Modelo arquitectónico para soluciones basadas en Blockchain

Juan David Muñoz Garzón
Astrid Carolina Ordoñez Guerrero

Director: PhD. Julio Ariel Hurtado Alegría

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Sistemas
Grupo de Investigación y Desarrollo en Ingeniería de Software (IDIS)
Línea de Investigación en Ingeniería de Software
Popayán, diciembre de 2021

Modelo arquitectónico para soluciones basadas en Blockchain

Monografía de Trabajo de Grado para optar al título de Ingeniero de sistemas

Juan David Muñoz Garzón
Astrid Carolina Ordoñez Guerrero

Director: PhD. Julio Ariel Hurtado Alegría

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Sistemas
Grupo de Investigación y Desarrollo en Ingeniería de Software (IDIS)
Línea de Investigación en Ingeniería de Software
Popayán, diciembre de 2021

NOTA DE ACEPTACIÓN

_____

_____

_____

_____

PRESIDENTE DEL JURADO

_____

JURADO

Popayán, 2021

Tabla de contenido

Índice de figuras

Índice de tablas

1. INTRODUCCIÓN

1.1 PLANTEAMIENTO DEL PROBLEMA

En los últimos 10 años se ha estudiado la forma de realizar transacciones o intercambios de una forma más rápida y segura a través del concepto de "BlockChain", una tecnología software de arquitectura descentralizada (bases de datos distribuidas) basada en transacciones, que tiene como objetivo mejorar la confiabilidad de la información y resolver el problema de realizar transacciones sin intermediarios, es decir, permitir que dos personas interactúen directamente entre sus sistemas, sin la necesidad de una tercera parte [1]. De acuerdo con F. Casino *et al.* [2], la tecnología Blockchain brinda grandes oportunidades de negocio más allá de las criptomonedas, como es el caso del manejo confiable de la información en sectores tales como salud, educación, sector energético, seguridad entre otros [2].

Ahora bien, la viabilidad de la implementación de esta tecnología en diferentes áreas de negocio no es clara, debido a que aún requiere resolver muchos aspectos relacionados con la privacidad, adaptabilidad, seguridad, trazabilidad, escalabilidad, transparencia, integridad, e interoperabilidad y sin dejar de lado las deficiencias o limitaciones que puede tener según el área de implementación [3][1][4]. Por otra parte, A. S. Bruyn *et al.* [1] indican que muchos de estos aspectos no sólo se relacionan con las estructuras y algoritmos, sino por la forma en que un sistema de Blockchain debe estructurarse, es decir, su arquitectura software [2][4]. Por lo anterior, se requiere de conocimiento y práctica para entender aspectos de la ingeniería de software y así construir y evaluar soluciones que adopten el paradigma Blockchain, particularmente aspectos arquitectónicos [5] relacionados con sus atributos de calidad, preocupaciones, tácticas y patrones de arquitectura [6].

Así mismo los autores X. Xu *et al.* [3] empezaron a estudiar el Blockchain como un nuevo tipo de conector arquitectónico de software que proporciona una infraestructura compartida para almacenar datos y ejecutar programas (conocidos como contratos inteligentes) [3], dicho término hace referencia a contratos que se ejecutan y se hacen cumplir a sí mismos de manera automática y autónoma, cuando se dan las condiciones previamente programadas. Sin embargo, debido a las propiedades y limitaciones de Blockchain, la tecnología no se ajusta en todos los casos, algunas de las restricciones detectadas son: i) Blockchain ha limitado la capacidad de almacenamiento, ya que contiene una historia completa de todas las transacciones a través de todos los participantes de la red Blockchain. ii) las aplicaciones Blockchain podrían tener datos sensibles, pero la información sobre Blockchain está diseñada para ser accesible a todos los participantes. iii) todos los contratos inteligentes pueden ser llamados por todos los participantes Blockchain por defecto, una función permiso, podría ser desencadenada por usuarios no autorizados accidentalmente, que se convierte en una vulnerabilidad de aplicaciones basadas en Blockchain [6].

El problema radica en cómo relacionar atributos de calidad, tácticas y patrones arquitecturales durante el diseño y mantenimiento de una arquitectura, requerida para una aplicación basada en Blockchain al momento de llevarlo a la práctica, debido a que la tecnología aún no ha sido estudiada sistemáticamente, y hay poca comprensión acerca del impacto de la adopción del Blockchain en una arquitectura [7], porque si bien hay varios enfoques que estudian la relación entre las decisiones arquitectónicas y los atributos de calidad en los sistemas centralizados, la investigación [7] se encuentra en una etapa temprana para los sistemas descentralizados [5]. Debido a eso, en este trabajo de grado buscamos resolver la siguiente pregunta de investigación

¿Qué aspectos arquitecturales, en términos de cualidades, incumbencias, tácticas y patrones arquitectónicos que soportan la tecnología Blockchain deben ser articulados al momento de evaluar y aplicar este tipo de tecnología a una solución software para soportar algún proceso de negocio basado en transacciones?

1.2 OBJETIVOS

1.2.1 Objetivo General

Diseñar un modelo arquitectónico general y reutilizable para soluciones de software basadas en tecnología de Blockchain (MAB).

1.2.2 Objetivos Específicos

- Identificar y caracterizar desde la literatura los aspectos[1] arquitectónicos relevantes con el Blockchain y sus relaciones.
- Especificar MAB a nivel de diseño arquitectónico a través de un conjunto de vistas y haciendo uso de los aspectos arquitectónicos identificados y sus relaciones.
- Evaluar la idoneidad de MAB mediante un estudio de caso en el cual es desarrollada una solución basada en Blockchain.

1.3 METODOLOGÍA Y DESCRIPCIÓN DEL TRABAJO

Para alcanzar los objetivos de este trabajo de investigación se tuvo en cuenta la estructura de algunos trabajos tales como: el proceso definido por J. Hurtado [8] denominado “Método científico en ingeniería de software-MCIS” el cual está compuesto por 3 fases (Exploración, formulación y ejecución) cuyos principales hitos se componen de la comprensión del problema, formulación y ejecución, los cuales consideramos son la base fundamental para el desarrollo de un trabajo de investigación. además, este proceso está apoyado en este proyecto por otras metodologías y directrices tales como: las directrices presentadas por Petersen et al.[9], la metodología enfocada en el desarrollo de revisiones sistemáticas de Kitchenham et al.[10] y para la recuperación y evaluación de la arquitectura es utilizado el método de investigación de estudio de caso propuesto por Runeson and Host [11].

Nuestro trabajo de investigación se compone de varios ítems, inicialmente una descripción general de temas de importancia para la comprensión y guía del proyecto, cuyos temas principales se basan en el entendimiento de la tecnología Blockchain y en la arquitectura de software. Partimos del conocimiento comprendiéndolo como un todo y es por ello que realizamos un mapeo sistemático enfocado en la arquitectura Blockchain dividida en varios momentos y dirigida mediante la directriz y metodología mencionada en el párrafo anterior, nuestro mapeo sistemático se caracteriza por la comprensión de hallazgos y unificación de los mismos obteniendo datos importantes de la arquitectura de la tecnología y permitiéndonos consolidar todo ese conocimiento en un solo proyecto. Además, fue la base principal que nos dio pie para proponer un modelo arquitectónico y desarrollar dos casos de estudio. El primero parte de la recuperación de la arquitectura de una plataforma Blockchain utilizando métodos y herramientas

de evaluación, la unificación de estos métodos y herramientas para la construcción del caso nos lleva a realizar la actividad en varios momentos que son: contextualización, presentación, identificación y priorización de los escenarios, finalmente recuperación visual, evaluación y recuperación del rationale; posteriormente se realiza el análisis de la arquitectura recuperada y se describen sus resultados. El segundo consistió en la evaluación de un modelo arquitectónico propuesto, cuyo objetivo principal seria obtener una retroalimentación de la aplicación del modelo en un proyecto para la gestión de salud ver su viabilidad y descripción de resultados.

La monografía está compuesta por 6 capítulos. En el capítulo 2: Se presenta el marco conceptual y estado del arte del proyecto enfocado en Blockchain y la arquitectura de software, el cual es el primer paso para dar solución al primer objetivo específico que es: "Identificar y caracterizar desde la literatura los aspectos[1] arquitectónicos relevantes con el Blockchain y sus relaciones ".

El capítulo 3: es un complemento al capítulo 2 ya que ampliamos el espectro de información mediante la realización de un mapeo sistemático descrito previamente.

En el capítulo 4: Se incluye un modelo arquitectónico, compuesto por patrones, atributos de calidad y tácticas, que representa una base conceptual para el diseño de aplicaciones Blockchain.

En el capítulo 5: Se realiza la evaluación de la arquitectura de una plataforma Blockchain (Hyperledger) aplicando el método de evaluación ATAM Y para la recuperación de la arquitectura el método EV-AR. En este capítulo se incluye el estudio exploratorio y holístico que permitieron la construcción empírica del mismo. Además, se incluye un caso de estudio exploratorio y holístico para la recopilación de información y descripción de resultados de la aplicación del modelo arquitectónico presentado en el capítulo 4.

Finalmente, en el capítulo 6 se presentan las conclusiones, limitaciones y trabajos futuros de este proyecto de investigación

2. MARCO CONCEPTUAL Y ESTADO DEL ARTE

2.1 ARQUITECTURA DE SOFTWARE

2.1.1 Definiciones clave

La arquitectura es un conjunto de componentes (patrones, abstracciones coherentes, atributos de calidad etc.) los cuales permiten tener un marco de referencia de un software para poder llevar a cabo su desarrollo, dichos componentes varían de un software a otro según los requisitos más relevantes y restricciones que se hayan obtenido dependiendo del caso, todo esto se efectúa de manera abstracta para definir las prácticas a aplicar y determinar la viabilidad de diferentes propuestas candidatas a ser implementadas. Una de las principales definiciones está dada por L. Bass *et al.* [5] de referencia para este trabajo es: “La arquitectura de software es la estructura o estructuras del sistema, incluyendo: sus componentes software, las propiedades visibles de dichos componentes y las relaciones entre ellos”.

2.1.2 Decisiones arquitecturales: Atributos, Tácticas, Patrones Arquitectónicos e Incumbencias.

2.1.2.1 Cualidades (Atributos de calidad).

Los atributos de calidad especifican la calidad de las respuestas del sistema [12], puede interpretarse como el grado en que dichos atributos satisface los requisitos de sus usuarios, se definen en los requisitos no funcionales siendo entre otros: confiabilidad, facilidad de uso, rendimiento, capacidad de mantenimiento [5][13]. Para verificar el cumplimiento de los atributos de calidad se espera que “dado un estado del sistema y una entrada específica, la salida debe estar dentro de los límites especificados” L. Bass *et al.* [5].

2.1.2.2 Tácticas.

Las tácticas son técnicas usadas para tomar decisiones de diseño generales que facilitan alcanzar los atributos de calidad esperados e interactúan positiva o negativamente con los patrones arquitectónicos; normalmente las tácticas se implementan a través de los patrones arquitectónicos [13]. Las tácticas se pueden categorizar en: tácticas en tiempo de diseño, por ejemplo, las usadas para la reutilización de código o mantenibilidad; también están las tácticas en tiempo de ejecución, actúan directamente con un atributo de calidad por ejemplo las usadas para establecer la seguridad o confiabilidad del sistema, estas dos clases de tácticas es fundamental seleccionarlas durante el diseño de la arquitectura de un sistema [13]. La implementación de estas puede resultar tan sencillo como utilizar un patrón totalmente compatible, así como podría llegar a ser una tarea ardua que necesite de adaptaciones al patrón o de la reestructuración de las estructuras ya definidas.

2.1.2.3 Patrones de arquitectura.

Los patrones arquitectónicos son soluciones generales a una arquitectura de software, relacionando los componentes, de tal forma que cumplan con las exigencias de los atributos de calidad. El éxito de un patrón arquitectónico depende en gran parte de la manera en cómo éste se implemente. Diferentes arquitecturas de software con características similares pueden utilizar un mismo patrón [14].

2.2 METODOS DE ARQUITECTURA

El desarrollo de arquitectura de software cuenta con varias etapas como lo son: análisis, diseño, implementación y a su vez también la evaluación y recuperación de la arquitectura, en cada una de dichas etapas se encuentran diversos métodos, cada uno de ellos basados en las diferentes concepciones requeridas para identificar las actividades y procesos que nos permitan cumplir con las especificaciones o requerimientos iniciales para generar una arquitectura.

A su vez los métodos ofrecen diferentes técnicas que nos ayudan a facilitar el proceso, la técnica seleccionada debe contener actividades que se complementen y sean lo suficientemente sólidas para ayudarnos a la comprensión y hallazgo de los atributos de calidad y tácticas que contribuyen a la arquitectura, lo cual es fundamental a la hora de realizar la evaluación de la arquitectura [15].

2.3 EVALUACIÓN DE ARQUITECTURA

Realizar una evaluación de la arquitectura es la manera más económica de evitar desastres, esta ayuda a encontrar debilidades en una etapa temprana de la construcción de un software, donde aún son factibles los cambios que pueden afectar de manera drástica el desarrollo. La evaluación de una arquitectura no produce resultados cuantitativos, pero sí proporciona mejoras y pautas que nos ayudan a prever los problemas que se puedan presentar a lo largo del desarrollo obteniendo el máximo beneficio, algunas de las cualidades que se tienen en cuenta al realizar una evaluación son: desempeño, escalabilidad, seguridad y modificabilidad. Como resultado se obtiene una lista priorizada de los atributos de calidad requeridos para la arquitectura que está siendo evaluada, riesgos y beneficios. Existen varias técnicas de evaluación de arquitecturas para llevar a cabo un desarrollo tales como SAMM, ATAM, ARID etc. En los estudios de caso de este trabajo se va a utilizar el método de evaluación ATAM (Architecture Tradeoff Analysis Method) [14] el cual obtiene su nombre no solo porque nos dice cuando una arquitectura particular satisface las metas de calidad, sino que también provee ideas de cómo esas metas de calidad interactúan entre ellas y como realizan concesiones mutuas (tradeoffs).

2.4 RECUPERACIÓN DE ARQUITECTURAS

La recuperación de arquitecturas de software se basa principalmente en la ingeniería inversa la cual analiza un sistema para identificar sus componentes, interrelaciones y así brindar un bosquejo representativo en un nivel de abstracción superior, esto con el fin de extraer artefactos de diseño que en lo posible sean independientes de la implementación del sistema [16]. Con la recuperación de arquitecturas se busca realizar un diseño el cual permite comprender que hace

un sistema, como lo hace, para que lo hace, entre otras necesidades [16]. En esta tesis se aplicará esta técnica para el caso de estudio donde se recuperará la arquitectura de una plataforma Blockchain: Hyperledger.

2.5 DOCUMENTACIÓN DE ARQUITECTURAS

Documentar arquitecturas de manera eficaz es una parte de gran relevancia en la ingeniería de software; si la documentación no se entiende o se malinterpreta puede causar serios problemas en el cumplimiento de los objetivos a la hora de implementar el sistema [17]. Un enfoque muy conocido y bastante practico es el concepto de vistas arquitecturales [18] debido a que refleja el ciclo de vida del desarrollo de software abstraído de los de preocupaciones, actividades y modelos de arquitectura (Reconstructing Architectural Views from Legacy Systems). Este trabajo documenta bajo el concepto de vistas una arquitectura recuperada de una plataforma Blockchain [17].

2.6 BLOCKCHAIN

2.6.1 Orígenes

Blockchain, en la práctica nace con la iniciativa de bitcoin en el año 2013, sin embargo, Satoshi Nakamoto ya conceptualiza dicha tecnología en el 2008 y los mecanismos de trabajo del Blockchain, están basados en llaves públicas y privadas que se derivan de un documento de Merkle de 1980. Gran parte de la criptología y técnicas indican que Blockchain tiene sus orígenes en los años 90's, por lo que se dice que su invención es de esa fecha. Dicha tecnología ha tenido gran acogida entre las personas, cada vez es más evidente que se incrementa el desarrollo de aplicaciones basadas en Blockchain para beneficiarse de todas sus cualidades, por lo cual se ha manifestado que va a ser cada vez más importante en el futuro [1].

2.6.2 Funcionamiento

Blockchain, como su nombre lo indica es una cadena de bloques (nodos enlazados) de forma distribuida; ya que al realizar una operación esta será soportada por diferentes ordenadores o máquinas servidoras, siendo estos los nodos de la cadena para su ejecución. Posteriormente se crea un enlace entre ellos lo cual permite realizar transacciones de forma segura sin la necesidad de terceros, manteniendo el anonimato entre los usuarios. En otras palabras, Blockchain puede servir como un libro de contabilidad abierta y distribuida que permite grabar las transacciones entre dos partes (*Transacción centralizada - BlockChain Figura 2.1*) [4][19], *además, se* caracteriza por elementos fundamentales tales como la redundancia de datos, comprobación de requisitos de transacción, registro de las transacciones ordenados secuencialmente en bloques, entre otros [2]. Según F. Casino *et al.*[2], la tecnología Blockchain está versionada de la siguiente manera:

- Blockchain 1.0 que incluye aplicaciones que permiten transacciones de criptomonedas digitales.
- Blockchain 2.0, que incluye SC(smart contracts) y un conjunto de aplicaciones que se extienden más allá de las transacciones en criptomonedas.
- Blockchain 3.0, que incluye aplicaciones en áreas como gobierno, salud, ciencia e IoT.

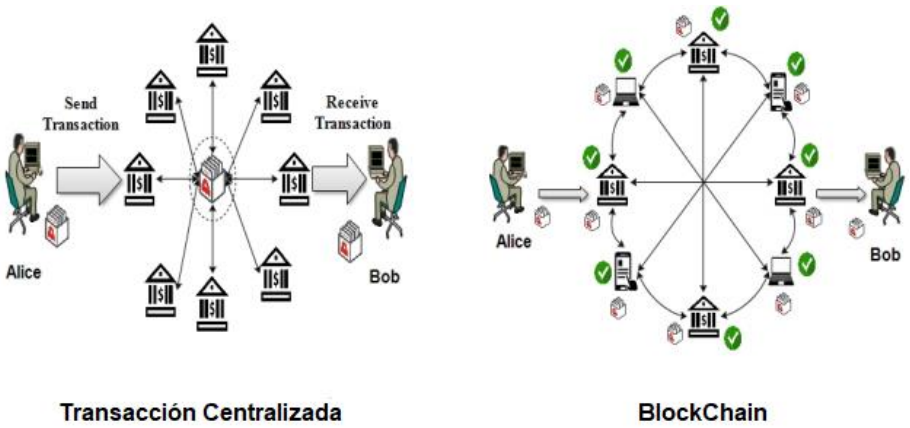


Figura 1: ***Transacción centralizada - BlockChain.***
Tomado de **S. Aggarwal, R. Chaudhary et al [20]**

2.6.3 Trabajos relacionados y Aspectos Arquitecturales del Blockchain

De acuerdo con H. O. W. The *et al.*[21], las implementaciones modernas de Blockchain tienen que adaptarse a algunos desafíos técnicos y limitaciones requeridas para la tecnología, por ejemplo el tamaño del bloque y ancho de banda; y cubrir cualidades esperadas del Blockchain tales como, seguridad, privacidad, desempeño, la facilidad de uso, la integridad de los datos y la escalabilidad. La adopción de la tecnología Blockchain en organizaciones a partir de la escogencia de una arquitectura flexible y correcta tiene grandes beneficios como lo son: incorporar seguridad de datos y funciones de confidencialidad; además, el uso de herramientas y marcos ayudan a conducir a la coexistencia, integración e interoperabilidad, los cuales empujan a las organizaciones a integrarse con Blockchain. En un estudio realizado por Viriyasitavat y D. Hoonsopon [22], sugieren una arquitectura de procesos de negocio en la era de Blockchain, la cual proporciona persistencia, validez, capacidad de auditoría, evita los intermediarios y proporciona flexibilidad. Ellos utilizan el protocolo PBFT (consenso) porque garantiza seguridad, vida útil y cierto grado de tolerancia a fallas. La arquitectura se divide en dos capas: la capa de procesos de negocios administrada por tecnologías de administración de negocios, incluidos el sistema de administración de negocios, la selección y composición de servicios, los lenguajes de especificación de flujo de trabajo de servicio y control de cumplimiento para monitoreo en tiempo real, y la capa Blockchain de procesos de negocio donde los procesos de negocio se codifican en contratos inteligentes. La variable de nodos en este contrato inteligente indica nodos involucrados para realizar el consenso [23].

X. Xu et al. [24] proponen una taxonomía que captura las principales características arquitectónicas de diversas soluciones Blockchain, para ayudar con el diseño y evaluación de su impacto en las arquitecturas software. F. Wessling *et al.* [7] empiezan a estudiar algunas tácticas que deben ser consideradas en la tecnología Blockchain para lograr una buena negociación entre las decisiones arquitectónicas y los atributos de calidad, separados en dos aspectos; tácticas a nivel de diseño arquitectónico y tácticas de implementación. Finalmente Wessling et al. [25], en un nuevo trabajo, plantea un enfoque relacionado a los atributos y elementos arquitectónicos que deben incorporarse en tecnologías Blockchain, para ello proponen una taxonomía de propiedades Blockchain, un diagrama de flujo y un primer boceto para conseguir un proyecto arquitectónico híbrido identificando a los participantes, las relaciones de confianza y las interacciones. Los artículos mencionados en este apartado y cada uno de sus aportes son

pequeños avances para un gran tema por esclarecer como lo es la arquitectura Blockchain, de la cual aún hay mucho por explorar.

2.7 PLATAFORMAS BLOCKCHAIN

Dia a día en la industria se ha impulsado el desarrollo de nuevas plataformas Blockchain diseñadas para entornos privadas donde se les permite a los participantes autenticarse, a diferencia de las primeras plataformas Blockchain que fueron públicas y que permite a los participantes entrar y salir sin necesidad de una autenticación. La categorización de Blockchains como públicas o privadas es útil para identificar las principales características. En las Blockchain publicas cualquier nodo puede unirse y abandonar el sistema, por lo que la cadena de bloques es totalmente descentralizada, la mayoría de sistemas públicos de Blockchain emplean variantes de PoW para el consenso, este funciona bien en entornos públicos ya que protege de los ataques Sybil. Sin embargo, al no ser determinista y computacionalmente costoso no es adecuado para aplicaciones que deben manejar grandes volúmenes de transacciones de manera determinista [23]. Por otro lado, están las Blockchain privadas en la que la cadena de bloques impone una membresía estricta, como resultado cada nodo se autentica y su identidad es conocida por los otros nodos. Una propiedad clave de las cadenas de bloques privadas es que admiten contratos inteligentes, las cuales permiten expresar lógicas de transacciones altamente costosas [23].

Para realizar la selección de las plataformas a estudiar se tuvo en cuenta el estudio presentado en el artículo “Untangling Blockchain: A Data Processing View of Blockchain Systems”, en el cual describen BLOCKBENCH, que es un marco de referencia para evaluar cuantitativamente y comparar Blockchains privadas con contratos inteligentes de Turing. En dicho estudio se llevó a cabo una evaluación exhaustiva de tres cadenas de bloques principales que son: Ethereum, Hyperledger y Parity; ya que, se consideran los más maduros en términos de la base de código y base de usuarios. Además, tuvieron en cuenta 5 métricas importantes: Desempeño, que es el número de transacciones exitosas por segundo; Latencia, tiempo de respuesta por transacción; escalabilidad, que son los cambios en el desempeño, la latencia al aumentar la cantidad de nodos y la cantidad de cargas de trabajo concurrentes; tolerancia a fallas, cambios en el rendimiento y la latencia durante la falla del nodo; y seguridad, que es la relación entre el número total de bloques incluidos en la rama principal y el número total de bloques confirmados [23].

- ***Desempeño y latencia:*** *la prueba se realizó con 8 servidores y 8 clientes simultáneos durante un período de 5 minutos. en términos de rendimiento, Hyperledger supera a los otros dos en ambos puntos de referencia. La brecha entre Hyperledger y Ethereum se debe a la diferencia en los protocolos de consenso: uno se basa en PBFT mientras que el otro se basa en PoW. La brecha entre Parity e Hyperledger no se debe a protocolos de consenso, ya que se espera que el protocolo PoA de Parity sea más simple y más eficiente que PoW y PBFT* [23].
- ***Escalabilidad:*** *La tasa de solicitud del cliente (320 solicitudes por segundo para Hyperledger, 160 solicitudes por segundo para Ethereum y Parity) y aumentaron tanto la cantidad de clientes como la cantidad de servidores. Curiosamente, mientras que el rendimiento y la latencia de Ethereum se degradan casi linealmente más allá de 8 servidores, Hyperledger deja de funcionar más allá de 16 servidores. Los resultados hasta ahora indican que escalar tanto la cantidad de clientes como la cantidad de servidores degrada el rendimiento e incluso hace que Hyperledger falle. Para Ethereum, aunque está vinculado a la computación, todavía consume una cantidad modesta de recursos de red para propagar transacciones y bloques a otros nodos* [23].

- ***Tolerancia a Fallas y seguridad:*** *Para evaluar cuán resistentes son los sistemas ante fallas, ejecutaron los sistemas con 12 y 16 servidores, con 8 clientes durante más de 5 minutos, durante los cuales eliminaron 4 servidores en 250 segundos. Primero, Ethereum no se ve afectado por el cambio, lo que sugiere que los servidores fallidos no contribuyen significativamente al proceso de minería. En segundo lugar, el rendimiento de Parity tampoco se ve afectado. Esto se debe a que a cada nodo se le asigna el mismo intervalo de tiempo durante el cual puede generar bloques, por lo tanto, fallar 4 nodos en Parity significa que los 8 nodos restantes reciben segmentos de tiempo más grandes. Tercero, Hyperledger deja de generar bloques después de la falla en la red de 12 servidores, lo cual es como se esperaba porque la PBFT solo puede tolerar menos de 4 fallas en una red de 12 servidores. En la red de 16 servidores, Hyperledger todavía genera bloques, pero a una velocidad menor, que fueron causados por los servidores restantes que tuvieron que estabilizar la red después de las fallas sincronizando sus vistas. En el ataque realizado en el artículo a las tres plataformas, tanto Ethereum como Parity se bifurcan en 100 segundos y la vulnerabilidad aumenta a medida que pasa el tiempo mientras que Hyperledger, no presenta bifurcaciones, ya que su protocolo de consenso está demostrado que garantiza la seguridad* [23].

De acuerdo a los resultados arrojados se puede decir que: Hyperledger se desempeña consistentemente mejor que Ethereum y Parity en los puntos de referencia. Pero no puede escalar hasta más de 16 nodos; Ethereum y Parity son más resistentes a las fallas de nodos, pero son vulnerables a los ataques de seguridad que bifurcan la cadena de bloques; Los principales cuellos de botella en Hyperledger y Ethereum son los protocolos de consenso, pero para Parity el cuello de botella es causado por la firma de transacciones; Ethereum y Parity incurren en grandes gastos generales en términos de uso de memoria y disco. Su motor de ejecución también es menos eficiente que el de Hyperledger; El modelo de datos de Hyperledger es de bajo nivel, pero su flexibilidad permite una optimización personalizada para consultas analíticas. En conclusión, las plataformas Hyperledger Y Ethereum se desempeñan muy bien a nivel general. Para tomar una decisión de cual elegir se debe entrar en detalle a la necesidad del negocio para obtener especificaciones precisas y realizar una evaluación de cada plataforma.

Plataformas ——————— Características	**ETHEREUM**	**HIPERLEDGER**	**PARITY**
EJECUCIÓN INTELIGENTE DEL CONTRATO	Viene con su propia máquina virtual para ejecutar los códigos de byte EVM. EVM realiza un seguimiento de los recursos consumidos durante la ejecución del contrato, tanto en términos de CPU como de memoria, y se cargan a la cuenta del remitente de la transacción. EVM	Optando por la portabilidad utiliza contenedores DOCKERS para ejecutar sus contratos. Específicamente, un contrato se puede escribir en cualquier idioma, que luego se compila en código nativo y se empaqueta en una imagen de	Aunque Ethereum y Parity usan el mismo motor de ejecución es decir EVM. Parity está más optimizada, por lo tanto, es más eficiente en el cálculo y la memoria [23].

	también realiza un seguimiento de los cambios de estado intermedios y los revierte si no hay fondos suficientes para pagar la ejecución. Ethereum incurre en una sobrecarga de memoria grande [23].	Docker. Cuando se carga el contrato, cada nodo inicia un nuevo contenedor con esa imagen. Hyperledger es mucho más eficiente en términos de velocidad y uso de memoria [23].	
CONTRATOS INTELIGENTES	Ethereum se encuentra entre las primeras cadenas de bloques que ofrecen contratos inteligentes completos de Turing [23].	Contratos inteligentes completos de TURING [23]	Contratos inteligentes completos de TURING [23]
LENGUAJE DE CONTRATO DE CONSENSO	SOLIDITY, SERPENT, LLL [23].	Admite lenguajes de programación de alto nivel como JAVA Y GO. Sin embargo, sus interfaces de clave valor con Blockchain necesitan lógicas de aplicación adicionales para mapear estructuras de datos de alto nivel en tuplas de clave valor [23].	SOLIDITY, SERPENT, LLL [26].
API	API JSON-RPC que devuelven detalles de transacciones y saldos de cuentas en un bloque específico [26].	no tiene API para consultar estados históricos [26].	API JSON-RPC que devuelven detalles de transacciones y saldos de cuentas en un bloque específico [23].
MADUREZ	Es más maduro en términos de base de código, base de usuarios y comunidad de desarrolladores [23].	Base de código y base de usuario [23].	Base de código y base de usuario [23].
RENDIMIENTO	Incurren en grandes gastos generales en	Se desempeña consistentemente	incurren en grandes gastos generales en

	términos de uso de memoria y disco [23].	mejor que Ethereum y Parity [23].	términos de uso de memoria y disco. Parity procesa las transacciones a una tasa constante [23].
TOLERANCIA A FALLAS	Resistente a fallas de nodos [23].	Deja de funcionar después de la falla en la red de 12 servidores debido al protocolo PBFT, que solo puede tolerar menos de 4 fallas en una red de 12 servidores [23].	Resistente a fallas de nodos [23].
ESCALABILIDAD	El rendimiento y la latencia de Ethereum se degradan casi linealmente a partir de 8 servidores [23].	Tiene problemas de escalabilidad. Hyperledger deja de funcionar a partir de 16 servidores ya que los nodos no llegan a un consenso sobre ningún lote de transacciones. Es decir, tener más servidores significa más mensajes intercambiados y mayores gastos generales [23].	El rendimiento y la latencia de Parity se mantiene. Parity cambia el rendimiento por escalabilidad al mantener los estados en la memoria [23].
SEGURIDAD	vulnerables a los ataques de seguridad que bifurcan la cadena de bloques, lo que significa están altamente expuestos a doble gasto o ataques [23].	El protocolo de consenso garantiza la seguridad, sin embargo, Hyperledger tarda más en recuperarse de los ataques que Ethereum y Parity [23].	vulnerables a los ataques de seguridad que bifurcan la cadena de bloques, lo que significa están altamente expuestos a doble gasto o ataques [26].
OPEN SOURCES	SI [26] https://github.com/ethereum	SI [26]. https://github.com/hyperledger/fabric	SI [26]. https://github.com/paritytech/

SOPORTE Y DOCUMENTACIÓN	SI. https://ethereum.org/es/	SI https://www.hyperledger.org/	SI https://www.parity.io/
CONSENSO	POW: fuerza bruta para sacar el hash del bloque, es un protocolo hibrido que tiene como objetivo mejorar el rendimiento. es tolerante al fracaso bizantino, pero es de naturaleza probabilística: es posible que se agreguen dos bloques al mismo tiempo, creando una bifurcación en la cadena de bloques [23].	PBFT-based: es un protocolo trifásico. En la fase de *preparación previa*, un líder difunde un valor para que otros nodos lo confirmen. A continuación, en la *preparación* fase, los nodos transmiten los valores que están a punto de confirmar. Por último, se confirma el valor de fase comprometida cuando hay más de dos tercios de los nodos que están de acuerdo en la fase anterior. PBFT está vinculado a la comunicación, pero logra seguridad y vida en redes parcialmente sincrónicas [23]. el protocolo asegura que una vez que se agrega un bloque, es final y no puede ser reemplazado o modificado [23].	POA(Proof-of-Authority): utiliza POA, en el que algunos nodos predefinidos se consideran autoridades de confianza y pueden proponer los siguientes bloques. luego usa la programación round-robin para asignar a cada nodo de autoridad una ventana de tiempo durante la cual puede proponer bloques [23]. POA Se utilizan en Blockchains privadas donde mejoran el PBFT mediante la ejecución de consenso en redes más pequeñas llamadas federadas [27].

Tabla 1. Estudio Comparativo Plataformas Blockchain. Fuente propia

3. ARQUITECTURA BLOCKCHAIN: UN MAPEO SISTEMATICO DE LA LITERATURA

3.1 PLANIFICACIÓN DEL MAPEO SISTEMATICO

Debido a que en los estudios realizados sobre la tecnología Blockchain la muestran como tecnología emergente y considerando la novedad y dispersión sobre el conocimiento de la arquitectura Blockchain, surge la necesidad de realizar un mapeo sistemático que permita Identificar y caracterizar desde la literatura los aspectos arquitectónicos relevantes con el Blockchain y sus relaciones. Un mapeo sistemático de la literatura es un proceso que se realiza en varias etapas mediante el cual se logra obtener información necesaria de un tema en específico. Dicha información nos sirve para adquirir una visión global del tema de interés [10] y nos ayuda a responder a las preguntas de investigación que hemos planteado. Para el desarrollo de este mapeo sistemático se siguieron las directrices de mapeo-revisiones sistemáticas presentadas por Petersen et al. [9] y la metodología enfocada en el desarrollo de revisiones sistemáticas de Kitchenham et al. [10]. El resultado de este estudio tiene como fin presentar la información suficiente que nos genere los aportes necesarios para la realización de un modelo arquitectónico Blockchain (MAB), además sirve como punto de partida para la realización de trabajos futuros.

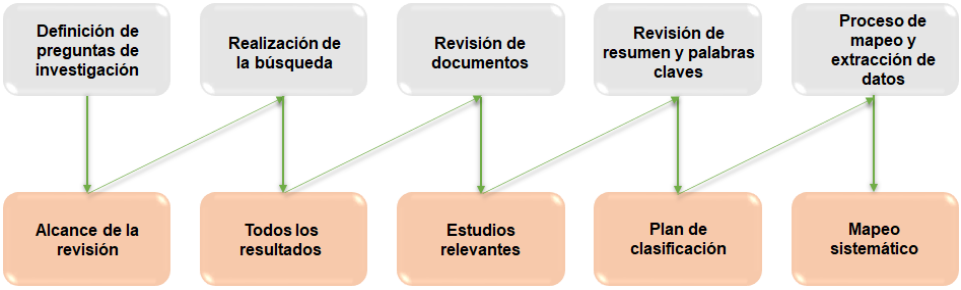


Figura 2. Proceso de Mapeo Sistemático propuesto por Petersen et al. [9]

A continuación, se desarrolla el mapeo sistemático teniendo en cuenta las 5 etapas presentadas en la figura 2: 1) Preguntas de investigación, 2) llevar a cabo la búsqueda de estudios primarios, 3) Revisión de documentos teniendo en cuenta los criterios de inclusión y exclusión, 4) Selección de estudios primarios, 5) Extracción de datos. Cabe destacar que la realización de cada una de estas etapas tiene un resultado que sirve como insumo para continuar con la siguiente etapa, cuyo resultado final del proceso es el mapeo sistemático.

3.1.1 Preguntas de investigación

Si bien el objetivo general de este trabajo se basa en diseñar un modelo arquitectónico que represente un conocimiento reutilizable para soluciones de software basadas en tecnología de Blockchain (MAB), las preguntas de investigación aquí planteadas buscan dar solución al primer objetivo específico, que es el paso inicial para generar el conocimiento suficiente, tener una visión integral del tema y poder dar seguimiento a los demás objetivos específicos.

Las preguntas de investigación que buscamos responder en este mapeo sistemático son:

Q1: ¿Cuál es la frecuencia de publicación de investigación sobre la arquitectura de la tecnología Blockchain?
Q2: ¿Qué publicaciones presentan como las plataformas responden a las cualidades buscadas por el enfoque Blockchain?
Q3: ¿Cuáles aspectos arquitecturales, tales como cualidades, incumbencias, tácticas y patrones arquitectónicos fueron considerados en el planteamiento, diseño y evaluación de arquitecturas basadas en Blockchain?
Q4: ¿Cómo están articulados estos aspectos arquitectónicos en las publicaciones que los incluye?
Q5: ¿Cuáles son los principales problemas que evidencia la tecnología Blockchain?

3.1.2 Fuente de datos y estrategia de búsqueda

La estrategia de búsqueda se elabora teniendo en cuenta una cadena de búsqueda automatizada, las bases de datos seleccionadas fueron: ACM Digital Library, IEEE Digital Library, ScienceDirect y Scopus. Dichas bases de datos se escogieron debido a que sus contenidos son cientifico-tecnicos y tienen amplia cobertura de publicaciones en el área de conocimiento ciencias de la computación. Conviene destacar que la cadena de búsqueda se empleó en el título, el *abstract y* las palabras claves de las bases de datos mencionadas anteriormente, como se puede observar en la Tabla 2, cabe destacar que en la base de datos ScienceDirect se redujo la cadena de búsqueda, debido a que esta acepta un máximo de 8 operadores booleanos.

Para la realización de las cadenas de búsqueda se tuvo como referencia los criterios de PICOC (Population, Intervention,Comparison, Outcomes y Context), los cuales son utilizados generalmente en la formulación de las preguntas de investigación, pero también sirven como guía para estructurar las cadenas de búsqueda ya que nos ayuda a tener en cuenta todos los temas que requerimos para nuestra investigación, además, nos permite asociarlos de manera ordenada y metódica [28]. Los 5 ítems que componen el termino PICOC hacen referencia a: la población; es donde se recopila la evidencia, la intervención; son las tecnologías; herramientas o procedimientos presentes en este estudio, la comparación; con la cual se compara la intervención, los resultados; los cuales no son simplemente estadísticos, sino también significativos desde el punto de vista práctico, contexto; es una visión ampliada de la población [29].

CONCEPTO	TERMINOS
Población	“Blockchain”
Intervención	“software architectur*”
Comparación	"architectural decision" "quality Attribute" "concern" "tactic" "pattern" "platform" "case Study" "solution"
Resultados	Relationships
Contexto.	“Architectural Development” “Reference Architecture”

Tabla 2. Tabla criterios PICOC para la Cadena de Búsqueda. Fuente propia

En la tabla que se presenta a continuación la cadena de búsqueda aplicada a las bases de datos científicas y el número de artículos obtenidos por cada base de datos.

Cadena de Búsqueda	Base de Datos Científica	# Artículos Encontrados
blockchain AND ((software AND architectur*) OR "Reference Architecture") AND ("architectural decision" OR "quality Attribute" OR "concern" OR "tactic" OR "pattern" OR "platform" OR "case Study" OR "solution")	ACM Digital Library	46
	IEEE Digital Library	95
	ScienceDirect	16
	Scopus	51
	Otras Fuentes	17
	TOTAL	225

Tabla 3. Cadena de Búsqueda. Fuente propia

Teniendo en cuenta que requeríamos obtener la mayor cantidad de información de los estudios publicados, decidimos realizar una búsqueda manual en Google Académico en donde se seleccionaron los artículos que consideramos tenían información relevante respecto a la arquitectura Blockchain, dichos estudios encontrados se anexaron a los estudios ya obtenidos de las bases de datos científicas escogidas y corresponden con el apartado otras fuentes que se encuentra en la tabla 3. Cadena de Búsqueda.

Finalmente, cabe destacar que en las búsquedas realizadas no se tuvieron en cuenta periodos de tiempo determinados, dicha búsqueda fue realizada en el periodo comprendido entre 01 de abril de 2020 y el 22 de abril del 2020, además se excluyeron los estudios que estaban asociados a otras áreas.

3.1.3 Criterios de selección de estudios primarios

Los criterios de selección de estudios primarios son esenciales para delimitar cuáles estudios son relevantes y cumplen con un umbral determinado de calidad. Así mismo los estudios que no cumplen con los criterios de selección y cumplen con uno o más criterios de exclusión, deben ser excluidos sin realizar una evaluación exhaustiva [30]. Los criterios de inclusión y exclusión que tuvimos en cuenta surgieron para responder a las preguntas de investigación planteadas en el numeral 3.1.1.

3.1.3.1 Criterios de inclusión

En la evaluación realizada a los estudios encontrados, se dice que es relevante si cumple con todos los criterios de inclusión mencionados a continuación:

- **CI1:** La investigación presenta el uso o análisis de patrones tácticas, cualidades o incumbencias (concerns) en el contexto de la tecnología Blockchain.
- **CI2:** Estudios que tengan en cuenta las plataformas, soluciones arquitectónicas de Blockchain o basadas en Blockchain.
- **CI3:** Artículos completos publicados en revistas, conferencias, congresos o talleres de prestigio con revisión por pares.

3.1.3.2 Criterios de exclusión

Si en la evaluación realizada a los estudios encontrados, no se cumplió con alguno de los criterios descritos a continuación, el estudio fue excluido de la investigación**.**

- **EC1:** Estudios no revisados por pares.
- **EC2:** Estudios que no permiten acceso al texto completo.
- **EC3:** Estudios del mismo grupo de investigadores que tratan el mismo tema desde diferentes perspectivas.
- **EC4:** Estudios que analizan los resultados de estudios primarios realizados por otros investigadores.

3.2 SELECCIÓN DE ESTUDIOS PRIMARIOS

Para la selección de los estudios primarios se realizó la revisión de los artículos aplicando 3 filtros y en cada uno de ellos se tuvieron en cuenta los criterios de inclusión y exclusión descritos anteriormente.

- **Primer filtro (Titulo y resumen):** Se realizo la revisión del título y resumen de las publicaciones encontradas por cada base de datos, incluyendo la búsqueda en otras fuentes y se eliminaron todas las publicaciones duplicadas.
- **Segundo filtro (Introducción y conclusiones):** Como segundo paso, aquellas publicaciones elegidas a partir del primer filtro se sometieron a la revisión de la introducción y conclusiones, seleccionando los que cumplieran con los criterios de inclusión y exclusión.
- **Tercer filtro (Texto Completo):** Finalmente, aquellas publicaciones seleccionadas fueron sometidas a un análisis y lectura del texto completo, de lo cual obtuvimos las publicaciones de interés para el tema a tratar.

TOTAL, ARTICULOS SELECCIONADOS	PRIMER FILTRO	SEGUNDO FILTRO	TERCER FILTRO
138	54	32	19

Tabla 4. Filtrado de artículos. Fuente propia

3.3 ESTRATEGIA DE EXTRACCIÓN DE DATOS

Según la metodología de Petersen et al.[9] se sugiere que los artículos deben ser explorados solo en aquellos casos en el que el abstract y titulo no son bien específicos. Pero en esta investigación se realiza la lectura de texto completo a los artículos que han pasado por los dos primeros filtros descritos anteriormente ya que el solo hecho de analizar el título y abstract no basta para dar respuesta a las preguntas de investigación planteadas**.**

Para extraer la información de interés se decide elaborar una plantilla de extracción de datos para cada uno de los estudios. En dicha plantilla se registra información principal tal como: información bibliográfica y el conocimiento relacionado con el problema de investigación: aspectos de la arquitectura Blockchain, sus plataformas y las relaciones. Es decir, se recopila toda la información que se encuentre asociada a las preguntas de investigación.

3.4 PROTOCOLO DE EJECUCIÓN

En la fase de ejecución del mapeo sistemático de la literatura damos paso al desarrollo de cada uno de los puntos establecidos anteriormente, los cuales fueron divididos en cinco etapas que son:

- **Etapa 1:** Se aplicó la cadena de búsqueda a cada una de las 4 bases de datos científicas seleccionadas, en las que se obtuvieron un total de 252 resultados, posteriormente se eliminaron los artículos duplicados haciendo uso de la herramienta Mendeley dando como resultado 138 artículos, a estos se les aplicó el primer filtro (revisión de título y abstract) y los criterios de inclusión y exclusión, de lo cual resultaron 54 artículos de investigación. Cabe destacar que en esta etapa se hizo una reasignación de los artículos encontrados, debido a que la base de datos Scopus contiene artículos publicados en otras bases de datos por lo cual se revisó cada uno de estos para identificar a que base de datos pertenecía para realizar la posterior reasignación, lo que ocasiona un cambio significativo en la cantidad de artículos presentes en cada una de las bases de datos ya que inicialmente más de la mitad de los artículos estaban asignados a la base de datos Scopus.
- **Etapa 2:** Se aplico el segundo filtro (introducción y conclusiones) a los artículos resultantes de la anterior etapa, teniendo como resultado 32 artículos de investigación, denominados “candidatos a estudios primarios”.
- **Etapa 3:** En esta etapa se realizó el mismo proceso de aplicar criterios de selección, pero en este caso empleando el tercer filtro (texto completo) teniendo como resultado un total de 19 artículos de investigación, a los cuales se les denomina “artículos primarios”.
- **Etapa 4:** Teniendo en cuenta que la finalidad de esta etapa es encontrar cuales artículos impactan más en nuestra área de investigación, se realizó la evaluación de calidad a cada uno de los artículos primarios.
- **Etapa 5:** Finalmente se realiza la extracción de datos y la caracterización de los estudios.

3.5 RESULTADOS Y ANALISIS DE RESULTADOS

Finalmente, se presenta los resultados y análisis de resultados obtenidos al realizar el mapeo sistemático de la literatura, cabe destacar que esto se logró mediante la extracción de información de los estudios primarios. El protocolo de extracción de dicha información se basó en realizar una tabla en la que se identificaron algunos campos tales como: título del artículo, fecha de publicación, aspectos arquitectónicos, aportes, relación de aspectos arquitectónicos, plataformas, características de las plataformas, resultados y conclusiones. Los cuales sirvieron como guía para tomar de cada artículo la información necesaria para dar respuesta a las preguntas de investigación planteadas

La siguiente gráfica permite representar un porcentaje basado en la cantidad de resultados obtenidos de cada una de las bases de datos seleccionadas en el numeral 3.1.2 fuentes de datos y estrategias de búsqueda. En esta gráfica se puede observar que más de la mitad de estudios encontrados corresponde a las bases de datos IEEE y ACM.

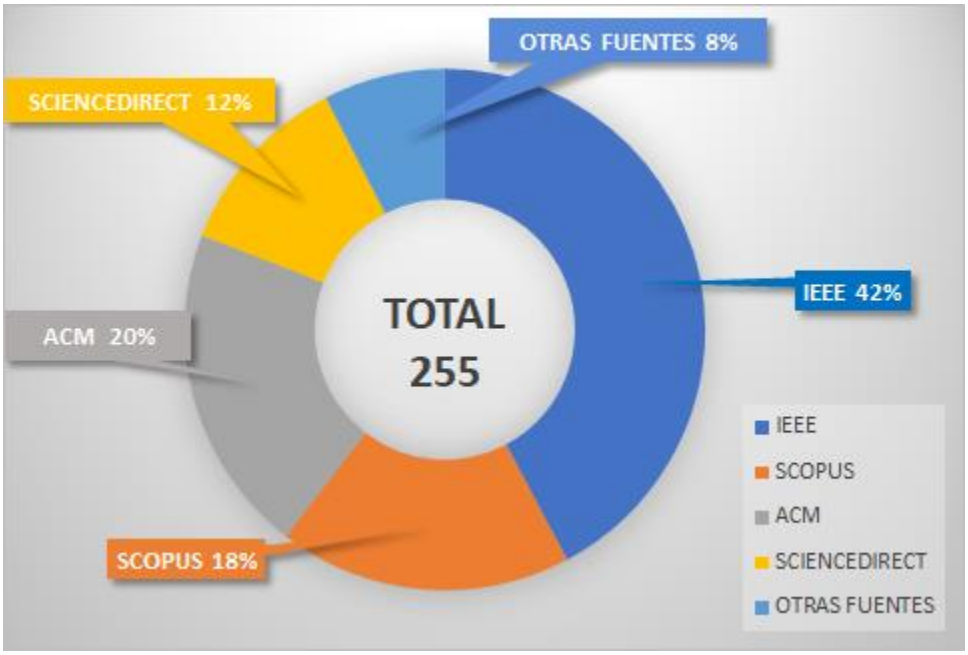


Figura 3. Resultados iniciales obtenidos de cada una de las bases de datos. Fuente propia

3.5.1 Q1: ¿Cuál es la frecuencia de publicación de investigación sobre la arquitectura de la tecnología Blockchain?

En la figura 4 se puede observar la distribución anual en las diferentes bases de datos de publicaciones realizadas que tienen que ver con la arquitectura Blockchain. Es preciso destacar que dicho gráfico hace referencia a los estudios primarios seleccionados en fases anteriores, como se puede observar los estudios corresponden al periodo de tiempo entre 2016 y 2019 con un mayor índice de publicación entre 2018 y 2019 correspondiente al 89.47% de las publicaciones. Esto quiere decir que el estudio de las arquitecturas de la tecnología Blockchain es muy reciente, que ha tenido gran acogida en los últimos años y es un tema en el que aún se puede explorar mucho.

Figura 4. distribución de investigaciones seleccionadas en el dominio arquitectura Blockchain. Fuente propia.

3.5.2 Q2: ¿Qué publicaciones presentan cómo las plataformas responden a los aspectos buscados por el enfoque Blockchain?

En el estudio realizado encontramos que 16 de las 19 publicaciones primarias describen algún aspecto arquitectónico de alguna de las plataformas Blockchain, entre las más comunes se encuentra Bitcoin, Ethereum y Hyperledger y solo algunos pocos estudios hacen referencia a plataformas como Parity y Ripple. La figura 5 representa el porcentaje correspondiente a la cantidad de artículos que describen la plataforma, cabe destacar que algunos artículos describen más de una plataforma.

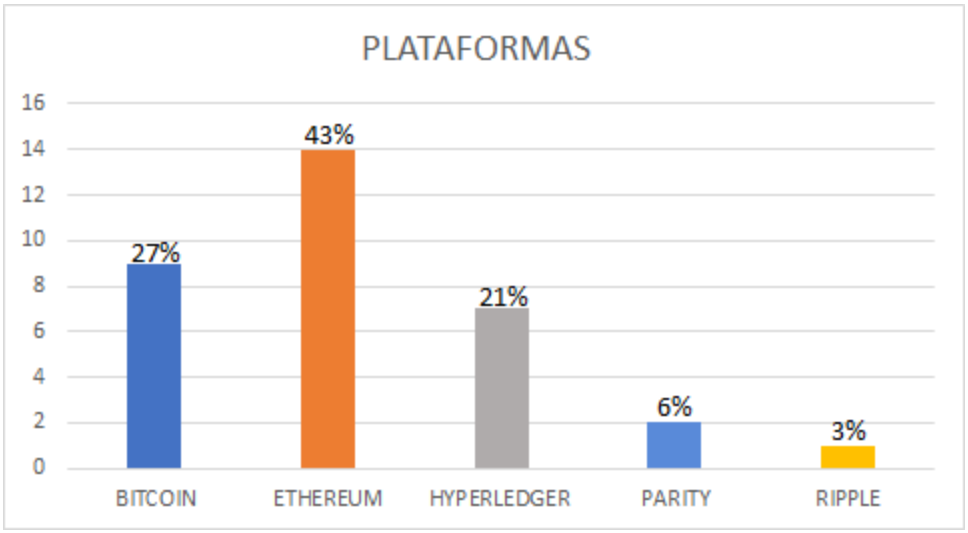


Figura 5. Distribución de plataformas Blockchain en estudios primarios. Fuente propia

Como mencionamos anteriormente las plataformas Bitcoin, Ethereum y Hyperledger son las más recurrentes en los artículos de Blockchain, por ende, nos centraremos en ellas para realizar la descripción de las cualidades de estas tres plataformas, además son las más populares debido a que enmarcan los cambios presentes en las dos generaciones de Blockchain. En la primera generación se encuentra Bitcoin que está vinculada a la aparición de Blockchain, la cual fue creada principalmente para almacenar y transferir monedas digitales con lo cual se proporcionó el primer libro público que permite almacenar transacciones financieras firmadas criptográficamente [31][24]. La segunda generación de Blockchains proporciona una infraestructura programable de propósito general con un libro público que registra los resultados computacionales, además, dieron lugar a los contratos inteligentes lo que permite poder configurar las transacciones [24] en esta generación se encuentran las plataformas Hyperledger y Ethereum. Por su parte podemos ver en la figura 5 que Ethereum es la plataforma con mayor porcentaje 43% esto es debido a algunos aspectos tales como: 1) Ethereum es la cadena de bloques más utilizada debido a su capacidad para implementar y ejecutar contratos inteligentes, 2) Ethereum fue pionera en emitir códigos para dichos contratos inteligentes [32] y 3) actualmente es la plataforma más destacada para construir Dapps [33]. Aunque Bitcoin está en segundo lugar en la figura 5 con un porcentaje del 27% este pertenece a la Blockchain de primera generación y solo admite contratos inteligentes básicos por ende podríamos decir que Hyperledger es la segunda plataforma Blockchain más usada y que cumple con las cualidades buscadas por el enfoque

Blockchain, pero para sustentar esta teoría vamos abordar algunos aspectos de las tres plataformas.

1. **Mecanismo de consenso:**
   El mecanismo de consenso es fundamental en la tecnología Blockchain dado que su función principal es determinar el consenso de todas las transacciones y el estado actual del sistema [24].Este mecanismo garantiza que las transacciones solo sean agregadas una vez si son válidas. En la actualidad existen tres modelos dominantes que son: prueba de trabajo, prueba de participación y tolerancia a fallas bizantinas. La prueba de participación en comparación con la prueba de trabajo es mejor debido a que se usa menos poder de cómputo en la minería, pero la latencia es más baja y las cadenas de bloques basados en tolerancia a fallas bizantinas ofrecen una garantía de consistencia más fuerte y una menor latencia, este protocolo normalmente solo se usa en Blockchains autorizados [31].

   - **Prueba de trabajo (POW):** Es utilizado por Bitcoin y Ethereum los cuales ofrecen garantía probabilística en términos de inmutabilidad de las transacciones registradas [24]. En este mecanismo los mineros resuelven problemas computacionales complejos con lo cual garantizan la validez de las nuevas transacciones.
   - **Tolerancia a fallas bizantinas (PBFT):** Este mecanismo garantiza el consenso a pesar del comportamiento arbitrario de los participantes, es decir agrega un nuevo bloque si más de 2/3 de todos los pares de validación presentan la misma respuesta [34]. Hyperledger utiliza PBFT.
   - **Prueba de participación:** Es un mecanismo alternativo a la POW que le otorgaría a los participantes derechos para minar en proporción a la cantidad de monedas que tengan dentro de la red Blockchain. Además, La cadena de bloques de prueba de participación brindan protección contra ataques maliciosos ya que ejecutar un ataque requiere que los atacantes cuenten con una gran cantidad de dinero y por lo general los participantes que cuentan con el dinero no atacan debido al concepto doble gasto, es decir que con el tiempo los ataques disminuyen el valor de la criptomoneda y el valor de su participación [34].

2. **Contrato inteligente:**
   Uno de los conceptos más importantes dentro de la tecnología Blockchain son los contratos inteligentes conocidos también como smart contract. Un contrato inteligente se puede definir como protocolos informáticos que facilitan, verifican y hacen cumplir digitalmente los contratos celebrados entre dos o más partes en Blockchain [31]. Ahora bien, Ethereum es la primera plataforma que admite contratos inteligentes avanzados de propósito general (Turing- completo) cuyo lenguaje principal utilizado en la cadena de bloques es solidez [35] y las transacciones son enviadas a Ethereum mediante Virtual Machine (EVM) para ejecutar métodos [31]. Los contratos inteligentes que se ejecutan en Hyperledger se llaman Chaincode que se puede escribir en cualquier lenguaje de programación y se ejecutan en contenedores (Dockers) [34], en cuanto a bitcoin es la primera criptomoneda que permite contratos inteligentes básicos, y estos se validan solo si se cumplen ciertas condiciones [36].

3. **Restricciones de Permiso:**
   Las restricciones de permiso se encargan de decidir si los mineros pueden hacer su trabajo (agregar nuevas transacciones validas a los bloques, agregar los bloques a la estructura de datos y propagar los bloques a la red Blokchain) sin permiso o si están restringidos y necesitan permiso para ser autorizados. Los sistemas totalmente

descentralizados incluyen cadenas de bloques sin permiso y permitidos, la plataforma Blockchain está basada en permisos, y de esa misma forma está basada Hyperledger [24] por lo que las transacciones son realizadas por usuarios predefinidos. Por otro lado, bitcoin y Ethereum no necesitan permisos, por lo cual están completamente abiertos, es decir los nuevos usuarios pueden unirse en cualquier momento a la red, validar las transacciones y extraer bloques [6].

4. **Permisos de acceso a datos:**
   Los permisos de acceso a datos se refieren a quien puede ver(leer) los datos de las transacciones. Las cadenas de bloques se pueden dividir en 3 tipos.

   - **Blockchain pública:** Con el uso de las cadenas de bloques públicas se obtiene una mayor transparencia de la información y capacidad de auditoría, pero se sacrifica la privacidad de la información [6]. Las Blockchains públicas han sido utilizadas para construir la mayoría de criptomonedas y en esta instancia se encuentra Bitcoin y Ethereum.
   - **Blockchain de consorcio:** El Blockchain de consorcio se usa para varias organizaciones, el derecho de leer la cadena de bloques puede estar público o restringido a los participantes de la cadena de bloques de la red Blockchain [6].
   - **Blockchain privada:** Las Blockchain privadas son utilizadas para una sola organización, en estas Blockchains solo los participantes que están predefinidos tienen acceso directo a los datos [37]. Este es el caso de Hyperledger.

5. **Escalabilidad / Desempeño:**
   Blockchain proporciona una escalabilidad limitada debido a que contiene un historial completo de todas las transacciones entre todos los participantes, por lo cual el tamaño de Blockchain crece cada vez más [24]. La escalabilidad de los sistemas Blockchain está compuesta por dos factores que son escalabilidad de nodos y escalabilidad de desempeño y entre ellos existe una compensación. Las cadenas de bloques públicas tienen una escalabilidad limitada, estas hacen la compensación hacia la escalabilidad de los nodos mediante su protocolo de consenso POW y solo pueden manejar en promedio de 3 a 20 transacciones por segundo [19], en esta se encuentra las plataformas Bitcoin y Ethereum. En cuanto a las plataformas que hacen uso del protocolo PBFT como es el caso de Hyperledger, realizan la compensación hacia la escalabilidad del rendimiento [34]. En términos comparativos Hyperledger tiene un alto nivel de escalabilidad.

6. **Regulación centralizada (Gobernanza):**
   Las plataformas Blockchain como cualquier otra plataforma o sistema requiere de mantenimiento y desarrollo por lo cual la gobernanza se refiere a las personas o persona que tiene la autoridad para la toma de decisiones a nivel del protocolo central. Hyperledger está a cargo de linux foundation, Ethereum está a cargo de Ethereum Developers y Bitcoin por community/miners [6].

7. **Anonimato:**
   El anonimato en una red Blockchain está estrechamente relacionado con las restricciones de permiso de la cadena de bloques, es decir, en una red Blockchain sin permiso los participantes son anónimos por que ocultan su identidad detrás de seudónimos que es su dirección de billetera publica, como en bitcoin y Ethereum, por el contrario, en una red con permisos como Hyperledger los participantes generalmente se conocen entre sí [31].

8. **Moneda nativa:**
   Algunas Blockchains cuenta con moneda nativa como lo es el caso de Bitcoin que utiliza su moneda “Bitcoin” y Ethereum que usa el “Eter”, pero en el caso de Hyperledger no cuenta con una moneda propia.

   En la siguiente tabla se resumen cada una de los aspectos mencionadas anteriormente de las plataformas Bitcoin, Ethereum y Hyperledger.

CARACTERISTICAS	BITCOIN	ETHEREUM	HYPERLEDGER
CONSENSO (SEGURIDAD)	POW	POW	PBFT
CONTRATO INTELIGENTE	Contratos inteligentes básicos	Turing completo, ejecución Ethereum Virtual Machine (EVM), lenguaje Solidez	Chaincode, ejecución contenedores (Dockers), cualquier lenguaje de programación
RESTRICCIONES DE PERMISO (SEGURIDAD)	Sin permiso	Sin permiso	Permisionado
RESTRICCIÓN DE ACCESO A DATOS (CONFIDENCIALIDAD	Público	Público o Privado	Privado
ESCALABILIDAD/ DESEMPEÑO	Alta escalabilidad de nodos - Baja escalabilidad de desempeño	Alta escalabilidad de nodos - Baja escalabilidad de desempeño	Alta escalabilidad de desempeño - Baja escalabilidad de nodos
REGULACIÓN CENTRALIZADA (NEGOCIO/GOBERNANZA)	community/miners	Ethereum Developers	linux foundation
ANONIMATO (CONFIDENCIALIDAD)	Si	Si	No
MONEDA NATIVA	Bitcoin	Éter	No

Tabla 5. Cualidades de las plataformas Blockchain. Fuente propia

5.3.3 Q3: ¿Cuáles aspectos arquitecturales, tales como cualidades, incumbencias, tácticas y patrones arquitectónicos fueron considerados en el planteamiento, diseño y evaluación de arquitecturas basadas en Blockchain?

Para dar respuesta a esta pregunta fue necesario realizar una revisión literaria minuciosa en la cual se tuvo como referencia el libro “software architecture in practice (3 edition)” [5] donde se mencionan aspectos importantes como atributos de calidad y sus tácticas, y el libro” quality attributes” [7] del cual tuvimos en cuenta los atributos de calidad y las preocupaciones(Concerns) asociadas. Esto con el fin de tener el conocimiento necesario que nos llevara a poder abstraer partes de la arquitectura Blockchain que no son tan explicitas en todos los contextos. Ahora bien en lo explicado anteriormente se puede notar que no

hablamos de patrones arquitectónicos y es debido a que en la revisión de la literatura que se realizó inicialmente pudimos evidenciar y destacar uno de los artículos que se denomina ”A pattern collection for Blockchain based application” [31] en el cual se encuentran bien definidos 15 patrones de diseño Blockchain subdivididos en 4 tipos, dicho artículo también se tomó como referencia para lograr encontrar en los demás artículos la utilización o descripción de estos patrones, evidenciamos que nuestra apreciación fue muy acertada debido a que en la mayoría de los artículos hubo al menos uno de ellos.

Una vez descrita la forma que utilizamos para abstraer los aspectos arquitectónicos de la literatura, podemos entrar en materia. Los atributos de calidad, patrones de diseño, tácticas y preocupaciones se encuentran ilustrados en las tablas que se presentan a continuación. Inicialmente tenemos una tabla en la cual relacionamos cada uno de los 19 artículos con una letra esto con el fin de asociar cada uno de los aspectos arquitectónicos encontrados con su respectivo artículo en las tablas siguientes. Además, es importante mencionar cinco artículos que se centran en aspectos específicos de la arquitectura, uno de ellos es el que mencionamos anteriormente de X. Xu et al.[31], el segundo y tercer articulo tratan tácticas propias de Blockchain [7][25], el tercer artículo es el de Medellin et al.[38] en el que realizan compensaciones de los atributos de calidad más importantes con base en las directrices que ha publicado la SEI(software engineering institute) y al mismo tiempo hacen énfasis en las compensaciones más frecuentes en Blockchain. El quinto y último artículo es el de Xu et al.[6] en el cual proponen considerar Blockchain como conector de software.

Como mencionamos anteriormente en la primera tabla encontramos la lista de los 19 artículos seleccionados en el mapeo sistemático. En las siguientes tablas hablaremos un poco de los 5 artículos relacionándolos con los aspectos arquitectónicos en los que se centran.

	ARTÍCULO
A	A Blockchain-Based Micro Economy Platform for Distributed Infrastructure Initiatives
B	A Comparison of Performance between Fully and Partially Decentralized Applications
C	A Blockchain-Based Collocation Storage Architecture for Data Security Process Platform of WSN
D	A Pattern Collection for Blockchain-based Applications
E	A Taxonomy of Blockchain-Based Systems for Architecture Design
F	A-Discussion-on-Blockchain-Software-Quality-Attribute-Design-and-Tradeoffs
G	Analysis of Architectural Variants for Auditable Blockchain-based Private Data Sharing
H	Applying Design Patterns in Smart Contracts
I	Blockchain Solution Reference Architecture (BSRA)
J	Blockchain-Enabled Smart Contracts Architecture, Applications, and Future Trends
K	Design Pattern as a Service for Blockchain Applications
L	Designing Blockchain-based Applications A Case Study for Imported Product Traceability
M	Engineering Software Architectures of Blockchain-Oriented Applications
N	Evaluating Suitability of Applying Blockchain
O	How Much Blockchain Do You Need Towards a Concept for Building Hybrid DApp Architectures

P	Poster Architecture Reconstruction and Evaluation of Blockchain Open Source Platform
Q	SmartInspect Solidity Smart Contract Inspector
R	The Blockchain as a Software Connector
S	Towards Blockchain Tactics Building Hybrid Decentralized Software Architectures

Tabla 6. Artículos. Fuente propia

ATRIBUTOS DE CALIDAD:

La definición de atributo de calidad que nos proporciona el libro de Bass et al.[5] es:" Un atributo de calidad (QA) es una propiedad medible o comprobable de un sistema que se utiliza para indicar qué tan bien el sistema satisface las necesidades de sus partes interesadas". Ahora bien, teniendo en cuenta dicha definición se encontraron los atributos de calidad ilustrados en la **Tabla 7. Atributos de calidad**, donde se puede observar que los más recurrentes son el desempeño, la escalabilidad y la seguridad en la arquitectura Blockchain. Estos tres atributos son los más recurrentes debido a que son los más afectados cuando se toman decisiones de diseño, es decir, la escalabilidad se ve afectada por el tamaño y frecuencia del bloque, la seguridad por el protocolo de consenso y el desempeño por la estructura de los datos [39]. Para profundizar más en el desempeño se sabe que el tamaño de Blockchain cada vez es mayor, ya que este hace una replicación completa de los datos por lo cual el almacenamiento de dichos datos es un desafío constante, pero a su vez esto hace que tengamos disponibilidad total de la información, para abordar esta problemática se ha introducido los términos on-chain y off-chain que son los datos dentro y fuera de la cadena de lo cual se profundizara más adelante. Por otro lado, Blockchain proporciona una plataforma confiable que permite el intercambio de información de manera segura [25] esto es posible debido a que, en lugar de confiar en un componente central, la confianza es depositada en un protocolo de igual a igual (peer to peer) basado en primitivas criptográficas [38] que a su vez garantiza la integridad de los datos. Los demás atributos como lo son interoperabilidad, modificabilidad testeabilidad usabilidad y extensibilidad podemos ver que no son tan recurrentes, pero fueron tenidos en cuenta debido a que el estudio realizado por Medellin et al.[38] hace compensaciones(trade-off) interesantes de ellos.

ATRIBUTOS DE CALIDAD	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**	**O**	**P**	**Q**	**R**	**S**	**total**
Desempeño	x	x	x	x	x	x	x		x	x		x		x		x			x	13
Escalabilidad	x	x	x	x	x		x		x	x	x			x				x		11
Seguridad	x		x	x	x	x	x			x	x	x	x			x	x	x	x	14
Disponibilidad					x	x	x		x			x								5
Confiabilidad		x	x	x					x			x							x	6
Integridad			x		x		x	x		x		x		x				x		8
Interoperabilidad				x	x	x		x	x											5
Modificabilidad				x		x		x								x				4
Testeabilidad						x														1
Usabilidad						x							x							2
Extensibilidad						x														1
Auditabilidad				x	x		x			x				x				x		6
TOTAL	3	3	5	7	7	8	6	3	5	5	2	5	2	4	0	3	1	4	3	

Tabla 7. Atributos de calidad. Fuente propia

Uno de los artículos importantes mencionados anteriormente es el presentado por Medellin et al. [38] el cual tiene como objetivo segmentar y realizar compensaciones de los atributos de calidad claves que un diseñador debe tener en cuenta cuando se desea hacer uso de la tecnología Blockchain y proporcionan una matriz que permite asociar las compensaciones entre atributos de calidad.

Cuando se desea diseñar una arquitectura debemos tener en cuenta las compensaciones que se dan entre los atributos de calidad ya que estos habitualmente entran en conflicto uno con el otro. Por ejemplo: interoperabilidad vs rendimiento, si diseñamos un sistema con un alto nivel de abstracción y cohesión de la interfaz, esto podría ocasionar que el sistema utilice más ciclos lo cual tendría un impacto en el rendimiento. Es por ello que se debe diseñar pensando en que las compensaciones entre atributos de calidad ofrezcan una solución que cumpla razonablemente con los requisitos establecidos por el sistema. Ahora bien, en el artículo se tuvieron en cuenta 7 atributos de calidad que son: disponibilidad, interoperabilidad, modificabilidad, rendimiento, seguridad, testeabilidad, usabilidad y extensibilidad siendo este último opcional. Estos atributos son claves para determinar la calidad de un sistema de software.

A continuación, citamos las compensaciones más importantes realizadas por Medellin et al.[38] para el diseño de la arquitectura Blockchain:

> *"1) Almacenamiento vs Computación: la cantidad de almacenamiento requerida por el bloque afectará los requisitos de computación; un bloque más grande requerirá más cálculos para cifrar, ya que hay más elementos para cifrar. La compensación de diseño en este caso es usabilidad frente a rendimiento.*
>
> *2) Anonimato vs Confianza: si las partes no se conocen, lo más probable es que necesiten verificar las identidades adecuadas mediante el uso de claves públicas / privadas. Este paso adicional requerirá un patrón de diseño diferente. La compensación de diseño en este caso es interoperabilidad frente a seguridad.*
>
> *3) Variaciones de incentivos: por definición, el patrón requiere validación distribuida antes de que se puedan agregar bloques. Los participantes suelen estar incentivados a realizar validaciones para llevar a cabo esta función. En las cadenas públicas, esto significa la asignación de valor a esos participantes lo que implica rastrear el valor en la red. De manera similar, en privado, los esquemas de incentivos varían, en la forma más simple podría significar que el participante debe validar los bloques anteriores antes de que se agregue el suyo. La compensación de diseño en este caso es extensibilidad vs usabilidad.*
>
> *4) Grado de distribución: en los patrones tradicionales de Nakamoto, los participantes guardan copias de la cadena de bloques. Sin embargo, existen diferentes patrones que pueden requerir que los contratos inteligentes residan fuera de línea, en la cadena en un nodo centralizado o de otro tipo. La compensación de diseño en este caso es interoperabilidad frente a extensibilidad.*
>
> *5) Escalabilidad frente a latencia: estos dos atributos están estrechamente relacionados con el rendimiento. En la mayoría de los casos, los sistemas están diseñados para cumplir con un cierto tiempo de devolución a los usuarios que se especifica en los requisitos. Cuando esos límites se superan en una escala mayor (más volúmenes), el procesamiento de bloques comienza a retrasarse, en algunos casos mucho más allá de los requisitos esperados. La compensación de diseño en este caso es Disponibilidad vs Rendimiento.*
>
> *6) Inmutabilidad frente a funcionalidad del proceso: esto ocurre principalmente en la implementación de contratos inteligentes. Uno de los atributos clave de la arquitectura es que es un registro permanente. La inclusión de contratos inteligentes puede violarlos (al tener datos fuera*

de línea o modificar bloques anteriores) o preservarlos (al reafirmar el contrato inteligente, los estados anteriores y el nuevo estado). En este caso, la compensación de diseño es seguridad frente a usabilidad.

7)Selección del algoritmo de consenso: esta compensación se relaciona con la selección del enfoque de consenso para la validación. En Nakamoto, por ejemplo, el PoW constituye tanto un enfoque de seguridad (51% de la computación) como la validación del bloque (cifrado / búsqueda del "nonce"), mientras que, en otros enfoques, esto podría variarse para proporcionar un nivel de precisión y protección. eso puede no ser tan intenso computacionalmente. La compensación de diseño en este caso es seguridad frente a rendimiento."

En la **Tabla 8. compensaciones** de atributos de calidad se muestra la relación de las compensaciones entre los atributos de calidad claves para el diseño de la arquitectura Blockchain, la forma de entenderla es la siguiente: en el caso de la compensación disponibilidad vs rendimiento estas se relacionan con el numero 5 el cual hace referencia a la descripción del numeral 5 de las compensaciones descritas anteriormente.

	DISPONIBILIDAD	INTEROPERABILIDAD	MODIFICABILIDAD	RENDIMIENTO	SEGURIDAD	TESTEABILIDAD	USABILIDAD	EXTENSIBILIDAD
DISPONIBILIDAD	X			5				
INTEROPERABILIDAD		X			2			4
MODIFICABILIDAD			X					
RENDIMIENTO	5			X	7		1	
SEGURIDAD		2		7	X		6	
TESTEABILIDAD						X		
USABILIDAD				1	6		X	3
EXTENSIBILIDAD		4					3	X

Tabla 8. Compensaciones. Fuente Propia

PATRONES DE DISEÑO:

Los patrones de diseño que contemplamos en esta sección son los encontrados en el artículo "A pattern collection for Blockchain based application" [31] que como explicamos anteriormente se encuentran agrupados en 4 tipos que son: patrones del mundo externo, los cuales describen diferentes formas para que Blockchain se comunique con el mundo externo(ejemplo: comunicación de datos con componentes dentro del sistema); patrones de

gestión de datos; para datos tanto dentro como fuera de Blockchain, patrones de seguridad; para aplicaciones basadas en Blockchain y patrones estructurales y contractuales; definen las dependencias entre los contratos inteligentes y su comportamiento. Como se sabe un patrón de diseño es la representación de las formas que nos permiten solucionar problemas recurrentes de diseño. Por ello existen múltiples patrones que se pueden aplicar de acuerdo al momento de diseño del sistema.

PATRONES DE DISEÑO	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**	**O**	**P**	**Q**	**R**	**S**	**total**
INTERACTION WITH EXTERNAL WORLD PATTERNS																				
Oracle				x	x					x		x		x				x		6
Reverse Oracle				x	x							x						x		4
Legal and Smart Contract Pair				x				x		x		x						x		5
DATA MANAGEMENT																				
Encrypting On-Chain Data	x		x	x	x	x	x	x		x	x	x		x				x		12
Tokenisation	x			x	x				x	x	x		x	x						8
Off-Chain Data Storage	x			x	x		x	x	x	x	x	x		x				x		11
State Channel	x			x	x				x	x								x		6
SECURITY																				
Multiple Authorization				x	x			x			x	x						x		6
Off-Chain secret enable dynamic authorization			x	x		x	x				x	x						x		7
X-Cofirmation				x	x													x		3
STRUCTURAL PATTERNS OF CONTRACT																				
Contract Registry				x						x	x	x						x		5
Data Contract				x				x				x								3
Embedded Permission				x	x						x	x								4
Factory Contract Store				x				x				x								3
Incentive Execution				x																1
TOTAL	4	0	2	15	9	2	3	7	3	7	7	12	1	4	0	0	0	6	0	XX

Tabla 9. Patrones de diseño Blockchain. Fuente propia

TÁCTICAS:

De acuerdo a la IEEE (Institute of Electrical and Electronics), una táctica es una decisión de diseño que influencia el control de la respuesta de un atributo de calidad, es decir un arquitecto debe escoger las tácticas adecuadas que le permitan alcanzar la calidad definida del sistema. Algunos autores como Bass et al.[5] han implementado y desarrollado una guía de tácticas por atributos de calidad con respecto al desarrollo de software, es por ello que en primera instancia tomamos como referencia este libro para encontrar algunas tácticas debido a que en ninguno de los artículos hablan de forma explícita de ellas. Haciendo uso de esta técnica pudimos encontrar algunas tácticas que son las que se muestran a continuación en la **Tabla 10 Tácticas** las cuales están separadas por atributos de calidad**.**

TACTICAS	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	tota
DESEMPEÑO																				
Limit event response	x				x													x		3
Maintain Multiple copies of computations	x			x	x		x		x	x	x	x		x				x	x	11
Increase resources					x	x												x		3
SEGURIDAD																				
Verify message delay	x		x	x	x		x	x			x	x		x			x	x		11
Encrypt Data			x	x	x	x	x	x	x	x		x		x				x		11
DISPONIBILIDAD																				
TimesTamp	x	x	x							x								x		5
Replication	x			x	x		x	x	x			x		x				x	x	10
TOTAL	5	1	3	4	6	2	4	3	3	3	2	4	0	4	0	0	1	7	2	

Tabla10. Tácticas. Fuente propia

Los articulos “¿How much Blockchain do you need? towards a concept for building hybrid Dapp architectures” [25] y “Towards Blockchain tactics building decentralized software architectures” [8]. presentan un enfoque para definir qué elementos de una arquitectura de sistemas existentes podrían beneficiarse al incluir la implementación de la tecnología Blockchain, para ello crean un borrador arquitectónico cuyos pasos consisten en identificar los participantes, sus relaciones de confianza e interacciones. Se adicionan estos estudios en este apartado ya que específicamente en el segundo articulo nos encontramos con que esos pasos definidos en el primer artículo están propuestos como tácticas de Blockchain y en este sentido definen dos niveles en las que consideran el diseño arquitectónico (identificar participantes, identificar relaciones de confianza e interacciones) y además un nivel de implementación en el que se definen a su vez 3 etapas que son: determinar escenarios de uso, simular costos operativos y seleccionar patrones de diseño. Sin embargo, nosotros los catalogamos como recomendaciones o pasos de construcción de aplicaciones Blockchain.

PREOCUPACIONES(CONCERNS).

En el campo de la ingeniería de software las preocupaciones se definen como: “los parámetros por los cuales los atributos de un sistema son juzgados, especificados y medidos” [6]. En cuanto a las preocupaciones de Blockchain solo logramos identificar algunas para los atributos de calidad rendimiento y seguridad ya que son los más recurrentes en los artículos.

PREOCUPACIONES (CONCERNS)	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	total
RENDIMIENTO																				
rendimiento	x	x	x	x																4
latencia		x	x	x	x	x	x			x		x				x		x		10
inmutabilidad	x			x	x			x	x		x	x		x				x		9
modos		x	x	x																3
SEGURIDAD																				
trazabilidad		x		x			x	x	x			x						x		7
integridad		x		x			x	x		x	x	x		x				x		9
No-repudio				x		x	x					x	x							5
confidencialidad				x				x			x	x		x						5
disponibilidad				x			x	x												3
TOTAL	2	5	3	9	2	2	5	5	2	2	3	6	1	3	0	1	0	4	0	

Tabla 11. Preocupaciones. Fuente propia

Para concluir esta pregunta es importante dar una breve descripción del artículo “The Blockchain as a Software Connector” de los autores X. Xu et al. [6], debido a que ellos consideran la cadena de bloques como un conector de software lo que les permite realizar consideraciones importantes respecto al rendimiento resultante y algunos atributos de calidad como lo son, por ejemplo: seguridad, privacidad, escalabilidad y sostenibilidad. Como se sabe un conector de software es un mecanismo de interacción para los componentes y en sistemas distribuidos nos permiten lograr las propiedades del sistema. La tecnología Blockchain como conector podría mejorar algunos aspectos como lo son la transparencia y la trazabilidad de la información, aunque también hay que tener en cuenta los contras, siendo uno de estos el aumento de la latencia en la comunicación cuando se aplica minería de datos. Cada nodo de Blockchain está dividido en dos capas, la capa de “Aplicación” y la de “Conector Blockchain” (**Figura 6: Descripción general de Blockchain como conector utilizando un estado**), una parte de la aplicación se implementa dentro del conector Blockchain en forma de contratos inteligentes. La parte de la aplicación que se implementa fuera de la capa del conector puede alojar datos y la lógica de la aplicación para interactuar con la cadena de bloques a través de transacciones. Para implementar la tecnología Blockchain como conector de software es necesario tener en cuenta una de las principales características arquitectónicas, donde se tiene que decidir qué funcionalidades se implementan dentro de la cadena y que otras deben mantenerse fuera de ésta.

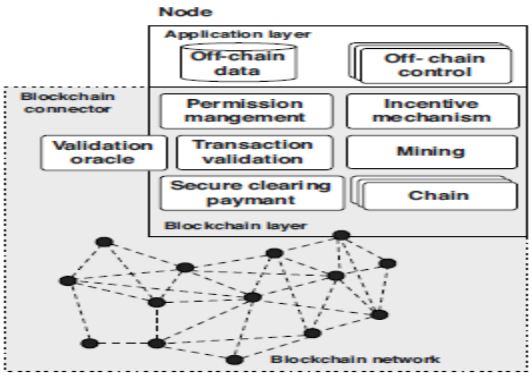


Figura 6. Descripción general de Blockchain como conector utilizando un estado. Tomado de X. Xu et al.[6]

En el estudio se tuvieron en cuenta los aspectos de un conector que son: la comunicación y coordinación a través de transacciones, oráculos de validación y contratos inteligentes, servicios de facilitación específicos, que incluyen gestión de permisos, pagos seguros basados en criptografía, validación de transacciones, extracción e incentivos.

5.3.4 Q4: ¿Cómo están articulados estos aspectos arquitectónicos en las publicaciones que los incluye?

La articulación de los aspectos arquitectónicos mencionados anteriormente se realizó de la siguiente manera: se tomó como base referencial los patrones de diseño mencionados en el artículo" A pattern collection for Blockchain based application" [31] en el cual se mencionan 15 patrones de diseño. Después de realizar el estudio respectivo a estos patrones encontramos que en los 18 artículos restantes se habla de al menos uno de estos patrones ya sea de forma explícita o implícita, debido a esto se tomó la decisión de realizar un catálogo de patrones en la que se evidencie la articulación de los aspectos arquitectónicos tomando como punto de partida dichos patrones de diseño. Inicialmente realizamos una breve descripción del patrón que se formara teniendo en cuenta principalmente la descripción del artículo que lo expone y apoyamos esa descripción con información encontrada en otros artículos, además tenemos en cuenta la plantilla de documentación de patrones que se muestra en el libro de Bass et al. [5] en la cual se establece que un patrón arquitectónico debe contener la relación entre los siguientes aspectos: un contexto, un problema y una solución, además, tuvimos en cuenta aspectos tales como consecuencias (pros, contra y compromisos entorno a los atributos de calidad), tácticas y usos conocidos. Cabe destacar que los patrones de diseño que tuvimos en cuenta para realizar dicha descripción y articulación son los que tienen una base de información sólida.

- **Patrón 1:** Encrypting On-Chain Data

    - **Contexto:** Los sistemas basados en Blockchain pueden tener datos o información que no permitan ser accedidos por todos los participantes del sistema, como, por ejemplo: en el caso de las historias clínicas, que solo el participante a quien corresponde la historia clínica puede acceder a ella.

    - **Problema:** Blockchain no cuenta con usuarios privilegiados en ninguna de sus tres Blockchains (públicos, privados y consorcio), es por ello que por ejemplo en el caso de los Blockchain públicos estos cuentan con transparencia y auditabilidad de la información [24] por lo cual esto se convierte en falta de privacidad de los datos ya que estos se encuentran expuestos a todos los participantes de la red.

    - **Solución:** Para lograr la privacidad de los datos e información que agrega los participantes a la Blockchain se debe hacer uso del cifrado de datos o hashes criptográficos antes de ingresar los datos a la Blockchain. Esto se logra cuando uno de los participantes crea una clave secreta y la distribuye fuera de la cadena, una vez distribuida esta clave quien la tenga podrá acceder y descifrar la información[36] [19]. Es decir, la clave secreta con la que se cifra la información es la misma que se utiliza para descifrarla, este es uno de los posibles métodos que se puede utilizar para cifrar los datos denominado criptografía simétrica, pero también se puede hacer uso de la criptografía asimétrica.

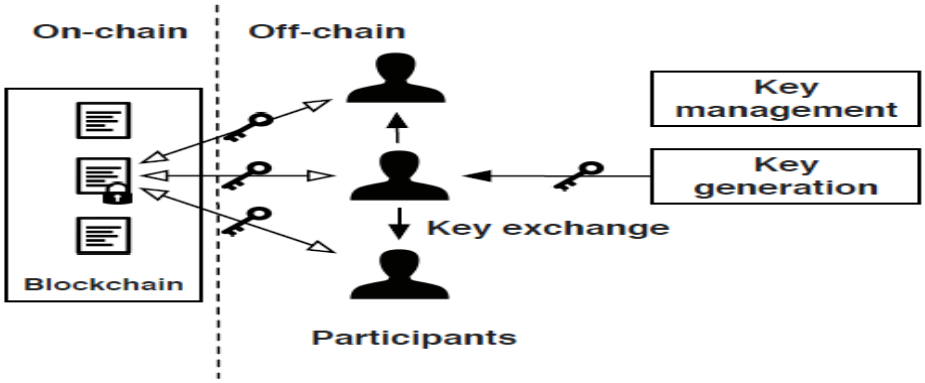


Figura 7. Encrypting On-chain Data Pattern.
Tomado de Xu X et al. [31]

- **Consecuencias**

  Pros:

  - **Seguridad:** La información de acceso público se cifra por lo cual los participantes que no tenga la clave secreta no podrá tener acceso la transacción ni descifrar la información [19], es decir, los datos que han sido guardados confidencialmente en Blockchain no podrán ser accedidos por participantes sin la clave secreta [37].

  Contra:

  - **Seguridad:** La transferencia de la clave secreta debe hacerse fuera de la cadena independientemente del método de cifrado que se utilice (simétrico o asimétrico) por lo cual si la gestión de las claves no se hace de forma correcta dichas claves se pueden ver comprometidas y otros usuarios podrían acceder a la información[37] [3].
  - **Inmutabilidad:** Los datos permanecerán en la cadena de bloques para siempre incluso si dichos datos se cifran, es por eso que aunque en el momento sean totalmente seguros y permiten acceso permanente a los participantes que tienen la clave secreta [37], con el tiempo el método de cifrado puede ser ineficiente y los datos podrían quedar expuestos ante ataques maliciosos.

- **Compromisos(concesiones):**
  - **Inmutabilidad vs Seguridad:** Como sabemos los datos en Blockchain son inmutables, es decir los datos no cambian y se mantienen permanentes en la cadena de bloques, esto hace que dichos datos sean seguros en el sentido que no están prestos al cambio, pero puede significar un defecto para los datos que se requiere no sean visibles para todos los participantes de la red, ya que para esto generalmente se utilizan métodos de cifrado de datos que se pueden ver vulnerados con el tiempo.

- **Tácticas**
  - **Cifrar Datos (Seguridad):** El cifrado de los datos o hashes criptográficos se utiliza para evitar el acceso no autorizado a la información disponible en la cadena de datos [24].

- **Usos conocidos**
  - **MLGBlockchain:** firma digital criptográfica.
  - **Oraclize:** Oraclize permite a los desarrolladores de contratos inteligentes cifrar los parámetros de sus consultas localmente mediante el uso de una clave pública antes de pasarlos a un contrato inteligente [34].

- **Patrón 2:** Off-Chain Data Storage

  - **Contexto:** Una de las principales decisiones de diseño que debemos tener en cuenta al usar una cadena de bloques es el almacenamiento de información que va dentro y va fuera de la cadena [39][3].

  - **Problema:** Blockchain garantiza la integridad de los datos y es por ello que algunas aplicaciones toman la decisión de incluirla en sus diseños e implementaciones, pero debido a la replicación completa de los datos, Blockchain tiene una capacidad límite de almacenamiento por bloque, es por ello que las aplicaciones deben tener en cuenta un equilibrio apropiado de los datos que se incluyen en la cadena y los que quedan fuera de la cadena [24][6]. Además, Blockchain no garantiza la integridad e inmutabilidad de los datos que se encuentran fuera de la cadena.

  - **Solución:** La Seguridad de los datos fuera de la cadena es garantizado mediante la utilización de un valor hash que es almacenado en la cadena, dicho valor hash es único para cada fragmento de datos, es decir si los datos almacenados fuera de la cadena son modificados incluso si es solo un bit el valor hash de este nuevo fragmento de datos va a ser muy diferente al almacenado en cadena, es de esta manera que se garantiza la integridad de los datos fuera de la cadena [36][37] y la integridad del valor hash agregado en cadena es garantizado por Blockchain [37][40]. Una técnica común muy utilizada es almacenar los metadatos en cadena y los datos grandes y privados mantenerlos fuera de la cadena [6].

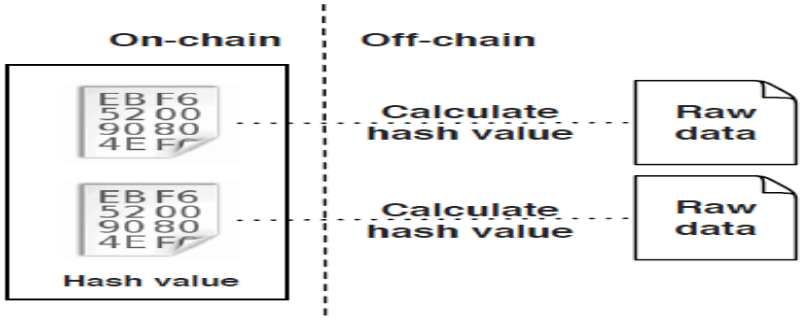


Figura 8. Off-chain Data Storage Pattern.
Tomado de Xu X et al. [31]

  - **Consecuencias.**
    Pros:
    - **Integridad:** Blockchain garantiza la integridad de los datos haciendo uso de un valor hash que representa los datos sin procesar [36]. La integridad de los datos se verifica mediante reglas algorítmicas y técnicas criptográficas [24] y para ello es necesario hacer uso del valor hash que se encuentra almacenado en la cadena de datos.

- **Escalabilidad:** Blockchain proporciona una escalabilidad limitada debido a que cada bit de datos se replica en todos los nodos, donde estos se mantienen de forma permanente, por ello cuando se hace uso del almacenamiento fuera de la cadena se aumenta la escalabilidad [36].

Contra:

- **Seguridad:** Los datos sin procesar que son almacenados fuera de la cadena podrían no ser tan seguros como en Blockchain debido a que estos datos podrían ser modificados y por ende el hash almacenado en cadena no coincidiría, lo que generaría perdida de información [36].

- **Compromisos:**
  - **Seguridad vs integridad:** Como los datos sin procesar se almacenan fuera de la cadena, se pueden eliminar o perder. Solo su valor hash permanece permanentemente en la cadena de bloques [36], lo que generaría un agujero de seguridad.
  - **Seguridad vs Escalabilidad**: Como los datos sin procesar se encuentran fuera de cadena, la escalabilidad de la aplicación aumenta, pero de la misma forma disminuye la seguridad de dichos datos, esto es debido a que la información se puede perder si se realizan modificaciones de los datos fuera de cadena ya que el valor hash no correspondería con el que se encuentra almacenado en cadena.
  - **Escalabilidad vs Latencia:** Estos dos compromisos se encuentran estrechamente relacionados al atributo de calidad “desempeño”, por lo cual cuando almacenamos datos fuera de la cadena, estamos liberando carga lo cual a su vez corresponde a una disminución de la latencia [5] y un aumento de la escalabilidad [36] lo cual según el estudio de Medellín et al [38] se traduce en una compensación de diseño entre la disponibilidad vs rendimiento.

- **Tácticas:**
  - **Verificar la integridad del mensaje (Seguridad):** Esta táctica emplea como por ejemplo valores hashes para verificar la integridad de los mensajes o archivos [5]. En Blockchain se hace uso de los hashes para verificar la integridad de los datos que se almacenan fuera de la cadena [24].
  - **Aumentar los recursos (Desempeño):** Al hacer uso del almacenamiento fuera de cadena se aumentan los recursos y se libera carga de almacenamiento en Blockchain ya que, este hace una replicación completa de toda la información [24], es por ello que esta táctica nos puede ayudar a disminuir la latencia [5] y mejorar la escalabilidad [36].

- **Usos conocidos:**
  - **Siaicoin, Storj y Filecoin:** Son plataformas que permiten a los participantes hacer uso del almacenamiento fuera de cadena de archivos de Blockchain de manera encriptada y replicada [36].

- **Patrón 3:** Tokenization

  - **Contexto:** Los tokens son utilizados como medios para representar el costo de activos de mucho valor, con ello se disminuye el riesgo del manejo financiero total

de los activos [41]. Es decir, los tokens carecen de un valor especifico a diferencia de las criptomonedas, generalmente tiende a confundirse estos términos, pero hacen referencia a contextos distintos. Los tokens pueden representar cualquier cosa existente en el mundo real.

- **Problema:** Los tokens que representan activos deben ser la fuente autorizada de los activos correspondientes, es decir, estos deben preocuparse por la forma en que se van a representar sus activos de información ya que están creados para representar cualquier cosa del mundo, por ejemplo: una casa, acciones de empresas, entre otros [33]. Por su lado la criptomoneda es un medio de intercambio de valores que opera en su propia Blockchain [38] y si bien los tokens se pueden utilizar como medio de intercambio ese no es su fin porque como ya mencionamos están creados para representar cualquier cosa. Es decir, el problema de los tokens radica en cómo garantizar que estos representen los activos físicos correspondientes.

- **Solución:** Blockchain proporciona lo necesario para el desarrollo de los tokens, es por ello que se deben tener en cuenta dos fases importantes, la primera la Blockchain y criptomoneda que nos permita desarrollar el token y la segunda es definir una estructura de datos en un contrato inteligente en el que se especifique todo lo que se puede hacer con el token, un token en Blockchain es la fuente autorizada del activo físico, es por ello que el usuario puede canjear dicho token para obtener la propiedad del valor asociado [41].

- **Consecuencias:**

  Pros:
  - **Seguridad:** Los tokens aumentan la seguridad al disminuir el riesgo de manejo financiero de propiedades de alto valor, ya que estos permiten reemplazarlos por equivalentes, además Blockchain proporciona una infraestructura confiable que permiten la creación de tokens autorizados de los activos o propiedades correspondientes.
  - **Confiabilidad:** El modelo de Blockchain en compañía de los contratos inteligentes nos provee una infraestructura confiable, la cual nos proporciona tokens autorizados y a su vez administra y controla los tokens que se encuentran en circulación [41].

  Contra:
  - **Integridad:** Como se sabe la integridad de los datos o de los tokens está garantizado por Blockchain, pero este no puede garantizar la autenticidad del activo físico que representa el token.

- **Compromisos:**
  - **Integridad vs Seguridad y Confiabilidad:** Blockhain nos proporciona integridad, seguridad y confiabilidad de los datos que se encuentran dentro de la cadena, sin embargo, cuando hacemos uso de los tokens autorizados para representar activos físicos, sacrificamos la integridad de datos asociados a dichos activos, pero a su vez ganamos seguridad y

confiabilidad ya que no realizamos manejo directo de esos activos de información que pueden llegar a tener un alto valor.

- **Tácticas:**
  No encontradas.

- **Usos conocidos:**
  - **Digix:** Este proyecto proporciona la infraestructura necesaria para crear tokens que están respaldados por activos físicos en las cadenas de bloques y sirven para rastrear la propiedad del oro como una propiedad física [25].
  - **Aragon:** Es un proyecto de Ethereum que busca des intermediar la creación y el mantenimiento de estructuras organizativas, en este proyecto los tokens se utilizan para representar la participación en la organización [34].

- **Patrón 4:** Off-Chain secret enable dynamic authorization

  - **Contexto:** Las aplicaciones basadas en Blockchain (públicas o privadas) requieren de una gestión de permisos para autorizar algunas actividades y a los participantes dentro de la red [24], dichos participantes son desconocidos al realizar la primera transacción.

  - **Problema:** Un contrato inteligente consiste en lo siguiente: múltiples autoridades, enlace dinámico y permisos integrados. Como se mencionó anteriormente en las aplicaciones basadas en Blockchain se requiere una gestión de permisos para autorizar algunas actividades, pero dicha autoridad puede ser desconocida cuando se envía una primera transacción a Blockchain, al implementar el contrato inteligente correspondiente o cuando la transacción es enviada a la cadena de bloques [37]. Blockchain utiliza firma digital para la autenticación y la respectiva autorización de la transacción [36], además de esto Blockchain no admite el enlace dinámico en situaciones donde la dirección de un participante no se encuentra definida inicialmente en el contrato inteligente o la respectiva transacción [37], es decir, para autorizar una segunda transacción todos los participantes deben definirse en la primera transacción antes de que esta sea agregada a la cadena de bloques.

  - **Solución:** Si un participante no se encuentra definido inicialmente en el contrato inteligente una vez se realiza la primera transacción se puede usar un secreto (clave hash) fuera de la cadena lo cual permite habilitar la autorización dinámica lo que significa que el acceso a los datos se encuentra protegido por una clave hash que es enviada junto con la información que contenga el contrato inteligente y quien contenga dicha clave o secreto fuera de la cadena podrá obtener o adquirir los datos de la información que contiene el contrato inteligente [37]. Cabe destacar que una vez revelado el secreto este no va a poder ser usado en otras transacciones y si múltiples transacciones se encuentran bloqueadas con el mismo secreto, al desbloquear una se desbloquean las demás.

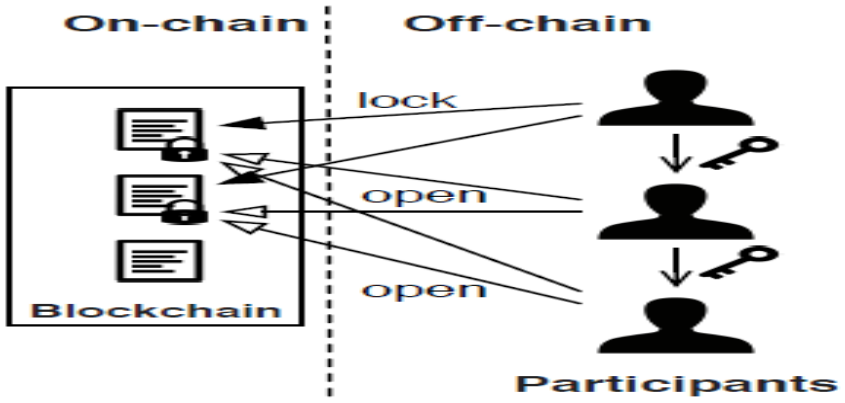


Figura 9. Off-chain Secret Enabled Dynamic Authorization Pattern. Tomado de Xu X et al. [31]

- **Consecuencias:**
  Pros:
  - **Interoperabilidad:** La clave secreta se puede intercambiar de cualquier manera fuera de la cadena.

  Contra:
  - **Seguridad:** Al intercambiar la clave secreta fuera de la cadena Blockchain no puede garantizar la seguridad de la misma.

- **Compromisos:**
  - **Seguridad vs interoperabilidad:** La interoperabilidad que permite el patrón influye en la seguridad del contenido del contrato inteligente ya que al permitirnos compartir el secreto fuera de cadena, este queda expuesto y Blockchain no puede garantizar su seguridad.

- **Tácticas:**
  - **Cifrar Datos (Seguridad):** El cifrado de los datos o hashes criptográficos se utiliza para que el participante no definido desde la primera transacción y que contenga dicho secreto pueda acceder a la información contenida en el contrato inteligente [24].

- **Usos conocidos:**
  - **Red Raiden:** Es una red de canales de pago fuera de cadena que permite realizar transferencias seguras y es una solución de escalamiento, permitiendo así pagos escalables, rápidos y a costos muy bajos, Este es complementario a Ethereum [3].

- **Patrón 5:** State Channel

  - **Contexto:** Blockchain cuenta con el potencial necesario para realizar micropagos a un costo muy bajo, los micropagos son pagos que pueden ser tan pequeños como porciones de céntimo de dólar. Pero aun cuando el costo de las transacciones de micropagos es bajo, es preciso preguntarnos si vale la pena almacenar estas transacciones en Blockchain.

- **Problema:** Debido a la arquitectura descentralizada de Blockchain este cuenta con algunas limitaciones principales que son: el rendimiento y la privacidad. El rendimiento de la cadena de bloques depende directamente de la implementación de la cadena de bloques, es decir, si es de consorcio, privado o público. En Blockchains de consorcio o privadas esta característica se puede configurar para obtener un mayor rendimiento, pero en el caso de las Blockchains públicas como lo es el caso de Bitcoin su rendimiento es menor [24] ya que en promedio estas Blockchains pueden manejar de 3 a 20 transacciones por segundo [24] e incluso las transacciones pueden demorar varios segundos o hasta una hora. Debido al largo tiempo de compromiso y las altas tarifas de transacción en Blockchains públicas, donde estas tarifas no están ligadas al monto de la transacción, se puede decir que es inviable realizar micropagos a través de la red Blockchain.

- **Solución:** Como se ha mencionado anteriormente la escalabilidad afecta principalmente a Blockchains publicas ya que cadenas de bloques como lo son Bitcoin y Ethereum solo pueden manejar en promedio de 3 a 20 transacciones por segundo [6], es por ello que algunos proyectos han buscado mejorar este atributo de calidad, una de las soluciones que se están estudiando es aumentar el tamaño de los bloques para permitir que los mineros agreguen más transacciones a dichos bloques [6]. Por lo cual, teniendo en cuenta la situación planteada anteriormente se considera no es factible el almacenamiento de cada transacción de micropagos en Blockchain, debido a que el valor monetario asociado a cada transacción es muy bajo. Por lo tanto, una solución es crear un canal de pago entre dos participantes [6], en el caso por ejemplo de Bitcoin esta permite realizar transacciones fuera de la cadena [24], una vez que ambas partes se ponen de acuerdo y desean cerrar el canal de estado de micropagos y así finalizar la transferencia de valor, se envía la transacción a la cadena de bloques de Bitcoin [31][6]. A modo general el canal de estado de pago mantiene los estados intermedios de los micropagos y solo permite almacenar en cadena el pago final. Cabe destacar que la frecuencia de las transacciones se realiza mediante el acuerdo realizado entre los participantes.

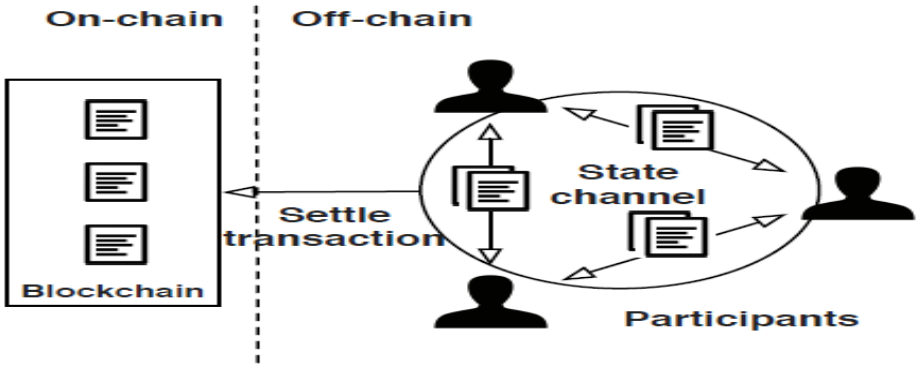


Figura 10. State Channel Pattern.
Tomado de Xu X et al. [29]

- **Consecuencias.**
  Pros:
  - **Escalabilidad:** Con este patrón se busca equilibrar la escalabilidad debido a que Blockchain tiene una escalabilidad limitada porque cada bit de datos es replicado en todos los nodos y es permanente [24], así que al realizar

los micropagos fuera de cadena se evita realizar la replicación de datos de dichas transacciones.

- **Desempeño:** En lo que corresponde al desempeño, una importante decisión de diseño es cuales datos deben mantenerse en cadena y cuales fuera de cadena [24], es por ello que en el caso de realizar los micropagos fuera de la cadena, este proceso no está limitado por las configuraciones de Blockchain como lo son: el tamaño del bloque, intervalo del bloque, límite de gas, etc. Lo que permite obtener un desempeño mayor al que se puede lograr estando dentro de la cadena.
- **Latencia:** La latencia en sistemas Blockchain se ve afectada por la minería [6], es por ello que al realizar las transacciones de micropagos fuera de la cadena evitamos realizar este proceso y de esa manera se mejora el tiempo de respuesta de las transacciones ayudando a preservar la privacidad de los datos y disminuyendo la latencia existente en la Blockchain [6].
- **Confidencialidad:** La confidencialidad o privacidad de las transacciones de micropagos es garantizada en Blockchain siempre y cuando estas sean realizadas fuera de la cadena debido a que estas no son visibles para los participantes de la Blockchain. Como se sabe todas las transacciones intermedias de micropagos se realizan fuera de la cadena, esto quiere decir que solo la última transacción queda almacenada en cadena.

Contra:

- **Integridad:** Blockchain no puede garantizar la integridad de la información almacenada fuera de cadena debido a que no cuenta con la inmutabilidad que proporciona Blockchain a los datos almacenados dentro de la cadena, por lo tanto, las transacciones de micropagos pueden no ser tan confiables.

- **Compromisos:**
  - **Integridad vs Escalabilidad/Desempeño:** Al realizar las transacciones de micropagos fuera de la cadena se sacrifica la integridad de los datos al no contar con una de las principales cualidades de la Blockchain que es la inmutabilidad, pero a la vez ganamos en escalabilidad ya que no se realiza la replicación completa de estos datos dentro de la cadena, además, el desempeño de las transacciones fuera de cadena es mucho más ágil, al no estar limitado por las configuraciones de Blockchain.
  - **Confidencialidad vs Escalabilidad/Desempeño:** Cuando las transacciones de micropagos son realizadas fuera de cadena obtenemos confidencialidad y privacidad de los datos debido a que esta información no se encuentra replicada en toda la red, además, de eso ganamos en escalabilidad y desempeño debido al mismo factor mencionado.

- **Tácticas:**
  - **Aumentar los recursos (Desempeño):** Al realizar los micropagos fuera de la cadena se están aumentando recursos, es por ello que hay un mejor desempeño y escalabilidad en el sistema Blockchain ya que estos

micropagos no se ven afectados por estas características propias de la red, además esta táctica nos permite disminuir la latencia [5].

- **Reducir complejidad computacional (Desempeño):** reducimos la sobrecarga computacional en la cadena de bloques al realizar las transacciones de micropagos fuera de la cadena [24].
- **Mantener múltiples copias (Desempeño):** almacenamos información en el canal de estado y al final hay información que se encuentra en la Blockhain, se tienen las dos copias, pero únicamente se hace uso de la cadena de bloques cuando se necesita [6].

- **Usos conocidos:**
  - **The Lightning Network:** El Lightning Network o red de canales de pagos es una red descentralizada para micropagos fuera de la cadena, para Bitcoin y otras criptomonedas. Esta red de canales de pago establece una transacción de firma múltiple entre los dos participantes generando así el canal de micropagos y de esta manera poder transferir el valor de la cadena [24], es decir, los micropagos se habilitan cuando se establece un canal de pago bidireccional, mediante el compromiso de los participantes. El canal de pago puede cerrarse enviando una transacción a la cadena de bloques, siendo esta la transacción final.

  - **The Raiden Network:** Es complementario a la cadena de bloques Ethereum y similar a The Lightning Network [42]. The Raiden Network es una solución de escalamiento fuera de la cadena Que permite que se realicen micropagos escalables a bajos costos y de manera rápida [28]. La idea principal es evitar el cuello de botella del consenso utilizado por Ethereum, aprovechando los beneficios de una red de canales de pago fuera de la cadena que permite transferir de manera segura su valor monetario, dicho valor es depositado a los canales de pago mediante el uso de contratos inteligentes.

- **Patrón 6:** Multiple Authorization

  - **Contexto:** En las aplicaciones basadas en Blockchain, algunas actividades (representada por una transacción) podrían basarse en múltiples autoridades [3] [37], es decir, deben ser autorizadas por múltiples direcciones de Blockchain [3]. por ejemplo, una transacción monetaria puede requerir la autorización múltiple de direcciones Blockchains

  - **Problema:** La principal problemática está relacionada a la disponibilidad de las autoridades, es decir que las direcciones que autorizan una actividad (Representada por una transacción) [3][37] no puedan decidirse o ponerse de acuerdo debido a la disponibilidad de las autoridades. Otra variante se puede expresar en términos de recuperación de claves perdidas. Al ser Blockchain un entorno descentralizado no contamos con un proveedor, es decir, un sistema centralizado que nos permita recuperar contraseñas, por tal motivo el usuario es el único encargado de recordar su par de claves (pública y privada). Si el usuario pierde las claves, esto implica la perdida permanente del control sobre una cuenta y a su vez de los contratos inteligentes [3].

- **Solución:** El mecanismo de múltiples autoridades permite al participante controlar los contratos inteligentes utilizando más de una dirección de Blockchain, con esto se logra reducir el riesgo de perder el control o dominio sobre sus contratos inteligentes en caso de perder la clave privada o comprometida [3]. Los servicios de múltiples autoridades se centran en transacciones, que necesitan ser autorizadas por múltiples direcciones de Blockchain [37] es por ello, que para permitir más dinamismo en el sistema el conjunto de direcciones para la autorización no se decide antes de que se envié la transacción a la red Blockchain o se implemente el contrato inteligente. Como en este patrón hay múltiples autoridades en una red Blockchain, este puede proporcionar flexibilidad para lograr una mejor cooperación [43]. Una transacción es válida solo cuando hay suficientes firmas de las autoridades y además podemos considerar este patrón como un mecanismo que nos permite salvaguardar nuestra clave privada o comprometida, ya que como se menciona anteriormente Blockchain no nos proporciona esta funcionalidad [43]. En Bitcoin, la autorización múltiple o firma múltiple requiere más de una clave privada para autorizar una transacción y en el caso de Ethereum el mecanismo de autorización múltiple se implementa como un contrato inteligente, específicamente una firma múltiple M de N que define M de N claves públicas son necesarias para autorizar una transacción [43]. Con este método si hay N direcciones de autoridad y el umbral es M, entonces, al menos M claves privadas entre las N claves privadas deben mantenerse de manera segura, para de esta forma evitar perder el control [37].

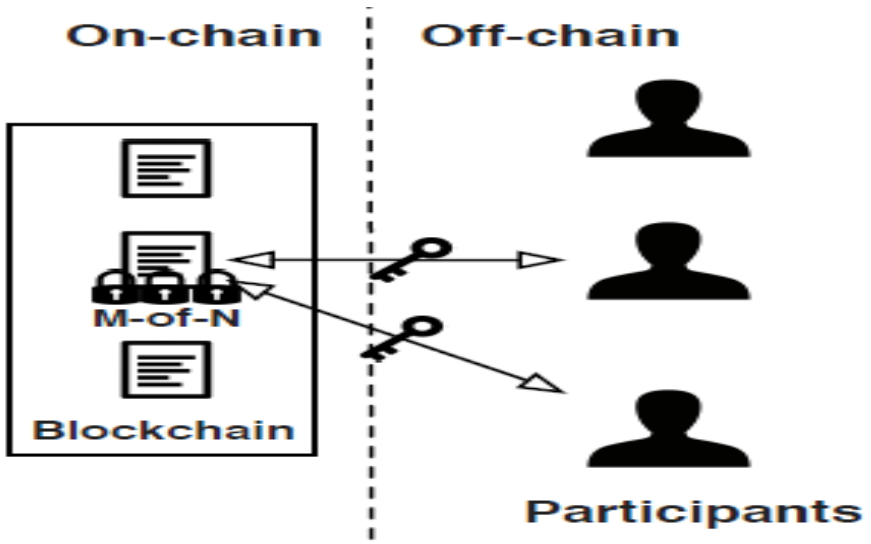


Figura 11. Multiple Authorization Pattern.
Tomado de Xu X et al. [31]

- **Consecuencias.**
  Pros:
  - **Seguridad:** En Blockchain no es posible recuperar claves, al perder las claves se pierde acceso a los contratos inteligentes, es por ello que la autorización múltiple permite tener menos riesgo de pérdida al utilizar más de una dirección de Blockchain [44].

Contra:

- **Seguridad:** A pesar que la firma múltiple, permite tener más control, se debe tener en cuenta que al menos M claves privadas entre las N claves privadas deben mantenerse de manera segura para así evitar perder el control [37], esto quiere decir que, si se escogen pocas claves privadas, la seguridad se puede ver comprometida.
- **Disponibilidad:** En el caso por ejemplo de Ethereum en el que se necesita una firma múltiple M de N, en la cual se necesita M de N claves públicas para autorizar una transacción, donde M es el umbral [37]. Entonces se podría decir que esto permite garantizar una mayor disponibilidad de las direcciones asociadas.

- **Compromisos:**
  - **Seguridad vs Disponibilidad:** a mayor seguridad se necesitar incrementar la disponibilidad de los nodos y por lo tanto se necesita mejorar ese aspecto, a nivel físico se necesita tener más infraestructura.

- **Tácticas:**
  - **Autenticar actores (Seguridad):** Esta táctica apoya la característica principal del patrón que es la autenticación de múltiples autoridades, como se menciona en el libro de Bass et al. [5] *"La autenticación significa asegurarse de que un actor, en este caso las direcciones autenticadas sean seguras, es decir que se pueda validar que sea realmente quien se pretende ser".*

- **Usos conocidos:**
  - **Multisignature:** Es un mecanismo de firma múltiple proporcionado por Bitcoin.
  - **Multisignature wallet:** Es un mecanismo de firma múltiple, escrito en solidity, el cual es ejecutado en Ethereum.

- **Patrón 7:** Oracle

  - **Contexto:** Desde la perspectiva de la arquitectura de software Blockchain puede verse como un conector dentro de un gran sistema software, es decir, Blockchain es un conector complejo basado en la red [6], ya que un conector es un mecanismo de interacción para los componentes. Los conectores en sistemas distribuidos son elementos claves que nos permiten lograr atributos de calidad importantes tales como: rendimiento, confiabilidad, seguridad, entre otros [6]. Para casos generales los diferentes componentes de la arquitectura pueden coordinar sus cálculos a través de la cadena de bloques. Para lograrlo, es posible enviar transacciones a los contratos inteligentes para invocar las funciones que han sido definidas en estos [6], pero en el caso de requerir utilizar Blockchain para casos diferentes a los servicios financieros convencionales, las aplicaciones creadas en Blockchain pueden necesitar de servicios externos.

  - **Problema:** El entorno de ejecución de las diferentes Blockchain es autónomo, solo puede acceder a la información presente en una transacción o en el historial de transacciones de la cadena de bloques [24], estos entornos garantizan la seguridad de la red Blockchain. Los contratos inteligentes se ejecutan en entornos

tales como: EVM en el caso de Ethereum y contenedores (Dockers) en el caso de Hyperledger, los cuales no permiten importaciones externas [34], estos solo pueden acceder a la información presente en las cadenas de bloques. Es decir, el estado de los sistemas externos no es directamente accesible a los contratos inteligentes, sin embargo, estos contratos inteligentes en ocasiones necesitan acceder a dichos estados externos.

- **Solución:** Como se ha mencionado Blockchain por sí solo no permite realizar la conexión al mundo externo, es por ello que para abordar esta limitación se hace necesario el uso de un Oracle. Oracle permite introducir condiciones que no se pueden expresar en contratos inteligentes ejecutados en el entorno de Blockchain, los Oracles son fuentes de datos confiables, para proporcionar estados externos sobre el mundo real en forma de una transacción ya que cualquier información que no es generada dentro de la cadena mediante una transacción debe ser incluida como datos adjuntos a una transacción de manera segura y confiable [34], cuando la validación de una transacción depende del estado externo, se solicita a Oracle que verifique el estado externo, esto podría bloquear el procesamiento de la transacción hasta que el Oracle de validación que posee una o varias de las direcciones predefinidas verifique una condición sobre el estado externo [3]. Dicho Oracle también se puede implementar dentro de una red Blockchain como un contrato inteligente con un estado externo que se inyecta periódicamente en el Oracle por fuera de la cadena.

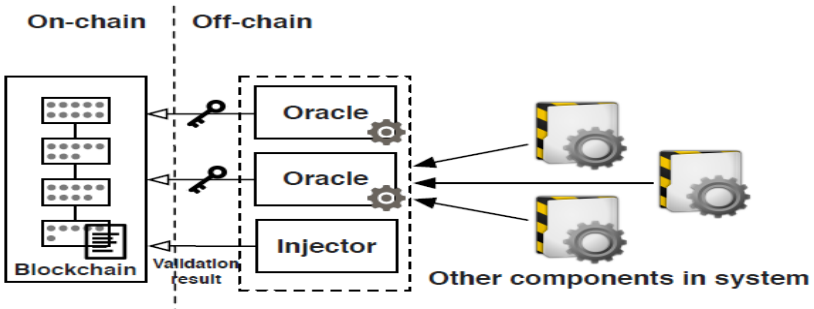


Figura 12. Oracle Pattern
Tomado de Xu X et al. [31]

- **Consecuencias:**
  Pros:
  - **Disponibilidad:** Al introducir el concepto de Oracle a Blockchain se hace posible tener acceso a los datos de mundo externo, es decir esto se convierte en disponibilidad de la información.
  - **Interoperabilidad:** La interoperabilidad en el sistema se da debido a que el entorno cerrado de Blockchain se encuentra conectado con el mundo externo a través de Oracle.

  Contra:
  - **Seguridad:** Los estados externos en las transacciones no pueden ser completamente validados por los mineros, es por ello que cuando los mineros validan la transacción junto al estado externo, estos confían en Oracle para verificar su valides, pero podría suceder que si bien dentro de la cadena las transacciones son inmutables, fuera de esta el estado de validación puede cambiar después de haber sido agregadas las

transacciones a la cadena de bloques, afectando a su vez la disponibilidad de la información que se encuentra fuera de la cadena.

- **Confiabilidad:** Cuando se hace uso de Oracle, se introduce un tercero de confianza en el sistema. El Oracle seleccionado debe ser confiable para todos los involucrados en las transacciones.

Compromisos:

- **Seguridad vs Disponibilidad:** Como los mineros no pueden validar por completo el estado externo de la información confían en Oracle con lo cual se introduce un tercero de confianza que afectaría la seguridad del sistema, pero aumentaría la disponibilidad al permitir la conexión con el mundo real.

- **Tácticas:**
  - **Separar entidades (Seguridad):** Oracle nos permite hacer la separación física de los datos, es decir, nos permite acceder a información del mundo externo. Con Oracle estamos permitiendo que Blockchain se extienda a más aplicaciones, se amplía el espectro de aplicación de la Blockchain.
  - **Verificar mensajes de integridad (Seguridad):** Oracle se encarga de la verificación de la información del estado externo y este proporciona el resultado al validador(minero), quien toma en cuenta el resultado proporcionado por oracle al validar la transacción.

- **Usos conocidos:**
  - **Oracle:** En Bitcoin un Oracle de validación automatizado puede implementarse como un servidor fuera de la red Blockchain, el cual tiene su propio par de claves [6].
  - **Orisi:** Permite a los participantes involucrados en un contrato seleccionar un conjunto de Oracle en el que se sientan cómodos antes de usar dicho contrato para luego firmar un contrato que requiere un cierto número de firmas de validación de Oracle [6].

- **Patron 8:** Contract Registry

  - **Contexto:** En todos los sistemas o aplicaciones existentes de software se hace necesario realizar actualizaciones, lo que da como resultado nuevas versiones del mismo, sin embargo, Blockchain no funciona de la misma manera por ello, para lograr obtener esta funcionalidad se hace necesario que las funciones, disparadores, condiciones o lógica empresarial definida en los contratos inteligentes [24] se puedan actualizar para de esta manera poder corregir errores y cumplir nuevos requisitos.

  - **Problema:** Los almacenes de datos compartidos, exportan una interfaz básica que contiene un CRUD (Crear, Leer, Actualizar y eliminar), Blockchain es un almacén de datos solo para agregar, ya que en este no están permitidas las actualizaciones, cualquier cambio que queramos realizar es agregado a la cadena de bloques como una nueva transacción [6]. Una vez se ha definido o implementado un contrato inteligente en Blockchain, este no puede ser modificado

o actualizado [43] esto es debido a la característica de inmutabilidad que proporciona Blockchain.

- **Solución:** Un contrato de registro realizar actualizaciones sobre los contratos inteligentes existentes, dicha actualización es posible realizarla de la siguiente manera: El creador de un contrato puede registrar el nombre y la dirección del nuevo contrato en el contrato de registro, después de haber sido creado dicho contrato. El invocador de un contrato registrado recupera la última versión del nuevo contrato inteligente desde el contrato de registro, de esta manera las funciones correspondientes al contrato registrado pueden actualizarse, esto se hace actualizando la versión anterior del contrato con la nueva versión, cabe destacar que, si los contratos inteligentes tienen dependencias con otros contratos, estas no se rompen al realizar la actualización. Además de esto se dice que un contrato de registro representa un acuerdo legal, el cual está vinculado al contrato inteligente en cadena mediante la adicción de la dirección del contrato inteligente en el acuerdo legal y agregando el hash del contrato legal a una variable del contrato inteligente, así establecemos un puente entre el acuerdo legal y el contrato inteligente [3].

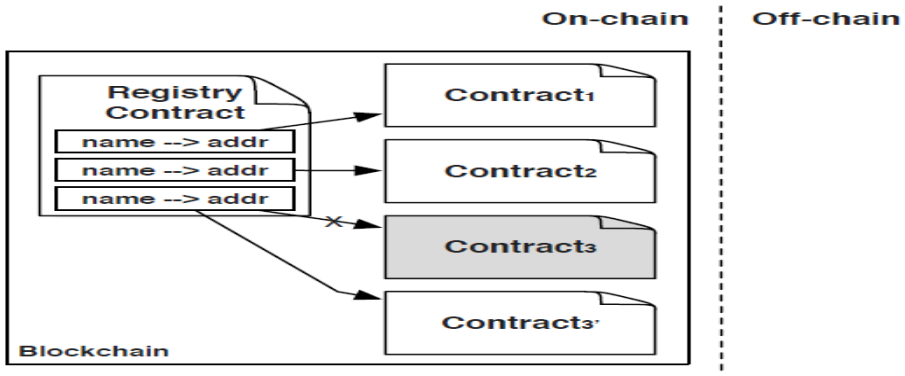


Figura 13. Contract Registry Pattern. Tomado de Xu X et al. [31]

- **Consecuencias:**

  Pros:

  - **Modificabilidad:** El contrato inteligente que se encuentra asociado a un contrato de registro puede ser modificado o actualizado mediante una nueva versión sin romper las dependencias del contrato inteligente.

  Contra:

  - **Modificabilidad:** Aunque el contrato de registro nos permite la modificación de los contratos inteligentes, estos también tienen algunas restricciones, ya que la actualización sigue estando limitada si las funciones que están definidas en el contrato inteligente son llamadas o dependen de otros contratos, cabe destacar que aun cuando se presente este caso la implementación de la función puede actualizarse, pero la firma de la función no es posible modificarla sin romper las dependencias.
  - **Inmutabilidad:** Una de las principales características de la Blockchain es la inmutabilidad, por ello cada bit de datos incluidos los contratos inteligentes almacenados en Blockchain son inmutables, característica que se ve vulnerada al aplicar este patrón.

- **Compromisos:**
  - **Inmutabilidad vs Modificabilidad:** Cuando hacemos uso del patrón contrato de registro se ve vulnerada la característica de inmutabilidad de Blockchain debido a que de acuerdo a estas características los datos no deberían ser modificados ni actualizados, sin embargo, teniendo en cuenta que todo puede cambiar en el tiempo y que las funciones definidas en los contratos inteligentes también pueden cambiar se hace necesario el uso de este patrón.

- **Tácticas:**
  - **Actualización de software (disponibilidad):** La táctica actualización de software es crucial en este patrón debido a que en todo su contexto se habla de modificación o actualización, además el objetivo de esta táctica es lograr actualizaciones de manera que no se vean afectados los servicios [5], es decir cumple con las características del patrón.

- **Usos conocidos:**
  - **ENS 43:** Es un servicio de nombres de Ethereum que se implementa como contratos inteligentes ENS mantiene un mapeo entre los contratos inteligentes en cadena y los recursos fuera de cadena y los nombres simples y legibles por humanos. ENS se puede ver como un registro de contrato integrado en una plataforma Blockchain, que es accesible para todos.
  - **Regis 44:** Es una aplicación en el navegador que facilita la creación, implementación y administración de registros como contratos inteligentes en la cadena de bloques Ethereum.

En la descripción de patrones diseño que se acaban de presentar no se tuvieron en cuenta 7 de los 15 patrones mencionados en el artículo Xu X et al.[31] los cuales son: Confirmation X, Reverse oracle, data contract, embedded permission, incentive execution, Legal and Smart Contract Pair y Factory Contract Store. Esto es debido a que no se encontró información sólida en los demás artículos que nos llevara apoyar la descripción inicial que encontramos en el artículo base, además, cabe destacar que el no haber sido descritos minuciosamente en este trabajo no significa que carezcan de importancia o que no sean necesarios.

5.3.5 Q5: ¿Cuáles son los principales problemas que evidencia la tecnología Blockchain?

Los inicios de la tecnología Blockchain no fueron muy satisfactorios, pero a medida que ha pasado el tiempo la perspectiva respecto a Blockchain ha cambiado de tal forma que ha tenido gran acogida en los últimos años y muchos sistemas existentes han buscado la manera de incluirla en sus desarrollos para beneficiarse de sus propiedades únicas como lo son: el no repudio, la integridad, la transparencia, la inmutabilidad de los datos, entre otros [3]. Blockchain aporta muchos beneficios a los sistemas, pero la principal preocupación de diseño tiene que ver precisamente con dos de sus principales beneficios que son la inmutabilidad y la replicación completa de los datos, ya que debido a esto el tamaño del libro mayor sigue creciendo cada vez más [31]. Además, Los principales problemas de la

Blockchain son la escalabilidad, la privacidad de los datos y el desempeño [3][31], pero esta cuenta con muchas más limitaciones, algunas de ellas se describen a continuación:

1. **Desempeño:** El desempeño de los sistemas Blockchain está ligado a la implementación de la cadena de bloques, es decir, si es una Blockchain pública, privada o de consorcio. Por ejemplo, es posible obtener un mejor desempeño en Blockchains privadas o de consorcio debido a que permiten una mejor configuración de este atributo [3][45], en cambio en Blockchains públicas el desempeño del sistema es más limitado, por ello, las transacciones pueden demorar varios minutos o incluso una hora debido al largo tiempo de compromiso y las altas tarifas transaccionales (estas no necesariamente están ligadas al monto de la transacción) [31]. En cuanto a los contratos inteligentes estos son ejecutado en serie por validadores y mineros por lo que el rendimiento también se ve afectado, ya que no se aprovechan las arquitecturas actuales diseñadas para que dichas ejecuciones se realicen de forma simultánea o en paralelo lo que permitiría un mejor desempeño del sistema [34].

2. **Escalabilidad:** La escalabilidad y el desempeño están estrechamente ligados es por ello que la escalabilidad es otra de las limitaciones Blockchain, pero antes de presentar la limitación de la escalabilidad en dicha tecnología, es importante conocer que es la escalabilidad en Blockchain. Este término hace referencia a la cantidad de transacciones que la red puede procesar. Ahora bien, la escalabilidad de Blockchain es limitada porque cada bit de datos se replica en todos los nodos y estos datos se mantienen permanentes en la red, es por eso que los límites de la escalabilidad se presentan en los siguientes aspectos I) El tamaño de los datos en Blockchain, II) La tasa de procesamiento de transacciones y III) La latencia de la transmisión de los datos y los compromisos [43]. Las cadenas de bloques públicas como Bitcoin y Ethereum solo admiten de 3 a 20 transacciones por segundo, es por ello que buscan mejorar la escalabilidad tomando algunas decisiones de diseño como, por ejemplo: aumentar el tamaño del bloque en Bitcoin de 1MB a 8MB para permitir que los mineros incluyan más transacciones en los bloques [24].

3. **Privacidad de los datos:** La privacidad de los datos es una limitación de Blockchain debido a que la información que se encuentra en la cadena de bloques está disponible para todos los participantes, especialmente en Blockchains publicas ya que no hay usuarios privilegiados y la información es accesible para todos [31][8].

4. **Almacenamiento:** Blockchain cuenta con capacidad limitada de almacenamiento esto es debido a que cada participante de la red cuenta con una réplica local de todo el historial de transacciones [23][3] por ello, el almacenamiento de grandes cantidades de datos es casi imposible en Blockchain [46] por lo que se hace necesario el uso de almacenamiento dentro y fuera de la cadena denominado (on-chain, off-chain) [36], los beneficios de hacer uso de esta práctica es que se aprovechan todas las propiedades de Blockchain y se hace frente a la limitación de almacenamiento [47], pero por otro lado también cuenta con un problema al hacer uso del enfoque de almacenamiento fuera de la cadena y es que se requiere tener confianza en la persona que realiza el cifrado de la datos, además si no se crea un registro de auditoría, los datos cifrados que se encuentran fuera de la cadena no impiden por sí solos el acceso a ellos [31].

5. **Latencia:** Blockchain depende de una red de nodos para realizar la verificación de las transacciones, es por ello que entre él envió de la transacción y la verificación de la misma la latencia se ve afectada, generalmente el tiempo requerido para este fin es alrededor

de 1 hora (intervalo de bloque de 10 minutos con tiempo de inclusión y confirmación de 5 bloques) en Bitcoin, y alrededor de 3 minutos (intervalo de bloque de 14 segundos con 11 bloques de confirmación) en Ethereum [34].

6. **Irreversibilidad:** Cuando se implementa un contrato inteligente, este se vuelve irreversible, es decir, una vez finalizado el contrato inteligente no se puede modificar, es por ello que si ocurre un error en un contrato inteligente no hay una forma directa de solucionarlo [31].

7. **Integridad:** Blockchain cuenta con cinco propiedades principales que son: la inmutabilidad, el no repudio, integridad de los datos, transparencia e igualdad de derechos, sin embargo en el caso de la integridad de los datos aunque es garantizada gracias a reglas algorítmicas y técnicas criptográficas [24][36] esta propiedad puede verse comprometida debido a nodos no confiables o al remitente, esto se da cuando los hashes almacenados en cadena no corresponden con los datos no cifrados [40], además, usar Blockchain para garantizar la integridad de los datos es relativamente costoso en comparación con otros mecanismos que garantizan la persistencia de los datos [48].

4. ARQUITECTURA BLOCKCHAIN: UN MODELO ARQUITECTONICO PARA SOLUCIONES BASADAS EN BLOCKCHAIN

La tecnología Blockchain está en constante avance y crecimiento, esto da lugar a estar continuamente buscando mejores prácticas y estructuras arquitecturales que permitan mejorar los conceptos bases sobre la cual se construye. Bajo esta premisa se ha trabajado en este proyecto en proponer un modelo arquitectural basado en la tecnología Blockchain que permita agilizar la comprensión los conceptos arquitecturales sobre los cuales se fundamenta esta tecnología. Este trabajo se fundamenta en el hecho de que existe información arquitectural sobre la tecnología Blockchain, pero difícil de entender debido a que no se encuentra organizada y estructurada de una manera simple, metodológica y entendible.

El objetivo de esta investigación se centra en proponer una guía arquitectónica estructurada metodológicamente, donde se encuentran articulados patrones, tácticas y atributos de calidad con el fin de brindarle a las personas interesadas en implementar, adaptar o simplemente estudiar la tecnología Blockchain; una base de conocimiento simplificada que les permita agilizar y abstraer una idea para sus necesidades con la tecnología. Es importante destacar principalmente que no hay una única solución para aplicaciones basadas en Blockchain esto es debido a que cada aplicación requiere de aspectos diferentes en su arquitectura, también es importante decir que es posible tener una arquitectura de referencia, pero para ello se requiere de una mayor maduración de la disciplina y de la participación de la industria en consorcio con la academia Linux Fundation entre otras.

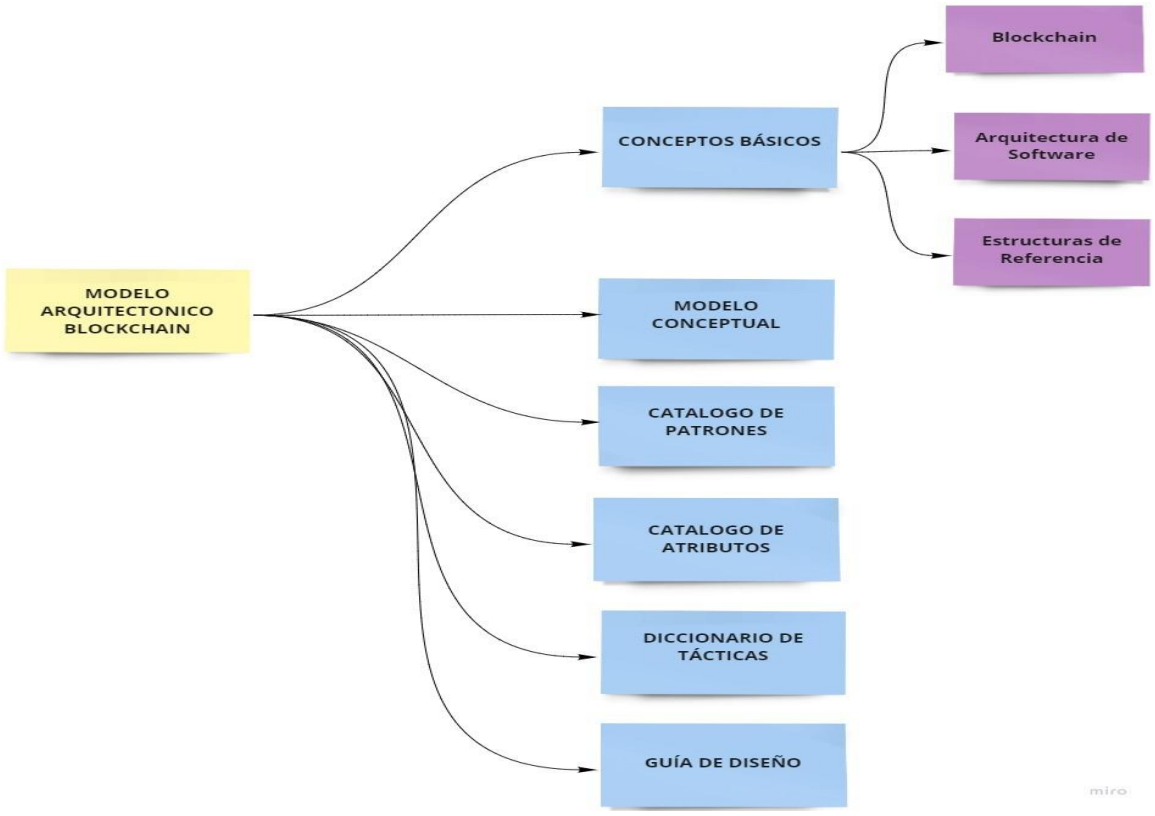


Figura 14. Modelo Arquitectónico Blockchain. Fuente propia

La arquitectura Blockchain, como se muestra en la figura (de arriba), está estructurada en 6 paquetes metodológicos, que se describen a continuación:

1. **Conceptos básicos:** Que relacionan la tecnología Blockchain con las arquitecturas de software. En este paquete se describen los conceptos básicos que debe conocer una persona para comprender el modelo arquitectónico.

2. **Modelo Conceptual:** es una representación de los artefactos involucrados en el modelo, donde se ilustra la relación e influencia entre ellos.

3. **Catálogo de Patrones:** Es una estructura metodológica de patrones arquitecturales en el contexto de la tecnología Blockchain, estos crean relaciones con los paquetes de catálogos de atributos de calidad y el de diccionario de tácticas.

4. **Catálogo de Atributos de calidad:** Es una estructura metodológica de atributos de calidad que se requieren en las bases de la tecnología Blockchain.

5. **Diccionario de Tácticas:** Describe las principales tácticas arquitecturales que se encuentran en la tecnología Blockchain

6. **Guía de Diseño:** Describe el proceso de aplicación del modelo arquitectónico.

A continuación, se describe en detalle cada uno de los paquetes metodológicos.

4.1 CONCEPTOS BASICOS DE LA ARQUITECTURA BLOCKCHAIN

En este primer paquete metodológico se establecen los conceptos de la tecnología Blockchain, arquitecturas de software y estructuras de referencia con el objetivo de dar claridad conceptual a las personas, empresas, arquitectos etc. Que estén interesadas en conocer la estructura arquitectural básica de una plataforma Blockchain. El paquete consta de los sub-paquetes “Conceptos de la Blockchain”, “Conceptos de la Arquitectura de Software” y “Conceptos de Estructuras de Referencia” se presentan en la **Figura 15. Conceptos básicos** y se describen a continuación.

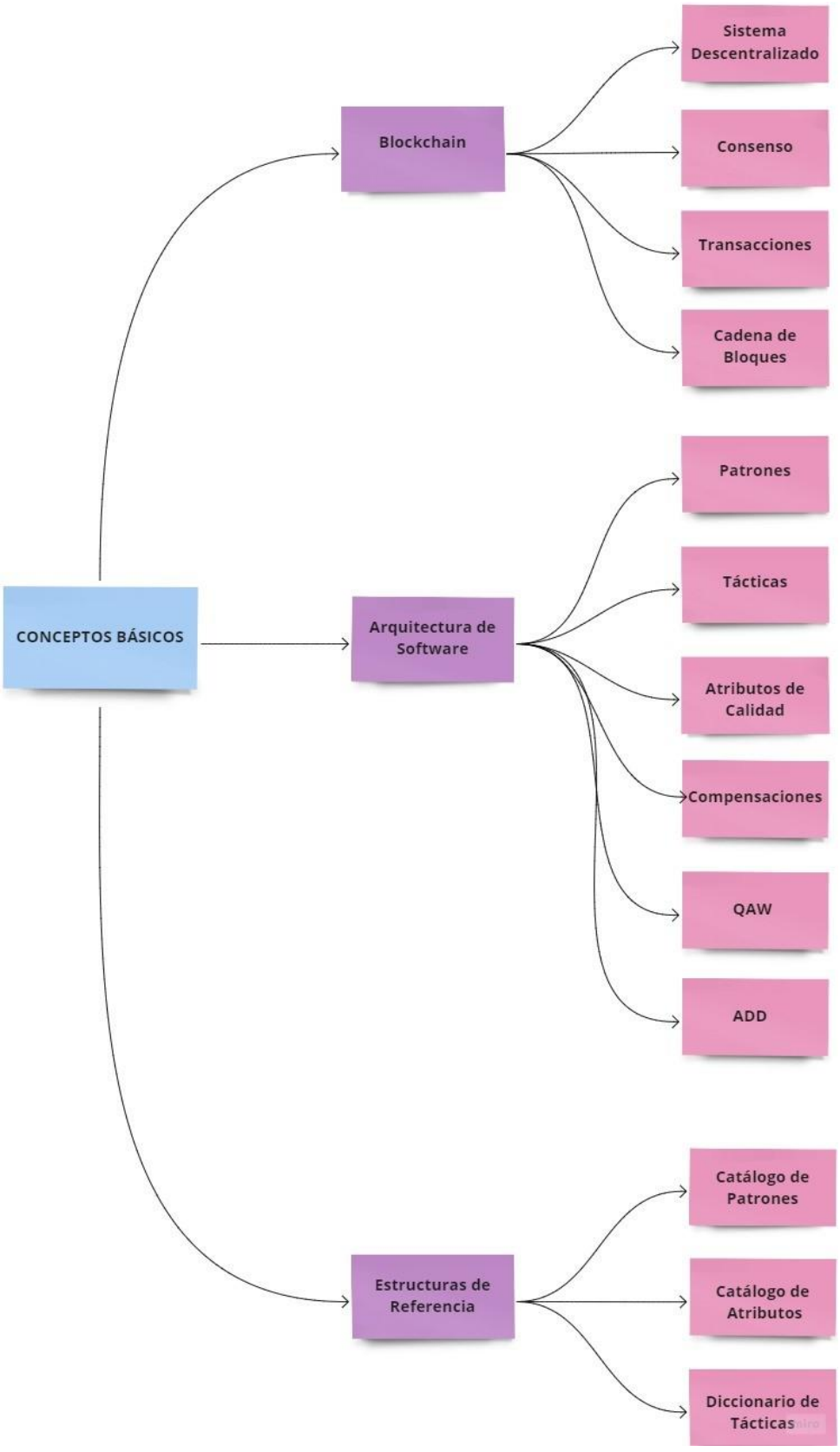


Figura 15. Conceptos básicos. Fuente propia

4.1.1 conceptos de la Blockchain

- **Sistema descentralizado:** Los sistemas descentralizados se caracterizan por no requerir de intermediarios de confianza o un punto de integración central para realizar transacciones [40].

- **Consenso:** El mecanismo de consenso es fundamental en la tecnología Blockchain dado que su función principal es determinar el consenso de todas las transacciones y el estado actual del sistema [24]. Este mecanismo garantiza que las transacciones solo sean agregadas una vez si son válidas. En la actualidad existen tres modelos dominantes que son: prueba de trabajo, prueba de participación y tolerancia a fallas bizantinas.

- **Transacciones:** Una transacción es una operación que almacena información en la Blockchain [31]**.** Cuando la transacción firmada se envía a un nodo este cumple la función de validarla, una vez sea validada la distribuye a los demás nodos que realizan el mismo proceso propagándola a sus pares hasta que la transacción se replica en toda la red [24]

- **Cadena de Bloques:** La cadena de bloques es una lista ordenada de bloques los cuales agrupan transacciones, cada bloque contiene un hash propio y un hash que lo encadena al bloque anterior [24]

4.1.2 conceptos de la arquitectura de software

- **Patrones:** Los patrones arquitectónicos son soluciones generales a una arquitectura de software, relacionando los componentes, de tal forma que cumplan con las exigencias de los atributos de calidad. El éxito de un patrón arquitectónico depende en gran parte de la manera en cómo éste se implemente. Diferentes arquitecturas de software con características similares pueden utilizar un mismo patrón [1].

- **Tácticas:** Las tácticas son técnicas usadas para tomar decisiones de diseño generales que facilitan alcanzar los atributos de calidad esperados e interactúan positiva o negativamente con los patrones arquitectónicos; normalmente las tácticas se implementan a través de los patrones arquitectónicos [13].

- **Atributos de Calidad:** Los atributos de calidad especifican la calidad de las respuestas del sistema [10], puede interpretarse como el grado en que dichos atributos satisface los requisitos de sus usuarios, conocidos también como requerimientos no funcionales [5].

- **Compensaciones:** Las compensaciones son formas sistemáticas de relacionar atributos de calidad, lo cual proporciona una base sólida para evaluarlos y cuyo objetivo principal es tomar decisiones que favorezcan el sistema [44]**.**

- **QAW (Quality Attribute Workshop):** Este método consiste en incluir a los stakeholders al inicio del ciclo de vida de un proyecto con el fin de descubrir y documentar atributos de calidad basado en escenarios [49].

- **ADD (Attribute-Driven Design):** El método ADD sirve para definir una arquitectura basada en la captura de escenarios realizada previamente y está enfocada en el proceso de diseño de atributos de calidad, haciendo uso de tácticas y patrones que los satisfagan [50].

4.1.3 conceptos de estructuras de referencia

A continuación, se presenta la estructura de referencia para el catálogo de patrones, atributos de calidad y el diccionario de tácticas:

4.1.3.1 Estructura catálogo de patrones

En este catálogo enumeramos 6 patrones, entre ellos arquitectónicos y de diseño que como sabemos estos patrones son reutilizables y están diseñados específicamente para dar solución a problemas de la arquitectura de software [31]. Blockchain es visto como un sistema grande, donde los patrones cumplen un rol fundamental por lo que nos pueden ayudar a obtener un diseño y arquitectura robusta de aplicaciones o sistemas basados en Blockchain. Es por ello que los 6 patrones descritos son los que consideramos más útiles y de los cuales logramos tener mayor información. Este catálogo no pretende ser exhaustivo, pero si tuvimos en cuenta varios aspectos, tales como: Nombre, Tipo, contexto, problema, solución, diagrama arquitectural de la solución, usos conocidos, tácticas, atributos de calidad, compensaciones, consecuencias y finalmente un ejemplo de su aplicabilidad.

Al definir los patrones somos conscientes de que se pueden generar variaciones en su aplicabilidad, ya que estos no consideran preposiciones del todo o nada, dichas especificaciones de patrones generalmente son estrictas pero sabemos que los arquitectos en la práctica pueden optar por hacer pequeñas variaciones que podrían ocasionar otras compensaciones que sean de su beneficio, es por ello, que nuestra propuesta será puesta en marcha, con la finalidad de tener una retroalimentación en cuanto al uso estricto de la definición de los patrones y las variaciones que se puedan presentar.

1. **Nombre:** El Nombre del patrón define la esencia del patrón en el contexto de Blockchain de manera sucinta. El nombre es importante debido a que debe hacer parte del vocabulario de Blockchain.
2. **Tipo:** El tipo define si es patrón arquitectónico o de diseño, es importante identificar a que tipo corresponde cada patrón Blockchain para hacer una clasificación adecuada.
3. **Contexto:** Ámbito o escenario en el cual se presenta la necesidad de aplicar el patrón Blockchain.
4. **Problema:** Descripción general de la problemática que pretende solucionar el patrón, es decir, este ítem responde a la pregunta ¿Cuál es la problemática que presenta Blockchain y a la cual se pretende dar solución mediante el uso de patrones arquitectónicos o de diseño?
5. **Solución:** como el patrón da solución a la problemática descrita, en esta parte se describen las tareas y formas para la implementación del patrón, en este ítem respondemos a las siguientes preguntas: ¿Qué hace este patrón arquitectónico?, ¿En qué se basa? Y ¿Cómo soluciona la problemática presentada en el ítem anterior?
6. **Diagrama arquitectural de la solución:** El diagrama arquitectural de la solución es una representación gráfica que define la descripción del patrón Blockchain, en la cual se tienen en cuenta aspectos tales como: componentes y relaciones**.**
7. **Usos conocidos:** Plataformas o herramientas que han hecho uso del patrón Blockchain especificado. Descripción general de cómo lo usaron.
8. **Tácticas:** Nombre de las tácticas asociadas al uso del patrón Blockchain, en este catálogo no se describen debido a que hay un diccionario dedicado a ello.
9. **Atributos de calidad:** Nombre de los atributos de calidad asociados al patrón Blockchain, en este catálogo no se describen debido a que hay un catálogo dedicado a ello.

10. **Compensaciones:** Se definen las compensaciones entre atributos de calidad derivadas de las decisiones arquitecturales, dichas compensaciones pueden ser tanto positivas como negativas.
11. **Consecuencias:** ¿Cómo logra el patrón el objetivo?, ¿Cuáles son las ventajas, inconvenientes y resultados obtenidos al aplicar el patrón?, ¿Qué aspectos de la estructura están prestas a modificaciones?
12. **Ejemplo de aplicación:** En este ítem buscamos dar un ejemplo de aplicación muy concreto, en el que se especifiquen aspectos tales como: dificultades, técnicas, y aspectos a tener en cuenta a la hora de aplicar el patrón Blockchain.

4.1.3.2 Estructura diccionario de tácticas

Las tácticas son decisiones de diseño que influyen tanto positiva como negativamente en los atributos de calidad, además, afectan las respuestas del sistema frente a los estímulos proporcionados, es por ello que se presentan las compensaciones entre atributos, pero en este diccionario de tácticas no hemos considerado dichas compensaciones ya que estas difieren de los patrones que es donde se construyen. En este diccionario se pretende dar una definición orientada a Blockchain de las tácticas que hemos logrado identificar y relacionar frente a cada patrón.

1. **Nombre:** Nombre de la táctica, este nombre se define teniendo como base el libro de Bass et al.[5].
2. **Definición:** Definición oficial del libro de Bass et al.[5] y definición orientada a Blockchain de las tácticas identificadas.

4.1.3.3 Estructura catálogo de atributos de calidad

En esta sección proporcionamos un catálogo de atributos de calidad que está estrechamente ligado a los patrones definidos en el primer catálogo, con este pretendemos proporcionar técnicas de diseño definidas como tácticas las cuales nos permiten alcanzar dichos atributos de calidad y describir algunos aspectos de ellos tales como: Nombre, definición, contexto y un escenario general que nos ayudara a entender mucho mejor el atributo, dicho escenario está compuesto por una fuente de estímulo, un estímulo, artefactos y respuesta.

1. **Nombre:** Nombre del atributo de calidad
2. **Definición:** En la literatura se pueden encontrar diferentes taxonomías y/o definiciones de atributos de calidad debido a que estas se basan en su aplicabilidad, es decir, si tenemos por ejemplo en cuenta la seguridad podríamos decir que un sistema es seguro en cuanto a algunos aspectos o fallas y frágil con respecto a otras. La definición de los atributos de calidad aquí especificados será en lo posible relacionados directamente al funcionamiento de la Blockchain.
3. **Contexto:** Ámbito o escenario en el cual se presenta el atributo de calidad.
4. **Escenario general:**
    i. **Fuente de estímulo:** Fuente que genera el estímulo.
    ii. **Estimulo:** Eventos que se presentan en el sistema.
    iii. **Artefactos:** Parte del sistema al que es aplicado el requisito.
    iv. **Respuesta:** Como responde el sistema al estímulo.

4.2 MODELO CONCEPTUAL

El modelo arquitectónico basado en soluciones Blockchain fue construido teniendo en cuenta varios aspectos como lo son los hallazgos del mapeo sistemático de la literatura y el caso de estudio, el cual comprende la recuperación de la arquitectura de la plataforma Hyperledger Fabric. Todo este conocimiento obtenido se organizó teniendo en cuenta las pautas conceptuales que ofrece el libro de Bass et al.[5]. La estructura del modelo pretende mostrar un catálogo de patrones arquitectónicos, facilitando su comprensión y relaciones asociadas al catálogo de atributos de calidad y diccionario de tácticas. Esto con el objetivo de agilizar la contextualización desde el punto de vista arquitectónico de las bases de la tecnología Blockchain. El modelo comprende los siguientes aspectos presentados en la **Figura 16. Modelo Arquitectónico Blockchain**.

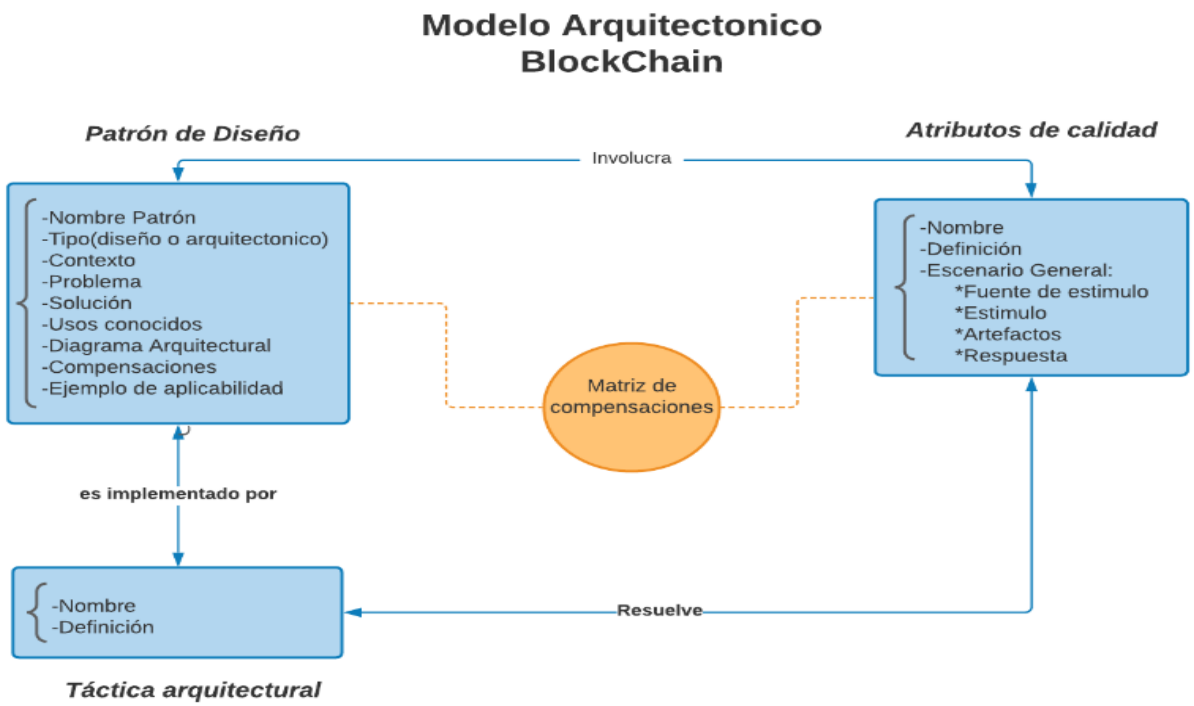


Figura 16. Modelo Arquitectónico Blockchain. Fuente propia

4.2.1 Metodología

Teniendo en cuenta dos enfoques de la tesis que son el mapeo sistemático y el caso de estudio de Hyperledger los cuales nos brindan una base de información importante sobre patrones arquitectónicos, nos dimos a la tarea de describir una metodología que nos ayudara a escoger los patrones más óptimos y con mayor base de información para realizar un buen análisis definición de los mismos y asociación a atributos de calidad y tácticas, es por ello que para la escogencia de patrones definimos dos filtros.

- **Primer filtro:** se tienen en cuenta todos los patrones encontrados basándonos específicamente en la solución de la cuarta pregunta de investigación del mapeo sistemático y hallazgos encontrados mediante la solución del caso de estudio. Cabe destacar que este punto se seleccionan los patrones que tienen mayor base de información.

- **Segundo filtro:** Se hace un cruce de información de los dos enfoques con el fin de encontrar mayor información que respalde los patrones seleccionados en la primera fase, y seleccionamos los patrones que tengan respaldo tanto de la revisión de la literatura como del caso de estudio.

4.3 CATÁLOGO DE PATRONES

1. Patrón 1:

- **Nombre:** Peer to Peer
- **Tipo:** Patrón de Arquitectura
- **Contexto:** Es una de las características más importantes de la red Blockchain, debido a que permite que la información se almacene de forma descentralizada siendo compartida en cada uno de los participantes de la red, esto permite eliminar el factor centralizado de confianza.
- **Problema:** Las transacciones comúnmente se realizan con el siguiente flujo: un usuario requiere realizar una transacción y un servidor que tiene almacenados los datos de las cuentas, recibe la transacción y la procesa. Esta arquitectura puede tener ciertas falencias como lo son: La información se encuentra centralizada, teniendo así un solo punto de fallo, también puede presentar saturación o lentitud en el procesamiento de datos; a su vez existe un solo punto de confianza lo cual puede tener problemas de seguridad si los datos son interceptados.
- **Solución:** La tecnología Blockchain como base de su arquitectura brinda la implementación del patrón arquitectónico peer to peer, el cual aporta a la solución a varios problemas, descentralizando la información y almacenando copias de ella en todos los peers de la red, mejorando varios aspectos como lo son la seguridad de la información, la tolerancia a fallos entre otros.
- **Diagrama arquitectural de la solución:**

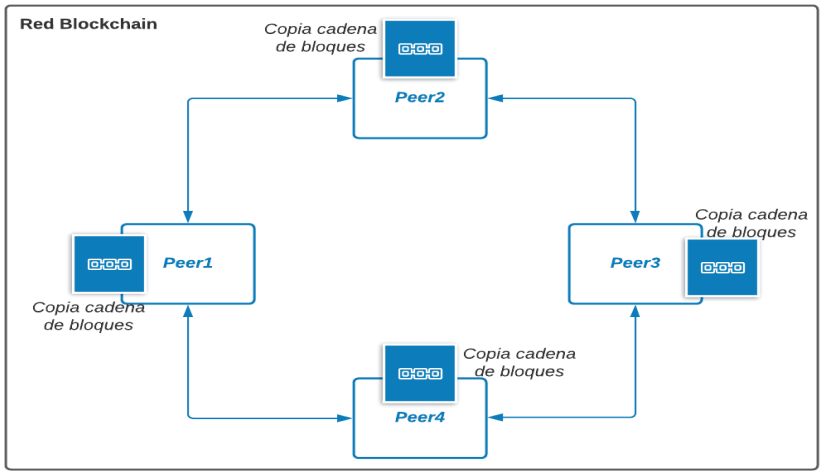


Figura 17. Peer to Peer. Fuente propia

- **Usos conocidos:**

  - **Bitcoin, Ethereum, DodgeCoín:** Estas son algunas de las plataformas que implementan el patrón peer to peer, debido a que todas las plataformas que se encuentran construidas con la tecnología Blockchain lo implementan.

- **Tácticas:**

  - Redundancia activa (Disponibilidad)
  - Resincronización de estados (Disponibilidad)
  - Replicación (Disponibilidad)

- **Atributos de calidad:**

  - Disponibilidad
  - Escalabilidad
  - Seguridad
  - Rendimiento

- **Compensaciones:**

  - **Seguridad vs Rendimiento:** La seguridad de la información con el patrón peer to peer es alta pero afecta el rendimiento al procesar transacciones debido a que requiere que múltiples participantes ejecuten procesos para hacerla válida.
  - **Escalabilidad vs Rendimiento**: La escalabilidad es una característica que se le da muy bien al patrón debido a que se le pueden añadir cuantos nodos deseen, pero esto afecta el rendimiento debido a que entre más amplia la red más latencia puede presentarse retrasando el procesamiento de información.

- **Ejemplo de aplicación**:

  - **Hyperledger fabric:** En Hyperledger fabric se integran los peers que van a hacer parte de la red, se les brinda una copia de los datos del libro mayor y si es necesario también se le asigna un contrato inteligente para el procesamiento de transacciones; si la configuración de la red está por defecto el peer queda conectado a los demás participantes y permite el endoso o consulta de transacciones[51].

2. **Patrón 2:**

- **Nombre:** Oráculo

- **Tipo:** Patrón arquitectónico.

- **Contexto:** Desde la perspectiva de la arquitectura de software Blockchain puede verse como un conector dentro de un gran sistema software, es decir, Blockchain es un conector complejo basado en la red [6], ya que un conector es un mecanismo de interacción para los componentes. Los conectores en sistemas distribuidos son elementos claves que nos permiten lograr atributos de calidad importantes tales como: rendimiento, confiabilidad,

seguridad, entre otros [6]. Para casos generales los diferentes componentes de la arquitectura pueden coordinar sus cálculos a través de la cadena de bloques. Para lograrlo, es posible enviar transacciones a los contratos inteligentes para invocar las funciones que han sido definidas en estos [6], pero en el caso de requerir utilizar Blockchain para casos diferentes a los servicios financieros convencionales, las aplicaciones creadas en Blockchain pueden necesitar de servicios externos.

- **Problema:** El entorno de ejecución de las diferentes Blockchain es autónomo, sólo puede acceder a la información presente en una transacción o en el historial de transacciones de la cadena de bloques [24], estos entornos garantizan la seguridad de la red Blockchain. Los contratos inteligentes se ejecutan en entornos tales como: EVM en el caso de Ethereum y contenedores (Dockers) en el caso de Hyperledger, los cuales no permiten importaciones externas [34], estos solo pueden acceder a la información presente en las cadenas de bloques. Es decir, el estado de los sistemas externos no es directamente accesible a los contratos inteligentes, sin embargo, estos contratos inteligentes en ocasiones necesitan acceder a dichos estados externos.

- **Solución:** Como se ha mencionado, Blockchain por sí solo no permite realizar la conexión al mundo externo, es por ello que para abordar esta limitación se introduce un nuevo componente denominado oráculo. El oráculo permite agregar condiciones que no se pueden expresar en contratos inteligentes ejecutados en el entorno de Blockchain, los oráculos son fuentes de datos confiables, para proporcionar estados externos sobre el mundo real en forma de una transacción ya que cualquier información que no es generada dentro de la cadena mediante una transacción debe ser incluida como datos adjuntos a una transacción de manera segura y confiable [34], cuando la validación de una transacción depende del estado externo, se solicita al oráculo que verifique el estado externo, esto podría bloquear el procesamiento de la transacción hasta que el oráculo de validación que posee una o varias de las direcciones predefinidas verifique una condición sobre el estado externo[3]. Dicho oráculo también se puede implementar dentro de una red Blockchain como un contrato inteligente con un estado externo que se inyecta periódicamente en el oráculo por fuera de la cadena.

- **Diagrama Arquitectural de la solución:**

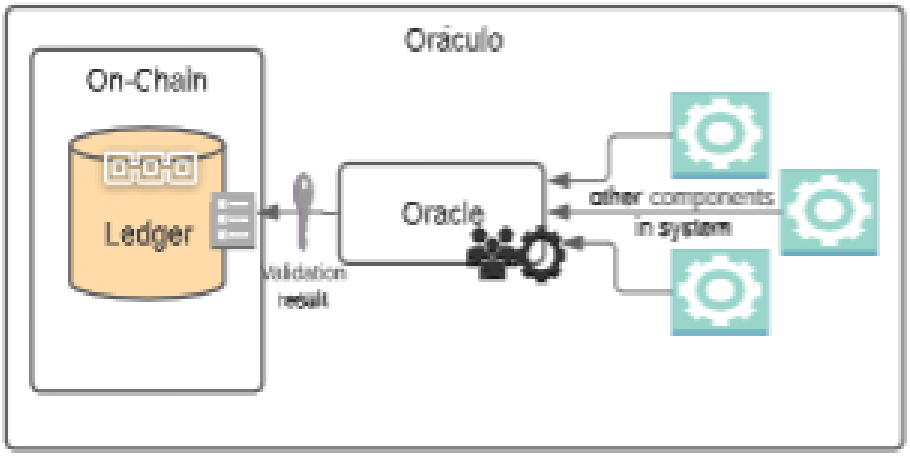


Figura 18: Oráculo
Basado X. Xu et al.[31]

- **Usos conocidos:**

  - **Oracle**: En Bitcoin un Oráculo de validación automatizado puede implementarse como un servidor fuera de la red Blockchain, el cual tiene su propio par de claves[6].
  - **Orisi**: Permite a los participantes involucrados en un contrato seleccionar un conjunto de Oracle en el que se sientan cómodos antes de usar dicho contrato para luego firmar un contrato que requiere un cierto número de firmas de validación de Oracle[6].

- **Tácticas:**

  - Separar entidades (Seguridad).
  - Verificar mensajes de integridad (Seguridad).

- **Atributos de calidad:**

  - Disponibilidad.
  - Interoperabilidad.
  - Seguridad.
  - Confiabilidad.

- **Compensaciones:**

  - **Seguridad vs Interoperabilidad:** Como los mineros no pueden validar por completo el estado externo de la información deben confiar en el Oráculo, con lo cual se introduce un tercero de confianza que afectaría la seguridad del sistema, pero sería más interoperable al facilitar la conexión con el mundo externo.

- **Ejemplo de aplicación:**

  - Un ejemplo de aplicación se puede dar al utilizar un servicio de terceros independiente (oráculo) para conectar el entorno de ejecución de Blockchain con el mundo externo. El oráculo podría consultar y validar los datos externos necesarios para una función de contrato inteligente y luego enviar esos datos a dicho contrato inteligente mediante una transacción. Es decir, cuando la validación de una transacción depende del estado externo, se solicita al oráculo que verifique el estado externo e inyecte el resultado en la cadena de bloques en una transacción firmada. Dichos datos proporcionados por el oráculo son confiables para el contrato inteligente [6].

3. Patrón 3:

- **Nombre:** Intermediario de Confianza.

- **Tipo:** Patrón arquitectónico.

- **Contexto:** Muchas veces en los sistemas basados en Blockchain es necesario restringir transacciones o comunicación entre participantes de la red, con el fin de garantizar la

privacidad de información entre las partes interesadas en ella y restringirla del resto de participantes.

- **Problema:** Las plataformas Blockchain privadas requieren variadas formas de privacidad de la información, normalmente con el fin de brindar seguridad y privacidad a redes de participantes. Por ejemplo, se puede requerir que cierta información no sea conocida por todos los participantes por cuestión de privacidad.

- **Solución:** En las plataformas Blockchain privadas, para restringir el acceso y manipulación de la información se hace uso de un patrón arquitectónico basado canales de comunicación, el cual tiene un comportamiento muy similar al patrón publicador subscriptor[52], estos canales permiten transportar información y a la vez aislarla de los participantes que no se encuentren conectados al canal, este patrón tiene una capa de seguridad opcional para el ofuscamiento de datos que consiste en hacer uso del cifrado de datos [51] al igual que las Blockchain públicas, ofreciendo una capa adicional de seguridad, porque además de que los participantes estén conectados al canal, deben tener la llave para poder descifrar la información.

- **Diagrama Arquitectural de la solución:**

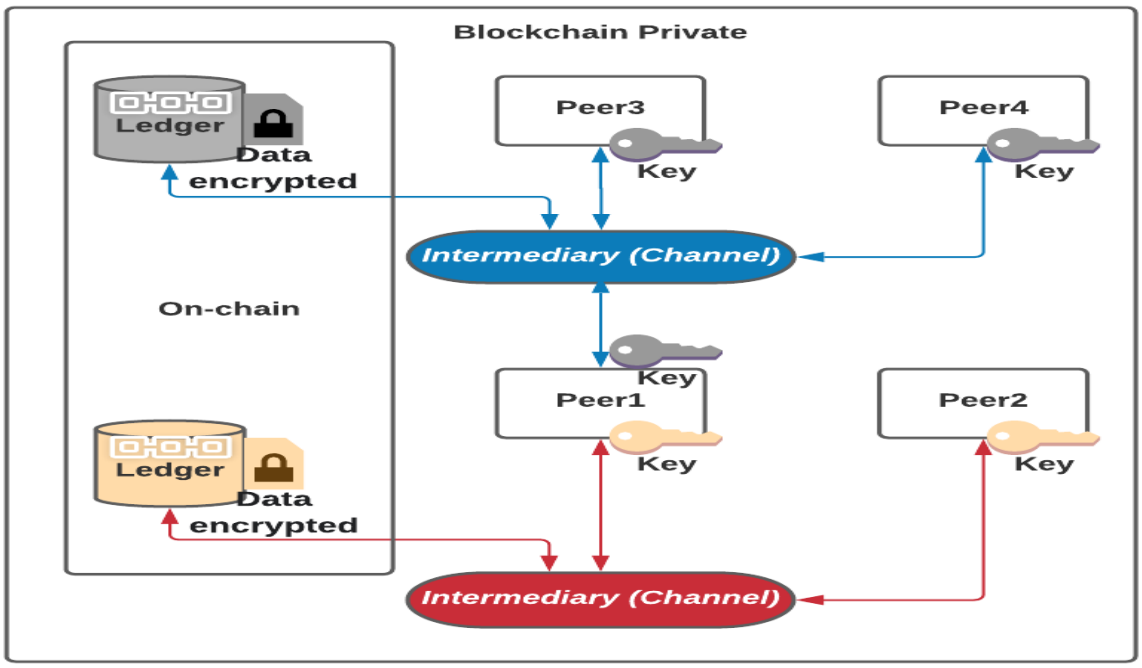


Figura 19. Intermediario de Confianza. Fuente propia

- **Usos conocidos:**

  - **HyperLedger**: En Hyperledger se le llama canal a este intermediario

- **Tácticas:**

  - Utilizar intermediario (Modificabilidad)
  - Cifrar datos (Seguridad)

- **Atributos de calidad:**

  - Seguridad.
  - Escalabilidad.
  - Interoperabilidad.
  - Rendimiento.

- **Compensaciones:**

  - **Desempeño vs Seguridad:** Cuando se aplica el patrón intermediario de confianza aumenta la seguridad, permitiendo el acceso a la información sólo a los participantes de la red que se encuentren conectados a él. Pero esta operación tiene un costo de procesamiento lo cual se ve reflejado en el desempeño produciendo mayores tiempos de respuesta.
  - **Interoperabilidad vs Seguridad:** Este patrón nos brinda la facilidad de separar organizaciones dentro de una red Blockchain, pero también permite que las organizaciones sean interoperables brindando la posibilidad de conectar un par a más de un canal, esto aumenta la interoperabilidad entre organizaciones, pero se puede ver afectada la seguridad al compartir datos privados de una organización a otra.

- **Ejemplo de aplicación:**

  - **Hyperledger Fabric:** Este framework brinda la posibilidad de implementar el patrón por medio de canales, donde se conectan los participantes de la red, el cual también sirve de puente para comunicar diferentes organizaciones.

Patrón 4:

- **Nombre:** Almacén Off-Chain

- **Tipo:** Patrón de Arquitectura

- **Contexto:** Una de las principales decisiones de diseño que debemos tener en cuenta al usar una cadena de bloques es el almacenamiento de información dentro y fuera de la cadena [39][3].

- **Problema:** Blockchain garantiza la integridad de los datos y es por ello que algunas aplicaciones toman la decisión de incluirla en sus diseños e implementaciones, pero debido a la replicación completa de los datos, Blockchain tiene una capacidad límite de almacenamiento por bloque, es por ello que las aplicaciones deben tener en cuenta un equilibrio apropiado de los datos que se incluyen en la cadena y los que quedan fuera de la cadena [6][24].(Ejemplo, en Ethereum el tamaño máximo de un bloque es de 30 millones de gas, y en Hyperledger Fabric el tamaño máximo es de 1 mega). Además, Blockchain no garantiza la integridad e inmutabilidad de los datos que se encuentran fuera de la cadena.

- **Solución:** La Seguridad de los datos fuera de la cadena es garantizado mediante la utilización de un valor hash que es almacenado en la cadena, dicho valor hash es único para cada fragmento de datos, es decir si los datos almacenados fuera de la cadena son

modificados incluso si es solo un bit el valor hash de este nuevo fragmento de datos va a ser muy diferente al almacenado en cadena, es de esta manera que se garantiza la integridad de los datos fuera de la cadena [36][37] y la integridad del valor hash agregado en cadena es garantizado por Blockchain [37][40]. Una técnica común muy utilizada es almacenar los metadatos en cadena y los datos grandes y privados mantenerlos fuera de la cadena [6].

- **Diagrama arquitectural de la solución:**

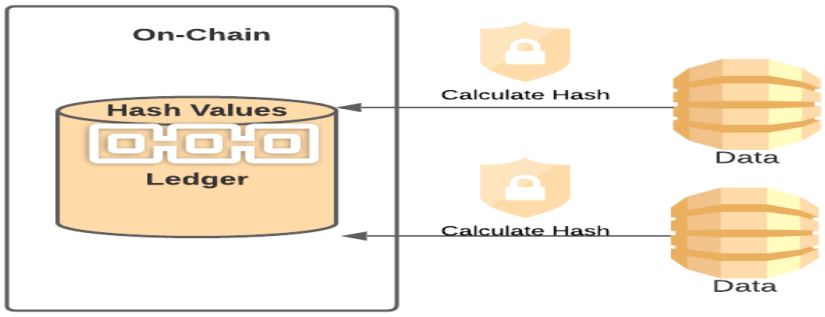


Figura 20: Almacén Off-Chain
Basado X. Xu et al.[31]

- **Usos conocidos:**

  - **Siaicoin, Storj y Filecoin:** Son plataformas que permiten a los participantes hacer uso del almacenamiento fuera de cadena de archivos de Blockchain de manera encriptada y replicada [36].

- **Tácticas:**

  - Verificar la integridad del mensaje (Seguridad)
  - Aumentar los recursos (Desempeño)

- **Atributos de calidad:**

  - Integridad
  - Inmutabilidad
  - Escalabilidad
  - Seguridad
  - Rendimiento

- **Compensaciones:**

  - **Seguridad vs integridad:** Como los datos sin procesar se almacenan fuera de la cadena, se pueden eliminar o perder. Solo su valor hash permanece permanentemente en la cadena de bloques [34], lo que generaría un agujero de seguridad.
  - **Seguridad vs inmutabilidad:** Como los datos sin procesar se almacenan fuera de la cadena, se puede cambiar sin conciencia de la Blockchain. Solo su valor hash es inmutable en la cadena de bloques [34], con lo que sólo se puede garantizar en forma limitada si se ha corrompido la inmutabilidad de la información fuera de la cadena.

- **Seguridad y Escalabilidad**: Como los datos sin procesar se encuentran fuera de cadena, la escalabilidad de la aplicación aumenta, pero de la misma forma disminuye la seguridad de dichos datos, esto es debido a que la información se puede perder si se realizan modificaciones de los datos fuera de cadena ya que el valor hash no corresponde con el que se encuentra almacenado en cadena.

- **Ejemplo de aplicación**

  - **Hyperledger Fabric:** En Hyperledger Fabric se implementan funciones encargadas de categorizar por tamaño los datos que llegan a formar parte de una transacción; los datos que excedan el tamaño de bloque establecidos serán procesados de la siguiente manera: se calcula el hash que se van a escribir en el libro mayor, y los datos se almacenan en una base de datos compatible con los protocolos de esta plataforma como los son *CouchDB* y *StateDB*[51].

Patrón 5:

- **Nombre:** Múltiple Autorización

- **Tipo:** Patrón arquitectónico.

- **Contexto:** En las aplicaciones basadas en Blockchain, algunas actividades (representada por una transacción) podrían basarse en múltiples autoridades [37][3], es decir, deben ser autorizadas por múltiples direcciones de Blockchain [3]. Por ejemplo, una transacción monetaria puede requerir la autorización múltiple de direcciones Blockchains.

- **Problema:** La principal problemática está relacionada a la disponibilidad de las autoridades, es decir que las direcciones que autorizan una actividad (Representada por una transacción) [3][37] no puedan decidirse o ponerse de acuerdo debido a la disponibilidad de las autoridades. Otra variante se puede expresar en términos de recuperación de claves perdidas. Al ser Blockchain un entorno descentralizado no contamos con un proveedor, es decir, un sistema centralizado que nos permita recuperar contraseñas, por tal motivo el usuario es el único encargado de recordar su par de claves (pública y privada). Si el usuario pierde las claves , esto implica la pérdida permanente del control sobre una cuenta y a su vez de los contratos inteligentes [3].

- **Solución:** El mecanismo de múltiples autoridades permite al participante controlar los contratos inteligentes utilizando más de una dirección de Blockchain, con esto se logra reducir el riesgo de perder el control o dominio sobre sus contratos inteligentes en caso de perder la clave privada o comprometida [3]. Los servicios de múltiples autoridades se centran en transacciones, que necesitan ser autorizadas por múltiples direcciones de Blockchain [37] es por ello, que para permitir más dinamismo en el sistema el conjunto de direcciones para la autorización no se decide antes de que se envié la transacción a la red Blockchain o se implemente el contrato inteligente. Como en este patrón hay múltiples autoridades en una red Blockchain, este puede proporcionar flexibilidad para lograr una mejor cooperación [43]. Una transacción es válida solo cuando hay suficientes firmas de las autoridades y además podemos considerar este patrón como un mecanismo que nos permite salvaguardar nuestra clave privada o comprometida, ya que como se menciona anteriormente Blockchain no nos proporciona esta funcionalidad [43]. En Bitcoin, la autorización múltiple o firma múltiple

requiere más de una clave privada para autorizar una transacción y en el caso de Ethereum el mecanismo de autorización múltiple se implementa como un contrato inteligente, específicamente una firma múltiple M de N que define M de N claves públicas son necesarias para autorizar una transacción [3]. Con este método si hay N direcciones de autoridad y el umbral es M, entonces, al menos M claves privadas entre las N claves privadas deben mantenerse de manera segura, para de esta forma evitar perder el control [37].

- **Diagrama arquitectural de la solución:**

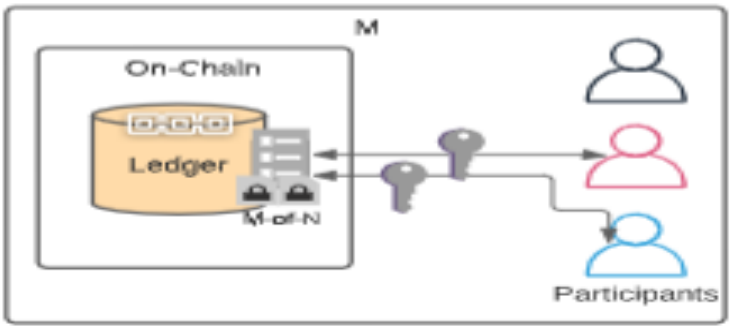


Figura 21: Múltiple Autorización
Basado X. Xu et al. [31]

- **Usos conocidos:**
  - **Multisignature:** Es un mecanismo de firma múltiple proporcionado por Bitcoin. La cual requiere que todos los participantes incluidos estén de acuerdo para que pueda ocurrir cualquier transacción.
  - **Multisignature wallet:** Es un mecanismo de firma múltiple, escrito en Solidity, el cual es ejecutado en Ethereum

- **Tácticas:**
  - Autenticar actores (Seguridad)
  - Redundancia activa (Disponibilidad)

- **Atributos de calidad:**
  - Seguridad
  - Disponibilidad

- **Compensaciones:**
  - **Seguridad y Disponibilidad:** a mayor seguridad vamos a necesitar incrementar la disponibilidad de los nodos y por lo tanto se necesita mejorar ese aspecto, a nivel físico se necesita tener más infraestructura.

- **Ejemplo de aplicación:**
  - **Hyperledger Fabric:** Este framework, implementa el patrón de múltiple autorización, con una política de endoso, donde establece la cantidad de direcciones Blockchain que deben aprobar la transacciones para que sea válida; la aprobación se hace simulando que

la transacción se realizara, si la prueba es superada cada dirección Blockchain devuelve la transacción con la firma digital, lo que quiere decir que está respaldada por esa dirección.

Patrón 6:

- **Nombre:** Autorización de participantes

- **Tipo: Patrón arquitectónico**

- **Contexto:** Las aplicaciones basadas en Blockchain (públicas o privadas) requieren de una gestión de permisos para autorizar algunas actividades y a los participantes dentro de la red [24], dichos participantes son desconocidos al realizar la primera transacción.

- **Problema:** Un contrato inteligente consiste en lo siguiente: múltiples autoridades, enlace dinámico y permisos integrados. Como se mencionó anteriormente en las aplicaciones basadas en Blockchain se requiere una gestión de permisos para autorizar algunas actividades, pero dicha autoridad puede ser desconocida cuando se envía una primera transacción a Blockchain, al implementar el contrato inteligente correspondiente o cuando la transacción es enviada a la cadena de bloques [37]. Blockchain utiliza firma digital para la autenticación y la respectiva autorización de la transacción [36], además de esto Blockchain no admite el enlace dinámico en situaciones donde la dirección de un participante no se encuentra definida inicialmente en el contrato inteligente o la respectiva transacción [37], es decir, para autorizar una segunda transacción todos los participantes deben definirse en la primera transacción antes de que esta sea agregada a la cadena de bloques.

- **Solución:** Si un participante no se encuentra definido inicialmente en el contrato inteligente una vez se realiza la primera transacción se puede usar un secreto (clave hash) fuera de la cadena lo cual permite habilitar la autorización dinámica lo que significa que el acceso a los datos se encuentra protegido por una clave hash que es enviada junto con la información que contenga el contrato inteligente y quien contenga dicha clave o secreto fuera de la cadena podrá obtener o adquirir los datos de la información que contiene el contrato inteligente [37]. Cabe destacar que una vez revelado el secreto este no va a poder ser usado en otras transacciones y si múltiples transacciones se encuentran bloqueadas con el mismo secreto, al desbloquear una se desbloquean las demás.

- **Diagrama arquitectural de la solución:**

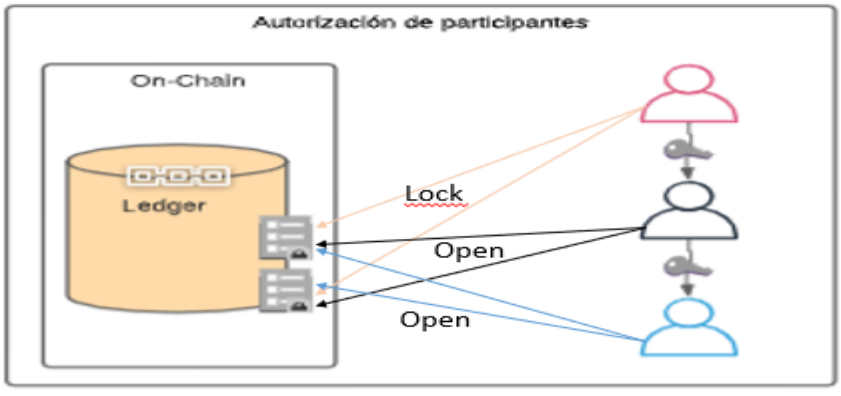


Figura 22: Autorización de participantes

Basado X. Xu et al.[31]

- **Usos conocidos:**
    - **Red Raiden:** Es una red de canales de pago fuera de cadena que permite realizar transferencias seguras y es una solución de escalamiento, permitiendo así pagos escalables, rápidos y a costos muy bajos, Este es complementario a Ethereum [42].
- **Tácticas:**
    - Cifrar Datos (Seguridad)
- **Atributos de calidad:**
    - Interoperabilidad
    - Seguridad
- **Compensaciones:**
    - **Seguridad vs interoperabilidad:** La interoperabilidad que permite el patrón influye en la seguridad del contenido del contrato inteligente ya que al permitirnos compartir el secreto fuera de cadena, este queda expuesto y Blockchain no puede garantizar su seguridad.
- **Ejemplo de aplicación:**
    - **Hyperledger Fabric:** En esta plataforma el uso del patrón se aplica utilizando una entidad certificadora, la cual es la encargada de proporcionar una firma digital a cada participante de la red, para poder llevar a cabo cualquier operación dentro de ella. Esta entidad certificadora se encuentra aislada en un cluster de servidores los cuales comparten la misma base de datos para realizar un seguimiento de las identidades y certificados. Se conecta a la arquitectura de Hyperledger mediante Api Rest o SDK [51].

4.4 CATALOGO DE ATRIBUTOS DE CALIDAD

- **Integridad:** Según Bass et al.[5] La integridad es la propiedad de que los datos o servicios no están sujetos a manipulación no autorizada. Teniendo en cuenta esto podemos decir que una de las principales características de Blockchain es la integridad ya que se protege en contra de la modificación y/o eliminación de la información almacenada en red, asegurando la autenticidad de la información y el no repudio. Además, al contar con otras características tales como la trazabilidad e inmutabilidad de los datos se puede garantizar dicha integridad.
    - **Escenario general:**
        - **Fuente de estímulo:** Aplicación cliente
        - **Estímulo:** Una transacción que ingresa a la red Blockchain
        - **Artefactos:** Blockchain
        - **Ambiente:** Ejecución

        - **Respuesta:** Se guarda el bloque en cadena y se actualiza el estado a todos los participantes, manteniendo la integridad de los datos que se encontraban inicialmente almacenados y los nuevos, en el libro mayor.
        - **Medida de la respuesta:** El esfuerzo que requerirá el atacante será de X [personas-mes] (X depende del nivel de integridad esperado, en una Blockchain pública como bitcoin X tiende a infinito).

- **Seguridad:** Según Bass et al.[5] La seguridad es una medida de la capacidad del sistema para proteger los datos y la información de accesos no autorizados a la vez que proporciona acceso a personas y sistemas autorizados. La seguridad está estrechamente ligada a la confidencialidad de la información, es por ello que algunas medidas de seguridad en Blockchain son: Proteger contra el acceso no autorizado (Blockchains privadas) y la encriptación de datos (Blockchains públicas).

    - **Escenario general:**
        - **Fuente de estímulo:** Atacante, usuario malicioso o usuario no autorizado
        - **Estímulo:** Ataque o Intento no autorizado de acceder a información
        - **Artefactos:** Blockchain
        - **Ambiente:** Ejecución
        - **Respuesta:** No permitir la obtención de información relevante y confidencial del sistema.
        - **Medida de la respuesta:** La probabilidad de que un usuario no autorizado, malicioso o atacante obtenga información relevante debe ser inferior a X (X depende del nivel de seguridad esperado).

- **Disponilidad:** Según Bass et al. [5] La disponibilidad se refiere a una propiedad de software que está allí y lista para llevar a cabo su tarea cuando la necesita. Para el caso de Blockchain en específico la disponibilidad está muy ligada a la información que se encuentra en la red y en quienes pueden acceder a dicha información dependiendo de la Blockchain (públicas o privadas).

    - **Escenario general:**
        - **Fuente de estímulo:** Información
        - **Estímulo:** Disponibilidad de la información
        - **Artefactos:** Blockchain
        - **Ambiente:** Ejecución
        - **Respuesta:** En Blockchains públicas la información se encuentra disponible para cualquier participante de la red, mientras que en Blockchains privadas la disponibilidad de esta información está basada en permisos establecidos que generalmente es dada en grupos.
        - **Medida de la respuesta:** Tiempo medio entre fallos, tiempo recuperación de un fallo, alfa = tmf/(tmf + tr)

- **Interoperabilidad:** Según Bass et al.[5] La interoperabilidad se refiere al grado en que dos o más sistemas pueden intercambiar información útil a través de interfaces en un contexto particular. Si bien al momento no se ha logrado la interoperabilidad entre sistemas Blockchain como podría ser Bitcoin y Ethereum, existen métodos o patrones que nos permiten conectar con el mundo exterior como lo es la interacción con bases de datos externas cuando se requiere almacenar información que supera el tamaño del bloque (Almacén Off-Chain),

también se evidencia en la aplicación del patrón arquitectónico oráculo, entre otros. De esta manera se ve la interoperabilidad reflejada en la Blockchain.

- **Escenario general:**
  - **Fuente de estímulo:** Un sistema realiza una solicitud para interoperar con el otro
  - **Estímulo:** Una solicitud para intercambiar información entre los sistemas que interoperan
  - **Artefactos:** Sistemas que deseen interoperar
  - **Ambiente:** Desarrollo
  - **Respuesta:** Intercambio de información entre los sistemas
  - **Medida de la respuesta:** Esfuerzo de poner a interoperar la solución con nuevo sistema centralizado inferior a X personas-mes (Entre menos esfuerzo dedicado a la integración, es más interoperable)

- **Confiabilidad:** Según Bass et al.[5] La confiabilidad es la capacidad de evitar fallas que son más frecuentes y más graves de lo que es aceptable. La tecnología Blockchain maneja un porcentaje de confiabilidad alto debido a muchos factores, uno de ellos es su principio descentralizado lo que permite que las operaciones se ejecuten en varios nodos eliminando el único punto de entrada y de fallo, dando la posibilidad de que la mayoría de transacciones se ejecuten sin problemas en los tiempos determinados.
  - **Escenario general:**
    - **Fuente de estímulo:** Aplicación cliente
    - **Estímulo:** Una transacción o consulta ingresadas a la red Blockchain en determinados periodos de tiempo
    - **Artefactos:** Blockchain
    - **Ambiente:** Ejecución
    - **Respuesta:** La transacción o consulta se lleva a cabo exitosamente.
    - **Medida de la respuesta:** Probabilidad de que un sistema no tenga fallos al procesar la transacción o consulta solicitada en un periodo de tiempo.

- **Inmutabilidad:** Se refiere a la capacidad de un sistema para mantener sus datos sin ninguna alteración después de ser almacenados.

  - **Escenario general:**
    - **Fuente de estímulo:** Aplicación cliente
    - **Estímulo:** Solicitud maliciosa o no autorizada de manipulación de bloques
    - **Artefactos:** Blockchain
    - **Ambiente:** Ejecución
    - **Respuesta:** Los bloques de la red Blockchain permanecen en su estado original no se permite su modificación.
    - **Medida de la respuesta:** La probabilidad de que un usuario no autorizado, malicioso o atacante modifique la información de la cadena de bloques, debe ser inferior a X (X depende del nivel de inmutabilidad esperado).

- **Escalabilidad:** Se refiere a la capacidad de un sistema para aumentar la capacidad de procesamiento sin tener que ser modificado (más allá de la replicación en la instalación).

  - **Escenario general:**

        - **Fuente de estímulo:** Añadir pares a la red
        - **Estímulo:** Crecer una red Blockchain en el número de clientes y de flujo de transacciones en un Y por ciento.
        - **Artefactos:** Red de nodos
        - **Ambiente:** Despliegue
        - **Respuesta:** La red Blockchain es capaz de integrar nuevos nodos a la red sin modificar la solución.
        - **Medida de la respuesta:** Esfuerzo E, tiempo T

- **Rendimiento:** Se trata del tiempo y la capacidad del sistema de software para cumplir con los requisitos de tiempo. Cuando ocurren eventos, el sistema, o algún elemento del sistema, debe responder a ellos a tiempo. El rendimiento es un atributo variante en la tecnología Blockchain, debido a que se ve afectado por varios factores, aun así, esta tecnología aplica diferentes decisiones arquitecturales como lo son el procesamiento de datos en paralelo, el almacenamiento de transacciones por bloques, entre otros, esto con el fin de satisfacer lo mejor posible este atributo.

    - **Escenario general:**
        - **Fuente de estímulo:** Aplicación cliente
        - **Estímulo:** Añadir una transacción a la Blockchain
        - **Artefactos:** Blockchain
        - **Ambiente:** Ejecución
        - **Respuesta:** La red Blockchain es capaz de almacenar la transacción en un tiempo determinado
        - **Medida de la respuesta:** Tiempo T

4.5 DICCIONARIO DE TÁCTICAS

Las definiciones proporcionadas a continuación están basadas en las descripciones encontradas en el libro de Bass et al.[5]. Con el fin de complementar dichas definiciones decidimos darle un enfoque hacia Blockchain, proporcionando una descripción breve de cuál es la manera en que se puede utilizar cada una de las tácticas en dicha tecnología.

- **Tácticas de seguridad:**

    - **Separar entidades:** La separación de diferentes entidades dentro del sistema se puede hacer a través de la separación física en diferentes servidores que están conectados a diferentes redes; el uso de máquinas virtuales o un "espacio de aire", es decir, al no tener conexión entre diferentes partes de un sistema. Finalmente, los datos confidenciales se separan con frecuencia de los datos no sensibles para reducir las posibilidades de ataque de aquellos que tienen acceso a datos no sensibles. Una de las formas en que se hace uso de esta táctica en Blockchain es mediante un patrón denominado oráculo el cual nos permite hacer la separación física de los datos, es decir, nos permite acceder a información del mundo externo. Con Oracle estamos permitiendo que Blockchain se extienda a más aplicaciones, se amplía el espectro de aplicación de la Blockchain.

- **Verificar mensajes de integridad:** Esta táctica emplea técnicas como sumas de comprobación o valores hash para verificar la integridad de los mensajes, archivos de recursos, archivos de implementación y archivos de configuración. Una suma de comprobación es un mecanismo de validación en el que el sistema mantiene información redundante para los archivos y mensajes de configuración, y utiliza esta información redundante para verificar el archivo o mensaje de configuración cuando se usa. Un valor de hash es una cadena única generada por una función de Hashing cuya entrada podría ser archivos de configuración o mensajes. Incluso un ligero cambio en los archivos o mensajes originales produce un cambio significativo en el valor de hash. Partiendo de esta descripción podemos darnos cuenta que el patrón oráculo cumple con esta definición ya que el oráculo se encarga de la verificación de la información del estado externo y este proporciona el resultado al validador(minero), quien toma en cuenta el resultado proporcionado por el oráculo al validar la transacción. Además de eso, las Blockchains públicas también hacen uso de esta táctica mediante el cifrado de datos, con este método solo quien tiene la clave privada puede tener acceso a la información, en caso de las Blockchains privadas tienen otros métodos tales como canales.

- **Cifrar datos:** Los datos deben estar protegidos del acceso no autorizado. La confidencialidad se logra usualmente aplicando algún tipo de cifrado a los datos y a la comunicación. El cifrado proporciona protección adicional a los datos mantenidos de forma persistente más allá de los disponibles a partir de la autorización. Los enlaces de comunicación, por otro lado, pueden no tener controles de autorización. En tales casos, el cifrado es la única protección para pasar datos a través de enlaces de comunicación de acceso público. El enlace puede ser implementado por una red privada virtual (VPN) o por un Secure Sockets Layer (SSL) para un enlace basado en la web. El cifrado puede ser simétrico (ambas partes usan la misma clave) o asimétrico (claves públicas y privadas). Esta es una de las tácticas más recurrentes en Blockchain ya que el cifrado de los datos o hashes criptográficos se utiliza para evitar el acceso no autorizado a la información disponible en la cadena de datos, además también es muy común que se utilice por ejemplo en contratos inteligentes para que el participante no definido desde la primera transacción y que contenga dicho secreto pueda acceder a la información contenida en el contrato.

- **Autenticar Actores:** Autenticación significa asegurarse de que un actor (un usuario o una computadora remota) sea realmente quién o lo que pretende ser. Las contraseñas, las contraseñas de un solo uso, los certificados digitales y la identificación biométrica proporcionan un medio para la autenticación. En Blockchains públicas existe un patrón que permite la múltiple autorización, es decir la autenticación de múltiples autoridades permitiendo así más control sobre los contratos inteligentes y transacciones, a su vez en Blockchains privadas también tenemos aspectos tales como los canales, que nos permiten tener privacidad de la información ya que solo son accesibles por los participantes autorizados en dicho canal.

- **Tácticas de disponibilidad:**

  - **Redundancia activa:** Esto se refiere a una configuración en la que todos los nodos (activos o de reserva redundantes) en un grupo de protección reciben y procesan entradas idénticas en paralelo, lo que permite que las piezas de reserva redundantes mantengan el estado síncrono con los nodos activos. Debido a que el repuesto

redundante posee un estado idéntico al procesador activo, puede asumir el control de un componente fallido en cuestión de milisegundos. El caso simple de un nodo activo y un nodo de repuesto redundante se conoce comúnmente como redundancia 1 + 1 ("uno más uno"). La redundancia activa también se puede usar para la protección de instalaciones, donde los enlaces de red activos y en espera se utilizan para garantizar una conectividad de red de alta disponibilidad. Esta táctica se aplica en la red Blockchain cuando se ejecutan validación o almacenamiento de transacciones, donde los nodos de la red encargados de esta tarea procesan las peticiones en forma paralela y son capaces de tolerar el fallo de algún nodo.

- **Resincronización:** resincronización de estados es un socio de reintroducción de las tácticas de preparación y reparación de redundancia activa y redundancia pasiva. Cuando se usa junto con la táctica de redundancia activa, la resincronización de estado se produce de manera orgánica, ya que los componentes activos y en espera reciben y procesan entradas idénticas en paralelo. En la práctica, los estados de los componentes activos y en espera se comparan periódicamente para garantizar la sincronización. Esta comparación puede basarse en un cálculo de comprobación de redundancia cíclica (suma de comprobación) o, para sistemas que proporcionan servicios críticos para la seguridad, un cálculo de resumen de mensaje (una función hash de una vía). Cuando se usa junto con la táctica de redundancia pasiva (repuesto dinámico), la resincronización de estado se basa únicamente en la información de estado periódico transmitida desde el (los) componente (s) activo (s) al (los) componente (s) en espera (s), generalmente a través del punto de control. Esta táctica es muy común en la Blockchain debido a que la red se sincroniza, manteniendo el estado actualizado en todos los nodos; en caso de algún fallo y una posterior recuperación la táctica es capaz de actualizar el estado en el nodo recuperado, manteniendo la red Blockchain en armonía.

- **Replicación:** La replicación es la forma más simple de votar; Aquí, los componentes son clones exactos el uno del otro. Tener múltiples copias de componentes idénticos puede ser eficaz para proteger contra fallas aleatorias de hardware, pero esto no puede proteger contra errores de diseño o implementación, en hardware o software, porque no hay ninguna forma de diversidad incorporada en esta táctica. Esta táctica se aplica en la Blockchain cuando se agregan nodos a la red, donde cada nodo posee una copia del estado, permitiendo la descentralización, esto se traduce en que todos los nodos de la red son clones y algunos tienen tareas adicionales según la configuración de la red.

- **Tácticas de desempeño**

  - **Aumentar los recursos:** Los procesadores más rápidos, los procesadores adicionales, la memoria adicional y las redes más rápidas tienen el potencial de reducir la latencia. El costo generalmente es una consideración en la elección de recursos, pero aumentar los recursos es definitivamente una táctica para reducir la latencia y, en muchos casos, es la forma más económica de obtener una mejora inmediata. Esta táctica en la Blockchain se ve reflejada en varios aspectos, uno de ellos es cuando se requiere manejo de datos que superan el tamaño del bloque o se encuentran en el límite, se recurre a almacenar los datos en una persistencia externa y referenciar lo que se almacenó; otro punto de aplicación es aumentar los recursos de forma equitativa en los nodos que conforman la red para reducir la latencia y tener una red equilibrada.

- **Tácticas de modificabilidad:**

  - **Utilizar un intermediario:** Dada una dependencia entre la responsabilidad A y la responsabilidad B (por ejemplo, llevar a cabo A primero requiere llevar a cabo B), la dependencia se puede romper utilizando un intermediario. El tipo de intermediario depende del tipo de dependencia. Por ejemplo, un intermediario de publicación/ suscripción eliminará el conocimiento del productor de datos de sus consumidores. También lo hará un repositorio de datos compartido, que separa a los lectores de una parte de los datos de los escritores de esos datos. En una arquitectura orientada a servicios en la que los servicios se descubren entre sí mediante una búsqueda dinámica, el servicio de directorio es un intermediario. Esta táctica se refleja más que todo en las Blockchain con autorización o privadas, las cuales implementan la táctica para proteger la información de nodos que no sean requeridos, utilizando un intermediario que se encarga de conectar los nodos interesados a un canal para la transferencia de datos.

4.6 GUÍA DE APLICACIÓN MODELO ARQUITECTÓNICO BLOCKCHAIN

La guía de MAB (Modelo Arquitectónico Blockchain) tiene como objetivo proveer a las personas interesadas en trabajar con la tecnología Blockchain, una orientación de la aplicabilidad del modelo, por medio de una serie de 11 pasos propuestos, los cuales se obtuvieron y se refinaron en base de los métodos arquitecturales QAW y ADD [49][50]. A continuación, se presenta la guía de diseño.

Figura 23. Guía de diseño MAB. Fuente propia

- **Paso 1: Establecer Equipo de Diseño**

En este paso se conforma el equipo que va a llevar a cabo el diseño del sistema con base en el MAB, además se presentan los interesados especificando la experiencia que ha desarrollado a lo largo de su carrera y su relación con el sistema que se va a diseñar. Se debe designar uno o más interesados que lideren el diseño del sistema ya que serán los encargados de mantener un flujo correcto además de aportar en la solución de inconvenientes.

- **Paso 2: Presentación del negocio**

Después del paso 1, un integrante del equipo de diseño que tenga claridad sobre el negocio procede a presentar el contexto de este en detalle, especificando las preocupaciones, restricciones, requisitos funcionales, la misión del proyecto y atributos de calidad del sistema; en un periodo máximo de una hora. Los demás integrantes del equipo que escuchan la presentación deben estar atentos a cualquier información relevante que puede dar paso a un requisito de alto impacto que se refinaran en el siguiente paso.

- **Paso 3: Identificar Cualidades Relevantes**

En el paso 2, los interesados capturan información sobre las cualidades relevantes del sistema. En este paso se tiene en cuenta toda la información recopilada por cada uno de los interesados, la cual será unificada en una lista a cargo de los lideres, se espera que esta lista este compuesta por preocupaciones, restricciones requisitos funcionales, atributos de calidad, objetivos y cualquier información considerada relevante. Se socializa la lista generada con todos los interesados del equipo con el fin de refinarla (adicionar, corregir, eliminar) y obtener un enfoque en la lluvia de ideas.

- **Paso 4: Lluvia de Ideas de Escenarios**

Una vez socializada la lista del paso 3, los lideres proceden a guiar a los interesados en la creación de escenarios bien formados, donde como mínimo exista un estímulo un entorno y una respuesta. La idea es que en los escenarios generados exista una representación descrita de cada uno de las cualidades identificadas en el paso anterior.

- **Paso 5: Priorización de Escenarios**

En este paso se realiza la lista de escenarios obtenida en el paso anterior, y los lideres organizan una votación por cada uno de ellos, donde participan todos los interesados del equipo de diseño. Cada interesado tiene un número determinado de votos el cual se es el total de lo que obtengamos del 30 % de los escenarios generados.

- **Paso 6: Consolidación y Refinamiento de Escenarios Priorizados en Base al MAB**

En este paso los lideres documentan de una manera más detallada los escenarios priorizados, teniendo en cuenta la relación que tienen con los artefactos propuestos por el MAB, para documentar con el mayor detalle posible y lograr escenarios bien formados y consolidados, esto se hace teniendo en cuenta los siguientes 7 puntos:

1. Estímulo: la condición que afecta al sistema
2. Respuesta: la actividad que resulta del estímulo
3. Fuente de Estímulo: la entidad que generó el estímulo
4. Ambiente: la condición en la que se produjo el estímulo
5. Atributos: son los atributos relevantes que se ven afectados por el escenario
6. Artefacto Estimulado: el artefacto que fue estimulado
7. Medida de respuesta: medida con la cual se evaluará como el sistema debería responder

Una vez se documentan los puntos anteriores se da a conocer la estructura de cada escenario a los interesados, con el fin de escuchar preguntas, propuestas o sugerencias.

- **Paso 7: Seleccionar tácticas, del diccionario de tácticas del MAB**

En este paso es importante que estén todos los interesados del equipo de diseño con el fin de seleccionar tácticas del diccionario MAB las cuales puedan aplicar en los escenarios priorizados.

- **Paso 8: Seleccionar vistas y patrones del catálogo de patrones MAB**

Una vez seleccionan las tácticas en el paso anterior, se van a crear relaciones con los patrones del catálogo MAB, la tarea de los lideres es dar a conocer a los interesados estos patrones con su estructura metodológica, con el fin de escuchar propuestas, preguntas, mejoras y en base a eso seleccionar los que más se adapten a los escenarios.

- **Paso 9: Instanciar patrón MAB**

Una vez seleccionado los patrones del catálogo MAB, se procede a realizar la instanciación de cada uno de ellos, con el objetivo de satisfacer los escenarios siendo esta la prioridad del negocio. El catálogo provee gran información para llevar a cabo este paso adicional a esto es aconsejable que un líder con amplia experiencia en arquitectura de software sea el responsable de guiarlo debido a que tiene gran peso en los resultados de la solución.

- **Paso 10: Evaluar la solución en Términos de Compensaciones MAB**

En este paso se hace una evaluación de la solución con todo el equipo de diseño esto con el fin de detectar compensaciones descritas en el catálogo de patrones MAB, y se procede a documentar cada una de ellas.

- **Paso 11: Documentar Diseño, Decisiones y Justificación**

Por último, se realiza una retrospectiva de todos los pasos, con el fin de documentar entre todo el equipo el diseño obtenido, las decisiones que se tomaron para llegar a esa solución y justificaciones en las cuales consensuaron para tomar esas decisiones. Se recomienda usar el modelo de vistas para llevar a cabo la documentación formal.

5. ESTUDIOS DE CASO: ARQUITECTURA BLOCKCHAIN

5.1 INTRODUCCIÓN

Para la realización de este proyecto, además de estudiar la literatura, es importante conocer desde lo concreto y en forma empírica acerca de la arquitectura subyacente de soluciones Blockchain. Para esto se desarrollan dos estudios de caso, empírico-descriptivos. El primero orientado a la recuperación y evaluación de la arquitectura de una plataforma Blockchain y el segundo orientado a la evaluación del MAB propuesto.

En ambos estudios se utilizó el método de investigación de estudio de caso en ingeniería de software, siguiendo los lineamientos de Runeson and Host [11]. La investigación en la ingeniería de software mediante estudios de caso tiene como objetivo estudiar en su contexto, aspectos del desarrollo de software, cómo es realizado y las partes interesadas [53].

Finalmente, en la realización de los estudios de caso se siguieron los pasos presentados en la **Figura 24. Pasos generales de un estudio de Caso.**

Figura 24. Pasos generales de un estudio de caso.
Basado [11]

- **ESTUDIO DE CASO 1 RECUPERACIÓN Y EVALUACÍON ARQUITECTONICA DE UNA PLATAFORMA BLOCKCHAIN**

  Holístico, Descriptivo, 1 unidad de análisis.
  En este estudio de caso se propone recuperar y evaluar el modelo arquitectónico de una plataforma Blockchain, desde una perspectiva de los atributos de calidad (y las incumbencias), tácticas arquitectónicas, patrones arquitectónicos, hasta su aplicabilidad en proyectos de desarrollo software reales.

  De acuerdo a la clasificación de Yin et al.[53] el estudio de caso es holístico, puesto que estudia un solo caso de recuperación como un todo en términos de los aspectos de arquitectura que se evalúan, es descriptivo porque busca describir el fenómeno del diseño a través de los resultados plausibles en vistas arquitectónicas y las decisiones de diseño tomadas para llegar a ellas.

- **ESTUDIO DE CASO 2 EVALUACIÓN DE MAB**

Holístico, Descriptivo, 1 unidad de análisis.

En este estudio de caso se propone evaluar el grado de satisfacción hacia los atributos de calidad de un modelo arquitectónico propuesto, aplicado a un proyecto de software real.

De acuerdo a la clasificación de Yin et al. [53]el estudio de caso es holístico con una unidad de análisis, puesto que estudia el modelo propuesto, aplicado a dos soluciones en términos arquitectónicos en dos contextos diferentes, es descriptivo porque busca describir el fenómeno del grado de satisfacción a través de los resultados plausibles en la implementación de las dos soluciones.

A continuación, se detalla el plan, ejecución y reporte de cada estudio de caso

5.2 ESTUDIO DE CASO 1: RECUPERACION Y EVALUACIÓN ARQUITECTONICA DE HYPERLEDGER EN EL CONTEXTO DE UNA SOLUCION EN EL SECTOR SALUD

5.2.1 Antecedentes

De acuerdo a los resultados arrojados por la revisión de la literatura, investigación de la documentación oficial y el respaldo de la comunidad de desarrolladores, las plataformas que responden mejor a las cualidades buscadas por el enfoque Blockchain son: Hyperledger, Ethereum y Parity. Aunque existen muchas más estas resultan ser las más documentadas. Partiendo de esto se selecciona Hyperledger Fabric para este estudio de caso descriptivo. La decisión fue tomada gracias a la investigación tanto en desempeño y escalabilidad, como en la documentación existente de la plataforma y la comunidad que la respalda. En este caso Ethereum se tiene una documentación muy escasa referente a cómo está construida la plataforma y las decisiones que estuvieron involucradas en ello; por otro lado, Hyperledger Fabric es el caso contrario debido a que está respaldada por la fundación Linux la cual tiene un largo y exitoso historial de proyectos de código abierto, ofreciendo una amplia documentación que brinda un panorama más claro de la interacción de sus componentes. Mencionando un poco los resultados obtenidos en la literatura se tienen los siguientes pro y contras de las 3 plataformas seleccionadas: aunque Hyperledger se desempeña consistentemente mejor que Ethereum y Parity en puntos de referencia bajo los cuales se evaluaron, presenta dificultades para escalar a más de 16 nodos; Ethereum y Parity son más tolerantes a las fallas de nodos, pero son vulnerables a los ataques de seguridad que bifurcan la cadena de bloques; Los principales cuellos de botella en Hyperledger y Ethereum son los protocolos de consenso, pero para Parity el cuello de botella es causado por la firma de transacciones; Ethereum y Parity incurren en grandes gastos generales en términos de uso de memoria y disco. Su motor de ejecución (el cual es el mismo en Ethereum y Parity) también es menos eficiente que el de Hyperledger; El modelo de datos de Hyperledger es de bajo nivel, pero su flexibilidad permite una optimización personalizada para consultas analíticas. Por lo anterior y por ser Hyperledger la plataforma más madura tanto en términos de su base de código, base de usuarios y comunidad de desarrolladores que la respaldan, es la plataforma seleccionada para realizar este estudio.

Para desarrollar el caso, se escogió ATAM (Architecture Tradeoff Analysis Method), para evaluar con mayor profundidad, aspectos referentes a la arquitectura, correspondientes a varios atributos de calidad, además está definido en forma clara por un conjunto de pasos importantes [54]. Adicional a esto se recupera parte de su “rationale” a través de la aplicación del método de evaluación de arquitectura EVAR [55] que es un método basado en ATAM para la recuperación

de arquitecturas y que se ajusta a las necesidades del proyecto.

5.2.2 preguntas de investigación

El diseño de investigación parte de varias preguntas ya propuestas al inicio de este documento. Particularmente, para este estudio, la pregunta de investigación del proyecto evidenció la necesidad de identificar las cualidades (y sus incumbencias), tácticas y patrones arquitectónicos de la solución arquitectónica de Hyperledger Fabric. Así la pregunta derivada que para este estudio concreto sobre Hyperledger Fabric busca responder es: ¿Cuáles aspectos arquitecturales, en términos de cualidades, incumbencias, tácticas y patrones arquitectónicos fueron considerados en el modelo arquitectónico de Hyperledger Fabric? ¿Cómo están articulados estos aspectos arquitectónicos al momento de evaluar y aplicar este tipo de tecnología a una solución software para soportar algún proceso de negocio basado en transacciones?

5.2.3 Contexto del caso

En este caso se tiene como objetivo realizar la recuperación y evaluación arquitectónica de una plataforma Blockchain, se inició por el framework Ethereum donde con días de investigación en documentación oficial, y aplicación de herramientas visuales de recuperación, no se obtuvieron buenos resultados debido a la escasez de documentación técnica, donde se buscaba evidencias de la articulación de sus componentes arquitectónicos, además de la incompatibilidad de las herramientas de recuperación visual debido a que esta plataforma se encuentra construida con diversos lenguajes como lo son C++ y Python lo cual dificulta la abstracción visual para las herramientas. Teniendo en cuenta todos los impedimentos, con ayuda de nuestro director se tomó la decisión de investigar el siguiente framework en la lista priorizada de los antecedentes encontrados el cual es Hyperledger Fabric que es una red privada o de autorización creada y mantenida por Linux oundation. Para realizar posteriormente una evaluación sistemática. Durante el estudio de la plataforma Hyperledger Fabric se hace uso de herramientas de ingeniería inversa para la recuperación de la arquitectura de un sistema, las cuales nos proveen vistas arquitectónicas del sistema y su *rationale* se consigue a partir de las fuentes establecidas siguiendo el método EVAR. Posteriormente a la recuperación, se realiza una evaluación objetiva de la arquitectura de Hyperledger Fabric se utiliza el método de evaluación ATAM que tiene como proyecto base un sistema del sector salud enfocado al traslado de pacientes, para esto tenemos un experto en el tema el cual hace parte del equipo de evaluación. Esta metodología está conformada por un conjunto de pasos: presentación, investigación, análisis, testing y reportes; los cuales nos ayudan a elegir una adecuada arquitectura de un sistema software. La idea es que los resultados de ATAM permitieran establecer un conocimiento arquitectónico de Hyperledger para brindar una solución concreta en el sector salud.

5.2.4 Diseño de los indicadores y mediciones

De acuerdo al objetivo del estudio de caso, y las preguntas de investigación fue diseñado el indicador, las métricas e instrumento a emplear, la **Tabla 12. Diseño de Estudio Caso Descriptivo** relaciona estos elementos para el estudio de caso descriptivo.

Objetivo	Preguntas	Indicadores	Aspectos Cualitativos	Instrumentos
Estudiar Hyperledger Fabric como solución arquitectónica	¿Cuáles aspectos arquitecturales, en términos de cualidades, incumbencias, tácticas y patrones arquitectónicos fueron considerados en la solución arquitectónica de Hyperledger Fabric?	AK= Conocimiento de la Arquitectura	A (conjunto de atributos) D(drivers) T(tácticas) P(patrones)	Informe de recuperación Informe de evaluación
	¿Cómo están articulados estos aspectos arquitectónicos al momento de evaluar y aplicar este tipo de tecnología a una solución software para soportar algún proceso de negocio basado en transacciones?	AK+R= Conocimiento de la Arquitectura con rationale	LAT (matriz de trazabilidad atributos X tácticas) MTP (matriz de trazabilidad tácticas X patrones)	Informe de recuperación Informe de evaluación

Tabla 12. Diseño de Estudio Caso Descriptivo. Fuente propia

5.2.5 Desarrollo del caso

Para la ejecución del estudio de caso se plantearon las siguientes tareas basadas en las metodologías: recuperación y evaluación, para obtener una guía de arquitectura para Blockchain. La primera tarea fue:

- Recuperación de la arquitectura (Preparando los insumos para la evaluación de la arquitectura)
    - Objetivos claros para la recuperación de la arquitectura.
    - Explorar y seleccionar la herramienta que apoya la recuperación de las vistas de la arquitectura de Hyperledger Fabric.
    - Recuperar la arquitectura de Hyperledger Fabric, usando las herramientas seleccionadas.
    - Documentar la arquitectura de Hyperledger Fabric.

- Evaluación de la arquitectura:
    - Evaluación a profundidad de Hyperledger Fabric, sobre un sistema orientado en el sector de la salud (véase Anexo A), utilizando el método ATAM.
    - Objetivos claros para la evaluación de la arquitectura.
    - Analizar la plataforma Hyperledger Fabric para obtener información de su funcionamiento y los componentes que lo conforman.
    - Identificar las diferentes decisiones arquitecturales que se tuvieron en cuenta para la construcción de Hyperledger Fabric.
    - Documentar la evaluación de Hyperledger Fabric.

5.2.6 Dinámica del proyecto

Las actividades para la recuperación de la arquitectura siendo este el primer enfoque del estudio de caso se basaron en técnicas de ingeniería inversa obteniendo la mayor parte de información en la documentación oficial del framework Hyperledger Fabric y de una herramienta de recuperación visual la cual obtuvo resultados analizando el repositorio de código abierto donde se encuentra alojado, esto con el fin de obtener la información directamente de la fuente de creación desde un punto de vista de los investigadores. En base al lenguaje en el que se encuentra construido el framework siendo este Golang se realizó una investigación de herramientas visuales que fueran compatibles con el lenguaje y que permitiera la recuperación gráfica de arquitectura.

El análisis de este estudio se inició con la lectura de procesos, técnicas y métodos para la recuperación de arquitecturas, donde se toma como referencia la metodología EVAR que es propuesta como trabajo de grado por los ingenieros Yuli Andrea Ordoñez Guzmán y Edwar Alejandro Giraldo Muñoz. Todo el proceso se realizó de manera empírica. Para escoger la herramienta visual de recuperación utilizada, primero se realizó una investigación de las posibles herramientas que se podían utilizar; se tomó una lista del trabajo de grado anteriormente mencionado debido a que tenían una investigación en el ámbito funcional y documental asesorada por profesores de la Universidad, de este trabajo se tomaron 2 herramientas que a grandes rasgos por su descripción podían aplicar para la recuperación estas son Softwarenaut y MOOSE, además se realizó una investigación de herramientas que fueran específicas para el lenguaje en el cual está construido la plataforma se encontró 1 candidata que se ajustaba a las especificaciones de la plataforma esta es GoPlantUML [56]. Se procede a realizar pruebas sobre las herramientas para conocerlas y obtener observaciones de estas, en este proceso encontramos muchos inconvenientes como la falta de documentación oficial, en la mayoría de ocasiones era incompleta, lo cual obligaba a recurrir a foros de discusión. Se probaron las 3 herramientas, pero 2 de ellas no arrojaron ningún resultado estas son Softwarenaut, y Moose; la tercera herramienta GoPlantUML se pudo probar exitosamente, aunque sus resultados no aportaron gran valor a la obtención de la arquitectura. En la **Tabla 13. Herramientas de recuperación visual** se muestran los detalles de cada una de las herramientas.

Herramienta	**Detalles**
GoPlantUML	• Trabaja sobre código fuente Golang. • Procesa el código y lo abstrae a un diagrama de clases PlantUML. • Genera estructuras e interfaces, así como la relación entre ellas • Reconstruyo la vista de implementación de la plataforma
Softwarenaut	• Trabaja sobre el metamodelo

	• FAMIX. Permite ver los componentes de un sistema y sus relaciones, además de proveer un mecanismo para navegar dentro de cada componente. • Depende de herramientas que realicen la traducción del código fuente al metamodelo FAMIX. • No fue posible traducir el código de la plataforma al metamodelo FAMIX
MOOSE	• Es una plataforma que permite el análisis de aplicaciones y datos, mediante la visualización y exploración. • Moose contiene importadores internos que le permiten recibir. • datos o código fuente en Smalltalk, XML, MSE. • Requiere transformar el código a un metamodelo FAMIX • No fue posible traducir el código de la plataforma al metamodelo FAMIX

Tabla 13. Herramientas de recuperación visual. Fuente propia

Para el segundo enfoque se puso en práctica la metodología de evaluación ATAM, llevando a cabo las 2 fases que se componen de 9 pasos, esto conforma la metodología adicional se organizó un grupo de personas interesados en el tema, cada una de las personas tuvo un rol asignado siendo estos:

- **PhD. Julio Ariel Hurtado –** Evaluador de la arquitectura, se encargará de llevar a cabo la ejecución de la evaluación en contraste con los atributos de calidad propuesto.

- **Edgar Dulce Villareal –** Cliente, el cual se encargará de explicar la finalidad y comportamiento que debería tener el sistema a implementar en este caso la gestión de información en el traslado de pacientes además de las expectativas de la evaluación.

- **Juan David Muñoz –** Arquitecto de la plataforma Hyperledger Fabric quién realizó la recuperación y quien presenta a los demás miembros cómo está construida junto con algunas decisiones de diseño y la forma en que interactúa en ejecución.

- **Astrid Carolina Ordoñez –** Arquitecta de software, quién realizó la recuperación, expondrá sus puntos de vista acerca de las decisiones detrás de la plataforma Hyperledger Fabric, en el contexto del caso.

- **Giovanna –** Desarrolladora e investigadora en tecnologías BlockChain. Expondrá sus puntos de vista desde la perspectiva del desarrollo.

Los pasos de la metodología se analizaron y se describieron según el rol de participación siendo estos:

- **Fase 1**

    - **Paso 1 – Presentar la metodología ATAM (véase Anexo B)**: se presenta la metodología a todos los interesados y se explica el proceso a seguir con los roles y responsabilidades de cada uno en el proyecto, junto con los resultados a obtener De la ejecución de la metodología ATAM se espera obtener un informe el cual tenga los beneficios y riesgos

que pueden presentarse al desarrollar el sistema con la plataforma Hyperledger Fabric, los cuales se pondrán en contraste para tomar una decisión

- **Paso 2 – Presentar las pautas de negocio:** se presenta el sistema desde el punto de vista de negocio detallando las principales funcionalidades restricciones y metas definidas para el sistema. Para este paso es necesaria la intervención del cliente Edgar Dulce Villareal el cual expondrá las funciones más importantes del sistema, restricciones y metas esperadas del sistema, esto brindara una visión acerca de los atributos de calidad que se presentan sirviendo así de guía para el resto de la evaluación.

- **Paso 3 – Presentar la arquitectura:** se presenta la arquitectura de software, junto con las restricciones, técnicas, estilos utilizados, la interacción de los componentes y si necesita otros sistemas para el funcionamiento. Para este paso es necesaria la intervención de Juan David Muñoz que cumple con el rol de representante del arquitecto, el cual expondrá la arquitectura de Hyperledger Fabric gracias a la recuperación que se realizó en base al código y documentación esto incluye cómo está construida la plataforma, las decisiones de diseño, los componentes, conectores y la forma en cómo interactúan.

- **Paso 4 – Identificar Propuestas Arquitectónicas**: se presentan los estilos o propuestas arquitectónicas utilizadas que definirán las principales características de la plataforma. Para este paso es necesaria la intervención del tesista Juan David Muñoz donde presentará las propuestas y principales características de la plataforma.

- **Paso 5 – Generar el Árbol de Utilidad:** se presenta las características más importantes del sistema identificando los escenarios del árbol y clasificándolos según dos dimensiones las cuales son: la importancia de la característica para el éxito o funcionamiento correcto del sistema y la dificultad que requiere la implementación para lograrlo, donde las calificaciones varían entre Alto, Medio y bajo para cada dimensión. En este paso se requiere la participación del tesista Juan David Muñoz (Arquitecto) y Edgar Dulce Villareal (Cliente), tesista Astrid Carolina Ordoñez (Arquitecta) y opcionalmente la del PhD Julio Ariel Hurtado (Evaluador).

- **Paso 6 – Analizar las Propuestas Arquitectónicas:** se presenta la evaluación de las propuestas arquitectónicas junto con el árbol de utilidad, evaluando la influencia de cada propuesta para la obtención o no del atributo de calidad requerido, se identifican riesgos, puntos de sensibilidad y cohesión. Para este paso es necesaria la intervención del PhD. Julio Ariel Hurtado el cual es el evaluador de la arquitectura y encargado de llevar a cabo este proceso.

- **Fase 2**

  - **Paso 7 – Lluvia de Ideas:** se involucra todo el equipo de trabajo para identificar nuevos escenarios, los que también se priorizan y se comparan con los identificados en el árbol

de utilidad generado. Para este paso es necesario que se involucre todo el equipo, Tesista Juan David Muñoz (Arquitecto), Tesista Astrid Carolina Ordoñez (Arquitecta), profesor Edgar Dulce (Cliente), PhD. Julio Ariel Hurtado (Evaluador).

- **Paso 8 – Analizar las Propuestas Arquitectónicas:** se realiza lo mismo que en el paso 6 para el nuevo árbol de utilidad.

- **Paso 9- Presentar Resultados:** se presentan los resultados a todos los involucrados y se entrega la documentación generada por las salidas de ATAM que incluyen documento de propuestas arquitectónicas, conjunto de escenarios priorizados, conjunto de preguntas basadas en atributos, árbol de utilidad, riesgos más la relación de impacto a las pautas de negocio establecidas y no riesgos documentados.

5.2.7 Resultados obtenidos

La obtención de resultados se divide en dos partes, la primera se enfoca en la recuperación de la arquitectura de la plataforma Hyperledger Fabric representándola por medio de vistas y relacionando sus atributos, tácticas y patrones; la segunda parte plasma la retroalimentación y artefactos de la evaluación de los aspectos arquitecturales recuperados contrastados con un proyecto de desarrollo, esto se realizó por medio del método ATAM [14].

5.2.7.1 Recuperación

Después de hacer una investigación en la documentación oficial y el repositorio de código abierto se logró identificar la siguiente lista de patrones, tácticas y atributos de calidad que comprende la plataforma Hyperledger Fabric, apoyado por el libro de Bass et al. [5] para identificar y relacionar cada uno de estos aspectos arquitectónicos

- **Patrones**
  - Publicador suscriptor
  - Niveles múltiples
  - Autorización
- **Tácticas**
  - Autorizar actores
  - Autenticar actores
  - Identificar actores
  - Encriptar datos
  - Mantener múltiples copias de datos
  - Mantener múltiples copias de maquinas
  - Resincronización de estado
  - Utilizar intermediario

- **Atributos de calidad**

    - Seguridad
    - Escalabilidad
    - Desempeño
    - Disponibilidad

Los aspectos arquitecturales mencionados anteriormente, se lograron extraer de la arquitectura, la cual se recuperó en base al siguiente proceso

Después de hacer un recorrido en la documentación oficial y el repositorio de código abierto se logró identificar las siguientes vistas arquitectónicas que componen la plataforma Hyperledger Fabric.

1. Vista Lógica

Para la construcción de esta vista fue necesario recopilar la información que se encuentra en la documentación oficial de Hyperledger Fabric, obteniendo una arquitectura modular basada en ejecución de órdenes (Execute-Order-Validate).

Para un primer entendimiento del sistema se presentará un bosquejo de los componentes modulares de alto nivel en la ***Figura 25. Bosquejo Modular***.

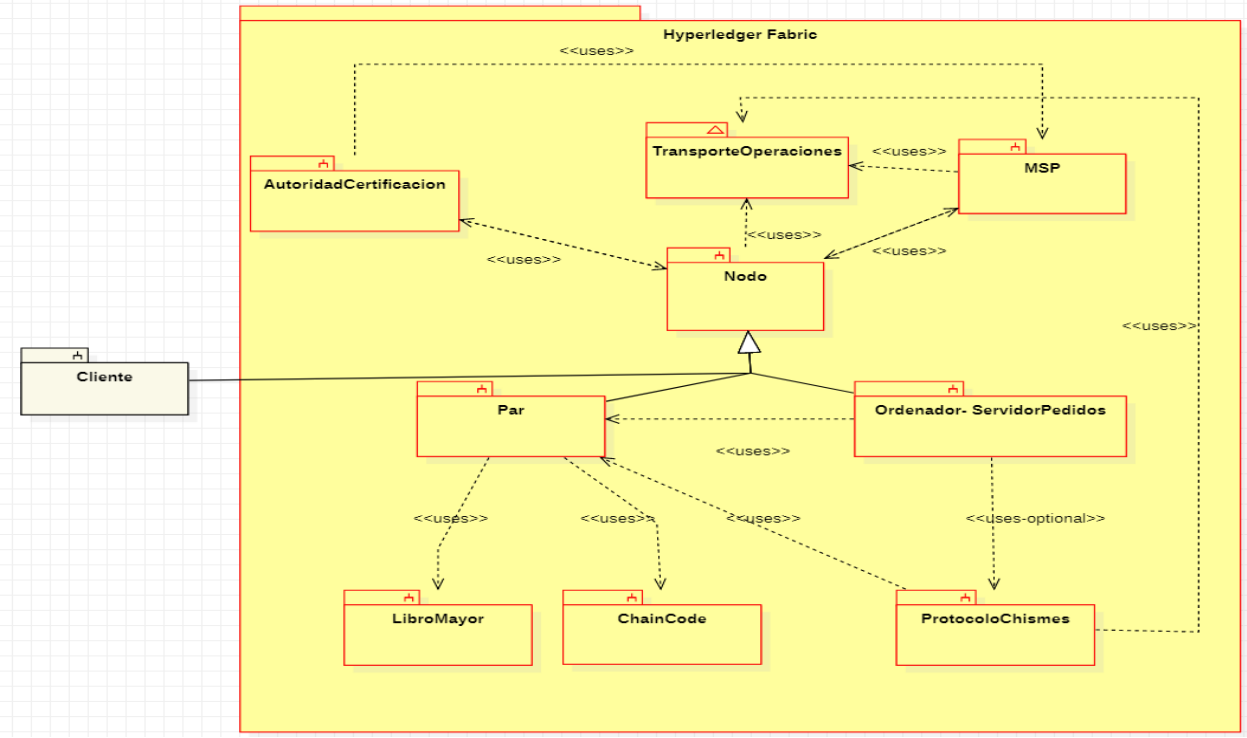


Figura 25. Bosquejo Modular. **Fuente propia**

Una descripción de los subsistemas presentados en la ***Figura 25. Bosquejo Modular***. Se presenta en la **Tabla 15. Descripción subsistemas**.

Módulo	Descripción
Autoridad certificadora	Encargada de emitir certificados a los pares una vez creados
MSP	Proveedor de servicios de membresía (MSP) componente encargado de proporcionar credenciales a los clientes y pares para que participen de una red Hyperledger Fabric.
Transporte de Operaciones	Es el encargado de suscribir pares y organizaciones para el transporte de información
Par	Son los encargados de tener libros de contabilidad y contratos inteligentes. Exponen un conjunto de api que permiten interactuar con los servicios
Cliente	Es el encargado de lanzar las transacciones que se van a ejecutar en la red
Ordenador (Servidor de pedidos)	Un colectivo definido de nodos que ordena las transacciones en un bloque y luego distribuye los bloques a los pares conectados para su validación y confirmación
Protocolo de Chismes	El protocolo de chismes se encarga de gestionar el descubrimiento de pares y pertenencia a los canales; difundir los datos del libro mayor a todos los pares del canal y sincronizar el estado del libro mayo a todos los pares del canal
Chain Code (Contrato inteligente)	Chaincode o contrato inteligente es el encargado de recibir las invocaciones de un cliente externo a la red, para administrar el acceso y modificaciones al libro mayor
Libro Mayor	Se compone de dos partes, la cadena de bloques y la base de datos del estado. La cadena de bloques es inmutable, la base de datos del estado contiene el valor actual del conjunto de pares clave-valor que han sido agregados, modificados o eliminado por las transacciones comprometidas en la cadena de bloques

Tabla 15. Descripción subsistemas. Fuente propia

Para un nivel más cercano de detalle se refinaron los subsistemas que más información brindaban

- **Libro Mayor**
    - **Cadena de bloques:** Se encarga de almacenar las transacciones que se han llevado a cabo ya sean validas o invalidas, cabe mencionar que esta cadena es de estado inmutable.
    - **Base de datos:** es una base de datos que contiene el valor actual del conjunto de pares clave-valor que han sido agregados, modificados o eliminados por el conjunto de transacciones validadas y confirmadas en la cadena de bloques.

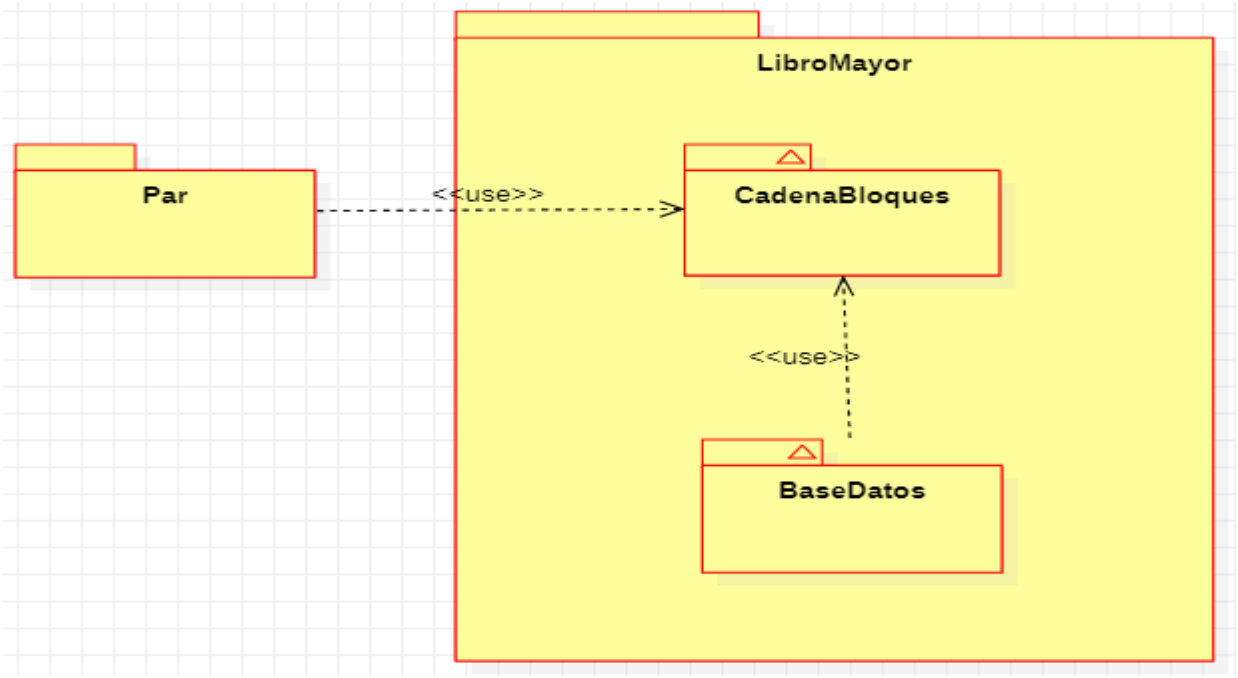


Figura 26. Descomposición del Libro Mayor. **Fuente propia**

- **MSP**
    - **Emisión de Certificados:** Se encarga de emitir los certificados a las entidades certificadoras para poder pertenecer a la red.
    - **Validación de Certificados:** Se encarga de validar la identidad de los pares que se conectan al canal. En caso de que exista más de una organización, con la identidad que presenta el par se sabe a cuál pertenece.
    - **Autenticación de Usuarios:** Es el encargado de permitir o no el acceso a la red dependiendo si el cliente o par que se desea conectar cumple con las políticas de seguridad establecidas en la implementación.

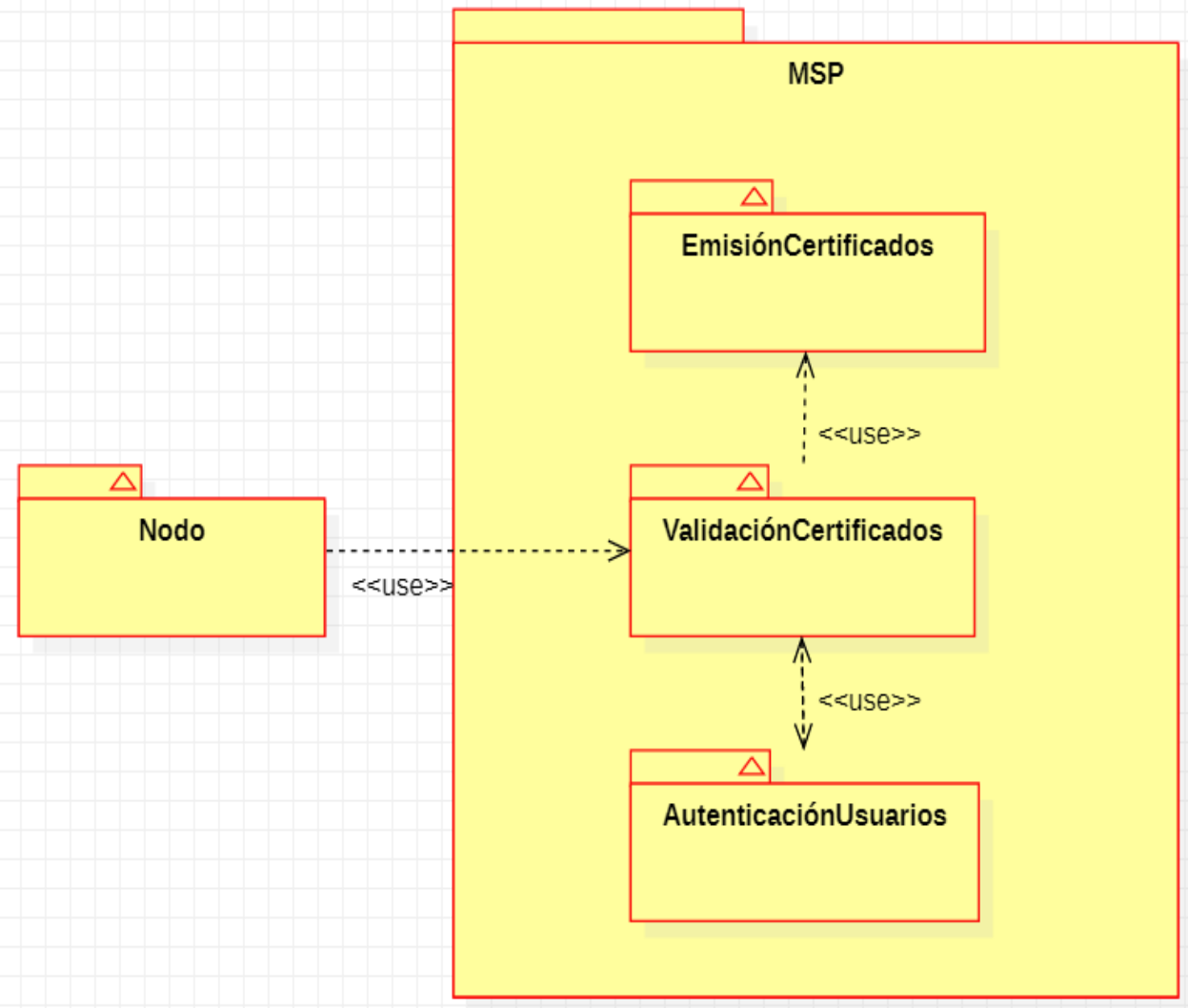


***Figura 27. Descomposición del MSP*. Fuente propia**

- **Ordenador -. Servidor de Pedidos**
    - **Control de Acceso:** Se encarga de restringir quién puede leer y escribir datos en el servidor de pedidos y también quien puede configurarlos
    - **Construcción de Transacciones:** Se encarga de recibir las transacciones y crear bloques de transacciones para posteriormente distribuir a todos los pares del canal
    - **Distribución de Transacciones**: Se encarga de distribuir los bloques de transacciones a los pares conocidos en el canal.

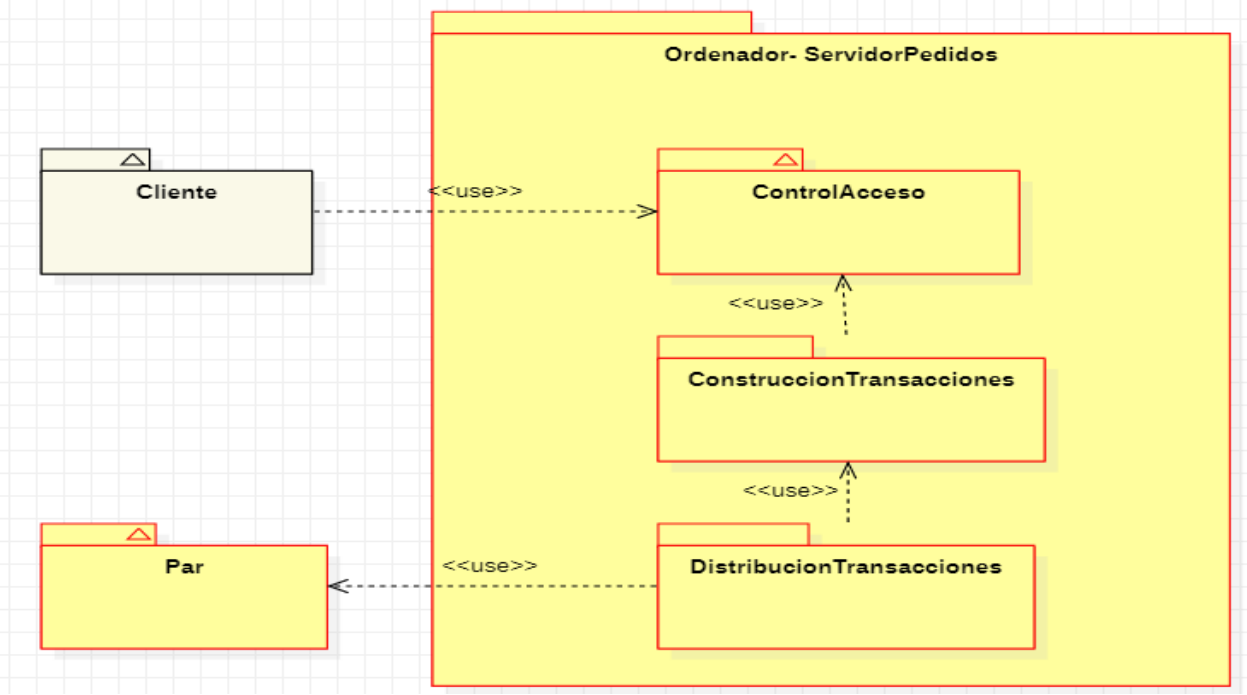


Figura 28. Descomposición del Ordenador – Servidor de Pedidos. **Fuente propia**

- **Protocolo de chismes**
    - **Administrar la Conexión de Pares:** identifica continuamente los pares disponibles conectados al canal y también detecta los que se han desconectado
    - **Difusión de Datos:** Difunde los datos contables en todos los pares del canal, los pares que se encuentren fuera de sincronía, identifica los bloques que faltan y copia los datos correctos.
    - **Actualización Nuevos:** Se encarga de actualizar al estado actual del libro a los pares que son nuevos en la red, permitiendo la transmisión de estado de punto a punto.

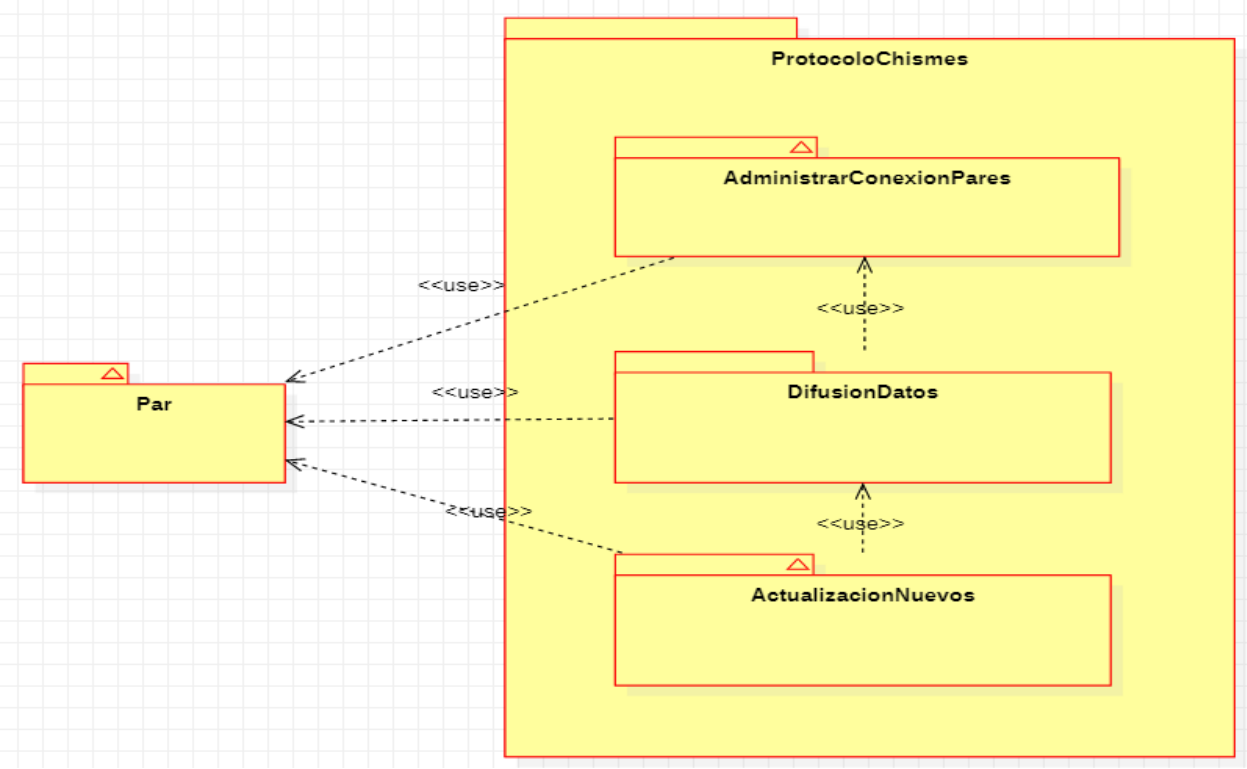


***Figura 29. Descomposición del Protocolo de chismes*. Fuente propia**

2. Vista de Procesos

Los procesos que se ejecutan en Hyperledger Fabric para llevar a cabo una transacción los podemos dividir en 3 fases.

- **Fase 1: Propuesta**

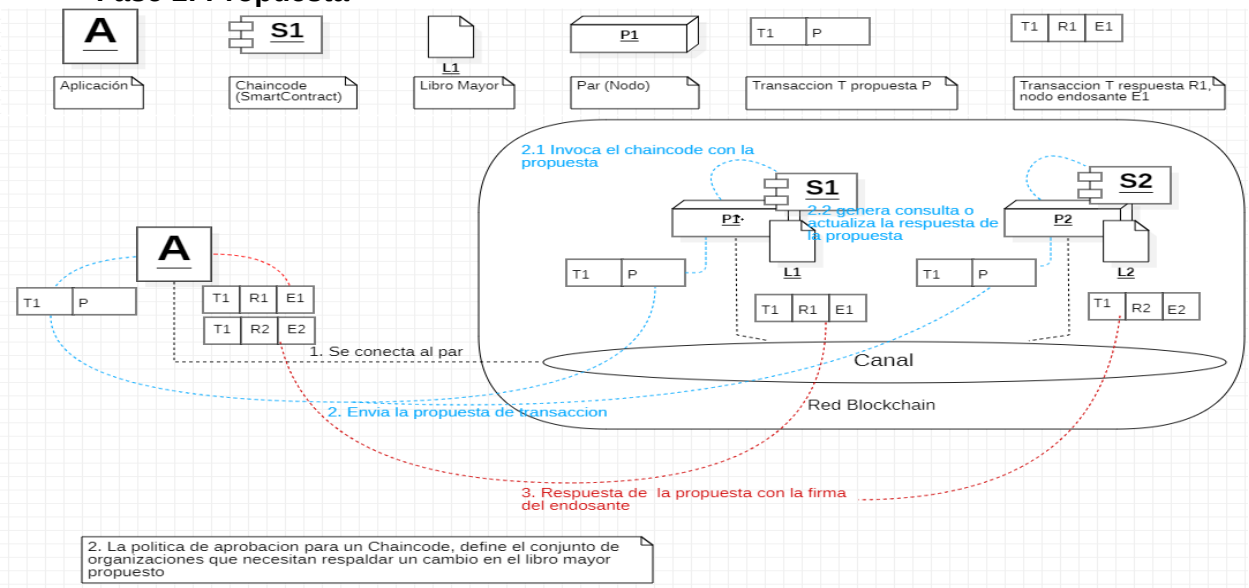


Figura 30. Propuesta. Fuente propia

- **Fase 2: Empaque**

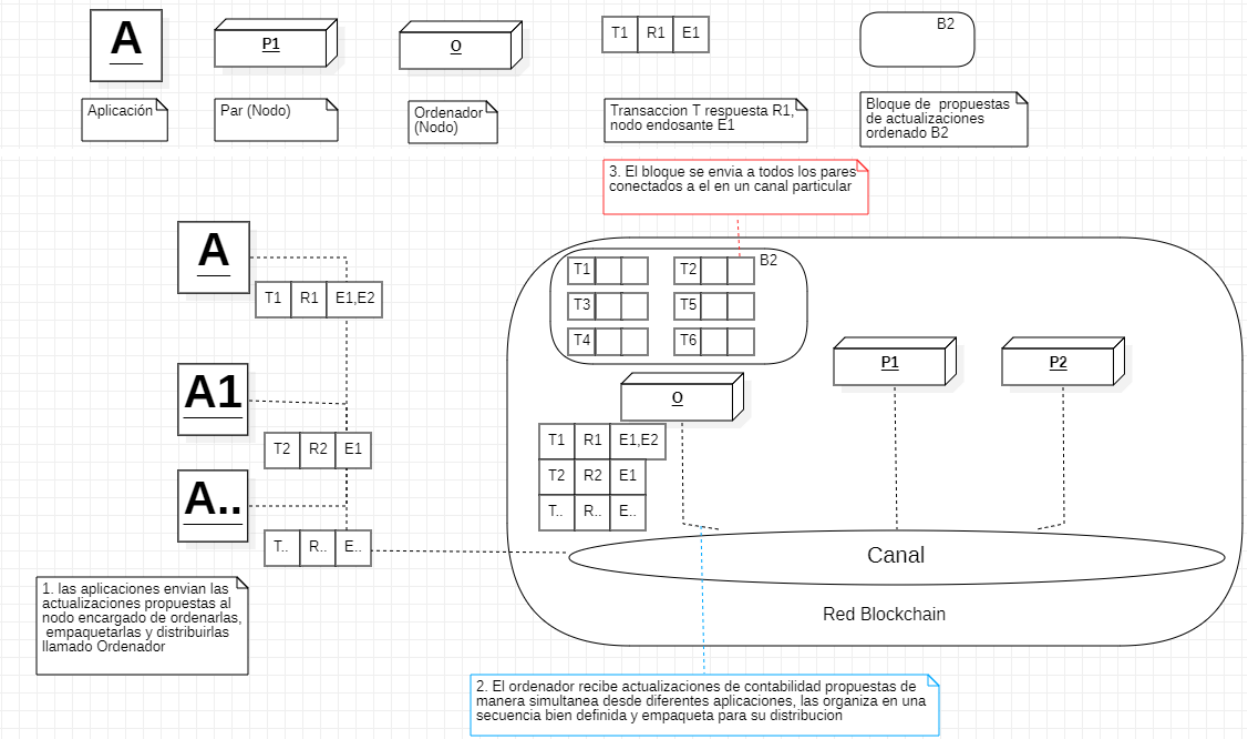


Figura 31. Empaque. Fuente propia

- **Fase 3: Validación**

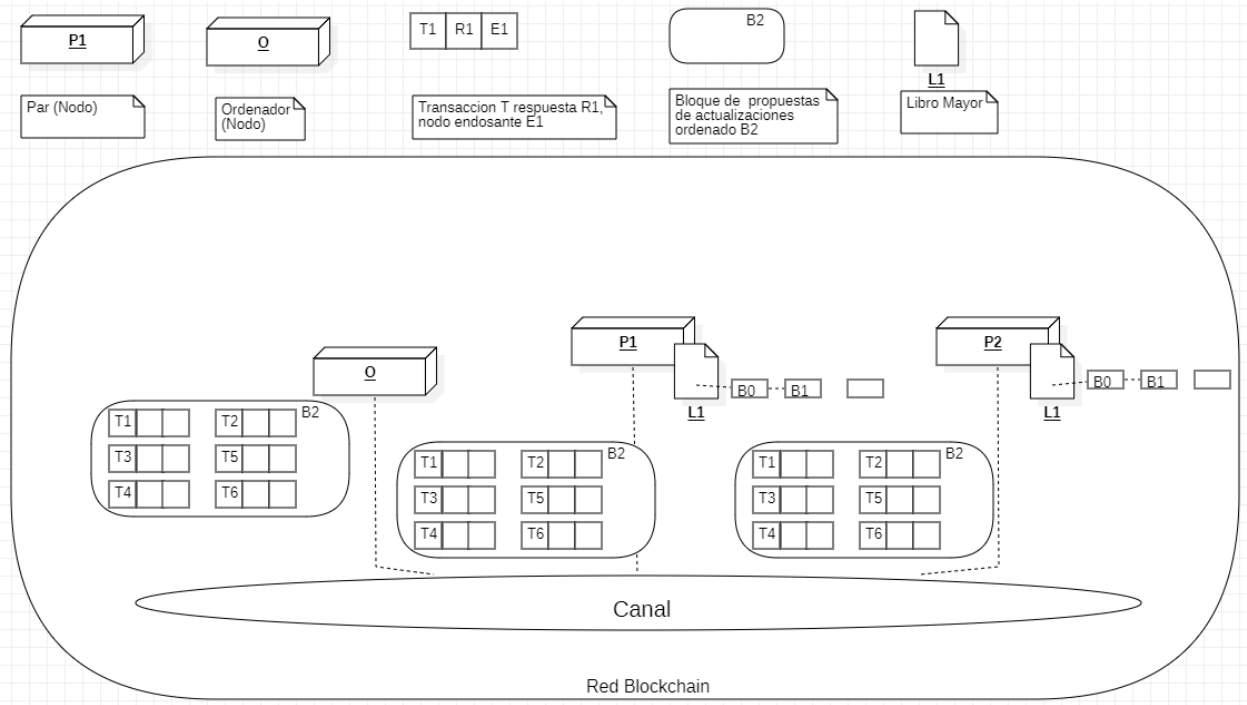


Figura 32. Validación. Fuente propia

3. Vista de Implementación

Para este punto se muestra la vista de implementación del **Servidor de Pedidos** obtenida gracias a la herramienta de análisis de código GoPlantUML.

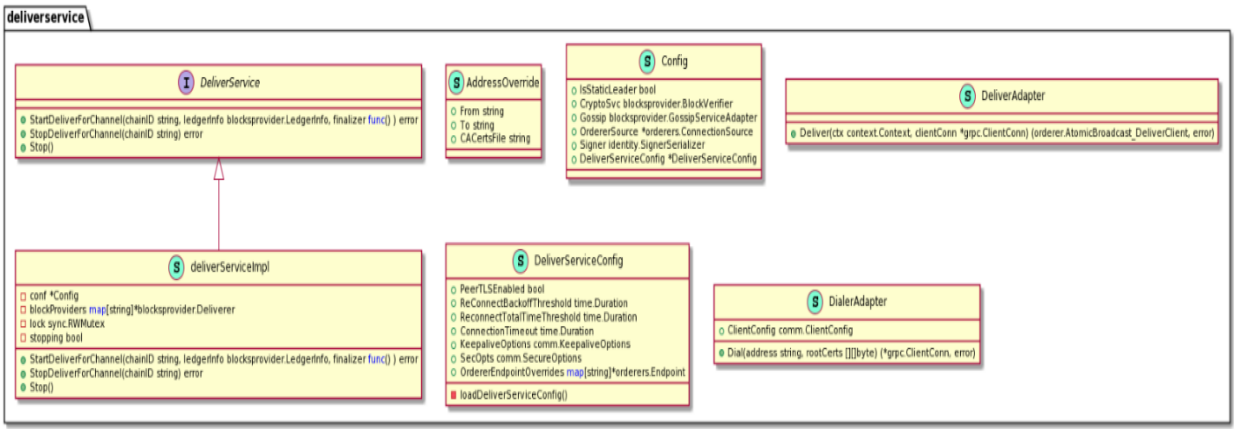


Figura 33. Vista de implementación del servidor de pedidos. Generado por GoPlantUML [56]

5.2.7.2 Evaluación de la arquitectura en un sistema de traslado de pacientes en el sector salud, con ATAM

En esta sección del caso de estudio se ejecutó el método de evaluación ATAM con los roles y fases previamente descritos, se utilizó como arquitectura de referencia la recuperada en la primera parte y como proyecto a desarrollar un sistema de traslado de pacientes en el departamento de Nariño propuesto por el integrante del equipo de evaluación Edgar Dulce Villareal, se necesitaron de 2 secciones virtuales la primera con una duración de 4 horas y la segunda con una duración de 2 horas 20 minutos (vease Anexo D), para llevar acabo la evaluación donde se obtuvieron los siguientes artefactos como resultados.

- **Árbol de utilidad**

Se expusieron los atributos de calidad que impactan el software por medio de escenarios que pueden presentarse al momento de entrar en operación, posterior a ello se priorizaron para evaluarlos de manera específica, este paso se llevó a cabo con todo el equipo de evaluación establecido, esta priorización se etiqueto de la siguiente manera:

- H: Prioridad Alta
- M: Prioridad Media
- L: Prioridad Baja

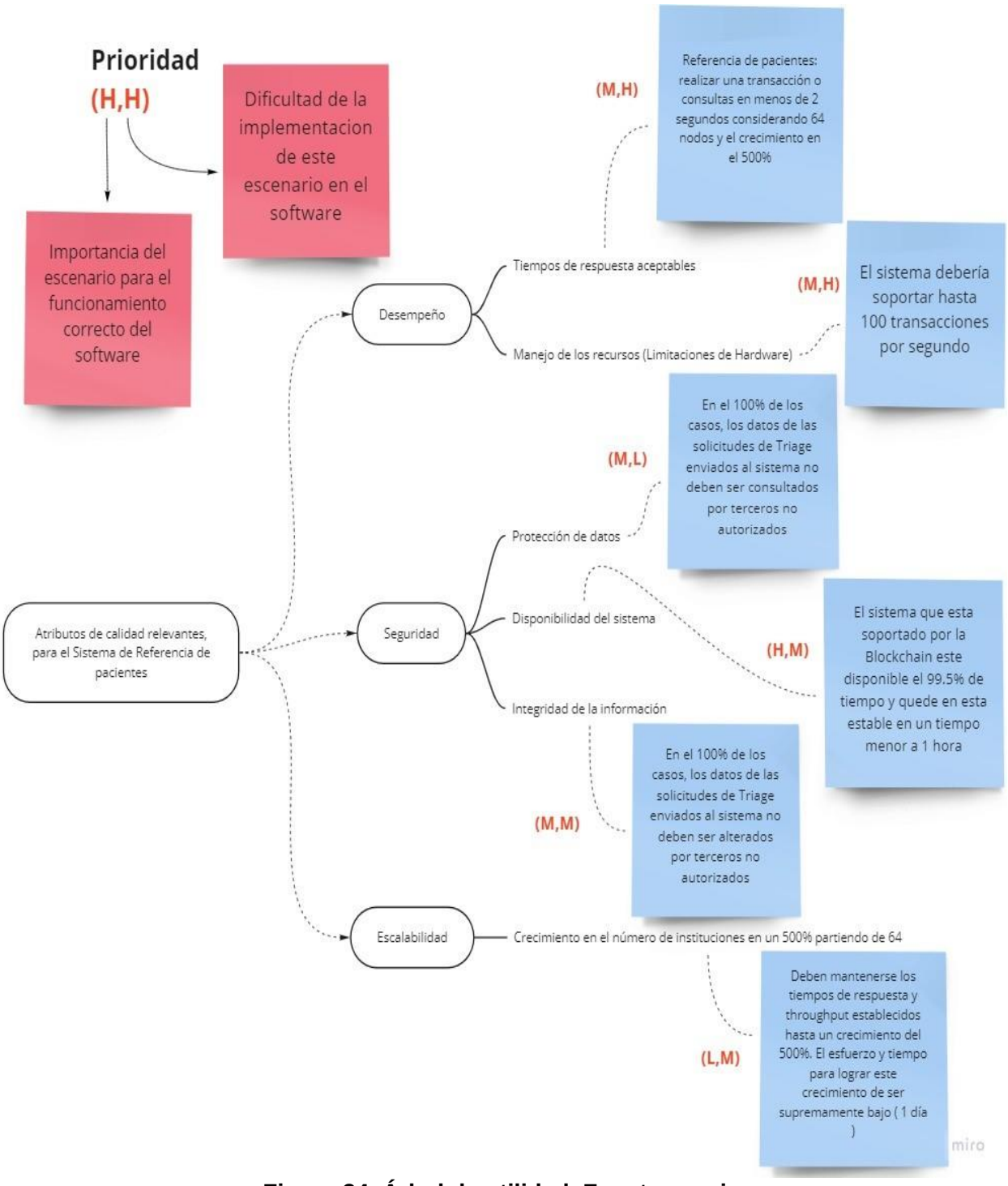


Figura 34. Árbol de utilidad. Fuente propia

- **Análisis Propuestas Arquitectónicas**

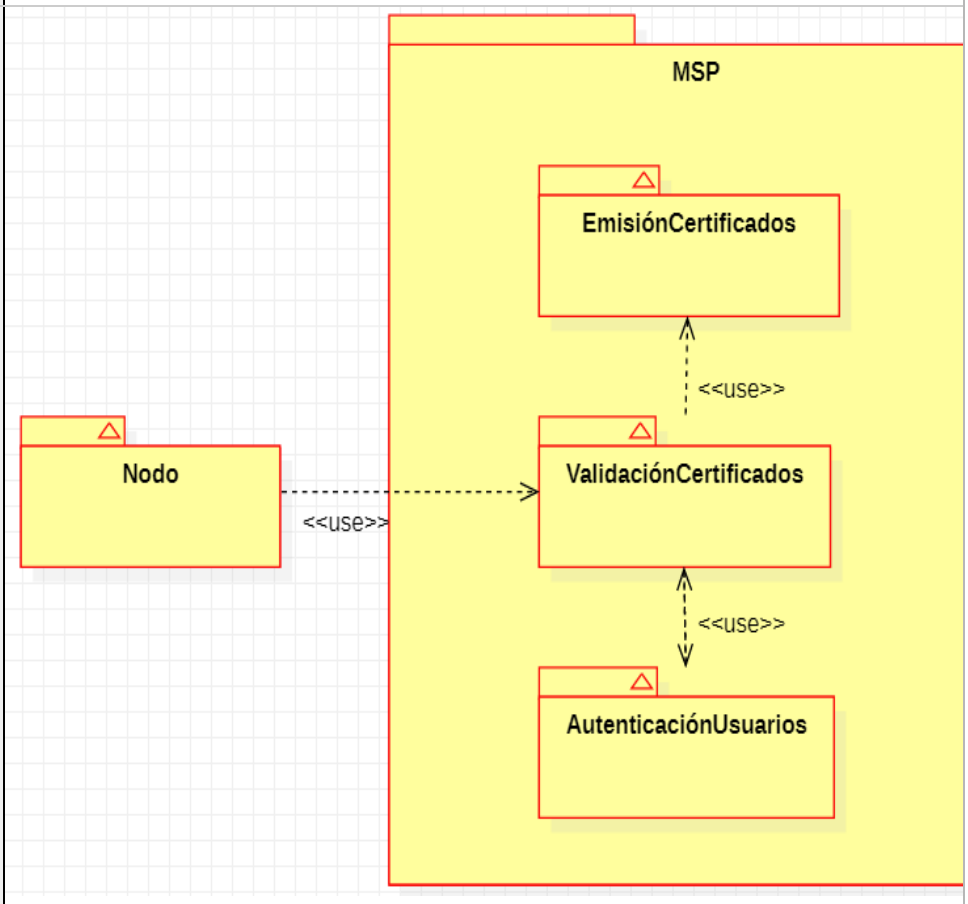
Del árbol de utilidad previamente construido se escogieron 2 atributos y 3 escenarios que representan un mayor impacto al software tanto en funcionalidad como en implementación, estos son

Atributo de calidad	Refinamiento del Atributo	Escenario	Priorización
Seguridad	Protección de datos	En el 100% de los casos, los datos de las solicitudes de Triage enviados al sistema no deben ser consultados por terceros no autorizados	(H, L)
	Integridad de la información	En el 100% de los casos, los datos de las solicitudes de Triage enviados al sistema no deben ser alterados por terceros no autorizados.	(H, M)
Desempeño	Tiempos de respuesta aceptable	Referencia de pacientes: realizar una transacción o consultas en menos de 2 segundos considerando 64 nodos y el crecimiento en el 500%.	(M, H)

Tabla 16. Análisis Propuestas Arquitectónicas. Fuente propia

El siguiente paso fue evaluar los escenarios con la arquitectura recuperada para detectar puntos de sensibilidad, compensaciones y riesgos (véase Anexo C) que pudiera tener Hyperledger en el momento de implementar el software bajo esta plataforma. Este paso se llevó a cabo con todo el equipo de evaluación establecido, los resultados obtenidos fueron los siguientes.

Escenario # 1:	En el 100% de los casos, los datos de las solicitudes de Triage enviados al sistema no deben ser consultados por terceros no autorizados		
Atributos	Seguridad		
Ambiente	Operación normal		
Estímulos	Solicitud de consulta de una persona no autorizada		
Respuesta	Acceso denegado		
Decisiones Arquitecturales	**Sensibilidad**	**Compensación**	**Riesgos**

(MSP) Servicio de proveedor de membresía	S1	C2	R1, R2
Servidor de pedidos		C3	R2, R3
Canal	S1		
Razonamiento	S1. Aumenta la confiabilidad C2. Aumenta la reusabilidad, pero afecta el desempeño C3. Aumenta la confiabilidad, pero afecta la disponibilidad R1. Configuración maliciosa del proveedor (Aumenta el riesgo de inseguridad de los accesos a la red) R2. Permisos de clientes a los usuarios (Aumenta el riesgo de inseguridad en los accesos a la red) R3 Asignación de permisos a participantes (Aumenta la inseguridad de la configuración del servidor de pedidos)		
Diagrama Arquitectura			

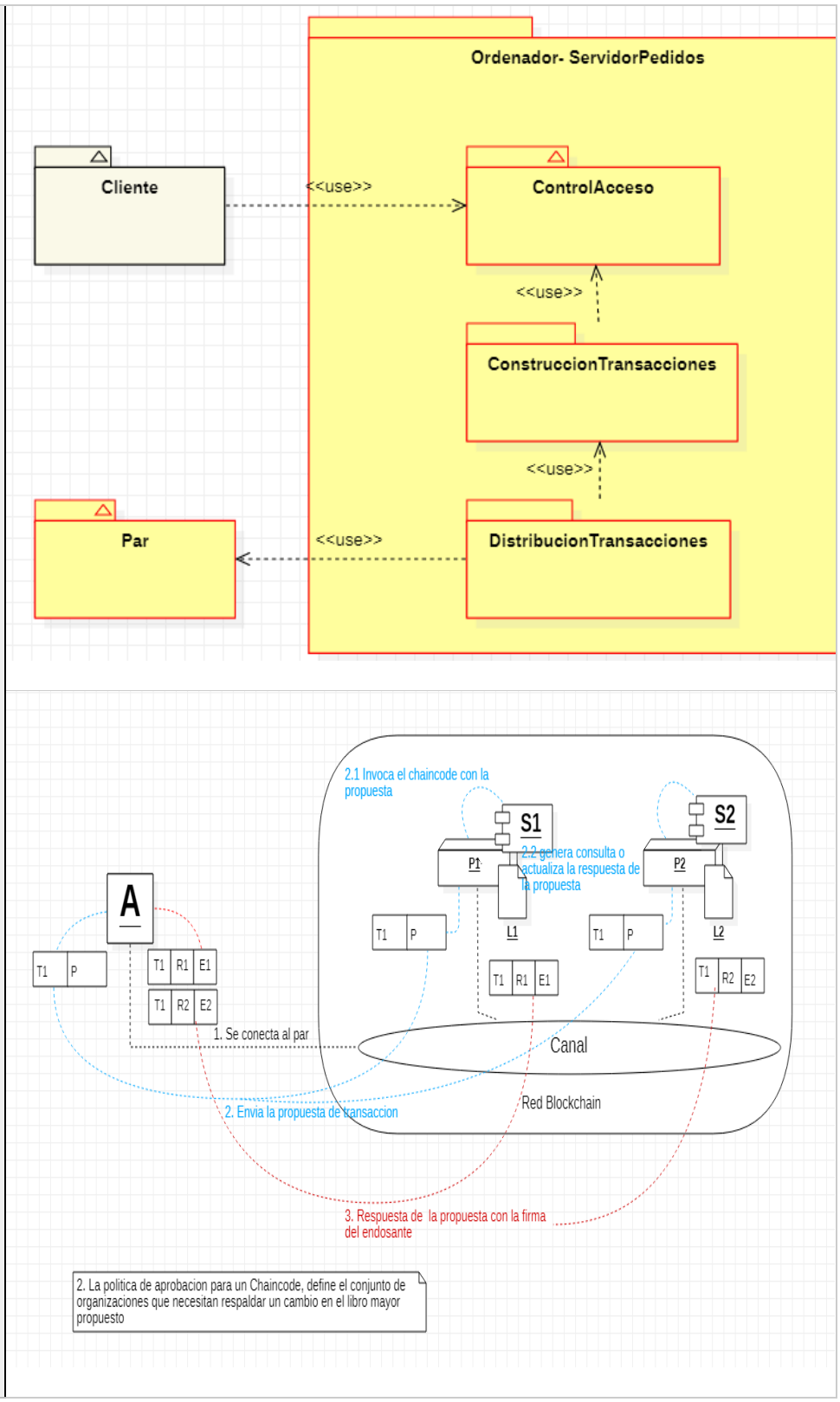
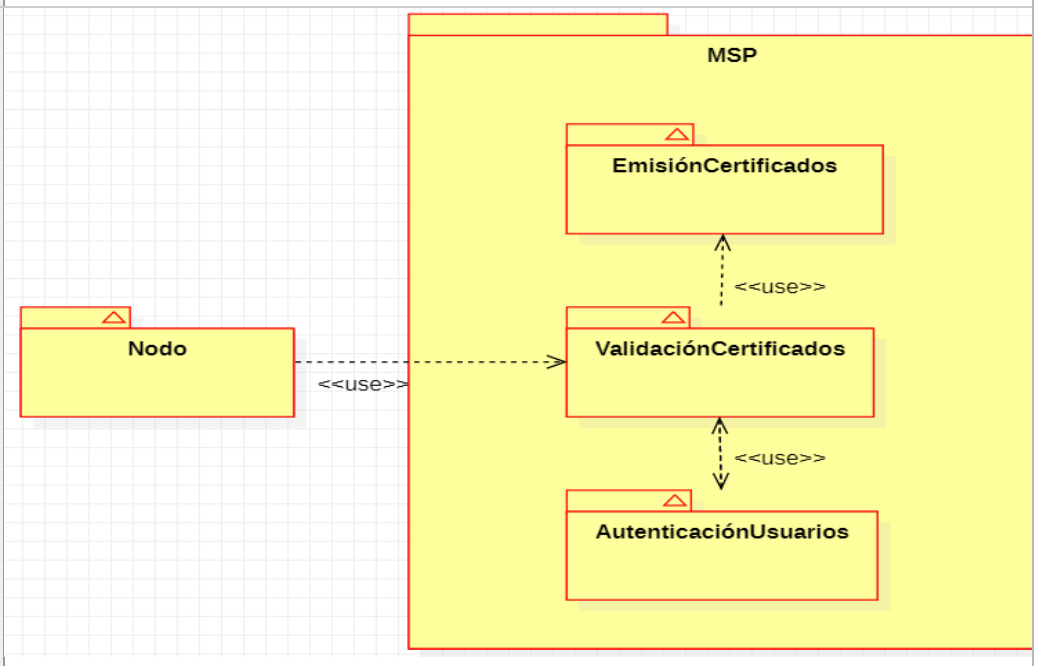


Tabla 17. Escenario 1. Fuente propia

Escenario #2:	En el 100% de los casos, los datos de las solicitudes de Triage enviados al sistema no deben ser alterados por terceros no autorizados.		
Atributos	Seguridad		
Ambiente	Operación normal		
Estímulos	Solicitud de modificación de datos por una persona no autorizada		
Respuesta	Acceso denegado		
Decisiones Arquitecturales	**Sensibilidad**	**Compensación**	**Riesgos**
(MSP) Servicio de proveedor de membresía	S1	C2	R1, R2
Servidor de pedidos		C3	R2, R3
Razonamiento	S1. Aumenta la confiabilidad C2. Aumenta la reusabilidad, pero afecta el desempeño C3. C1. Aumenta la confiabilidad, pero afecta la disponibilidad R1. Configuración maliciosa del proveedor (Aumenta el riesgo de inseguridad en los accesos a la red) R2. Permisos de clientes a los usuarios (Aumenta el riesgo de inseguridad en los accesos a la red) R3 Asignación de permisos a participantes (Aumenta la inseguridad de la configuración del servidor de pedidos)		
Diagrama Arquitectura			

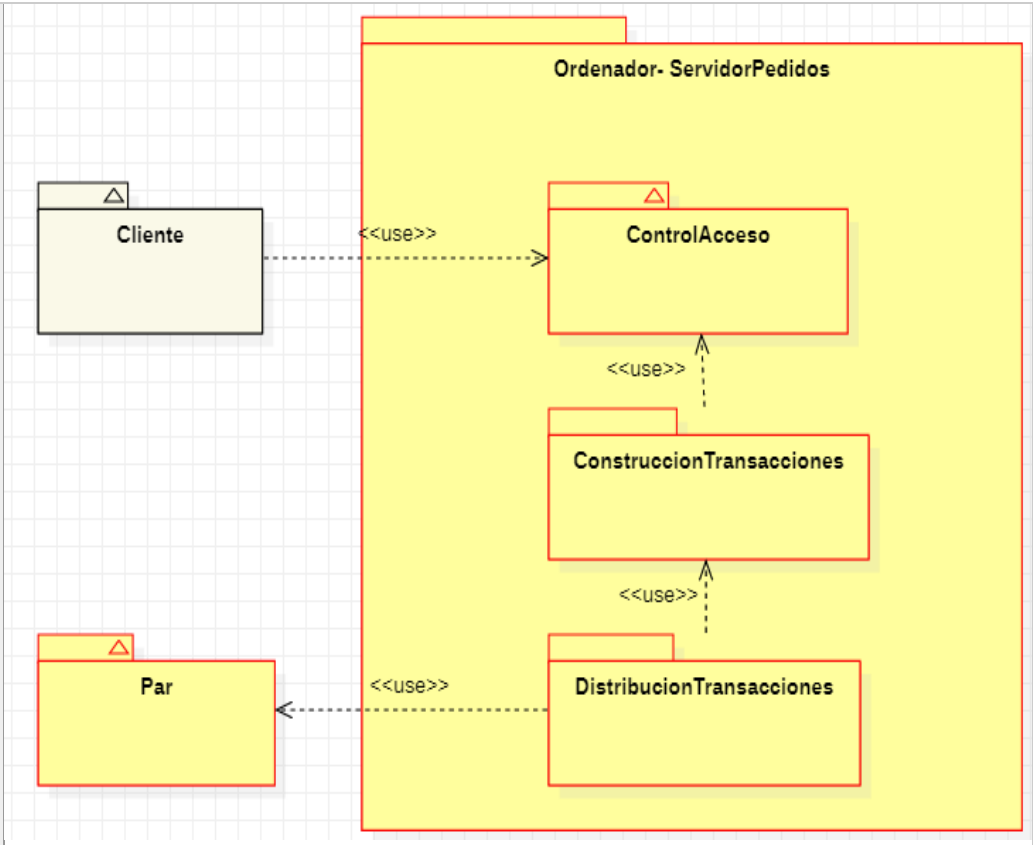


Tabla 18. Escenario 2. Fuente propia

Escenario # 3:	Referencia de pacientes: realizar una transacción o consultas en menos de 2 segundos considerando 64 nodos y el crecimiento en el 500%.		
Atributos	Desempeño		
Ambiente	Operación normal		
Estímulos	Solicitud de registro o consulta de datos de pacientes		
Respuesta	Operación de registro o consulta exitoso en el tiempo estimado		
Decisiones Arquitecturales	**Sensibilidad**	**Compensación**	**Riesgos**
Servidor de pedidos		C1	R3
Protocolo de chismes	S1, S2, S3		

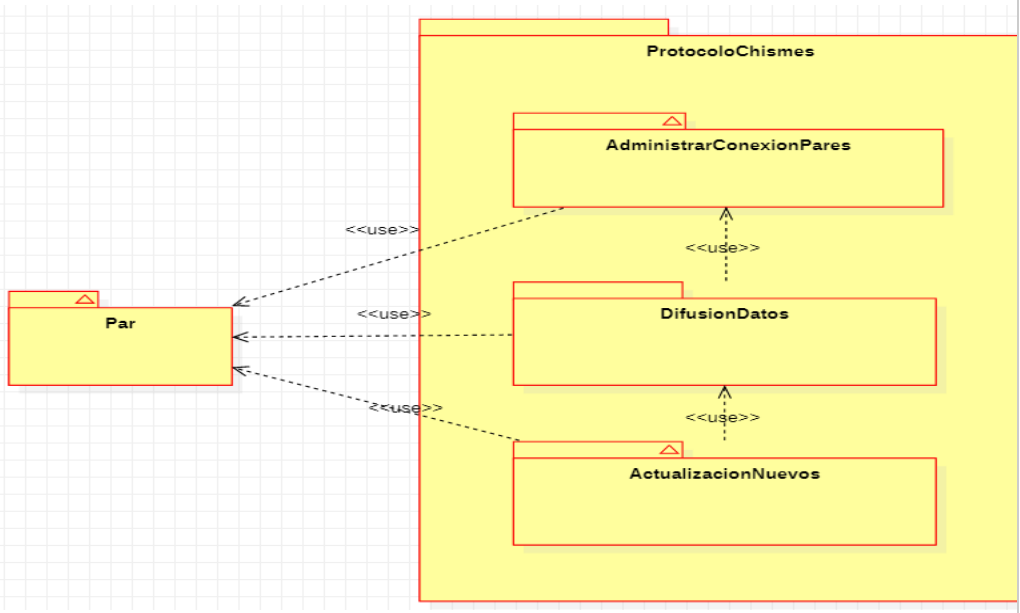
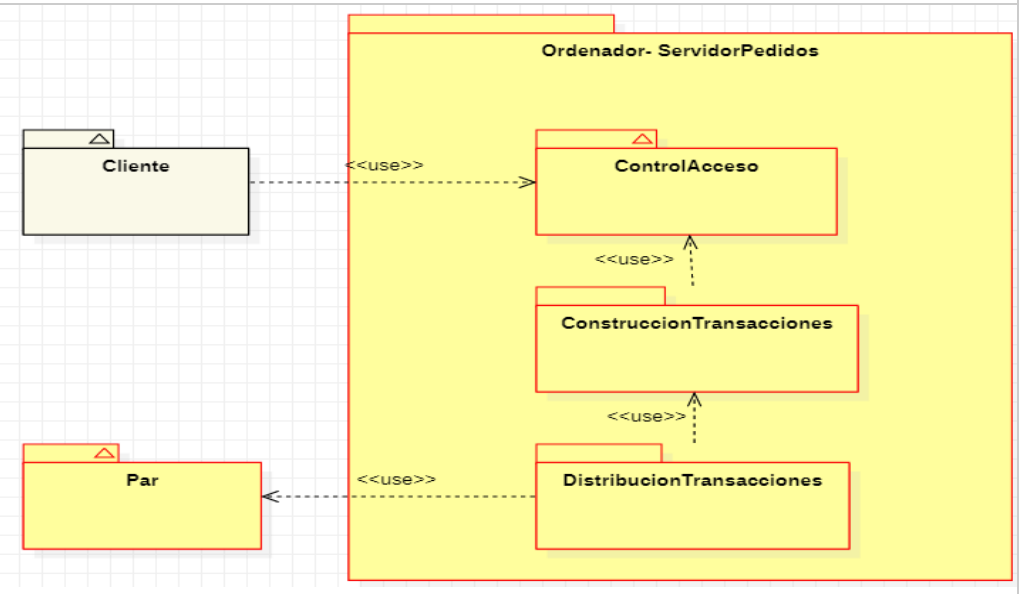
Razonamiento	S1. Aumenta la facilidad de escalar S2. Aumenta la mantenibilidad S3. Aumenta el performance C1. Aumenta la confiabilidad, pero afecta la disponibilidad R3. En caso de fallo, la transacción no se lleva acabo
Diagrama Arquitectura	 

Tabla 19. Escenario 3. Fuente Propia

Además de generar escenarios y refinarlos también se analizó la relación de atributos de calidad con tácticas y la relación de tácticas con patrones con el fin de identificar cada uno de estos ítems en la arquitectura base, tomando como apoyo el libro de Bass et al. [5] para relacionar los aspectos arquitectónicos con los resultados obtenidos, lo cual se ilustra en las siguientes tablas.

Atributos ---------- Tácticas	Authorize actors	Authenticate an Actor	Identify an Actor	Encrypt Data	Maintain Multiple Copies of Data	Maintain Multiple Copies of Computations	State Resynchronization	Increase resources
Security	**x**	**x**	**x**	**x**				
Scalability							**x**	**x**
Performance					**x**	**x**		

Tabla 20. Atributos X Tácticas. Fuente propia

Tácticas ---------- Patrones	Peer to Peer	MSP
Authorize actors		**X**
Authenticate an Actor		**X**
Identify an Actor		**X**
Maintain Multiple Copies of Data	**X**	
Maintain Multiple Copies of Computation	**X**	

Tabla 21. Táctica X Patrones. Fuente propia

5.3 ESTUDIO CASO 2: EVALUACIÓN DE UN MODELO ARQUITECTONICO EN UN EL DOMINIO DEL SECTOR DE SALUD

De acuerdo a la información obtenida al realizar el mapeo sistemático y el desarrollo del caso de estudio basado en la recuperación y evaluación de la arquitectura de una plataforma Blockchain se obtuvo como resultado aspectos arquitecturales de gran impacto tanto en Blockchain públicas como privadas. Por este motivo se tomó la decisión de realizar este estudio de caso para estructurar y evaluar la información obtenida.

5.3.2 preguntas de investigación

El diseño de investigación parte de varias premisas que se tuvieron en cuenta como punto de partida para formular la pregunta de investigación, todas estas enfocadas a la evaluación oportuna de nuestra propuesta, algunas de ellas tenían relación a: la utilidad, intención de uso y complejidad a la hora de aplicar nuestra propuesta en un proyecto real. Así, la pregunta derivada que buscamos responder es: ¿Cuál es la utilidad del catálogo en el diseño de una solución arquitectural basada en Blockchain en los contextos de gestión de salud?

5.3.3 Contexto del caso

En este estudio de caso se tiene como objetivo evaluar la utilidad de MAB (Modelo Arquitectónico Blockchain) mediante la aplicación del modelo en un proyecto enfocado a la gestión de salud que tiene como respaldo la propuesta doctoral "un mecanismo arquitectónico para la interoperabilidad de los sistemas de gestión de información en salud con la tecnología Blockchain", este proyecto produce un alto impacto en el sector salud, directamente en la gestión de información sensible, asociando el estudio de caso con Blockchain, con esto buscan generar un clima de confianza al momento de tomar decisiones derivadas de la consulta, actualización, transmisión, trazabilidad y completitud de la información que se maneja. Su objetivo principal es mejorar la gestión de la información referente a la salud de las personas, mediante la aplicación de la tecnología Blockchain para salvaguardar la misma, optimizando la toma de decisiones médicas hacia el paciente, apoyados por el Hospital Universitario Departamental de Nariño. Para esto, quieren realizar una evaluación de la plataforma de Blockchain de código abierto Hyperledger Fabric, respecto a las cualidades relevantes para la gestión de información sensible en salud.

5.3.4 Diseño de los indicadores y mediciones

De acuerdo al objetivo del estudio de caso, y las preguntas de investigación fue diseñado el indicador, las métricas e instrumento a emplear, la Tabla 22 relaciona estos elementos para el estudio de caso descriptivo.

Objetivo	Preguntas	Indicadores	Aspectos Cualitativos y Cuantitativos	Instrumentos
Realizar la evaluación del modelo	¿Cuál es la utilidad del catálogo en el	EV= Evaluación del modelo	AUP (aspectos arquitecturales	Informe de evaluación

arquitectónico propuesto	diseño en una solución arquitectural basadas en Blockchain en el contexto de gestión de salud?	arquitectónico rationale	utilizados del MAB) V (Viabilidad de los aspectos arquitecturales) S&M (Sugerencias y modificaciones a realizar)	

Tabla 22. Diseño caso de estudio MAB. Fuente propia

5.3.5 Desarrollo del caso

Para la ejecución del estudio de caso se plantearon las siguientes tareas basadas en las metodologías: propuesta y evaluación, para obtener un modelo de arquitectura para Blockchain refinado. Las tareas fueron:

- Diseño de la solución haciendo uso de MAB: Entrega y puesta en contexto de un borrador usable (sitio web Google site. véase Anexo E), del modelo arquitectónico propuesto para la puesta en marcha y validación de la propuesta.

- Evaluación del modelo arquitectónico entregado
    - Objetivos claros para la evaluación de la arquitectura.
    - Analizar e identificar las diferentes decisiones de diseño que se tuvieron en cuenta en el proyecto.
    - Tomar nota de las sugerencias para realizar ajustes al modelo.
    - Documentar la evaluación del modelo arquitectónico propuesto.

5.3.6 Resultados obtenidos

a. Resultados cualitativos

El encargado del proyecto enfocado en traslados de pacientes del sector salud Edgar Dulce recorrió el catálogo de patrones y sus relaciones con el catálogo de atributos de calidad y taticas el cual le permitió realizar un primer diseño donde se evidencia varios aspectos arquitecturales extraídos del MAB, a continuación, se presenta el diseño

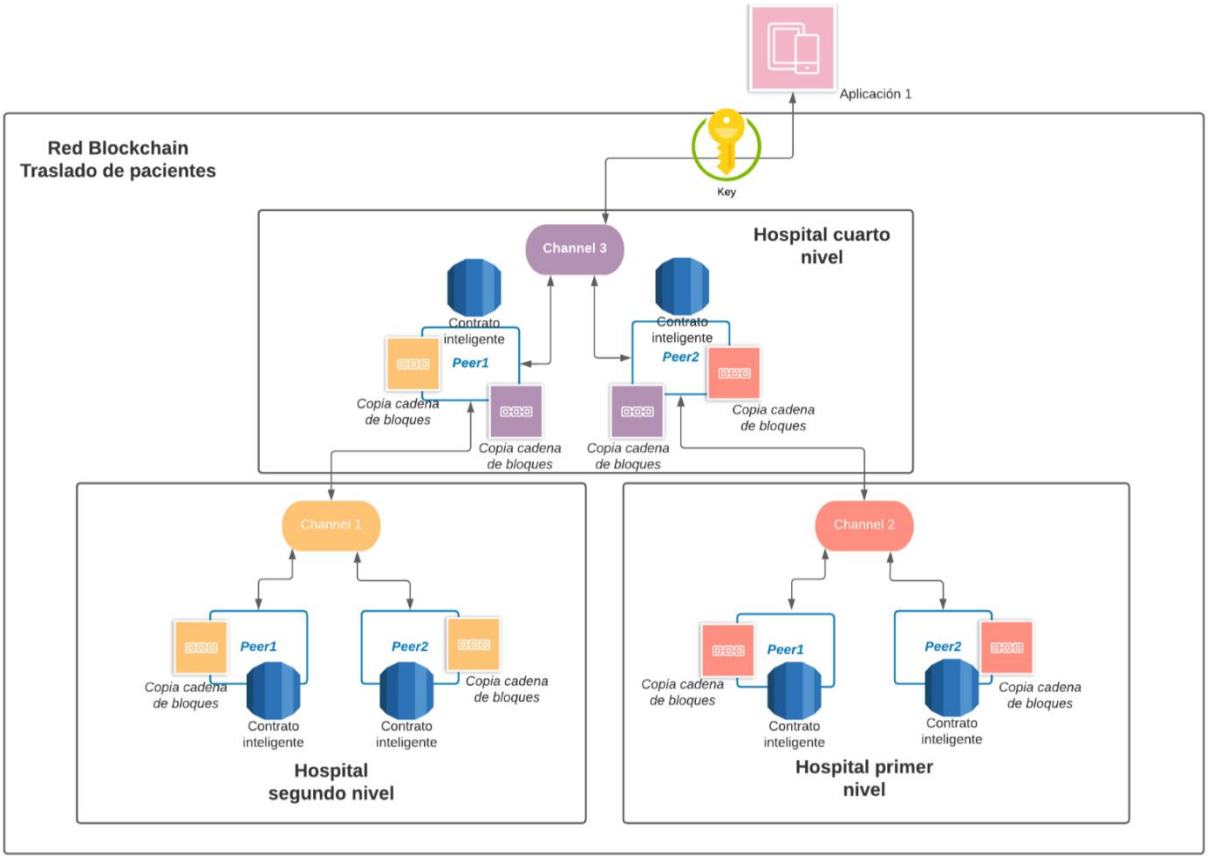


Figura 35. Red Blockchain traslado de pacientes. Fuente propia

En base al diseño anterior se realizó un listado de patrones, tácticas y atributos de calidad que se utilizaron del MAB, además de especificar el rationale de cada uno.

- **Patrones**

  - **Intermediario de confianza:** Se tomó la decisión de usar este patrón debido al grado de sensibilidad de información que se comparte en la red y la facilidad que brinda para aislar y a la misma ves comunicar diferentes organizaciones, en este caso hospitales.

  - **Autorización de participantes:** El uso de este patrón se basa en la seguridad en el momento de obtener acceso a los datos mediante aplicaciones que interactúan con la red, en este caso la información que se puede consultar o ingresar de cada uno de los pacientes que son remitidos entre hospitales

- **Tácticas**

  - **Utilizar intermediario:** Se utilizo junto con el patrón intermediario de confianza con la finalidad de tener alta seguridad en el trasporte de información

  - **Autorizar actores:** Esta táctica permite autorizar o denegar el acceso a información de aplicaciones que lo soliciten

- **Identificar actores:** Esta táctica permite identificar quien está intentando ingresar a la red que junto con la táctica anterior generan una capa de seguridad al patrón de autorización de participantes

- **Atributos de calidad**

  - **Seguridad**: Se requiere que la red tenga un comportamiento constante en cuanto a seguridad, brindando confianza en el manejo de datos sensibles de cada paciente

  - **Escalabilidad**: El traslado de pacientes puede involucrar varios hospitales hasta llegar el caso de realizar traslados departamentales, es por eso que la red debe poder escalar en caso de requerir comunicación con más hospitales.

Sugerencias y Aportes

Además de brindarnos el conocimiento aplicado del MAB, Edgar Dulce también nos dio retroalimentación como parte del proceso de uso del modelo.

- El trabajo es interesante desde el punto de vista de la ingeniería de software, aborda una problemática clara dentro de esta rama de la ingeniería.
- Utiliza una metodología consolidada y muy difundida dentro del campo de la ingeniería de software.
- Realiza una clasificación interesante y fácilmente de comprender, clasificando patrones arquitecturales, que se describen de una manera clara mediante una problemática que se presenta y una posible solución que da el patrón si se lograra implementar, además, muestra un diagrama arquitectural de la solución planteada que ayuda a comprender aún más las utilidades y alcances de cada uno de los patrones.
- También, cuenta con un diccionario de tácticas alineadas a los patrones arquitectónicos, ayudando a lograr las compensaciones necesarias para proyectar una mejor solución arquitectural.
- Así mismo, muestra un catálogo de atributos de calidad, su definición, y un escenario típico con sus características.

En conclusión, es un documento que realiza un gran aporte a la ingeniería de software en el campo de Blockchain, sobre el cual existen varios vacíos en la literatura. El documento ayuda a comprender de una mejor forma los patrones, tácticas y atributos de calidad y la forma en que estos influyen en las decisiones arquitecturales de posibles soluciones que utilicen Blockchain. Muchas gracias por todo su tiempo, conocimiento y esfuerzo utilizados para lograr este importante proyecto de investigación.

Como sugerencia nos mencionó que en la guía usable (sitio web Google site) que se le presento, también se referenciaran las fuentes puesto que esto solo lo teníamos en el documento formal.

b. Resultados cuantitativos

Una vez analizado los resultados de la aplicabilidad del MAB en este primer diseño del proyecto, en términos de patrones, tácticas y atributos de calidad, podemos dar un porcentaje de viabilidad de este conjunto de aspectos arquitecturales, siendo este de un 100% debido a que todos se encuentran comprendidos en el modelo y fueron usados de la manera propuesta

6. CONCLUSIONES, LIMITACIONES, TRABAJOS FUTUROS Y LECCIONES APRENDIDAS

En este trabajo de grado se construyó y evaluó un modelo arquitectónico, basado en el estudio de la literatura y profundización empírica, de donde se extrajo y estructuró la información.

6.1 CONCLUSIONES

Para la construcción del modelo fue necesario abordar diferentes fuentes de información que aportaran conocimiento desde el enfoque arquitectural. Inicialmente el estudio de la literatura fue de vital importancia, lo que nos permitió conocer, analizar y entender ventajas y desventajas de los diferentes aspectos arquitecturales de la tecnología Blockchain, permitiendo extraer y estructurar la información. Además, se discutieron preocupaciones y problemas presentes en la arquitectura Blockchain, parte de nuestro trabajo se enfoca principalmente en el cruce de evidencia entre patrones, atributos de calidad y tácticas, el cual como forma práctica y útil presentamos haciendo uso de los patrones como factor principal ya que estos son soluciones concretas que involucra tácticas y atributos de calidad, por lo cual un conjunto de patrones asociados a sus preocupaciones conforman una herramienta importante en la arquitectura de software.

La construcción de un modelo arquitectónico está acompañada de un arduo estudio de la literatura para lograr abstraer el conocimiento necesario, en este caso en específico nos basamos en dos premisas principales, la primera el mapeo sistemático específicamente los resultados obtenidos en la pregunta de investigación Q4: ¿Cómo están articulados estos aspectos arquitectónicos en las publicaciones que los incluye? Y el segundo el estudio de caso de la evaluación de una plataforma Blockchain (Hyperledger), con estos estudios previos logramos identificar y abstraer algunos patrones arquitecturales y algunas relaciones importantes con atributos de calidad, tácticas, y definir en un contexto más amplio cada uno de ellos. El modelo arquitectónico fue evaluado en un sistema real de gestión de la salud, pero somos conscientes de que, para convertir nuestro aporte en una base de conocimiento sólido, este debe ser evaluado en muchos más sistemas para validar su viabilidad. Sin embargo, consideramos que el modelo arquitectónico Blockchain consolida información valiosa que puede ser de gran aporte para los arquitectos.

En el primer estudio de caso se realiza una investigación empírica desde diferentes enfoques de la ingeniería inversa con la finalidad de encontrar como se encuentran articulados los patrones, tácticas y atributos de calidad de una plataforma Blockchain, se puede evidenciar que es un buen método para profundizar en el diseño arquitectural que conlleva el desarrollo de un sistema debido a que se logran resultados relevantes, permitiendo tomar decisiones en caso de que se requieran conocer aspectos de diseño arquitectónico puntuales para realizar una implementación o adopción de un sistema.

Para la evaluación de los resultados obtenidos en este trabajo de grado se realizó un segundo estudio de caso con la finalidad de analizar la viabilidad de usar el MAB en un proyecto real enfocado en el sector salud que se encuentra en fase de análisis y diseño. La evaluación generó resultados prematuros debido a los tiempos del proyecto y entrega de este trabajo de grado, a pesar de esto se obtiene una buena retroalimentación donde se evidencia que el MAB puede

prometer mucho si se aplica de manera metodológica en un proyecto con un equipo de trabajo bien constituido.

6.2 LIMITACIONES

El modelo arquitectónico propuesto fue evaluado mediante un estudio de caso en un proyecto real para el traslado de pacientes del sector salud, pero consideramos necesario realizar la validación del modelo en más proyectos que verifiquen la viabilidad y utilidad del mismo. Esta parte no se desarrolló debido al alcance del proyecto. La evaluación del modelo debe realizarse de manera metódica acompañada de expertos en arquitectura y aplicada en proyectos reales.

En la recuperación arquitectónica que se realizó en el primer caso de estudio, las técnicas de ingeniería inversa recomiendan apoyarse de herramientas que puedan abstraer del código diagramas arquitecturales con el fin de poder obtener su rationale de una manera más rápida, esta parte tuvo varios inconvenientes debido a la falta de información y documentación oficial, por otra parte la escases de herramientas que se adaptaran al lenguaje en el cual se encontraba construido la plataforma, además de que soportaran la complejidad que implica la implementación de un framework.

Un estudio de caso en un escenario tan especifico es insuficiente para generalizar el valor del MAB. Sin embargo, el catálogo fue construido con una mayor amplitud, para lo cual se requerirá de estudios en los siguientes años que permitan validarlo y mejorarlo en la práctica.

Finalmente, la delimitación de la tesis para poder llevarla a cabo en el tiempo establecido.

6.3 TRABAJOS FUTUROS

Realizar una aplicación basada en Blockchain aplicando los conceptos y aspectos representados en el modelo arquitectónico propuesto, el cual nos permita fortalecer la información y mejorar los aspectos de aplicabilidad de la guía.

Probar de manera masiva el modelo arquitectónico para obtener una retroalimentación completa de sus ventajas, desventajas y así mismo realizar las modificaciones pertinentes. Todo esto con el fin de consolidar nuestro catalogo como una base de conocimiento.

6.4 LECCIONES APRENDIDAS.

Al realizar un mapeo sistemático de la literatura es importante seguir una metodología sistemática para conducir el estudio, tener claro el enfoque y las palabras claves para así obtener resultados concretos que apoyen nuestro estudio.

Para la construcción de conocimiento plasmada en un modelo arquitectónico es importante realizar la validación u evaluación amplia del mismo con expertos en el tema con el fin de consolidar la base conceptual.

Los escenarios son un mecanismo que nos ayuda a detectar puntos de sensibilidad, compensaciones riesgos, funcionalidad, atributos de calidad, tácticas o temas de interés de un sistema software.

Las herramientas de visualización de la arquitectura son de gran importancia a la hora de realizar la recuperación de la arquitectura debido a que nos ayudan en este proceso, pero es importante tener en cuenta que la herramienta a utilizar debe ser analizada según las características del sistema para obtener un resultado satisfactorio.

Es importante contar con personas expertas en un ámbito para este caso en la construcción de un sistema software para obtener mejores resultados a la hora de definir las decisiones arquitecturales tenidas en cuenta en X sistema. De esta manera se logra que dichas personas indaguen sobre las decisiones tomadas en una arquitectura y se genere una definición más completa.

REFERENCIAS BIBLIOGRÁFICAS.

[1] A. S. Bruyn, “Blockchain an introduction Research paper,” 2017.

[2] F. Casino, T. K. Dasaklis, and C. Patsakis, “A systematic literature review of blockchain-based applications: Current status, classification and open issues,” *Telemat. Informatics*, vol. 36, no. November 2018, pp. 55–81, 2019, doi: 10.1016/j.tele.2018.11.006.

[3] X. Xu, Q. Lu, Y. Liu, L. Zhu, H. Yao, and A. V. Vasilakos, “Designing blockchain-based applications a case study for imported product traceability,” *Futur. Gener. Comput. Syst.*, vol. 92, no. October, pp. 399–406, 2019, doi: 10.1016/j.future.2018.10.010.

[4] S. Porru, A. Pinna, M. Marchesi, and R. Tonelli, “Blockchain-oriented software engineering: Challenges and new directions,” *Proc. - 2017 IEEE/ACM 39th Int. Conf. Softw. Eng. Companion, ICSE-C 2017*, no. June, pp. 169–171, 2017, doi: 10.1109/ICSE-C.2017.142.

[5] L. Bass, P. C. Clements, and R. Kazman, *Software Architecture in Practice*. 1997.

[6] X. Xu *et al.*, “The blockchain as a software connector,” *Proc. - 2016 13th Work. IEEE/IFIP Conf. Softw. Archit. WICSA 2016*, no. April, pp. 182–191, 2016, doi: 10.1109/WICSA.2016.21.

[7] F. Wessling, C. Ehmke, O. Meyer, and V. Gruhn, “Towards Blockchain Tactics: Building Hybrid Decentralized Software Architectures,” *Proc. - 2019 IEEE Int. Conf. Softw. Archit. - Companion, ICSA-C 2019*, pp. 234–237, 2019, doi: 10.1109/ICSA-C.2019.00048.

[8] J. A. Hurtado, “Toward a Scientific Method in Software Engineering (Position Paper),” p. 2, 2009.

[9] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, “Systematic mapping studies in software engineering,” *12th Int. Conf. Eval. Assess. Softw. Eng. EASE 2008*, no. February 2015, 2008, doi: 10.14236/ewic/ease2008.8.

[10] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, “Systematic literature reviews in software engineering - A systematic literature review,” *Inf. Softw. Technol.*, vol. 51, no. 1, pp. 7–15, 2009, doi: 10.1016/j.infsof.2008.09.009.

[11] P. Runeson and M. Höst, “Guidelines for conducting and reporting case study research in software engineering,” *Empir. Softw. Eng.*, vol. 14, no. 2, pp. 131–164, 2009, doi: 10.1007/s10664-008-9102-8.

[12] T. L. Alves, P. Silva, and M. S. Dias, “Applying ISO/IEC 25010 Standard to prioritize and solve quality issues of automatic ETL processes,” *Proc. - 30th Int. Conf. Softw. Maint. Evol. ICSME 2014*, no. January, pp. 573–576, 2014, doi: 10.1109/ICSME.2014.98.

[13] N. B. Harrison and P. Avgeriou, “How do architecture patterns and tactics interact? A model and annotation,” *J. Syst. Softw.*, vol. 83, no. 10, pp. 1735–1758, 2010, doi: 10.1016/j.jss.2010.04.067.

[14] R. Kazman, M. Klein, and P. Clements, “ATAM : Method for Architecture Evaluation,” *Cmusei*, vol. 4, no. August, p. 83, 2000, doi: (CMU/SEI-2000-TR-004, ADA382629).

[15] L. Dobrica and E. Niemelá, “A survey on software architecture analysis methods,” *IEEE Trans. Softw. Eng.*, vol. 28, no. 7, pp. 638–653, 2002, doi: 10.1109/TSE.2002.1019479.

[16] E. J. Chikofsky and J. H. Cross, “Reverse Engineering and Design Recovery: A Taxonomy,” *IEEE Softw.*, vol. 7, no. 1, pp. 13–17, 1990, doi: 10.1109/52.43044.

[17] J. Stafford and P. Clements, “Producing Software Architecture Documentation to Suit Your Needs,” pp. 33–33, 2007, doi: 10.1109/wicsa.2007.34.

[18] P. Clements, D. Garlan, R. Little, R. Nord, and J. Stafford, “Documenting software architectures: Views and beyond,” *Proc. - Int. Conf. Softw. Eng.*, no. January 2015, pp. 740–741, 2003, doi: 10.1109/icse.2003.1201264.

[19] L. Feng, H. Zhang, L. Lou, and Y. Chen, “for ’ ata 6 ecurity 3 rocess 3 latform of WSN,” *2018 IEEE 22nd Int. Conf. Comput. Support. Coop. Work Des.*, pp. 75–80, 2018, doi:

10.1109/CSCWD.2018.8465319.
[20] S. Aggarwal, R. Chaudhary, G. S. Aujla, N. Kumar, K. K. R. Choo, and A. Y. Zomaya, “Blockchain for smart communities: Applications, challenges and opportunities,” *Journal of Network and Computer Applications*, vol. 144. Academic Press, pp. 13–48, 15-Oct-2019, doi: 10.1016/j.jnca.2019.06.018.
[21] H. O. W. The, P. Of, I. Can, and A. Blockchain, “Leveraging Architectural Principles,” pp. 16–20.
[22] W. Viriyasitavat and D. Hoonsopon, “Blockchain characteristics and consensus in modern business processes,” *J. Ind. Inf. Integr.*, vol. 13, pp. 32–39, 2019, doi: 10.1016/j.jii.2018.07.004.
[23] T. T. A. Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi, and J. Wang, “Untangling Blockchain: A Data Processing View of Blockchain Systems,” *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 7, pp. 1366–1385, 2018, doi: 10.1109/TKDE.2017.2781227.
[24] X. Xu *et al.*, “A Taxonomy of Blockchain-Based Systems for Architecture Design,” *Proc. - 2017 IEEE Int. Conf. Softw. Archit. ICSA 2017*, no. April, pp. 243–252, 2017, doi: 10.1109/ICSA.2017.33.
[25] F. Wessling, C. Ehmke, M. Hesenius, and V. Gruhn, “How much blockchain do you need?: Towards a concept for building hybrid DApp architectures,” *Proc. - Int. Conf. Softw. Eng.*, pp. 44–47, 2018, doi: 10.1145/3194113.3194121.
[26] M. Pereira, M. Toscano, and P. Villar, “Plataformas blockchain y escenarios de uso,” p. 149, 2019.
[27] C. Marshall and P. Brereton, “Tools to support systematic literature reviews in software engineering: A mapping study,” *Int. Symp. Empir. Softw. Eng. Meas.*, no. October 2013, pp. 296–299, 2013, doi: 10.1109/ESEM.2013.32.
[28] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*, vol. 9783642290. 2012.
[29] T. Meline, “Selecting Studies for Systemic Review: Inclusion and Exclusion Criteria,” *Contemp. Issues Commun. Sci. Disord.*, vol. 33, no. Spring, pp. 21–27, 2006, doi: 10.1044/cicsd_33_s_21.
[30] F. Nasrallah, W. Zidi, M. Feki, S. Kacem, N. Tebib, and N. Kaabachi, “Biochemical and clinical profiles of 52 Tunisian patients affected by Zellweger syndrome,” *Pediatr. Neonatol.*, vol. 58, no. 6, pp. 484–489, 2017, doi: 10.1016/j.pedneo.2016.08.011.
[31] X. Xu, C. Pautasso, L. Zhu, Q. Lu, and I. Weber, “A pattern collection for blockchain-based applications,” *ACM Int. Conf. Proceeding Ser.*, 2018, doi: 10.1145/3282308.3282312.
[32] F. Wessling and V. Gruhn, “Engineering Software Architectures of Blockchain-Oriented Applications,” *Proc. - 2018 IEEE 15th Int. Conf. Softw. Archit. Companion, ICSA-C 2018*, pp. 45–46, 2018, doi: 10.1109/ICSA-C.2018.00019.
[33] J. Kramer, J. M. Van Der Werf, J. Stokking, and M. Ruiz, “A Blockchain-Based Micro Economy Platform for Distributed Infrastructure Initiatives,” *Proc. - 2018 IEEE 15th Int. Conf. Softw. Archit. ICSA 2018*, pp. 11–20, 2018, doi: 10.1109/ICSA.2018.00010.
[34] S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, and F. Y. Wang, “Blockchain-Enabled Smart Contracts: Architecture, Applications, and Future Trends,” *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 49, no. 11, pp. 2266–2277, 2019, doi: 10.1109/TSMC.2019.2895123.
[35] K. M. Kina-Kina, H. E. Cutipa-Arias, and P. Shiguihara-Juarez, “A comparison of performance between fully and partially decentralized applications,” *Proc. 2019 IEEE 26th Int. Conf. Electron. Electr. Eng. Comput. INTERCON 2019*, 2019, doi: 10.1109/INTERCON.2019.8853524.
[36] V. Reniers, B. Lagaisse, D. Van Landuyt, R. Lombardi, P. Viviani, and W. Joosen, “Analysis of architectural variants for auditable blockchain-based private data sharing,” *Proc. ACM Symp. Appl. Comput.*, vol. Part F1477, no. i, pp. 346–354, 2019, doi:

10.1145/3297280.3297316.
[37] Q. Lu, X. Xu, Y. Liu, and W. Zhang, “Design pattern as a service for blockchain applications,” *IEEE Int. Conf. Data Min. Work. ICDMW*, vol. 2018-Novem, pp. 128–135, 2019, doi: 10.1109/ICDMW.2018.00025.
[38] J. M. Medellin and M. A. Thornton, *A Discussion on Blockchain Software Quality Attribute Design and Tradeoffs*, vol. 285. Springer International Publishing, 2019.
[39] R. Viswanathan, D. Dasgupta, and S. R. Govindaswamy, “Blockchain solution reference architecture (BSRA),” *IBM J. Res. Dev.*, vol. 63, no. 2, pp. 1–12, 2019, doi: 10.1147/JRD.2019.2913629.
[40] S. K. Lo, X. Xu, Y. K. Chiam, and Q. Lu, “Evaluating Suitability of Applying Blockchain,” *Proc. IEEE Int. Conf. Eng. Complex Comput. Syst. ICECCS*, vol. 2017-Novem, no. February 2019, pp. 158–161, 2018, doi: 10.1109/ICECCS.2017.26.
[41] A. Pinna and S. Ibba, “A blockchain-based decentralized system for proper handling of temporary employment contracts,” *Adv. Intell. Syst. Comput.*, vol. 857, pp. 1231–1243, 2019, doi: 10.1007/978-3-030-01177-2_88.
[42] “Raiden Network.” [Online]. Available: https://raiden.network/. [Accessed: 01-Apr-2021].
[43] Y. Liu, Q. Lu, X. Xu, L. Zhu, and H. Yao, “Applying design patterns in smart contracts: A case study on a blockchain-based traceability application,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10974 LNCS, pp. 92–106, 2018, doi: 10.1007/978-3-319-94478-4_7.
[44] M. Barbacci, M. H. Klein, T. a Longstaff, and C. B. Weinstock, “Quality Attributes,” *Softw. Eng. Institute, Carnegie Mellon Univ. Pittsburgh, Pennsylvania,Technical Rep. C. , 1995*, vol. 18, no. 3, pp. 215–233, 1995.
[45] J. Pereira, M. M. Tavalaei, and H. Ozalp, “Blockchain-based platforms: Decentralized infrastructures and its boundary conditions,” *Technol. Forecast. Soc. Change*, vol. 146, pp. 94–102, 2019, doi: 10.1016/j.techfore.2019.04.030.
[46] M. G. Mayorga *et al.*, “No 主観的健康感を中心とした在宅高齢者における 健康関連指標に関する共分散構造分析Title,” *J. Chem. Inf. Model.*, vol. 6, no. 2, pp. 1689–1699, 2016, doi: 10.1017/CBO9781107415324.004.
[47] Z. Hoque, T. Rana, Z. Hoque, and T. Rana, “Study design and methods,” *Cost Manag. Nonprofit Volunt. Organ.*, pp. 80–91, 2019, doi: 10.4324/9780429059056-6.
[48] I. Romdhani, *Confidentiality and Security for IoT Based Healthcare*. Elsevier Inc., 2017.
[49] M. Barbacci, R. Ellison, A. Lattanze, J. Stafford, C. Weinstock, and W. Wood, “Quality Attribute Workshops (QAWs), Third Edition,” no. August, 2003.
[50] R. Wojcik and P. Clements, “Attribute-Driven Design (ADD), Version 2 . 0,” no. November, 2006.
[51] “Modelo de Hyperledger Fabric — documentación de hyperledger-fabricdocs - master.” [Online]. Available: https://hyperledger-fabric.readthedocs.io/es/latest/fabric_model.html#privacidad. [Accessed: 19-Dec-2021].
[52] “Chaincode namespace — hyperledger-fabricdocs master documentation.” [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.2/developapps/chaincodenamespace.html#channels. [Accessed: 19-Dec-2021].
[53] R. K. Yin, “CASE STUDY RESEARCH Design and Methods Second Edition.”
[54] A. Sukoco, Marzuki, and A. Cucus, “Concept of quality measurement system software based on standard ISO 9126 and ISO 19011,” *Proceeding 2012 Int. Conf. Uncertain. Reason. Knowl. Eng. URKE 2012*, pp. 105–108, 2012, doi: 10.1109/URKE.2012.6319520.
[55] E. A. G. Munoz, Y. A. O. Guzman, and J. A. H. Alegria, “ATAM-AR: Un método de Recuperación de Arquitecturas basado en ATAM,” *2015 10th Colomb. Comput. Conf. 10CCC 2015*, pp. 79–85, 2015, doi: 10.1109/ColumbianCC.2015.7333415.

[56] “GitHub - jfeliu007/goplantuml: PlantUML Class Diagram Generator for golang projects.” [Online]. Available: https://github.com/jfeliu007/goplantuml. [Accessed: 19-Dec-2021].