfc5637153b815	6719b1115644f98048db810f 0228d4eb19f71f10ecc60b85 ecab13992c26be94	SI

Tabla 7.3: Comparación de los hash del certificado original 7.13 y el modificado 7.15

En los tres ensayos se obtuvieron los mismos resultados al modificar internamente los certificados, esto se debe al modificar 1 píxel los bits del documento digital cambia por ende la el hash generado es otro, cuando se sube un documento al sistema este genera el hash y los busca en el smart contract, si se encuentra almacenado significa que una cuenta de la FCE de la UNaM se encargó de subirla en algún momento, en caso contrario no se encontrará el hash por ende el certificado no es válido ya que pudo ser emitido por un ente diferente o persona no autorizada.

7.5. Consideraciones Detectadas

El desarrollo del sistema no confirma que a través de la generación de los hashes un documento digital sea la única manera de validarlos. Los estándares investigados y otros sistemas aplicaban este método. También hay que considerar que la prueba del sistema se realizó en una Blockchain de prueba, por ende no se puede confirmar que el sistema funciona en todas las Blockchain. Si se requiere su uso es necesario probar que la Blockchain soporte el lenguaje de programación Solidity.

Se tomaron en cuenta los sistemas ya existentes con los estándares investigados y se optó por utilizar funcionamientos similares en los sistemas y que toda la información esté almacenada en la Blockchain y no solo en el documento. Los ensayos realizados demuestran que cambiando el nombre del archivo no altera el hash, por ende tolera la modificaciones que no intervienen con la integridad del documento.

Por otro lado, la implementación del sistema se puede anexar con un sistema externo que gestione los documentos y los suba de manera automática en la Blockchain, permitiendo automatizar todo el proceso.

CAPÍTULO 8

Conclusiones

Los documentos digitales, sin distinguir el tipo información que contiene, requieren de mecanismos o sistemas que permitan validarlos íntegramente. En el desarrollo del presente trabajo se expuso que existen diversas prácticas para respaldar el contenido de documentos digitales tales como los certificados digitales y firma digital pero carecen de los resguardos que sí posee la tecnología Blockchain.

Tanto a nivel internacional, nacional y provincial existen antecedentes y una fuerte tendencias de utilizar la tecnología Blockchain para el resguardo y validación de documentaciones digitales y diversos activos, lo que puede aplicarse en los documentos digitales que respalden la realización de actividades de extensión en la UNaM.

A efectos de buscar una solución al problema planteado, se desarrolló un sistema que permite a la entidad sellar en la Blockchain los documentos digitales en su poder para que posteriormente usuarios interesados consulten sobre la validez de los mismos, todo lo cual refleja la inmutabilidad de los datos y/o si existen modificaciones en los documentos digitales (conforme los ensayos realizados).

Por todo lo expuesto existe evidencia suficiente para no rechazar la hipótesis planteada en el presente trabajo sin perjuicio de que existen futuras líneas de investigaciones, tales como:

1. Investigación de Blockchain para uso académico con bajos costos (con el fin de reconocer las Blockchains que permitan desplegar sistemas académicos o soluciones que permitan aprovechar la tecnología Blockchain).
2. Análisis de documentos digitales originales con métodos de validaciones, a fin de conocer cuántos documentos digitales utilizan, métodos que respalde su integridad, permitiendo incentivar el uso de sistemas como Blockchain para fortalecer la seguridad de los documentos.
3. Anexar la propuesta del sistema de validación con sistema de gestión documental, para generar y validar los documentos digitales de manera automatizadas en la Secretaría de Extensión de la Facultad de Ciencias Económicas de la UNaM, (esto permitirá reducir el proceso que realiza la Secretaría para generar los documentos y anexando a la propuesta de sistema de validación, tendrían los documentos respaldados por la Blockchain).

Bibliografía

- Academia, B. (2018). *“¿Qué es Proof of Stake? Programador explica”*. Youtube. Descargado 08/05/2021, de https://www.youtube.com/watch?v=2_vVII1K5yE
- Ambito. (2021). *“cuáles son las mejores wallets para criptomonedas”*. ambito. Descargado 09/05/2021, de <https://www.ambito.com/wallets/cuales-son-las-mejores-criptomonedas-n5182273>
- AvanzaExportador. (2009). *“certificados electrónicos y digitales. firma electrónica”*. Youtube. Descargado 06/11/2020, de <https://www.youtube.com/watch?v=EU6vgU077xU&t=208s>
- Back, A. (2002). *“hashcash - a denial of service counter-measure”*. , 10. Descargado de <ftp://sunsite.icm.edu.pl/site/replay.old/programs/hashcash/hashcash.pdf>
- Badreddin, O., Rivera, A. G., y Malik, A. (2018). *“blockchain Fundamentals and Development Platforms”*. IBM Corp, 3.
- BFA. (s.f.-a). *“aplicaciones en BFA”*. Descargado 13/11/2020, de <https://bfa.ar/bfa/aplicaciones>

- BFA. (s.f.-b). “BFA como funciona”. Descargado 13/08/2020, de <https://bfa.ar/bfa/como-funciona>
- BFA. (s.f.-c). “protocolos de consenso”. Descargado 05/08/2020, de <https://bfa.ar/Blockchain/protocolos-de-consenso>
- BFA. (s.f.-d). “que es BFA”. Descargado 13/11/2020, de <https://bfa.ar/bfa/que-es-bfa>
- BFA. (s.f.-e). “smart Contracts”. Descargado 05/08/2020, de <https://bfa.ar/Blockchain/smart-contracts>
- BFA. (s.f.-f). “trazabilidad de alimentos”. Descargado 09/06/2021, de <https://bfa.ar/Blockchain/casos-de-uso/trazabilidad-de-alimentos>
- BlockCerts. (s.f.-a). “FAQ”. blockcerts. Descargado 17/05/2021, de <https://www.blockcerts.org/guide/faq.html>
- BlockCerts. (s.f.-b). “introduction BlockCerts”. Descargado 12/11/2020, de <https://www.blockcerts.org/guide/>
- Bragagnolo, S., Rocha, H., Denker, M., y Ducasse, S. (2018). “SmartInspect: solidity smart contract inspector”. En *2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE)* (pp. 9–18). Campobasso: IEEE. doi: 10.1109/IWBOSE.2018.8327566
- Brys, C. (2019). “la Cadena de Bloques Blockchain: Un abordaje comprensible a su definición y posibles usos” (Inf. Téc.).
- Cheng, J.-C., Lee, N.-Y., Chi, C., y Chen, Y.-H. (2018). “blockchain and smart contract for digital certificate”. En *2018 IEEE International Conference on Applied System Invention (ICASI)* (pp. 1046–1051). Chiba, Japan: IEEE. doi: 10.1109/ICASI.2018.8394455
- Choo, K.-K. R., Dehghantanha, A., y Parizi, R. M. (Eds.). (2020). “blockchain Cybersecurity, Trust and Privacy” (Vol. 79). Gewerbestrasse 11, 6330 Cham, Switzerland: Springer International Publishing. doi: 10.1007/978-3-030-38181-3
- Clementín, F. (2021). “la provincia argentina de Misiones podría emitir su propia stablecoin”. criptonoticias. Descargado 26/07/2021, de <https://www.criptonoticias.com/regulacion/provincia-argentina>

- misiones-podria-emitar-propia-stablecoin/
- Cámara de Representantes de la Provincia de Misiones. (2020). *"Programa Misionero de Inovación Financiera con Tecnología Blockchain y Criptomoneda"*. Descargado 21/10/2020, de <http://www.diputadosmisiones.gob.ar/nuevo/archivos/proyectos/P55666.pdf>
- Coinsenda. (2019). *"tipos de Wallet"*. medium. Descargado 09/05/2021, de <https://medium.com/@coinsenda/tipos-de-wallet-8054a6418d3c>
- Cripto247. (2021). *"2021, el año de DeFi: las tendencias que se vienen y a qué prestarle atención"*. cripto247. Descargado 11/08/2021, de <https://www.cripto247.com/comunidad-cripto/2021-el-ano-de-defi-las-tendencias-que-se-vienen-y-a-que-prestarle-atencion-199138>
- CriptoMonedasTV. (2018). *"entrevista: BlockCerts con Daniel Páramo"*. Youtube. Descargado 12/11/2020, de <https://www.youtube.com/watch?v=eJ9lCuEcUBU>
- D'Agostino, A. (2020). *"exclusivo: este el proyecto de ley para impulsar criptomonedas, activos digitales y el ePeso en la Argentina"*. iproup. Descargado de <https://www.iproup.com/economia-digital/18143-bitcoin-como-es-el-proyecto-de-ley-de-criptomonedas>
- Dannen, C. (2017). *"introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners"*. Berkeley, CA: Apress. doi: 10.1007/978-1-4842-2535-6
- Drescher, D. (2017). *"blockchain Basics"*. Berkeley, CA: Apress. doi: 10.1007/978-1-4842-2604-9
- Edublocs. (2019). *"nueve universidades utilizan blockcerts para crear un acreditaciones digitales"*. edublocs. Descargado 12/11/2020, de <https://www.edublocs.org/noticias/params/post/1783300/nueve-universidades-utilizan-blockcerts-para-crear-un-acreditaciones-digita>
- Educacionit. (s.f.). *"Carrera Blockchain"*. educacionit. Descargado 12/05/2021,

- de <https://www.educacionit.com/carrera-blockchain>
- Ethereum. (s.f.). “solidity”. soliditylang. Descargado 12/05/2021, de <https://docs.soliditylang.org/en/v0.8.4/>
- Ethereum. (2020). “¿Qué es Ethereum?”. ethereum.org. Descargado 09/11/2020, de <https://ethereum.org/es/what-is-ethereum/>
- Ethereum. (2021). “TRANSACTIONS”. ethereum. Descargado 05/04/2021, de <https://ethereum.org/en/developers/docs/transactions/>
- García Rojas, W. A. (2008). “implementación de firma digital en una plataforma de comercio electrónico” (Tesis Doctoral, Universidad Católica del Perú, Lima, Perú). Descargado de <https://tesis.pucp.edu.pe/repositorio/handle/20.500.12404/352>
- Gauchi Risso, V. (2012). “aproximación teórica a la relación entre los términos gestión documental, gestión de información y gestión del conocimiento”. *Revista española de Documentación Científica*, 35(4), 531–554. doi: 10.3989/redc.2012.4.869
- Jimenez, D. (2020). “proyecto Colmena basado en tecnología Blockchain se llevó a cabo en concurso NAVES 2020”. cointelegraph. Descargado 12/08/2021, de <https://es.cointelegraph.com/news/colmena-project-based-on-blockchain-technology-received-a-special-mention-in-the-naves-2020-contest>
- Jirgensons, M., y Kapenieks, J. (2018). “blockchain and the Future of Digital Learning Credential Assessment and Management”. *Journal of Teacher Education for Sustainability*, 20(1), 145–156. Descargado 27/06/2020, de <https://content.sciendo.com/doi/10.2478/jtes-2018-0009> doi: 10.2478/jtes-2018-0009
- Joaquín López Lérda, J. J. M. P. (2016). “la economía de Blockchain”. Kindle.
- Kelly, J., Lauer, M., Prinster, R., y Zhang, S. (2018). “investigation of Blockchain Network Security”. , 15.
- King, S., y Nadal, S. (2012). “PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake”. *artículo autopublicación*, Agosto, 6.
- La Honorable Cámara de Diputados y el Senado de la Nación. (2020). “PRO-

- YECTO DE LEY REGULACIÓN DE CRIPTOACTIVOS". Descargado de <https://www4.hcdn.gob.ar/dependencias/dsecretaria/Periodo2020/PDF2020/TP2020/6055-D-2020.pdf>
- Lab, M. M. (2016). *"what we learned from designing an academic certificates system on the blockchain"*. medium. Descargado 17/05/2021, de <https://medium.com/mit-media-lab/what-we-learned-from-designing-an-academic-certificates-system-on-the-Blockchain-34ba5874f196#.4m4bmwcm0>
- Mayor, C. E. O. (2017). *"Blockchain la nueva base de datos no SQL en Big Data"*. UNIVERSIDAD DE GUADALAJARA – CUCEA FACULTAD DE INGENIERIA(UNIVERSIDAD LIBRE DE COLOMBIA), 32.
- MetaMask. (2021). *"metamask introduction"*. metamask. Descargado 21/05/2021, de <https://docs.metamask.io/guide/#why-metamask>
- Mozilla. (s.f.). *"trabajando con JSON"*. mozilla. Descargado 13/07/2021, de <https://developer.mozilla.org/es/docs/Learn/JavaScript/Objects/JSON>
- Nakamoto, S. (2008). *"bitcoin: A Peer-to-Peer Electronic Cash System"*. , 9.
- Noticiasdel6. (2020). *"proyecto Colmena: un modelo de recuperación de residuos"*. noticiasdel6. Descargado 12/08/2021, de <https://www.noticiasdel6.com/proyecto-colmena-un-modelo-de-recuperacion-de-residuos/>
- OpenCerts. (s.f.). *"frequently Asked Questions"*. opencerts. Descargado 23/07/2021, de <https://www.opencerts.io/faq>
- OpenCerts. (2018). *"gestión de certificados académicos sobre la blockchainEthereum"*. Universitat Oberta de Catalunya. Descargado 12/11/2020, de <https://informatica.blogs.uoc.edu/gestion-de-certificados-academicos-sobre-la-blockchain-ethereum>
- Palomeque, A. J. G. (2015). *"implementación de certificados y firmas digitales para sistemas de información transaccionales en una empresa gubernamental"* (Tesis Doctoral, Escuela Superior Politécnica Del Litoral, Guayaquil Ecuador).

- Descargado de <https://www.dspace.espol.edu.ec/handle/123456789/30019>
- Preukschat, A., Várez, J. L., Kuchkovsky, C., Lardies, G. G., García, D. D., y Íñigo Molero. (2018). *“blockchain: la revolución industrial de internet”*. Barcelona: Gestión 2000. (OCLC: 1079886042)
- Rambo, A., Jara, J., Estigarribia, O., Sydniuk, R., y Brys, C. (2021). “blockchain para el Gobierno Digital”. , 10.
- Raskin, M. (2017). “the law and legality of smart contracts”. 1 *Georgetown Law Technology Review* 304, 1, 37. doi: <https://ssrn.com/abstract=2959166>
- Remix. (s.f.). *“Deploy & Run transactions in the Blockchain”*. remix. Descargado 12/05/2021, de <https://remix-project.org/>
- Retamal, C. D., Roig, J. B., y Muños Tapia, J. L. (2017). “la blockchain: fundamentos, aplicaciones y relación con otras tecnologías disruptivas”. *Dialnet*, 33–40.
- Rezaeighaleh, H., y Zou, C. C. (2019). “new Secure Approach to Backup Cryptocurrency Wallets”. En *2019 IEEE Global Communications Conference (GLOBECOM)* (pp. 1–6). Waikoloa, HI, USA: IEEE. doi: 10.1109/GLOBECOM38437.2019.9014007
- Sadouskaya, K. (2017). *Adoption of Blockchain Technology in Supply Chain and Logistics* (Tesis Doctoral, XAMK). Descargado de <https://www.theseus.fi/bitstream/handle/10024/126096/Adoption%20of%20Blockchain%20Technology%20in%20Supply%20Chain%20and%20Logistics.pdf?sequence=1>
- Satoh, A., y Inoue, T. (2007). “ASIC-hardware-focused comparison for hash functions MD5, RIPEMD-160, and SHS”. *Integration*, 40(1), 3–10. doi: 10.1016/j.vlsi.2005.12.006
- Schollmeier, R. (2002). “a definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications”. En *Proceedings First International Conference on Peer-to-Peer Computing* (pp. 101–102). Linköping, Sweden: IEEE Comput. Soc. doi: 10.1109/P2P.2001.990434
- Sheinix, C. (2020). *“bitcoin basico - criptografía de llave pública llave privada -*

- explicación sencilla*". Youtube. Descargado 06/11/2020, de <https://www.youtube.com/watch?v=mtDqU0yvwus>
- Torres, A. D. L. (2020). "blockchain: características y estado actual. Posible efecto sobre la auditoría. Blockchain: characteristics and current state. Possible effect on the audit". , 33.
- Truffle. (s.f.-a). "*ganache overview*". trufflesuite. Descargado 12/11/2020, de <https://www.trufflesuite.com/docs/ganache/overview>
- Truffle. (s.f.-b). "*TRUFFLE OVERVIEW*". trufflesuite. Descargado 12/05/2021, de <https://www.trufflesuite.com/docs/truffle/overview>
- UNaM. (2012). "*Estatuto de la Universidad Nacional de Misiones X Asamblea Universitaria*".
- VUEJS. (s.f.). "*introduction*". vuejs. Descargado 02/06/2021, de <https://v3.vuejs.org/guide/introduction.html>
- VUEJS. (2019). "*overview*". vuejs. Descargado 25/05/2021, de <https://cli.vuejs.org/guide/>
- Vázquez, S. (2020). "episodio clínico en blockchain de Ethereum". *Universitat Oberta de Catalunya (UOC)*, 56.
- Web3js. (2016). "*web3.js - Ethereum JavaScript API*". Descargado 12/05/2021, de <https://web3js.readthedocs.io/en/v1.3.4/>

ANEXO *A*

Anexos del Desarrollo

A.1. Enlaces de Interés

Los enlaces se agregaron el día 13/08/2021.

1. El enlace donde se desarrolló el front end del sistema es : https://github.com/agustinbritez/Sistema_Validacion.git.
2. Enlace de Smart Contract publicado en Ropsten: <https://ropsten.etherscan.io/address/0x7006882779C21D8246C82989F813237f78A781b1>
3. Enlace de Remix : <https://remix.ethereum.org/>
4. Enlaces de páginas grifos (obtener ETH en Ropsten) : <https://faucet.metamask.io/> o <https://faucet.ropsten.be>.

A.2. Métodos de relevamientos

Para relevar información acerca de la facultad, se realizó una entrevista formal mediante comunicación remota con el encargado de la Secretaría de Extensión, también comunicación informal mediante chats, y observaciones, se utilizó documentos referenciados en la bibliografía como el estatuto de la UNaM.

A.3. Comunicaciones Personales y Observaciones

Se realizó charlas personales con el encargado de la Secretaría de extensión de la FCE, se realizó una entrevista, y conversaciones por mensaje de textos los días 02,03,04 de octubre del 2021.

Las observaciones realizadas en la FCEQyN y FCE también sirvieron como aporte para el entendimiento del trabajo realizado por las ambas facultades.

A.3.1. Entrevista a Responsable de la Secretaría de Extensión de la Facultad de Ciencias Económicas

La entrevista fue realizada el 2 de Octubre del 2020, la información personal del entrevistado se mantiene anónimo.

1. ¿Actualmente que actividades debe realizar la Secretaría de Extensión?.

Se encarga de participar en cualquier evento o actividad que sea necesario la interacción con individuos académicos, no académicos, el objetivo principal es conectar, educar y debatir con exterior de la facultad, por ello se dictan cursos, eventos, charlas y no todos ellos son específicamente dedicado a estudiantes universitarios. Hay casos de capacitaciones que hemos hecho inclusive a empleados de algunas empresas de la región. El objetivo es incrementar los conocimientos del ambiente que rodea la universidad en sí. La gestión y logística de todos los materiales necesarios para efectuar una la charla o evento. Encargado de la coordinación de los eventos que certifiquen cualquier actividad que se requiera, como certificaciones a charlas, congreso. Para iniciar un evento por parte de algún docente, no docente, funcionarios o secretarios deben tener la aprobación mediante algún instrumento.

2. ¿Que significan los instrumentos?

Los instrumentos son los registros sobre la aprobación de algún evento mediante las disposiciones o resoluciones, estos documentos también impulsan la generación de las actividades.

Pero por otro lado para la aprobación de algún evento se pueda llevar en marcha es necesario que estos estén dentro del marco de programa o proyecto. Por lo general cuando hablamos de programas son los planes esperados a largo plazo. Mientras que los proyectos, son parte de los programas para el cumplimiento de objetivos en plazos más cortos. Entonces para que un evento sea aprobado debe encontrarse dentro de los planes de la universidad.

Si la actividad propuesta no está alineado a los planes o programas, la única opción es que sea aprobada mediante la disposición del decano.

Una vez aprobado el evento, se realiza toda la logística, en caso de que el evento sea muy grande se reservan las aulas más grandes, se establecen fechas para efectuar el evento, se calculan los presupuestos. Se planea la cantidad de tiempo que durara, etc.

Los eventos pueden ser congresos, charlas o cualquier actividad que se plante, puede haber diferentes charlas en un evento y recibir un certificado por cada una que asistió o simplemente un certificado para todas.

3. ¿Que documentación o tipo de documentos utilizan o manejan? (formato papel o digital).

Gestionamos documentos desde la parte del docente, alumnos, y no docentes, ya que la creación de una actividad puede venir desde el grupo de alumnos, como del docente o inclusive personal no académico. Los documentos son por lo general en formato físico o papel, pero también con la situación de la cuarentena se empezó a utilizar el token con pendrive, que nos permite enviarnos documentos digitales con las firmas de las autoridades. Los documentos son para la creación de algún evento en particular, para las peticiones de materiales o herramientas, también debemos hacer informes para reservar lugares de los eventos para una fecha específica, por otro lado manejamos la gestión de los certificados de los estudiantes que participan en una actividad o también a los disertantes.

Por último también se presentan presupuestos que conlleva realizar las actividades o los eventos en la facultad se manejan distintos tipos de documentos, pero en nuestra área con frecuencia empleamos para la gestión de una actividad.

4. ¿Como se gestionan los eventos o actividades?

Como dije anteriormente un evento puede ser iniciado por parte de un estudiante, docente o personal no docente, pero para ello la actividad

debe formar parte de un programa de la facultad o estar en disposiciones o resoluciones, no obstante para que el proyecto sea aprobado se necesita que estén alindados a los fines de la facultad, en casos que la actividad sea extraordinaria se evalúa por los responsables de la Secretaría de Extensión para rechazarla o aprobarla.

Cuando la actividad o el evento es aprobado, el personal administrativo inicia con las elevaciones para la gestión del sitio donde se van a realizar los eventos, inicia con los preparativos para seleccionar a los expositores o disertantes, la duración del evento, la publicidad, marketing, con todos los preparativos que son necesarios para el evento. Hay casos que son eventos como una juntas cortas, que no necesitan mucha gestión y por otro lado eventos enormes que hay que movilizar muchos sectores.

5. ¿La certificación de los eventos o actividades como se realizan?

Anteriormente, se realizaban eventos o actividades donde no se entregaban certificados, pero como resultado ¿qué sucedió?, disminuyo significativamente el número de participantes, entonces se volvió a la entrega de certificados.

Los certificados pueden ir dirigidos a cualquier individuo, tanto como estudiantes, docentes, no docentes. Porque la actividad se ejecuta para un público en particular o para todo público dependiendo el caso.

Para que los participantes puedan constatar que han asistido a un evento, se le entregan certificados, que pueden ser de asistencias, por exámenes, o cualquier actividad que se haya planificado para el evento.

Para la certificación lo que hacemos es tomar los datos de los posibles participantes primero creando un formulario de Google y publicándolo en todas nuestras redes sociales, antes este registro se hacía manualmente y nos consumía mucho tiempo. Sin embargo con la utilización de esta herramienta nos facilitó mucho la obtención de datos. A veces nos piden que hagamos estadísticas si la cantidad de participantes son estudiantes, docentes o no docentes. En otros casos, estadísticas de edades, si trabajan

o estudian y con el Google formulario podemos hacerlo de una manera más eficiente.

6. ¿Cual es el proceso que realizan para emitir los certificados ?

Los procesos que realizamos para emitir los certificados son los siguientes:

Unos días antes del evento se hace el modelado de los certificados, por ejemplo se toma un modelo antiguo o se crea uno nuevo donde se deja sin completar los espacios de las firmas y datos de los participantes. Después lo que hacemos es pedirle a las autoridades es firmas escaneadas para integrarlas al modelo del certificado, también nos encargamos de la gestión del lugar, conseguir los disertantes etc. Pero siguiendo con el proceso como mencione, se hace publicidad del evento mediante una página o se usa las redes sociales para anunciar las fechas del evento y un link o un formulario de Google para que los estudiantes se puedan registrar, no todos los eventos son de entrada libre, algunos hay que comprar las entradas. El día del evento se vuelven a verificar los usuarios que asistieron, para preparar los certificados con los datos de los usuarios que asistieron. Al finalizar el evento se entrega a los participantes el certificado de asistencia y en caso de que sea necesario los certificados de exámenes aprobados, estos certificados en algunos casos lo imprimimos y entregamos a los participantes, en otras situaciones hemos entregado de manera digital mediante correo electrónico. Para los certificados de examen aprobados, si son pocos participantes se entrega en el mismo evento, pero por lo general tarda un poco más de tiempo evaluarlos.

7. ¿Que problema puede percibir en cuanto a la validación de los certificados?

Sabemos que el tipo de validación mediante las firmas escaneadas no representan un método muy fiable para validar los documentos porque alguien que maneje un poco de informática puede editar los datos. Sería necesario tener algún método para poder validar los certificados de

manera sencilla, porque tampoco es conveniente complicar aún más el proceso de certificación.

8. ¿En el caso que algún estudiante quiera revalidar un certificado emitido hace tiempo, como lo hacen?

En esos casos generalmente el estudiante se acerca con el certificado y lo sellamos para dar validez al certificado, pero si el certificado es virtual se busca en nuestra base de datos y lo volvemos a enviar a su correo, o si el certificado no lo toman como válido se lo imprime y se lo sella pero eso sucede casos particulares.

A.4. Código Smart Contract

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >0.5.99 <0.8.0;
pragma experimental ABIEncoderV2;

contract validationSystem {
    /*******Modifier***** */

    modifier onlyOwnerOrg() {
        require(
            ownerOrg == msg.sender,
            "Only Organisation Owner can call this function"
        );
        _;
    }

    /*******Struct***** */

    struct Area {
        uint256 id;
        // bool editable;
        address ownerArea;
        string name;
        string description;
        uint256 state_id;
        uint256[] idEvents;
    }

    // name= Verificado mean= El documento esta verificado
    struct State {
```



```
        uint256 id;
        string name;
        string mean;
    }
    // mapping (uint256=>string) Reason;
    struct Document {
        //hash
        string idHash;
        uint256 state_id;
        uint256 event_id;
        //date of expired timestamp
        uint256 expiration;
        string reasonState;
        //new Version of document
        string newDocument;
    }

    struct Event {
        uint256 id;
        string name;
        string description;
        uint256 state_id;
        string startEvent;
        string endEvent;
        //Propietario que puede hacer cambios al evento
        uint256 area_id;
        //storage idhash
        string[] idDocuments;
    }

    //*****
    // only one organisation for avoid that use a smart
    contract for upload many files
```

```
string private organitaton;

address private ownerOrg;
//all states
State[] private states;
//one owner many areas
mapping(address => uint256[]) public ownerArea;
//all areas
Area[] private areas;
//all events
Event[] private events;
// hash => document
mapping(string => Document) private documents;

/***** Methods *****/
constructor() {
    ownerOrg = msg.sender;
    uint256[] memory idEvents;
    //La primer area todos las areas borradas hacen
    referencia a este
    Area memory zero = Area(0, address(this), "null",
    "null", 1, idEvents);
    areas.push(zero);

    State memory _state = State(0, "deleted", "deleted");
    State memory _state2 = State(1, "actived", "actived");
    State memory _state3 = State(2, "expired", "expired");

    states.push(_state);
    states.push(_state2);
    states.push(_state3);
    string[] memory str;
```

```

        Event memory _event = Event(0, "null", "null", 0, "", "
        zero.id, str);
        events.push(_event);
    }

/*****Organisation*****/
function setOrganisation(string memory org) public payable
onlyOwnerOrg {
    organitaton = org;
}

function editOwnerOrg(address newOwner) public payable
onlyOwnerOrg {
    ownerOrg = newOwner;
}

/*****State*****/

function addState(string memory name, string memory mean)
    public
    payable
    onlyOwnerOrg
    returns (uint256)
{
    uint256 id = states.length;
    State memory newState = State(id, name, mean);
    states.push(newState);
    return id;
}

function editState(uint256 id,string memory name,string
memory mean)

```

```

        public
        payable
        onlyOwnerOrg
        returns (
            uint256,
            string memory,
            string memory
        )
    {
        require((states.length > id) && (id > 1));
        (states[id].name = name);
        (states[id].mean = mean);
        return (id, states[id].name, states[id].mean);
    }

    /*****Area*****/
    function addArea(address _ownerArea,string memory _name,
    string memory _description) public payable onlyOwnerOrg {
        uint256 _id = areas.length;
        uint256[] memory _events;
        Area memory _newOwner =
            Area(_id, _ownerArea, _name, _description, 1,
            _events);

        areas.push(_newOwner);
        ownerArea[_ownerArea].push(_id);
    }

    function editArea(uint256 _id_area,string memory _name,
    string memory _description,uint256 _id_state) public
    payable {
        require(

```

```

        (ownerOrg == msg.sender) ||
            (areas[_id_area].ownerArea == msg.sender)
    );

    if (bytes(_name).length > 0) {
        areas[_id_area].name = _name;
    }

    if (bytes(_description).length > 0) {
        areas[_id_area].description = _description;
    }

    if (_id_state < states.length) {
        areas[_id_area].state_id = _id_state;
    }
}

function changeOwnerArea(uint256 _id_area, address
_newOwner)
    public
    payable
    onlyOwnerOrg
{
    require((areas.length > _id_area) && (_id_area > 0));
    //get oldest owner
    address _oldOwner = areas[_id_area].ownerArea;
    //change oldest area_id by 0
    for (uint256 index = 0; index < ownerArea[_oldOwner]
.length; index++) {
        if (ownerArea[_oldOwner][index] == _id_area) {
            ownerArea[_oldOwner][index] = 0;
            break;
        }
    }
}

```

```

    }

    ownerArea[_newOwner].push(_id_area);
    areas[_id_area].ownerArea = _newOwner;
}

/*****Event*****/

function addEventFull(uint256 _area_id,string memory name,
string memory description,string memory _startDate,
string memory _endDate) public payable returns (uint256 id)
{
    require(
        (areas[_area_id].ownerArea == msg.sender) ||
        (ownerOrg == msg.sender)
    );

    uint256 _id = events.length;
    string[] memory _document;

    Event memory evento =
        Event(
            _id,
            name,
            description,
            1, //state active
            _startDate,
            _endDate,
            _area_id,
            _document
        );

```

```
events.push(evento);

areas[_area_id].idEvents.push(_id);

return _id;
}

function editEventFull(uint256 _event_id,string memory name
memory description,string memory _startDate,string memory
_endDate,uint256 area_id,uint256 state_id) public payable
returns (uint256 id) {
    require(
        (areas[events[_event_id].area_id].ownerArea ==
        msg.sender) ||
        (ownerOrg == msg.sender)
    );
    require((state_id < states.length));
    require((area_id < areas.length));

    if (bytes(name).length > 0) {
        events[_event_id].name = name;
    }
    if (bytes(description).length > 0) {
        events[_event_id].description = description;
    }

    if (bytes(_startDate).length > 0) {
        events[_event_id].startEvent = _startDate;
    }
    if (bytes(_endDate).length > 0) {
        events[_event_id].endEvent = _endDate;
    }
}
```

```

        events[_event_id].state_id = state_id;

        if (events[_event_id].area_id != area_id) {
            uint256 leng = areas[events[_event_id].area_id]
                .idEvents.length;
            for (uint256 i = 0; i < leng; i++) {
                if (
                    areas[events[_event_id].area_id]
                        .idEvents[i] ==
                    events[_event_id].area_id
                ) {
                    areas[events[_event_id].area_id]
                        .idEvents[i] = 0;
                    break;
                }
            }
            areas[area_id].idEvents.push(_event_id);
            events[_event_id].area_id = area_id;
        }

        return _event_id;
    }

    //*****Document*****/
    function addDocumentEvent(string memory _idHash,uint256
        _event_id,uint256 _state_id,string memory _reasonState,
        uint256 _expiration) public payable {
        require(
            (areas[events[_event_id].area_id].ownerArea ==
            msg.sender) ||
            (msg.sender == ownerOrg)

```



```

    );
    require((states.length > _state_id) && (_state_id > 0))
    //only idHash not exists
    require(bytes(documents[_idHash].idHash).length == 0);

    Document memory _newDocument =
        Document(_idHash, _state_id, _event_id, _expiration,
            _reasonState, "");
    documents[_idHash] = _newDocument;
    events[_event_id].idDocuments.push(_idHash);
}

function addAllDocumentsEvent(string[] memory hashid,
uint256 _event_id, uint256 _state_id, string memory
_reasonState, uint256 _expiration) public payable {
    require(
        (areas[events[_event_id].area_id].ownerArea == msg.
            sender) || (msg.sender == ownerOrg)
    );
    require((states.length > _state_id) && (_state_id > 0))
    //only idHash not exists
    require(hashid.length > 0);
    bytes memory tempEmptyStringTest;
    for (uint256 i = 0; i < hashid.length; i++) {
        tempEmptyStringTest = bytes(documents[hashid[i]]
            .idHash);
        if (tempEmptyStringTest.length == 0) {
            Document memory _newDocument =
                Document(hashid[i], _state_id, _event_id,
                    _expiration, _reasonState, "");
            documents[hashid[i]] = _newDocument;
            events[_event_id].idDocuments.push(hashid[i]);
        }
    }
}

```

```

        }
    }
}

function editAllDocumentsEvent(string[] memory hashid,
uint256 _event_id,uint256 _state_id,string memory
_reasonState,uint256 _expiration) public payable {
    require(
        (areas[events[_event_id].area_id].ownerArea ==
        msg.sender) || (msg.sender == ownerOrg)
    );

    require((states.length > _state_id));
    //only idHash not exists
    require(hashid.length > 0);
    uint256 c = 0;
    string[] memory hashidFiltro = new string[](hashid
    .length);

    for (uint256 i = 0; i < hashid.length; i++) {
        if (bytes(documents[hashid[i]].idHash).length > 0) {
            if (
                areas[events[documents[hashid[i]].event_id]
                .area_id].ownerArea == msg.sender) {
                hashidFiltro[c] = hashid[i];
                c++;
            }
        }
    }
    hashid = new string[](c);

    for (uint256 i = 0; i < hashid.length; i++) {

```

```
        hashid[i] = hashidFiltro[i];
    }

    for (uint256 i = 0; i < hashid.length; i++) {
        //add only not exists

        string memory hashAux;
        //delete passed event
        for (
            uint256 j = 0;
            j < events[documents[hashid[i]].event_id]
                .idDocuments.length;
            j++
        ) {
            hashAux = events[documents[hashid[i]].event_id]
                .idDocuments[j];

            if (keccak256(bytes(hashAux)) == keccak256(
                bytes(hashid[i]))) {
                events[documents[hashid[i]].event_id]
                    .idDocuments[j] = "";
                break;
            }
        }
    }

    documents[hashid[i]].event_id = _event_id;
    events[_event_id].idDocuments.push(hashid[i]);

    documents[hashid[i]].state_id = _state_id;

    documents[hashid[i]].reasonState = _reasonState;
    documents[hashid[i]].expiration = _expiration;
```

```

    }
}

//change state_id and reason of the state
function changeStateDocument(string memory _idHash,uint256
_state_id,string memory _reasonState) public payable {
    require(
        (areas[events[documents[_idHash].event_id].area_id]
        .ownerArea ==
            msg.sender) || (msg.sender == ownerOrg)
    );
    require((states.length > _state_id));
    //only idHash not exists
    require(bytes(documents[_idHash].idHash).length != 0);

    documents[_idHash].reasonState = _reasonState;
    documents[_idHash].state_id = _state_id;
}

//mando valores para la version vieja, asi modifico si quiero
//por ejemplo cambiar el estado o cambiar la razon del estado,
function newVersionDocument(string memory _idHash_old,string
memory _idHash_new) public payable {
    require(bytes(documents[_idHash_old].idHash).length > 0);
    require(bytes(documents[_idHash_new].idHash).length > 0);
    require(keccak256(bytes(_idHash_old)) != keccak256(bytes(
        _idHash_new)));
    documents[_idHash_old].newDocument = _idHash_new;
}

/*****Getters of attributes*****/

```

```
function getOrganisation() public view returns (string memory) {
    return organisation;
}

function getOwnerOrg() public view returns (address) {
    return ownerOrg;
}

function getState(uint256 _id) public view returns (uint256,
string memory name, string memory mean)
{
    return (states[_id].id, states[_id].name, states[_id]
.mean);
}

function getLengthStates() public view returns (uint256) {
    return states.length;
}

function getAreaOfOwner(address _id, uint256 _area_index)
public view returns (uint256)
{
    return ownerArea[_id][_area_index];
}

function getLengthAreaOfOwner(address _id) public
view returns (uint256) {
    return ownerArea[_id].length;
}

function getArea(uint256 _id) public view returns
( uint256 id, address owner, string memory name,
```

```
string memory description, uint256 state_id,
uint256 cantEvents)
{
    return (
        areas[_id].id,
        areas[_id].ownerArea,
        areas[_id].name,
        areas[_id].description,
        areas[_id].state_id,
        areas[_id].idEvents.length
    );
}

function getLengthAreas() public view returns
(uint256) {
    return areas.length;
}

function getLengthEventsOfArea(uint256 _id_area)
public view
returns (uint256)
{
    return areas[_id_area].idEvents.length;
}

function getEventOfArea(uint256 _id_area, uint256
_id_event_index) public view returns (uint256)
{
    return areas[_id_area].idEvents[_id_event_index];
}

function getAllEventsOfArea(uint256 _id_area) public
```

```
view returns (uint256[] memory)
{
    return areas[_id_area].idEvents;
}

function getEvent(uint256 _id) public view returns
(uint256 id,string memory name,string memory description,
uint256 state_id,uint256 area_id,string memory startEvent,
string memory endEvent)
{
    return (
        _id,
        events[_id].name,
        events[_id].description,
        events[_id].state_id,
        events[_id].area_id,
        events[_id].startEvent,
        events[_id].endEvent
    );
}

function getCantDocumentEvent(uint256 _id) public view
returns (uint256) {
    return (events[_id].idDocuments.length);
}

function getLengthEvents() public view returns (uint256) {
    return events.length;
}

function getDocument(string memory _idHash) public view
returns (string memory idHash,uint256 state_id,
```

```
uint256 event_id,string memory reasonState,uint256
expiration,string memory newDocument)
```

```
{
    return (
        documents[_idHash].idHash,
        documents[_idHash].state_id,
        documents[_idHash].event_id,
        documents[_idHash].reasonState,
        documents[_idHash].expiration,
        documents[_idHash].newDocument
    );
}
```

```
function getDocumentsOfEvent(uint256 _id_event) public
view returns (string[] memory)
```

```
{
    return events[_id_event].idDocuments;
}
```

```
function checkDocument(string memory _idHash) public
view returns (bool) {
```

```
    return bytes(documents[_idHash].idHash)
        .length > 0 ? true : false;
}
```

```
function checkDocuments(string[] memory idHashes)
public view returns (bool[] memory)
```

```
{
    bool[] memory checks = new bool[](idHashes.length);
    for (uint256 i = 0; i < idHashes.length; i++) {
        checks[i] = bytes(documents[idHashes[i]].idHash)
            .length > 0;
    }
}
```



```
        }  
        return checks;  
    }  
}
```