UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS

TESIS DE GRADO

MODELO DE CERTIFICACIÓN DE CONTRATOS INTELIGENTES APLICANDO LA TECNOLOGÍA BLOCKCHAIN CASO: (CERTIFICACIONES CITES)

Para Optar al Título de Licenciatura en Ingeniería de Sistemas

MENCIÓN: INFORMÁTICA Y COMUNICACIONES

POSTULANTE: Raquel Apaza Alberto

TUTOR METODOLÓGICO: M.Sc. Enrique Flores Baltazar

TUTOR ESPECIALISTA: Ing. Ramiro Kantuta Limachi

TUTOR REVISOR: Ing. Elías Carlos Hidalgo Mamani

EL ALTO – BOLIVIA
2020

DEDICATORIA

A Dios por darme vida, salud, fuerza, fortaleza para terminar esta tesis de grado.

A mis padres Domingo e Rosminda Juana que sin ellos no hubiera logrado una meta más en mi vida profesional, gracias por todo el amor cariño apoyo a lo largo de mi vida.

A mis hermanas Lizet Leticia y Dayana por el apoyo que me brindaron siempre.

A mis tutores, amigos y a todos aquellos que me apoyaron en el desarrollo de este trabajo.

AGRADECIMIENTOS

Agradecer a Dios por todas las bendiciones recibidas por darme salud fuerza para poder lograr la culminación de mi tesis de grado, por la confianza en seguir adelante y cumplir mi meta.

Agradecer con mucho afecto a mis distinguidos Tutores:

A mi Tutor Metodológico M.Sc. Ing. Enrique Flores Baltazar por compartir sus conocimientos, brindarme sus orientaciones, sugerencias con paciencia y motivación durante el desarrollo de la tesis de grado.

A mi Tutor Especialista Ing. Ramiro Kantuta Limachi por su apoyo, confianza, tiempo y motivación brindada.

A mi Tutor Revisor Ing. Elías Carlos Hidalgo Mamani por su disponibilidad de tiempo, su acertada orientación y observaciones brindadas en la revisión de la tesis de grado.

A la Universidad Pública de El Alto por haberme brindado una formación académica.

Y finalmente a mis amigos y compañeros, por los momentos compartidos en aulas a lo largo de estos años y de esta manera lograr realizar nuestras metas.

RESUMEN

La presente tesis de grado se desarrolló para aplicar la tecnología Blockchain en la emisión de certificación cites de permisos de importación, exportación, reexportación de especies de nuestro país, en atención a la problemática de un modelo de certificación de contratos inteligentes poco eficiente.

Para la certificación cites se tiene un gran movimiento de información y documentación mismo que son certificados mediante un sistema limitado a falta de funciones que faciliten los contratos dificultando varios factores, entre los cuales se encuentran falsificación, lentitud, provocando retrasos en la emisión de la certificación.

Es sí que se plantea desarrollar e implementar un sistema de apoyo a la certificación Cites con la utilización de la metodología Uwe(uml-based web engineering), y herramientas de desarrollo de aplicación web como lenguaje de programación phyton, javascript, framework-bootstrap, framework-codelgniter framework-flask, json, Asimismo, para determinar la calidad del sistema web desarrollado, se hace uso de los factores de calidad iso/iec 25000

Finalmente se realiza el análisis de costos utilizando Cocomo II y su modelo de estimación post arquitectura el cual es más detallado y se aplica cuando la arquitectura del proyecto está completamente definida.

Palabras Claves: Modelo, Certificación, cites, Blockchain, seguridad, contratos inteligentes, criptografía, hash Sha-256.

ABSTRACT

This undergraduate thesis was developed to apply Blockchain technology in the issuance of citations certification of import, export, and re-export permits for species in our country, in response to the problems of an inefficient smart contract certification model.

For citations certification there is a great movement of information and documentation that are certified through a limited system in the absence of functions that facilitate contracts, hindering various factors, among which are falsification, slowness, causing delays in the issuance of certification.

It is indeed proposed to develop and implement a support system for Cites certification with the use of the Uwe methodology (uml-based web engineering), and web application development tools such as the phyton programming language, javascript, framework-bootstrap, framework-codelgniter framework-flask, json, Also, to determine the quality of the developed web system, use is made of the quality factors iso / iec 25000

Finally, the cost analysis is performed using Cocomo II and its post-architecture estimation model, which is more detailed and is applied when the project architecture is fully defined.

Keywords: Model, Certification, citations, Blockchain, security, smart contracts, crypto, Sha-256 hash.

ÍNDICE GENERAL

CAPÍTULOI:MARCO PRELIMINAR **Pág.**

ÍNDICE DE FIGURAS

CAPITULO II **Pág.**

CAPITULO III

ÍNDICE DE TABLAS

CAPITULO I **Pág.**

CAPITULO II

CAPITULO III

CAPITULO I

MARCO PRELIMINAR

1.1. INTRODUCCIÓN

En la actualidad las organizaciones han comprendido la importancia de implementar algún sistema como medio de apoyo a la certificación de contratos el cual les sirva como soporte para la toma de decisiones en la emisión correspondiente y tener control sobre los datos y la información de diversos documentos que se ejecutan por lo que permite mejora en disponibilidad y confiabilidad.

En los últimos años, con la aparición del bitcoin, ha supuesto una revolución no sólo en el ámbito económico, en el que ha tenido un éxito notable afianzándose como “moneda” alternativa y generado incluso la aparición de otras monedas virtuales o criptomonedas, sino también en el ámbito tecnológico donde su protocolo Blockchain se ha establecido como el referente del concepto de registro distribuido e inmutable, reduciendo costos velocidad, alcance. En enero de 2009, surgió una tecnología que muchos expertos llaman “la segunda generación de internet”, la consideran la tecnología más relevante de “la cuarta revolución industrial” y creen que tendrá un impacto social tan grande como lo tuvo el internet, esta tecnología se llama Blockchain, o cadena de bloques. Una de sus aplicaciones más relevantes los Smart Contracts. (Sanchez, 2018)

Un “contrato inteligente” es un código de computadora que se ejecuta en la parte superior de una cadena de bloque, la cual contiene un conjunto de reglas bajo las cuales las partes de ese “contrato inteligente” acuerdan interactuar entre sí. Por tanto, si se cumplen las reglas predefinidas, el acuerdo se aplica automáticamente. El código de “contrato inteligente” facilita, verifica y hace cumplir la negociación o ejecución de un acuerdo o transacción.

La Cites, a través del ministerio del medio ambiente realizan la emisión de un certificado cite ya sea de importación, exportación, reexportación u otros de especies amenazadas de fauna y flora de nuestro país. Ya que para la solicitud se controla la documentación, requisitos que deben cumplir para obtener el

certificado. Por ello, el desarrollo de este trabajo de investigación, tiene como finalidad, plantear un modelo de certificación de contratos inteligentes aplicando la tecnología Blockchain para mejorar el servicio de la emisión de certificaciones cites y así contar con rapidez mayor seguridad.

Para el desarrollo de este modelo se aplica la metodología de desarrollo UWE, porque está basada en el Proceso Unificado y UML para el desarrollo de aplicaciones Web, utilizando herramientas como el gestor de base de datos PostgresSQL, Manejador de Base de Datos (PGADMIN), Servidor Web Apache, Framework-Bootstrap, Lenguaje de programación Php, Json, Lenguaje de Programación Python, Framework Flask SQLAlchemy, url_for, render_template, request, redirect, sesión, datetime, json.

1.2. ANTECEDENTES

1.2.1. Antecedentes Académicos

Para la presente investigación se realiza un análisis de trabajos de investigación desarrollados sobre este tema, obteniendo información que se relaciona con el trabajo de investigación.

Internacionales

- (Alvarez, 2016). “Diseño y desarrollo de propiedades inteligentes aplicación de la blockchain a Internet de las Cosas”. Implementar un sistema que mantenga un registro de propiedades público y distribuido, haciendo uso de la cadena de bloques de Bitcoin para mantener el registro de los datos y del protocolo Open Assets para representar, mediante monedas digitales, la emisión y transferencia de activos, mantener una copia propia de la cadena de bloques de Bitcoin, que será verificada y se utilizará para extraer los datos relativos a las propiedades. Universidad Autónoma de Madrid Escuela Politécnica Superior.

- (Urdaneta, 2018). “Diseño de la arquitectura de un sistema de contratos inteligentes basada en la tecnología Blockchain”. Diseñar la arquitectura de un sistema de contratos inteligentes, basada en la tecnología de Blockchain, que permita la gestión del registro de personas en el sistema de educación colombiano para reducir el riesgo de fraude en el proceso de registro de personas en el sistema de educación en colombiano. Colombia.

Nacionales

- (Koller, 2017). “Modelo de votación electrónico con Blockchain”. El sistema de votación electrónica que él propone puede beneficiar a la democracia, pues hay seguridad en la integridad de los datos, mantiene el anonimato y hay transparencia en la información. Entonces, se podría contar con una votación segura e infraudable. Ese voto automáticamente se replica en todas las mesas, por lo cual desde el momento en que se emitió no va a poder ser cambiado nunca. Cochabamba: Bolivia.
- (Morales, 2017). “Sistema para la gestión de historias clínicas con Blockchain”. Muchas veces un asegurado necesita hacerse atender en otro departamento al que reside o acudir a servicios externos del Seguro Universitario, es necesario contar con las historias clínicas completas resguardando la confidencialidad, integridad y disponibilidad de los datos. Explica que con este sistema los médicos podrán tener las historias clínicas completas cuando lo necesiten y asegurar el intercambio confiable de información entre ellos, evitando que sea modificada por centros o personas sin autorización. Cochabamba: Bolivia

1.3. PLANTEAMIENTO DEL PROBLEMA

1.3.1. Problema Principal

Para la emisión de certificación cites existe un gran movimiento de información y documentación respecto a la importación, exportación, reexportación de

animales, plantas vivos o muertos de nuestro país, los mismos son registrados llenando un formulario y controlando a través de un sistema web de certificación cites ya que cuenta con falencias inseguridad. Lo mencionado se analiza en un árbol de problemas, el cual se describe en la Anexo A.1.

De Anexo A.1, se observa que el problema central se encuentra en la inadecuación del proceso de emisión de certificación cites el cual es producido por las siguientes causas y sus respectivos efectos: identificación de dificultad en el proceso de emisiones de certificaciones cites si bien se tiene un sistema cites (sistema que se maneja actualmente) en uso el cual tienen las funciones de aprobar su petición de emitir un certificado, por lo cual se tiene la dependencia y manipulación de datos durante los procesos el cual provoca modificaciones, falsificaciones por lo que existe inseguridad de información.

¿De qué manera mejorar el proceso de emisión de certificación CITES de contratos inteligentes con la tecnología Blockchain y que se beneficie a los clientes adquiriendo así el certificado cite de forma rápida segura?

1.3.2. Problemas Secundarios

- Falta de un modelo de contratos inteligentes para las emisiones de certificaciones cites por riesgo perdida de información, costos, demora.
- Falta de un sistema de certificación vía web aplicando Blockchain.
- Actualmente no se está implementando la Blockchain debido a que no se investiga más sobre el avance de la tecnología.
- Riesgo de falsificación y alteración de la información por qué no se está viendo el uso de cifrado, criptografía, hash.
- No aplicación de nuevas tecnologías debido a falta de interés sobre la revolución tecnológica que es lo que nos espera más adelante que se aplica en otros países tecnológicos.

1.4. OBJETIVOS

El objetivo del presente trabajo de investigación nace con una respuesta a las necesidades problemáticas con las que se cuenta, luego de un análisis de los problemas y su relación causa-efecto se realizó el árbol de objetivos (Ver Anexo A.2).

1.4.1. Objetivo General

Desarrollar un modelo de certificación basado en contratos inteligentes con la tecnología Blockchain, para mejorar el funcionamiento de emisiones de certificados CITES.

1.4.2. Objetivos Específicos

- Implementar los contratos inteligentes para las certificaciones cites.
- Implementar un sistema de certificación vía web con Blockchain.
- Diseñar la arquitectura de solución para la generación de certificados digitales.
- Estudiar los algoritmos criptográficos que contribuyen a la seguridad.
- Realizar un estudio sobre la tecnología Blockchain.

1.5. HIPÓTESIS

Con la ingeniería de software y la tecnología Blockchain se obtendrá un modelo de certificación de contratos inteligentes aplicado a Cites con una eficiencia del 90%.

1.5.1. Identificación de variables

Variable Independiente: Tecnología Blockchain.

Variable Dependiente: Modelo de certificación cites.

Variable Interviniente: Seguridad de Bloques.

1.5.2. Operacionalización de variables

Tabla N° 1.1: Operacionalización de variables independientes

Variables	Definición Conceptual	Dimensión	Indicadores	Instrumento
Tecnología Blockchain	Cadena de bloques tecnología informática usada para generar y conservar un registro certero y verificable de acontecimientos digitales en el que se incluyen todas las transacciones.	Uso de bloques.	Uso de nuevas Tecnologías.	Acceso a internet.
		Información confiable.	Investigación sobre Blockchain.	Investigación.

Nota. - Elaboración propia

Tabla N° 1.2: Operacionalización de variables intervinientes

Variables	Definición Conceptual	Dimensión	Indicadores	Instrumentos
Seguridad de Bloques	Registro de Transacciones encriptadas, cifradas.	Uso de criptografía, hash.	Seguridad de datos.	Observación

Nota. - Elaboración propia

En la Tabla 1.3. Se muestra la operacionalización de variables dependientes de la hipótesis.

Tabla N° 1.3: Operacionalización de variables dependientes

Variables	Definición Conceptual	Dimensión	Indicadores	Instrumentos
Modelo	Representación de un objeto, su propósito es ayudar a explicar, entender o mejorar un sistema.	Uso de Librerías.	Instrucciones, pasos a seguir.	Observación
Certificación Cites	Convención sobre el comercio internacional de especies amenazadas de fauna y flora.	Uso de documentos.	Interfaz Navegabilidad Interactividad	Encuesta Cuestionario Entrevistas

Nota. - Elaboración propia

1.6. JUSTIFICACIONES

1.6.1. Justificación Técnica

Este trabajo se justifica técnicamente ya que actualmente no se cuentan con este tipo de modelo en nuestro país por falta de investigación no se realiza el uso de esta tecnología, con este modelo de la Blockchain se podrá aplicar a la certificación cites contratos inteligentes, ya que busca utilizar las características de seguridad, privacidad, seguimiento planteadas en la tecnología de Blockchain ya que fortalece la integridad de la información y la transparencia operacional.

1.6.2. Justificación Social

El presente trabajo tiene un impacto social asociado puesto que propone una herramienta con la capacidad de solucionar necesidades de los clientes para la emisión de certificación cites que permitan obtener información rápida mediante la herramienta tecnológica Blockchain para mejorar este proceso y así brindar seguridad, confidencialidad.

1.6.3. Justificación Científica

Este trabajo se justifica científicamente con el aporte de la tecnología Blockchain al caso de emisión de certificación cites de importación, exportación, reexportación.

1.7. METODOLOGÍA

- **Metodología Científico**

  El método científico es una creación humana, creada artificialmente para crear conocimiento científico, puesto que el hombre no está dotado de manera natural para conocer científicamente. Es el camino para producir conocimiento objetivo, es un modo razonado de indagación establecido en forma deliberada y sistemática, que está constituido por una serie de etapas o pasos para producir conocimiento.

  Las etapas a utilizar son:

  1. Planteamiento del problema: ¿Cómo se puede apoyar el proceso de emisión de certificación CITES de contratos inteligentes con la tecnología Blockchain?
  2. Composición del Marco Teórico: Comprobación recogida de datos.
  3. Formulación de la Hipótesis: Con la ingeniería de software y la tecnología blockchain se obtendrá un modelo de certificación de contratos inteligentes aplicado a CITES con una eficiencia del 90%.

4. Constatación de la hipótesis: Variable Independiente: Tecnología Blockchain, Variable Dependiente: Modelo de certificación cites, Variable Interviniente: Seguridad de Bloques.
5. Conclusiones y resultados: Implementar un modelo de certificación aplicando la tecnología blockchain y contratos inteligentes aplicado a la certificación CITES. (Asuad, 2014)

- **Metodología UWE**

  Metodología UWE es una metodología detallada para el proceso de autoría de aplicaciones con una definición exhaustiva del proceso de diseño que debe ser utilizado, este proceso, iterativo e incremental, incluye flujos de trabajo y puntos de control, y coinciden con las propuestas en el Proceso Unificado de Modelado. Cuentan estas fases:

  1. Captura, análisis y especificación de requisitos
  2. Diseño del sistema
  3. Codificación del software
  4. Pruebas
  5. La Instalación o Fase de Implementación
  6. El Mantenimiento. (Castro, 2014)

- **Métricas de Calidad de Software**

  Principal objetivo de los ingenieros de software es producir sistemas, aplicaciones de alta calidad. Para las evaluaciones que se quieran obtener.

  Iso/Iec 25000- SQuaRE: ISO 25000 es una familia de normas que tiene como objetivo la creación de un marco de trabajo común para evaluar la calidad de un producto de software. (Molina, 2015). Se encuentra compuesta por cinco divisiones:

1. La ISO/IEC 2500n-División de gestión de calidad
2. La ISO/IEC 2501n-División de modelo de calidad
3. La ISO/IEC 2502n-División de mediciones de calidad
4. ISO/IEC 2503n-División de requisitos de calidad.
5. La ISO/IEC 2504n-División de evaluación de calidad

- **Modelo de calidad del producto (calidad interna y externa):** Categoriza propiedades de calidad de producto del sistema/software en ocho características. (Sierra, 2017).

    1. Adecuación Funcional
    2. Eficiencia de Rendimiento
    3. Compatibilidad
    4. Usabilidad
    5. Confiabilidad
    6. Seguridad
    7. Mantenibilidad
    8. Portabilidad

- **Modelo Constructivo de Costos**

    Modelo de formulación matemática con un fuerte componente de base empírica, principalmente utilizado para estimación de costos en los proyectos de software el modelo está orientado a la magnitud del producto final, está basado en estimaciones matemáticas, mide el “tamaño” del proyecto y utiliza las líneas de código como unidad de medida. Dos de los aspectos fundamentales del modelo COCOMO son los submodelos y los modos de desarrollo. Incluye submodelos son tres: básico, intermedio y

detallado. Por su parte, los modos de desarrollo son también tres: orgánico, semi-acoplado y empotrado. (Boehm, 1981).

1.8. HERRAMIENTAS

- ✓ **Gestor de base de datos PostgresSQL**

  PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacionales. PostgreSQL almacena los datos de usuarios, así como también los datos de los grupos dentro de sus propios catálogos de sistema. De esta manera, cualquier conexión a PostgreSQL debe ser realizada con un usuario específico, y cualquier usuario puede pertenecer a uno o más grupos definidos. (Garavito, 2007)

- ✓ **Manejador de Base de Datos (Pgadmin)**

  PgAdmin es una aplicación con interfaz gráfica para gestionar bases de datos PostgreSQL, siendo la más completa y popular con licencia Open Source. Está diseñado para darle respuesta a las necesidades de la mayoría de los usuarios, desde la escritura de consultas simples en SQL hasta el desarrollo de bases de datos complejas. (Ortiz, 2013)

- ✓ **Framework-Bootstrap**

  Es un Framework de twitter para desarrollo de aplicaciones web, sencillo y ligero basta con un fichero CSS y uno JavaScript. Basado en los últimos estándares de desarrollo de Web HTML5, CSS3 y JavaScript/JQuery, Plugins de jQuery. (Pavon J. , 2014).

- ✓ **Lenguaje de Programación Python**

  Python es un lenguaje de código abierto, de alto nivel y multiparadigma, soporta orientación a objetos, programación imperativa y programación funcional. Al ser multiplataforma puede ser ejecutado en un windows, un mac, un linux, un teléfono móvil, una raspberry pi o controlar una placa de arduino. (Hashmania, 2019)

- ✓ **Lenguaje de Programación Php**

  PHP: **H**ypertext **P**reprocessor Lenguaje de script que se ejecuta en el servidor el resultado se devuelve al navegador como HTML capacidad para generar páginas de contenido dinámico, HTML, imágenes, videos, PDF, XML, etc. crear, abrir, leer, escribir, cerrar ficheros en el servidor, Utilizar bases de datos, procesar datos de formularios enviar y recibir cookies control de acceso de usuarios al sitio web encriptar datos. (Pavon J. , 2013)

- ✓ **Framework Flask**

  Flask es un framework minimalista escrito en Python, concebido para facilitar el desarrollo de Aplicaciones Web bajo el patrón MVC, de forma rápida y con un mínimo número de líneas de código. Flask es simple y se limita a lo que toda aplicación Web necesita: Un sistema de enrutamiento y un sistema de platillas. (Saavedra, 2018)

- ✓ **Json**

  JSON es un formato de datos muy ligero basado en un subconjunto de la sintaxis de JavaScript: literales de matrices y objetos. Como usa la sintaxis JavaScript, las definiciones JSON pueden incluirse dentro de archivos JavaScript y acceder a ellas sin ningún análisis adicional como los necesarios con lenguajes basados en XM. (Quiroga, 2011).

1.9. LÍMITES Y ALCANCES

1.9.1. Limites

El modelo diseñado realiza el uso de la tecnología Blockchain para la emisión de certificación cites ya sea de importación, exportación, reexportación, no se desarrolla el uso de otras tecnologías.

1.9.2. Alcances

La presente tesis tiene como alcance la implementación de un modelo de certificación aplicando la tecnología Blockchain y contratos inteligentes orientado a la certificación CITES.

1.10. APORTES

Este trabajo nos permite aplicar el modelo mediante la investigación, análisis, pruebas e implementación, aplicados a la certificación CITES.

- Con el uso de la tecnología Blockchain se enviará información de manera rápida, segura, confiable.
- Blockchain usará la criptografía ya que se enviará información encriptada.

CAPITULO II

MARCO TEÓRICO

CONCEPTUAL

2.1. INTRODUCCIÓN

En este capítulo tiene por objetivo construir el marco teórico conceptual sobre el cual se sustenta la investigación, conocer los elementos conceptuales del presente trabajo, realizar el uso de herramientas tecnológicas que actualmente pueden ser utilizadas, que tiene la finalidad de construir un modelo optimo que ayude el proceso de emisiones de certificaciones cites aplicando la tecnología Blockchain los Smart contracts se describe a continuación una breve conceptualización.

CITES es la Convención sobre el Comercio Internacional de Especies Amenazadas de Fauna y Flora Silvestres. También se conoce como el Convenio de Washington, pues de firmó en Washington D.C. La CITES se firmó el 3 de marzo de 1973 y entró en vigor el 1 de julio de 1975. (cites, 2010)

Blockchain como una contabilidad pública de persona a persona que se mantiene a través de una red distribuida y que no requiere ninguna autoridad central ni terceras partes que actúen como intermediario. Como se puede apreciar no hay gran diferencia entre las definiciones expuestas, aunque un hecho interesante es que ha sido considerado como una base de datos, o libro en el que es posible guardar información de manera segura de todos los participantes o usuarios de esta. Cada transacción en la Blockchain se verifica por consenso de la mayoría de los participantes en el sistema. Una vez integrado en el sistema, la información nunca puede ser borrada (Nachiappan, 2016).

Los “smart contracts” también conocidos como contratos digitales o contratos inteligentes, son programas informáticos, basados en criptografía, que facilitan, verifican y hacen cumplir de forma automática la negociación de un contrato sin necesidad de tener un documento contractual. Estos contratos inteligentes ejecutan automáticamente las cláusulas contractuales cuando se cumplen, o incumplen, las correspondientes condiciones pre-programadas, asociadas a activos reales. El programa puede definir reglas estrictas y sus consecuencias, de la misma manera que un documento legal tradicional. Pero a diferencia de

este, puede hacer uso de la información del sistema y tomar las acciones necesarias en base a lo descrito en el contrato. (Sanchez, 2018)

2.2. ANTECEDENTES DE LA CITES

CITES (convención sobre el comercio internacional de especies amenazadas de fauna y flora silvestres) es un acuerdo internacional al que los países a través de sus gobiernos se adhieren voluntariamente como aquellas organizaciones de integración económica internacional, formando parte de esta institución. La misma tiene la finalidad de velar por el comercio internacional de especímenes de animales y plantas silvestres que no establezcan un peligro para su supervivencia.

Las Partes al adherirse a la CITES deben aplicar los mandatos de la Convención, promulgando su propia legislación nacional para garantizar que la CITES se aplique a escala nacional. Estos países ofrecen el marco legislativo para el desenvolvimiento de la convención, sin menoscabar su legislación.

Los objetivos estratégicos institucionales de los cites son: regular el comercio internacional de especímenes de especies de fauna y flora silvestres con algún grado de amenaza, es decir regular su exportación, importación o re-exportación.

2.2.1. Estructura Organizacional

La estructura de organización (Ver Anexos A.3.) consta de niveles jerárquicos los cuales son:

- Las Partes (Países miembros) en la Cites se denominan colectivamente como la conferencia de las partes, cada dos a tres años, se reúnen para examinar la aplicación de la convención.
- El Comité Permanente
    - Brinda orientación política a la Secretaría CITES (aplicación de la Convención)
    - Supervisa la administración del presupuesto de la Secretaría.

- Coordina y supervisa la labor de los comités y los grupos de trabajo
- Realiza otras tareas encomendadas por la Conferencia de las Partes
- Prepara proyectos de resolución para presentarlos a la consideración.

- Los comités de Fauna y Flora son comités de expertos, que aportan en los conocimientos biológicos y especializados sobre las especies de fauna y flora que están (o podrían estar) sujetas a controles comerciales CITES. Su finalidad es proporcionar apoyo técnico en la toma de decisiones sobre estas especies. Secretaría CITES con sede en Ginebra, Suiza es administrada por PNUMA.
- La secretaria:
  - Coordina, y asesora el funcionamiento de la Convención
  - Asiste en las comunicaciones y el control de aplicación de la Convención para garantizar que se respetan las disposiciones
  - Publica nuevas ediciones de los Apéndices I, II y III, cuando hay modificaciones.
  - Organiza reuniones de la Conferencia de las Partes y de los comités de carácter permanente.
  - Formular recomendaciones sobre la aplicación de la Convención
  - Distribuye información a las Partes, mediante Notificaciones (en español, francés e inglés).

Resoluciones Ministeriales de Autoridad Científica Cites (Ver Anexo A.4.)

Fue firmada el 3 de marzo de 1973 en Washington y entro en vigencia el 1 de julio de 1975. En Bolivia se firma el 23 de diciembre de 1974 y es ratificada mediante Decreto Ley N° 16464 del 17 de marzo de 1979, posteriormente se eleva a rango de Ley N° 1255 de 5 de julio de 1991. En la actualidad 175 países

son parte de esta Convención, acorando adoptar sus disposiciones. (Antezana, 2012).

2.2.2. Tipos de documentos Cites

2.2.2.1. Permisos de Exportación

- Los permisos de exportación sólo pueden ser emitidos por la Autoridad Administrativa, siempre y cuando la Autoridad Científica haya manifestado que esa exportación no perjudicará la supervivencia de esa especie
- La Autoridad Administrativa debe haber verificado que el espécimen fue obtenido legalmente

2.2.2.2. Permisos de Importación

- (Sólo aplicable a especímenes de especies incluidas en el Apéndice I)
- Los permisos de importación únicamente pueden ser expedidos por la Autoridad Administrativa, y después de que la Autoridad Científica haya manifestado que los fines de la importación no serán perjudiciales para la supervivencia de dicha especie
- Nota: al adoptar medidas domésticas más estrictas algunas Partes (por ejemplo, los Estados miembros de la Unión Europea) también requieren documentos de importación para especímenes de especies del Apéndice II.

2.2.2.3. Certificados de Reexportación

Los certificados de reexportación únicamente pueden ser expedidos por la Autoridad Administrativa, y sólo cuando dicha autoridad haya verificado que los especímenes fueron importados de conformidad con las disposiciones de la Convención.

2.2.2.4. Otros certificados

- Éstos se utilizan en algunos casos particulares.
- Especímenes criados en cautividad o reproducidos artificialmente

Figura N° 2.1: Certificado cite

Fuente: cites, s. (2010). Cites

2.2.3. Ventajas Cites

- Regulación internacional del comercio de especies silvestres, efectiva y constante, para su conservación y uso sostenible
- Cooperación internacional sobre comercio y conservación, legislación y cumplimiento, así como gestión de recursos y ciencias de la conservación.
- Participación, como agente global, en la gestión y conservación de las especies silvestres a nivel internacional. (cites, 2010).

Cites convención sobre el comercio internacional de especies amenazadas de la flora y fauna se encarga de controlar bajo un sistema de permisos y certificados que se emitan tras la consecución de criterios claramente predefinidos estos documentos que son de importación exportación reexportación ya que deben ser especificados.

2.3. MODELO

Los modelos son una representación o concepción de la realidad y generalmente simplifican al sistema real. Por ello al establecerlos hay que determinar con claridad, cuales atributos o propiedades se desea analizar o representar, y verificar si dichas propiedades están incorporadas en él. Pueden tener poca o ninguna semejanza con la apariencia real del sistema que representan, pero simbólicamente reproducen los elementos esenciales de esa realidad. El modelo es un esquema, que trata de imitar al mundo real en forma simplificada ayudan en la descripción, análisis y en la fase de solución del problema.

2.3.1. Clasificación de Modelo

- Diagramáticos o explicativos: se utiliza para mostrar la relación entre los distintos componentes de un sistema y cuando se requiere analizar su estructura y funcionamiento.
- Analíticos o matemáticos: si la característica de sus componentes, su estructura y sus interacciones pueden ser matemáticamente definidas.

2.3.2. Etapas a cumplir para el desarrollo del Modelo

- Delimitación del sistema
- Identificación de los elementos constitutivos
- Análisis de sus componentes
- Síntesis final. (Mattion, 2015)

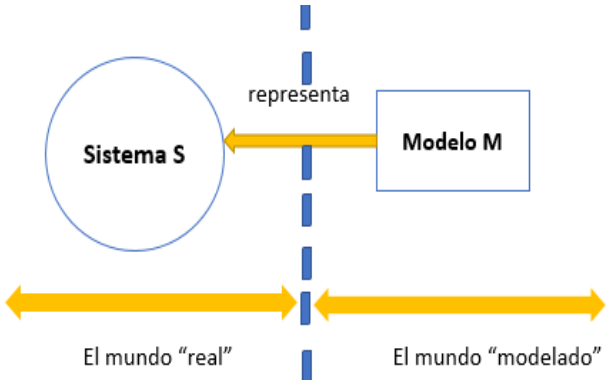


Figura N°2.2: Modelo

Fuente: Elaboración propia

Un modelo es una abstracción de un sistema o entidad del mundo real. Una abstracción es una simplificación, que incluye sólo aquellos detalles relevantes para algún determinado propósito.

- Sistema: Colección de elementos, posiblemente divididos en subsistemas, organizados para lograr un propósito. Está descrito por un conjunto de modelos.
- Modelo: Simplificación completa y auto consistente de la realidad, creado para comprender mejor un sistema.
- Vista (Arquitectural): Proyección de la organización y estructura de un modelo de un sistema, centrada en un aspecto. Incluye un subconjunto de los elementos incluidos en el modelo

- Diagrama: Representación gráfica de un conjunto de elementos del modelo y sus relaciones. En UML generalmente corresponde a un grafo conexo de nodos (elementos) y arcos (relaciones). (Rumbaugh, 2007:5).

Un modelo es una representación de un objeto, sistema o idea, de forma diferente al de la entidad misma su propósito del modelo es ayudar a explicar, entender o mejorar un sistema, puede ser una réplica exacta o una abstracción de las propiedades dominantes del objeto.

2.4. CERTIFICACIÓN

La certificación es un procedimiento destinado a que un organismo independiente y autorizado, valide o dictamine la calidad del sistema aplicado por una organización, partiendo y verificando si la misma cumple o no lo dispuesto por un determinado referencial o modelo de calidad, reconocido y oficial. Es un proceso de evaluación de conformidad, que permite dar como resultado un informe escrito en relación a un producto, una persona, o una organización, asegurando que el mismo responde a ciertos requisitos, características, y/o exigencias. (Wikipedia/Certificacion, 2018).

El certificado es un tipo de texto administrativo empleado para constatar un determinado hecho. En el proceso de solicitud de un puesto de trabajo, en especial si se trata de una institución oficial, los certificados son fundamentales para demostrar la formación y la experiencia. Es un tipo de texto que se produce normalmente a instancias de quien lo recibe, y por una persona con autoridad suficiente dentro de la institución para establecer que se ha cumplido con lo afirmado en el documento. Si llega haber alguna irregularidad o falsedad en lo declarado, puede ser penado por la ley. (Wikipedia/Certificado, 2019)

La certificación es una garantía que se entrega ya que tiene como objetivo afirmar la autenticidad de un documento, la certificación de un contrato que se requiere adquirir en una entidad pública privada.

2.5. CONTRATOS INTELIGENTES O SMART CONTRACT

En 1994, Nick Szabo, un jurista y criptógrafo, se dio cuenta de que un libro mayor descentralizado podría ser utilizado para realizar contratos digitales. En este formato los contratos pueden ser convertidos a código, guardados y replicados en el sistema y supervisados por la red de computadoras que corre el programa blockchain. Los contratos inteligentes nos ayudan a intercambiar dinero, propiedades, activos o cualquier bien de valor de una manera sencilla, evitando los gastos por el servicio de intermediarios y sin revelar ningún tipo de información confidencial sobre las partes y/o naturaleza de la transacción.

- ✓ Autonomía: Es uno mismo quien accede al acuerdo. No existen intermediarios, por lo que también se ahorra dinero.
- ✓ Confianza: Los documentos están encriptados en un shared-ledger. No pueden perderse.
- ✓ Backup: Todos los miembros de la red tienen los documentos duplicados.
- ✓ Rapidez: Ahorro de tiempo en el procesamiento de documentos, papeleos manuales, etc.
- ✓ Seguridad: Obtenida gracias a la criptografía. (Navarro B. , 2016).

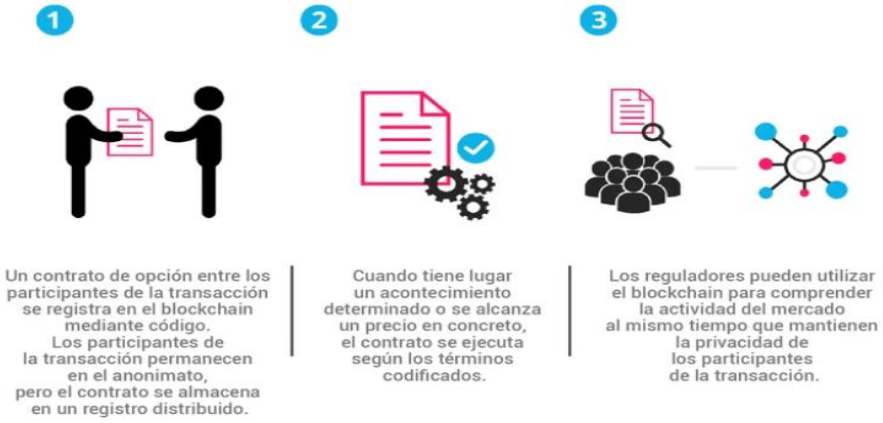


Figura N° 2.3: Ejecución de los Smart Contracts

Fuente: Navarro, B. (2016). Blockchain y sus aplicaiones

Los “SmartContract” representa los contratos inteligentes que se pueden definir en esta Blockchain. Un contrato inteligente en este programa es simplemente una transacción que tiene que ser confirmada tanto por el remitente como por el receptor, y, además, se ejecutará dadas unas condiciones determinadas; en este caso, pasada una fecha. Cada contrato posee:

- Un ID (hash), que al igual que en el caso de las transacciones, se calcula a partir de los demás datos. Además, este ID servirá para evitar que se manipule algún dato del contrato antes de ser ejecutado.
- La fecha en la cual se debe ejecutar el contrato. Este dato se expresa en milisegundos, tomando como referencia el 01/01/1970.
- La cantidad a enviarse.
- Las claves públicas del remitente y del receptor.
- La firma de la transacción pendiente, firmada por el remitente con su clave privada.
- Un número entero “id” que evita que el hash del contrato se repita en caso de que sus datos (fecha, cantidad, remitente, receptor…) sean idénticos a un contrato ya existente. (Stefanescu, 2019).

Los contratos inteligentes, son programas informáticos**.** No están escritos en lenguaje natural, sino en código virtual. Son un tipo de software que se programa, como cualquier otro software, para llevar a cabo una tarea o serie de tareas determinadas de acuerdo a las instrucciones previamente introducidas en el contrato almacenando sus direcciones.

2.6. TECNOLOGÍA BLOCKCHAIN

2.6.1. Blockchain

Blockchain es un registro de información distribuido tipo P2P (*Peer-to-Peer*) en donde los diferentes participantes no tienen por qué confiar los unos en los otros, puesto que hay un protocolo de consenso que garantiza la seguridad y la

veracidad de las transacciones. Otra de las características principales, y sin duda una de las más relevantes, es la inmutabilidad de la cadena; en Blockchain no es posible editar o borrar información. La estructura de este registro, consistente en conjuntos de transacciones que son organizados y almacenados en bloques. Los bloques están ordenados cronológicamente y tienen un número de bloque, un código alfanumérico conocido como *hash* sobre el que profundizaremos más adelante y están firmados digitalmente por la persona que propone o valida el bloque. (Lopez, 2018)

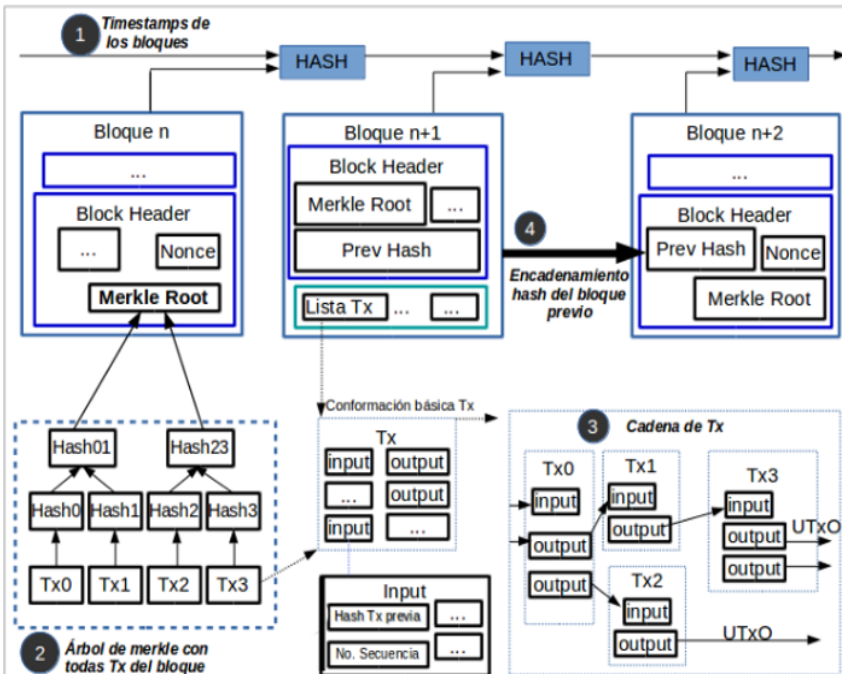


Figura N° 2.4: Estructura de la Cadena de Bloques

Fuente: Albuixech, R. (2018). Estudio de la tecnologia blockchain

1. El servidor P2P distribuido de marca de tiempo (timestamp) se implementa utilizando POW incrementando un nonce en el bloque hasta lograr llegar a un hash con el nivel de dificultad del reto (un hash con una cantidad determinada de ceros). El hash del bloque anterior se guarda en el bock header de cada bloque.
2. El uso de funciones hash para enlazar bloques y transacciones anteriores con un bloque/transacción en particular y el ahorro de espacio en disco

cuando se utiliza el árbol de Merkle con el hash de las transacciones apareadas.

3. Todo input corresponde a un output y el entrelazamiento entre transacciones se realiza a través de un campo en los inputs. Cada output de una transacción solo puede ser referenciado una vez por un input de una transacción subsiguiente.

4. Cada bloque en la cadena contiene un hash del bloque anterior creando así una cadena de bloques desde el primer bloque hasta el último bloque en la cadena más larga, Así se asegura que no puede ser modificado un bloque especifico sin antes modificar el bloque que lo tiene registrado y todos los bloques subsiguientes a este. (Albuixech, 2018)

2.6.2. Funcionamiento de Blockchain

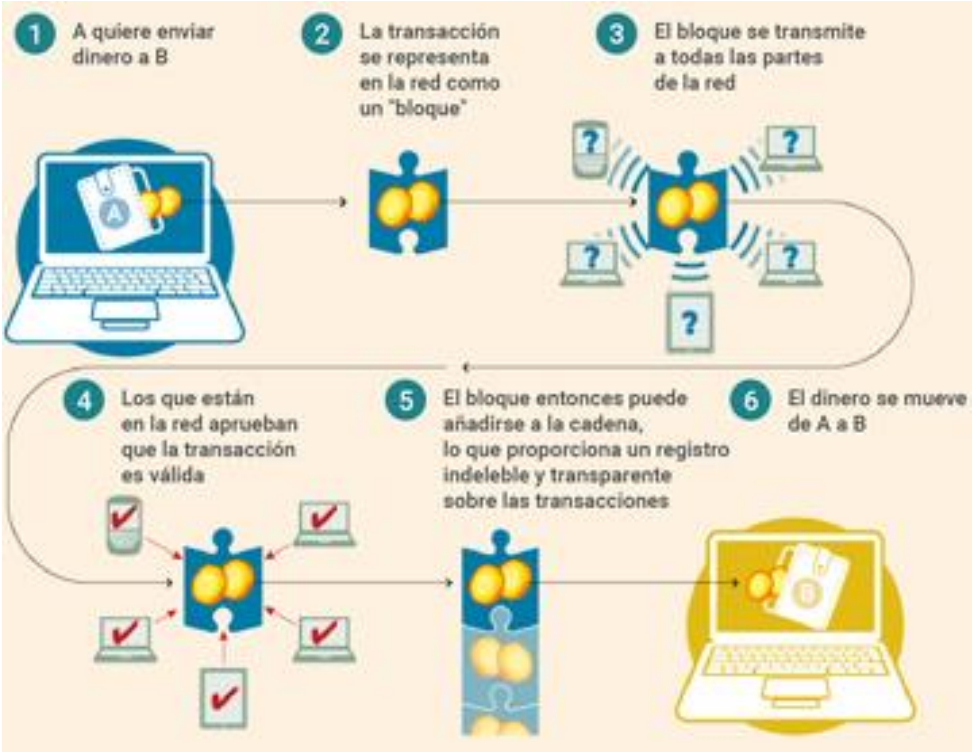


Figura N° 2.5: Funcionamiento de Blockchain

Fuente: Ortega. (2010). Bitcoin

2.6.3. Elementos de Blockchain

Para entender el alcance de la tecnología Blockchain hay que conocer los elementos básicos de que se compone. Son los siguientes:

- **Un nodo:** puede ser un ordenador personal o, según la complejidad de la red, una mega computadora. Con independencia de la capacidad de cómputo, todos los nodos han de poseer el mismo software/protocolo para comunicarse entre sí. De otro modo no podrán conectarse ni formar parte de la red de una Blockchain, sea ésta pública, privada o híbrida. Si en una Blockchain pública estos nodos no tienen por qué identificarse, en una Blockchain privada los nodos se conocen entre sí, pudiendo también ser iguales entre ellos.
- **Un protocolo estándar:** en forma de software informático para que una red de ordenadores (nodos) pueda comunicarse entre sí. Existen protocolos muy conocidos, como el TCP/IP para internet o el SMPT para el intercambio de correos electrónicos. El protocolo de una Blockchain funciona de la misma forma: otorga un estándar común para definir la comunicación entre los ordenadores participantes en la red.
- **Una red entre pares o P2P (Peer-to-Peer, en inglés):** Se trata de una red de nodos conectados directamente en una misma red, elimina la necesidad de un servidor central y la de una relación cliente/servidor, pasando a funcionar como una arquitectura de red entre partes iguales sin jerarquías y distribuidas en las cuales las aplicaciones pueden comunicarse entre sí intercambiando información sin la intervención de un ente controlador central.
- **Un sistema descentralizado:** a diferencia de un sistema centralizado, donde toda la información está controlada por una única entidad, aquí son todos los ordenadores conectados los que controlan la red porque todos son iguales entre sí; es decir, no hay una jerarquía entre los nodos, al menos en una blockchain pública. En una privada sí puede haber jerarquía. (Preukschat, 2017)

2.6.4. Componentes de Blockchain

2.6.4.1. Bloques

Blockchain es un registro de todas las transacciones que se empaquetan en bloques que los mineros luego se encargan de verificar. Luego serán agregadas a la cadena una vez terminada su validación y distribuidas a todos los nodos que forman la red. Un bloque es un conjunto de transacciones confirmadas e información adicional que se ha incluido en la cadena de bloques. Cada bloque que forma parte de la cadena (menos el primer bloque que inicia la cadena) está formado por:

- Un código alfanumérico que enlaza con el bloque anterior
- El “paquete” de transacciones que incluye
- Otro código alfanumérico que enlazará con el siguiente bloque.

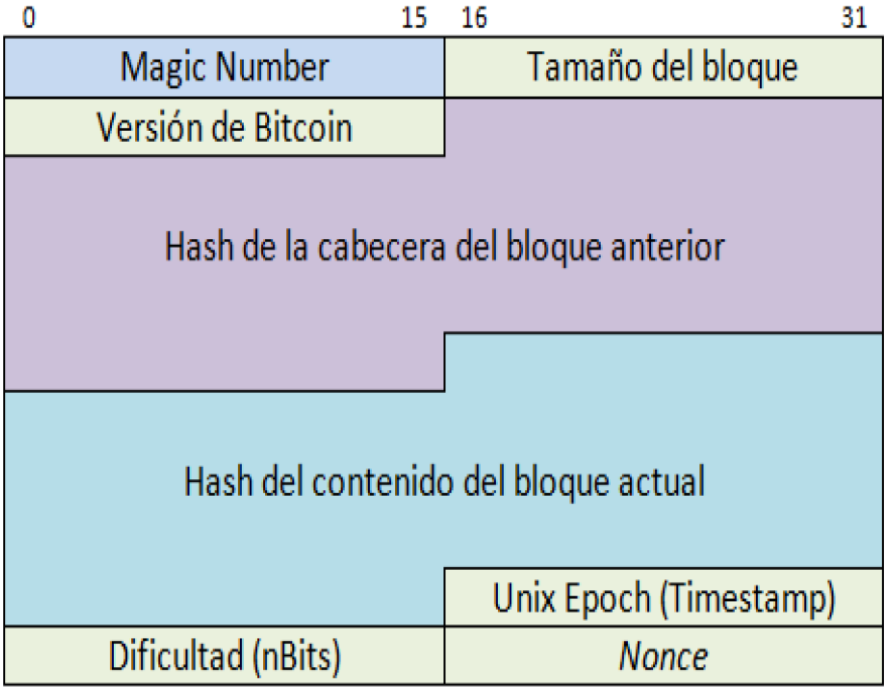


Figura N° 2.6: Estructura del Bloque

Fuente: Albuixech, R. (2018). Estudio de la tecnologia blockchain

La estructura de los bloques de transacción es la siguiente:

- Magic number: Valor por defecto en 0Xd9b4bef911. No es un dato específico del protocolo.
- Blocksize: Cantidad de bytes del bloque.
- Blockheader: Cabecera del bloque y cadena. Incluye los datos siguientes:
  - Version: Versión del bloque.
  - HashPrevBlock: Hash del bloque anterior.
  - HashMerkleRoot: Hash de la raíz del árbol de Merkle.

    Hash->SHA256(Magic number, Nonce, Data, Time, HashPrevBlock)
  - Time: Marca de tiempo de creación del bloque.
  - Bits: Especificación de la complejidad del bloque.
  - Nonce: Nonce que resuelve el PoW, campo de 4 bytes, generalmente comienza con 0 que incrementa itera.
- Transaction counter: Número de transacciones del bloque.
- Transactions: Lista de transacciones incluidas en el bloque.
- Data: Datos almacenados en el bloque.

2.6.4.2. Mineros

Cualquier dispositivo o nodo participante de la red que este encargado de comprobar, validar, registrar y propagar los bloques tras la resolución de desafíos matemáticos del protocolo de consenso para determinar su validez, pudiendo recibir incentivos del sistema por esta labor en forma de activos o tokens Los mineros son ordenadores dedicados que aportan su poder computacional a la red para verificar las transacciones que se llevan a cabo. Son computadoras que se encargan de autorizar la adición de los bloques de transacción.

2.6.4.3. Nodos

Son computadoras conectadas a la red utilizando el mismo software/protocolo para comunicarse entre sí ya que almacena distribuye una copia actualizada en tiempo real del blockchain. Cada vez que un bloque se válida y se añade a la cadena, el cambio es comunicado a todos los nodos y este se añade a la copia que cada uno almacena. Algunos, conocidos como mining pools o grupos de minería, se encargan además de escuchar nuevas transacciones y agruparlas en bloques para proponerlos como trabajo a los mineros, que luego de ser confirmados son propagados a la red y añadidos a la cadena. De otro modo no podrán conectarse ni formar parte de la red de una blockchain.

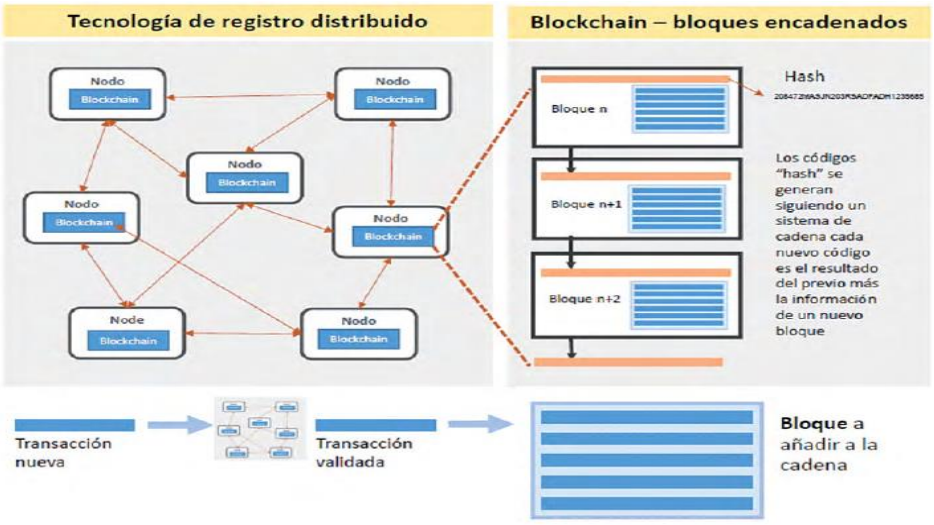


Figura N°2.7: Nodos de Blockchain

Fuente: Junestrand. (2018). Blockchain en el sector publico

2.6.4.4. Transacciones

Las transacciones son la parte central de Bitcoin, todas las demás partes están construidas para asegurar que las transacciones sean creadas, propagadas en la red P2P, verificadas y agregadas al Blockchain. Las transacciones son estructuras de datos las cuales almacenan transferencias de valor entre participantes del sistema. Cada transacción es almacenada como una entrada en el Blockchain.

Estructura de la transacción

Una transacción es en esencia una estructura de datos que almacena las transferencias de valor desde un origen, llamado entrada, a un destino, denominado salida. Las entradas y salidas de una transacción no están asociadas a cuentas o identidades. En su lugar, son un monto de satoshis los cuales sólo pueden ser gastados por el propietario de la dirección de destino, esto mediante el uso de la llave privada asociada a esta. (Becerra, 2015)

Tabla N° 2.1: Estructura de Transacción

Tamaño	Campo	Descripción
4 bytes	Versión	Reglas a las cuales se apega la transacción
1-9 bytes	Total de entradas	El número de entradas que se incluyen
Variable	Entradas	Una o más entradas de la transacción
1-9 bytes	Total de salidas	El número de salidas que se incluyen
Variable	Salidas	Una o más salidas de la transacción
4 bytes	Bloqueo	Una fecha en formato UNIX o un número de bloques

Nota. - Nakamoto, S. (2016). Bitcoin

2.6.5. Protocolos de Consenso Blockchain

El protocolo de consenso es el procedimiento mediante el cual se elige a un nodo para proponer un nuevo bloque. Se pretende que esta elección sea aleatoria, aunque no todos los participantes tienen la misma probabilidad de ganar el "sorteo". Se describen algunos de los procedimientos más utilizados como protocolos de consenso.

- Proof-of-Work (PoW). PoW Para implementar un servidor de sellado de tiempo distribuido de forma peer-to-peer, necesitaremos emplear un sistema de proof-of-work similar al Hashcash de Adam Back, más que al de los periódicos o los post Usenet. La proof-of-work consiste en escanear en busca de un valor que cuando fue hasheado, al igual que con SHA-256, el hash comience con un número de cero bits. El trabajo medio que

hace falta es exponencial en el número de cero bits requeridos y puede verificarse ejecutando un único hash.

- Proof-of-Stake (PoS). Tanto PoS como los protocolos que mencionaremos a continuación consisten en asignar mayor probabilidad de ganar el sorteo a aquellos que tienen más activos en la red. Aquí no hay personas compitiendo por validar el bloque, tampoco hay recompensa para quien lo consigue.
- Leased-Proof-of-Stake (LPoS). LPoS una variación de PoS en la que usuarios con poco capital pueden ceder sus probabilidades de ganar el sorteo. En caso de que el nodo en el que hayan delegado resulte el ganador y haya algún tipo de recompensa por el minado, se reparte proporcionalmente entre él y las personas que lo apoyaron.
- Proof of Importance (PoI). PoI funciona como PoW pero asignando la probabilidad de ser elegido en función de la actividad transacciones, balance o reputación en la red del nodo en lugar de su dinero. La idea es la misma, premiar con el derecho de proposición de bloques a las personas que más interesadas están en el buen funcionamiento de la cadena, de forma que no les convenga proponer bloques maliciosos que puedan perjudicarla. (Lopez, 2018).

Tabla N° 2.2: Protocolos y Algoritmos Blockchain

Propiedad	PoW	PoS	PBFT	DPOS	Ripple	Tendermint
Administración de la identidad del nodo	Abierta	Abierta	Autorizado	Abierta	Abierta	Autorizado
Ahorro de energía	No	Parcial	Sí	Parcial	Sí	Sí
Poder tolerado del adversario	<25% de poder de cómputo	<51% de estaca	<33.3% de réplicas defectuosas	<51% de validadores	<20% de réplicas defectuosas en UNL	<33% de poder de votación bizantina
Ejemplo	Bitcoin	Peercoin	Hyperledger Fabric	Bitshares	Ripple	Tendermint

Nota. - Marquez. (2018). Blockchain el auge de las transacciones

Blockchain cadena de bloques un sistema en el cual se pueden hacer transacciones seguras entre personas sin la necesidad se intermediarios un registro enlazados y cifrados una certificación de la información mediante consenso por los nodos

2.7. INTRANET

Es la red de redes, es decir, la red de computadoras más extendida del planeta, que conecta y comunica a millones de personas en todo el mundo. Estos cables se presentan en muchas formas: desde cables de red local (varias máquinas conectadas en una oficina o campus) a cables telefónicos convencionales, digitales y de fibra óptica que forman las "carreteras" principales. Es un esquema de red donde se utiliza, sobre una plataforma de red organizacional o inter-organizacional existente, con el medio de transporte entre sus unidades, el protocolo de internet, el TCP/IP, aquellas tecnologías avanzadas de publicación electrónica WEB(http), correo electrónico(email), archivos y recursos compartidos (file server, printer server, backup server, ftp server), acceso remoto (vpn, ssh). (Martin, 2014)

2.7.1. Tabla de comparación de Internet Intranet y Extranet

Tabla N° 2.3: Comparación de Internet Intranet Extranet

Nombre	Concepto	Características	Ventajas	Desventajas
Internet	Es una red de redes que permite la interconexión descentralizada de computadoras a través de protocolos TCP/IP.	Universal está extendido en todo el mundo. Fácil de usar no es necesario saber informática.	Comunicación fácil. Conexión desde cualquier parte del mundo. Interactúa con personas de todo el mundo.	Poca privacidad. Piratería. Dependencia o vicio. Virus y spam. Información no veraz.
Intranet	Red de ordenadores privada basada en los estándares de internet.	Confidencialidad garantiza que los datos no sean comunicados incorrectamente.	Ahorro de tiempo y costes para su puesta en marcha, a medio plazo supone un ahorro de tiempo y coste, mejor rapidez y eficacia.	Muchas intranets bien constituidas tecnológicament e resultan un fracaso.
Extranet	Es una red privada virtual resultante de la interconexión de dos o más intranets que utiliza internet.	Proceso y flujo de trabajo más agiles intercambia grandes volúmenes de datos utilizados.	Consultas online de pedidos, de niveles de stock, de productos de condiciones de compra/venta.	Pueden ser caros de aplicar y mantener dentro de una organización el sistema necesita ser controlado.

Nota. - Martin. (2014). Intranet

2.8. WEB 2.0

Web social comprende aquellos sitios web que facilitan compartir información, la interoperabilidad, el diseño centrado en el usuario y la colaboración en la World Wide Web Web 2.0 permite a los usuarios interactuar y colaborar entre si, como creadores de contenido. La red social conocida como web 2.0 pasa de ser un simple contenedor o fuente de información; la web en este caso se convierte en una plataforma de trabajo colaborativo.

2.8.1. Componentes

Componentes principales de la Web 2.0 son estas 4:

Comunicación

- **Conversaciones.** Actitud de las empresas y organizaciones a relacionarse directamente y de forma trasparente con los consumidores, gracias a las nuevas formas de comunicación.
- **Transparencia.** Abrirse al mundo, a otros puntos de vista, compartir toda la información posible y minimizar los “secretos”, ayudar a los demás y a uno mismo, sean personas o empresas.
- **Recomendaciones.** Es un mundo sobresaturado de información, sistemas de filtrado colaborativo y participativo permiten generar gran cantidad de recomendaciones fiables.
- **Compartir.** Frente a los entornos cerrados y la informática individual, compartir información en cualquier formato redunda en beneficio para todos.

Interacción

- **Interfaces enriquecidas.** Formas avanzadas de que un usuario interactúe con una aplicación o página web determinada, con funciones o nuevas posibilidades útiles, manteniendo la simplicidad aparente.

- **Folksonomías.** Metodología de clasificación en la que los propios usuarios emplean ‘tags’ o etiquetas de modo descentralizado para objetos diversos tales como fotografías, paginas, videos, audio textos.
- **Movilidad.** Manera e definir la posibilidad de acceder a un servicio, aunque el usuario cambie de lugar de acceso o de dispositivo.
- **La red como plataforma.** Muchos servicios dejan de ser aplicaciones encerradas en el ordenador personal para estar disponibles y ser usados, ‘vía web’, dese cualquier lugar.
- **Páginas de inicio personalizadas.** Puntos de inicio para el navegador, personalizables con módulos y contenidos diversos.

Contenido

- **Datos e Información.** El contenido es el rey porque existen nuevas posibilidades de compartirlo, llevarlo de un lado a otro hacer remezclas, etiquetarlo y encontrarlo.
- **Contenido generado por el usuario.** La información generada, publicada y compartida por los individuos hacen que surjan nuevos servicios basados principalmente en este tipo de contenidos.
- **Economía de la atención.** Ante la sobredosis informativa de la actualidad, lo más valioso que tienen las personas suele ser su tiempo. Por lo tanto, su moneda de cambio es la atención.
- **Periodismo ciudadano.** Además de consumir información, el usuario escribe weblogs, toma fotos, graba videos, lo comparte, filtra y comenta.
- **Tags.** Etiquetas o palabras clave que describen o se asocian a diversos tipos de objetos de información y que sirven para clasificarlos.

Sociedad

- **Redes sociales.** Redes en cuya estructura los nodos individuales son personas que mantienen relaciones, como amistad intereses comunes.

- **Reputación/confianza.** Cuando el usuario es el protagonismo, su reputación influye en lo que le rodea.
- **Computación social.** Utilización el "colectivo" para realizar tareas de computación costosas o complejas.
- **Software social.** Herramientas que basan su experiencia en las necesidades o fines de comunicación de las personas.
- **Participación.** La participación de los individuos y forma activa es la razón de la existencia de muchos servicios. (Herrera, 2014)

2.8.2. Cuadro Comparativo diferencias de la Web 1.0, 2.0, 3.0

Tabla N° 2.4: Comparación de la Web 1.0, 2.0, 3.0

Web 1.0	Web 2.0	Web 3.0
Software de escritorio estático	Software de escritorio transformado en una aplicación Web	Aplicación Web con mucho AJAX (Asincronical Java and XML)
Maneja un lenguaje HTML para su construcción	Web que respeta los estándares de XHTML	Pondrán trabajar todas juntas
	Separación de contenido del diseño con uso de hojas de estilo	Serán relativamente pequeñas
	Permite la sindicación de contenidos	Gestionarán datos que estarán "en la nube"

Nota. - Herrera. (2014). Web 2.0 componetes

2.9. INGENIERÍA DE SOFTWARE

La ingeniería de software es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software; es decir, la aplicación de la ingeniería al software el estudio de enfoques. El fundamento para

la ingeniería de software es la capa proceso. El proceso de ingeniería de software es el aglutinante que une las capas de la tecnología y permite el desarrollo racional y oportuno del software de cómputo. El proceso define una estructura que debe establecerse para la obtención eficaz de tecnología de ingeniería de software. El proceso de software forma la base para el control de la administración de proyectos de software, y establece el contexto en el que se aplican métodos técnicos, se establecen puntos de referencia, se asegura la calidad y se administra el cambio de manera apropiada.

Figura N° 2.8: Ingeniería de Software

Fuente: Pressman, R. S. (2010). Ingenieria del software

2.9.1. Modelo de desarrollo de Software

El proceso de ingeniería de software y cada una de éstas se encuentra definida por un conjunto de tareas que identifica las tareas del trabajo que deben realizarse, los productos del trabajo que se producirán, los puntos de aseguramiento de la calidad que se requieren y los puntos de referencia que se utilizarán para evaluar el avance. Proceso de Software tareas que tienen que ser realizadas para producir un producto de alta calidad desarrollo de software. Se relacionan según determinados (modelos, métodos, principios, herramientas). (Pressman, 2010).

Actividades en el proceso de desarrollo de software

- Análisis de Requerimientos
- Especificación
- Diseño
- Programación
- Integración y Gestión de Configuraciones
- Validación y Verificación
- Prototipaje

2.9.1.1. Metodologías Tradicionales

Las metodologías tradicionales de desarrollo de software son orientadas por planeación. Inician el desarrollo de un proyecto con un riguroso proceso de elicitación de requerimientos, previo a etapas de análisis y diseño. Con esto tratan de asegurar resultados con alta calidad circunscritos a un calendario.

En las metodologías tradicionales se concibe un solo proyecto, de grandes dimensiones y estructura definida; se sigue un proceso secuencial en una sola dirección y sin marcha atrás; el proceso es rígido y no cambia; los requerimientos son acordados de una vez y para todo el proyecto, demandando grandes plazos de planeación previa y poca comunicación con el cliente una vez ha terminado ésta.

2.9.1.2. Metodologías Agiles

Las metodologías ágiles son flexibles, pueden ser modificadas para que se ajusten a la realidad de cada equipo y proyecto. Los proyectos ágiles se subdividen en proyectos más pequeños mediante una lista ordenada de características. Cada proyecto es tratado de manera independiente y desarrolla un subconjunto de características durante un periodo de tiempo corto, de entre dos y seis semanas. La comunicación con el cliente es constante al punto de requerir un representante de él durante el desarrollo. Los proyectos son altamente colaborativos y se adaptan mejor a los cambios; de hecho, el cambio en los requerimientos es una característica esperada y deseada, al igual que las

entregas constantes al cliente y la retroalimentación por parte de él. Tanto el producto como el proceso son mejorados frecuentemente. (Navarro, 2013).

2.9.1.3. Comparación entre Metodologías

La Tabla N° 2.5 se muestra aspectos relevantes de las metodologías de desarrollo tradicional contrastándolas con los aspectos relevantes de las metodologías de desarrollo ágil.

Tabla N° 2.5: Metodologías tradicionales vs agiles

Metodologías Tradicionales	Metodologías Ágiles
Predictivos	Adaptativos
Orientados a procesos	Orientados a personas
Proceso rígido	Proceso flexible
Se concibe como un proyecto	Un proyecto es subdividido en varios proyectos más pequeños
Poca comunicación con el cliente	Comunicación constante con el cliente
Entrega de software al finalizar el desarrollo	Entregas constantes de software
Documentación extensa	Poca documentación

Nota. - Navarro. (2013). Metodologias Agiles

2.10. METODOLOGÍAS

2.10.1. Método Cientifico

El método científico es una creación humana, creada artificialmente para crear conocimiento científico, puesto que el hombre no está dotado de manera natural para conocer científicamente. Es el camino para producir conocimiento objetivo, es un modo razonado de indagación establecido en forma deliberada y

sistemática, que está constituido por una serie de etapas o pasos para producir conocimiento.

2.10.1.1. Pasos de Método Científico

1. **Planteamiento del problema:** Consiste en la recopilación de hechos acerca de un problema o fenómeno natural que despierta nuestra curiosidad. Las observaciones deben ser lo más claras y numerosas posible, porque han de servir como base de partida para la solución. Delimitación clara y precisa el objeto de investigación.
2. **Composición del Marco Teórico:** Selección de teorías, conocimientos científicos, métodos y procedimientos para describir, explicar objetivamente el objeto de investigación en su estado histórico actual y futuro.
3. **Formulación de la Hipótesis:** Se plantea una descripción sobre los eventos y una explicación preliminar de las causas o razones de un evento o fenómeno. Es una afirmación razonada tentativa la cual debe contrastarse con los hechos y fenómenos reales.
4. **Constatación de la hipótesis:** Es la actividad mediante la observación, experimentación, documentación, encuesta y análisis sistemático, permite comprobar o demostrar adecuadamente si una hipótesis es falsa o verdadera.
5. **Conclusiones y resultados:** Resultados de la investigación, juicios sobre la falsedad o veracidad de las hipótesis utilizadas. Concordancia de los datos y análisis con relación a la hipótesis seleccionada. (Asuad, 2014).
    a. Concordancia total = Hipótesis correcta.
    b. Concordancia parcial = Hipótesis parcial.
    c. Concordancia falsa = Hipótesis falsa.

2.10.2. Métodologia Uwe

UWE es una metodología basada en el Proceso Unificado y UML para el desarrollo de aplicaciones Web, cubre todo el ciclo de vida de las aplicaciones Web.

2.10.2.1. Características de la metodología Uwe

La metodología UWE define vistas especiales representación gráficamente por diagramas en UML, tales como el modelo de navegación y el modelo de presentación. Los diagramas pueden adaptar como mecanismos de extensión basados en estereotipos que proporciona UML. Estos mecanismos de extensión son los que UWE utiliza para definir estereotipos que son los que finalmente se utilizaran en las vistas especiales para el modelado de aplicaciones Web.

2.10.2.2. Modelo que propone Uwe

- **Modelo Lógico-Conceptual.** UWE apunta a construir un modelo conceptual de una aplicación Web, se debe llevar a cabo de acuerdo con los casos de uso que se definen en la especificación de requerimientos.
- **Modelo de Navegación**. Consta de la construcción de dos modelos de navegación, el **modelo del espacio de navegación** especifica que objetos serán visitados por el navegador a través de la aplicación y el **modelo de la estructura de navegación** define como se relacionarán.
- **Modelo de presentación**. Describe dónde y cómo los objetos de navegación y accesos primitivos serán presentados al usuario.
- **Interacción Temporal**. Presenta los objetos que participan en la interacción y la secuencia de los mensajes enviados entre ellos.
- **Escenarios Web**. Permiten detallar la parte dinámica del modelo de navegación, especificando los eventos que disparan las situaciones, definen condiciones y explícitamente incluyen las acciones que son realizadas. Junto con el modelo de interacción temporal, los escenarios Web proveen la representación funcional dinámica del modelo de navegación.

- **Diagramas.** Los diagramas usados por UWE, son diagramas UML puro. Entre los más importantes tenemos: Diagramas de estado, de Secuencia, de colaboración y diagramas de Actividad.

2.10.2.3. Fases o etapas del Modelado UWE

- **Captura, análisis y especificación de requisitos:** Durante esta fase, se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir la aplicación web.
- **Diseño del sistema:** Se basa en la especificación de requisitos producido por el análisis de los requerimientos (fase de análisis), el diseño define cómo estos requisitos se cumplirán, la estructura que debe darse a la aplicación web.
- **Codificación del software:** Durante esta etapa se realizan las tareas que comúnmente se conocen como programación; que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación elegido, todo lo diseñado en la fase anterior.
- **Pruebas:** Las pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código.
- **La Instalación o Fase de Implementación:** Es el proceso por el cual los programas desarrollados son transferidos apropiadamente al computador destino, inicializados, y, eventualmente, configurados; todo ello con el propósito de ser ya utilizados por el usuario final. Esto incluye la implementación de la arquitectura, de la estructura del hiperespacio, del modelo de usuario, de la interfaz de usuario.
- **El Mantenimiento:** es el proceso de control, mejora y optimización del software ya desarrollado e instalado, que también incluye depuración de errores y defectos que puedan haberse filtrado de la fase de pruebas de control. (Castro, 2014).

2.11. ARQUITECTURA DEL SOFTWARE

La arquitectura del software alude a “la estructura general de éste y a las formas en las que ésta da integridad conceptual a un sistema”. En su forma más sencilla, la arquitectura es la estructura de organización de los componentes de un programa (módulos), la forma en la que éstos interactúan y la estructura de datos que utilizan. Sin embargo, en un sentido más amplio, los componentes se generalizan para que representen los elementos de un sistema grande y sus interacciones. Una meta del diseño del software es obtener una aproximación arquitectónica de un sistema. Ésta sirve como estructura a partir de la cual se realizan las actividades de diseño más detalladas. Un conjunto de patrones arquitectónicos permite que el ingeniero de software resuelva problemas de diseño comunes.

2.11.1. Patrón Modelo Vista Controlador

Según (Vicente, 2011), la modelo vista controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica del negocio de una aplicación de la interfaz del usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello, el Modelo Vista Controlador propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado, define componentes para la representación de la información, y por otro lado para la interacción de usuarios. Este patrón de diseño busca en las ideas la reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y posterior mantenimiento.

El patrón del Modelo Vista Controlador fue una de las primeras ideas den el campo de las interfaces graficas de usuario y uno de los primeros trabajos en describir e implementar aplicaciones software en términos de sus diferentes funciones. De manera genérica, los componentes del Modelo Vista Controlado son los siguientes:

- **El modelo.** Es la representación de la información con la cual el sistema opera, por lo tanto, gestiona todos los accesos a dicha información, tanto a consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica del negocio). Envía a la vista aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario. Las peticiones de accesos a manipulación de información llegan del modelo a través del controlador.
- **El controlador.** Responde a eventos (usualmente acciones de usuario) e invoca peticiones al modelo cuando se hace alguna solicitud sobre la información. También pude enviar comandos a su vista asociada si se solicitan un cambio en la forma en que se presenta el modelo, por tanto, se podría decir que el controlador hace el intermedio entre la vista y el modelo.
- **La vista.** Presenta el modelo (información y lógica del negocio) en un formato adecuado para interactuar (usualmente la interfaz del usuario) por tanto requiere de dicho modelo la información que debe representar como salida.

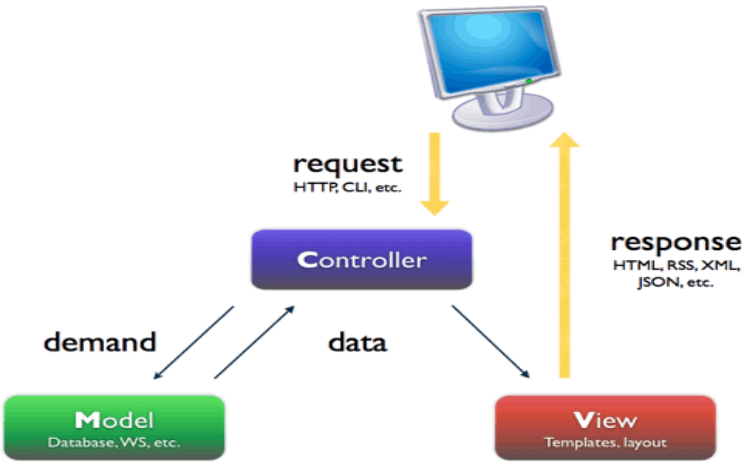


Figura 2.9: Modelo Vista Controlador

Fuente: Vicente, E. (2011). Modelo Vista Controlador

2.12. HERRAMIENTAS DE DESARROLLO

2.12.1. Gestor de Base de Datos Posgresql

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto- Relacionales. PostgreSQL almacena los datos de usuarios, así como también los datos de los grupos dentro de sus propios catálogos de sistema. De esta manera, cualquier conexión a PostgreSQL debe ser realizada con un usuario específico, y cualquier usuario puede pertenecer a uno o más grupos definidos. (Garavito, 2007)

2.12.2. Lenguaje de Programación

2.12.2.1. Python

Python es un lenguaje de código abierto, de alto nivel y multiparadigma, soporta orientación a objetos, programación imperativa y programación funcional. Al ser multiplataforma puede ser ejecutado en un windows, un mac, un linux, un teléfono móvil, una raspberry pi o controlar una placa de arduino. (Hashmania, 2019)

2.12.2.2. Php

PHP: **H**ypertext **P**reprocessor Lenguaje de script que se ejecuta en el servidor el resultado se devuelve al navegador como HTML capacidad para generar páginas de contenido dinámico, HTML, imágenes, videos, PDF, XML, etc. crear, abrir, leer, escribir, cerrar ficheros en el servidor, Utilizar bases de datos, procesar datos de formularios enviar y recibir cookies control de acceso de usuarios al sitio web encriptar datos. (Pavon J. , 2013)

2.12.3. Framework para el desarrollo Web

2.12.3.1. Framework - Bootstrap

Es un Framework de twitter para desarrollo de aplicaciones web, sencillo y ligero basta con un fichero CSS y uno JavaScript. Basado en los últimos estándares de desarrollo de Web HTML5, CSS3 y JavaScript/JQuery, Plugins de jQuery. (Pavon J. , 2014).

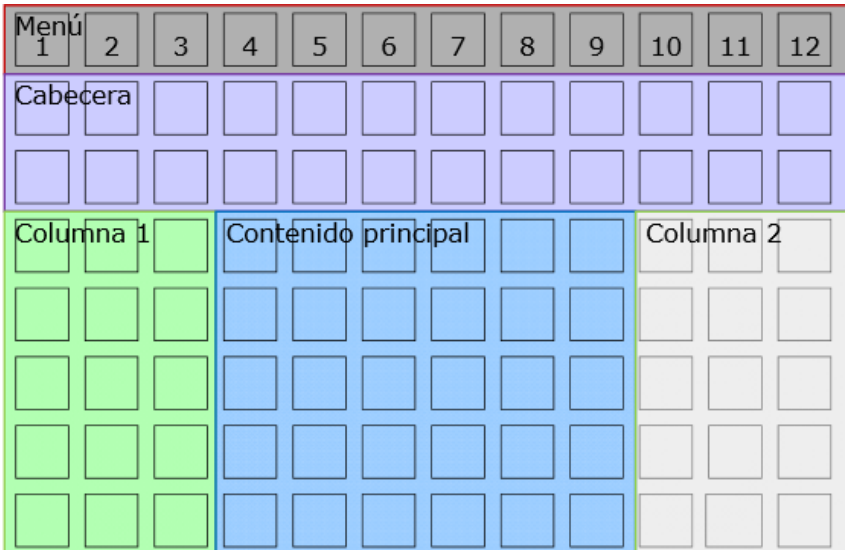


Figura N° 2.10: Esquema de Bootstrap

Fuente: Pavon, 2014. Aplicaciones Web/Sistemas

2.12.4. Framework Flask

Flask es un framework minimalista escrito en Python, concebido para facilitar el desarrollo de Aplicaciones Web bajo el patrón MVC, de forma rápida y con un mínimo número de líneas de código. Flask es simple y se limita a lo que toda aplicación Web necesita: Un sistema de enrutamiento y un sistema de platillas.

Estructura

Los únicos valores predeterminados que existen en Flask (aunque pueden ser cambiados con parámetros al crear el objeto de la aplicación) son los directorios static y templates, que deben estar en el mismo directorio donde existe la aplicación. (Saavedra, 2018)

2.12.5. Json

JSON es un formato de datos muy ligero basado en un subconjunto de la sintaxis de JavaScript: literales de matrices y objetos. Como usa la sintaxis JavaScript, las definiciones JSON pueden incluirse dentro de archivos JavaScript y acceder a ellas sin ningún análisis adicional como los necesarios con lenguajes basados en XM. (Quiroga, 2011).

2.13. MÉTRICAS DE CALIDAD DE SOFTWARE

La ingeniería de software es desarrollar y producir software de alta calidad. Para lograr este objetivo, es fundamental aplicar métodos y herramientas efectivos dentro del contexto de un proceso maduro de desarrollo de software. Para las evaluaciones que se quieran obtener es necesario la utilización de medidas técnicas, que evalúan la calidad de manera objetiva.

2.13.1. Estándar Iso/Iec 25000

ISO/IEC 25000 conocida como SQuaRE (Software Product Quality Requirements and Evaluation), que tiene por objetivo la creación de un marco de trabajo para definir los requisitos y evaluar la calidad del producto software, sustituyendo a las anteriores ISO/IEC 9126 e ISO/IEC 14598 y convirtiéndose así en el referente a seguir.

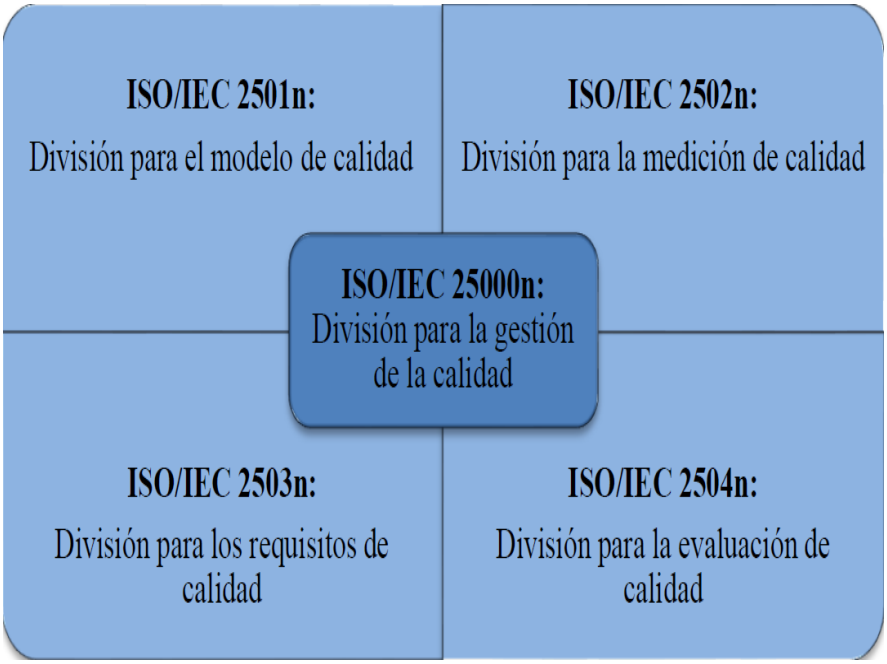


Figura N° 2.11: División de Iso/iec 25000

Fuente: García, 2015

La ISO/IEC 25000 se encuentra compuesta de varias partes o divisiones, entre las que podemos destacar:

1. **La ISO/IEC 2500n-División de gestión de calidad:** las normas que forman este apartado definen todos los modelos, términos y definiciones referenciados por todas las otras normas de la familia 25000. Actualmente esta división de encuentra formada por:

   ISO/IEC 25000, contiene el modelo de la arquitectura de SQuaRE, la terminología de la familia, un resumen de las partes, los usuarios previstos y las partes asociadas, así como los modelos de referencia.

   ISO/IEC 25001, establece los requisitos y orientaciones para gestionar la evaluación y especificación de los requisitos del producto software.

2. **La ISO/IEC 2501n-División de modelo de calidad**: las normas de este apartado presentan modelos de calidad detallados incluyendo las características para calidad interna, externa y en uso del producto software. Esta división está formada por:

   ISO/IEC 25010, determina las características de calidad del producto software que se pueden evaluar. En total son 8 las características de calidad que identifica: funcionalidad, rendimiento, compatibilidad, usabilidad, fiabilidad, seguridad, mantenibilidad y portabilidad.

   ISO/IEC 25012, define un modelo general para la calidad de los datos, aplicable a aquellos datos que se encuentran almacenados de manera estructurada y forman parte de un sistema de información.

3. **La ISO/IEC 2502n-División de mediciones de calidad**. las normas pertenecientes a esta división incluyen un modelo de referencia de calidad del producto software, definiciones matemáticas para las mediciones de calidad y una guía práctica para su aplicación. Presenta aplicaciones de métricas para la calidad externa, interna y en uso del producto software. Actualmente esta división está formada por:

   ISO/IEC 25020, presenta una explicación introductoria y un modelo de referencia común a los elementos de medición de la calidad. También proporciona una guía para que los usuarios seleccionen o desarrollen y apliquen medidas propuestas por normas ISO.

ISO/IEC 25021, define y especifica un conjunto recomendado de métricas base y derivadas que puedan ser usadas a lo largo de todo el ciclo de vida del desarrollo software.

ISO/IEC 25022, define específicamente las métricas para realizar la medición de la calidad en uso del producto.

ISO/IEC 25023, define específicamente las métricas para realizar la medición de la calidad de productos y sistemas software.

ISO/IEC 25024, define específicamente las métricas para realizar la medición de la calidad de datos.

4. **ISO/IEC 2503n-División de requisitos de calidad**. Las normas que forman este apartado ayudan a especificar requisitos de calidad. Está compuesto por:

   ISO/IEC 25030, las normas que forman esta división ayudan a especificar los requisitos de calidad. Estos requisitos pueden ser usados en el proceso de especificación de requisitos de calidad para un producto software que va a ser desarrollado o como entrada para un proceso de evaluación.

5. **La ISO/IEC 2504n-División de evaluación de calidad:** Incluye normas que proporcionan requisitos, recomendaciones y guías para llevar a cabo el proceso de evaluación del producto software. Esta división está formada por:

   ISO/IEC 25040, define el proceso de evaluación de la calidad del producto software.

   ISO/IEC 25041, describe los requisitos y recomendaciones para la implementación práctica de la evaluación del producto software desde el punto de vista de los desarrolladores, de los adquirentes y de los evaluadores independientes. (Molina, 2015).

2.13.2. Modelo de calidad (calidad interna y externa)

Categoriza propiedades de calidad de producto del sistema/software en ocho características: adecuación funcional, eficiencia de rendimiento, compatibilidad,

usabilidad, confiabilidad, seguridad, mantenibilidad y portabilidad. Cada característica está compuesta de un conjunto de subcaracterísticas relacionadas.

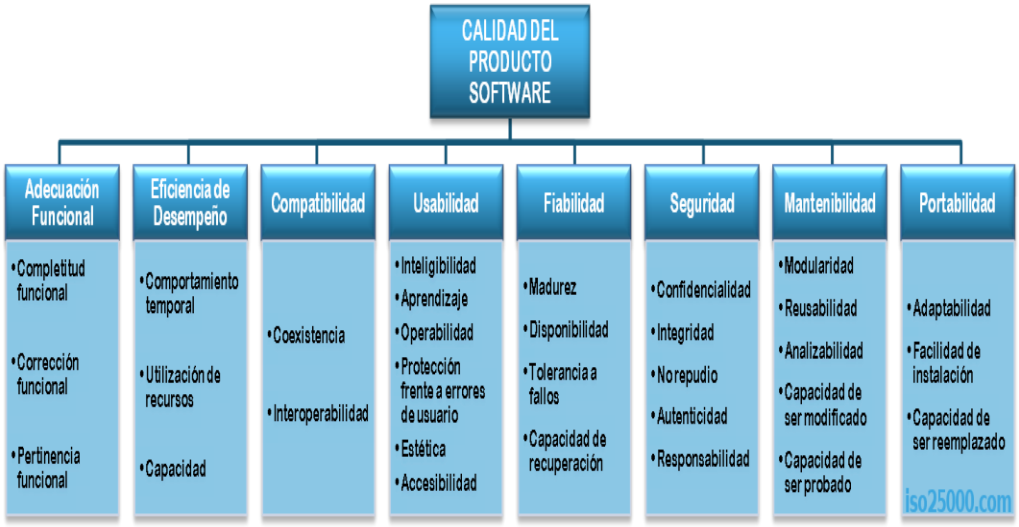


Figura N° 2.12: Iso/iec 25000

Fuente: Sierra, 2017

El modelo de calidad del producto puede ser aplicado solo a un producto de software, o a un sistema de computación que incluye software, debido a que muchas de las subcaracterísticas son relevantes tanto para el software como para los sistemas. A continuación, la descripción de cada característica y subcaracterísticas del modelo de calidad del producto:

1. **Adecuación Funcional:** representa la capacidad del producto o sistema software para proporcionar las funciones necesarias para satisfacer al usuario. Contempla las siguientes subcaracterísticas:

   Completitud funcional: capacidad del sistema software para proporcionar un conjunto de funcionalidades apropiadas para cubrir todas las tareas y objetivos determinados por el usuario.

   Exactitud funcional: capacidad del sistema software para proporcionar los resultados correctos con el grado necesario de precisión.

2. **Fiabilidad:** capacidad del producto o sistema software para realizar las funciones específicas cuando se utiliza bajo ciertas condiciones y periodos de tiempo determinadas. Contempla las siguientes subcaracterísticas:

   Madurez: capacidad del sistema software para satisfacer las necesidades de fiabilidad durante el funcionamiento normal.

   Disponibilidad: capacidad de un sistema software de estar operativo y accesible para su uso cuando se necesite.

   Tolerancia a Fallos: capacidad de un sistema software para operar cuando se presenten fallos.

   Recuperabilidad: capacidad de un sistema software para reestablecer el estado del sistema y recuperar datos que se hayan afectado, en caso de interrupción o fallo.

3. **Eficiencia en el Desempeño:** capacidad de un producto o sistema software de proporcionar un rendimiento apropiado, respecto a la cantidad recursos utilizados bajo determinadas condiciones. Contempla las siguientes subcaracterísticas:

   Comportamiento Temporal: capacidad de un sistema software para proporcionar los tiempos de respuesta y procesamiento apropiados.

   Utilización de Recursos: capacidad en que un sistema software utiliza las cantidades y tipos de recursos adecuados.

   Capacidad: capacidad de un sistema software de cumplir con los requisitos determinados.

4. **Facilidad de Uso:** capacidad del producto o sistema software para que sea entendido, aprendido, agradado y usado por el usuario. Contempla las siguientes subcaracterísticas:

Capacidad de reconocer su adecuación: capacidad del sistema software que permite al usuario entender si el software es adecuado para sus necesidades.

Capacidad para ser entendido: capacidad del sistema, que permite al usuario entender si el software es adecuado para alcanzar sus objetivos determinados.

Operatividad: capacidad de un sistema software que permite al usuario operarlo y controlarlo con facilidad.

Protección contra errores del usuario: capacidad en que el sistema brinda la protección necesaria contra errores que realizan los usuarios.

Estética de la Interfaz del usuario: capacidad en que la interfaz de usuario llega a satisfacer y agradar al usuario.

Accesibilidad técnica: capacidad del sistema software para que se permita ser utilizado por usuarios con determinadas discapacidades.

5. **Seguridad:** capacidad de proteger la información y los datos, de manera que personas o sistemas no autorizados puedan tener acceso para consultas o actualizaciones. Contempla las siguientes subcaracterísticas:

   Confidencialidad: capacidad de proteger la información y el acceso a datos no autorizados, ya sea de manera accidental o intencional.

   Integridad: capacidad de un producto, sistema o componente software para evitar accesos no autorizados a datos o programas de computación.

   No – repudio: capacidad para demostrar que los eventos han ocurrido, de manera que dichos eventos no puedan ser refutados posteriormente.

   Responsabilidad: capacidad de dar seguimiento a las acciones que fueron realizadas por una entidad.

   Autenticidad: capacidad de demostrar la identidad de un sujeto o un recurso.

6. **Compatibilidad:** capacidad de dos o más sistemas software, para llevar a cabo sus funciones intercambiando información mientras comparten el mismo entorno. Contempla las siguientes subcaracterísticas:

   Co-Existencia: capacidad de un sistema software para coexistir en un entorno en el cual comparten recursos comunes con otro software independiente.

   Interoperatividad: capacidad de dos o más sistemas software para intercambiar información y utilizar dicha información.

7. **Mantenibilidad:** capacidad del sistema software para ser modificado o actualizado debido a necesidades evolutivas y correctivas. Contempla las siguientes subcaracterísticas:

   Capacidad de ser Analizado: facilidad con la que se puede llevar a cabo un análisis del impacto de una determinada modificación en el sistema.

   Capacidad de ser Modificado: capacidad del sistema para permitir que sea modificado sin causar daños o reducir la calidad del producto existente.

   Capacidad de ser Probado: facilidad de realizar pruebas a un sistema o componente software, para determinar si se han cumplido con los requerimientos establecidos.

8. **Portabilidad:** capacidad de un sistema o componente software de ser trasladado de un entorno a otro sin que esto afecte la funcionalidad de cada sistema. Contempla las siguientes subcaracterísticas:

   Adaptabilidad: capacidad de un sistema software de ser adaptado a distintos entornos.

   Capacidad de ser Instalado: capacidad de un sistema para que pueda ser fácilmente instalado y/o desinstalado.

Capacidad de ser Reemplazado: capacidad del sistema software para ser utilizado en lugar de otro sistema en el mismo entorno y cumpliendo con el mismo objetivo. (Sierra, 2017).

2.14. ANÁLISIS DE COSTOS DE SOFTWARE COCOMO

2.14.1. Cocomo II

Entre los distintos métodos de estimación de costes de desarrollo de software, el modelo COCOMO (COnstructive COst MOdel) desarrollado por Barry M. Boehm, se engloba en el grupo de los modelos algorítmicos que tratan de establecer una relación matemática la cual permite estimar el esfuerzo y tiempo requerido para desarrollar un producto. COCOMO es un modelo de formulación matemática con un fuerte componente de base empírica, principalmente utilizado para estimación de costos en los proyectos de software el modelo está orientado a la magnitud del producto final, está basado en estimaciones matemáticas, mide el "tamaño" del proyecto y utiliza las líneas de código como unidad de medida. Dos de los aspectos fundamentales del modelo COCOMO son los submodelos y los modos de desarrollo. Incluye submodelos son tres: básico, intermedio y detallado. Por su parte, los modos de desarrollo son también tres: orgánico, semi-acoplado y empotrado. (Boehm, 1981)

Por tanto, COCOMO define tres modos de desarrollo o tipos de proyectos:

- **Orgánico:** proyectos relativamente sencillos, menores de 50 KDLC líneas de código, en los cuales se tiene experiencia de proyectos similares y se encuentran en entornos estables.
- **Semi-acoplado:** proyectos intermedios en complejidad y tamaño (menores de 300 KDLC), donde la experiencia en este tipo de proyectos es variable, y las restricciones intermedias.
- **Empotrado:** proyectos bastante complejos, en los que apenas se tiene experiencia y se engloban en un entorno de gran innovación técnica. Además, se trabaja con unos requisitos muy restrictivos y de gran volatilidad.

Y por otro lado existen diferentes modelos que define COCOMO:

- **Modelo básico:** Se basa exclusivamente en el tamaño expresado en LDC.
- **Modelo intermedio:** Además del tamaño del programa incluye un conjunto de medidas subjetivas llamadas conductores de costes.
- **Modelo avanzado:** Incluye todo lo del modelo intermedio además del impacto de cada conductor de coste en las distintas fases de desarrollo.

2.14.2. Estimación del desarrollo del Esfuerzo

E = Esfuerzo = a KLDC e * FAE (persona x mes)

Dónde:

- **E=** es el esfuerzo estimado. Representa los meses-persona necesarios para ejecutar el proyecto.
- **KLDC** es el tamaño del software a desarrollar en miles de líneas de código.
- **a** y **e** son coeficientes que varían según el Modo de Desarrollo (Orgánico, Semicopado, Empotrado).

2.14.3. Estimación del desarrollo del Tiempo

T = Tiempo de duración del desarrollo = c Esfuerzo d (meses)

2.14.4. Estimación del desarrollo del Número de personas

P= Personal = E/T (personas)

Estos modos de desarrollo permiten utilizar cuatro valores constantes. Estos valores constantes, codificados aquí como “a”, “b”, “c” y “d”, son propuestos por el modelo COCOMO para complementar las ecuaciones de cálculo usadas en el modelo.

Tabla N° 2.6: Esquema de modos de desarrollo de software

MODO	a	b	c	d
Orgánico	2.40	1.05	2.50	0.38
Semilibre	3.00	1.12	2.50	0.35
Rígido	3.60	1.20	2.50	0.32

Nota. - Boehm. (1981). Estimacion de costo de software

Las ecuaciones incluidas, son las utilizadas para los submodelos básico e intermedio. Estas ecuaciones se utilizan para calcular el esfuerzo nominal en personas/mes (E), tiempo estimado en meses (T) y personal requerido (P). No se incluyen las ecuaciones para el submodelo detallado, por razones de espacio dentro del desarrollo de la propuesta del enfoque pedagógico descrita en este trabajo.

Tabla N° 2.7: Valores constantes por modo de desarrollo

Modo de desarrollo	*a*	*b*	*c*	*d*	Mes-Hombre (nominal)	Tiempo de desarrollo (nominal)
Orgánico	3.2	1.05	2.5	0.38	$E_i = 3.2 * KLOCS^{1.05}$	$T_d = 2.5 * E_i^{0.38}$
Semi-acoplado	3.0	1.12	2.5	0.35	$E_i = 3.0 * KLOCS^{1.12}$	$T_d = 2.5 * E_i^{0.35}$
Acoplado	2.8	1.20	2.5	0.32	$E_i = 3.2 * KLOCS^{1.05}$	$T_d = 2.5 * E_i^{0.32}$

Nota. - Boehm. (1981). Estimacion de costo de software

Los multiplicadores de esfuerzo, utilizados en la ecuación de esfuerzo del submodelo intermedio, son quince agrupados en cuatro grandes categorías: atributos de producto, atributos de computador, atributos personales y atributos del proyecto.

Cada uno de estos multiplicadores de esfuerzo, tiene una valoración que se clasifica en una escala de 6 valores desde “muy bajo”, “bajo”, “nominal”, “alto”, “muy alto” y “extraordinariamente alto”. Estos multiplicadores de esfuerzo ajustan

el valor real del esfuerzo. Los factores seleccionados se agrupan en cuatro categorías:

- **Atributos del producto de software**

RELY Confiabilidad Requerida

DATA Tamaño de la Base de Datos

CPLX Complejidad del Producto

- **Atributos del hardware**

TIME Restricción del Tiempo de Ejecución

STOR Restricción del Almacenamiento Principal

VIRT Volatilidad de la Máquina Virtual

TURN Tiempo de Respuesta de la computadora expresado en horas

- **Atributos del personal involucrado en el proyecto**

ACAP Capacidad del Analista

AEXP Experiencia en Aplicaciones Similares

PCAP Capacidad del Programador

VEXP Experiencia en la máquina virtual

LEXP Experiencia en el Lenguaje de Programación

- **Atributos propios del proyecto**

MODP Prácticas Modernas de Programación

TOOL Uso de Herramientas de Software

SCED Cronograma de Desarrollo Requerido

Tabla N° 2.8: Ecuaciones por tipo de modelo Cocomo

Atributos	Valor					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
Atributos de software						
Fiabilidad	0,75	0,88	1,00	1,15	1,40	
Tamaño de Base de datos		0,94	1,00	1,08	1,16	
Complejidad	0,70	0,85	1,00	1,15	1,30	1,65
Atributos de hardware						
Restricciones de tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria virtual			1,00	1,06	1,21	1,56
Volatilidad de la máquina virtual		0,87	1,00	1,15	1,30	
Tiempo de respuesta		0,87	1,00	1,07	1,15	
Atributos de personal						
Capacidad de análisis	1,46	1,19	1,00	0,86	0,71	
Experiencia en la aplicación	1,29	1,13	1,00	0,91	0,82	
Calidad de los programadores	1,42	1,17	1,00	0,86	0,70	
Experiencia en la máquina virtual	1,21	1,10	1,00	0,90		
Experiencia en el lenguaje	1,14	1,07	1,00	0,95		
Atributos del proyecto						
Técnicas actualizadas de programación	1,24	1,10	1,00	0,91	0,82	
Utilización de herramientas de software	1,24	1,10	1,00	0,91	0,83	
Restricciones de tiempo de desarrollo	1,23	1,08	1,00	1,04	1,10	

Nota. - Boehm. (1981). Estimacion de costo de software

2.15. PRUEBAS DE LA CAJA BLANCA Y NEGRA

2.15.1. Caja Negra

Las pruebas de caja negra se centran en probar el programa usando su interfaz externa, sin preocuparnos de la implementación del mismo. Lo fundamental es comprobar que los resultados de la ejecución del programa son los esperados, en función de las entradas que recibe. Es decir, solo interesa si realiza las funciones que se esperan de él. Las técnicas más usuales que siguen el método de caja negra son:

- **Particiones de equivalencia**: Consiste en dividir el dominio de entrada de un programa en clases de equivalencia de los que se pueden derivar casos de prueba, donde la prueba de un valor representativo de la misma sea extrapolable al que se conseguiría probando cualquier valor de la clase.
- **Análisis de valores límite**: En este caso, a la hora de implementar un caso de prueba, se van a elegir como valores de entrada, aquellos que se encuentra en el límite de las clases de equivalencia. Se justifica en la constatación de que para una condición de entrada que admite un rango de valores es más fácil que existan errores en los límites que en el centro. Ejemplo: un campo numérico que requiera una cifra de 3 dígitos, tendrá como valores límites inferiores 99 y 100, y como valores límites superiores 999 y 1000.
- **Valores típicos de error**: desarrolla casos de prueba con ciertos valores susceptibles de causar problemas, esto es, valores típicos de error, y valores especificados como no posibles. La determinación de los valores típicos de error se realiza en función de la naturaleza y funcionalidad del programa a probar, por lo que depende en buena medida de la experiencia del diseñador de la prueba.

2.15.2. Caja Blanca

Las pruebas de caja blanca se centran en probar el comportamiento interno y la estructura del programa, examinando la lógica interna del mismo. Este tipo de pruebas, se basan en unos criterios de cobertura lógica, cuyo cumplimiento determina la mayor o menor seguridad en la detección de errores. Las principales técnicas de caja blanca son:

- **Prueba de interfaz:** este tipo de prueba debe ser la primera en realizar. Se basa en analizar el flujo de datos que pasa a través de la interfaz del módulo, tanto externa como interna, para asegurar que la información fluye de forma adecuada tanto hacia el interior como hacia el exterior del módulo que se está probando.
- **Pruebas de estructuras de los datos locales**: estas pruebas tienen como objetivo asegurar la integridad de los datos durante todos los pasos de la ejecución del módulo.
- **Prueba del camino básico**: permite obtener una medida de la complejidad lógica de un programa (complejidad celomática) y utilizar esa medida como guía para definir un conjunto básico de caminos de ejecución. Es decir, la prueba del camino básico se orienta a cubrir la ejecución de cada una de las sentencias, cada una de las decisiones y cada una de las condiciones en las decisiones, tanto en su vertiente verdadero como falsa.
- **Prueba de bucles**: comprueban la validez de las construcciones de los bucles. Determinar la validez de las construcciones de los bucles es fundamental para garantizar que es correcto el módulo que se está probando. (Pressman, 2010)

2.16. SEGURIDAD

2.16.1. Criptografía

Un procedimiento que, utilizando un algoritmo con clave (clave de cifrado), transforma un mensaje sin atender a su estructura lingüística o significado, de tal forma que sea incomprensible o, al menos, difícil de comprender, a toda persona que no tenga la clave secreta (clave de descifrado) del algoritmo empleado. En la blockchain, la criptografía tiene la responsabilidad de proveer un mecanismo infalible para la codificación segura de las reglas del protocolo que rigen el sistema. Es también fundamental para evitar la manipulación, hurto o introducción errónea de información en la cadena de bloques, así como la responsable de generar firmas e identidades digitales encriptadas.

Se definirá criptografía entonces como la ciencia de la escritura enigmática, en la cual se aplican técnicas, métodos, algoritmos y conocimientos matemáticos como álgebra y aritmética para procesar la información y lograr este modo de escritura.

La criptografía provee de confidencialidad para los datos, y es la base de varios protocolos, algunos de ellos también aseguran la integridad y autenticidad en los recursos. Un sistema criptográfico o criptosistema posee elementos que son utilizados para el cifrado y descifrado de la información, una clave que es parametrizada en dichos métodos, y el contexto en el cual se encuentran definidos factores como el mensaje a cifrar dato o información en texto plano y su representación ya cifrada criptograma.

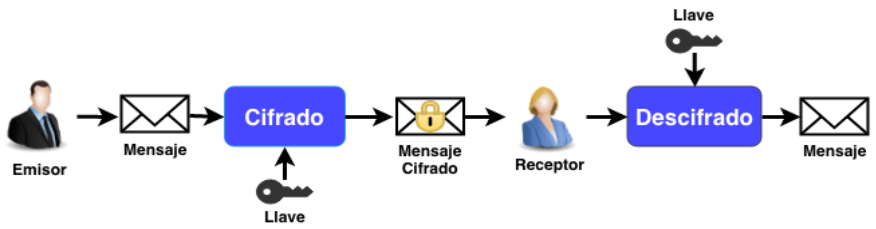


Figura N° 2.13: Esquema criptográfico

Fuente: Silva, R. (2019). Desarrollo de Blockchain

2.16.1.1. Criptografía Simétrica

Es ocultar cualquier dato dentro de un algoritmo cifrado con el cual el emisor puede enviar el mensaje oculto, la parte receptora tiene que tener el mismo algoritmo de cifrado y utilizarlo a la inversa. SI alguien en medio logra tener acceso a ese algoritmo criptográfico puede desencriptarlo sin problema. Este sistema criptográfico emplea la misma clave tanto para cifrar como para descifrar un mensaje. Este parámetro se supone secreto para que la información cifrada sea protegida ante el posible acceso de terceras partes, y se la define como un secreto compartido, ya que el correcto funcionamiento del criptosistema recae en la distribución confidencial de la clave a las entidades pertinentes. No obstante, esta necesidad de compartir la clave en “secreto” limita en muchas ocasiones el uso de este tipo de criptografía a la hora de poner en contacto entidades sin ningún tipo de relación previa.

Otra limitación que aborda este criptosistema es respecto a la necesidad de determinar con seguridad quien cifró o descifró un determinado mensaje ya que se utiliza la misma clave en estas dos funciones cifrado y descifrado, lo cual impide que pueda ser utilizada como mecanismo de no repudio o de autenticación de la identidad de la entidad cifrante.

Figura N° 2.14: Cifrado Simétrico

Fuente: Silva, R. (2019). Desarrollo de Blockchain

2.16.1.2. Criptografía Asimétrica

La criptografía asimétrica o criptografía de clave pública difiere principalmente de la simétrica en que las claves empleadas para las funciones de cifrado/descifrado no son únicas, sino que forman pares compuestos por una clave pública y una clave privada. Cada entidad pertinente posee un conjunto de claves de las cuales una de ellas prevalece protegida por su posesor y la otra se dispone de forma pública y visible por cualquier otra entidad tercera de allí su clasificación en pública y privada. La relación que prevalece entre el par de claves se entrelaza por medio de conceptos matemáticos/computacionales que aseguran y demuestran que resulta real y prácticamente imposible lograr descubrir una clave a partir del conocimiento de la otra, lo cual anula la necesidad de establecer secretos compartidos entre entidades ya que basta con tener acceso a una de las claves en este caso la pública.

Las características ofrecidas por este criptosistema, junto con la combinación de funciones de hashing ofrecen otras características primordiales que hacen a un sistema totalmente fiable: integridad y no repudio. No obstante, la agrupación de estos últimos mecanismos con sus respectivas características como producto resulta ni más ni menos que a una firma digital. (Silva, 2019).

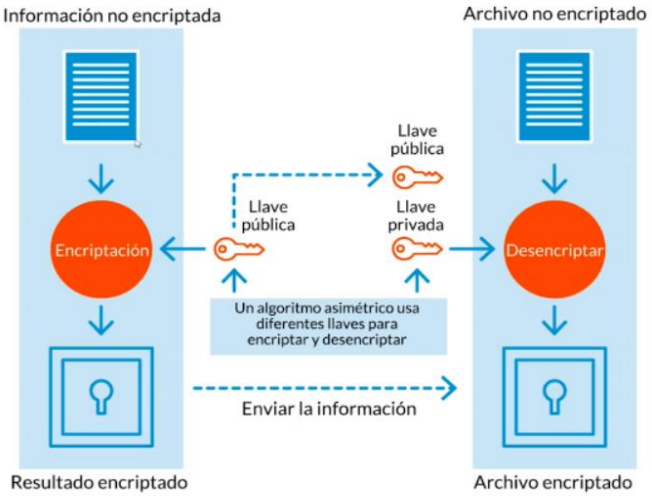


Figura N° 2.15: Cifrado Asimétrico

Fuente: Silva, R. (2019). Desarrollo de Blockchain

2.16.1.3. Hash Sha-256

La seguridad de un algoritmo hash es dependiente de su capacidad de producir un único valor para un conjunto de datos dados. Cuando una función hash produce el mismo valor para dos conjuntos de datos distintos, entonces se dice que se ha producido una colisión. Una colisión aumenta la posibilidad de que un atacante pueda elaborar computacionalmente conjuntos de datos que proporcionen acceso a información segura o para alterar ficheros de datos informáticos de tal forma que no cambiara el valor hash resultante y así eludir la detección. Una función hash 'fuerte' es aquella que es resistente a este tipo de ataques computacionales mientras que una función hash débil es aquella donde existe una creencia casi certera de que se pueden producir colisiones.

La función más empleada en el mundo de Blockchain suele ser SHA 256. El algoritmo usado por SHA-256 es uno de los más usados por su equilibrio entre seguridad y coste computacional de generación, pues es un algoritmo muy eficiente para la alta resistencia de colisión que tiene. En Blockchain, la función SHA-256 se emplea para garantizar la integridad de los registros incluidos en los bloques, en la creación de direcciones (envío y recepción de monedas) y como parte de la prueba de trabajo en el funcionamiento de la red. (Stefanescu, 2019).

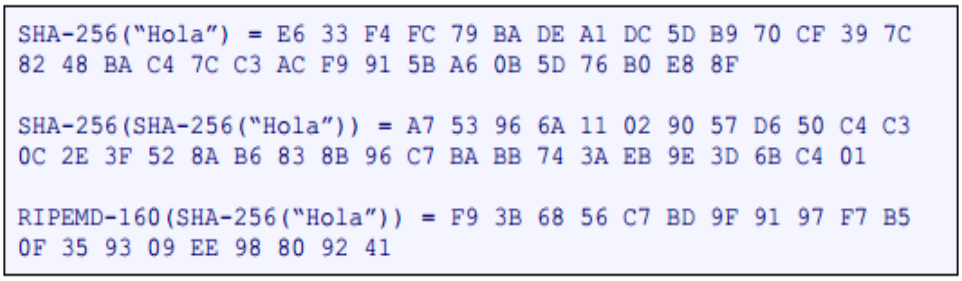


Figura N° 2.16: Cifrado Hash Sha-256

Fuente: Stefanescu, D. (2019).Blockchain estudio de alternativas

Los bloques que forman parte del blockchain son ordenados en la cadena por orden cronológico y tienen un código alfanumérico conocido como hash, que corresponde al bloque que los precede, gracias a ese hash todos están

referenciados por el bloque que los creo, por lo que solo los bloques que contienen un código valido son introducidos en la cadena y replicados a todos los nodos. Es gracias a este método lo que hace virtualmente imposible modificar un bloque que ha sido introducido ya hace un cierto tiempo. (Navarro B. , 2016).

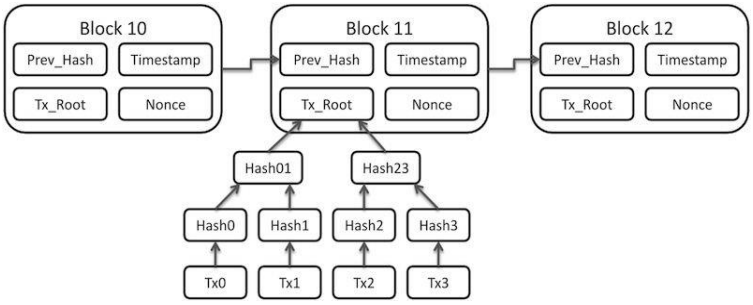


Figura N° 2.17: Seguridad de Bloques

Fuente: Navarro, B. (2016). Blockchain y sus aplicaciones

CAPITULO III

MARCO APLICATIVO

3.1. INTRODUCCIÓN

A lo largo de este capítulo se describe el fundamento teórico se mostrará de forma práctica, la construcción del modelo de certificación de contratos inteligentes aplicando la tecnología Blockchain para la emisión de certificación cites, donde se aplica el método de ingeniería, metodología Uwe, métrica de calidad de software iso /iec 25000 y el método de estimación de costo CocomoII.

El presente capitulo se describe la implementación de la metodología Uwe para las de fases de planificación: diseño del sistema, codificación del software, pruebas, la instalación o fase de implementación y el mantenimiento.

3.2. ESTRUCTURA DEL MODELO

Se muestra el funcionamiento del Modelo desde el momento que quiera realizar la emisión de certificación Cites hasta que se envía para su validación correspondiente en la Blockchain y así ejecutar el Smart Contract.

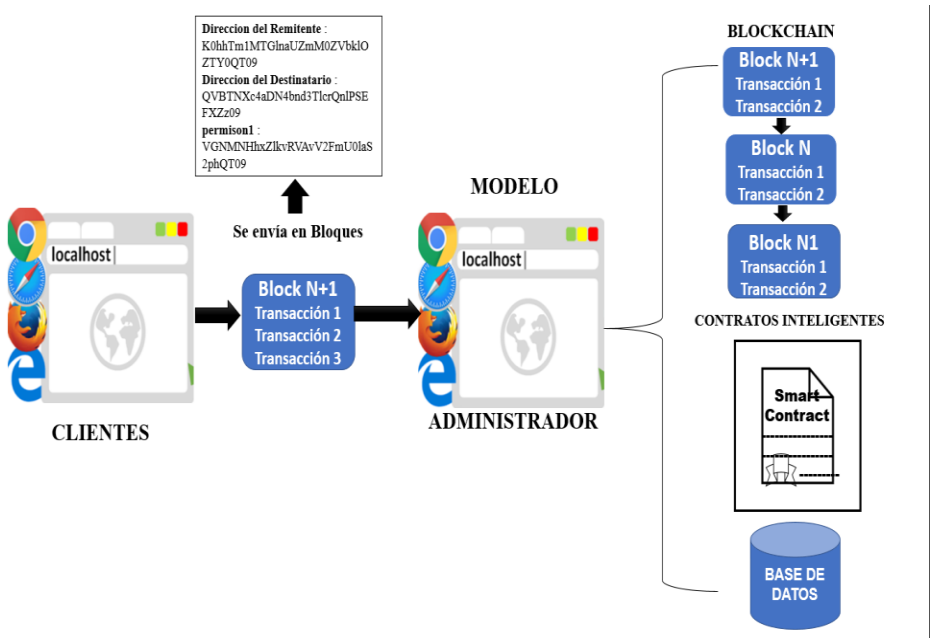


Figura N° 3.1: Estructura del Modelo

Fuente: Elaboración propia

3.2.1. Actores

Para emitir una certificación cites y se realice la solicitud que intervienen los siguientes actores:

- **Clientes**: Serán las personas que quieren emitir una certificación cite.
- **Administrador**: Serán los encargados de validar la certificación y decir si se emite el certificado cites o no.

3.2.2. Procesos de Emisión de Certificación Cites

La Autoridad Certificante en este caso Cites que permite que se puedan importar, exportar, reexportar especies amenazadas de fauna flora silvestres da el permiso de esta solicitud a los usuarios que desean emitir esta certificación. Ya que para enviar dicha solicitud debe ingresar a la página de cites y registrar todos estos requisitos:

1. Marcar en la casilla correspondiente el tipo de documento expedido (permiso de exportación, certificado de reexportación, permiso de importación u otro). Si se marca la casilla "otro", de indicarse el tipo de documento. Cada documento llevará un número único atribuido por la Autoridad Administrativa.

2. Para los permisos de exportación y los certificados de reexportación la fecha de expiración del documento no debe exceder los seis meses a partir de la fecha de expedición del mismo (un año para los permisos de importación).

3. Nombre y dirección **completa** del importador.

3a. Indicar el nombre completo del país.

4. Nombre y dirección **completa** del exportador/reexportador. Debe indicarse el nombre del país. El permiso o certificado serán válidos únicamente si llevan la firma del solicitante.

5. Las condiciones especiales pueden referirse a la legislación nacional o a las condiciones especiales fijadas por la Autoridad Administrativa expedidora para el

envío. Esta casilla puede utilizarse también para justificar la omisión de cierta información.

5a. Utilizar los siguientes códigos: **T**: comercial, **Z**: parque zoológico, **G**: jardín botánico, **Q**: circo y exhibición itinerante, **S**: científico, **H**: trofeo de caza, **P**: objeto personal, **M**: médico, **E**: educativo, **N**: reintroducción o introducción en el medio silvestre, **B**: cría en cautividad o reproducción artificial y **L**: aplicación de la ley / judicial / forense.

5b. Indicar el número de la estampilla de seguridad fijada en la casilla 13.

6. El nombre, la dirección y el país de la Autoridad Administrativa expedidora ya deben estar impresos en el formulario.

7-8. Indicar el nombre científico (género, especie y, según proceda, subespecie) del animal o de planta, tal como aparece en los Apéndices de la Convención o en las listas de referencia aprobadas por la Conferencia de las Partes, así como el nombre común de los mismos utilizado en el país que expide el permiso.

9. Describir, lo más exactamente posible, los especímenes objeto de comercio (animales vivos, pieles, flancos, carteras, zapatos, etc.). Si los especímenes están marcados (etiquetas, marcas de identificación, anillos, etc.), esté o no prescrito por una resolución de la Conferencia de las Partes (especímenes procedentes de establecimientos de cría en granjas, especímenes sujetos a cupos aprobados por la Conferencia de las Partes, especímenes de especies incluidas en el Apéndice I criados en cautividad con fines comerciales etc.), indicar el número y el tipo de marca y, si es posible, el sexo y la edad de los animales vivos.

10. Indicar el Apéndice (I, II o III) en el que está incluida la especie.

11.11 (Rev. CoP17), así como sus partes y derivados, exportadas con arreglo a las disposiciones del párrafo 5 del Artículo VII (especímenes de especies incluidas en el Apéndice I que hayan sido reproducidos artificialmente con fines no comerciales y especímenes de especies incluidas en los Apéndices II y III)

11. El número de especímenes y las unidades indicadas deberán ajustarse a la versión más reciente de las Directrices para la preparación y presentación de los informes anuales CITES.

11a. Indicar el número total de especímenes exportados en el año civil en curso (1 de enero a 31 de diciembre) (incluidos los cubiertos por este permiso) y el cupo anual correspondiente a la especie en cuestión. Esto se aplica tanto para los cupos establecidos por la Conferencia de las Partes como para los cupos nacionales.

12. El país de origen es el país en el que los especímenes fueron capturados o recolectados en la naturaleza, criados en cautividad o reproducidos artificialmente, salvo en el caso de los especímenes de plantas que dejen de cumplir los requisitos para gozar de una exención de las disposiciones de la Convención. En esos casos, se considera que el país de origen es el país en el que los especímenes dejan de cumplir los requisitos necesarios para la exención. Indicar el número del permiso o certificado del país exportador su fecha de expedición. En caso de que se desconozca toda o parte de la información, especifíquelo en la casilla 5. Esta casilla sólo debe ser rellenada en caso de reexportación.

12a. El país de la última reexportación es el país desde el que se reexportaron los especímenes antes de entrar en el país que expide el presente documento. Indicar el número del certificado de reexportación del país de la última reexportación y su fecha de expedición. En caso de que se desconozca toda o parte de la información, especifíquelo en la casilla 5. Esta casilla sólo debe ser rellenada en caso de reexportación de especímenes previamente reexportados.

12b. El "No. del establecimiento" es el número del establecimiento de cría en cautividad o reproducción artificial registrado. La "fecha de adquisición" se define en la Resolución Conf. 13.6 (Rev. CoP16) y se requiere únicamente para los especímenes reconvención.

13. Esta casilla debe ser rellenada por el funcionario que expide el permiso, indicando su nombre completo. La estampilla de seguridad debe ser fijada en esta casilla, validada con la firma del funcionario expedidor y un sello (preferentemente seco). El sello, la firma y el número de la estampilla de seguridad deben ser claramente legibles.

14. Esta casilla debe ser rellenada por el funcionario que inspecciona el envío en el momento de la exportación o reexportación. Indicar la cantidad de especímenes efectivamente exportados o reexportados. Anular las casillas no utilizadas.

15. Indicar el número del conocimiento de embarque o de la carta de porte aéreo, cuando el medio de transporte utilizado así lo requiera.

Una vez terminada el llenado registro de estos requisitos se envía la información usando la criptografía.

Direccion del Remitente : K0hhTm1MTGlnaUZmM0ZVbklOZTY0QT09
Direccion del Destinatario : QVBTNXc4aDN4bnd3TlcrQnlPSEFXZz09
numero_correlativo : MTE4UTNGR2RVV3NuMDNmdEkvNjFWQT09
fecha :
am1HUDB5U3E3aEx4MUUrL0NlZFFRcG1tdlU1UUFUMzNiRXZqM1BlemRlYz0=
observacion : N0RTSnZJejZtZUpxTnY2VEgxa1NEUT09
usuario : VFZBYlU2d0NKRCs5YitlZUN6d2ZUZz09
monto_deposito_bancario : eGwzMytHNVorY3RURXVYMmU4UjFvQT09
num_deposito_bancario : RlRONkszcGVRaUtGOHFadWZEUmZNZz09
tipo_moneda : UmJsN29vZElzVjZ3NU4vNnp3WFQ1QT09
digitalizacion : SnVzTmdwOUplUGFvMEJNVzJMa0hYdz09
estampilla : YXpSWFF1dDJERnNlOUdnZmZkUjBwUT09
num_cert : ZUpGSXN3UnRpNElKZlJnMGdUUjJ5QT09
anulado : NGlqakV3dFZBYkFXMm9yY3NtalJvUT09
num_cite : Yk04ZktWMW56Z01rYk9VTTdqL2h2UT09
estado : a0RURjlUQXdVR21KOFErL1BrZjFLZz09
apendice0 : c3g5Q1FBa2ZMVU40Rk91dzhtSmVjdz09
origen : YWkxUFRjbmtsbndWa2VWUW80b24wZz09
permison : dnlvVk1zRE9nNVRwNEhYeXdqYmptQT09

Figura N° 3.2: Registro de Cites

Fuente: Elaboración propia

3.3. PROCESOS DEL MODELO

3.3.1. Pasos de la Blockchain

3.3.1.1. Transacciones

La Blockchain recibe la petición de emisión de certificación cites a solicitud de los clientes y almacena primero en transacciones para ser añadida al bloque como se observa en la imagen.

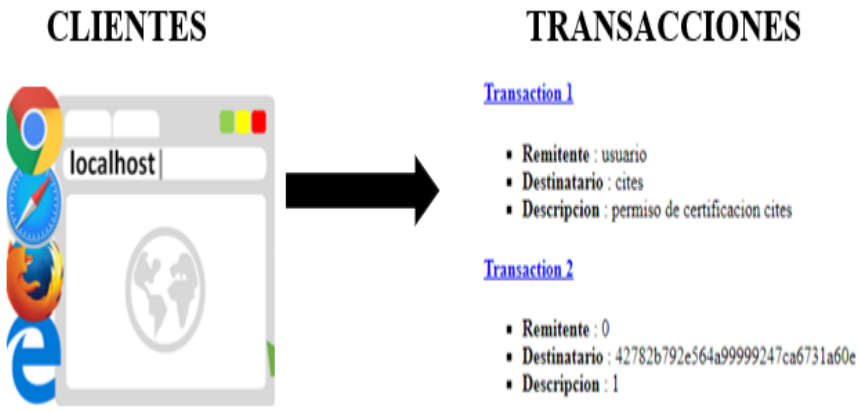


Figura N° 3.3: Almacenamiento en Transacciones

Fuente: Elaboración propia

Estructura de las transacciones en formato json ya que contiene la información enviada.

```
"transactions": [
    {
        "amount": "permiso",
        "recipient": "cites",
        "sender": "usuario"
    },
]
```

Tabla N° 3.1: Estructura de Transacción

Smart Contract	Descripción
Información	Contendrá las transacciones que se quieren aprobar enviadas por los clientes y transmitir con la Blockchain.
Remitente	Contendrá la url, dirección del cliente que quiere realizar la transacción.
Receptor	Contendrá la localización del remitente y su información para verificar y así poder validar su petición.

Nota. - Elaboración propia

3.3.1.2. Bloques

Se almacenan en bloques una vez recibidas las transacciones en formato Json ya que pueden contener varias transacciones.


```
{
    "index": 1,
    "previous_hash": "caa098d2b95e65a6e05713d52441689ad4849",
    "proof": 54600,
    "timestamp": 1590891844.4630187,
        "transactions": [ {
            "amount": "permiso",
            "recipient": "cites",
            "sender": "juan"    } ]
 }
```

Se la estructura de un bloque ya que tendrán los siguientes campos en la tabla N° 3.2.

Tabla N° 3.2: Estructura de Bloque

Bloques	Descripción
Index	Enumera el bloque, será una de las formas de identificar el bloque junto con el hash.
Previous_hash	Contiene el hash valido del bloque anterior para mantener la correlación de los bloques y evitar falsificaciones en las transacciones.
Proof	Es una limitación que se añade para dificultar la prueba de trabajo se define como un numero de ceros, que hay que añadir delante el hash del bloque para que dicho bloque sea válido, a mayor nonce mayor dificultad de encontrar el hash.
Timestamp	Marca de tiempo, para cuando se creó el bloque dia-mes-año.
Transactions	Contiene la información que fue enviada por el usuario que se transmitirá con el bloque.

Nota. - Elaboración propia

3.3.1.3. Hash Sha-256

Hacer el Bloque Inmutable: Para que no se pueda alterar el contenido la información es un algoritmo matemático con una serie de caracteres con una longitud fija.

Hash->SHA256(index, previous_hash, proof, timestamp, transactions)

Hash="caa098d2b95e65a0bf192784e05c0db6e05713d524418a9ef6f"

3.3.1.4. Blockchain

Ya implementada en bloques la información se encadenan los bloques para evitar alteración como y esto se hace gracias al hash previo del bloque que una vez generado no se puede modificar y si así lo hicieran el bloque.


```
{
    "chain": [
        {
          "index": 2,
          "previous_hash":"caa098d2b95e72784e05c024d48498a9ef6f",
          "proof": 54600,
          "timestamp": 1590891844.4630187,
          "transactions": [
               {
                   "amount": "permiso",
                   "recipient": "cites",
                   "sender": "usuario",
                   "importaor_emp": "bolivia"
              }
            ] }
  ],
    "length": 2
}
```


El modelo recibe las transacciones de la emisión de certificación cites sigue estos pasos:

Tabla N° 3.3: Estructura de Blockchain

Bloques	Descripción
Chain	La cadena de bloques.
Index	Enumera el bloque, será una de las formas de identificar el bloque junto con el hash.
Previous_hash	Contiene el hash valido del bloque anterior para mantener la correlación de los bloques y evitar falsificaciones en las transacciones.
Proof	Es una limitación que se añade para dificultar la prueba de trabajo se define como un numero de ceros, que hay que añadir delante el hash del bloque para que dicho bloque sea válido, a mayor nonce mayor dificultad de encontrar el hash.
Timestamp	Marca de tiempo, para cuando se creó el bloque dia-mes-año.
Transactions	Contiene la información que fue enviada por el usuario que se transmitirá con el bloque.
Length	Cuantos bloques hay en la cadena.

Nota. - Elaboración propia

3.3.1.5. Smart Contracts o Contratos Inteligentes

Los contratos inteligentes se ejecutan una vez enviadas las peticiones por los clientes al modelo y cuando acepta o almacena a la cadena de bloques se crea o ejecuta los contratos inteligentes almacenado la siguiente información la dirección del remitente, receptor, y sus transacciones como se observa en la tabla N° 3.4.

Tabla N° 3.4: Estructura de Smart Contracts

Smart Contracts
Contrato Numero :2 **Dirección Contrato :4d564ca5924d0562f732b8af0e23c5afdd3eeeb57ea9e7948035023244fecba9** • **Dirección del Remitente: K0hhTm1MTGlnaUZmM0ZVbklOZTY0QT09** • **Dirección del Destinatario: QVBTNXc4aDN4bnd3TlcrQnlPSEFXZz09** • **1. PERMISO/CERTIFICADO: dVlUUEkxQTRVMUUzaXd6MVpKaFZBdz09** • **2. Valido hasta el: N09hNzBDQjZodlQ2aE9DZU1vTjJuUT09** • **3.Importador (nombre, dirección): TzJMQ2VsRUM3V0g0SVdaR3d0b0dPdz09** • **3.a. País de importación: UDRHRU0xR2p1UG9ma2pLOFk2VWo2dz09** • **5.a. Propósito de la transacción: VWRINTNWSzRFTXJMNzB2MU5aRXVmdz09** • **A: ZGw3dlliQ1lha3FFQklUVW5OUU42QT09** • **Descripción: TzJMQ2VsRUM3V0g0SVdaR3d0b0dPdz09** • **Cantidad: TzJMQ2VsRUM3V0g0SVdaR3d0b0dPdz09** • **Unidad: c1pzUzVITE1BNDBIMk5OSnRwL00rdz09** • **U Otros: TzJMQ2VsRUM3V0g0SVdaR3d0b0dPdz09** • **Gestión cupo: c1pzUzVITE1BNDBIMk5OSnRwL00rdz09** • **12a/b. País de Origen: OTV6b1lNRFlZa001bzFpTmZaQTRwQT09** • **15. Conocimiento de embarque: dDNUdTFYWGZNOEtWV1MyaXZTbmhIMGpkZmtrdGZKcDJKZFM5NCthY**

Nota. - Elaboración propia

3.4. APLICACIÓN DE LA METODOLOGÍA UWE

Para el desarrollo del modelo se utilizó la metodología UWE sus fases se describen a continuación:

3.4.1. Captura, análisis y especificación de requisitos

En esta fase se especifican las características funcionales y no funcionales.

a) Requerimientos del Usuario

Se describen los requerimientos del usuario la tabla N° 3.5.

Tabla N° 3.5: Requerimientos del Usuario

REF.	FUNCIÓN	CATEGORÍA
R1	Realizar la solicitud de emisión de certificación cites.	**Evidente**
R2	Cumplir con todos los requisitos para emitir un certificado cite.	**Evidente**
R3	Registrar todos los datos que se pide para emitir la solicitud.	**Evidente**
R4	Se aprueba o rechaza la solicitud emitida por el usuario para la emisión de un certificado cite de importación, exportación, reexportación de alguna especie.	**Evidente**

Nota. - Elaboración propia

b) Requerimientos del Modelo

Se describen los requerimientos del Modelo, como se observa en la tabla N° 3.6.

Tabla N° 3.6: Requerimientos del Modelo

REF.	FUNCIÓN	CATEGORÍA
R1	El modelo debe ser el fácil manejo por el usuario	**Evidente**
R2	El modelo debe recibir información de manera segura y encriptada.	**Evidente**
R3	El modelo debe mostrar resultados confiables seguros.	**Evidente**

Nota. - Elaboración propia en base a ISO 25023

c) Requerimientos funcionales del Modelo de Certificación

Se describen los requerimientos funcionales, que realiza mostrando los eventos que se desarrolla dentro del Modelo como se observa en la tabla N° 3.7.

Tabla N° 3.7: Requerimientos Funcionales

REF.	FUNCIÓN	CATEGORÍA
R1	El modelo debe recibir peticiones del usuario para la correcta emisión de certificación cites.	**Evidente**
R2	El modelo recibe las transacciones en bloques.	**Evidente**
R2	El modelo recibe en bloques de manera cifrada.	**Evidente**
R4	El modelo almacena los bloques en la Blockchain y su base de datos.	**Evidente**
R5	El modelo ejecuta los Smart Contracts o contratos inteligentes.	**Evidente**

Nota. - Elaboración propia

3.5. DISEÑO DEL SOFTWARE

Se basa en el análisis de requerimientos, el diseño del modelo se define los requisitos deberán cumplir.

3.5.1. Casos de Uso

Se presentan los casos de uso para el funcionamiento del modelo de certificación cites de contratos inteligentes aplicando la tecnología Blockchain.

a) Caso de Uso Principal

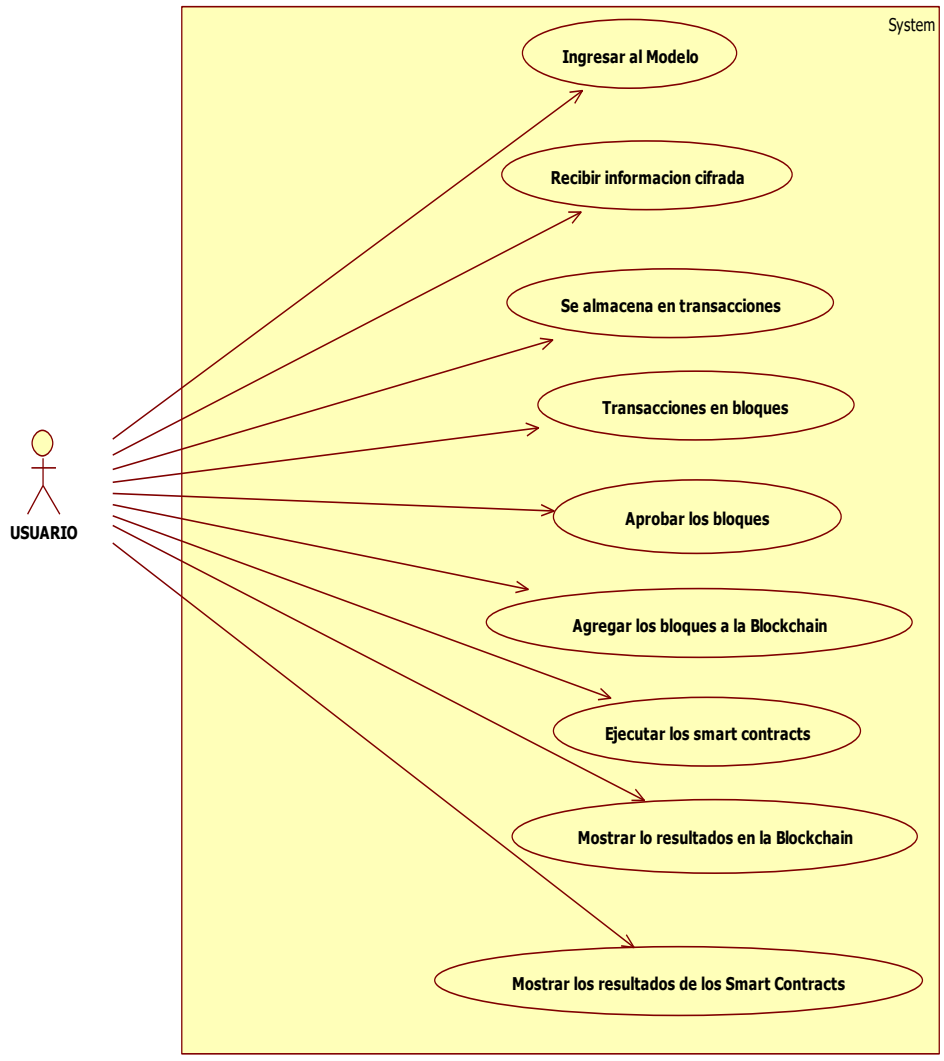


Figura N° 3.4: Caso de uso Principal

Fuente: Elaboración propia

b) Caso de Uso del Modelo

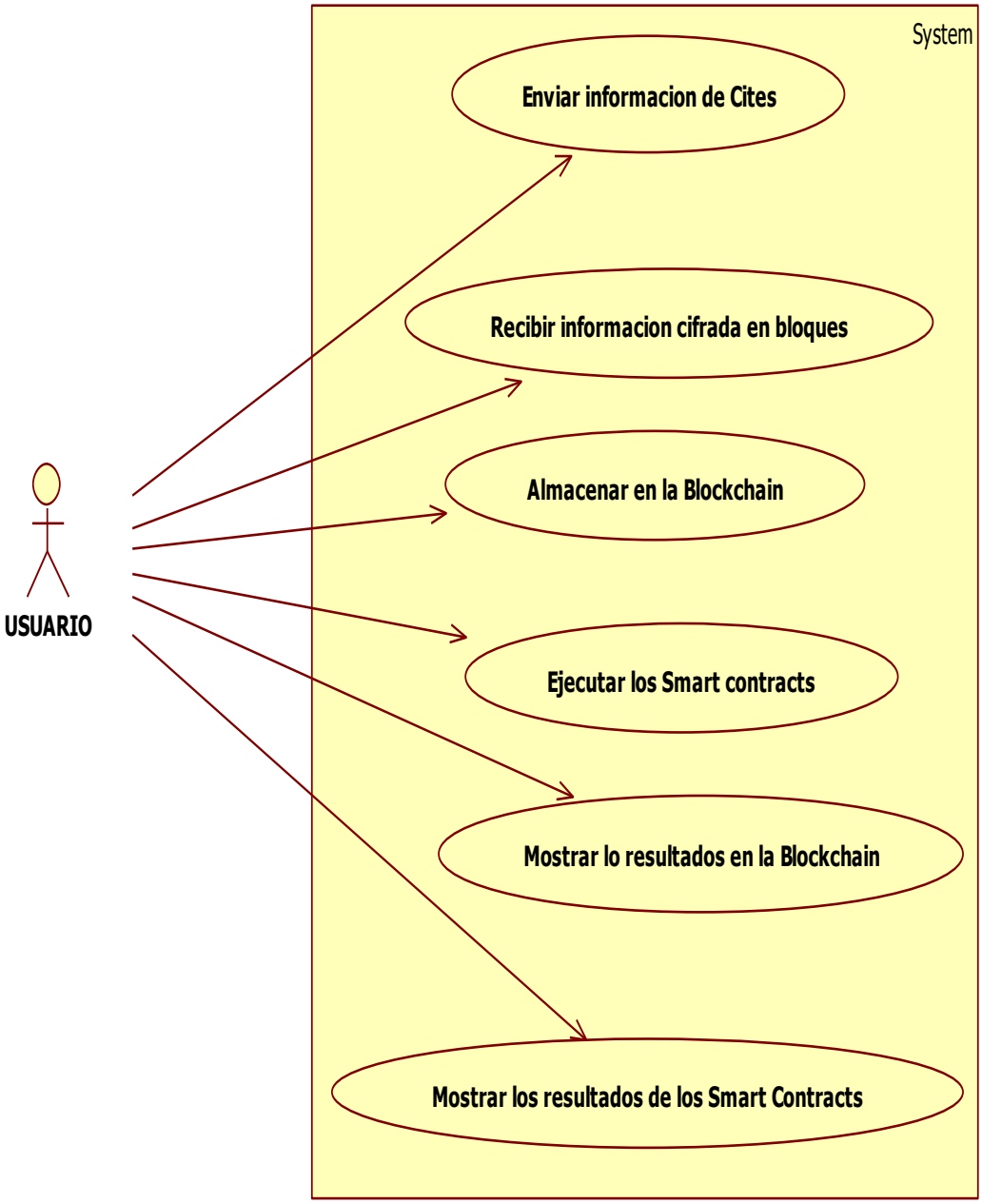


Figura N° 3.5: Caso de uso del Modelo

Fuente: Elaboración propia

3.5.2. Descripción de Casos de Uso

a) Caso de Uso Principal

Tabla N° 3.8: Descripción de Caso de Uso Principal

Casos de Uso	**Principal**
Actor	Usuario
Objetivo	Procesar la información de manera segura.
Descripción	Cuando el usuario ingrese al modelo podrá enviar información de manera segura.
FLUJO NORMAL	
1	El Modelo tiene la opción de recibir transacciones por usuarios.
2	El Modelo tiene la opción de validar las transacciones emitidas.
3	El Administrador tiene la opción revisar verificar las transacciones.
4	El Administrador tiene la opción de validar las transacciones si cumplen con los requerimientos para emitir una certificación.
5	Una vez recibida, verificadas las transacciones y validadas, el administrador podrá almacenar en la Blockchain y ejecutar los Smart Contracts.
FLUJO ALTERNATIVO	
4.1	Si el Administrador no valida la transacción, no se podrá ejecutar los Smart contracts.
RESULTADO ESPERADO	
El Administrador cuenta con el modelo para mejorar el proceso de la emisión de certificación Cites de contratos inteligentes con la Tecnología Blockchain.	

Nota. - Elaboración propia en base a ISO 25023

b) Caso de Uso de Obtención de Certificación Cites

Tabla N° 3.9: Obtención de requisitos

Casos de Uso	**Principal**
Actor	Usuario
Objetivo	Realizar pasos para el proceso almacenar en la Blockchain y Smart contracts.
Descripción	Cuando el Administrador ingrese al modelo se observa las transacciones emitidas por el usuario.
FLUJO NORMAL	
1	El Modelo tiene la opción de validar las transacciones.
2	El Usuario realiza la solicitud de misión de certificaciones cites.
3	El Modelo muestra las transacciones emitidas.
4	El Usuario accede a obtener el permiso del certificado Cite.
5	El Modelo muestra las transacciones en la cadena de bloques y los contratos inteligentes.
FLUJO ALTERNATIVO	
4.1	Si el usuario no accede a obtener el permiso, tiene que volver a enviar a solicitud ya que no debió cumplir con algún requisito.
RESULTADO ESPERADO	
El Usuario obtendrá el permiso de la certificación mostrada en los contratos.	

Nota. - Elaboración propia

3.5.3. Diagrama de Actividades

En el diagrama de actividades, se representa los pasos que se sigue para la ejecución del modelo como se puede observar el la Figura N° 3.6.

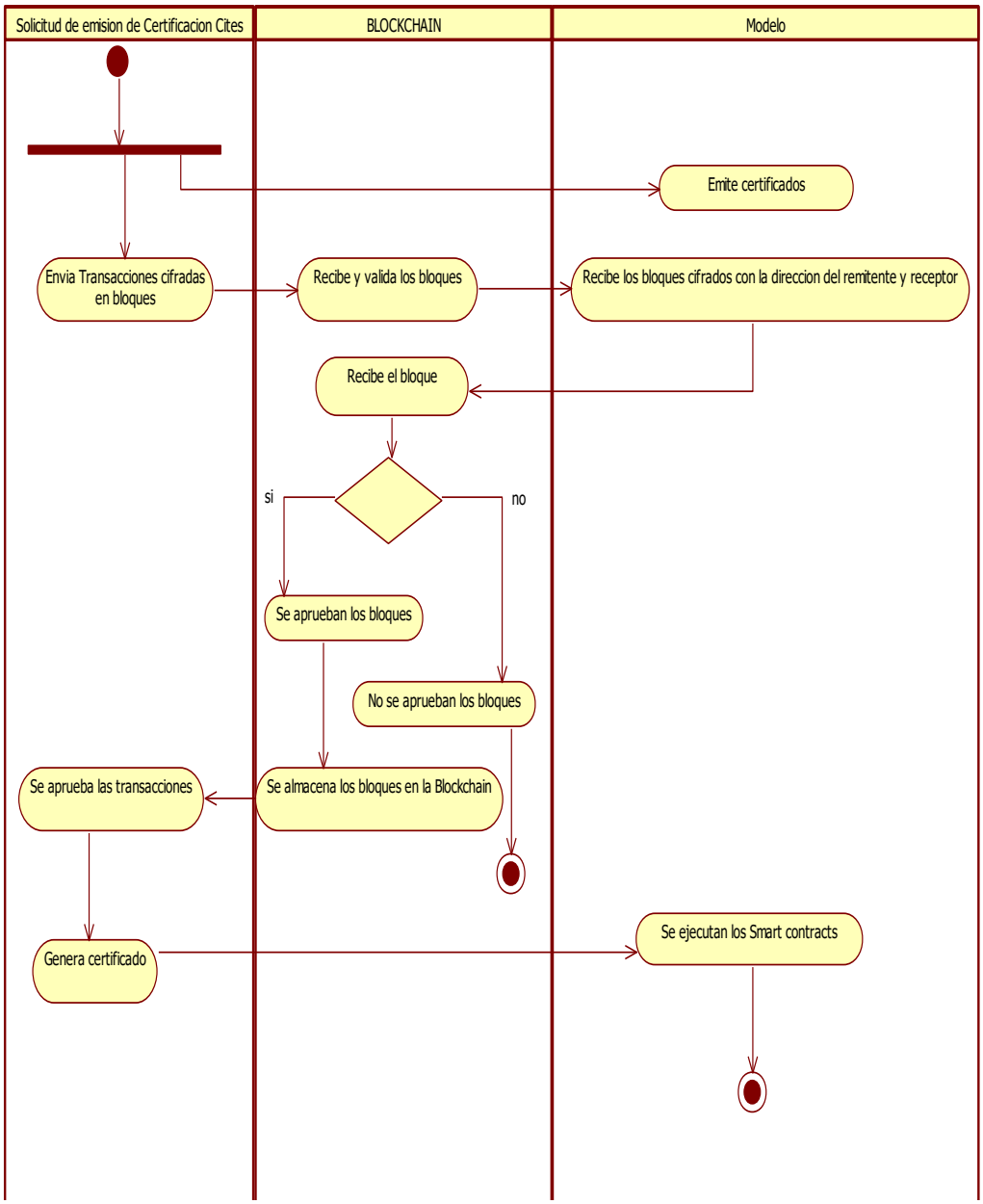


Figura N° 3.6: Diagrama de Actividades Principal

Fuente: Elaboración propia

3.5.4. Diagrama de Clases

El diagrama de clases, se representan las clases necesarias para el modelo haciendo el uso de la base de datos de cites, como se observa en la Figura N° 3.7.

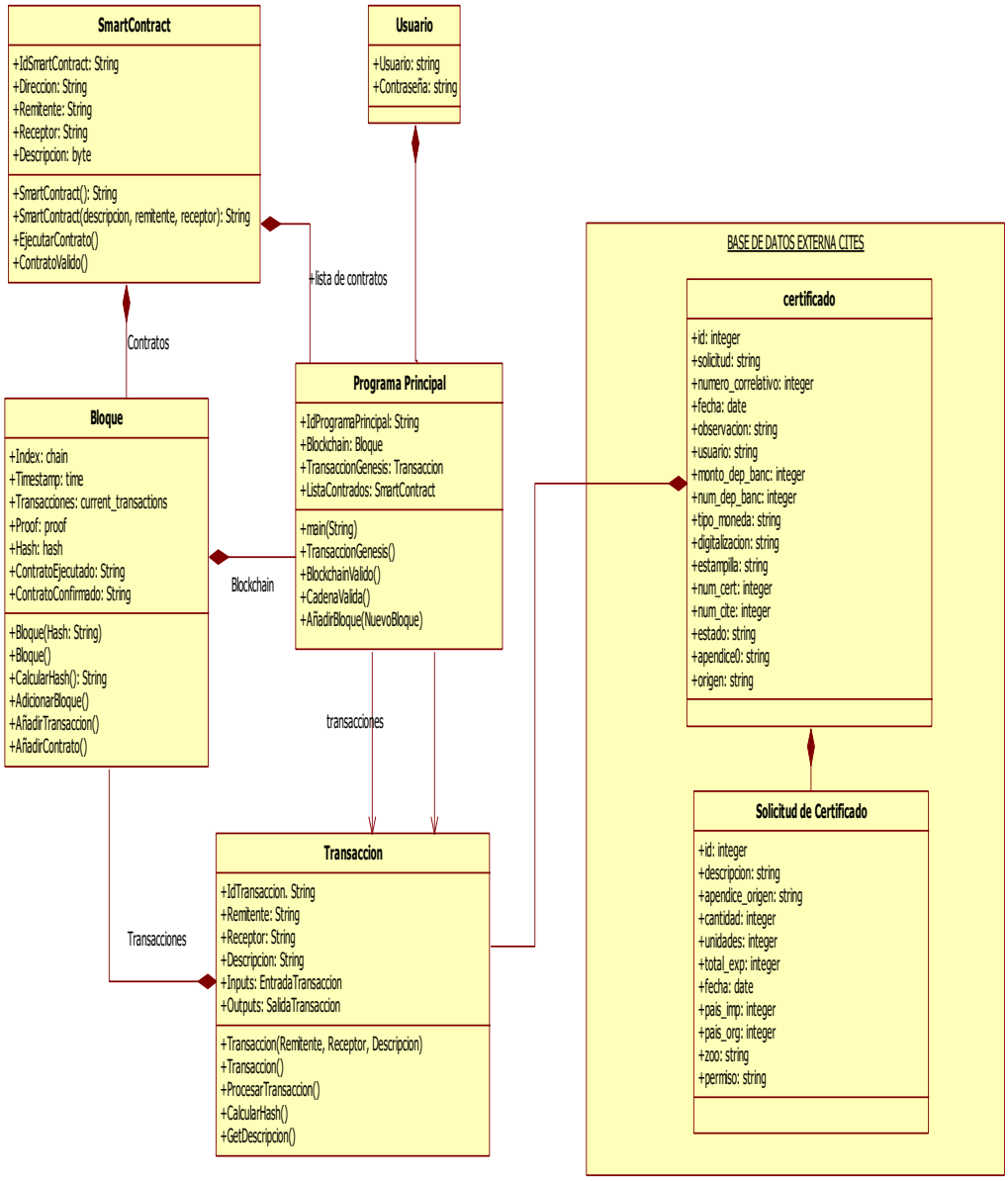


Figura N° 3.7: Diagrama de clases

Fuente: Elaboración propia

3.5.5. Diagrama de Secuencias

En el diagrama de secuencias se representa las secuencias del modelo como puede observarse en la Figura N° 3.8.

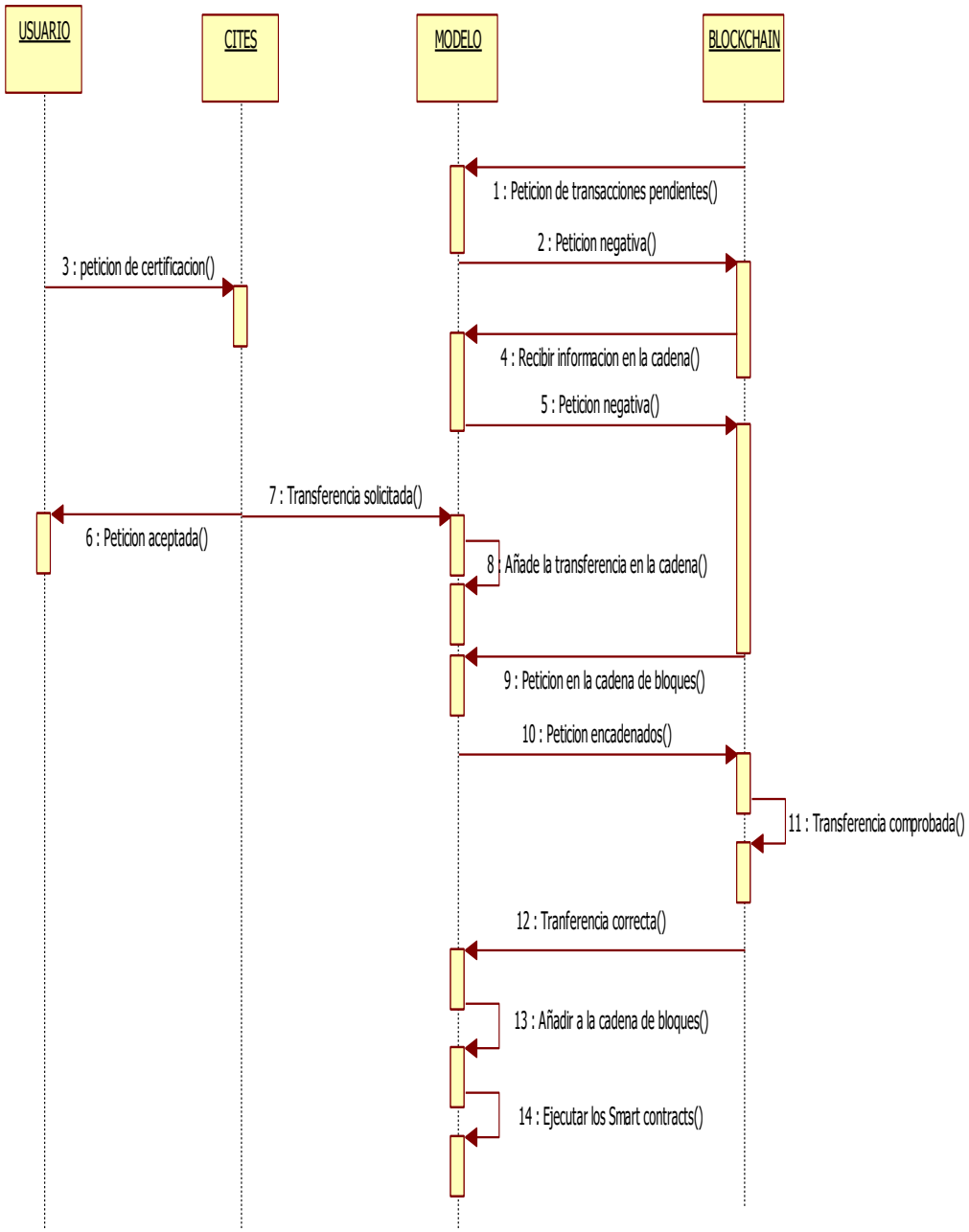


Figura N° 3.8: Diagrama de Secuencias

Fuente: Elaboración propia

3.5.6. Diagrama de Navegación

A continuación, se hace el modelado de navegación como interactuar con el modelo.

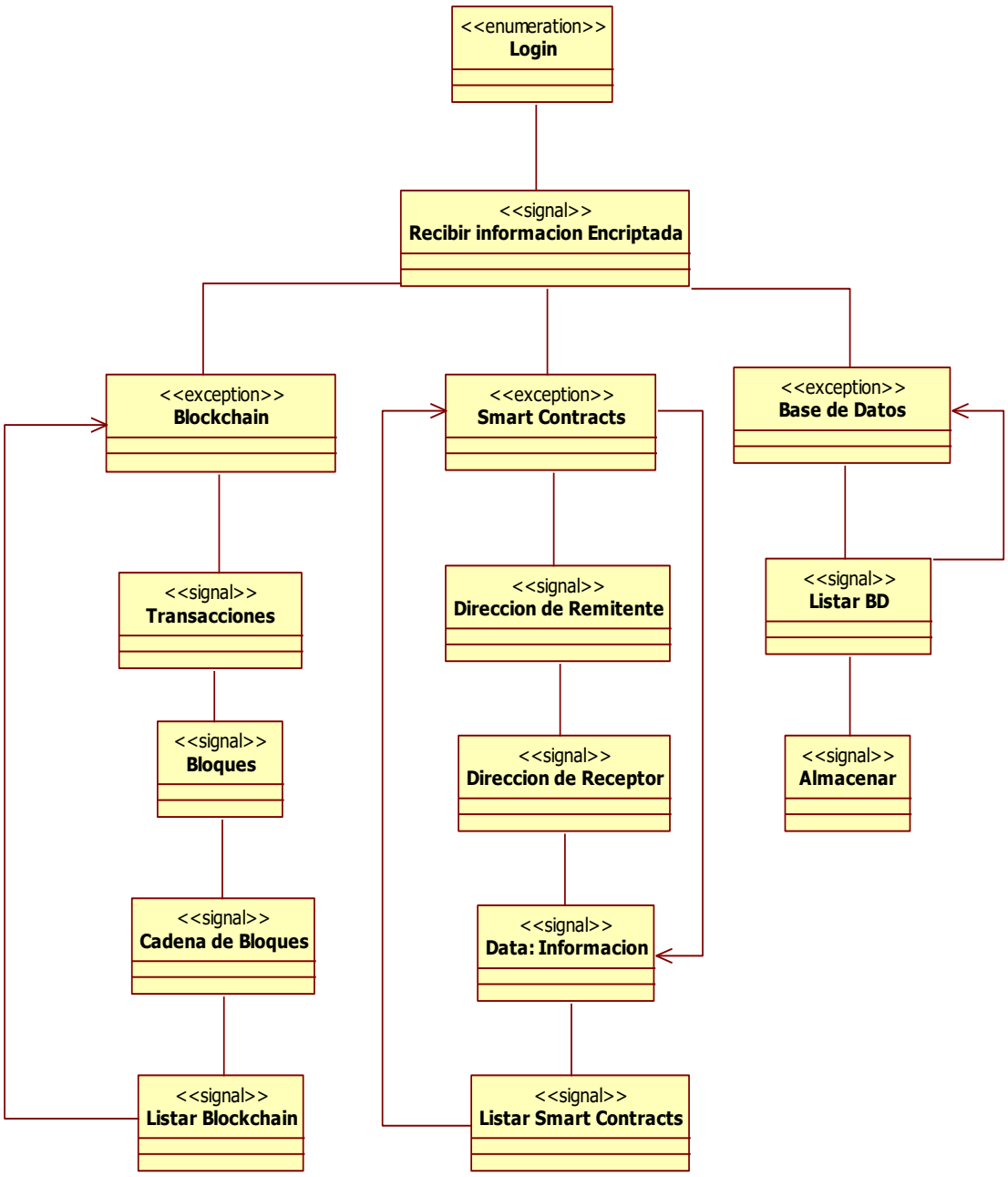


Figura N° 3.9: Diagrama de Navegación General

Fuente: Elaboración propia

3.5.7. Diagrama de Presentación

A continuación, se hace el modelado de Presentación de login, página principal, Blockchain, Smart contracts.

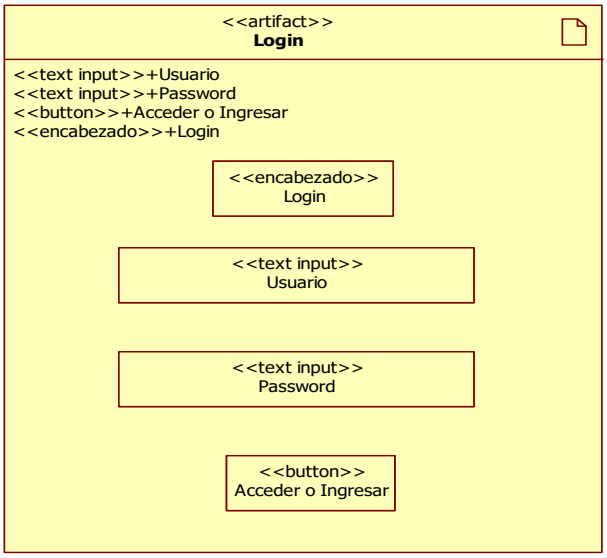


Figura N° 3.10: Diagrama de Presentación Login(inicio de sesión)

Fuente: Elaboración propia

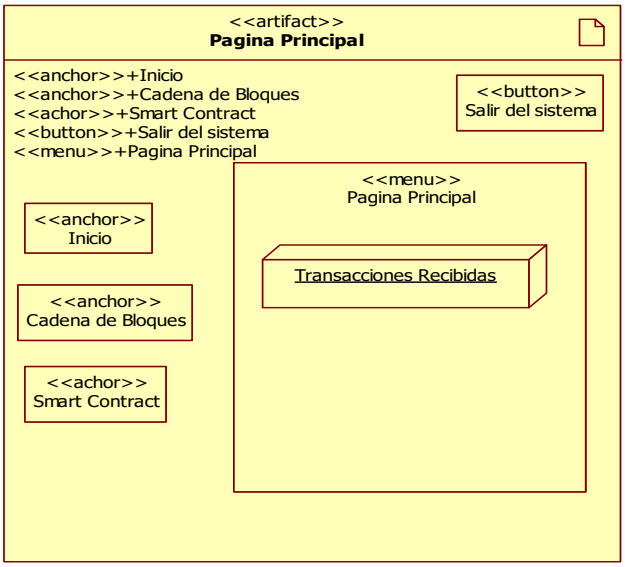


Figura N° 3.11: Diagrama de Presentación General

Fuente: Elaboración propia

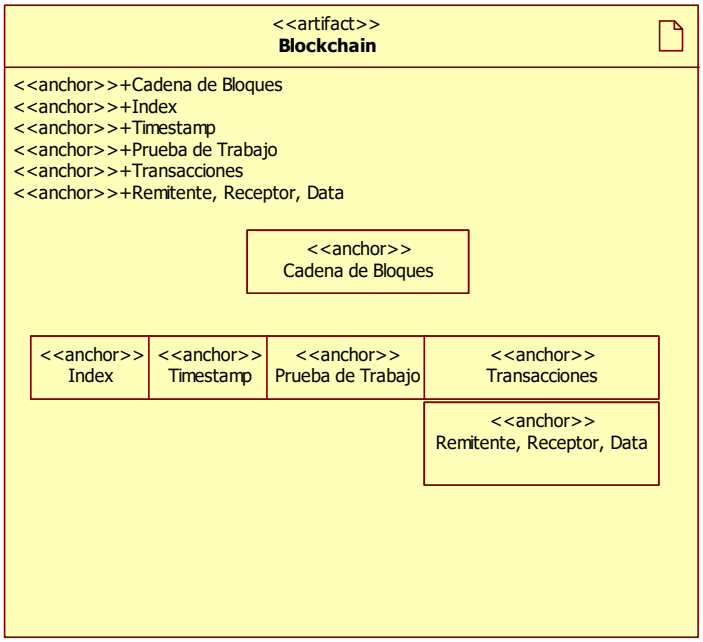


Figura N° 3.12: Diagrama de Presentación Blockchain

Fuente: Elaboración propia

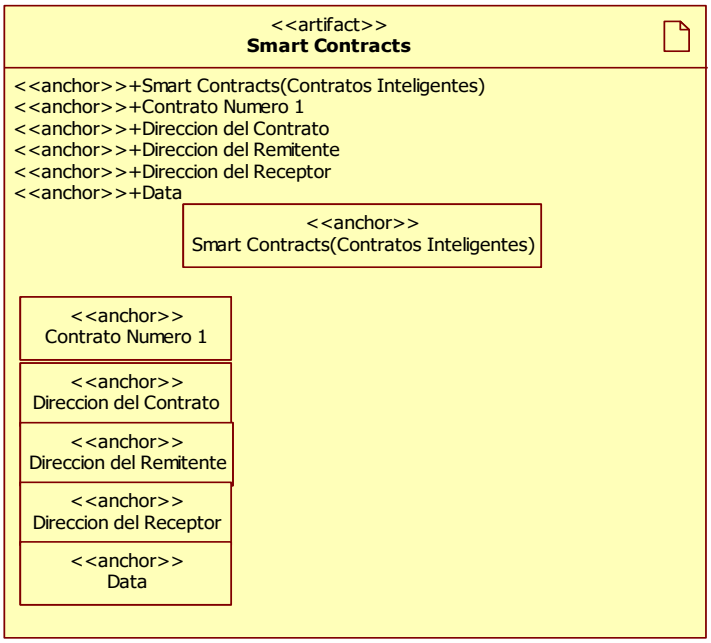


Figura N° 3.13: Diagrama de Presentación Smart Contracts

Fuente: Elaboración propia

3.6. REQUERIMIENTOS DE HARDWARE Y SOFTWARE

3.6.1. Hardware

Para obtener el buen funcionamiento del Modelo de Certificación se requieren las siguientes características en cuanto al hardware como se describe en la tabla.

Tabla N° 3.10: Descripción de Hardware

Hardware	Características
Equipo de Computación	Computadora de escritorio o Laptop
Memoria extraíble	Flash o Disco Duro

Nota. - Elaboración propia en base a ISO 25023

3.6.2. Software

Para obtener el buen funcionamiento del Modelo de Certificación se requieren las siguientes características en cuanto al software como se describe en la tabla.

Tabla N° 3.11: Descripción de Software

Requerimientos	Características
Sistema Operativo	- Windows 10
Lenguajes de Programación	- Python
Gestor de Base de Datos	- PostgresSQL
Framework	- Flask
Librerias	- Flask-SQLAlchemy, json - Flask url_for, render_template

Nota. - Elaboración propia

3.7. IMPLEMENTACIÓN

Ya realizadas las fases de especificación de requisitos, diseño del software, la codificación y las pruebas se implementa el modelo.

3.7.1. Captura de Ventanas del Modelo

3.7.1.1. Elaboración del Modelo

a) Pantalla de Inicio de sesión para acceso al Modelo, usuario = admin contraseña = admin

Figura N° 3.14: Inicio de sesión para ingresar al Modelo

Fuente: Elaboración propia

```
@app.route('/login/', methods=['GET', 'POST'])
def login():
    if request.method == 'GET':
        return render_template('login.html')
    else:
        u = request.form['username']
        p = request.form['password']
        data = User.query.filter_by(username=u, password=p).first()
        if data is not None:
            session['logged_in'] = True
            return redirect(url_for('index'))
        return render_template('index.html', message="Datos Incorrectos")

@app.route('/logout', methods=['GET', 'POST'])
def logout():
    session['logged_in'] = False
    return redirect(url_for('index'))
```

b) Pantalla del Modelo Blockchain

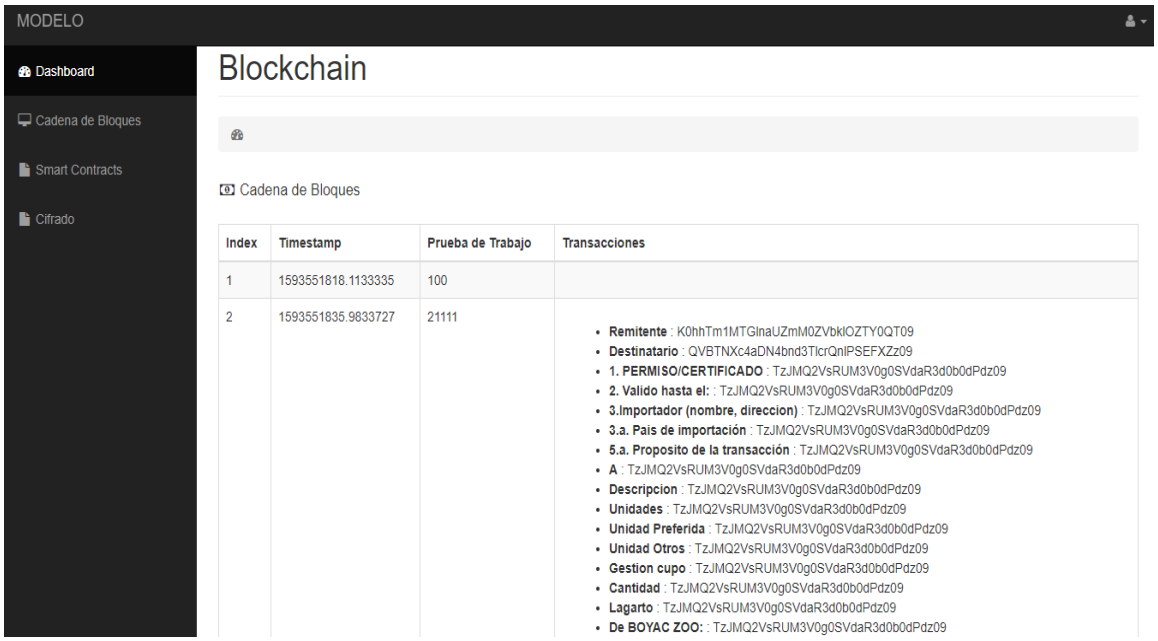


Figura N° 3.15: Pantalla Blockchain

Fuente: Elaboración propia

```
def new_cert(self):
        self.current_certf.append({
            'tipo': tipo,
            'especimen1': especimen1,
            'empresa_importador2': empresa_importador2,
            'pais_importacion': pais_importacion,
            'proposito': proposito,
            'especimen': especimen,
            'descripcion': descripcion,
            'cantidad': cantidad,
            'unidad': unidad,
            'unidad_otros': unidad_otros,
            'gestion': gestion,
            'lagarto': lagarto,
            'boyaca': boyaca,
            'boyacb': boyacb,
            'pais_origen': pais_origen,
            'puerto': puerto,
        })
        return self.last_block['index'] + 1
```

c) Pantalla de los Smart Contracts

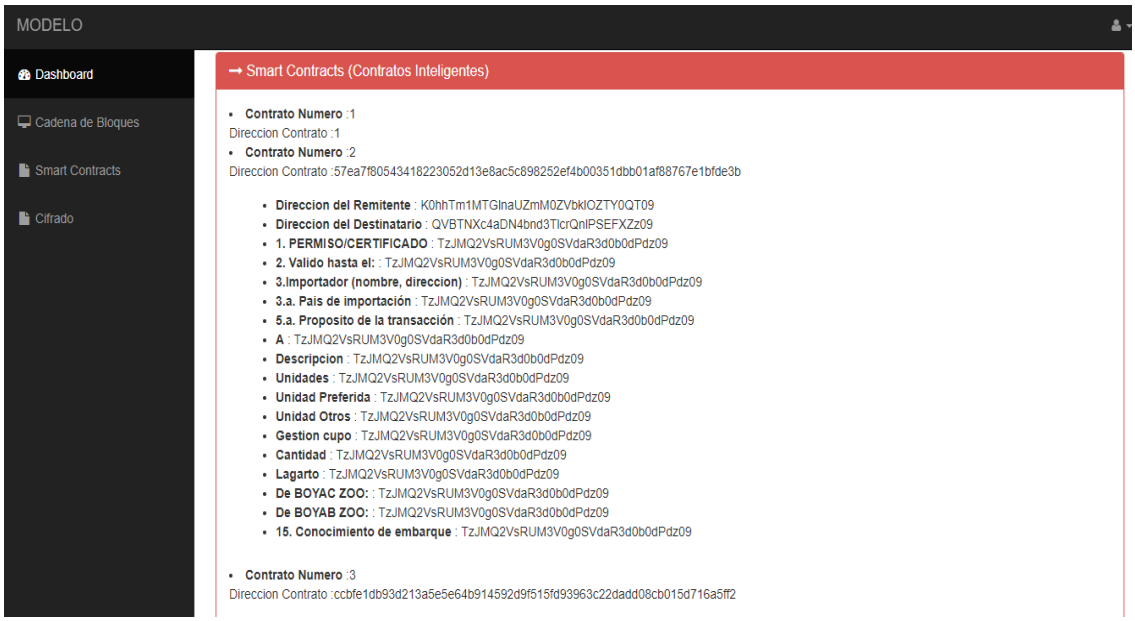


Figura N°.3.16: Pantalla Smart Contracts

Fuente: Elaboración propia

```
<tr class="table-primary">
    <li> <b>Contrato Numero</b> :{{ block.index }}</li>
     <b></b>Direccion Contrato</b> :{{ block.previous_hash }}<br>
     <td>
 {% for transaction in block.transactions %}
     <div class="panel-body">
        <ul>
        <li> <b>Direccion del Remitente</b> : {{transaction.sender}}</li>
         <b>Direccion del Destinatario</b> : {{transaction.recipi}}</li>
         <li> <b>1. PERMISO/CERTIFICADO</b> : {{transaction.tipo}}</li>
         <li> <b>2. Valido hasta el:</b> :{{transaction.especimen1}}</li>
         <li> <b>3.Importador</b>:{{transactio.empresa_importador2}}</li>
         <li> <b>Pais de importación</b> : {{tran.pais_importacion}}</li>
         <li> <b>5.a. Proposito</b> : {{transaction.proposito}}</li>
         <li> <b>A</b> : {{transaction.especimen}}</li>
         <li> <b>15. Conocimiento embarqu</b>:{{transaction.puerto}}</li>
         </ul>
       </div>
{% endfor %}
       </td>
</tr>
```

c) Backup Blockchain

Figura N° 3.17: Pantalla Backup Blockchain

Fuente: Elaboración propia

```
def resource_path(relative_path):
    try:
        base_path = sys._MEIPASS
    except Exception:
        base_path = os.path.abspath(".")
    return os.path.join(base_path, relative_path)
text_path = resource_path('blockChain.txt')
node_identifier = str(uuid4()).replace('-', '')
blockchain = Blockchain()
if os.stat(text_path).st_size > 0:
    file_object = open(text_path, 'rb')
    blockchain.chain = pickle.load(file_object)

    response = {
        'message': "New Block Forged",
        'index': block['index'],
        'transactions': block['transactions'],
        'proof': block['proof'],
        'previous_hash': block['previous_hash'],
    }
    return jsonify(response), 200
```

3.8. MÉTRICAS DE CALIDAD DEL SOFTWARE ISO/IEC 25000

La métrica de calidad ISO/IEC 25000 cuenta con sub características son las siguientes:

3.8.1. Adecuación Funcional

3.8.1.1. Completitud Funcional

- **Completitud de la implementación funcional**

  X = A / B => 4/5 => 0,8 => 80%

  A = Número de funciones que están incorrectas o que no fueron implementadas.

  B = Número de las funciones establecidas en la especificación de requisitos.

3.8.1.2. Exactitud Funcional

- **Exactitud**

  X = A/B => 7/8 => 0,875 => 88%

  A = Número de elementos de datos implementados con el estándar especifico de exactitud.

  B = Número total de elementos de datos implementados.

Tabla N° 3.12: Ponderación de Adecuación Funcional

Característica	Ponderación
Completitud de la implementación funcional	80%
Exactitud	88%
Total:	84%

Nota. - Elaboración propia en base a ISO 25023

3.8.2. Fiabilidad

3.8.2.1. Madurez

- **Eliminación de errores**

  X = A/B => 9/10 => 0,9 => 90%

  A = Número de fallas corregidas en la fase de diseño/codificación/pruebas.

  B = Número de fallas detectadas en las pruebas.

- **Cobertura de pruebas**

  X = 9/10 => 0,9 => 90%

  A = Número de casos de pruebas realizados en un escenario de operación durante la prueba.

  B = Número de casos de prueba a ser realizados para cubrir los requerimientos.

3.8.2.2. Disponibilidad

- **Tiempo de servicio**

  X = A/B => 4/5 => 0,8 => 80%

  A = Tiempo de servicio del sistema que se proporciona actualmente

  B = Tiempo de servicio del sistema regulado en el cronograma operacional

- **Tiempo medio de inactividad**

  X = A/T => 4/5 => 0,8 => 80%

  A = Número de fallos observados

  T = Tiempo total de inactividad

3.8.2.3. Tolerancia a Fallos

- **Prevención de fallas**

  X = A/B => 8/10 => 0,8 => 80%

  A = Número de ocurrencia de fallas evitadas contra los casos de pruebas de fallas iniciales.

  B = Número de casos de pruebas de fallas iniciales ejecutados durante las pruebas.

- **Anulación de operación incorrecta**

  X = A/B = 8/10 = 0,8 => 80%

  A = Número de operaciones incorrectas presentadas

  B = Número total de funciones implementadas para anular operaciones incorrectas

3.8.2.4. Recuperabilidad

- **Tiempo medio de recuperación**

  X = A / T => 8/10 => 0,8 = 80%

  A = Número de casos en los cuales se ha observado que el sistema entró en recuperación

  T = Tiempo que le tomó al sistema en recuperarse

Tabla N° 3.13: Ponderación de Fiabilidad

Característica	Ponderación
Eliminación de errores	90%
Cobertura de pruebas	90%
Tiempo de servicio	80%
Tiempo medio de inactividad	80%
Prevención de fallas	80%
Anulación de operación incorrecta	80%
Tiempo medio de recuperación	80%
Total:	83%

Nota. - Elaboración propia en base a ISO 25023

3.8.3. Eficiencia en el Desempeño

3.8.3.1. Comportamiento Temporal

- **Tiempo de respuesta**

  X = B - A = 10-9 => 100%

  A= Tiempo de envío de petición

  B = Tiempo en recibir la primera respuesta

- **Tiempo de espera**

  X = B - A => 10-9 => 100%

  A= Tiempo cuando se inicia un trabajo

  B = Tiempo en completar el trabajo

- **Rendimiento**

  X = A/T = 8/10 => 80%

  A= Número de tareas completadas

  T = Intervalo de tiempo

3.8.3.2. Utilización de Recursos

- **Utilización de los dispositivos de E/S**

  X =B-A => 4-3 => 100%

  A = Tiempo que los dispositivos de entradas y salidas pasan ocupados para realizar la tarea.

  B = Tiempo de operación

3.8.3.3. Capacidad

- **Número de peticiones online**

  X = A/T => 9/10 = 0,9 => 90%

  A= Número máximo de peticiones online procesada

  T = Tiempo de operación

- **Sistema de transmisión de ancho de banda**

  X = A/T => 8/10 => 0,8 => 80%

  A= Cantidad máxima de transmisión de datos

  T = Tiempo de operación

Tabla N° 3.14: Ponderación de Eficiencia en el desempeño

Característica	Ponderación
Tiempo de respuesta	100%
Tiempo de espera	100%
Rendimiento	80%
Utilización de los dispositivos de E/S	100%
Número de peticiones online	90%
Sistema de transmisión de ancho de banda	80%
Total:	92%

Nota. - Elaboración propia en base a ISO 25023

3.8.4. Facilidad de Uso

3.8.4.2. Capacidad para ser entendido

- **Efectividad de la documentación del usuario o ayuda del sistema**

  X = A / B => 9/10 => 0,9 = 90%

  A= Número de funciones descritas correctamente

  B = Número total de funciones implementadas

3.8.4.3. Protección contra errores del usuario

- **Verificación de entradas válidas.**

  X = A/B => 8/10 => 0,8 => 80%

  A= Número de ítems de entrada que son validados

  B = Número de ítems que necesitan ser validados

- **Prevención del uso incorrecto**

  X = A/B => 8/10 =>0,8 => 80%

  A = Número operaciones iniciales incorrectas

  B = Número de funciones implementadas para evitar fallos de funcionamiento provocados por un uso incorrecto

Tabla N° 3.15: Ponderación de Facilidad de Uso

Característica	Ponderación
Efectividad de la documentación del usuario o ayuda del sistema	90%
Verificación de entradas validas	80%
Prevención del uso incorrecto	80%
Total:	83%

Nota. - Elaboración propia en base a ISO 25023

3.8.5. Seguridad

3.8.5.1. Confidencialidad

- **Capacidad de control de acceso**

  X = A / B

  A = Número de diferentes tipos de operaciones ilegales detectados

  B = Número de tipos de operaciones ilegales en la especificación

  X = 9/10 = 0,9 => 90%

- **Encriptación de datos**

  X = A / B = > 10/10 => 100%

  A = Número de elementos de datos encriptados/ desencriptados correctamente

  B = Número de elementos de datos que requiere la encriptación/ desencriptación.

3.8.5.2. Integridad

- **Prevención de corrupción de datos**

  X = A / B

  A = Número de casos de corrupción de datos ocurridos en la actualidad

  B = Número de accesos donde se espera que ocurran daños de datos.

  X = 9/10 => 0,9 => 90%

3.8.5.3. Autenticidad

- **Métodos de autenticación**

  X = A => 90%

  A = Número de métodos de autenticación previstos

Tabla N° 3.16: Ponderación de Seguridad

Característica	Ponderación
Capacidad de control de acceso	90%
Encriptación de datos	100%
Prevención de corrupción de datos	90%
Métodos de autenticación	90%
Total:	93%

Nota. - Elaboración propia en base a ISO 25023

3.8.6. Compatibilidad

3.8.6.1. Interoperatividad

- **Conectividad con sistemas externos**

  X = A/B = 8/10 => 0,8 => 80%

  A= Número de interfaces implementadas con otros sistemas

  B = Número total de interfaces externas

- **Capacidad de intercambiar de datos**

  X = A/B => 9/10 => 0,9 => 90%

  A= Número de datos que se han intercambiado sin problemas con otro sistema

  B = Número total de datos que se intercambiarán

Tabla N° 3.17: Ponderación de Compatibilidad

Característica	Ponderación
Conectividad con sistemas externos	80%
Capacidad de intercambiar de datos	90%
Total:	85%

Nota. - Elaboración propia en base a ISO 25023

3.8.7. Mantenibilidad

3.8.7.1. Capacidad de ser Analizado

- **Capacidad de pistas de auditoría**

  X = A / B => 10/10 => 1 => 100%

  A = Número de datos realmente grabadas durante la operación

  B = Número de datos previstos a grabarse para controlar el estado del sistema durante la operación.

3.8.7.2. Capacidad de ser Modificado

- **Complejidad de modificación**

  X = A/T => 8/10 => 0,8 => 80%

  A = Número de modificaciones

  T = Tiempo de trabajo que le toma al desarrollador modificar

- **Índice de éxito de modificación**

  X = A/B => 8/10 => 80%

  A = Número de problemas dentro de un determinado período antes de mantenimiento

  B = Número de problemas en el mismo período después del mantenimiento.

3.8.7.3. Capacidad de ser Probado

- **Completitud funcional de funciones de pruebas**

  X = A/B = 9/10 => 0,9 => 90%

  A = Número de funciones de prueba implementadas

  B = Número de funciones de prueba requeridas

- **Capacidad de prueba autónoma**

  X = A/B = 9/10 => 0.9 => 90%

  A = Número de pruebas que están dependiendo de otros sistemas

  B = Número total de pruebas dependientes con otros sistemas

- **Capacidad de reinicio de pruebas**

  X = A/B = 8/10 => 0,8 => 80%

  A = Número de casos en los cuales el mantenedor puede pausar y restaurar las pruebas

  B = Número de casos de pausa en la ejecución de pruebas

Tabla N° 3.18: Ponderación de Mantenibilidad

Característica	Ponderación
Capacidad de pistas de auditoría	100%
Complejidad de modificación	80%
Índice de éxito de modificación	80%
Completitud funcional de funciones de pruebas	90%
Capacidad de prueba autónoma	90%
Capacidad de reinicio de pruebas	80%
Total:	87%

Nota. - Elaboración propia en base a ISO 25023

3.8.8. Portabilidad

3.8.8.1. Capacidad de ser Instalado

- **Eficiencia en el tiempo de instalación**

  X = A/T => 8/10 => 0.8 => 80%

  A = Número de reintentos al instalar el sistema

  T = Tiempo total transcurrido al instalar el sistema

- **Facilidad de instalación**

  X = A/B = 8/10 => 0.8 => 80%

  A = Número casos en que los usuarios tuvieron éxito al instalar el sistema cambiando proceso de instalación para su conveniencia

  B = Número total de casos en que los usuarios han intentado cambiar el proceso de instalación para su conveniencia.

Tabla N° 3.19: Ponderación de Portabilidad

Característica	Ponderación
Eficiencia en el tiempo de instalación	80%
Facilidad de instalación	80%
Total:	80%

Nota. - Elaboración propia en base a ISO 25023

3.8.9. Análisis de resultados

Tabla N° 3.20: Análisis de Resultados Iso 25000

N°	Característica	Preferencia	% de Preferencia
1	Adecuación Funcional	0.84	84%
2	Fiabilidad	0.83	83%
3	Eficiencia en el Desempeño	0.92	92%
4	Facilidad de Uso	0.83	83%
5	Seguridad	0.93	93%
6	Compatibilidad	0.85	85%
7	Mantenibilidad	0.87	87%
8	Portabilidad	0.80	80%
Evaluación de la Calidad Global		0.86	86%

Nota. - Elaboración propia en base a ISO 25000

Según [Pressman;2010], se define lo siguiente:

- Un nivel de aceptación satisfactorio, indica que los valores de preferencia se encuentran en el rango de 60-100.
- Un nivel de aceptación marginal, indica que los valores de preferencia se encuentran en el rango de 40-60.
- Un nivel de aceptación insatisfactorio, indica que los valores de preferencia se encuentran en el rango de 0-40.

Por lo tanto, la preferencia Global es de 86%. el trabajo es de nivel Satisfactorio.

3.9. EVALUACIÓN DE COSTOS DEL SOFTWARE

3.9.1. Modelo Cocomo II

3.9.1.1. Puntos de Función

Tabla N° 3.21: Calculo de punto de función no ajustados

Tipo de Función	Bajo	Intermedio	Alto	Total
Número de entradas de usuario	58	10	0	68
Número de salidas de usuario	15	15	15	45
Número de peticiones de usuario	5	10	34	49
Número de archivos	74	15	0	89
Número de interfaces externas	0	23	15	38
			TOTAL, UFP =	289

Nota. - Elaboración propia en base a Cocomo II

Tabla N° 3.22: Ponderación de factor de complejidad técnica

Numero de Factor	Factor	Valor 0 – 5
1	Mecanismos de recuperación y back-up confiables	4
2	Comunicación de Datos	3
3	Funciones de Procesamiento Distribuido	2
4	Performance	1
5	Configuración usada rigurosamente	2
6	Entrada de datos on-line	3
7	Factibilidad Operativa	3
8	Actualización de archivos on-line	2
9	Interfases Complejas	3
10	Procesamiento Interno Complejo	3
11	Reusabilidad	0
12	Fácil Instalación	3
13	Soporte de múltiples instalaciones	0
14	Facilidad de cambios y amigabilidad	3
	Factor de Complejidad Técnica (TCF)	32

Nota. - Elaboración propia en base a Cocomo II

Formula sobre el cálculo del Factor de Complejidad Técnica

$$\sum \text{Fi} = \text{TCF}$$

$$\sum \text{Fi} = \text{TCF} = 32$$

Con los cálculos obtenidos hasta el momento se procede a calcular el Punto de Función Ajustado reemplazando los datos en la siguiente fórmula de AFP.

$$\text{AFP} = \text{UFP} * (0.65 + (0.01 * \sum \text{Fi}))$$

$$\text{AFP} = 289 * (0.65 + (0.01*32))$$

$$\text{AFP} = 280.33$$

3.9.1.2. Aplicación de Cocomo II

Para la estimación del costo del proyecto, usara el método COCOMO II orientado a los puntos de función. Para determinar el costo de software desarrollado, se utiliza el resultado de punto de función, para convertir en líneas de código (LCD) con ayuda de la Tabla N° 3.23.

Tabla N° 3.23: Conversión de Puntos de Fusión a KLDC

LENGUAJE	NIVEL	FACTOR LDC/PF
C	2.5	128
JAVA	6	35
PL/I	4	80
VISUAL BASIC	7	46
ASP	9	36
PHP	11	29
VISUAL C++	9.5	34

Nota. - Elaboración propia en base a Pressman 2002

Para la conversión de PF a LDC, se aplica la siguiente formula:

KLDC = (AFP* Factor LDC/PF)/1000

KLDC = (280.33* 29) /1000

KLDC = 8.12957 [Miles de líneas de código]

Donde se deduce KLDC número (miles) estimado de líneas de código del Modelo

Los coeficientes que se usaran los valores que se detallan en la siguiente tabla N° 3.24.

Tabla N° 3.24: Coeficientes del Modelo Cocomo II

MODO	A	B	C	D
Orgánico	2.40	1.05	2.50	0.38
Semi Acoplado	3.00	1.12	2.50	0.35
Empotrado	3.60	1.20	2.50	0.32

Nota. - Elaboración propia en base a Cocomo II

Tabla N° 3.25: Costos de Cocomo II

Variable	Ecuación	Tipo
Esfuerzo Requerido por el proyecto	**$E = a * (KLDC)^b * FAE$**	**Personas / Mes**
Tiempo Requerido por el proyecto	**$T = c * (E)^d$**	**Meses**
Número de Personas requeridas para el desarrollo del proyecto	**NP = E/T**	**Personas**
Costo Total	**CT = Suelo Mes * NP*T**	**$us.**

Nota. - Elaboración propia en base a Cocomo II

Para hallar los valores de FAE se utiliza la Tabla N° 3.26.

Tabla N° 3.26: Calculo de atributos Fae

Atributos que afectan al coste	Valor					
	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
Atributos del Software						
Fiabilidad del software	0,75	0,88	1,00	1,15	1,40	
Tamaño base de datos		0,94	1,00	1,08	1,16	
Complejidad del producto	0,70	0,85	1,00	1,15	1,30	1,65
Atributos del Hadware						
Restricciones de tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria			1,00	1,06	1,21	1,56
Volatilidad de máquina virtual		0,87	1,00	1,15	1,30	
Tiempo de respuesta		0,87	1,00	1,07	1,15	
Atributos de Personal						
Capacidad de análisis	1,46	1,19	1,00	0,86	0,71	
Experiencia de aplicación	1,29	1,13	1,00	0,91	0,82	
Capacidad de programadores	1,42	1,17	1,00	0,86	0,70	
Experiencia en S.O. usado	1,21	1,10	1,00	0,90		
Experiencia lenguaje de programación	1,14	1,07	1,00	0,95		
Atributos del Proyecto						

Uso de técnicas de programación	1,24	1,10	1,00	0,91	0,82
Uso de herramientas de software	1,24	1,10	1,00	0,91	0,83
Restricciones de tiempo de desarrollo	1,23	1,08	1,00	1,04	1,10
TOTAL, FAE=			0,519		

Nota. - Elaboración propia en base a Cocomo II

- Cálculo del esfuerzo del desarrollo

$$E = a * (KLDC)^{b} * FAE$$

E = 2.40 * (8.12957) ^ 1,05 * 0,519

E = 11.24 personas /mes

- Cálculo tiempo de desarrollo

$$T = c * (E)^{d}$$

T = 2,50 * (11.24) ^ 0,38

T = 6.26 = 6 meses

- Personal promedio

NP = E/T

NP = 11.24 /6.26

NP = 1.79 = 2 personas

- Para calcular el costo final del modelo, en base a un sueldo promedio de 3000 Bs. mensual se aplica la siguiente formula:

$$CF = 3000*(NP*T)$$

$$CF = 3000*(12)$$

$$CF = 36000 \text{ Bs.} = 5216 \text{ \$us.}$$

En conclusión, de las fórmulas obtenidas que son:

- Costo de estimación del Modelo: 3000 Bs.
- Tiempo de desarrollo: 6 meses
- Número de personas para desarrollar el modelo: 2 personas

CAPITULO IV

PRUEBAS Y RESULTADOS

4.1. PRUEBA DE HIPÓTESIS

En el desarrollo de este capítulo se hizo el cálculo de la prueba de hipótesis planteada por el Modelo de certificación de Contratos Inteligentes con la Tecnología Blockchain, donde se utilizó la prueba de hipótesis.

4.1.1. Formulación de la hipótesis

Con la ingeniería de software y la tecnología Blockchain se obtendrá un modelo de certificación de contratos inteligentes aplicado a CITES con una eficiencia del 90%.

4.1.2. Estado de la Hipótesis

Para la demostración de la hipótesis se plantea lo siguiente:

Hipótesis

H1: “El modelo de certificación de contratos inteligentes con la aplicación o utilización de la tecnología Blockchain, permite la mejora en el proceso de la emisión de certificación cites con un nivel de confianza del 90%”.

Hipótesis Nula

Ho: “El modelo de certificación de contratos inteligentes con la aplicación o utilización de la tecnología Blockchain, no permite la mejora en el proceso de la emisión de certificación cites con un nivel de confianza del 90%”.

4.1.3. Prueba T-Student

Esta prueba T-Student es un modelo que se utiliza para contrastar la hipótesis sobre medias en muestras con la distribución normal, en otras palabras, lo que se busca es la determinación de las diferencias entre las medias de dos muestras que consiste en la evaluación antes y después.

4.1.4. Tamaño de la Muestra

Se realizó una muestra de 20 solicitudes emitidas por el usuario para la emisión de una certificación cite. Los cuales se distinguen de la siguiente manera, en la

tabla 4.1. se muestra los datos que se obtuvo del antes y después (solicitudes enviadas modificadas y no modificadas denotados como parámetros 0 y 1).

4.1.4.1. Procedimiento

A continuación, observamos los resultaos de la evaluación.

Tabla N° 4.1: Solicitud con y sin el uso de Blockchain

Numero	Muestra (X1)	Muestra (X2)
1	0	1
2	1	1
3	1	1
4	0	1
5	1	1
6	1	1
7	1	1
8	1	1
9	1	1
10	1	1
11	1	
12	1	1
13	0	1
14	0	1
15	0	1
16	0	1
17	0	1
18	1	1
19	1	1
20	1	1
Total	13	20

Nota. - Elaboración propia

- **Formulas a utilizar la Prueba T – Student presenta las siguientes fórmulas para el cálculo estadístico**

Donde:

n1 y n2 = Tamaños de la muestra

x1 = Solicitudes sin el uso de la tecnología Blockchain.

X2 = Solicitudes después del uso de la tecnología Blockchain.

Tomado en cuenta 0 = Datos modificados

Tomado en cuenta 1 = Datos no modificados

$$to = \frac{\overline{\mathrm{x1}} - \bar{\mathrm{x}}2}{\sqrt{S2 * \left[\left[\frac{1}{n1} + \frac{1}{n2}\right]\right]}}$$

Con n1+n2-2 grados de libertad.

Varianza común estimada:

$$S^2 = \frac{(n1-1)S_1^2 + (n2-1)S_2^2}{(n1+n2-2)}$$

Donde n1 y n2 tamaños muéstrales:

$$\overline{\mathrm{x1}}\ y\ \overline{\mathrm{x2}} = promedio\ de\ las\ muestras$$

Calculando Promedios

$$\overline{\mathrm{x1}} = \frac{\sum_1^n xi}{n} = \frac{13}{20} = 0.65$$

$$\overline{\mathrm{x2}} = \frac{\sum_1^n xi}{n} = \frac{20}{20} = 1$$

Cálculo de Varianzas

Se visualiza la tabla 4.2 con los siguientes datos de promedio que se obtuvo se calcula la varianza:

$$\overline{\mathrm{x1}} = 0.65;\ \overline{\mathrm{x2}} = 1$$

Tabla N° 4.2: Datos complementados para el cálculo de varianza

N°	M1 Xi	$(xi - \bar{x}1)$	$(xi - \bar{x}1)\ 2$	M2 Xi	$(xi - \bar{x}2)$	$(xi - \bar{x}2)\ 2$
1	0	-0,75	0.5625	1	0	0
2	1	0.25	0.0625	1	0	0
3	1	0.25	0.0625	1	0	0
4	0	-0.75	0.5625	1	0	0
5	1	0.25	0.0625	1	0	0
6	1	0.25	0.0625	1	0	0
7	1	0.25	0.0625	1	0	0
8	1	0.25	0.0625	1	0	0
9	1	0.25	0.0625	1	0	0
10	1	0.25	0.0625	1	0	0
11	1	0.25	0.0625	1	0	0
12	1	0.25	0.0625	1	0	0
13	0	-0.75	0.5625	1	0	0
14	0	-0.75	0.5625	1	0	0
15	0	-0.75	0.5625	1	0	0
16	0	-0.75	0.5625	1	0	0
17	0	-0.75	0.5625	1	0	0
18	1	0.25	0.0625	1	0	0
19	1	0.25	0.0625	1	0	0
20	1	0.25	0.0625	1	0	0
Total	13		4.75	20		0

Nota. - Elaboración propia

Cálculo de Varianza

$$S_1^2 = \frac{\sum(xi - \bar{x})\mathbf{2}}{n-1} = \frac{4.75}{19} = 0.25$$

$$S_2^2 = \frac{\sum(xi - \bar{x})\mathbf{2}}{n-1} = \frac{0}{19} = 0$$

Calculando la Varianza Común estimada

$$S^2 = \frac{(n1-1)S_1^2 + (n2-1)S_2^2}{(n1+n2-2)}$$

$$S^2 = \frac{(19) * 0.25 + (19) * 0}{31} = 0.153$$

Calculo estadístico "t" con $\overline{x1} = 0.65;\ \overline{x2} = 1$

$$t_{calculando} = \frac{\overline{x1} - \bar{x}2}{\sqrt{S2 * \left[\left[\frac{1}{n1} + \frac{1}{n2}\right]\right]}} = \frac{0.65 - 1}{\sqrt{0.153 * \left[\left[\frac{1}{13} + \frac{1}{20}\right]\right]}} = \frac{-0.35}{0.13} - 2.69$$

Nivel de Confianza y grados de libertad

∞ = 0.01; 1- 0.01 = 0.90 => Nivel de confianza

gl = 13+20-2 = 31 => grados de libertad

Uso de la tabla t –Student: Se busca lo siguiente en la tabla t-Student con los siguientes datos:

$$\frac{\infty}{2} = 0.005; gl = 31$$

Regla de Decisión, se tiene lo siguiente:

$$Si: t_{calculado} < t_{buscado} = Se\ rechaza\ Ho\ y\ se\ acepta\ H1$$

$$Si: t_{calculado} < t_{buscado} = Se\ rechaza\ Ho\ y\ se\ acepta\ H1$$

Por lo tanto, tenemos:

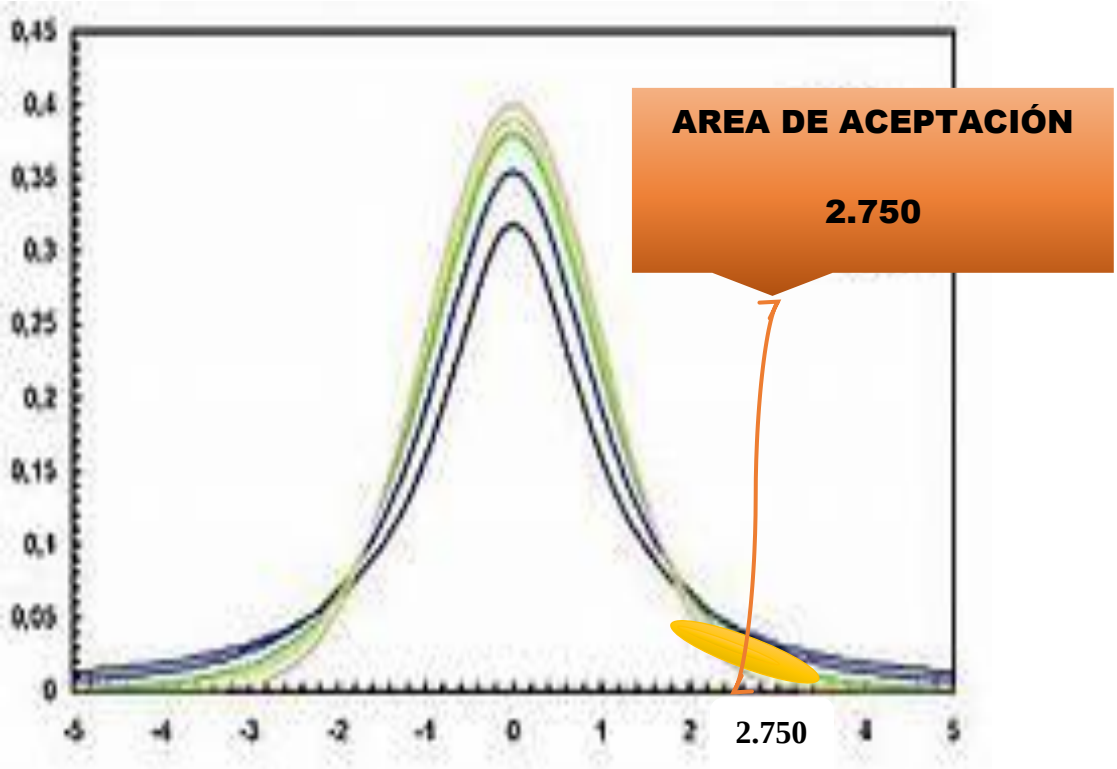
$$Si: t_{buscado} = 2.750\ y\ t_{calculado} = -2.69$$

Según la regla de decisión:

$$t_{calculado} < t_{buscado}$$

$$-2.69 < 2.750$$

Entonces se rechaza Ho y se acepta H1



4.1.4.2. Análisis de resultados

Tomado en cuenta el resultado final del porcentaje de solicitudes de emisiones de certificaciones de la muestra:

% Solicitudes sin el uso de la tecnología Blockchain = 65%

% Solicitudes con el uso de la tecnología Blockchain = 100%

Se observo que $t_{calculado} < t_{buscado}$ con un nivel de confianza del 90%, por lo cual según la regla de decisión de la prueba T-Student planteada, se llega a la conclusión que con el uso de la tecnología Blockchain es segura por lo que es aceptable para la solicitud de una certificación cites por lo tanto permite el mejor proceso de la emisión de certificación cites con un nivel del 90%.

4.2. PRUEBA DE CAJA NEGRA Y BLANCA

Al diseñar pruebas se descubre los errores de validación delo software encontrar el máximo de errores con la mínima cantidad de esfuerzo y tiempo.

4.2.1. Técnicas de Prueba de Caja Negra

Se trata de un enfoque que intenta descubrir diferentes tipos de errores que no se encuentran con los métodos de caja blanca.

Tabla N° 4.3: Pruebas de Caja Negra

N°	Propósito de prueba	Casos de Prueba		
		Datos de entrada 1	**Datos de entrada 2**	**Datos de entrada 3**
1	Que al ingresar al modelo genere resultados erróneos.	Ya que no se encuentra registrado, por eso genera el error.	No recuerda la contraseña, password invalido.	Quere ingresar sin loguearse, por eso devuelve la pina principal.
2	Que la Blockchain, ya no funcione correctamente.	Ya que el usuario intenta modificar la Blockchain y nos da el error.	No realizar cambios en la cadena validada, si no nos devuelve el error.	Querer registrar una transacción no valida, tal que devuelva error.

Nota. - Elaboración propia

4.2.2. Técnicas de Prueba de Caja Blanca

Para obtener el conjunto de caminos independientes se construirá el Grafo de Flujo asociado y se calculará su Complejidad referente al modelo. Posteriormente, se determinan los casos de prueba para cada camino básico,

luego se ejecutan y se comprueban los resultados, en las figuras y tablas ya que muestra las pruebas llevadas para cada módulo.

Grafo de flujo asociado al módulo procesos que sigue el Modelo. Por lo tanto, el diagrama de flujo V (G) asociado al módulo o código es el siguiente (ver figura. 4.1).

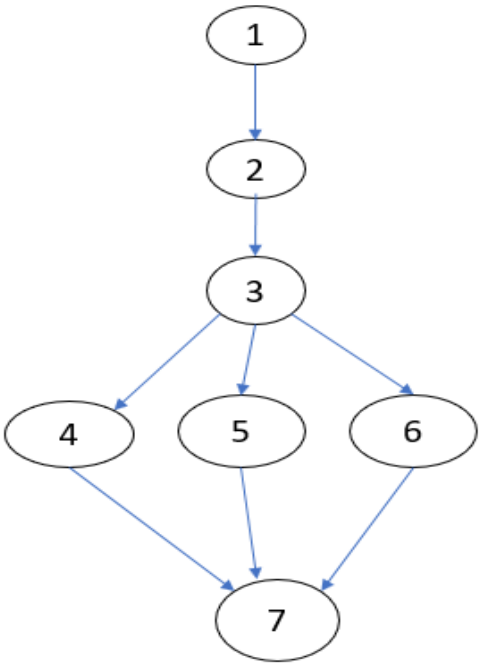


Figura N° 4.1: Grafo de flujo de modulo procesos del Modelo

Nota. - Elaboración propia

Cálculo de la Complejidad de los Procesos del Modelo

V (G)=A-N+2

Dónde:

A es el número de aristas y N es el número de nodos

En nuestro caso tenemos que A=8 y N=7. Entonces, se tiene que:

V(G)= 8-7+2 = 3

Por lo tanto, los procesos del Modelo son cuatro.

Determinación de los casos de prueba

Como V (G)= 3, existen tres caminos básicos. Estos son:

Camino básico 1: 1, 2, 3, 4, 7

Camino básico 2: 1, 2, 3, 5, 7

Camino básico 3: 1, 2, 3, 6, 7

Entonces, los casos de prueba se observan en la tabla N° 4.2:

Tabla N° 4.4: Casos de prueba de los procesos del Modelo

Caminos básicos	Descripción
1.	Se inicia la sesión o registra para ingresar al Modelo.
2.	Recibe datos enviados en transacciones, bloques por el usuario de manera cifrada.
3.	Revisa que cumpla con todos los requerimientos.
4.	Acepta las transacciones.
5.	Almacena en la Blockchain.
6.	Se ejecutan los Smart contracts.

Nota. - Elaboración propia

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

Una vez desarrollado y concluido el presente Trabajo de investigación denominado: “Modelo de Certificación de Contratos Inteligentes aplicando la tecnología Blockchain Certificaciones Cites”, se llegó a las siguientes conclusiones en base a los objetivos planteados.

- Se ha realizado la implementación de los Smart contracts o contratos inteligentes en la solicitud de las certificaciones cites.
- Se ha implementado un sistema de certificación vía web con Blockchain que faciliten su transmisión.
- Se ha diseñado la arquitectura de solución para la generación de certificados digitales.
- Se ha estudiado los algoritmos criptográficos que contribuyen a la seguridad que permite de manera segura, agregar la información de la certificación cites en transacciones, bloques, finalmente en la cadena de bloques encriptado con la función Sha-256.
- Se ha realizado la investigación sobre la tecnología Blockchain, por lo que se ha llegado a concluir este objetivo para lo cual se ha realizado estudios sobre sus pasos que sigue esta tecnología que almacena los datos en transacciones, bloques después en la cadena de bloques.
- Por último, se ha realizado el cumplimiento de objetivo general: “Diseñar un Modelo de Certificación de Contratos Inteligentes aplicando la tecnología Blockchain Certificaciones Cites”.

5.2. RECOMENDACIONES

A partir del estudio realizado e implementado el Modelo de Certificación Cites se recomienda lo siguiente:

- Como otra alternativa, se recomienda utilizar otras aplicaciones de nuevas tecnologías que ayuden el proceso de emisión de certificaciones cites y así evitar la falsificación de datos.
- Para una buena compresión de la presente investigación Modelo de certificación de contratos inteligentes, se recomienda la lectura sobre Blockchain.
- Para obtener una mejor seguridad en cites, se recomienda estudiar los algoritmos de encriptación más seguros.
- Se recomienda realizar el uso de los algoritmos de encriptación que coadyuve a cites.
- Para las futuras investigaciones de la emisión de certificaciones cites, se recomienda realizar una investigación más amplia tener precisión en los datos futuros.

BIBLIOGRAFÍA

BIBLIOGRAFÍA

Aguirre, J. (2018). Curso de Criptografía Aplicada. España.

Albuixech, R. (2018). Estudio de la tecnologia blockchain. España.

Alvarez, R. (2016). Diseño y desarrollo de propiedades inteligentes. Madrid.

Andrés, C. (2012). Pasos del método de Ingeniería.

Antezana, N. (2012). CITES Convención sobre el Comercio Internacional de Especies Amenazadas de Fauna y FLORA Silvestre. La Paz: Bolivia.

Astilla, M. (2009). Diseño de un Esquema de Comunicación Segura Utilizando como Criptosistema Simétrico a Triple DES y Asimétrico a RSA. México.

Asuad, N. (2014). Marco logico de la investigacion cientifica.

Basaldúa, D. (2005). Seguridad en Informática. México.

Becerra, A. (2015). Aspectos de Seguridad Bitcoin. Mexico.

Benoso, B. (2016). Cifrado y distribución de documentos vía web utilizando algoritmos Triple DES-96 y RSA. México.

Bermúdez, A. (2016). Estudio de la utilización de protocolos Blockchain en sistemas de votación electrónica. Barcelona.

Blanco, C. (s.f.). Ingeniería de Software.

Boehm. (1981). Estimacion de costo de software.

Castán, Y. (s.f.). Introducción al Método Científico y sus Etapas.

Castillo, E. (2017). Viabilidad de un Sistema Alternativo de envío de Remesas a través de Criptomonedas. Venezuela.

Castro, C. C. (2014). Metodologia Uwe.

Champagne, P. (2014). El Libro de Satoshi Blockchain. España.

cites, s. (2010). Cites.

Cortes, A. (2008). Capas de Aplicación . Cisco Networking Academy.

Garavito, J. (2007). PostgreSql. Bogota.

Guerrero, N. (s.f.). UWE en Sistema de Recomendación de Objetos de aprendizaje.

Gutiérrez, D. (2011). Arquitectura de Software. Universidad de los Andes.

Hashmania. (2019). Python para Neofitos.

Herrera. (2014). Web 2.0 componetes.

Junestrand. (2018). Blockchain en el sector publico.

Kindley, G. (2016). Redes de Computadoras Arquitectura Cliente – Servidor.

Koller, S. (2017). Modelo de votacion electronica con blockchain. Sistema de votacion electronica, 5,6.

Lopez, M. (2018). Blockchain como Desarrollador.

Marini, E. (2012). El Modelo Cliente/Servidor.

Marquez. (2018). Blockchain el auge de las transacciones.

MArquez. (2018). Blockchain el auge de las transacciones.

Martin. (2014). Intranet.

MArtin. (2014). Intranet.

Mattion. (2015). Modelo.

Méndez, G. (2009). Proceso Software y Ciclo de Vida. Madrid.

Molina, G. (2015). Aplicaciones segun ISO/IEC. Madrid.

Morales, S. (2017). Sistema para la gestion de historias clinicas con Blockchain. Historias clinicas con blockchain, 4.5.

Nakamoto, S. (2016). Bitcoin.

Navarro. (2013). Metodologias Agiles.

Navarro, B. (2016). Blockchain y sus aplicaiones. Paraguay.

Núñez, G. (2004). Arquitecturas de Software.

Olsina, L. (2005). Medición y Evaluación de Calidad en Uso de Aplicaciones Web.

Ortega. (2010). Bitcoin.

Ortiz, F. (2013). Administrador de Base de Datos Open source Postgresql . Ecuador.

Otazu, A. (s.f.). Un Modelo de Estimación de Proyectos de Software.

Pasini, A. (2018). Asistente para la Evaluación de Calidad de Producto de Software Según la Familia de Normas ISO/IEC 25000.

Pavon, J. (2013). Php.

Pavon, J. (2014). Aplicaciones Web/Sistemas Web. Madrid.

Pressman, R. S. (2010). Ingenieria del software. 7.

Preukschat, A. (2017). Blockchain la revolucion industrial de Internet. España.

Quiroga, M. (2011). Estructura Json.

Ramírez, D. (2019). Diseño y Desarrollo de Diplomas Académicos Digitales Mediante la Tecnología Blockchain.

Rey, M. (2015). Criptografía: Matemáticas para proteger la información.

Reyes, A. (2003). Seguridad del Software y Criterios de Evaluación. . Guatemala.

Rumbaugh, B. (2007). Modelo.

Saavedra, L. (2018). Flask. Bolivia.

Sanchez, D. (2018). Blockchain y el internet del valor. España.

Segura, M. (s.f.). Operacionalización de variables. DISAN.

Sierra, N. (2017). Modelo de calidad de software. Ecuador.

Silva, R. (2019). Desarrollo de Blockchain. Chile.

Sommerville, I. (2005). Ingenieria de Software. 7.

Stefanescu, D. (2019). Blockchain estudio de alternativas e implementaciones. Colombia.

Urdaneta, M. (2018). Diseño y arquitectura de un sistema de contratos intligentes basada en la tecnologia blockchain . Colombia.

Vedia, R. (2013). Proyecto Emisión Electrónica de Permisos CITES en los Países Miembros de la Organización del Tratado de Cooperación Amazónica (OTCA). Bolivia.

Vicente, E. (2011). Modelo Vista Controlador.

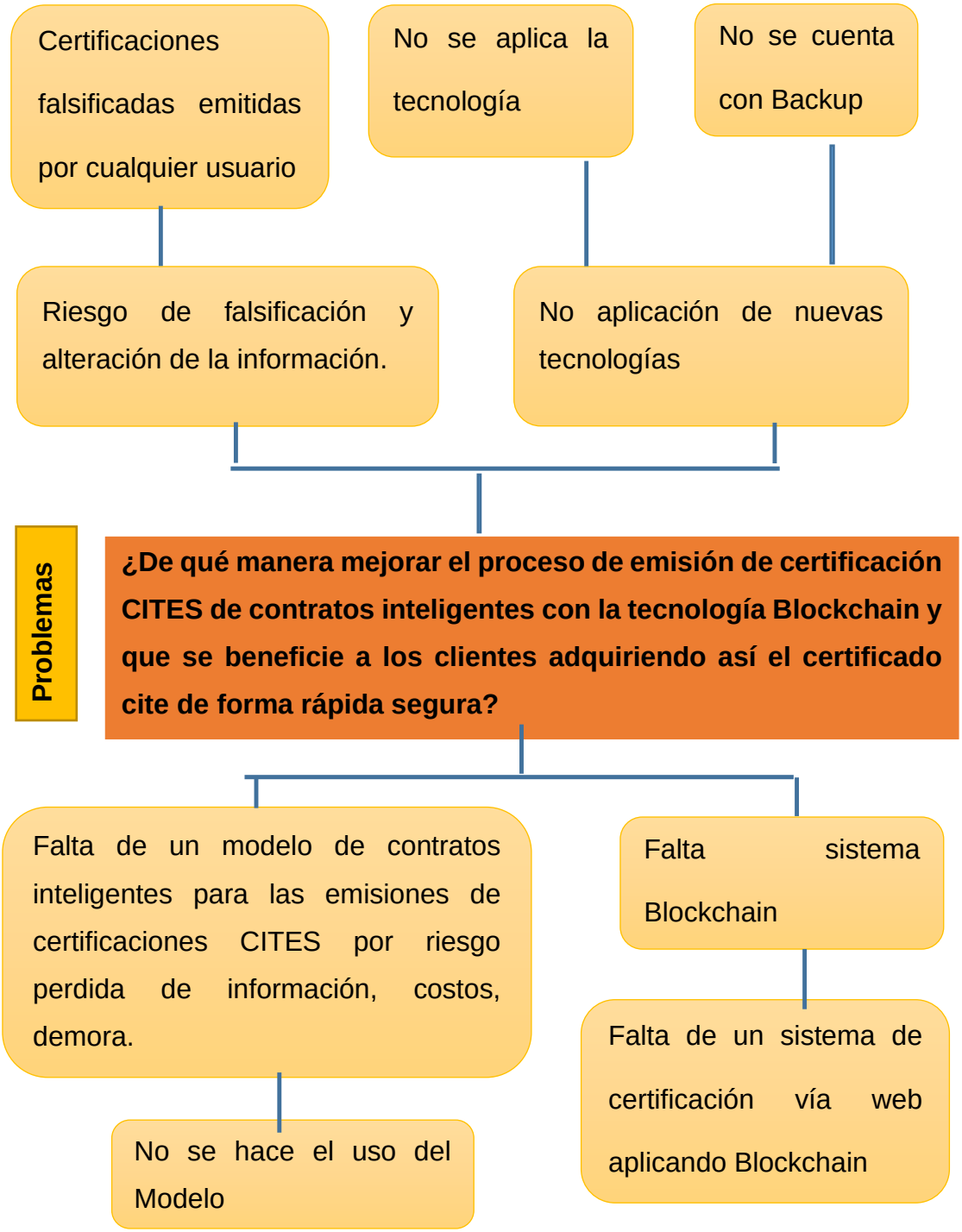
Wikipedia/Certificacion. (2018). Certificacion. Certificacion.

Wikipedia/Certificado. (2019). Certificacion. Certificacion.

ANEXOS

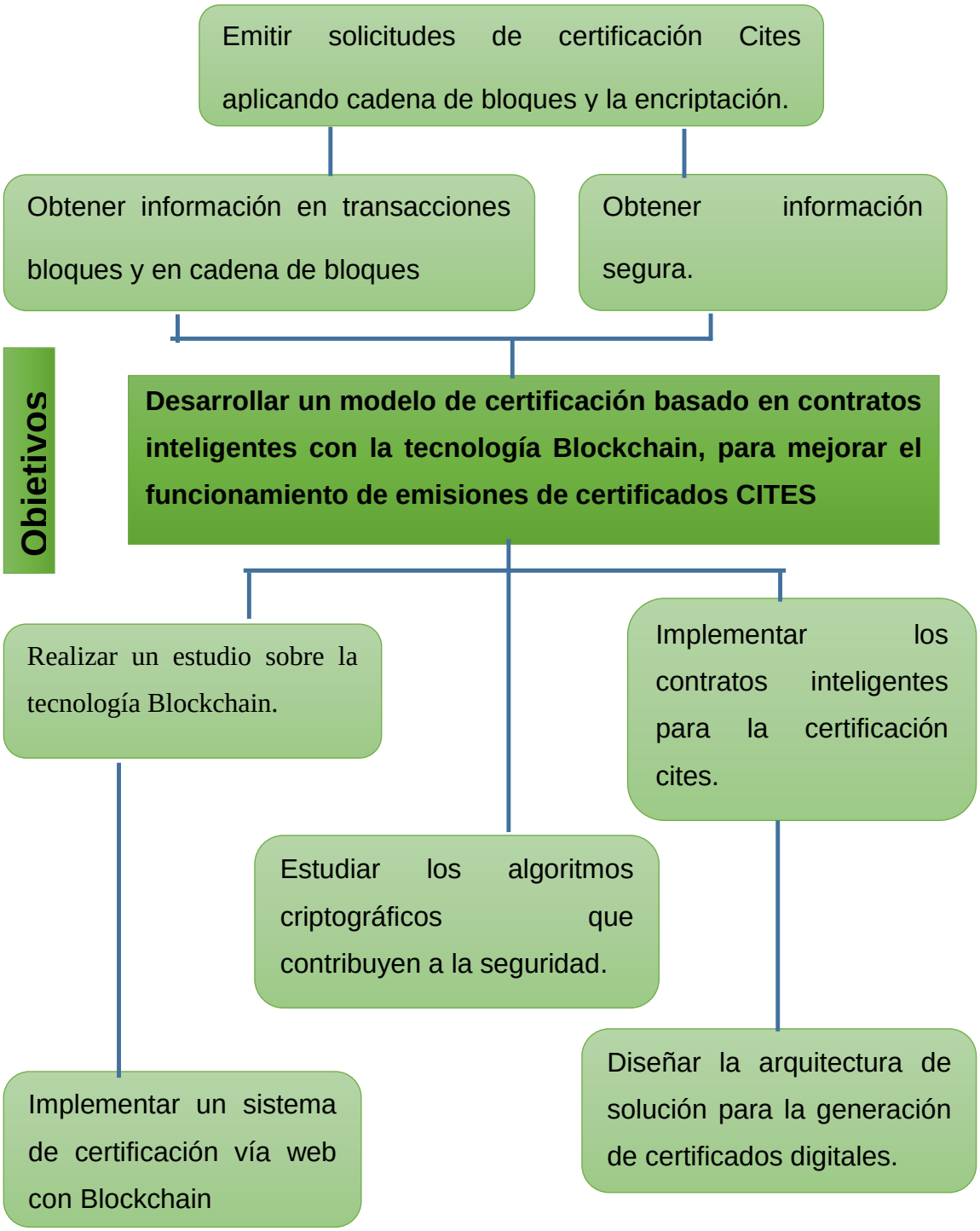
ANEXO A.1

ARBOL DE PROBLEMAS



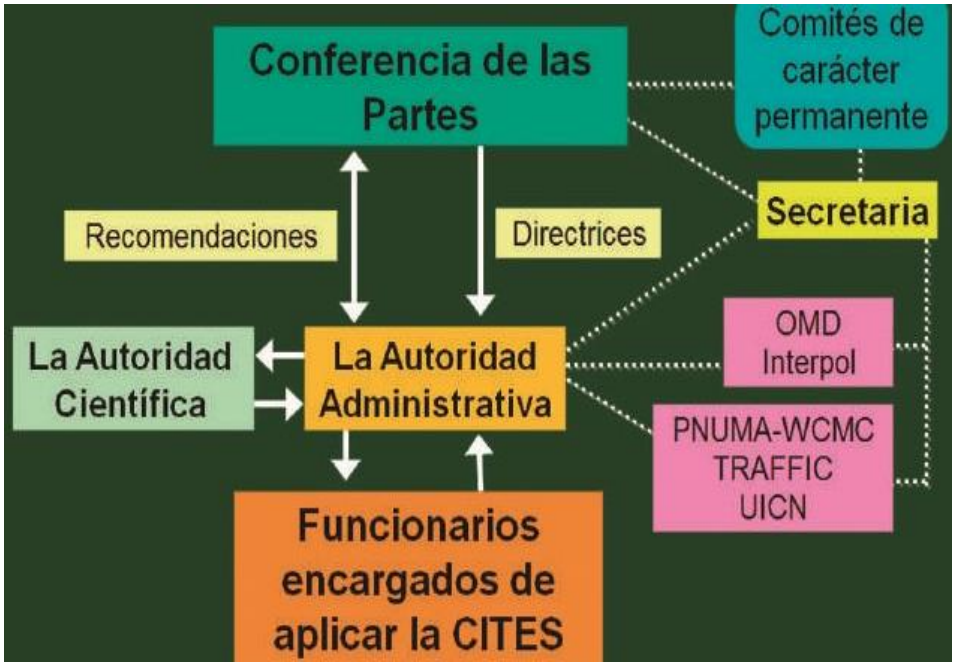
ANEXO A.2

ARBOL DE OBJETIVOS

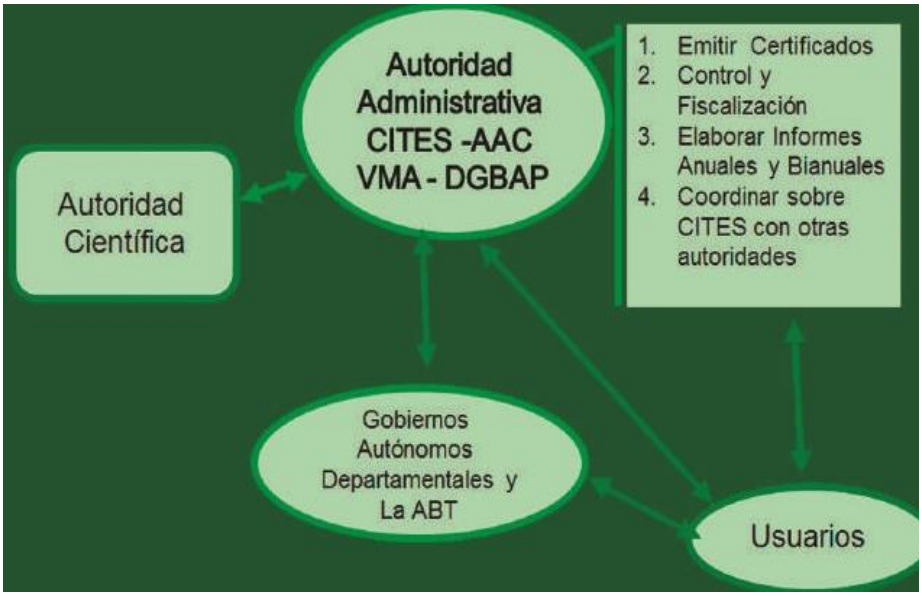


ANEXOS A.3

Estructura de la CITES

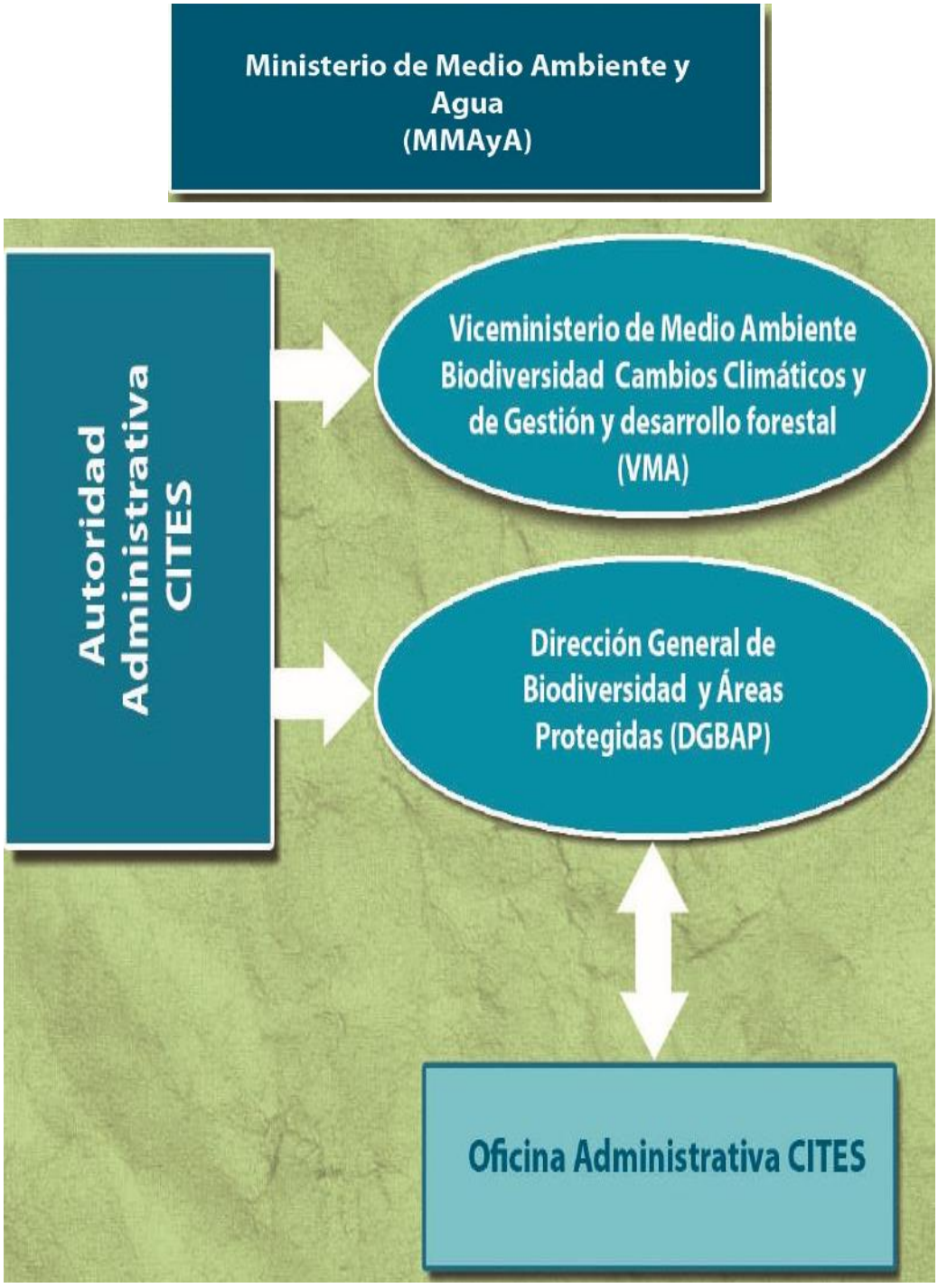


Resoluciones Ministeriales de designación de Autoridad Científica Cites



ANEXO A.4

Estructura de la CITES en Bolivia



MANUAL DE USUARIO

ANEXO A.5

Manual de Usuario

1. INGRESO AL SISTEMA CITES

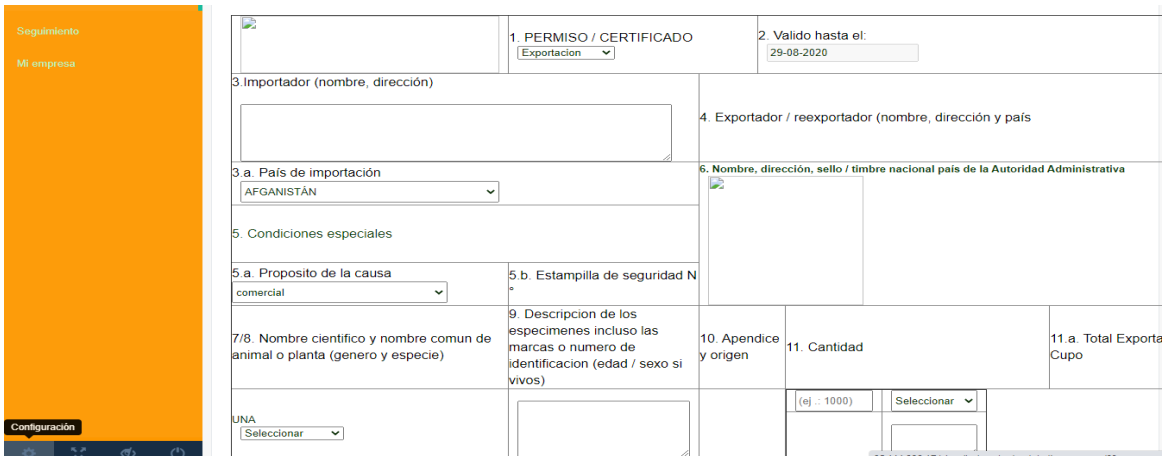
Se está aplicando el uso del sistema de Cites de Bolivia, ya que como se describió anteriormente se cuenta con este sistema, pero no cuenta con seguridad motivos por el cual se planteó esta problemática a continuación mostramos el funcionamiento de cites, también nuestro aporte a este sistema que es la Blockchain.

2. LOGIN DE USUARIO CITES

Primeramente, el usuario debe loquearse si está registrado para emitir una certificación cites.

3. FORMULARIO DE SOLICITUDES CITES

Después debe llenar este formulario que le permitirá registra y solicitar la certificación cites.



4. TRANSACCIONES ENVIADAS EN BLOQUES

string(41) "{"Mensaje":"Transaccion en el Bloque 3"} "

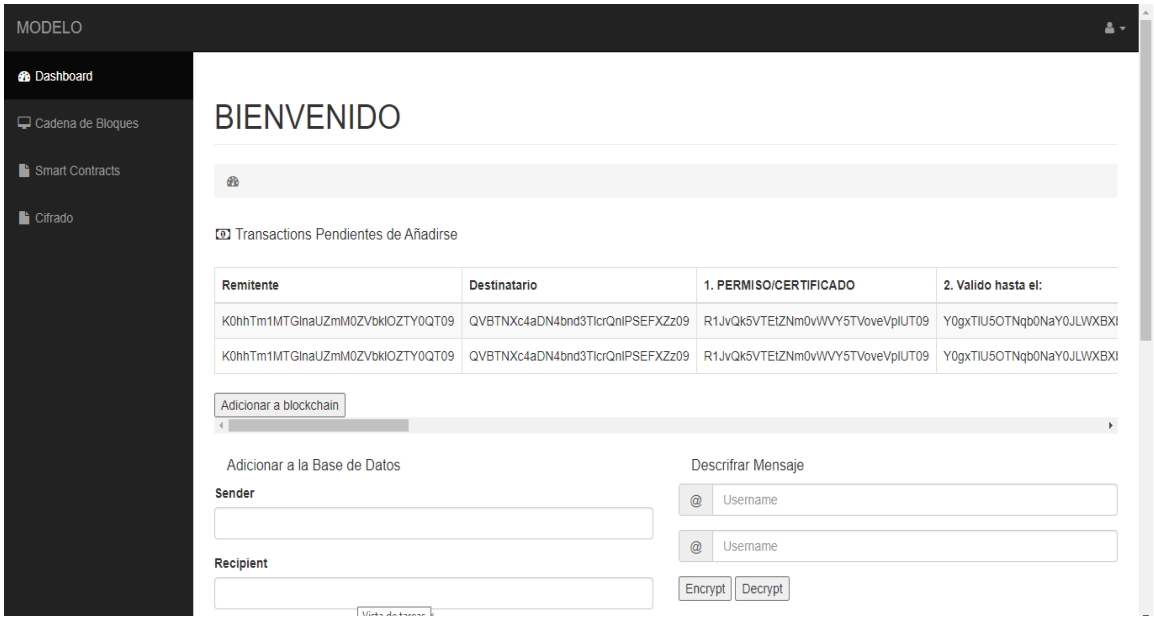
En esta sección se aplicó el trabajo del Modelo que en vez de realizar él envió a la base de datos se la envió al Modelo Blockchain, envía toda esa transacción y la almacena en bloques en este caso es el Bloque N° 3 como se muestra en la imagen.

5. LOGIN DEL MODELO

Primero debe loquearse para poder ingresar al Modelo.

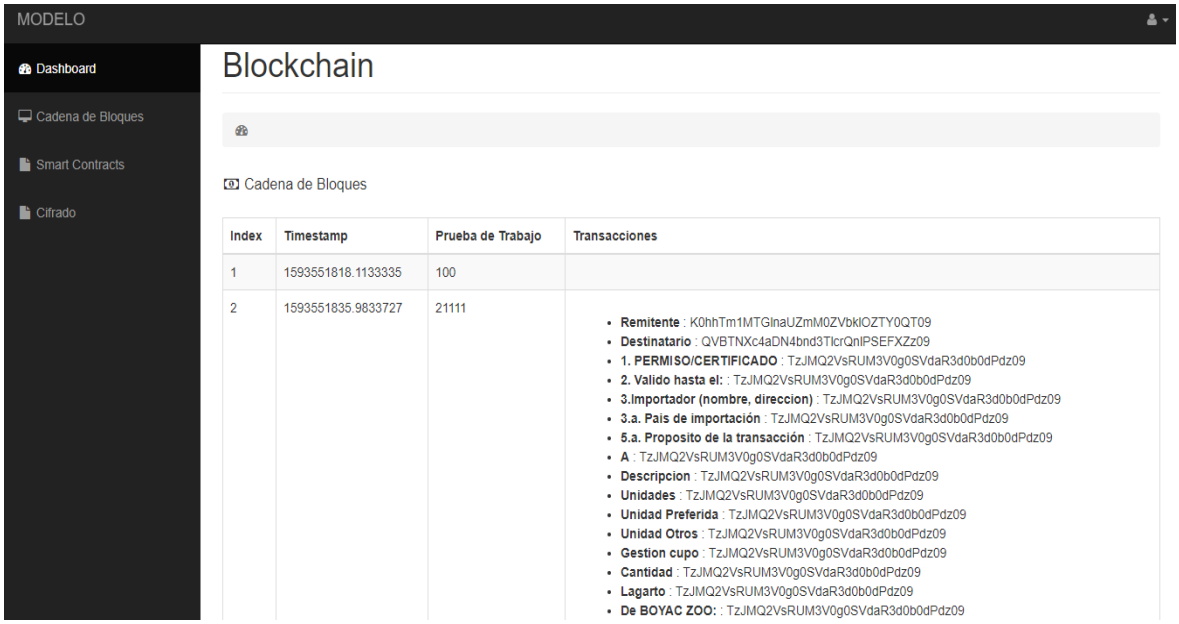
6. TRANSACCIONES RECIBIDAS

Después llega al modelo nos indica transacciones pendientes de añadirse en ese sector se verifica si cumple o no con todo si es así se la envía a la Blockchain



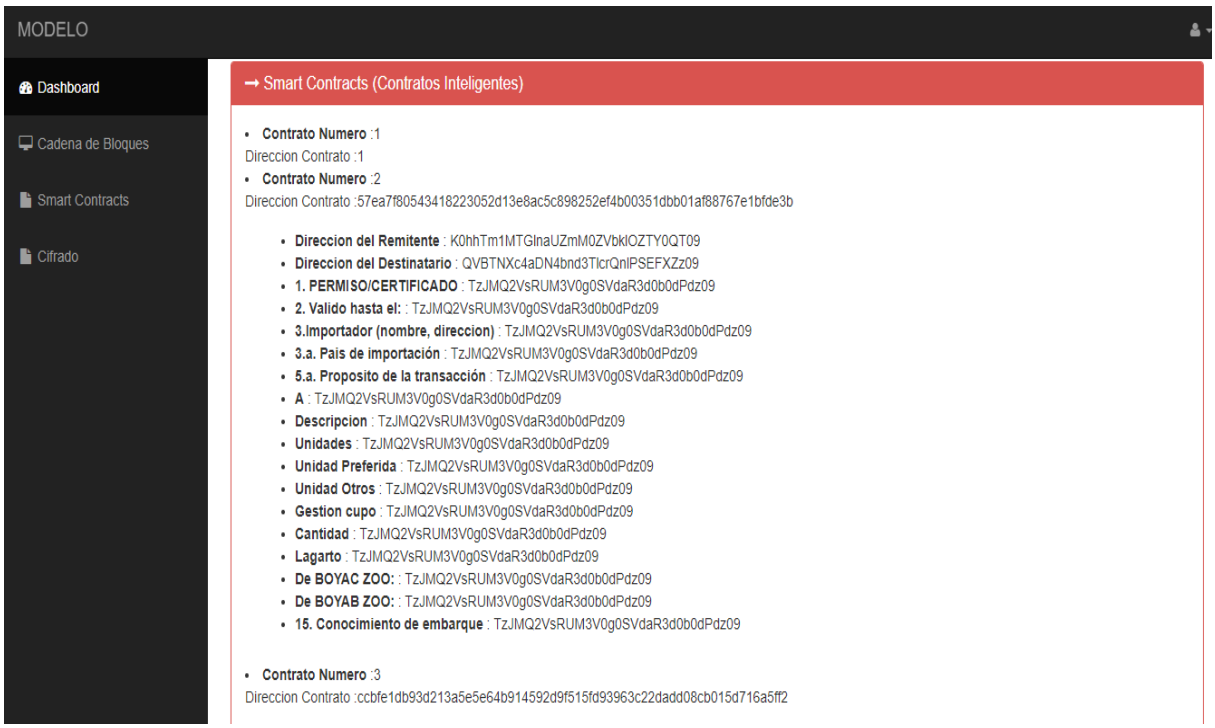
7. BLOCKCHAIN

Si se aprueba se la almacena en la Blockchain que funciona como su fuera un base de datos encadenada por un hash Sha-256.



8. SMART CONTRACTS

Y así se ejecutan los Smart Contracts en base a la Blockchain.



9. SMART CONTRACTS

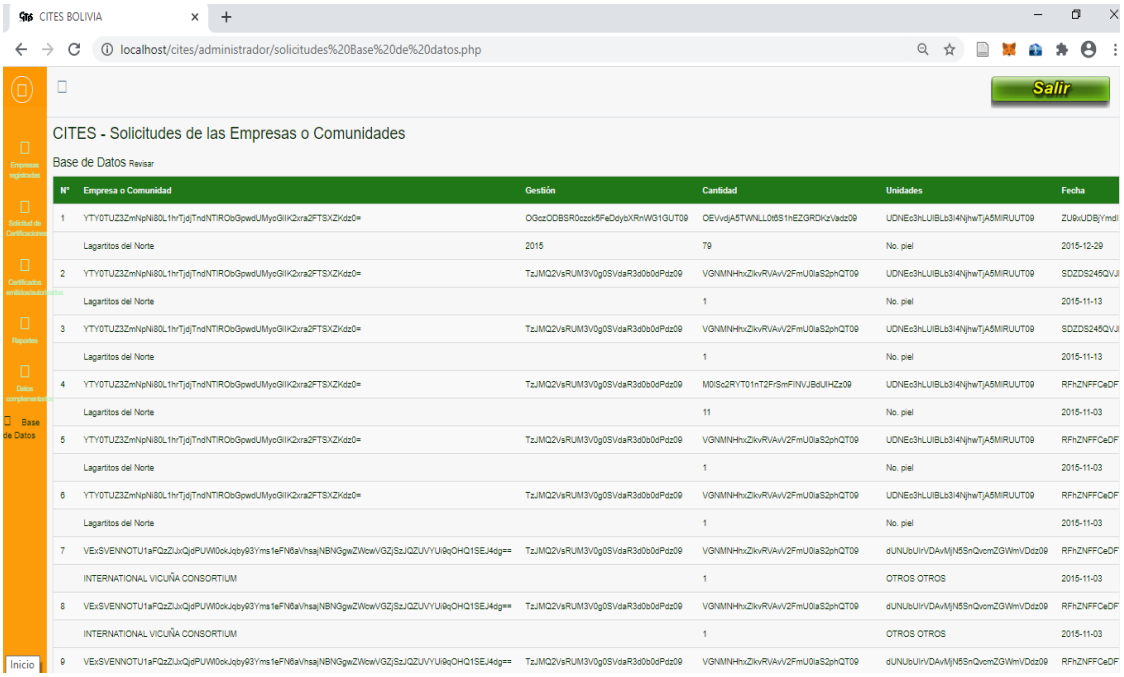
Y así se muestra el backup e la Blockchain.

10. LOGIN DE ADMINISTRADOR

Se está realizando también el modo administrativo para que pueda ver las pruebas o resultados, el administrador debe loquearse para comparar resultados, para así verificar en donde se está sufriendo cambios.

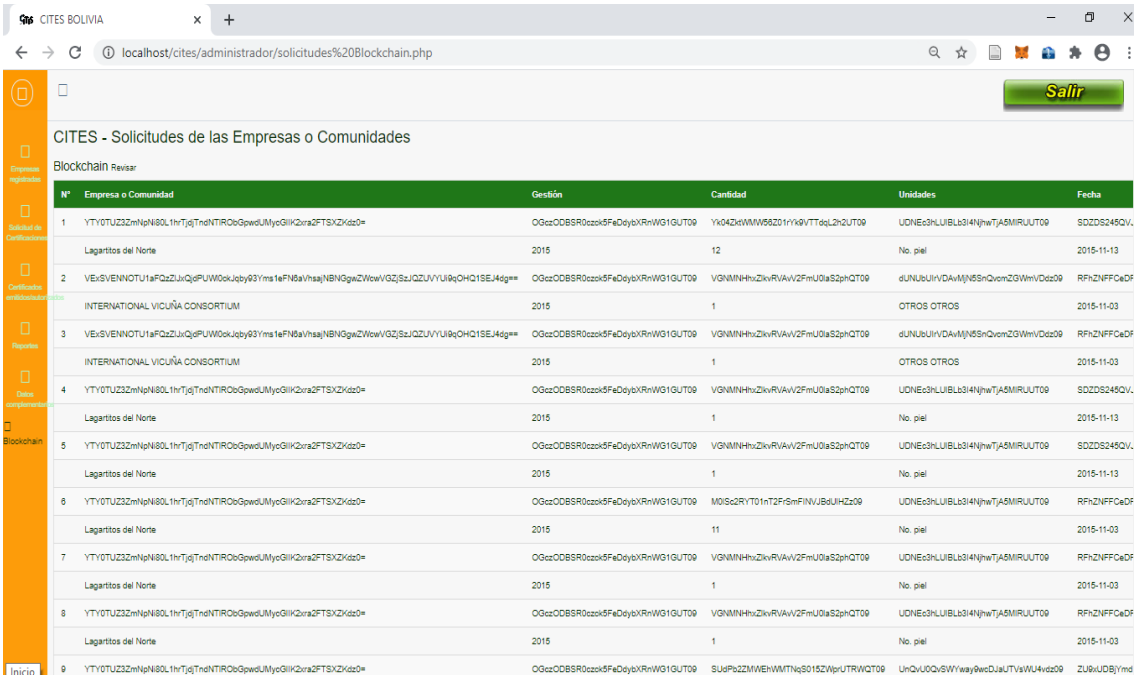
11. RESULTADOS DE LA BASE DE DATOS

Aquí se muestra los resultados comparaciones de la base de datos.



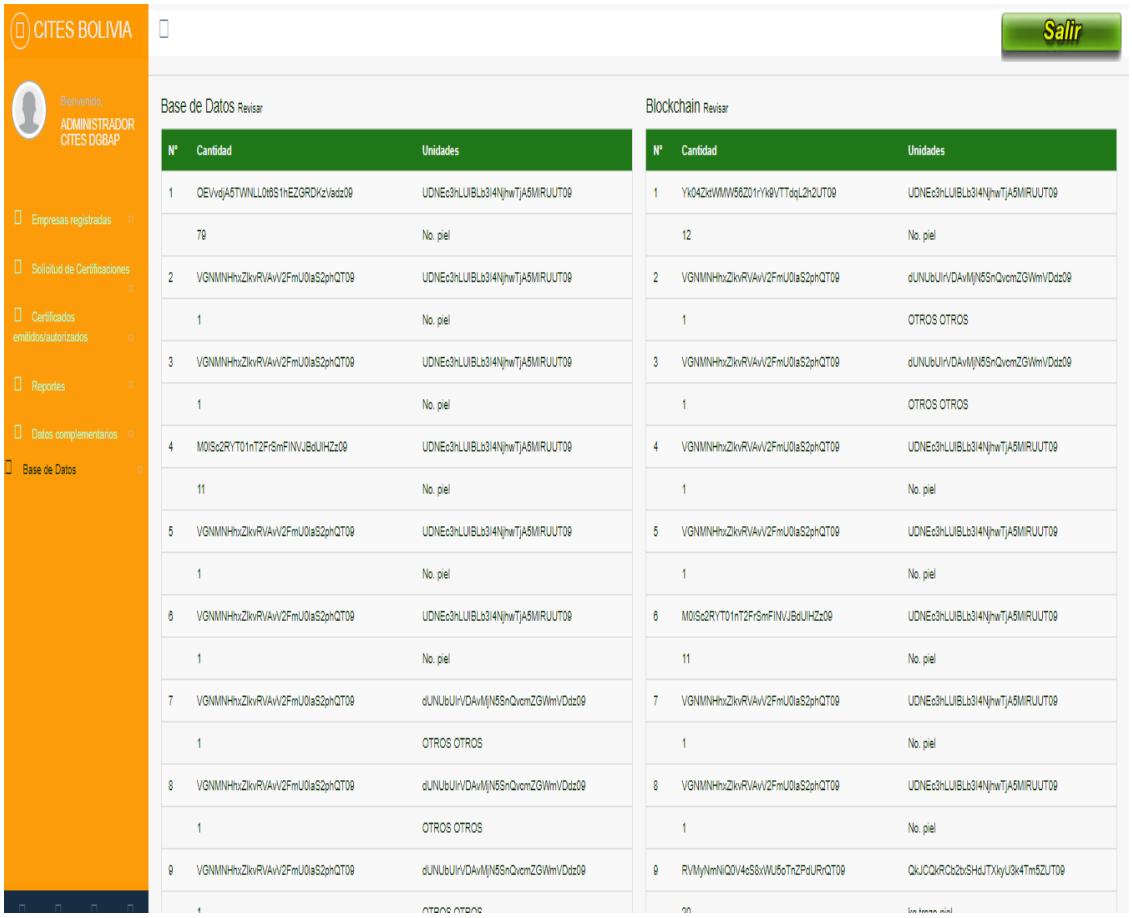
12. RESULTADOS DE LA BLOCKCHAIN

Aquí se muestra los resultados comparaciones de la Blockchain.



13. RESULTADOS COMPARATIVOS

Aquí se muestra los resultados comparaciones de la Blockchain y la Base de Datos, así el administrador podrá comparar los resultados y así verificar en que transacción se realizó la alteración de los datos.



Nº	Cantidad	Unidades
1	OEVvdjA5TWNLL0t6S1hEZGRDKzVadz09	UDNEc3hLUlBLb3l4NjhwTjA5MlRUUT09
	79	No. piel
2	VGNMNHhxZlkvRVAvV2FmU0laS2phQT09	UDNEc3hLUlBLb3l4NjhwTjA5MlRUUT09
	1	No. piel
3	VGNMNHhxZlkvRVAvV2FmU0laS2phQT09	UDNEc3hLUlBLb3l4NjhwTjA5MlRUUT09
	1	No. piel
4	M0lSc2RYT01nT2FrSmFlNVJBdUlHZz09	UDNEc3hLUlBLb3l4NjhwTjA5MlRUUT09
	11	No. piel
5	VGNMNHhxZlkvRVAvV2FmU0laS2phQT09	UDNEc3hLUlBLb3l4NjhwTjA5MlRUUT09
	1	No. piel
6	VGNMNHhxZlkvRVAvV2FmU0laS2phQT09	UDNEc3hLUlBLb3l4NjhwTjA5MlRUUT09
	1	No. piel
7	VGNMNHhxZlkvRVAvV2FmU0laS2phQT09	dUNUbUlrVDAvMjN5SnQvcmZGWmVDdz09
	1	OTROS OTROS
8	VGNMNHhxZlkvRVAvV2FmU0laS2phQT09	dUNUbUlrVDAvMjN5SnQvcmZGWmVDdz09
	1	OTROS OTROS
9	VGNMNHhxZlkvRVAvV2FmU0laS2phQT09	dUNUbUlrVDAvMjN5SnQvcmZGWmVDdz09

Nº	Cantidad	Unidades
1	Yk04ZktWMW56Z01rYk8VTTdqL2h2UT09	UDNEc3hLUlBLb3l4NjhwTjA5MlRUUT09
	12	No. piel
2	VGNMNHhxZlkvRVAvV2FmU0laS2phQT09	dUNUbUlrVDAvMjN5SnQvcmZGWmVDdz09
	1	OTROS OTROS
3	VGNMNHhxZlkvRVAvV2FmU0laS2phQT09	dUNUbUlrVDAvMjN5SnQvcmZGWmVDdz09
	1	OTROS OTROS
4	VGNMNHhxZlkvRVAvV2FmU0laS2phQT09	UDNEc3hLUlBLb3l4NjhwTjA5MlRUUT09
	1	No. piel
5	VGNMNHhxZlkvRVAvV2FmU0laS2phQT09	UDNEc3hLUlBLb3l4NjhwTjA5MlRUUT09
	1	No. piel
6	M0lSc2RYT01nT2FrSmFlNVJBdUlHZz09	UDNEc3hLUlBLb3l4NjhwTjA5MlRUUT09
	11	No. piel
7	VGNMNHhxZlkvRVAvV2FmU0laS2phQT09	UDNEc3hLUlBLb3l4NjhwTjA5MlRUUT09
	1	No. piel
8	VGNMNHhxZlkvRVAvV2FmU0laS2phQT09	UDNEc3hLUlBLb3l4NjhwTjA5MlRUUT09
	1	No. piel
9	RVMyNmNlQ0V4cS8xWU5oTnZPdURrQT09	QkJCQkRCb2txSHdJTXkyU3k4Tm5ZUT09

AVAL DE CONFORMIDAD

El Alto, 10 de Julio de 2020

Señor:
Ing. David Carlos Mamani Quispe
DIRECTOR DE LA CARRERA INGENIERÍA DE SISTEMAS
Presente. -

Ref.- Aval de Conformidad

Distinguido Ingeniero:

Mediante la presente tengo a bien comunicarle mi conformidad de la tesis de grado **"MODELO DE CERTIFICACIÓN DE CONTRATOS INTELIGENTES APLICANDO LA TECNOLOGIA BLOCKCHAIN" CASO: (CERTIFICACIONES CITES)**, que propone el postulante Univ. Raquel Apaza Alberto, con cedula de identidad 9116041 L.P. para su defensa publica, evaluación correspondiente a la materia Taller de Licenciatura II, de acuerdo a reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente.

.....

M.Sc. *Ing. Enrique Flores Baltazar*
TUTOR METODOLÓGICO TALLER II

AVAL DE CONFORMIDAD

El Alto, 10 de Julio de 2020

Señor:
Ing. Enrique Flores Baltazar
TUTOR METODOLÓGICO TALLER II
Presente. -

Ref.- Aval de Conformidad

Distinguido Ingeniero:

Mediante la presente tengo a bien comunicarle mi conformidad de la tesis de grado **"MODELO DE CERTIFICACIÓN DE CONTRATOS INTELIGENTES APLICANDO LA TECNOLOGIA BLOCKCHAIN" CASO: (CERTIFICACIONES CITES)**, que propone el postulante Univ. Raquel Apaza Alberto, con cedula de identidad 9116041 L.P. para su defensa publica, evaluación correspondiente a la materia Taller de Licenciatura II, de acuerdo a reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente.

.....

Ing. Ramiro Kantuta Limachi
TUTOR ESPECIALISTA

El Alto, 10 de Julio de 2020

Señor:
Ing. Enrique Flores Baltazar
TUTOR METODOLÓGICO TALLER II
Presente. -

Ref.- AVAL DE CONFORMIDAD

Distinguido Ingeniero:

Mediante la presente tengo a bien comunicarle mi conformidad de la tesis de grado **"MODELO DE CERTIFICACIÓN DE CONTRATOS INTELIGENTES APLICANDO LA TECNOLOGIA BLOCKCHAIN" CASO: (CERTIFICACIONES CITES)**, que propone la postulante Raquel Apaza Alberto, con cedula de identidad 9116041 L.P., para su defensa publica, evaluación correspondiente a la materia Taller de Licenciatura II, de acuerdo a reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente.

Ing. Elías Carlos Hidalgo Mamani
TUTOR REVISOR