

# Tesis maestria

*por* Fernando Castillo

---

**Fecha de entrega:** 02-may-2022 06:47p.m. (UTC-0500)

**Identificador de la entrega:** 1826663049

**Nombre del archivo:** Borrador-Turnitin\_2.pdf (4.18M)

**Total de palabras:** 19948

**Total de caracteres:** 109533

## INTRODUCCIÓN

14

Desde su creación hace más de 30 años, el **internet ha revolucionado el mundo tal y como lo conocemos y** actualmente influye en muchos ámbitos sociales, en especial en el campo del comercio electrónico donde se realizan transacciones financieras de manera online desde la comodidad del hogar. Cabe recalcar que los métodos de pagos online mayormente utilizados por las personas en la actualidad son: tarjetas proporcionadas por bancos, transferencias bancarias, pasarelas de pagos entre los que se destaca Paypal ampliamente utilizada por los negocios e-commerce [1] y finalmente las billeteras virtuales de criptomonedas empleadas principalmente para el trading y compra/venta de activos digitales [2].

Existe una constante que no puede dejarse de lado en cualquiera de las formas de pagos online anteriormente mencionadas y es que se han detectado un aumento progresivo de fraudes, estafas y robo de información tanto personal como de las tarjetas [3], estos problemas ocasionarían que las personas dejen de confiar en realizar compras online, afectando así a millones de aplicaciones Fintech.

Por tal razón, la comunidad científica ofrece soluciones aplicadas a la seguridad en las transacciones financieras online como encriptaciones avanzadas y aprobadas mundialmente como AES o RSA para la protección de la información desde el lado del cliente, base de datos criptográficas en la nube como IOTA stronghold utilizada para la protección de secretos digitales (tokens, passwords etc) [4] y el uso de los DLT (tecnología de contabilidad distribuida) como una nueva forma de protección de datos por las ventajas que ofrece como almacenamiento distribuido, uso de métodos criptográficos que garantizan seguridad, inmutabilidad y encriptación de la información [5]. Brindar seguridad en los pagos online es de especial importancia debido a que potenciaría la confianza de los usuarios en el uso de aplicaciones Fintech.

La motivación de esta investigación surge tras las alertas de robos, fraudes y estafas en transacciones financieras online ocurridas especialmente entre los años 2020-2021 debido a la aparición del COVID-19 [6], esta pandemia mundial ha sido positiva en cierta medida para la industria de pagos digitales, según cifras de Mastercard y Americas Market Intelligence [7], se duplicó el número de personas que se volcaron a las transacciones online pasando del 45% al 83%, la explicación para este comportamiento es sencillo, las

cuarentenas impuestas por los gobiernos mundiales obligaron a las personas a realizar pagos online, potenciando indirectamente el crecimiento exponencial de las aplicaciones Fintech [8].

El COVID-19 también afectó significativamente el mercado de las criptomonedas [9] detectándose un incremento de usuarios y de mercados Fintech que se volcaron al trading de estas [10] y a su vez el interés de los hackers por encontrar vulnerabilidades en estas [11].

Los datos generados por las aplicaciones Fintech durante las transacciones financieras son de alto valor y contienen información sensible en muchos aspectos [12] y es de conocimiento público por numerosos artículos citados anteriormente, los informes de robos de información, fraudes y estafas cometidas por estas aplicaciones que no implementan un sistema de seguridad robusto [13]. Por tal motivo, detectar estas vulnerabilidades en dichas aplicaciones es un objetivo primordial para los hackers de todo el mundo.

Estas vulnerabilidades se encuentran detalladas en el trabajo realizado por los autores Kaur, LashKari & Habibi [14], donde concluyeron que, hasta en la actualidad, siguen aún existiendo vulnerabilidades humanas, tecnológicas y transaccionales presentes en aplicaciones financieras. Los mismos autores Kaur, LashKari & Habibi [15] en otro de sus artículos dieron más ejemplos de amenazas ciberneticas y las motivaciones que impulsan estos incidentes, aplicaron varias metodologías de modelado de amenazas como STRIDE, TRIKE, VAST y PASTA para mitigar ataques en diferentes aplicaciones Fintech, sin embargo, esto no bastó para mitigar por completo todas las amenazas.

Finalmente, el trabajo de los autores Huh, Cho & Kim [16] donde se implementó un sistema de encriptación de datos utilizando RSA para la protección de llaves privadas generados por Ethereum, una de las plataformas blockchain más populares actualmente.

Se evidencia que, en los trabajos anteriormente citados, muchas plataformas Fintech no cuentan con la seguridad suficiente para realizar transacciones financieras, inclusive cuando estas transaccionan con criptomonedas [17], surgiendo soluciones como los contratos inteligentes o smart contracts para la mitigación de fraudes y estafas financieras, sobresaliendo Ethereum como la más utilizada para esta labor [18] [19]. Este asunto tan importante ha sido ignorado por la mayoría de empresas desarrolladoras de software por

el afán de lanzar aplicaciones Fintech y ganar mercado en estos tiempos de pandemia [20].

El trabajo realizado por Gatteschi [21] discute las ventajas y desventajas del blockchain y concluye que esta tecnología puede ser aplicada en cualquier sector, brindando grandes ventajas al sector Fintech [22]. Sin embargo, surgen varias limitantes sobre el uso de la tecnología blockchain demostradas por los autores Gatteschi y Mesengiser & Miloslavskaya [23] que podrían ser un problema a futuro para las aplicaciones Fintech y son el rendimiento, rentabilidad y sostenibilidad con el medio ambiente.

Con respecto al rendimiento, mientras más crece la red de blockchain, mayor será el tiempo de procesamiento de la transacción, bitcoin, por ejemplo, tiene la capacidad de procesar transacciones por segundo muy bajas dependiendo del congestionamiento de la red [24] a comparación de las 65.000 transacciones por segundo reportadas por la empresa Visa en el año 2021 [25], esto afectaría negativamente a las aplicaciones Fintech debido a que las mayorías de estas, son aplicaciones móviles y requieren que estas transacciones sean rápidas y sean reflejadas al usuario en el menor tiempo posible sin afectar la usabilidad.

Con respecto a la sostenibilidad ambiental, los autores Vries & Stoll [26] y Vries [27] analizaron los daños ambientales producidos por las criptomonedas mayormente desarrollados bajo la tecnología blockchain, donde concluyeron que estos daños son exponenciales para el medio ambiente. Esta limitante ocasionaría un problema para muchas aplicaciones, incluidas las Fintech, dado a que a futuro muchas personas, empresas o instituciones gubernamentales como el gobierno de China por ejemplo [28], rechacen utilizar, apoyar o colaborar con aplicaciones desarrolladas bajo la tecnología blockchain por el daño al medio ambiente que este ocasiona.

Con respecto a la rentabilidad, será menor a medida del crecimiento del blockchain dado a que genera un abismal consumo energético debido al tiempo que a estos le toman para resolver operaciones matemáticas complejas para concatenarse a la red [29] y a su vez generan residuos electrónicos [30] donde muchas empresas han optado por la utilización de sistemas SCADA [31] para facilitar el monitoreo y tener un control más eficiente de estos residuos electrónicos pero esto solamente mitiga el problema más no lo soluciona. Estos problemas se estudiaron mejor en la investigación realizado por Vries & Stoll [26] donde cuantificaron que toda la red del bitcoin genera por año una cantidad de 30,7

kilotoneladas de residuos electrónicos, que, según estos mismos autores, esta cantidad es comparable con los desperdicios generados por equipos electrónicos pequeños del país de Holanda.

Entre las soluciones propuestas a estas limitaciones se encuentran diseñar estrategias de sostenibilidad ambiental para el blockchain propuesta por Bai & Sarkis & Cordeiro [32], los mismo autores Vries & Stoll [26] dan una solución de sustituir el sistema de minera (el protocolo Proof-of-work) en su totalidad, dado a que según los estudios de evaluación de este protocolo realizados por los autores Nair & Dorai [33] y Gemeliarana & Sari [34], concluyeron que la seguridad fue alta pero de rendimiento bajo debido al costo eléctrico alto, surgiendo de aquí propuestas como proof-of-contribution [35] o el proof-of-stake [36].

Es indiscutible que la utilización del blockchain proporciona una solución robusta, gratuita y segura, sin embargo, aplicar solamente blockchain no es suficiente, hay que implementarla en conjunto con otros métodos de seguridad [37], esta problemática surge por la variedad de tecnologías de las cuales están desarrolladas las diferentes aplicaciones que requieren protecciones tanto a nivel de servidores como de aplicación. A raíz de esto surgió IOTA como solución a los problemas de rendimiento, rentabilidad y sostenibilidad presentes en blockchain pero esta tecnología igualmente presenta sus limitaciones ocasionadas por ser una tecnología relativamente nueva [38].

Basados en las afirmaciones anteriores, la presente investigación utilizará el DLT de IOTA como solución a las problemáticas expuestas por los autores [21], [23], [29], [30] y [26] gracias a la creación de IOTA que fue la primera criptomonedas que se creó fuera del sistema blockchain [39], en su lugar utiliza Tangle que a diferencia del blockchain, solamente necesita confirmar dos transacciones de diferentes participantes para poder concatenar su transacción dentro del nodo de Tangle [40], resultando ser rentable para ser utilizado en aplicaciones Fintech debido a la rapidez en la confirmación de las transacción.

El Tangle de IOTA hace posible que no exista la necesidad de utilizar la minería como en blockchain y con esto no se afectaría al medio ambiente, en lugar de esto utiliza los propios dispositivos clientes como dispositivos móviles o un arduino [41], [42] para ser verificadores de transacciones; una de las ventajas más sobresalientes para ser utilizado en el internet de las cosas (IoT) [43], [44], [45] y en transacciones financieras debido a

que no existen comisiones (fee) [46] que se carguen a las transacciones realizadas por los clientes en aplicaciones Fintech o en protocolos ligeros para dispositivos IoT [47], [48] en monitoreos con WSN [49] por citar algunos ejemplos.

Lastimosamente, los smart contracts de IOTA actualmente se encuentra en fase beta [50], lo que impide su implementación en un ambiente de producción, alternativas como Iotex blockchain son viable para aplicaciones Fintech debido a sus bajas comisiones de transacción en comparación a otras blockchain como Ethereum o Cardano [51].

27

En base al trabajo de Taylor & otros [52] donde se realizó una revisión sistemática de literatura de las ventajas de seguridad cibernetica ofrecidas por la utilización del blockchain y en base al trabajo realizado por Ali & otros [53] donde demuestran el estado actual de la utilización de los DLT en el sector financiero, se estableció el objetivo de esta investigación que busca la implementación de los DLT en aplicaciones Fintech para el almacenamiento seguro de las transacciones financieras, tomando en cuenta que la tecnología DLT estará presente en el futuro de la ciberseguridad financiera [54].

Por todo lo anteriormente redactado y con la intención de colaborar con el objetivo 3.7 propuesto en el plan nacional de desarrollo ecuatoriano [55] que incentiva a la producción y consumo ambiental de manera responsable con el fin de incrementar la productividad de tecnologías y así combatir con la obsolescencia programada y a su vez otorgar una adecuada utilidad a la información confidencial de los usuarios así como lo estipula el art. 66, #19 de la Constitución del Ecuador [56] y la Ley de Protección de Datos (LOPD) [57] se realizó esta investigación que tiene como objetivo la implementación de tecnologías de registros distribuidos en una arquitectura de microservicios de Google Cloud utilizando las plataformas de IOTA, IOTEX, Tatum para incrementar la probabilidad de ganar disputas de pagos por delitos informáticos (estafas y fraudes de primera persona) realizadas en transacciones financieras de una aplicación Fintech, partiendo de la hipótesis de que utilizar DLT en una arquitectura de microservicios cloud disminuye casos de estafas y fraudes, otorgando ventajas como seguridad, inmutabilidad, integridad, no repudio, disponibilidad y confidencialidad de los datos generados en las transacciones financieras de una aplicación Fintech.

Para el cumplimiento del objetivo detallado anteriormente, se diseñó una aplicación web y móvil las cuales se encuentran funcionando en arquitecturas cloud bajo la plataforma de Google, son diferentes instancias las cuales proporcionan una arquitectura basada en

procesamientos complejos de eventos (CEP) [58] y microservicios. Estos microservicios proporcionan las Apis necesarias para el procesamiento de datos a través del protocolo https y la interfaz de programación API-REST y a su vez estos se encargarán de realizar el almacenamiento de los datos en los DLT utilizando IOTA que gracias a su coste cero en sus almacenamientos será utilizado cuando se trate de transacciones financieras generales, se programarán smart contracts utilizando IoTex cuando se trate de compras realizadas en el marketplace y trading de criptomonedas y finalmente se utilizará NFT con Tatum como plataforma blockchain para la identidad digital de los usuarios al realizar transacciones con tarjetas de crédito.

La investigación se realizará en un ambiente de producción controlado, tomando como caso práctico las transacciones realizadas por los usuarios en la plataforma Pay2Meta. Luego de la aplicación de las pruebas pertinentes realizadas al finalizar la implementación de la propuesta, se concluye que el Tangle de la plataforma de IOTA y de igual forma el blockchain proporcionado por IoTex y la plataforma Tatum aumentaron la seguridad y disminuyeron casos de fraudes y estafas de primera persona realizadas por los usuarios en sus transacciones financieras dentro de la plataforma Fintech. Sin embargo, también se debe tener a consideración las altas vulnerabilidades que se encuentran presentes cuando se utilizan pasarelas de pagos desarrollados por terceros. Se recomienda que estos procesos de pagos no solamente dependan de las bondades ofrecidas por blockchain o Tangle sino que también estos pagos tengan certificación PCI DSS mínimo de nivel 3, encriptación de datos de extremo a extremo y una certificación de seguridad como es la ISO 21000:2013.

La siguiente investigación está estructurada en cuatro capítulos comenzando con la introducción donde se indica al lector lo que se va a desarrollar. El capítulo uno trata sobre la elaboración del estado de arte la cual está conformada por los antecedentes históricos, conceptuales y contextuales, todos enfocados a los objetos de estudios que son las Fintech y los DLT. El capítulo dos indica los métodos y metodologías que se utilizaron en la investigación como el tipo de estudio, los enfoques, la población y muestra, métodos teóricos, empíricos y técnicas estadísticas utilizadas. El capítulo tres muestra los resultados obtenidos, fundamentados en los aportes prácticos y teóricos obtenidos en el capítulo dos. El capítulo cuatro se discute los resultados obtenidos con énfasis en aspectos como hallazgos obtenidos, su relación con otros trabajos, conclusiones y sugerencias para

trabajos futuros. Finalmente, se elaboraron las conclusiones obtenidas de la investigación realizada y la bibliografía correspondiente.

## CAPÍTULO I: ESTADO DE ARTE

El presente capítulo está destinado a la elaboración del estado de arte, el cual está compuesta por los antecedentes históricos, antecedentes conceptuales y finalmente los antecedentes contextuales. Para el desarrollo del mismo, se ha realizado una revisión sistemática de literatura (SLR) tomando en cuenta el procedimiento de la guía metodológica propuesta por Barbara Kitchenham [59].

### 1.1 Preguntas de investigación.

4

Se elaboraron las siguientes preguntas para la búsqueda de información acerca de las tecnologías de registros distribuidos y su aplicación en las aplicaciones Fintech, la tabla 1 detalla el resultado de las preguntas y dimensiones seleccionadas.

Preguntas	Dimensiones
¿Qué tecnologías de registros distribuidos se han aplicado en las Fintech para disminuir casos de delitos informáticos?	Técnicas DLT, implementaciones de DLT en Fintech, delitos informáticos
¿Cómo se implementa la metodología ABCDE en conjunto con una arquitectura de microservicios en Google Cloud para el desarrollo de sistemas Dapps?	Metodología ABCDE, microservicios cloud.
¿Cómo se implementa microservicios para registros transaccionales de coste cero con IOTA Tangle e identidad digital mediante verificación biométrica y NFT con Tatum para incrementar la probabilidad de ganar disputas financieras en casos de fraudes en transacciones financieras?	IOTA Tangle, identidad digital con NFT, Tatum.
¿Cómo se implementa smart contracts en microservicios con IOTEX blockchain para disminuir el porcentaje de casos de estafas en transacciones financieras?	Smarts contracts, IOTEX blockchain.

Tabla 1: Preguntas de investigación para el SLR

Fuente: Elaboración propia

## **1.2 Proceso de búsqueda.**

Dentro del proceso de búsqueda, se seleccionaron las siguientes bases de datos propuestas por el instructivo de titulación de la maestría:

- IEEE Xplore
- Science Direct
- Taylor and Francis.
- Springer

## **1.3 Criterios de inclusión y exclusión.**

Dentro de los criterios de exclusión se consideraron los siguientes parámetros:

- Estudios duplicados.
- Estudios que no se incluyeron en las bases de datos de selección.
- Resultados de libros, cursos-

Dentro de los criterios de inclusión se consideraron los siguientes parámetros:

- Solo estudios primarios.
- Solo investigaciones con resultados.
- Escritos en inglés y español.
- Estudios de los últimos 5 años.
- Estudios de aplicación de DLT en aplicaciones financieras o Fintech.
- Deben ser journals o conference paper.
- Temas principales: DLT y ciberseguridad.

## **1.4 Cadena de búsqueda.**

La cadena de búsqueda se elaboró en base a las preguntas de investigación y se tomó en cuenta operadores lógicos como AND y OR y se seleccionó filtrando por aspectos como el título, palabras claves, metadatos etc, quedando de la siguiente manera:

*“Cybersecurity in Fintech” and (“Distributed Ledger Technologies” or “Blockchain” or “Tangle” or “Smart Contracts” or “IOTA” or “IOTEX”)*

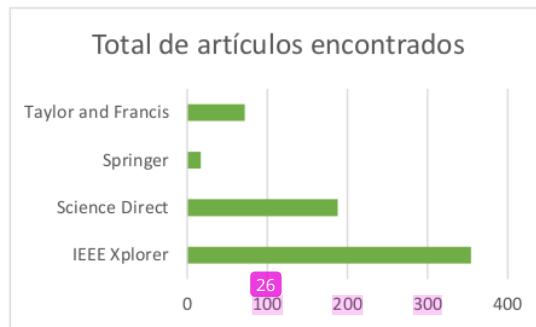
## **1.5 Selección de estudios y fase de revisión.**

Para la selección de estudios se usó las bases de datos y cadena de búsqueda previamente seleccionadas y formada, la tabla 2 muestra el resultado de este proceso.

Bases de datos	Total de artículos encontrados
IEEE Xplorer	354
Science Direct	188
Springer	17
Taylor and Francis	72
<b>Total</b>	<b>631</b>

Tabla 2: Total de artículos encontrados  
*Fuente:* Elaboración propia

En base a la tabla anterior se realizó el siguiente cuadro estadístico.



*Fuente:* Elaboración propia

En base al total de artículos encontrados en las diferentes bases de datos científicas, se realizó la fase de revisión partiendo del total de artículos, seguido de los filtrados de remover artículos duplicados, leer abstracts y títulos, aplicar criterios de exclusión e inclusión y finalmente leer el texto completo, la tabla 3 muestra el resultado de esta fase de revisión.

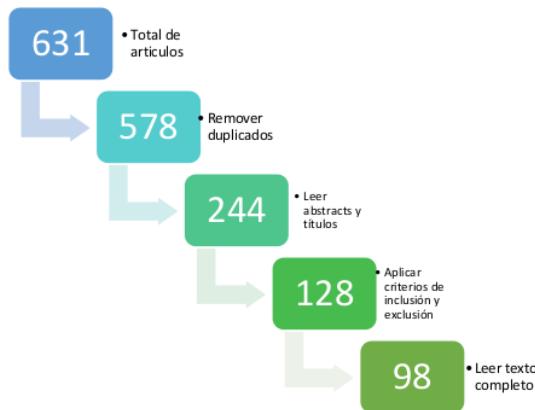


Tabla 3: Fase de revisión del SLR  
*Fuente:* Elaboración propia

## **1.6 Presentación de resultados.**

El resultado del SLR desarrollado se encuentra disponible en el Anexo 1, donde se detalla los 98 artículos científicos seleccionados para la elaboración de los antecedentes históricos, conceptuales y contextuales que se presentan a continuación.

## **1.7 Antecedentes históricos.**

Las transacciones financieras online tuvieron su nacimiento en el año 1979 gracias al inventor Michael Aldrich, pero su idea fue puesta en producción en el año 1984 cuando la señora Jane Snowball realizó una compra por VideoTex [60], uno de los primeros sistemas e-commerce que implementaron las ventas online [61] surgiendo desde este momento el término Fintech 1.0 [62].

Seguidamente por los años 90 con la aparición de las primeras aplicaciones Fintech como Paypal donde se implementaron pagos online, se da paso a las Fintech 2.0 con el objetivo de proporcionar soluciones al sector financiero y a su vez dar un gran salto en la industria tecnológica [63]. Pero a su vez, el número de estafas, fraudes y robo de información incrementaron en diversas formas por parte de hackers que aún siguen presentes en tiempos actuales tal y como lo detallan los autores [64], [14] y [15].

Con respecto a las estafas o fraudes, debido a que estas nuevas formas de pago implementadas en su mayoría por sistemas e-commerce para aquella época, no eran tecnológicamente maduras [65], muchas de las veces se firmaban contratos entre las partes interesadas para asegurarse de que nadie cometiera fraude. Cuando se menciona la palabra contrato, lo primero en que se piensa es en un papel escrito donde se establecen ciertas condiciones que, al ser leídas y aceptadas por las partes implicadas, los firmantes se comprometen a cumplir con dichas condiciones [66].

23

Desde los años 90 hasta la actualidad, se ha dado un importante avance en cuanto a la automatización, seguridad y garantías con respecto a los contratos físicos tradicionales debido al surgimiento de los smart contracts o contratos inteligentes que se llevan desarrollando desde 1997 gracias al criptógrafo Nick Szabo quién acuñó el término smart contract por primera vez, pero debido a las limitaciones tecnológicas de la época no fue factible su idea de desarrollar un sistema de pagos que llevase el concepto de los contratos tradicionales a lo digital [67]. Pero esta situación se volvió viable en el año 2009 con la aparición del bitcoin por Satoshi Nakamoto [68] gracias a la implementación de las Tecnologías de Registros Distribuidos (DLT, por sus siglas en inglés).

Antes del nacimiento del bitcoin, en el año 2008 las Fintech dieron un salto tecnológico con su versión 3.0, naciendo de aquí el término startups, que son empresas emergentes cuya característica principal es tener proyectos de rápido crecimiento y vertiginoso [69] entre ellos, proyectos de tipo Fintech que debido a la creciente popularidad del bitcoin, muchas de estas aplicaciones se enfocaron en el trading de criptomonedas y esto fue conocido como la blockchain 1.0 [70].

Como se mencionó anteriormente, la idea propuesta por Szabo de implementar contratos inteligentes para la mitigación de estafas y fraudes en su tiempo no era posible, pero gracias al surgimiento de la blockchain 2.0 en el año 2013 fue factible realizarlo. Esta nueva versión del blockchain permitió la aplicación de esta tecnología a nuevos campos de investigación con la inclusión de los smart contracts, microtransacciones, smart property, aplicaciones descentralizadas (Dapps), organización autónoma descentralizada (DAOs) y corporaciones autónomas descentralizadas (DACS) [70] [71], todas estas nuevas funcionalidades son prácticas para dar solución a posibles delitos informáticos en aplicaciones informáticas.

No cabe duda que la funcionalidad con mayor interés en el campo de las Fintech son los smart contracts dado al impulso que tuvo en el año 2014 gracias a la creación de Ethereum (plataforma open-source mayormente utilizada para programar contratos inteligentes [72]). Los smart contracts funcionan en un sistema descentralizado que no puede ser manipulado por ninguna de las partes implicadas en el contrato ni por organismos externos. El contrato se cumple por condiciones programadas, firmadas por las partes implicadas y enviada a una cadena de bloques donde se asegura inmutabilidad e indelebilidad [73] y este aspecto es conveniente para ser utilizada en compras por internet de un marketplace por citar un ejemplo práctico.

Debido a estos avances del blockchain, fue a partir del año 2015 que entidades financieras decidieran invertir en la infraestructura blockchain. Entre las entidades más destacadas se encuentran: J.P Morgan Chase que creó una división enfocada enteramente al blockchain [74] de las cuales se obtuvieron como resultado su propia blockchain privada denominada Quorum desarrollado bajo el código Ethereum [75] y en el año 2019 lanzaron su propia criptomoneda llamada JPMCoin [76]. Cabe recalcar que Quorum fue diseñado para satisfacer las necesidades de las instituciones financieras [77].

Otros casos significativos de implementación del blockchain en instituciones financieras se dio en el año 2016 por parte del Banco Santander de España, cuando inició sus pruebas en conjunto con la Empresa Ripple (creadora de la criptomoneda XRP [78]) para desarrollar servicios de pagos internacionales dando como resultado su servicio Fintech denominado Santander One Pay FX [79]; el banco The Hong Kong and Shanghai Banking Corporation (HSBC) de Reino Unido con su red privada blockchain FX Everywhere lanzada en el 2018, el Wells Fargo (EEUU) con su sistema Wells Fargo Digital Cash basado en blockchain R3, BTG Pactual (Brasil) con su token ReitBZ y Mitsubishi UFJ Financial Group (Japón) con su red privada blockchain Global Open Network y su criptomoneda MUFG Coin [80].

Pero no todo lo proporcionado por la blockchain 2.0 son ventajas, en los últimos 5 años se han elaborado artículos donde se detallan ciertos inconvenientes que a futuro serían un problema para todas las aplicaciones que utilicen blockchain y una de ellas es la rentabilidad [26]. Para que un nodo sea considerado como válido dentro de la red deberá ser aprobado por más del 50% de nodos en la red blockchain (one-cpu-one-vote) [81] lo que quiere decir que, mientras más crezca la red, mayor será el tiempo de procesar una transacción y esto ya no es tan rentable para aplicaciones desarrolladas por startups.

De igual forma sucede con las comisiones que se cobran por cada transacción en blockchain. Estas comisiones no están reguladas y varían dependiendo de varios factores como el congestionamiento de la red, el valor de la criptomoneda [82] agregando un costo adicional, muchas de las veces exageradamente alto, a las transacciones realizadas por los usuarios.

Como último inconveniente está el alto consumo de energía, esto se evidencia en los artículos elaborados por los autores [14], [15], [26], [27] & [32] y aunque existen soluciones como el Proof-of-work o Proof-of-stake para disminuir el consumo eléctrico, el problema de la sostenibilidad ambiental sigue presente en la actualidad.

Debido a estos problemas de rentabilidad, sostenibilidad y rendimiento documentados en los últimos años por la utilización de los DLT, en el año 2017 se dio paso a una próxima evolución del blockchain, conocido como la blockchain 3.0 que son redes creadas para soportar aplicaciones descentralizadas (Dapps) pero con la ventaja de tener mayor capacidad que las redes pioneras del blockchain (bitcoin y Ethereum) [83] , un producto de esta nueva tecnología es la red Cardano (criptomoneda ADA) [84].

Sin embargo, aunque estas nuevas redes que surgieron del blockchain 3.0 solucionan gran parte de los problemas ocasionados por la blockchain 1.0 y 2.0, aún siguen sin mitigarlas del todo, dando nacimiento al DLT IOTA como solución a todos los problemas mencionados anteriormente y es por esto que IOTA no es considerada un blockchain sino un Tangle basado en tecnología DAG (gráficos acíclicos dirigidos) [85].

Gracias al protocolo de consenso de IOTA, llamado FPC (Fast Probabilistics Consensus) [86], no existe distinción entre mineros y usuarios (ambos se consideran como nodos), haciendo que todos los nodos de la red sean participantes en operaciones computacionales que no requieren de mucho consumo de energía como almacenamiento y validaciones de transacciones, solucionando de esta manera el problema de la sostenibilidad ambiental dado por la tecnología blockchain.

Al no existir los mineros, ya no existe la necesidad de pagar por una comisión (fee) cada vez que se realiza una transacción. Cada transacción realizada con IOTA tiene un coste cero o también conocido como fee con valor cero [87], haciéndolo perfecto para ser utilizado en micropagos de IoT [88] o para aplicaciones Fintech.

En cuestión de la rentabilidad, IOTA no requiere que al menos el 50% de nodos de la red apruebe la transacción para unirla a la red. Cada usuario de IOTA puede realizar una transacción, pero para unirla a la red deberá validar al menos dos transacciones que antecederán a su nodo y posteriormente otro nodo validará la transacción inicial [89]. La ventaja de esto es que incrementa la rentabilidad en las transacciones realizadas en cualquier aplicación, en aspectos como velocidad, seguridad y escalabilidad.

Un aspecto negativo con respecto a IOTA, se debe a la carencia de implementación de los smart contracts, según el reporte del mes de octubre del 2021 de IOTA [50], los smart contract se encuentra actualmente en fase beta para los desarrolladores. Por lo tanto, Ethereum y Cardano son los más utilizados actualmente en la construcción de smart contracts [90].

Debido al surgimiento del COVID-19, las aplicaciones Fintech tuvieron un crecimiento considerable durante los años 2020-2021 [10]. Se registraron incrementos en la cantidad de usuarios que se inclinaron por realizar compras online e invertir en la bolsa de valores de criptomonedas [91], pero a su vez se detectaron un incremento de la ciberdelincuencia en estas aplicaciones [92], [93], [94], [95] & [96].

La implementación de los DLT en el campo de las Fintech, con todas las virtudes descritas anteriormente en esta investigación, surge como una medida extra de seguridad para dichas aplicaciones y aunque estas no logren solucionar todos los delitos informáticos por completo, es un esfuerzo adicional que la comunidad científica ofrece como protección a posibles ataques informáticos relacionados a las aplicaciones Fintech, como se muestra en el trabajo realizado por Angelis y Ribeiro da Silva [97] & Mohanta y otros [98].

Actualmente se está trabajando en la blockchain 4.0 en conjunto con la industria 4.0, que a pesar que en esta investigación no se utilizará esta tendencia, la característica de inclusión de la inteligencia artificial al blockchain [99] sería un gran avance para la mitigación de fraudes y estafas en transacciones financieras online. La figura 1 ilustra una síntesis de los antecedentes históricos elaborado para esta investigación.



Figura 1: Organización cronológica de los antecedentes de las fintech y blockchain.

Fuente: Elaboración propia

## 1.8 Antecedentes conceptuales.

### 1.8.1 Hipótesis de la investigación.

Para esta investigación se elaboraron dos hipótesis, una de investigación (Hi) y otra nula (Ho) que serán analizadas durante el desarrollo de la investigación y su validez se mostrarán en el capítulo IV en la sección de discusión de resultados obtenidos.

**Hi:** Si se utiliza las tecnologías de registros distribuidos (DLT) en arquitecturas informáticas entonces incrementaría la probabilidad de ganar disputas financieras en casos de estafas y fraudes de primera persona en transacciones financieras online de una aplicación Fintech.

**H<sub>0</sub>:** Si se utiliza las tecnologías de registros distribuidos (DLT) en arquitecturas informáticas entonces no incrementaría la probabilidad de ganar disputas financieras en casos de estafas y fraudes de primera persona en transacciones financieras online de una aplicación Fintech.

### 1.8.2 Red de categorías de las variables.

#### 1.8.2.1 Variable independiente.

- Tecnologías de registros distribuidos (DLT) en microservicios cloud.

#### 1.8.2.2 Variable dependiente.

- Disputas financieras por delitos informáticos (estafas y fraudes) en aplicaciones Fintech.

En la figura 2 se muestran las variables de investigación seleccionadas para la presente investigación.

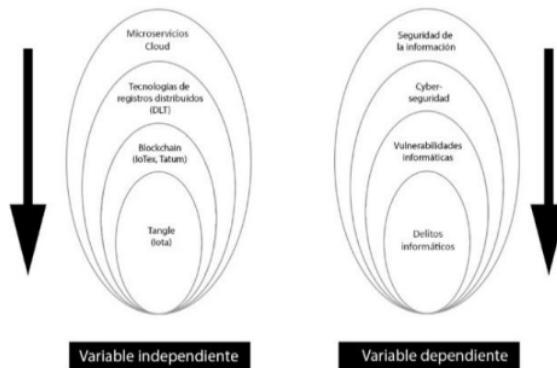


Figura 2: Variables dependientes e independientes seleccionadas

Fuente: Elaboración propia

### 1.8.3 Fundamentación teórica de la variable independiente.

Las tecnologías de registros distribuidos en microservicios cloud fue la variable independiente seleccionada para esta investigación y como estas ayudarían a la seguridad de los datos personales y financieros en aplicaciones Fintech, partiendo desde lo más general como son los microservicios cloud a lo más específico que son las tecnologías de registros distribuidos.

### 1.8.3.1 Microservicios cloud.

Las arquitecturas de microservicios implementadas en cloud computing son actualmente una de las tendencias más utilizadas para el desarrollo de software complejas [100] y distribuidas debido a su potencial de escalabilidad y seguridad para la información [101], esta afirmación viene fundamentada por los autores Hannousse & Yahiouche en su artículo [102] donde concluyeron que los microservicios nacieron con la finalidad de enfrentar la escalabilidad horizontal y vertical y los mantenimientos de los mismos mediante la utilización de patrones de diseños arquitectónicos. Sin embargo, las vulnerabilidades en sistemas basados en microservicios en aspectos como el no repudio, integridad y confidencialidad han aumentado [103], surgiendo los DLT como una nueva forma de protección para la información, esta afirmación acerca de los DLT viene sustentada por los trabajos realizados por Yang [104] y Sheng [105].

#### 1.8.3.1.1 Tecnologías de registros distribuidos (DLT).

Los DLT involucran varias tecnologías dando como resultado una base de datos que no es supervisada por ninguna entidad, es decir, es descentralizada, proporcionando la ventaja de aumentar la seguridad de los datos [106], ya que un hacker no podría acceder a esta información debido a que se encontraría distribuida en múltiples servidores. En la figura 3 se ilustra el funcionamiento de los DLT en un ambiente de sistemas DApps en el contexto de aplicaciones Fintech en un ledger centralizado en comparación con un ledger descentralizado.

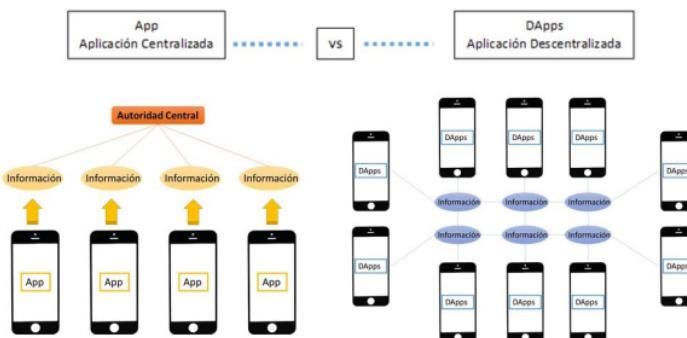


Figura 3: Ledger centralizado y descentralizado en un ambiente Fintech

Fuente: Elaboración propia

7

Entre las ventajas de los DLT, el autor Hashimy [107] detalla que mejoran la eficiencia en la distribución de la información, también reduce los costos debido a que una

institución ya no gastaría dinero en pagar servidores, sino que utilizaría el almacenamiento público de las redes de los DLT, al igual que la garantización de la inmutabilidad, trazabilidad, seguridad y transparencia de los datos almacenados.

En cuestión de su clasificación, el autor Zhuang [108] clasifica a los DLT en tres tipos, el blockchain, Tempo Ledger y DAG Ledger, en la figura 4 se muestra un organigrama elaborado por este mismo autor indicando los tipos de DLT y algunas tecnologías involucradas en ellas, es importante conocer esta clasificación debido a que en esta investigación se hará uso del blockchain y DAG como propuesta de solución y algo que llama la atención de la clasificación propuesta por Zhuang, es que coloca a IOTA como de tipo Tempo Ledger, esto entra a discusión con el autor Sadasiuvam [109] el cual indica que IOTA es un DAG al igual que HyperLedger Fabric que el autor Nawari [30] lo coloca de tipo blockchain y el autor Zhuang lo coloca de tipo DAG.

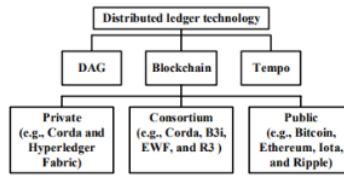


Figura 4: Clasificación de los DLT

Fuente: [108]

En la figura 5, Bahar [110] proporciona más características de los DLT, una de ellas es su amplia aplicación en diferentes campos como puede ser en la medicina, IoT, finanzas, industrias y mucho más, demostrando la gran versatilidad de esta tecnología en ser aplicadas en muchos dispositivos tecnológicos (smart watch, celulares, laptops, routers etc).

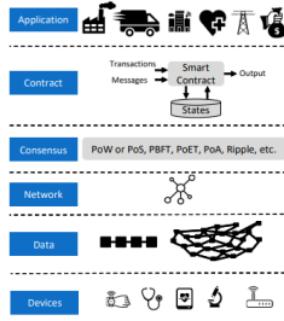


Fig. 2: DLT Stack.

Figura 5: Características de los DLT

Fuente: Adaptado de [110]

Actualmente existen dos estructuras en como la información en los DLT se distribuye dentro de la red, la primera es en forma de cadena de bloques como es el caso del blockchain y como DAG en el caso del Tangle [111] y cada una de ellas manejan sus propios protocolos de consenso entre las más destacadas se encuentran el proof-of-work,  
proof-of-stake, proof-of-contribution, FPC (IOTA) [112] y también ventajas únicas como la implementación de smart contracts muy baratas con IoTex blockchain [51] o un almacenamiento con coste cero con IOTA [43].

7

### 1.8.3.1.2 Blockchain.

El blockchain es considerado un libro de cuentas, donde cada registro es único, consensuado, distribuido y cifrado entre múltiples bloques que forman parte de la red [113]. El autor Feng [114] la define como una base de datos distribuida que utiliza el P2P ofreciendo seguridad y privacidad en las transacciones que se registran. Según Karaivanov [115], estas transacciones pueden ser económicas como costosas  
15 dependiendo del tipo de información que se deseé almacenar.

Para que la blockchain pueda funcionar requiere tener varios nodos que son considerados como mineros que se encargan de verificar estas transacciones utilizando diferentes protocolos de consenso para posteriormente validarlas y concatenarlas a la cadena de bloques [116].

#### 1.8.3.1.2.1 Tipos de Blockchain.

La blockchain se encuentra clasificada en dos ámbitos, como son los permisos de acceso y la privacidad en los usuarios para verificar transacciones dentro de la red, esta

clasificación se encuentra detallada a fondo en los trabajos realizados por [117] y [118] clasificándolos de la siguiente manera:

Permisos de acceso:

- Con permisos: Requiere autenticación para ingresar e interactuar con la red.
- Sin permisos: No requiere autenticación para ingresar e interactuar con la red.

Privacidad en transacciones:

- Transacciones públicas: cualquier persona puede ver las transacciones.
- Transacciones privadas: solo los usuarios pertenecientes a la red pueden ver las transacciones.

#### **1.8.3.1.2.2 Ventajas del blockchain.**

El autor Abdi & otros [119] detallaron en su trabajo muchas ventajas de la utilización de esta tecnología, convirtiéndola en primera opción para ser utilizada en muchos proyectos de diferentes áreas, entre las ventajas principales se destacan:

- Descentralización: las transacciones son procesados por múltiples servidores.
- Trazabilidad: los usuarios pueden estar pendientes del estado de sus transacciones.
- Transparencia: los datos no pueden ser alterados.
- Autonomía: los datos no son regulados por ninguna entidad.

#### **1.8.3.1.2.3 Plataformas blockchain**

Yang [118] y Nguyen [120] mencionan varias plataformas blockchain como:

- Bitcoin
- Ethereum
- Hyperledger Fabric
- Tatum
- IBM Blockchain
- Hydrachain
- Ripple
- R3 Corda
- Openchain

#### 1.8.3.1.2.4 IoTex.

IoTex es una infraestructura de blockchain cuya principal característica es su protocolo de consenso en tiempo real llamado Roll-DPoS [121] que permite una comunicación rápida y eficaz entre la blockchain y los millones de dispositivos conectados gracias a la web of things (WoT) [122] debido a que este protocolo utiliza un sistema de votación de minería de entre 21 a 50 delegados dentro de la blockchain y a su vez cada blockchain interactúa con diferentes dispositivos [123].

Gracias al protocolo Roll-DPoS se obtiene una red con un rendimiento significativamente más alta y de costo menor por cada transacción en comparación a otras blockchain [51], haciéndola perfecta para ser utilizado para smart contracts por su rapidez y bajo costo en comisiones. En la figura 6 se muestra un ejemplo de smart contract implementado con IoTex blockchain.



```
1 import solc from "solc";
2
3 const solidityFileString = `
4 pragma solidity ^0.4.16;
5
6 contract SimpleStorage {
7     uint storedData;
8
9     function set(uint x) public {
10         storedData = x;
11     }
12
13     function get() public view returns (uint) {
14         return storedData;
15     }
16 }
17 `;
18 const contractName = "SimpleStorage";
19 const output = solc.compile(solidityFileString, 1);
20 const abi = JSON.parse(output.contracts[contractName].interface);
21 const bytecode = output.contracts[contractName].bytecode;
```

Figura 6: Ejemplo de un smart contract con iotex

Fuente: Elaboración propia.

#### 1.8.3.1.2.5 Smart contracts.

Los contratos inteligentes o smart contracts son programas especiales que ejecutan instrucciones en redes distribuidas para almacenarlos en la blockchain y así asegurar que dicha información sea inmutable, transparente y seguras [124]. Para ejecutar un smart contract es necesario pagar una comisión, esta comisión varía dependiendo de la red blockchain que se utilice [125].

#### 1.8.3.1.2.6 Estándar ERC-721.

El estándar ERC-721 es un tipo especial de smart contract creado bajo la infraestructura de Ethereum con el objetivo de crear tokens únicos, no fungibles y no intercambiables.

Gracias a este estándar se han creado los NFT's y la identidad digital en aplicaciones informáticas [126].

#### **1.8.3.1.2.7 Estándar ERC-20.**

El estándar ERC-20 goza de una estructura pre-programada diseñada para facilitar la implementación de smart contract bajo cualquier blockchain de tipo Ethereum, por tal motivo es el más popular para implementar nuevos smart contracts [127].

#### **1.8.3.1.2.8 Solidity.**

10 Solidity es un lenguaje de programación considerada de alto nivel que hizo posible la creación de las Dapps debido a que con este lenguaje hizo posible la programación de los smart contracts que generalmente se las utiliza con el EVM de Ethereum [128].

#### **1.8.3.1.2.9 Tatum.**

Según la definición de su web oficial, Tatum “es una plataforma opensource para simplificar el desarrollo de aplicaciones DLT soportando más de 40 protocolos de blockchain y activos digitales en una misma API” [129]. Tatum admite las siguientes redes de blockchain para su desarrollo e implementación [130]:

- Mainnet. - red principal del blockchain.
- Testnet. - red de pruebas del blockchain.
- Virtual accounts. - cuentas virtuales pertenecientes a la red privada de Tatum.
- Base chain. – otras cadenas de blockchain pertenecientes a otras billeteras.

#### **1.8.3.1.3 Tangle DAG.**

Tangle es el núcleo de la tecnología IOTA así como el blockchain lo es para el bitcoin o Ethereum y a diferencia del blockchain que utiliza una cadena de bloques, Tangle utiliza los DAG (gráficos acíclicos dirigidos) [131] el cual brinda mayores ventajas en los DLT como eliminar la necesidad de utilizar mineros debido a que utiliza los propios dispositivos clientes como nodos [132], en la figura 7 se muestra gráficamente la diferencia entre la arquitectura blockchain y Tangle.

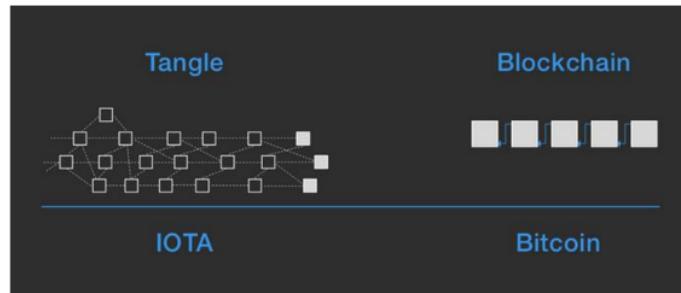


Figura 7: Arquitectura Blockchain vs Tangle

Fuente: Elaboración propia.

El funcionamiento de Tangle permite hacer transacciones offline y posteriormente concatenarse a la red, es decir, cuando una transacción es enviada a la red de Tangle, debe aprobar dos transacciones y esperar a que otra transacción la apruebe y así formará parte de la red, pero hasta eso los clientes pueden seguir enviando transacciones [133].

Entre las ventajas que ofrece Tangle, los autores [131], [132] & [133] concuerdan con la siguientes:

- Registra información de manera segura, transparente, inmutable.
- No cobra comisiones ya que no existe los mineros.
- Alta escalabilidad.
- Mejor rendimiento por la ejecución de transacciones en paralelo.
- Su arquitectura es más ligera que el del blockchain.
- Mientras más crezca el Tangle, más rápida será los procesos de verificación de transacciones.
- Descentralización y modular.

#### 1.8.3.1.3.1 IOTA

Gracias al Tangle fue posible la creación de IOTA y goza de todas las características previamente argumentadas en esta investigación como la no dependencia de mineros, alta escalabilidad, costo cero en comisiones y descentralización. Estas ventajas son posibles gracias al protocolo de consenso Fast Probabilistic Consensus (FPC), el cual según Popov & Buchanan lo definen como un protocolo de tecnología robusta, de consenso binario y baja complejidad para comunicarse entre nodos [134].

Iota es un DLT de código abierto que nació para solucionar los múltiples inconvenientes del blockchain como son problemas de rendimiento, medio ambiente y alto costos en

comisiones [135]. Su principal objetivo es la seguridad durante el flujo de la información en especial para el ambiente Iot [136].

Uno de los inconvenientes con Iota es que no es totalmente descentralizada, cuenta con un nodo origen llamado coordinador que se encarga principalmente de evitar ataques de red [137] [138] pero esto se quiere solucionar con el nuevo protocolo conocido como chrysalis con la salida de IOTA 2.0 nectar reléase [139].

La ejecución de los Smart contracts es también otro punto negativo por el momento en IOTA, pero en octubre del 2021, IOTA Foundation dió la noticia de que los Smart contract se encuentran en su fase beta [50] dando un gran paso sobre esta arquitectura.

#### **1.8.3.1.3.2 IOTA Stronghold**

Librería open-source escrita en Rust que utiliza una base de datos segura para proteger cualquier secreto digital de posibles hackers, como contraseñas, private keys etc y estas nunca sean revelados [140]. Gracias a esta librería, aumentaría la seguridad al momento de trabajar con contraseñas, llaves privadas o información sensible generadas en transacciones financieras Fintech.

#### **1.8.4 Fundamentación teórica de la variable dependiente.**

##### **1.8.4.1 La seguridad de la información.**

También conocida como S.I, nace para resguardar y proteger la información, donde se contempla un cúmulo de políticas de uso tanto preventivas como reactivas para el tratamiento de la información que se utilice dentro de alguna empresa y así evitar el acceso, utilización, divulgación o destrucción no autorizada de datos privados [141].

El objetivo principal de los S.I, según los autores Kirillova & otros [142] es garantizar de manera eficaz la protección de la información proveniente de los servicios, actividades, sistemas informáticos y comunicaciones dentro de una institución, protegiéndola contra violaciones que tengan que ver con la disponibilidad, integridad y confidencialidad de la información. Estos tres pilares se encuentran contemplados en la ISO/IEC 27001:2013 y para ponerlo en práctica las empresas identifican áreas con posibles vulnerabilidades de filtración de información, posteriormente evalúan los riesgos y finalmente otorgan los pasos necesarios para la reducción de los riesgos [143].

La detección de riesgos por lo general se los realiza en un ambiente de pruebas, el autor Wang [144] elaboró un marco tecnológico sobre la seguridad de la información realizados en un ambiente de pruebas, donde se contempla aspectos relevantes que pueden ser de utilidad en la seguridad de aplicaciones Fintech como es el no repudio, integridad, seguridad de los datos, confidencialidad, seguridad de la red y estructural, en la figura 8 se muestran más aspectos del mismo.

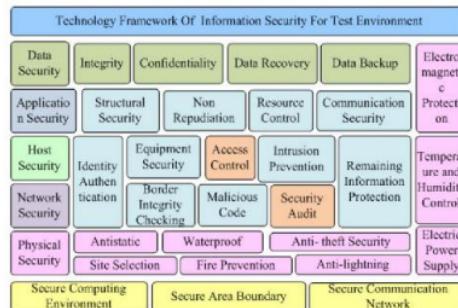


Figura 8: Framework de seguridad de la información para ambientes de pruebas

Fuente: [144]

#### 1.8.4.2 Cyber seguridad.

La seguridad informática, según la Asociación de Auditoría y Control de Sistemas de Información (ISACA), es un nivel adicional de protección para la información, con este nivel se trabaja para mitigar cualquier amenaza ya sea interna o externa durante las fases de procesamiento, transportación y almacenamiento de la información desde cualquier dispositivo [145].

Sin embargo, el autor Tirumala [146] indica que la ciberseguridad consiste en proteger sistemas donde se gestiona información privada y sensible provenientes de diferentes medios como puede ser computadoras personales, servidores, redes informáticas, dispositivos móviles entre otros, de ataques digitales por parte de hackers, que, por lo general, logran acceder a puntos que no poseen la protección suficiente para modificar, eliminar o acceder a información personal para posteriormente extorsionar a los usuarios.

Aunque a lo largo del tiempo se han implementado medidas de seguridad dentro del software, los ataques informáticos siguen ocurriendo debido al aumento de las personas en utilizar dispositivos conectados a internet [147] en especial en el ámbito IoT donde

amenazas de tipo DDos, man in the middle [148], filtración de datos, falsificación de dispositivos entre otras aún siguen presentes en la actualidad [149].

#### 1.8.4.3 Vulnerabilidades informáticas.

Las vulnerabilidades informáticas son todas aquellas que se originan cuando se produce un fallo o debilidad debido a una mala integración del software o hardware o simplemente limitaciones presentadas por la tecnología por la cual fue desarrollado el software [150]. Estas vulnerabilidades son explotadas por hackers accediendo sin autorización a diferentes sistemas informáticos mencionados anteriormente por el autor Tirumala, los atacantes una vez dentro del sistema, pueden comprometer los pilares de la seguridad de la información contemplados en la ISO/IEC 27001:2013.<sup>32</sup>

Según Tundis [151], las vulnerabilidades informáticas pueden ser de tipo teórica y real, la real es conocida como los exploits, son fallos que se encuentran en muchas aplicaciones y sistemas operativos que son solucionados en próximas versiones.

Con la llegada de la cloud computing, muchas aplicaciones, especialmente del ámbito web, migraron a estas arquitecturas, apareciendo nuevas vulnerabilidades de las cuales el autor Kumar [152] elaboró un organigrama jerárquico (ver figura 9) detallando aspectos a tener en cuenta sobre la seguridad en la cloud computing, como los requerimientos, amenazas, vulnerabilidades y contra medidas que se deben considerar al utilizar esta arquitectura.

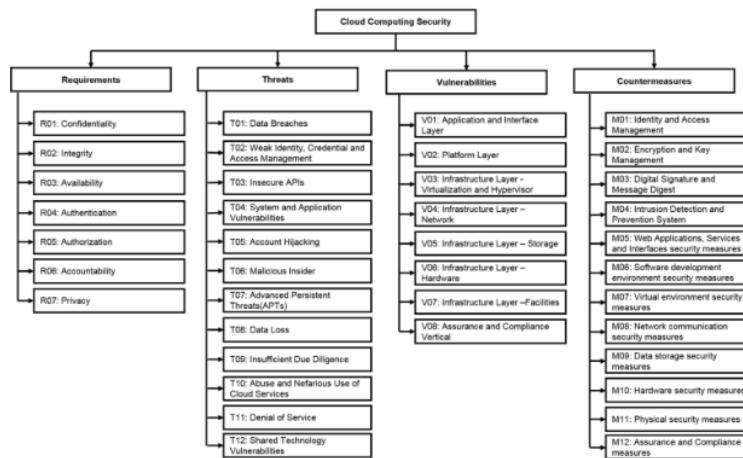


Figura 9: Seguridad en la cloud computing  
Fuente: [152]

#### **1.8.4.4 Ataques y vulnerabilidades en aplicaciones Fintech.**

A lo largo de los años, han existido muchos ataques y amenazas informáticas pero debido a que en esta investigación se centrará en las aplicaciones Fintech, se ha recapitulado aquellas vulnerabilidades que ponen en los pilares de la información dentro de estas aplicaciones.

##### **1.8.4.4.1 Carencia de cifrado de datos.**

Las aplicaciones Fintech gestionan información tanto personal como financiera de los usuarios, por tal motivo, se recomienda que toda información sensible viaje a través de la red desde las aplicaciones cliente hasta los servidores, de manera cifrada utilizando algún algoritmo de cifrado como AES, RSA, SHA256 [153] o un híbrido y en el caso de que los servidores estén en la cloud, el autor Yang [154] recomienda aplicar algoritmos de cifrado como el KP-ABE o el CP-ABE dentro del cloud storage. Aunque no existe un algoritmo de cifrado mejor o peor que otro, la selección de este algoritmo dependerá del contexto de la aplicación, por lo tanto, para la aplicación Fintech de “Pay2Meta” se ha optado por la utilización del algoritmo asimétrico RSA dado a su ventaja de utilizar una llave pública para el cifrado de datos desde las aplicaciones clientes y aunque un hacker realice un ataque de hombre de en medio (man-in-the-middle) jamás podrá desencriptar la información ya que para esto necesitaría la llave privada que se encuentra solamente en los servidores [155], en la figura 10 se observa de manera gráfica el funcionamiento del algoritmo RSA. Esta característica del RSA lo hace perfecta para ser utilizada en aplicaciones móviles, debido a que si un atacante realiza una ingeniería inversa a la app móvil solamente obtendría la llave pública y no haría nada con ese dato, caso contrario pasaría si se usase un algoritmo simétrico AES que utiliza la misma llave para cifrar y descifrar los datos [156], si un hacker la obtiene podría fácilmente desencriptar toda la información que fluya entre las aplicaciones clientes.



Figura 10: Algoritmo RSA

Fuente: Elaboración propia

#### **1.8.4.4.2 Carenica de doble factor de autenticación.**

La doble autenticación es una medida de seguridad extra implementado actualmente por muchas aplicaciones, debido a que aparte de solicitar las credenciales de email/usuario y password se requerirá de un código obtenido por aplicaciones de tercero o servicios de mensajería como SMS o email [157].

La carencia de un doble factor de autenticación en una aplicación Fintech es claramente una vulnerabilidad alta, por eso se recomienda implementarlo ya sea registrando un código PIN o solicitarlo por la aplicación de Google Authenticator [158].

#### **1.8.4.4.3 Ingeniería social.**

La ingeniería social está presente en cualquier aplicación, en especial donde se maneja comunidades de usuarios y flujo de dinero, los atacantes utilizan una serie de técnicas como el phishing para engañar a los usuarios, por lo general envían links donde se encuentran formularios solicitándole sus datos confidenciales o inyectándole malware e infectar sus dispositivos [159] y así robar información como claves, cuentas de usuarios, datos de tarjetas de crédito entre otros.

#### **1.8.4.4.4 Repudio de información.**

El no repudio de la información es uno de los principios de la seguridad informática y consiste en garantizar al receptor que el mensaje es enviado por el emisor original y no otra persona [160], este aspecto es importante en las aplicaciones Fintech, ya que tener la capacidad de demostrar que los usuarios realmente realizaron las transacciones financieras es vital para evitar fraudes o estafas.

#### **1.8.4.4.5 Carenica de seguridad en interfaces de programas de aplicación (API)**

Actualmente, la mayoría de aplicaciones se construye bajo la arquitectura de microservicios, donde la seguridad en las API es un aspecto primordial en dichas arquitecturas para proteger la confidencialidad de los datos que fluyen a través de estas API. Una API expuesta sin las seguridades suficientes son una de las principales causas de la filtración de datos confidenciales [161].

#### **1.8.4.4.6 Fraudes al utilizar tarjetas de créditos.**

Esta vulnerabilidad va d [161] con el no repudio de la información, si la aplicación Fintech no cuenta con mecanismos o algoritmos para demostrar el no repudio de los

usuarios al momento de utilizar sus tarjetas [162], claramente existirán los fraudes afectando económicamente a la empresa desarrolladora de la aplicación. Existen tres tipos de fraudes con tarjetas que se deben tener a consideración [163]:

- Fraude en primera persona: se comete cuando la persona dueña de la tarjeta realiza un pago online pero luego se dirige al banco y miente diciendo que él no realizó dicho pago.
- Fraude en segunda persona: se comete cuando un amigo o alguien cercano al dueño de la tarjeta realiza un pago online sin el consentimiento del dueño.
- Fraude en tercera persona: se comete cuando el dueño de la tarjeta desconoce por completo quien fue la persona que realizó un pago online, en este caso el dueño de la tarjeta es claramente una víctima de la ciberdelincuencia.

#### **1.8.4.4.7 Estafas al vender o comprar productos online.**

Empresas como Alibaba, Facebook, Instagram, Amazon han optado por la utilización de los marketplaces, que son aplicaciones donde muchas negocios ofertan sus productos y cualquier persona puede crearse una cuenta, provocando un aumento del índice de estafas en compras y ventas debido a que no existe un ente regulador que compruebe que estas tiendas son reales y que los productos que se ofertan sean verídicas, esta información ha sido comprobada en varios artículos elaborados entre los años 2020-2021 citados en la sección de antecedentes históricos de esta investigación.

#### **1.8.4.4.8 Metodología Agile Block Chain Dapp Engineering.**

La metodología ABCDE se fundamenta en los principios de una metodología ágil debido a que fue creada a partir de la metodología SCRUM por lo tanto utiliza varias prácticas como [164]:

- Enfoques de desarrollo interactivos e incrementales
- Historias de usuarios.
- Roles y reuniones.
- Diagrama derivado del UML para modelar eficazmente la estructura de datos de los smart contracts.
- Diagramas de secuencias para intercambiar mensajes entre las entidades del sistema.

- Utiliza dos flujos para las actividades, el primero tiene que ver con los contratos inteligentes y el segundo con los softwares que interactúan con los DLT.

Un punto a tomar en cuenta es que esta metodología considera dos tipos de integraciones, la del software entre los componentes de los DLT (smart contracts, biblioteca, estructura de datos etc) y los componentes fuera de los DLT como microservicios y aplicaciones web o móvil, naciendo de aquí un completo sistema DApp [165]. La metodología ABCDE utiliza actividades como el diseño, desarrollo, pruebas e integración con Smart contracts y software fuera de los DLT, documentar los Smart contracts utilizando diagramas para su posterior evaluación de seguridad y mantenimiento [166]. En la figura 11, se presenta de manera gráfica como es el flujo de actividades propuestos por la metodología ABCDE.

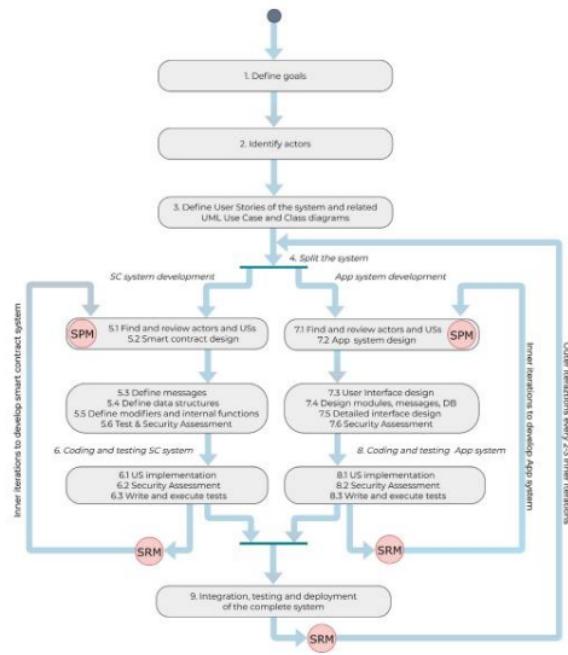


Figura 11: Proceso de la metodología ABCDE

Fuente: [164]

Con lo anteriormente mencionado por los autores acerca de la metodología ABCDE, se utilizará la misma en esta investigación porque quedó demostrado que son adecuadas para ser implementadas en aplicaciones basadas en DLT donde los requerimientos varían constantemente por la volatilidad de los DLT y también porque ofrece una metodología para la correcta utilización de los contratos inteligentes en DApps.

## **1.9 Antecedentes contextuales.**

### **1.9.1 Delimitación del contexto de investigación.**

La siguiente investigación se lo hará en un ambiente de producción controlado, tomando como caso práctico las transacciones realizadas por los usuarios en las funcionalidades de links de cobros, marketplaces y recarga de billetera ofrecidas por la plataforma Fintech “Pay2Meta”, que según su web oficial lo definen como un “eje de negocios digitales, enfocado principalmente a pequeños y medianos empresarios donde podrán comprar/vender productos o servicios, transaccionar con tarjetas de créditos y criptomonedas, poseer su propia billetera virtual, pagar servicios básicos entre otras funcionalidades” [167]. Su misión está enfocada en facilitar aspectos de negocios de los usuarios a través de procesos digitales de manera simple, rápida y segura. Su visión se centra en convertirse en el eje de negocios digitales más grande de América Latina [168], para esto, Pay2Meta requiere de la implementación de los DLT en los procesos financieros mencionados anteriormente para incrementar la seguridad de los datos transaccionales y a su vez mitigar los problemas de fraudes/estafas de primera persona detectadas en las funcionalidades de los marketplace y en la utilización de tarjetas de crédito dentro de la plataforma por parte de los usuarios. Mientras más va creciendo la plataforma, más seguridad se debe implementar tanto en el transporte como en el almacenamiento de los datos que son puntos potenciales de ataques para hackers.

### **1.9.2 Propuesta de solución.**

Desde su creación hasta la actualidad, se han detectado vulnerabilidades en las aplicaciones Fintech, especialmente entre los años 2020-2021 por la presencia del COVID-19 y aunque la comunidad científica ha realizado investigaciones para aumentar la seguridad en estas aplicaciones, aún siguen existiendo estas vulnerabilidades. La presente investigación pretende solucionar los problemas de estafas y fraudes de primera persona en aplicaciones Fintech tomando como caso práctico la plataforma Pay2Meta, por tal motivo, se diseñó una aplicación web y móvil las cuales se encuentran funcionando en arquitecturas cloud bajo la plataforma de Google, son diferentes instancias las cuales proporcionan una arquitectura basado en eventos y microservicios, estos microservicios proporcionan las APIs necesarias para el procesamiento de datos a través del protocolo https y la interfaz de programación API-REST y a su vez estos se encargarán de realizar el almacenamiento de los datos en los DLT.

La propuesta de solución consta de tres puntos:

- Crear una identidad digital, en la figura 12 se ilustra la propuesta de solución utilizando la verificación biométrica proporcionado por la plataforma MATI en conjunto con smart contract ERC-721 deployados en Iotex para posteriormente crear NFT's con Tatum blockchain y el resultado de esto almacenarlo en IOTA para asegurar su inmutabilidad.

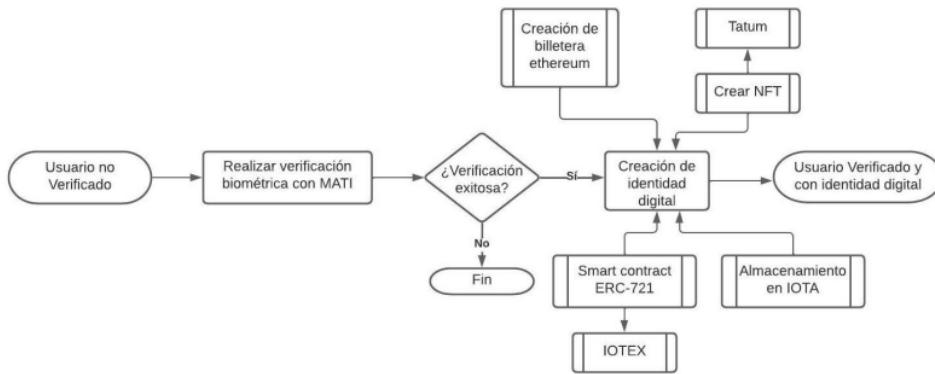


Figura 12: Diagrama de flujo del proceso de identidad digital

Fuente: Elaboración propia

- Utilizar los NFT del proceso anterior para las recargas de billeteras con tarjeta de crédito dentro de la plataforma, en la figura 13 se ilustra el proceso donde se hará uso de IOTA que gracias a su coste cero en sus almacenamientos se guardará información de transacciones financieras como ubicación, IP, dirección, últimas conexiones entre otras informaciones de los usuarios para posteriormente ser utilizado como soporte para defenderse ante un posible reclamo de fraude por parte de las entidades bancarias.

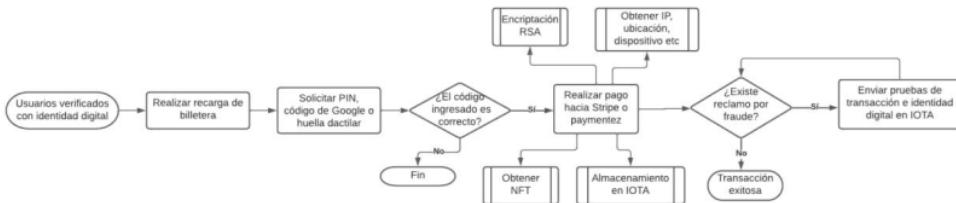


Figura 13: Diagrama de flujo del proceso de recarga de billetera

Fuente: Elaboración propia

- Finalmente, en la figura 14 se ilustra la utilización del smart contract ERC-20 para mitigar problemas de estafas utilizando IoTex blockchain cuando se trate de compras y ventas realizadas en el marketplace de productos/servicios y en el marketplace de criptomonedas donde se realizarán tradings y las transacciones financieras resultantes serán almacenadas en IOTA.

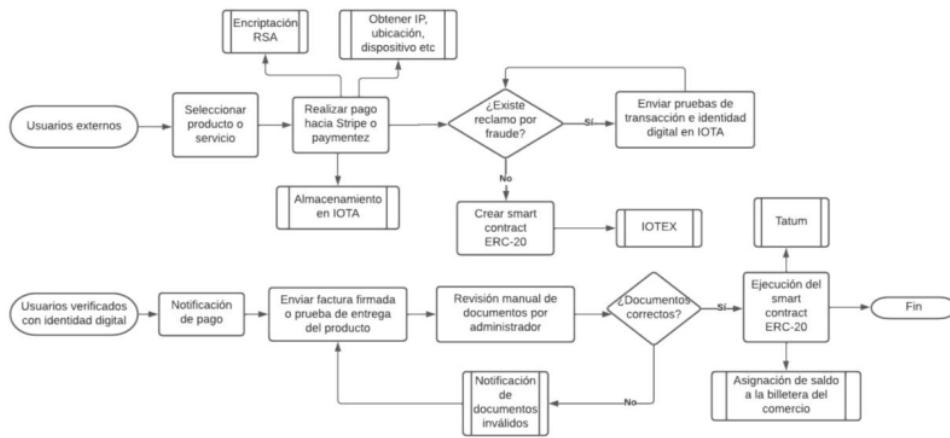


Figura 14: Diagrama de flujo del proceso del Marketplace

Fuente: Elaboración propia

Un resumen de lo anteriormente dicho se detalla en la tabla 4 donde consta las funcionalidades transaccionales donde se realizarán pruebas y se recolectarán los datos con sus respectivas propuestas de solución.

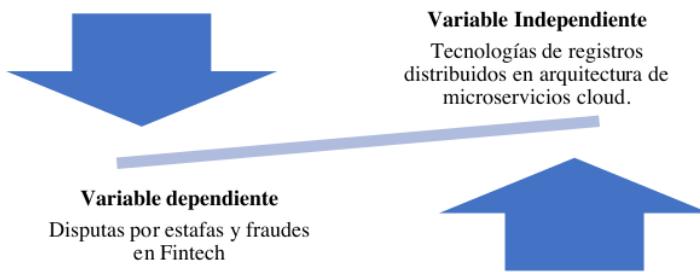
Nro	Funcionalidades transaccionales	Propuesta de solución
1	<ul style="list-style-type: none"> <li>Marketplace de productos/servicios y criptomonedas</li> <li>Links de cobros</li> </ul>	Smart contracts ERC-20 con Iotex y almacenamiento con Iota
2	Recarga de billetera con tarjetas de crédito	NFT con Tatum y almacenamiento con Iota
3	Identidad digital	Verificación biométrica con Mati, NFT con Tatum, Smart contract ERC-721 con Iotex y almacenamiento en Iota

*Tabla 4: Funcionalidades transaccionales de Pay2Meta*  
*Fuente:* Elaboración propia.

## CAPÍTULO II: MATERIALES Y MÉTODOS

### 2.1 Tipo de investigación seleccionada.

Dado a la revisión sistemática de literatura y al planteamiento de aspectos como el problema, objetivos y variables realizados anteriormente, se determinó que el nivel de profundidad para esta investigación sea de tipo correlacional debido a que se desea determinar qué tipo de correlación es (positiva, negativa o nula) entre la variable independiente que son las tecnologías de registros distribuidos en arquitectura de microservicios cloud y la dependiente que son las disputas por estafas y fraudes en aplicaciones Fintech, es decir, si se implementa los DLT en microservicios cloud, ayudaría a incrementar o no afectar la tasa de delitos informáticos por estafas y fraudes en aplicaciones Fintech. En la figura 15 se ilustra la correlación negativa entre las variables de la presente investigación.



*Figura 15: Correlación entre variables de la investigación*

*Fuente:* Elaboración propia.

### 2.2 Paradigma de investigación realizada.

Esta investigación se la realizó bajo un enfoque cuantitativo debido principalmente a que se manipularán valores numéricos como cantidad de transacciones, estafas, fraudes entre otros aspectos para su posterior medición con técnicas estadísticas y también esta investigación se ajusta al proceso cuantitativo propuesto por Sampieri <sup>1</sup> en su libro de <sup>28</sup> metodología de la investigación científica (ver figura 16) la misma que será utilizada para el cumplimiento de los objetivos propuestos.

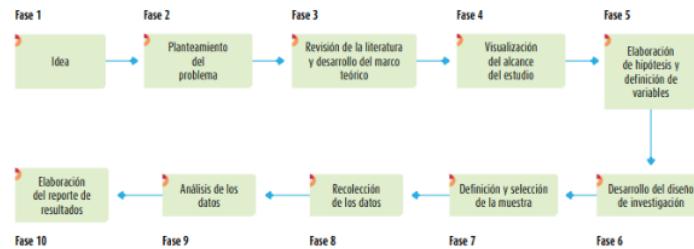


Figura 16: Fases del enfoque cuantitativo

Fuente: [169]

Finalmente, según el grado de uso de la variable, la investigación será de tipo cuasiexperimental por las razones de que en esta investigación no será posible seleccionar la población al azar sino que existirá un grupo determinado de transacciones etiquetadas potencialmente como fraudulentas o estafas, también porque la variable independiente será manipulada en un ambiente de producción con un grupo específico de sujetos obtenidos de las transacciones financieras realizadas por los usuarios de la aplicación de Pay2Meta para posteriormente verificar el comportamiento de los DLT ante distintas funcionalidades de la aplicación Fintech donde se han detectado casos de estafas y fraudes. En la tabla 5 se muestra las dos fases de experimentación que se utilizarán para esta investigación, con sus respectivas descripciones y aplicación.

Fases	Descripción	Aplicación
Baja transaccionalidad	Son aquellas transacciones que tomarán un tiempo en concatenarse a los DLT.	Se aplicarán en aquellas funcionalidades que requieran de la utilización de los smart contracts.
Alta transaccionalidad	Son aquellas transacciones que se concatenarán a los DLT ni bien termine la operación del usuario.	Se aplicarán en aquellas funcionalidades que requieran de la utilización de IOTA y NFT de Tatum.

Tabla 5: Fases de experimentación

Fuente: Elaboración propia.

### 2.3 Población y muestra de la investigación.

La población con la cual se trabajará en la presente investigación serán las transacciones detectadas como potencialmente fraudulentas o estafas realizadas por usuarios seleccionados para la fase beta de la plataforma Pay2Meta en los meses de enero, febrero, marzo y abril del 2022 y debido a que ya se tiene una cantidad fija de transacciones realizadas, se determinó que la población será finita. El tipo de muestreo será probabilístico sistemático debido a que la muestra se obtendrá de una población previamente definida por el investigador, para eso se utilizó la fórmula de tamaño de

muestra para poblaciones finitas mostrado en la figura 18, donde se estableció una población de 296 transacciones, un nivel de confianza del 99% con su respectivo valor z-score de 2.580, un margen de error de 3% y probabilidad de éxito y fracaso del 50%, los resultados de la aplicación de la fórmula se muestran a continuación:

n= muestra

N= población= 296

p= probabilidad a favor = 50%

q= probabilidad en contra = 50%

z= nivel de confianza = 2.580 (99%)

e= error de muestra = 3%

$$n = \frac{N * z^2 * p * q}{e^2 * (N - 1) + z^2 * p * q} = \frac{296 * 2.580^2 * 0.50 * 0.50}{0.03^2 * (296 - 1) + 2.580^2 * 0.50 * 0.50} = 255$$

*Figura 17: Cálculo de la muestra*

*Fuente: Elaboración propia.*

Obteniendo un resultado de 255 transacciones que deberán ser analizadas para la comprobación de la hipótesis.

#### **2.4 Método teórico utilizado.**

El método teórico seleccionado fue el hipotético-deductivo, debido a que esta investigación plantea una hipótesis y se requiere verificarla o refutarla. Entonces, analizando los pasos que conlleva este método, las mismas serán útiles para cumplir con el objetivo general de la investigación.<sup>24</sup> En la tabla 6 se ilustra el método hipotético-deductivo propuesto por el autor Rodríguez J. [170] el mismo que fue adaptado para esta investigación.

<b>Proceso del método hipotético-deductivo</b>	
<b>Observación</b>	Uso del instrumento de registro anecdótico en las muestras seleccionadas en la investigación obtenidas de las bases de datos de firebase y mysql.
<b>Elaboración de hipótesis</b>	Si se utiliza las tecnologías de registros distribuidos (DLT) en arquitecturas informáticas entonces incrementaría la probabilidad de ganar disputas financieras en casos de estafas y fraudes de

	primera persona en transacciones financieras online de una aplicación Fintech.
<b>Deducción de consecuencias</b>	Los casos de fraudes y estafas de primera persona serán menores o nulos con la implementación de los DLT.
<b>Experimentación</b>	<p>- <b>Pruebas:</b> Ejecución de los artefactos de software para estudiar la incidencia de una muestra obtenida de la población, en este caso las transacciones financieras, antes y después de la implementación de los DLT.</p> <p>- <b>Encuestas:</b> Aplicados a los usuarios que realicen compras/ventas en los marketplaces de la aplicación Fintech para conocer el nivel de satisfacción del producto obtenido y deducir si se produjeron estafas durante el proceso de compra/venta. El formato de la encuesta se ve reflejado en el anexo 2.</p>
<b>Refutación o verificación</b>	Se muestran los resultados obtenidos a través de estadística inferencial para comprobar o no la hipótesis planteada inicialmente.

Tabla 6: Proceso sistemático del método teórico utilizado

Fuente: Adaptado de [170]

## 2.5 Métodos empíricos utilizados.

31

Para esta investigación, se utilizaron los siguientes métodos empíricos:

- **Experimento**

Como se explicó en el punto 2.2, el tipo de investigación seleccionada es cuasiexperimental, por tal motivo se hará uso de este método en dos escenarios de experimentación planteadas en la tabla 5 con la población y muestra explicada en el punto 2.3. Para cada uno de estos escenarios, se diseñará e implementará diferentes arquitecturas de software donde se manipulará la variable independiente, es decir, se hará uso de distintos microservicios con DLT para posteriormente verificar el comportamiento de la variable dependiente.

- **Productos**

Se desarrollará en total tres artefactos entregables. El primer artefacto son todos los diseños de arquitecturas de microservicios en Google Cloud siguiendo la metodología ABCDE. El segundo artefacto serán los códigos fuentes

correspondientes a las funcionalidades más destacadas donde involucren smart contracts, NFT y registros con IOTA. Finalmente, el tercer artefacto serán la aplicación web y móvil desarrollados para testear las implementaciones.

- **SLR**

Se desarrollará un SLR usando la guía metodológica de B. Kitchenham para posteriormente elaborar un cuadro comparativo donde se seleccionará las tecnologías de registros distribuidos a utilizarse en la investigación.

- **Herramientas utilizadas**

Las herramientas seleccionadas para esta investigación se encuentran dividido en tres grupos tal y como se ilustra en la figura 18, el primer grupo será utilizado para el análisis de datos, siendo seleccionada la herramienta Excel y validándolo con R Studio. El segundo será utilizado para la recolección de datos, en este grupo se encuentran bases de datos como Firebase y Mysql de donde se obtendrán los registros transaccionales y Google Forms para obtener los resultados de las encuestas aplicados a los usuarios de la aplicación Fintech. El tercer grupo está enfocado a la realización de pruebas donde se encuentra JMeter que será utilizado para realizar pruebas funcionales, testeo de aplicaciones y servicios web; Postman para testeo de endpoints; Mythrill para el análisis de vulnerabilidades en Smart contracts y wireshark para el análisis de paquetes.



Figura 18: Herramientas utilizadas en la investigación  
Fuente: Elaboración propia

## 2.6 Técnicas estadísticas utilizadas.

Para el análisis de datos cuantitativos se usará la estadística inferencial para probar la hipótesis planteada y debido a que esta investigación es de tipo correlacional, se debe optar por seleccionar alguna técnica correlacional como Pearson, Spearman o Kendall, pero para esto primeramente se debe establecer si la investigación es paramétrica o no paramétrica aplicando pruebas de normalidad.

Las pruebas de normalidad aplicadas fueron las de Kolmogorov-Smirnov y Shapiro-Wilk utilizando el software estadístico SPSS en base a los datos transaccionales obtenidos de los meses de enero y febrero proporcionados por la aplicación Fintech <sup>33</sup> Pay2Meta, en la tabla 7 se ilustra el resultado de esta prueba, donde según los autores Vance & YanYan [171], cuando la población es mayor a 50 se debe optar por seleccionar los resultados de la prueba de Kolmogorov-Smirnov caso contrario seleccionar Shapiro-Wilk y en esta prueba la población o grados de libertad (gl) fue de 59 por tal motivo se seleccionará los resultados obtenidos de la prueba de Kolmogorov-Smirnov. Para conocer si los datos son normales, según la regla de Kolmogorov-Smirnov, si el nivel de significancia es mayor a 0.05 quiere decir que los datos son normales por lo tanto se deberá usar un análisis paramétrico donde se encuentra el coeficiente de correlación de Pearson, caso contrario no son normales y se debe optar por utilizar un análisis no paramétrico como pueden ser el coeficiente de correlación de Spearman o Kendall.

Pruebas de normalidad						
	Kolmogorov-Smirnov <sup>a</sup>			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
Cant_Transacciones	,133	59	,011	,934	59	,003
Tran_Frau_Est	,226	59	,000	,776	59	,000

Tabla 7: Pruebas de normalidad de Kolmogorov-Smirnov y Shapiro-Wilk

Fuente: Elaboración propia

<sup>4</sup>

Por tal motivo, según el nivel de significancia de la prueba de Kolmogorov-Smirnov mostrada en la tabla 7 que fue de 0.011 es menor a 0.05 se concluye que la técnica estadística para el análisis de datos cuantitativos que se usará para esta investigación será el coeficiente de correlación de Spearman.

## CAPÍTULO III: <sup>1</sup> RESULTADOS

En este capítulo se exponen los resultados obtenidos en el proyecto investigativo, iniciando con un SLR elaborado en el capítulo I sobre las tecnologías de registros distribuidos para seleccionar cuál de estas se utilizarán en la investigación mediante la realización de un cuadro comparativo, seguidamente de la aplicación de la metodología ABCDE para la elaboración de los sistemas Dapps sobre una arquitectura de microservicios en Google Cloud, posteriormente se detallará la implementación de IOTA, NFT y los smart contract con Iotex y finalmente se describirán las pruebas funcionales y no funcionales realizadas.

### 3.1 Selección de los DLT.

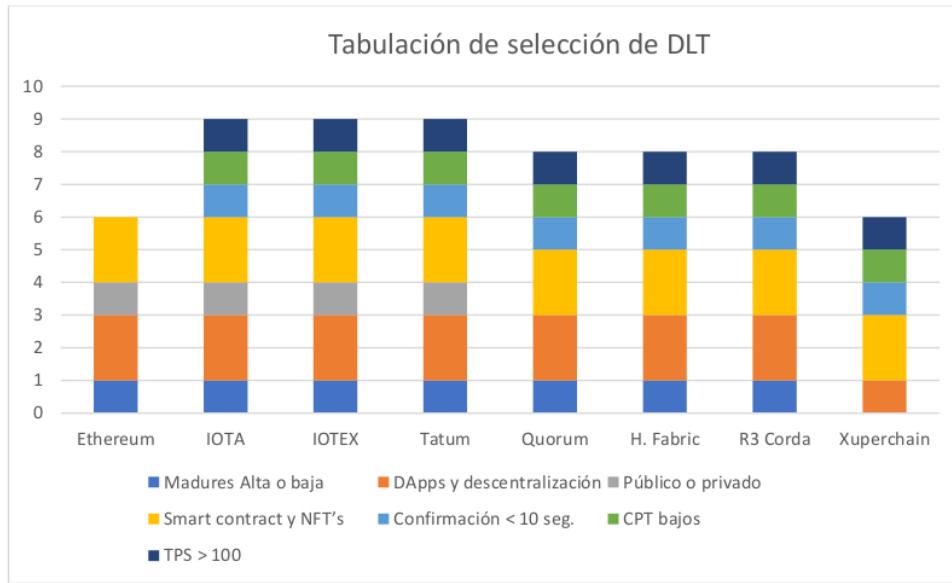
Actualmente existen varias plataformas DLT, cada una implementada en diferentes versiones de blockchain (1.0, 2.0 y 3.0) y de Tangle (DAG) ofreciendo diferentes características que podrían ser no ventajosas para ser implementadas en aplicaciones Fintech como comisiones altas, redes privadas, carencia de creación de smart contract o de NFT's entre otros. Tras una exhaustiva revisión sistemática de literatura usando la guía metodológica de B. Kitchenham cuyo resultado se ve reflejado en el Anexo 1 se ha podido realizar el siguiente cuadro comparativo ilustrado en la tabla 8 de algunas tecnologías DLT con sus características seleccionadas bajo criterios generales que ayudarían a cumplir con los objetivos propuestos para esta investigación las cuales son las siguientes:

- Tipo de red DLT.
- Madurez de la tecnología.
- Mecanismos de consenso.
- Costos de transacciones
- Aplicabilidad para Smart contract, NFT y Dapps.
- Tiempo de confirmación de transacción.
- Transacciones por segundo
- Descentralización.
- Lenguajes de programación soportados en sus APIs.
- Permisionado.

Tecnologías de Registros Distribuidos (DLT)							
Características	Ethereum	IOOTA	IOTEX	TATUM	Quorum	HYPERLEDGER FABRIC	r3.corda
<b>Madurez (años)</b>	Alto	Medio	Medio	Medio	Medio	Alto	Bajo
<b>Tecnología</b>	Blockchain	Tangle	Blockchain	Blockchain	Blockchain	Blockchain	Blockchain
<b>Mecanismo de consenso</b>	Proof-of-work (PoW)	FPC	Roll-DPoS	Depende del DLT.	Raft-consensus	Proof of Stake	TPOS
<b>Permisionado</b>	No	No	No	No	Solo validadores	Permisionado fino	Permisionado grueso
<b>Tipo de red</b>	Pública	Pública	Pública	Pública y privada	Privado y Federado	Privado y Federado	Privado y Federado
<b>Smart contracts</b>	Si	Si (fase beta)	Si	Si	Si	Si	Si (fase beta)
<b>Lenguajes</b>	Solidity	NodeJs, Rust	NodeJS, Python,	Solidity	Go, Python, Java	Kotlin, Java	Java
<b>NFT (tokens)</b>	Si	Si (fase beta)	Si	Si	Si	Si	Si (fase beta)
<b>Tiempo de confirmación</b>	14-15 segundos	< 10 segundos	5 segundos	< 5 segundos	4 a 10 segundos	5 a 10 segundos	2 a 5 segundos
<b>CPT (costos)</b>	21.000 gas	\$0.00	\$0.01	Depende del DLT.	\$0.20/h	\$0.15/h	\$0.15/h
<b>TPS (transacciones)</b>	-20 TPS	1000 TPS	2000 TPS	-100 TPS	-100 TPS	>2000 TPS	-170 TPS
<b>Dapps</b>	Si	Si	Si	Si	Si	Si	Si
<b>Descentralización</b>	Si	Si (fase beta)	Si	Si	Si	Si	No

Tabla 8: Cuadro comparativo de DLT  
Fuente: Elaboración propia

Entre los aspectos poco relevantes para la selección de los DLT en esta investigación se encuentra primeramente los lenguajes de programación para ser implementados, seguidamente del tipo de tecnología sea (blockchain, tangle o DAG) debido a que esta investigación trata sobre los DLT y todas las tecnologías mencionadas anteriormente lo son y tampoco importa los mecanismos de consenso debido a que cada DLT maneja consensos distintos y todas ofrecen tantas ventajas como desventajas. También se optó por aquellos DLT que tengan una madurez alta o media, descartando entonces XuperChain debido a que la mayoría de sus funcionalidades aún siguen en fase beta (NFT y smart contract por ejemplo) por ser relativamente nueva al ser lanzada en el año 2020. Sobre el permisionado, por cuestiones de descentralización y transparencia, se optaron por aquellos DLT que sean públicas o privadas, pero no federadas para respetar el principio de los DLT de no pertenecer a ninguna institución privada o gubernamental. Finalmente, otros aspectos a tener en cuenta son: deben contar con las funcionalidades de smarts contracts y NFT's, costo bajo en comisiones, transacciones por segundos mayores a 100, tiempo de confirmación de transacción menor a 10 segundos y soportar DApps y descentralización. En la figura 20 se ilustra la tabulación de cumplimiento de requisitos para selección bajo criterios de “si” y “no” con puntuación de 1 y 0 respectivamente.



*Figura 19: Tabulación de selección de DLT*

*Fuente: Elaboración propia*

Por tal motivo, para funcionales de registros transaccionales financieras se ha seleccionado la tecnología IOTA por su gran cantidad de TPS, coste cero en transacciones, madurez media y soporta múltiples lenguajes de programación para su integración con DApps. Para la implementación de contratos inteligentes se ha seleccionado Iotex blockchain debido a su coste de comisión relativamente baja para el deploy de dichos contratos a comparación del resto de tecnologías y también es una red pública, garantizando transparencia al momento de ejecutar los contratos. En cuestión de los NFT's para la identidad digital en las aplicaciones Fintech se ha seleccionado la plataforma Tatum debido a su bajo coste de comisión para implementar tokens no fungibles con una red blockchain a elección del programador y gracias a su red privada, las identidades digitales no serán visibles para todo el mundo.

### **3.2 Aplicación de la metodología ABCDE.**

Según lo detallado por el autor Tonelli [164] acerca del proceso de la metodología ABCDE, los pasos para el desarrollo de sistemas Dapps son los siguientes:

#### **3.2.1 Definición del objetivo del sistema.**

Implementar tecnologías de registros distribuidos en una arquitectura de microservicios de Google Cloud utilizando Blockchain, Tangle y la metodología ABCDE para incrementar la probabilidad de ganar disputas financieras en casos de estafas y fraudes de primera persona realizadas en transacciones financieras online de una aplicación Fintech.

#### **3.2.2 Identificación de actores.**

<b>Actores</b>	<b>Descripción</b>
Clientes registrados	Son usuarios registrados dentro del sistema, hacen uso de las diferentes funcionalidades del sistema interactuando como clientes.
Comercios registrados	Son usuarios registrados dentro del sistema, pero se diferencia de los clientes debido a que los comercios generan ganancias en los marketplaces.
Usuarios externos	Son usuarios que no pertenecen a la plataforma, pero realizan pagos a través de links de cobros generados por usuarios registrados.

Sistema	Es toda la arquitectura informática vista como un sistema, encargada de ejecutar eventos, enviar notificaciones, ejecutar smart contracts entre otras funcionalidades.
---------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 9: Identificación de actores del sistema

Fuente: Elaboración propia

### 3.2.3 Definir historias de usuarios, casos de usos y diagrama de clase.

Para la elaboración de las historias de usuarios se usará las plantillas proporcionadas por el marco de trabajo de SCRUM.

Historias de usuarios	
Número: 1	Usuario: Usuarios registrados.
<b>Nombre de historia:</b> Recargas de billeteras	
Prioridad: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 20	Iteración asignada: 1
<b>Programador responsable:</b> Ing. Fernando Castillo	
<b>Descripción:</b> Como usuario registrado de la plataforma quiero poder realizar recargas de billetera.	
<b>Validación:</b> El usuario cuando lo desee puede realizar recargas de billetera utilizando sus tarjetas de crédito o débito.	

Tabla 10: Historia de usuario #1

Fuente: Elaboración propia

Historias de usuarios	
Número: 2	Usuario: Usuarios externos y registrados.
<b>Nombre de historia:</b> Links de cobros	
Prioridad: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 40	Iteración asignada: 2
<b>Programador responsable:</b> Ing. Fernando Castillo	
<b>Descripción:</b> Como usuario registrado de la plataforma quiero crear links de pagos.	
<b>Validación:</b> El usuario cuando lo desee puede crear links de cobros para que usuarios externos a la plataforma puedan realizar pagos.	

Tabla 11: Historia de usuario #2

Fuente: Elaboración propia

Historias de usuarios	
<b>Número:</b> 3	<b>Usuario:</b> Usuarios y comercios registrados.
<b>Nombre de historia:</b> Marketplaces de criptomonedas y de productos o servicios.	
<b>Prioridad:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Puntos estimados:</b> 60	<b>Iteración asignada:</b> 3
<b>Programador responsable:</b> Ing. Fernando Castillo	
<b>Descripción:</b> Como comercio registrado de la plataforma quiero realizar anuncios de compras o ventas de criptomonedas y de productos o servicios para que usuarios registrados los compren.	
<b>Validación:</b> El comercio cuando lo deseé puede crear anuncios y los usuarios realizar transacciones de compra y venta en los marketplaces.	

Tabla 12: Historia de usuario #3

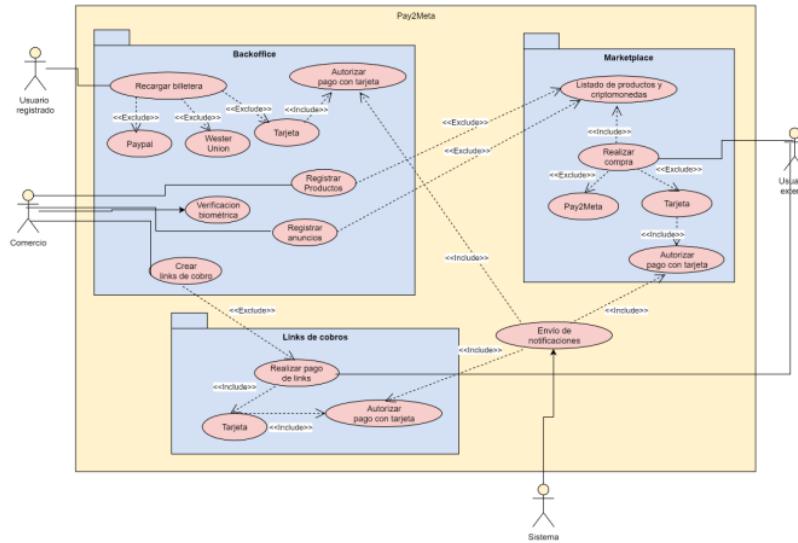
Fuente: Elaboración propia

Historias de usuarios	
<b>Número:</b> 4	<b>Usuario:</b> Usuarios registrados.
<b>Nombre de historia:</b> Identidad digital de usuarios.	
<b>Prioridad:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Puntos estimados:</b> 20	<b>Iteración asignada:</b> 4
<b>Programador responsable:</b> Ing. Fernando Castillo	
<b>Descripción:</b> Como usuario registrado de la plataforma quiero verificar mi identidad como usuario y obtener una identidad digital con NFT.	
<b>Validación:</b> El usuario cuando lo deseé puede verificar su identidad biométricamente y posteriormente generar su identidad digital con NFT.	

Tabla 13: Historia de usuario #4

Fuente: Elaboración propia

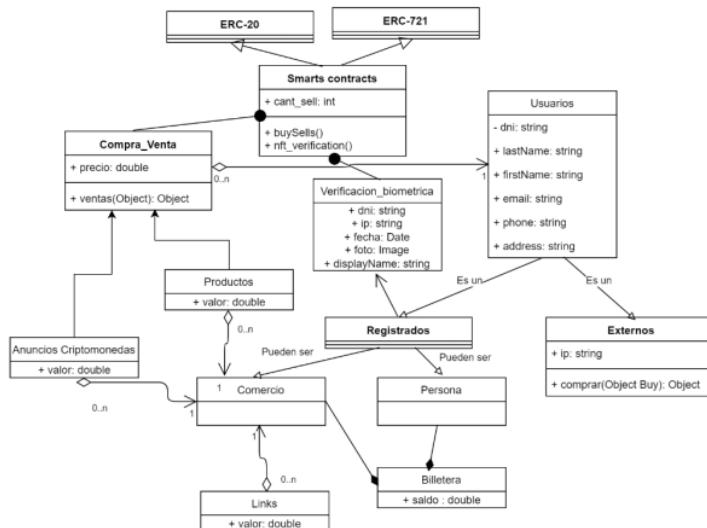
Una vez obtenidas las historias de usuarios, el siguiente paso fue diseñar los casos de usos. En la figura 20 se ilustra el caso unificado de estas historias de usuarios, con sus respectivos sub sistemas y actores.



*Figura 20: Caso de uso unificado*  
*Fuente: Elaboración propia*

6

Finalmente, se diseña el diagrama de clases tal y como se ilustra en la figura 21.



*Figura 21: Diagrama de clases*  
*Fuente: Elaboración propia*

### **3.2.4 Desarrollo de subsistemas.**

La metodología ABCDE sugiere dividir las aplicaciones en dos subsistemas, el primero hace referencia al desarrollo de las aplicaciones clientes como web o móvil y el segundo todo lo relacionado con aplicaciones con tecnologías de registros distribuidos.

#### **3.2.4.1 Subsistema de aplicaciones clientes.**

Este punto de la metodología ABCDE es dedicada al desarrollo de aplicaciones clientes <sup>34</sup> y abarca aspectos como el diseño arquitectónico, interfaces de usuario, diseño de módulos, base de datos y evaluación de seguridad de las aplicaciones.

##### **3.2.4.1.1 Diseño arquitectónico.**

La arquitectura utilizada fue una de microservicios usando Google Cloud, en la figura 22 se ilustra el diseño utilizado, el cual consta de varias aplicaciones clientes entre móvil y web, además todos los datos provenientes de estas aplicaciones clientes serán encriptadas con RSA hacia los api Gateway el mismo que se encargará de balancear y distribuir las peticiones https hacia los diferentes microservicios, estos microservicios se encargarán de desencriptar la información y volverla a encriptar con AES usando las llaves privadas obtenidas de la base de dato criptográfica de IOTA Stronghold para posteriormente almacenar la información en diferentes base de datos relationales y no relationales. Finalmente, estos microservicios realizarán envíos de notificaciones utilizando el balanceador de carga y cloud pub/sub de Google; y también conexiones con plataformas externas como Stripe, Paymentez; almacenamientos en IOTA y elaboración de smart contracts con Iotex y Tatum.

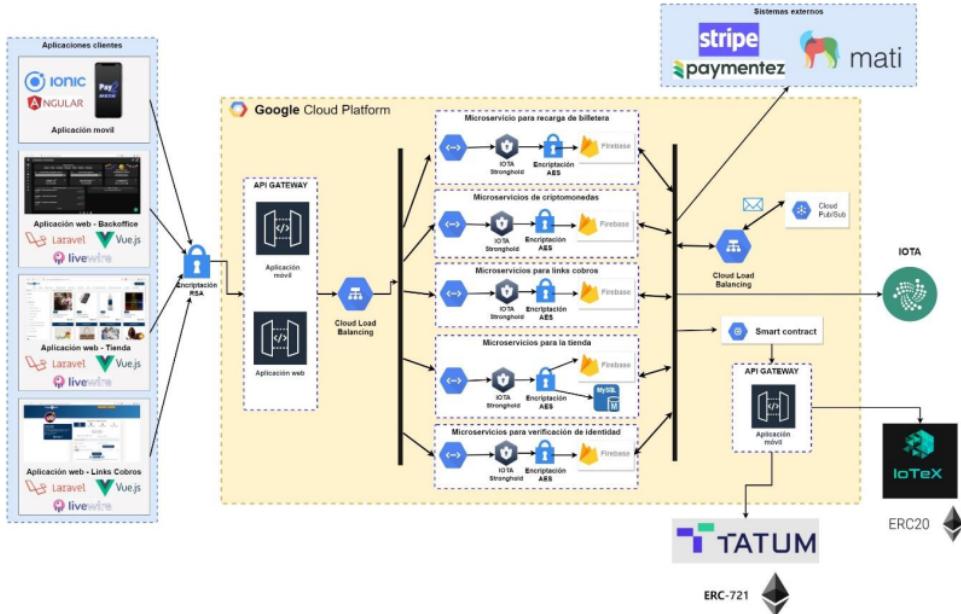


Figura 22: Diseño arquitectónico de las aplicaciones clientes  
Fuente: Elaboración propia

### 3.2.4.1.2 Diseño de las interfaces de usuario.

A continuación, se presenta las interfaces más importantes realizadas en los diferentes artefactos. La aplicación móvil ilustrada en la figura 23 fue realizada con el framework Angular 12 e IONIC 5.



Figura 23: Artefacto - aplicación móvil  
Fuente: Elaboración propia

El artefacto de links de cobros ilustrada en la figura 24 fue realizada con el framework Laravel 8 como tecnología backend, los frameworks Livewire y VueJS como tecnología frontend y firebase como base de datos.

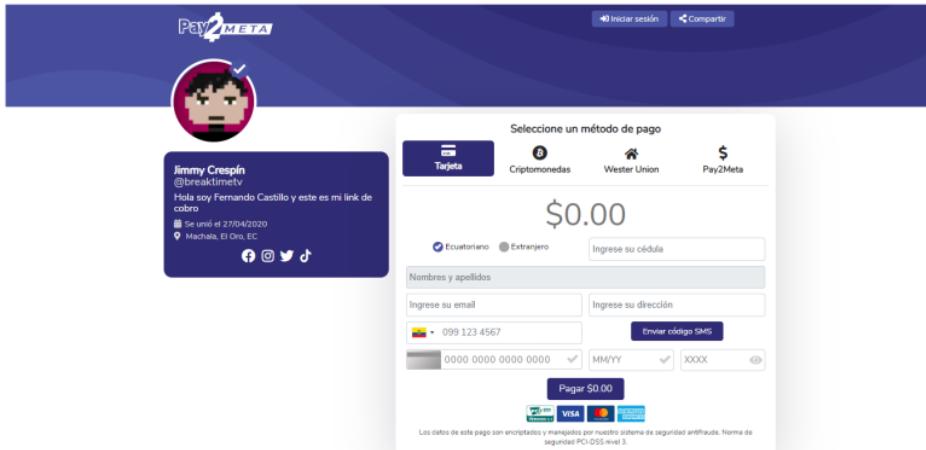


Figura 24:Artefacto - aplicación web de links de cobros

Fuente: Elaboración propia

El artefacto de marketplace ilustrada en la figura 25 fue realizada con el framework Laravel 8 como tecnología backend, los frameworks Livewire y VueJS como tecnología frontend y firebase y mysql como base de datos.

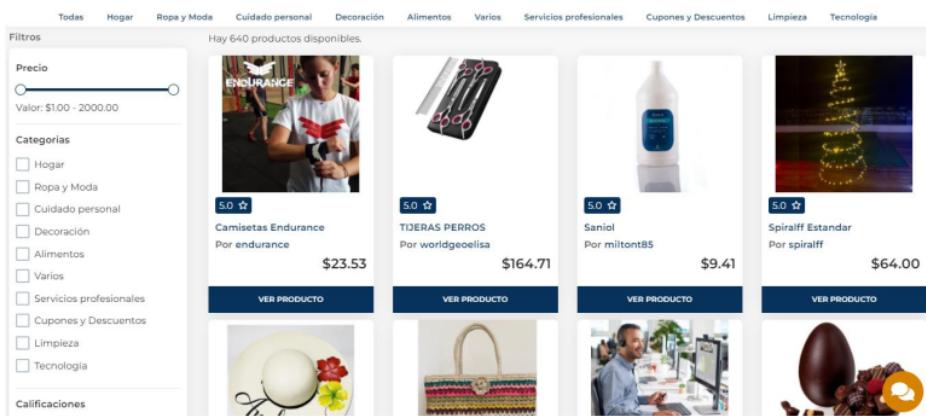


Figura 25:Artefacto - aplicación web marketplace

Fuente: Elaboración propia

El artefacto del backoffice ilustrada en la figura 26 fue realizada con el framework Laravel 8 como tecnología backend, los frameworks Livewire y VueJS como tecnología frontend y firebase como base de datos.

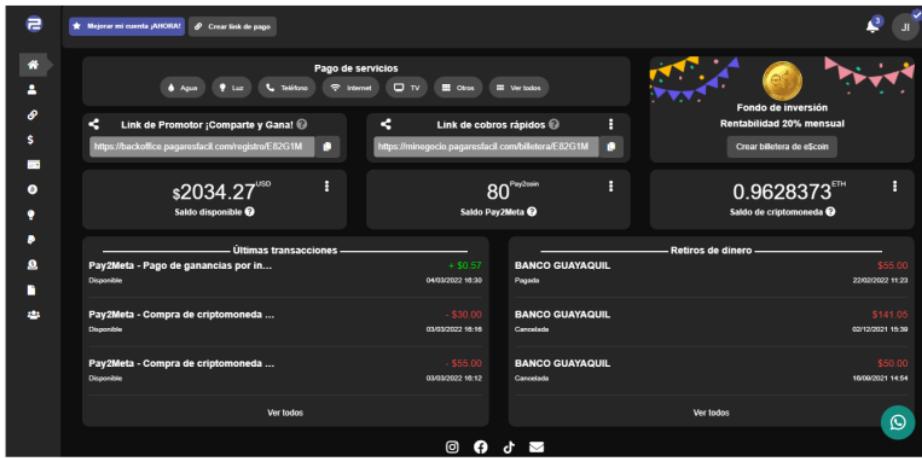


Figura 26:Artefacto - aplicación web backoffice

Fuente: Elaboración propia

El artefacto del trading con criptomonedas ilustrada en la figura 27 fue realizada con el framework Laravel 8 como tecnología backend, los frameworks Livewire y VueJS como tecnología frontend, firebase como base de datos y Tatum como tecnología DLT.

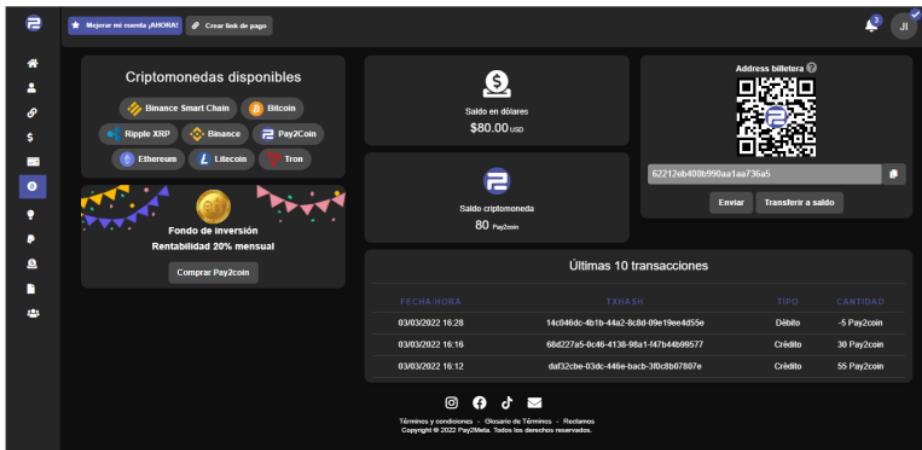


Figura 27:Artefacto - aplicación web trading criptomonedas

Fuente: Elaboración propia

### 3.2.4.1.3 Diseño de diagramas de procesos.

Fueron tres diagramas de procesos elaborados en esta investigación, el primero se ilustra en la figura 28 el cual consiste en el proceso de verificación biométrica de los usuarios, para eso se utiliza un sistema externo llamado Mati. Cuando termina la verificación, se procede a crear un QR de los datos del perfil del usuario verificado y esta imagen posteriormente se convertirá en una identidad digital utilizando un smart contract con

Iotex y NFT's de la plataforma Tatum y el resultado de esto se almacenará en IOTA para asegurar su inmutabilidad. Cabe recalcar que, para realizar esta identidad digital, los usuarios previamente deberán tener una billetera de criptomonedas de Ethereum, la misma que se puede crear en la plataforma Fintech estudiada y esta billetera deberá tener al menos un saldo de 5 centavos en ether para deployar el smart contract en Iotex.

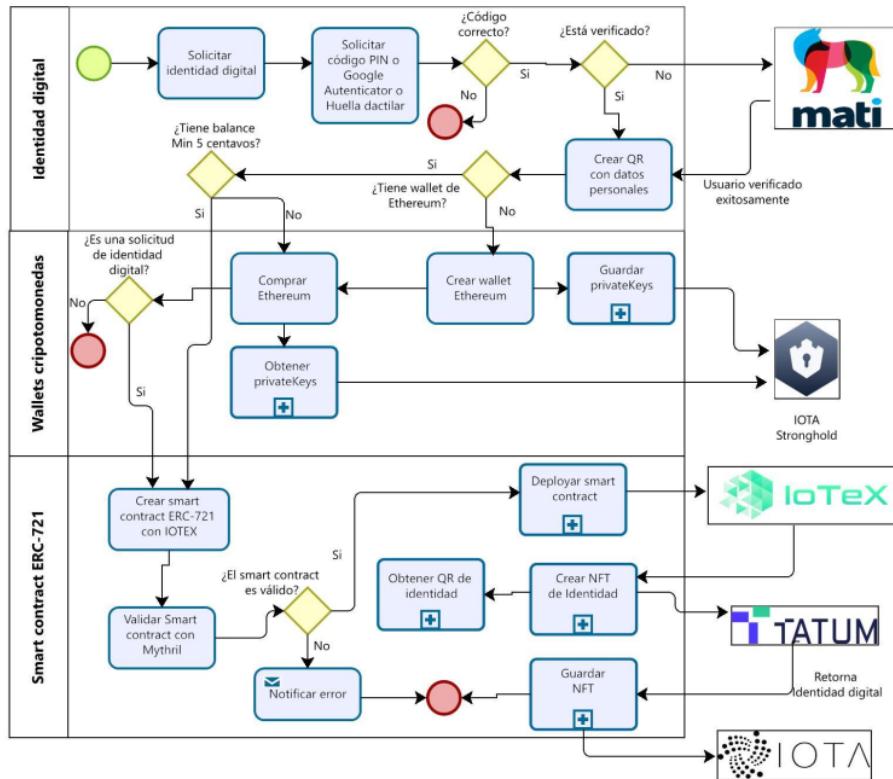


Figura 28: Identidad digital con NFT  
Fuente: Elaboración propia

El segundo proceso ilustrado en la figura 29 es acerca de la funcionalidad de recargar billetera ofrecida por la aplicación Fintech estudiada en esta investigación. Esta funcionalidad fue escogida por el motivo de que involucra pagos con tarjetas de crédito o débito de los usuarios verificados biométricamente para utilizar los NFT's creados y en conjunto con los datos de la transacción almacenadas en IOTA; tratar de disminuir los casos de fraude o si estas ocurren, tratar de ganar las disputas financieras ofreciendo esta información adicional a los bancos.

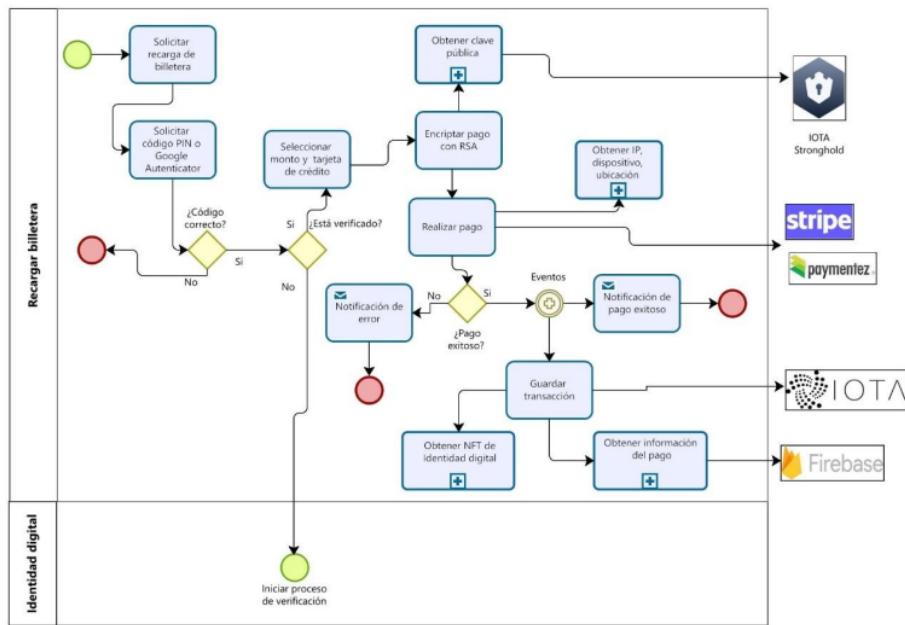


Figura 29: Proceso de recargar billetera con NFT e IOTA

Fuente: Elaboración propia

Finalmente, el tercer proceso ilustrado en la figura 30 es acerca de la funcionalidad de links de cobros y marketplace ofrecidas por la aplicación Fintech estudiada en esta investigación. Estas funcionalidades fueron escogidas por el motivo de que involucran a usuarios externos

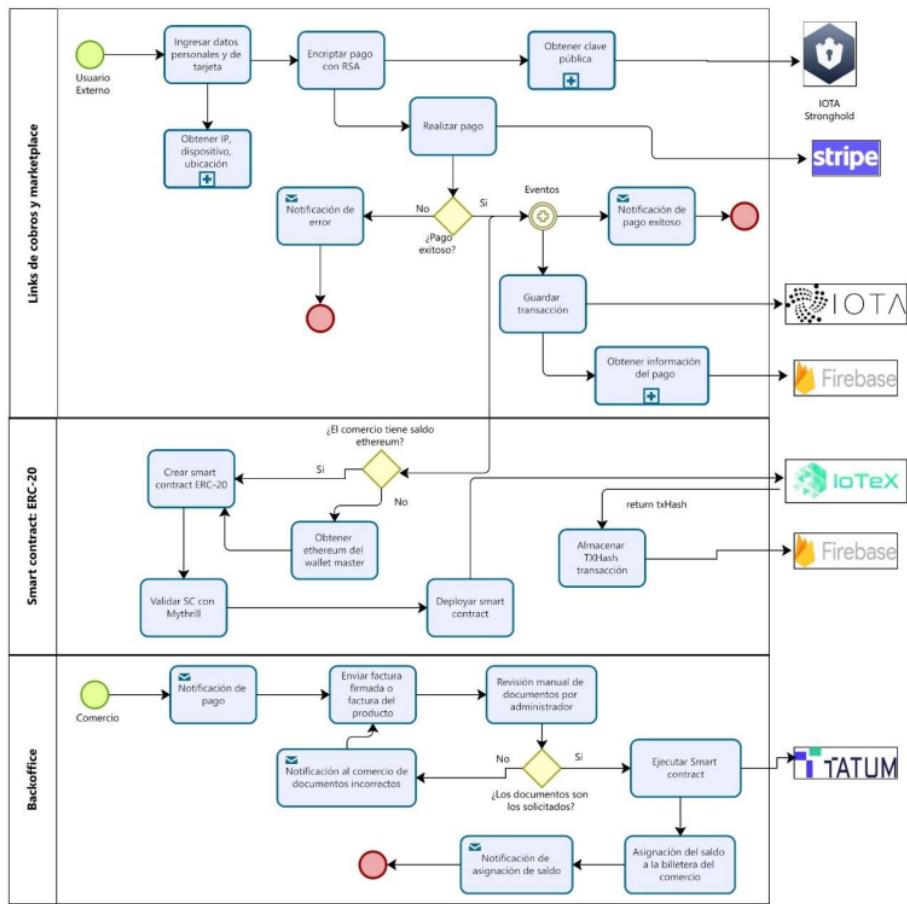


Figura 30: Proceso de compra/venta en Marketplace

Fuente: Elaboración propia

### 3.2.4.2 Subsistema DLT (IOTA, smart contracts y NFT).

#### 3.2.4.2.1 Programación de los smart contract, IOTA y NFT.

Para el registro de transacciones de coste cero con IOTA se utilizó el código de programación ilustrada en la figura 31 donde se detallan aspectos como la ip, características del dispositivo, coordenadas de ubicación, nombre del navegador, sistema operativo, fecha, hora, userId y la cantidad de la transacción que el usuario realizó en la plataforma Fintech Pay2Meta. Estos datos servirán posteriormente como prueba transaccional inmutable en caso de reportar fraude por parte de la entidad bancaria.

```

1 import { Request, Response } from 'express';
2 const { ClientBuilder } = require('@iota/client');
3
4 export async function saveTransaction(req: Request, res: Response) {
5   try{
6     const amount= req.body['amount'];
7     const type= req.body['type'];
8     const id= req.body['id'];
9     const addressReceive= req.body['addressReceive'];
10    const currency= req.body['currency'];
11    const addressSend= req.body['addressSend'];
12    const ip= req.body['ip'];
13    const ubicacion= req.body['ubicacion'];
14    const device= req.body['device'];
15    const name_browser_or_devices= req.body['name_browser_or_device'];
16    const origin= req.body['origin'];
17    const os_version= req.body['os_version'];
18    const userIdSend= req.body['userIdSend'];
19
20    const idWalletSend= req.body['idWalletSend'];
21    const idWalletReceive= req.body['idWalletReceive'];
22    let index = new TextEncoder().encode(id);
23
24    let createdAt = new Date().toLocaleString("es-ES", {timeZone: "America/Guayaquil",
25      year: 'numeric',month: '2-digit',day: '2-digit',hour: '2-digit',minute: '2-digit',second: '2-digit'});
26
27    let json= JSON.stringify({"ubicacion":ubicacion,"device":device,"name_browser_or_device":name_browser_or_device,
28    "origin":origin,"os_version":os_version,"idWalletReceive":idWalletReceive,"idWalletSend":idWalletSend,"userIdSend":userIdSend,
29    "amount": amount,"type": type,"id": id,"fecha": createdAt,"concepto":concepto,"addressReceive":addressReceive,"currency":currency,
30    "addressSend":addressSend,"ip":ip});
31
32    let data_encode = new TextEncoder().encode(json);
33    let typeNetwork ="mainnet";
34    let url="https://chrysalis-nodes.iota.org";
35    const client = new ClientBuilder()
36      .network(typeNetwork)
37      .node(url)
38      .build();
39    const responseIota = await client.message()
40      .index(index)
41      .data(data_encode)
42      .submit();
43
44    let url_iota= "https://explorer.iota.org/mainnet/message/"+responseIota["messageId"];
45    res.status(200).send({code: 200, url: url_iota});
46
47  }catch (error) {
48    res.status(200).send({code: 400, message: `Ocurrio un error: ${error}`})
49  }
50 }

```

Figura 31: Código de programación para almacenar una transacción en IOTA

Fuente: Elaboración propia

Con respecto a los smart contract ERC-20 se utilizó el lenguaje de programación solidity, en la figura 32 se ilustra una porción del código utilizado para las transferencias de criptomonedas de Ethereum cuando se cumple los acuerdos establecidos por el comercio y los compradores en los marketplaces de productos y de criptomonedas disponibles en la aplicación Fintech Pay2Meta.

```
● ● ●

1 pragma solidity ^0.8.0;
2 import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
3 contract Contrato is ERC20{
4     mapping(address => uint256) public amount;
5     uint256 totalAmount;
6     string tokenName;
7     string tokenSymbol;
8     uint256 decimal;
9     constructor() public{
10     totalAmount = 10000 * 10**18;
11     amount[msg.sender]=totalAmount;
12     tokenName="Ethereum";
13     tokenSymbol="ETH";
14     decimal=18;
15 }
16 function totalSupply() public view returns(uint256){
17     return totalAmount;
18 }
19 function balanceOf(address to_who) public view
20 returns(uint256){
21     return amount[to_who];
22 }
23 function transfer(address to_a,uint256 _value) public
24 returns(bool){
25     require(_value<=amount[msg.sender]);
26     amount[msg.sender]=amount[msg.sender]-_value;
27     amount[to_a]=amount[to_a]+_value;
28     return true;
29 }
30 }
```

Figura 32: Código de programación para smart contract ERC-20

Fuente: Elaboración propia

Con respecto a los smart contract ERC-721 se utilizó el lenguaje de programación python, en la figura 33 se ilustra una porción del código utilizado para la creación del NFT.

```

File Edit Selection View Go Run Terminal Help
NFT_PROYECTO ...
create_collectible.py X set_tokenuri.py
nft > scripts > simple_collectible > create_collectible.py ...
1 #!/usr/bin/python3
2 from brownie import SimpleCollectible, accounts, network, config
3 from scripts.helpful_scripts import OPENSEA_FORMAT
4
5 sample_token_url = "https://pay2meta.com/public/nft_100.json"
6
7 def main():
8     dev = accounts.add(config["wallets"]["from_key"])
9     print(network.show_active())
10    simple_collectible = SimpleCollectible[len(SimpleCollectible) - 1]
11    token_id = simple_collectible.tokenCounter()
12    transaction = simple_collectible.createCollectible(sample_token_url, {"from": dev})
13    transaction.wait(1)
14    print(
15        f"Listo, tu NFT fue generado exitosamente {token_id}"
16        f"\n{OPENSEA_FORMAT.format(simple_collectible.address, token_id)}"
17    )
18
19
20

```

Line 20, Col 1 | Spaces: 4 | UTF-8 | LF | Python | 3.9.5 64 bit | Go Live | R | C |

Figura 33: Código de programación para smart contract ERC-721

Fuente: Elaboración propia

### 3.2.4.2.2 Análisis de seguridad de los códigos de smart contracts.

Se utilizó la herramienta Mythril en su versión python que fue elaborado por Ethereum y Quorum para analizar la seguridad de los EVM bytecode de los smart contracts [73], en este caso los smart contracts programados con el estándar ERC-20 en el caso del marketplace y el estándar ERC-721 en el caso de la identidad digital, en la imagen 34 se ilustra el resultado de la validación del smart contract utilizado en los marketplaces y en la imagen 35 se ilustra el resultado del análisis del smart contract utilizado para la elaboración de los NFT.

```

> myth a C:\xampp\htdocs\nft_proyecto\marketplace_SC.sol -t 3
===== protected contract =====
SWC ID: 101
Security: High
Contract: Marketplace
Function name: transfer()
PC address: 543
Estimated Gas Usage: 454 - 888
The contract not can be violated by anyone.
Anyone can't violated this contract.

```

Figura 34: Análisis de seguridad del SC de Marketplace utilizando Mythril

Fuente: Elaboración propia

```
> myth a C:\xampp\htdocs\nft_proyecto\nft_SC.sol -t 3
==== protected contract ====
SWC ID: 104
Security: High
Contract: IdentidadDigital
Function name: createNFT()
PC address: 543
Estimated Gas Usage: 2100 - 2400
The contract not can be violated by anyone.
Anyone can't violated this contract.
```

Figura 35: Análisis de seguridad del SC de identidad digital utilizando Mythril

Fuente: Elaboración propia

### 3.2.5 Integración, pruebas y despliegue del sistema completo.

Para la protección de secretos digitales como llaves privadas, claves API, tokens entre otros se hará uso de la herramienta IOTA Stronghold, para esto se es necesario tener instalado en el equipo el lenguaje de programación Rust y posteriormente el paquete “cargo” que se encargará de ejecutar las librerías necesarias para el funcionamiento de Stronghold, en la figura 36 se ilustra el comando de instalación del paquete cargo.

```
fernando_castillo@instance-backoffice-web-vm:~/apps/proyecto$ cargo install --path .
```

Figura 36: Instalación del paquete Rust Cargo

Fuente: Elaboración propia

Una vez instalado el paquete cargo, el siguiente paso es instalar la herramienta stronghold utilizando el siguiente comando “cargo run –stronghold”, posteriormente se procede a realizar el almacenamiento de las claves privadas hacia la base de datos criptográfica de IOTA Stronghold, en la figura 37 se ilustra el comando para realizarlo, se puede utilizar un texto plano con todas las claves a encriptar o directamente escribir un arreglo de claves.

**Comando utilizado:** `stronghold encrypt --plain privateKey.txt --record_path "/home/fernando_castillo/apps/proyecto" --pass Fernando_1994.-`

```
fernando_castillo@instance-backoffice-web-vm:~/apps/proyecto$ stronghold encrypt --plain privateKey.txt --record_path /home/fernando_castillo/apps/proyecto --pass Fernando_1994.-
```

Figura 37: Comando para encriptar datos con IOTRA Stronghold

Fuente: Elaboración propia

Se utilizaron varias instancias de tipo Google Cloud y App Engine en la figura 38 se ilustra dos instancias para el almacenamiento de las aplicaciones web, eventos, notificaciones utilizadas para la aplicación Fintech Pay2Meta.

The screenshot shows the Google Cloud Platform Instances page. At the top, there are buttons for 'CREAR INSTANCIA', 'IMPORTAR VM', 'ACTUALIZAR', 'INICIAR/REANUDAR', 'DETENER', 'OPERACIONES', and 'MOSTRAR PANEL DE INFORMACIÓN'. A message at the top left says: 'La instancia "tienda-pagaresfacil" está sobreutilizada. Considera la posibilidad de cambiar al tipo de máquina custom (1 CPU virtual, 2.75 GB de memoria). Más información' with a 'DESCARTAR' button. Below this, there's a section titled 'INSTANCIAS PROGRAMA DE LA INSTANCIA' with a note: 'Las instancias de VM son máquinas virtuales altamente configurables para ejecutar cargas de trabajo en la Infraestructura de Google. Más información'. A filter bar allows entering a name or value for properties. The main table lists the following instances:

Estado	Nombre	Zona	Recomendaciones	En uso por	IP interna	IP externa	Conectar
OK	Instance-bacoffice-web-vm	us-west4-a		10.182.0.2 (nec)	34.125.150.227	SSH	⋮
OK	Instancia-bacoffice-pagar-es-facil	us-central1-a	Ahorra \$13/mes	10.128.0.5 (nec)	34.70.44.101	SSH	⋮
OK	Instancia-landing-pagar-es-facil	us-central1-a		10.128.0.8 (nec)	34.70.246.57	SSH	⋮
OK	landing-cripto-vm	europe-west3-a		10.156.0.2 (nec)	Ninguna	SSH	⋮
OK	odoo-1-m	us-central1-b		10.128.0.2 (nec)	34.67.172.168	SSH	⋮
OK	private-tienda-pagar-es-facil	us-central1-a		10.123.0.7 (nec)	35.226.75.128	SSH	⋮
OK	tienda-pagaresfacil	us-central1-a	Aumentar rend.	10.123.0.3 (nec)	35.226.102.0	SSH	⋮
OK	wordpress-1-vm	us-west4-b		10.182.0.4 (nec)	34.125.110.208	SSH	⋮
OK	wordpress-prueba-plugin-2-vm	us-west4-a		10.182.0.3 (nec)	Ninguna	SSH	⋮

Figura 38: Instancias de Google Cloud

1  
Fuente: Elaboración propia

En cuestión de los microservicios, en la figura 39 se ilustra varios de los microservicios implementados con Google Cloud y Firebase cloud functions, estos fueron deployados con los lenguajes de programación NodeJs 14 para funcionalidades transaccionales y Solidity 0.0.8 para cuestión de smart contracts y NFT.

The screenshot shows the Firebase Functions page for the project 'BackServicesPagos'. On the left, there's a sidebar with options like Authentication, Firestore Database, Realtime Database, Storage, Hosting, Functions, Machine Learning, Crashalytics, and Extensions. The main area is titled 'Functions' and shows a table of deployed functions:

Función	Activador	Region	Entorno de ejecución	Memoria	Tiempo de espera
bot	Solicitud <a href="https://us-central1/backservicespagos.cloudfunctions.net/">https://us-central1/backservicespagos.cloudfunctions.net/</a>	us-central1	Node.js 12	256 MB	1 min
consultas	Solicitud <a href="https://us-central1/backservicespagos.cloudfunctions.net/">https://us-central1/backservicespagos.cloudfunctions.net/</a>	us-central1	Node.js 14	1 GB	5 min
consultas2	Solicitud <a href="https://us-central1/backservicespagos.cloudfunctions.net/">https://us-central1/backservicespagos.cloudfunctions.net/</a>	us-central1	Node.js 14	1 GB	5 min
cryptos	Solicitud <a href="https://us-central1/backservicespagos.cloudfunctions.net/">https://us-central1/backservicespagos.cloudfunctions.net/</a>	us-central1	Solidity 0.8.0	1 GB	5 min
ext-firestore...	document.write <a href="#">atencionComisiones/document</a>	us-central1	Node.js 10	256 MB	1 min
ext-firestore...	document.write <a href="#">bankAccount/document</a>	us-central1	Node.js 10	256 MB	1 min
ext-firestore...	document.write <a href="#">consultarUnDeudor/document</a>	us-central1	Node.js 10	256 MB	1 min
ext-firestore...	document.write <a href="#">donave/document</a>	us-central1	Node.js 10	256 MB	1 min

Figura 39: Microservicios con Firebase cloud functions

Fuente: Elaboración propia

La aplicación de IOTA se ve reflejado en la figura 40 donde se almacenó el pago de una transacción realizada en la aplicación Pay2Meta, el resultado se puede observar en el siguiente link que proviene de la Mainnet de IOTA:

[https://explorer.iota.org/mainnet/message/08aba58ca3bbf5879cbe9368cc02e7ad9f1d392abf4c70495f1aaa277632b8b  
2abf4c70495f1aaa277632b8b](https://explorer.iota.org/mainnet/message/08aba58ca3bbf5879cbe9368cc02e7ad9f1d392abf4c70495f1aaa277632b8b)

Figura 40: Almacenamiento transaccional con IOTA  
Fuente: Elaboración propia

Con respecto a las encriptaciones, los DLT encriptan la información que circulan por la red WLAN pero no encriptan la información que proviene por la red LAN, por tal motivo se utilizó la librería de nodejs llamada jsencrypt para realizar encriptaciones de tipo RSA desde las aplicaciones clientes, para esto fue necesario que cada usuario tenga sus propias claves públicas y privadas, en la figura 41 se ilustra la manera de como generar una llave privada utilizando Openssl.

```
fernando_castillo@instance-backoffice-web-vm:~$ openssl genrsa -out rsa_1024_priv.pem 1024
```

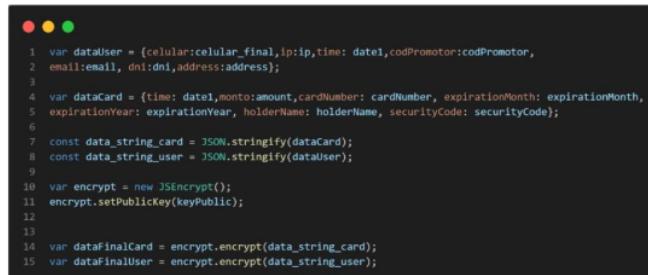
Figura 41: Generación de una llave privada RSA con openssl  
Fuente: Elaboración propia

Para la generación de la llave pública, en la figura 42 se ilustra la manera de como generar una llave pública utilizando igualmente openssl.

```
fernando_castillo@instance-backoffice-web-vm:~$ openssl rsa -pubout -in rsa_1024_priv.pem -out rsa_1024_pub.pem
```

Figura 42: Generación de una llave pública RSA con openssl  
Fuente: Elaboración propia

Finalmente, un ejemplo de encriptación desde el cliente se ve ilustrado en la figura 43 donde se utiliza la librería JSEncrypt conjuntamente con la llave pública generada para el usuario, se encripta con RSA información de la transacción como ip, datos del usuario e información financiera de la tarjeta de crédito. Esta información posteriormente será desencriptada por el servidor utilizando la llave privada del usuario almacenada en IOTA Stronghold.



```
1 var dataUser = {celular:celular_final,ip:ip,time: date1,codPromotor:codPromotor,
2 email:email, dni:dni,address:address};
3
4 var dataCard = {time: date1,monto:amount,cardNumber: cardNumber, expirationMonth: expirationMonth,
5 expirationYear: expirationYear, holderName: holderName, securityCode: securityCode};
6
7 const data_string_card = JSON.stringify(dataCard);
8 const data_string_user = JSON.stringify(dataUser);
9
10 var encrypt = new JSEncrypt();
11 encrypt.setPublicKey(keyPublic);
12
13
14 var dataFinalCard = encrypt.encrypt(data_string_card);
15 var dataFinalUser = encrypt.encrypt(data_string_user);
```

Figura 43: Proceso de encriptación con RSA  
Fuente: Elaboración propia

La aplicación y ejecución de los smart contracts ERC-20 se realizan al finalizar los procesos de compra/venta de los Marketplace de productos, servicios y criptomonedas que ofrece la plataforma Fintech Pay2Meta y para esto consta de los siguientes pasos:

- 1) Realización de compra/venta en los marketplaces de Pay2Meta.

Pay2Meta ofrece dos marketplaces donde los usuarios pueden realizar compras o ventas, <sup>3</sup> en la figura 44 se ilustra el proceso de compra en el Marketplace de productos y servicios <sup>3</sup> y en la figura 45 se ilustra el proceso de compra de criptomonedas.

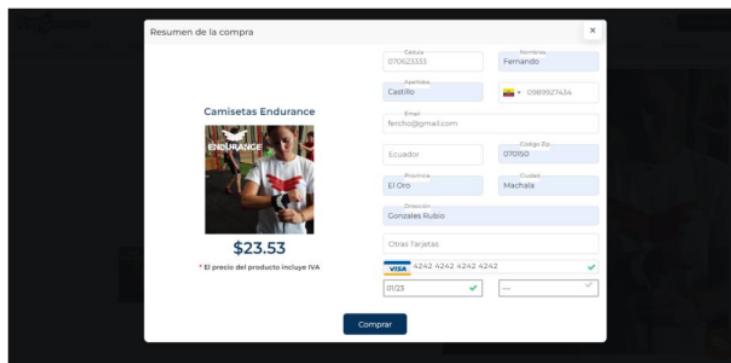


Figura 44: Marketplace de productos y servicios de Pay2Meta

Fuente: Elaboración propia

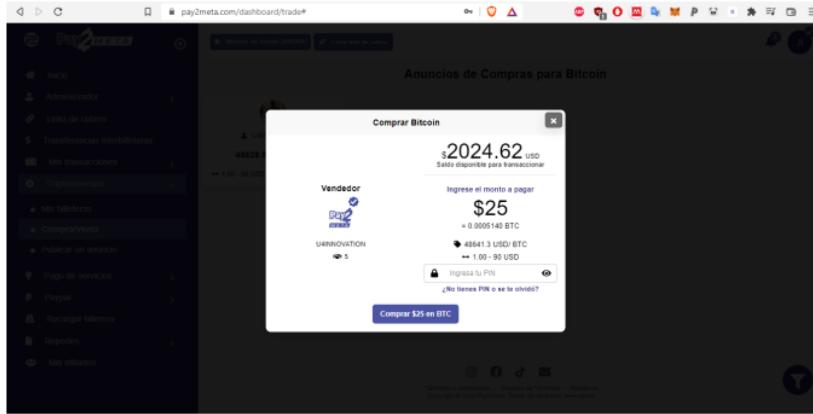


Figura 45: Marketplace de criptomonedas de Pay2Meta

Fuente: Elaboración propia

## 2) Compilación del código solidity en Iotex Studio web IDE.

Una vez realizado una compra o venta en los Marketplace se procede a la compilación del código del smart contract ERC-20, en la figura 46 se ilustra el proceso de compilar el smart contract con el IDE web que ofrece Iotex, esto da como resultado un bytecode y un código ABI que será necesario posteriormente para deployar el smart contract.

```

Solidity Compiler
Compiler: soljson-v0.5.13+commit
Enable Optimization
Compile Contract

Deploy & Run Transactions
Environment: JavaScript VM
Account: iot19...e7ac100.0
Gas Limit: 3000000
Value: 0 Raw
Compiled contracts: Select
Deployed Contracts

```

```

Solidity Compiler
Compiler: soljson-v0.5.13+commit
Enable Optimization
Compile Contract

Deploy & Run Transactions
Environment: JavaScript VM
Account: iot19...e7ac100.0
Gas Limit: 3000000
Value: 0 Raw
Compiled contracts: Select
Deployed Contracts

```

```

FILE          FILE      EDIT
FILE          erc20.sol    marketplace.sol x
EXPLORERS    erc20.sol
              test.sol
              erc20
              erc721.sol
              marketplace.sol

```

```

marketplace.sol
1 pragma solidity >=0.5.13;
2 import "openzeppelin/contracts/token/ERC20/ERC20.sol";
3 contract Marketplace is ERC20 {
4     mapping(address => uint256) public amount;
5     uint256 totalAmount;
6     string tokenName;
7     string tokenSymbol;
8     uint256 decimal;
9     constructor() public {
10         totalAmount = 1000 * 10**18;
11         amount[msg.sender] = totalAmount;
12         tokenName = "Iotex";
13         tokenSymbol = "IOT";
14         decimal = 18;
15     }
16     function totalSupply() public view returns(uint256){
17         return totalAmount;
18     }
19     function balanceOf(address to_who) public view
20     returns(uint256){
21         return amount[to_who];
22     }
23     function transfer(address to_a,uint256 _value) public
24     returns(bool){
25         amount[msg.sender] -= amount[msg.sender];
26         amount[msg.sender] += msg.sender._value;
27         amount[to_a] += amount[to_a]_value;
28         return true;
29     }
30 }

```

Figura 46: Compilación de código solidity y obtención del bytecode en Iotex

Fuente: Elaboración propia

## 3) Desplegar el smart contract con Iotex.

Una vez que se obtenga el código de bytes del paso anterior, el siguiente paso es deployar el smart contract enviándolo a la red de cadena de bloques de IoTeX. En la figura 47 se

ilustra el código en nodeJS para deployar un smart contract con Iotex, el código otorga un hash que sería el address del contrato que se utilizará en el último paso.

Figura 47: Código fuente para deployar un smart contract con Iotex

*Fuente: Elaboración propia*

4) Ejecutar la transferencia del token ERC-20 con Tatum.

El último paso es realizar la transferencia del pago en criptomonedas al comercio, esto se realiza una vez que se haya completado todo el proceso del marketplace, para esto se ejecutará el endpoint ofrecido por la plataforma blockchain de Tatum donde se establece sobre qué cadena de blockchain se hará la transferencia, en este caso ETH, también se

define a cual address se enviará, la cantidad en criptomonedas, el address del contrato generado anteriormente y finalmente la llave privada del dueño de la billetera de Ethereum, a continuación se presenta el endpoint de tipo POST que se utilizó para la transferencia de criptomonedas conjuntamente con los parámetros de envío.

5

**Post:** <https://api-eu1.tatum.io/v3/blockchain/token/transaction>

```
{  
  "chain": "ETH",  
  "to": "0x10a21aa67e615ddd1e63aa8d106c684279453ff5",  
  "amount": "0.0096222",  
  "contractAddress": "0x45871ED5F15203C0ce791eFE5f4B5044833aE10e",  
  "fromPrivateKey": "mi_llave_privada_de_mi_billetera"  
}
```

El resultado de ejecutar el endpoint se ve reflejado en la figura 48 donde se ilustra la transacción del pago realizado en la ropsten de pruebas de Ethereum:

<https://ropsten.etherscan.io/tx/0x04c72bf9f80ab620750f17313839cf4da2b670304d9a145a6388d26a4c2b3b83>

The screenshot shows the Etherscan interface for the Ropsten Testnet. The transaction details page displays the following information for the transaction 0x04c72bf9f80ab620750f17313839cf4da2b670304d9a145a6388d26a4c2b3b83:

- Transaction Hash: 0x04c72bf9f80ab620750f17313839cf4da2b670304d9a145a6388d26a4c2b3b83
- Status: Success
- Block: 12115835 | 28316 Block Confirmations
- Timestamp: 7 days 59 mins ago (Mar-21-2022 09:14:23 PM +UTC)
- From: 0x6bc435c110f999504c591dd61151b54f43ee72f
- To: 0x10a21aa67e615ddd1e63aa8d106c684279453ff5
- Value: 0.0096222 Ether
- Transaction Fee: 0.000125 Ether

Figura 48: Transferencia de criptomoneda Ethereum

Fuente: Elaboración propia

Para la aplicación de la identidad digital con NFT, primeramente, los usuarios deben realizar una verificación biométrica utilizando la plataforma Mati, en la figura 49 se ilustra el primer paso de esta verificación biométrica donde el usuario deberá elegir la forma en cómo se verificará, si escaneando un código QR o un link enviado a su email.



*Figura 49: Primer paso de verificación biométrica  
Fuente: Elaboración propia*

El siguiente paso, como se muestra en la figura 50 es enviar una foto de la parte delantera y trasera del documento de identidad y también un video de unos 5 segundos mostrando el rostro, como se ilustra en la figura 51.



*Figura 50: Segundo paso de verificación biométrica  
Fuente: Elaboración propia*



Figura 51: Tercer paso de la verificación biométrica

Fuente: Elaboración propia

Una vez finalizado la verificación biométrica, hay que esperar un tiempo en el que el sistema a través de inteligencia artificial valide los documentos enviados. El resultado de la verificación biométrica es un documento pdf que se lo puede observar en el siguiente link:

<https://firebasestorage.googleapis.com/v0/b/backservicespagos.appspot.com/o/verificacionBiometrica.pdf?alt=media&token=bf4baf87-32f7-40cf-a2a8-75ce3ddd3249>

Finalmente, dentro del perfil del usuario se genera un código QR con su identidad verificada biométricamente, en la figura 52 se ilustra el resultado.

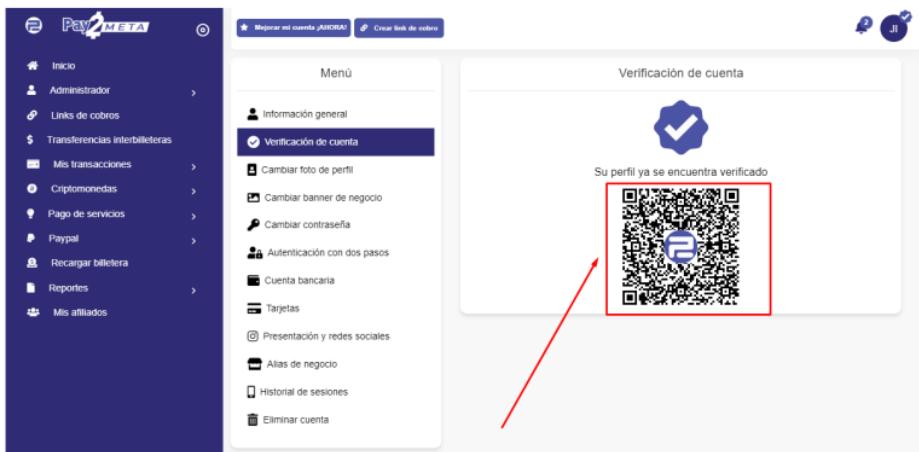


Figura 52: Perfil de usuario verificado biométricamente  
Fuente: Elaboración propia

El código QR generado, ilustrado en la figura 53, redireccionada a un registro de IOTA para asegurar la inmutabilidad de la verificación del usuario. El siguiente paso es convertir la imagen del QR en una identidad digital con NFT.



Figura 53: Código QR del usuario verificado

Fuente: Elaboración propia

Al igual que en los smart contract ERC-20, el código utilizado para los NFT hay que compilarlo y deployarlo con Iotex y obtener el código ABI y el bytecode para posteriormente obtener el address del contrato. Con ese address del contrato se procede a utilizar el endpoint de Tatum para acuñar un nuevo token ERC-721, un ejemplo del endpoint junto con los parámetros utilizados son los siguientes:

5

**Post:** <https://api-eu1.tatum.io/v3/nft/mint>

```
{  
  "chain": "ETH",  
  "to": "0x10a21aa67e615ddd1e63aa8d106c684279453ff5",  
  "contractAddress": "0x1860Cf5A199892BC527A0698e7be08a7C6Bc064",  
  "url": "https://pay2meta.com/public/nft_100.json",  
  "authorAddresses": [  
    "0x4eec1a0a2ae9bb1d9601101d429005b10da994f3"  
  ],  
  "fromPrivateKey": "mi_llave_privada",  
}  
}
```

De manera opcional, el usuario puede publicar su NFT en la plataforma Opensea, el resultado se ilustra en la figura 54 y se lo puede observar en el siguiente link:

<https://opensea.io/assets/0x495f947276749ce646f68ac8c248420045cb7b5e/7523418735891926846006348340043757648305025013495208722884814318985145221121/>

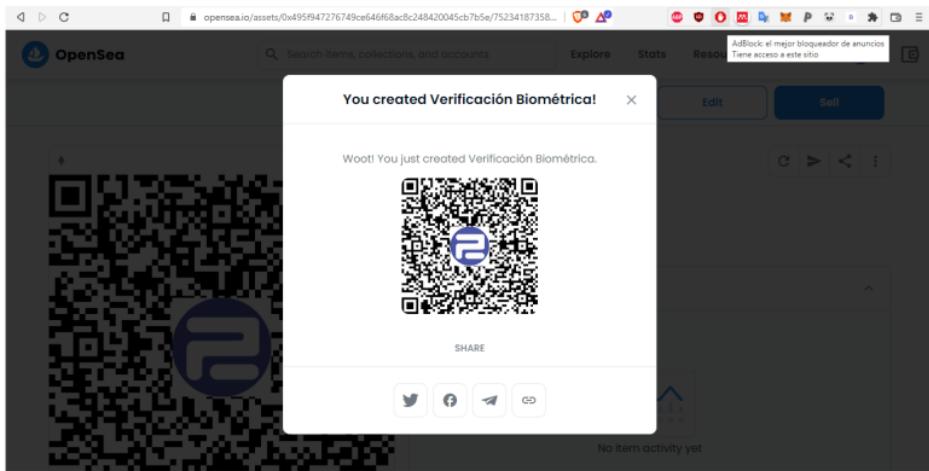


Figura 54: Publicación del NFT en OpenSea

Fuente: Elaboración propia

## CAPÍTULO IV: DISCUSIÓN DE RESULTADOS

El siguiente capítulo discute los principales resultados obtenidos en la investigación, comenzando con la sección 4.1 que trata sobre los principales hallazgos encontrados, seguido de la sección 4.2 donde se detallan la relación de los resultados obtenidos con trabajos previos realizados por otros autores, la sección 4.3 que trata sobre trabajos futuros detectados al finalizar la investigación y finalmente las conclusiones y recomendaciones.<sup>29</sup>

### 4.1 Hallazgos fundamentales.

Al finalizar la siguiente investigación se logró implementar de manera exitosa tres tecnologías de registros distribuidos en una arquitectura de microservicios de Google Cloud. Fueron en total tres instancias de Google Cloud utilizados para el despliegue de las aplicaciones clientes, es este caso dos aplicaciones web y una aplicación móvil, también se utilizaron 12 microservicios implementados con firebase cloud functions programados en NodeJS 14, Python 3.5 y solidity 0.8 para los procesos transaccionales, notificaciones, conexiones y almacenamiento con IOTA, ejecución de los smart contract ERC-20 y ERC-721 con Iotex y Tatum blockchain.

Tras la correcta implementación del sistema mencionado anteriormente se obtuvieron los siguientes hallazgos fundamentales:

#### 4.1.1 Usuarios verificados y con identidad digital con NFT.

Pay2Meta realizó un pre registro de usuarios durante los meses de enero, febrero y marzo donde se les permitieron realizar la verificación biométrica y la obtención de su identidad digital con NFT para verificar su cuenta dentro de la plataforma. En la figura 55 se ilustran la cantidad de usuarios que se pre-registraron en la plataforma, obteniendo un total de 385 usuarios registrados de los cuales el 97% (375 usuarios) realizaron la verificación biométrica y de estos usuarios verificados biométricamente, el 98% (368 usuarios) optaron por obtener su identidad digital con NFT.<sup>20</sup>

Debido al alto porcentaje de usuarios verificados biométricamente y que optaron por tener una identidad digital con NFT demuestra el alto interés de los usuarios por tener sus cuentas verificadas dentro de una plataforma fintech y a su vez por tener sus propias identidades digitales dentro del mundo de la blockchain.

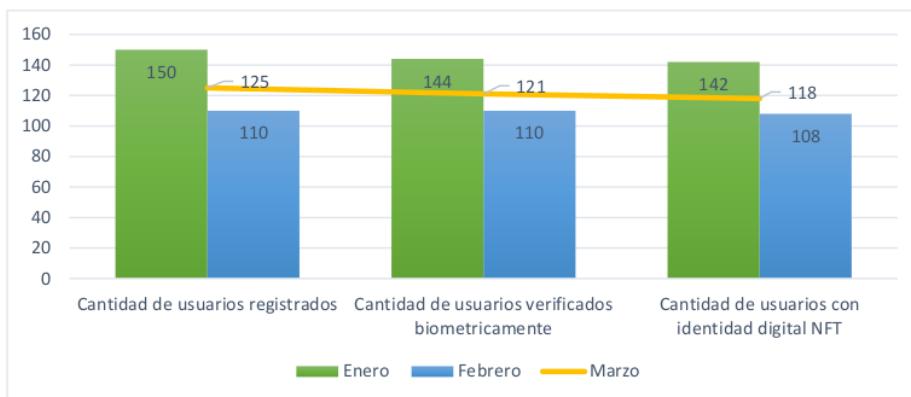


Figura 55: Usuarios registrados, verificados y con NFT en los meses de enero, febrero y marzo del 2022

Fuente: Elaboración propia

#### 4.1.2 Transacciones hacia IOTA y smart contracts ejecutados.

La muestra utilizada en esta investigación fue de un total de 255 transacciones entre los <sup>12</sup> meses de enero, febrero y marzo del 2022, donde se reportaron un 100% de efectividad en la ejecución de los endpoints para el almacenamiento de las transacciones hacia IOTA, en la figura 56 se ilustra los resultados de la ejecución de los contratos inteligentes donde se detectó un 92% de efectividad en la ejecución de los smart contracts, otros 3% de transacciones no se realizaron por detección de pago fraudulento por parte de los clientes, un 4% debido a que el comercio no cumplió con su parte del acuerdo y el 1% restante se debió a que el comercio posee saldo insuficiente en su billetera de criptomonedas Ethereum.

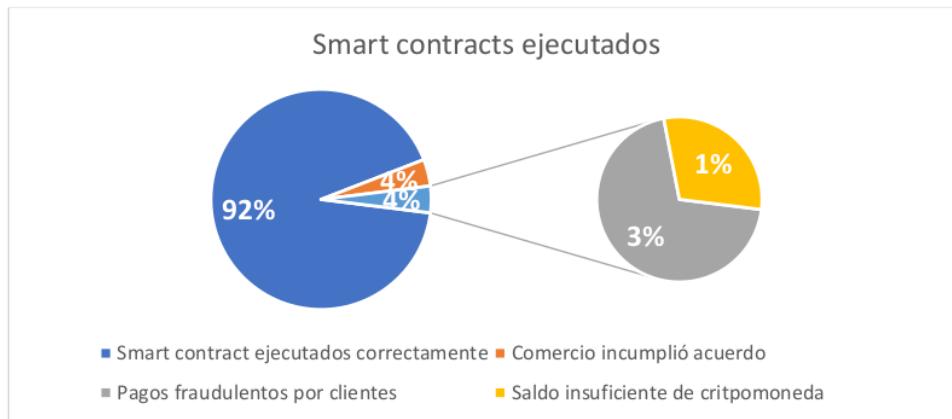


Figura 56: Transacciones hacia IOTA y smart contracts ejecutados

Fuente: Elaboración propia

#### 4.1.3 Seguridad del envío de datos desde aplicaciones clientes.

La seguridad de los datos se encuentra garantizados dentro del Tangle o Blockchain debido al uso de sus protocolos de consenso y a los diferentes algoritmos de encriptación que utilizan cada uno de ellos, pero ningún DLT garantiza la seguridad de los datos enviadas desde las aplicaciones clientes hacia los DLT por tal motivo es importante realizar un correcto análisis del tráfico de la red para prevenir ataques del tipo hombre en el medio y así evitar la escucha y desciframiento de los datos enviados para su posterior almacenamiento en los DLT. Para testear la seguridad del envío de datos desde aplicaciones clientes, se utilizó una de las funcionalidades ofrecidas por la plataforma Pay2Meta llamada links de billetera, en la figura 57 se ilustra el pago exitoso de cinco dólares realizados con una tarjeta de crédito VISA en uno de los links de billeteras de Pay2Meta.

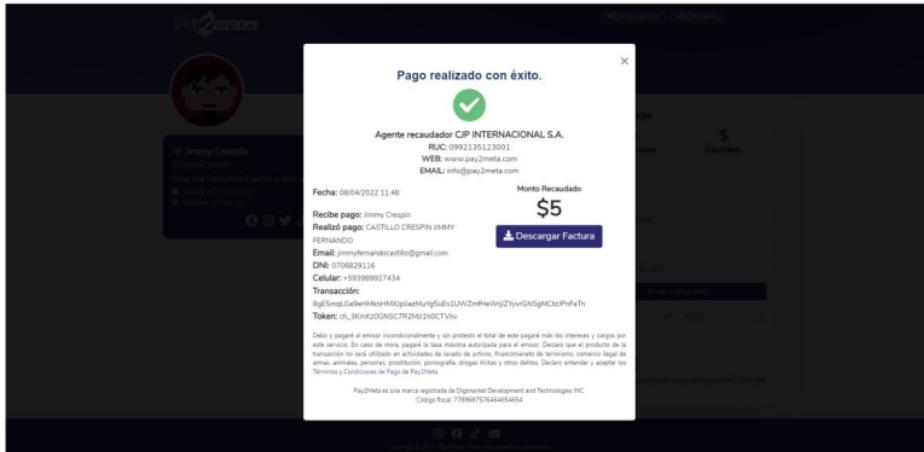


Figura 57: Confirmación de pago con tarjeta de crédito Visa

Fuente: Elaboración propia

Los datos enviados a través del formulario de pago fueron encriptados con RSA, en la figura 58 se ilustra los datos obtenidos al momento de realizar la petición Post, en este caso los datos de la tarjeta con la que se realizó el pago y los datos personales del cliente, ambos parámetros fueron encriptados con RSA.  
<sup>9</sup>

The screenshot shows the Wireshark interface with a selected RSA-encrypted packet. The packet details pane displays the following information:

- dataFinalCard:** BBSvQ\Abryr0AcQ0Nje5/Z1aEfMzTph768r3SE+yqbz0Bgb40PM1umjtNeddWGHwj8Jxwnvk3UwHM
- dataFinalUser:** JL/94u55+PS6nD5MM0+Rz85WGmn7M5jGnmZqKfn7plRp4CCComPUxYmUK2hom56JrtZKfMqsRu
- pixq6p1Gs1aUktmhWkIZfrbLxdyF1P108zE8HfFmCh8PLuG9puYY0D1nxL1TP4IEaB31nhjao9uFaX3fzTtzgj**
- myR48eiT2I87i0p+dy5iqsnh7xEKPBpRjKJo3nnJ1hEXHpv8X3E1NmhbCsg8CFhxqM=**

Figura 58: Encriptación RSA en los pagos realizados con tarjetas de crédito

Fuente: Elaboración propia

Esta encriptación se la validó utilizando la herramienta wireshark, en la figura 59 se ilustra el tráfico http de tipo post donde se visualiza que los datos enviados están encriptados.

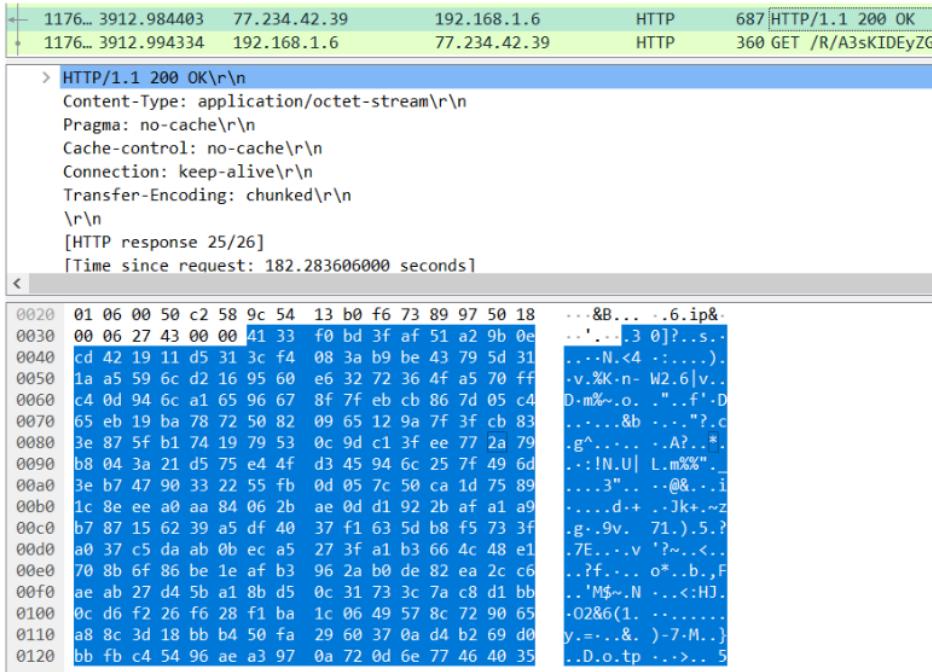


Figura 59: Tráfico de red en wireshark

Fuente: Elaboración propia

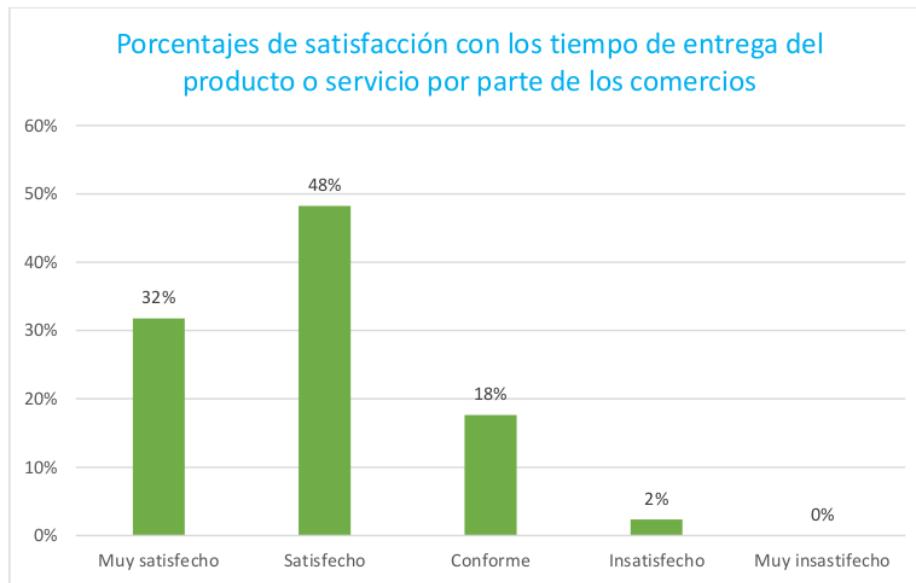
#### **4.1.4 Resultado de encuestas aplicadas.**

Las encuestas (ver anexo 2) fueron aplicadas a los usuarios que realizaron compras en los marketplaces de productos y criptomonedas ofrecidas por Pay2Meta para conocer la satisfacción del producto recibido por parte de los comerciantes y en base a estos resultados determinar potenciales estafadores a futuro.

El total de encuestados fueron 85 usuarios seleccionados a partir de las 255 transacciones que se tomaron como muestra. El resultado de cada pregunta se presenta a continuación.

##### **Pregunta #1 - ¿Qué tan satisfecho quedó con el tiempo de entrega del producto o servicio adquirido en nuestra plataforma?**

El resultado de esta encuesta ayudará a la aplicación de Pay2Meta a establecer límites máximos de entrega de los productos por parte de los comercios y su vez, los smart contracts también tengan un tiempo máximo para ejecutarse. En la figura 61 se ilustra el resultado de la encuesta aplicada para la pregunta #1 que estuvo enfocada al tiempo de entrega del producto o servicio, dando un resultado favorable del 48% de usuarios que quedarán satisfechos y un 32% muy satisfechos.

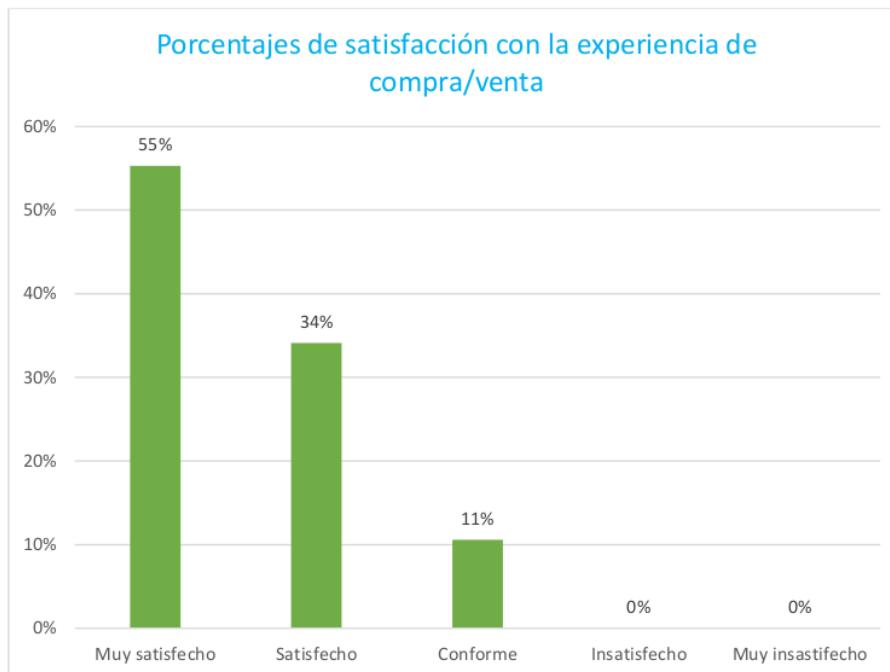


*Figura 60: Resultado de la encuesta del tiempo de entrega del producto o servicio*

*Fuente: Elaboración propia*

**Pregunta #2 - ¿Qué tan satisfecho quedó con la experiencia de compra/venta realizada en nuestra plataforma?**

El resultado de esta encuesta ayudará con el aspecto de usabilidad de la aplicación de Pay2Meta al momento de realizar una compra o venta dentro de los marketplaces. En la figura 62 se ilustra el resultado de la encuesta aplicada para la pregunta #2 que estuvo enfocada a la experiencia de compra o venta, dando un resultado favorable del 55% de usuarios que quedarán muy satisfechos y un 34% satisfechos.

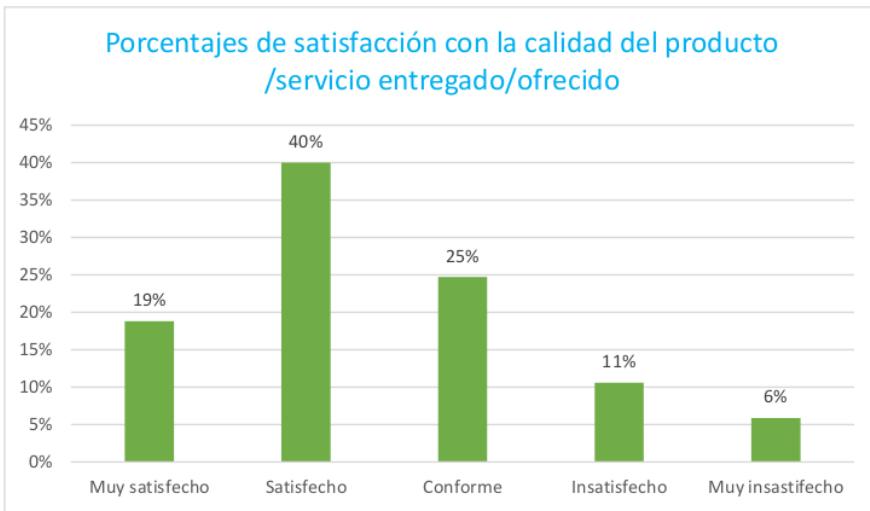


*Figura 61: Resultado de la encuesta de experiencia de compra/venta*

*Fuente: Elaboración propia*

**Pregunta #3 - ¿Qué tan satisfecho quedó con calidad del producto /servicio entregado/ofrecido en nuestra plataforma?**

El resultado de esta encuesta ayudará con el aspecto de estafas de la aplicación de Pay2Meta al momento de realizar una compra o venta dentro de los marketplaces. En la figura 63 se ilustra el resultado de la encuesta aplicada para la pregunta #3 que estuvo enfocada a la calidad del producto entregado, dando un resultado favorable del 40% de usuarios que quedarán satisfechos, pero no tan favorables con el 19% de usuarios que no quedaron muy satisfechos.



*Figura 62: Resultado de encuesta de la calidad del producto ofrecido*

*Fuente: Elaboración propia*

#### **4.1.5 Tiempo de aprobación de recargas de billetera con TDC.**

La plataforma Pay2Meta posee una funcionalidad de recargar billetera a través de pagos con tarjetas de créditos (TDC) para que los usuarios posean saldo dentro de la plataforma. La aprobación de estos pagos por parte de los administradores de Pay2Meta se lo realiza de manera manual en un lapso de tiempo entre los dos a tres días, esto es debido a que primeramente la plataforma de Pay2Meta analiza el pago a través de un sistema antifraude y posteriormente espera la confirmación por parte de la entidad bancaria que el pago no es fraudulento.

Sin embargo, con la implementación de la verificación biométrica con Mati e identidad digital con NFT en conjunto con las transacciones financieras almacenadas en IOTA realizadas en esta investigación, en caso de que se detecte que el pago es fraudulento por parte de la entidad bancaria. La plataforma Pay2Meta puede optar por disputar dicho pago enviando como pruebas los NFT de los usuarios y las transacciones almacenadas en IOTA. Por tal motivo, el tiempo de aprobación de estos pagos pasó de ser de tres días a cero días (ver tabla 14), estableciéndose una optimización del 100% en el tiempo de aprobación de pagos debido a que las recargas de billeteras ahora son instantáneas y no son aprobadas por algún administrador de la plataforma.

Tiempo de aprobación de recargas de billeteras por parte de los administradores de Pay2Meta antes de la implementación de los DLT	Tiempo de aprobación de recargas de billeteras por parte de los administradores de Pay2Meta después de la implementación de los DLT
2 o 3 días	0 días

Tabla 14: Tiempo de aprobación de pagos con TDC

Fuente: Elaboración propia

#### 4.1.6 Total de ganancias por meses.

Uno de los resultados que más benefició a la plataforma Pay2Meta fueron en sus ingresos mensuales, en la figura 64 se ilustra los ingresos iniciales <sup>2</sup> de los meses de enero, febrero, marzo y abril del año 2022, al igual que la cantidad de dinero en disputas ganadas y perdidas, las cantidades de dinero perdidas fueron menores con respecto a las cantidades de dinero disputadas ganadas siendo el total de ganancias por disputas ganadas de \$2.566,39, una cantidad alta a comparación con los \$559,36 perdidos, todo gracias a la implementación de los DLT que a través de la utilización de smart contract con Iotex, registros transacciones en IOTA e identidad digital con NFT se consiguió incrementar la probabilidad de ganar disputas financieras (ver tabla 15).

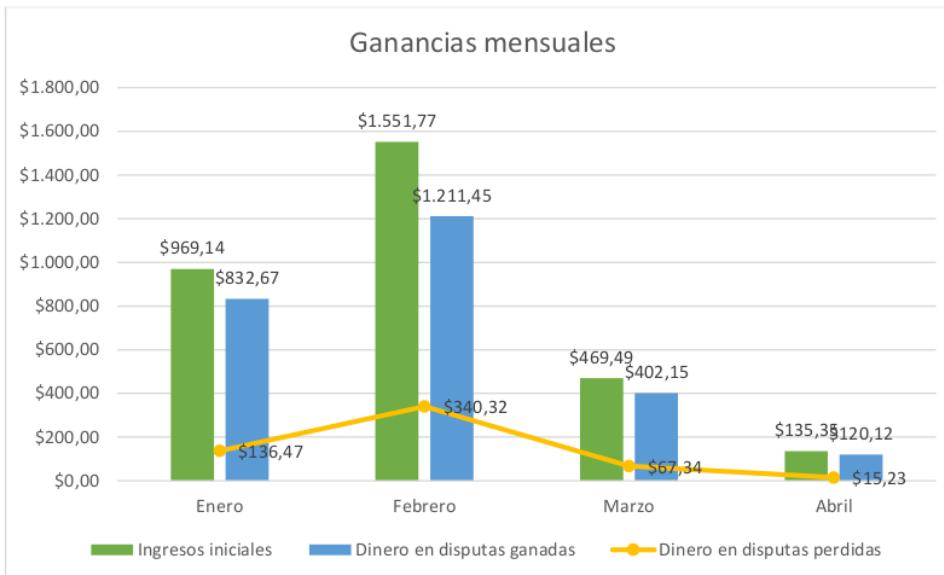


Figura 63: Total de ganancias mensuales

Fuente: Elaboración propia

#### **4.1.7 Probabilidad de ganar disputas financieras por fraudes.**

De las 255 transacciones tomadas como muestra, se incluyeron tantos pagos detectados como potencialmente estafas y fraudulentas entre los meses de enero, febrero, marzo y abril del año 2022, de las cuales 138 de esas transacciones son pagos fraudulentos, en la tabla 15 se ilustra las disputas ganadas que fueron de 125 frente a la cantidad de pagos fraudulentos detectados de 138, al igual que la cantidad de disputas perdidas que fueron 13 y el porcentaje de ganar estas disputas frente a las entidades bancarias del 90%.

Cantidad de pagos por fraude	Disputas ganadas	Disputas perdidas	Porcentaje de disputas ganadas por fraude
138	125	13	90%

Tabla 15: Probabilidad de ganar disputas financieras por fraudes

Fuente: Elaboración propia

#### **4.1.8 Aplicación del coeficiente de correlación de Spearman.**

Para determinar el tipo de correlación y verificar la hipótesis planteada en esta investigación se utilizó el coeficiente de Spearman debido a que en el epígrafe 2.6 se estableció que los datos a trabajar eran de tipo no paramétricas. En la tabla 16 se ilustra la asignación de rangos de los datos de los DLT ejecutados y fraudes o estafas ganados y está conformada primeramente por la columna semanas que representan las 14 semanas que duró la investigación.

Con respecto a la columna de DLT ejecutados, estas cantidades fueron obtenidos de un registro histórico en Firebase de la plataforma Pay2Meta, provenientes de la ejecución de los smart contracts ERC-20 cuando se realizaron las compras/ventas de los marketplaces en conjunto con los registros transaccionales con IOTA y NFT.

En la columna de fraudes y estafas ganadas, estas cantidades se obtuvieron por disputas ganadas con la plataforma de pagos Stripe, en casos de fraudes con tarjetas de crédito/débito y por disputas ganadas con los usuarios de tipo comercio de la plataforma Pay2Meta en casos de estafas.

La columna Rango A está formada por la cantidad de registros analizados (14 registros) asignando el número mayor al valor mayor de la columna de DLT ejecutados y así sucesivamente hasta llegar al valor menor que sería uno. Un ejemplo práctico sería el registro #3 que posee un valor de 44 en la columna DLT ejecutados, que se le asignó el

número 14 en la columna de Rango A, seguido del registro #2 con valor 25 en la columna DLT ejecutados asignándole el número 13 en la columna Rango A y así sucesivamente.

Con respecto a la columna de Rango B, sigue la misma secuencia realizada para la columna de Rango A, solamente que en vez de tomar los valores de la columna DLT ejecutados, se toma los valores de la columna de fraudes y estafas ganados. La columna “d” es la resta entre los rangos A y B y finalmente la columna “d^2” corresponde a la potencia dos de la columna “d”.

#	Semanas	DLT ejecutados	Fraudes y estafas ganados	Rango A	Rango B	d	d^2
1	1	20	32	11	14	-3	9
2	2	25	23	13	8	5	25
3	3	44	54	14	13	1	1
4	4	18	12	9	9	0	0
5	5	7	5	3	2	1	1
6	6	5	2	2	1	1	1
7	7	9	11	4	10	-6	36
8	8	12	8	6	3	3	9
9	9	2	4	1	4	-3	9
10	10	19	14	10	5	5	25
11	11	16	18	8	7	1	1
12	12	23	21	12	12	0	0
13	13	14	16	7	11	-4	16
14	14	11	15	5	6	-1	1
						<b>Suma</b>	<b>134</b>

Tabla 16: Asignación de rangos de los datos

Fuente: Elaboración propia

Una vez finalizado con la tabla de valores, se aplica la fórmula de Spearman ilustrado en la figura 64, donde:

Rs= coeficiente de spearman

D= la suma total obtenido de d^2

n= cantidad de sujetos que se clasifican

$$r_s = 1 - \frac{6 \sum D^2}{n(n^2 - 1)}$$

Figura 64: Fórmula de coeficiente de Spearman

Fuente: Elaboración propia

Entonces:

$$Rs = 1 - \frac{6(134)}{14(14^2-1)} = 1 - \frac{804}{2730} = 0.70$$

Para corroborar el resultado obtenido anteriormente se utilizó el software SPSS. En la tabla 17 se ilustra el resultado obtenido para Rho de 0,895.

Correlaciones			DLT_Ejecutados	Fraudes_estafas_ganados
Rho de Spearman	DLT_Ejecutados	Coeficiente de correlación	1,000	,895**
		Sig. (bilateral)	.	,000
		N	14	14
Fraudes_estafas_ganados		Coeficiente de correlación	,895**	1,000
		Sig. (bilateral)	,000	.
		N	14	14

\*\*. La correlación es significativa en el nivel 0,01 (bilateral).

Tabla 17: Correlación de Spearman en SPSS

Fuente: Elaboración propia

Debido a que el valor calculado manualmente del rs es igual a 0.70 y en SPSS fue de 0.89 se concluye que existe una correlación positiva significativa. Por tanto, mientras más se implementen los DLT en procesos financieros, también aumentarán la cantidad de disputas ganadas por fraudes o estafas de primera persona en pagos realizados en aplicaciones Fintech.

#### 4.1.9 Mediciones de rendimiento y pruebas de carga.

Para las pruebas de medición del rendimiento y pruebas de carga se utilizó una computadora personal como cliente, con las características mostradas en la tabla 18, donde se detallan el procesador, la memoria RAM y el sistema operativo.

CPU	AMD Ryzen 7 3200 2.30 Ghz
RAM	16 GB
Sistema operativo	Windows 11

Tabla 18: Características del equipo para pruebas locales de rendimiento y carga

Fuente: Elaboración propia

En cuestión de los microservicios con cloud functions, en la tabla 19 se ilustra las características del servidor utilizado, donde se especifican aspectos como la región donde

se encuentra, el entorno de ejecución, la memoria y el tiempo máximo de espera en ejecución.

Región del servidor	us-central1
Entorno de ejecución	NodeJS 16
Memoria	1 GB
Tiempo máximo de espera en ejecución	5 minutos

Tabla 19: Características del servidor de Google Cloud Functions

Fuente: Elaboración propia

Con respecto al servidor donde se encuentran alojado los proyectos clientes como la plataforma web del backoffice y Marketplace, en la tabla 20 y figura 65 se ilustran las características del mismo donde se especifica la ubicación del servidor, que tipo de máquina se utilizó, la cantidad de CPU virtual utilizada, así como el tamaño de la memoria y disco duro.

Ubicación del servidor	us-central1-a
Tipo de máquina	G1-small
Cantidad de CPU virtuales	4 CPU virtuales
Memoria	4 GB
Disco duro	20 GB

Tabla 20: Características del servidor de Google Cloud Platform

Fuente: Elaboración propia

The screenshot shows the Google Cloud Platform interface for managing instances. At the top, there's a navigation bar with the project name 'BackServicesPagos' and various menu items. Below this, the main content area is titled 'instancia-bac...' with tabs for 'DETALLES', 'OBSERVABILIDAD', 'INFORMACIÓN DEL SO', and 'CAPTURA'. The 'DETALLES' tab is selected, displaying the following data:

Nombre	instancia-pay2meta
ID de instancia	1146539163895667354
Descripción	Ninguna
Tipo	Instancia
Estado	Activa
Hora de creación	nov. 25, 2020, 1:37:41 p. m. UTC-05:00
Zona	us-central1-a
Plantilla de instancia	Ninguna
En uso por	Ninguno
Reservas	Elegir automáticamente
Etiquetas	goog-dm: instance-b...
Protección contra la eliminación	Inhabilitado
Servicio Confidential VM	Inhabilitado
Tamaño de estado preservado	0 GB

Below this, another section titled 'Configuración de la máquina' shows the following details:

Tipo de máquina	g1-small
Plataforma de CPU	Intel Haswell
CPU virtuales para proporción de núcleos	–
Núcleos visibles personalizados	–
Dispositivo de visualización	Inhabilitado Habilita esta opción para usar las herramientas de desarrollo.
GPU	Ninguna

Figura 65: Características del servidor utilizado en Google Cloud

Fuente: Elaboración propia

Los equipos anteriormente mencionados se los utilizaron para realizar pruebas de envíos de datos transaccionales a través de Postman, en la figura 66 se ilustra como a través de un endpoint de tipo Post se consiguió simular pagos transaccionales al mismo tiempo con una cantidad de 1000 usuarios obtenidos de la base de datos Firebase y con el card\_id de una tarjeta de crédito de pruebas. Esto se logró dividiendo procesos a través de microservicios y eventos proporcionados por Google Cloud Platform realizando pagos asincrónicamente, es decir en paralelo y al mismo tiempo como se ilustra en la figura 67 de los resultados en la consola de Firebase functions del endpoint utilizado.

Figura 66: Envío de transacciones con Postman  
Fuente: Elaboración propia

Figura 67: Consola de Firebase functions de las transacciones en ejecución

Fuente: Elaboración propia

La captura de información se lo realizó con JMeter y se determinó que existe una alta transaccionalidad debido a que se ejecutaron exitosamente todos los microservicios de Google Cloud con DLT, en la tabla 21 se ilustran los resultados obtenidos como tiempo

promedio de ejecución entre microservicios de 1s, el tiempo promedio en el almacenamiento de las transacciones en IOTA que oscilan en 20.5s, tiempo promedio de ejecución de los smart contracts de 3.5s y finalmente el tiempo total de la ejecución de transacciones con 1000 usuarios de 25s, que corresponde a lo mostrado en la figura 66 cuando se realizó la prueba con Postman con un resultado de 25.21 s. En la figura 68 se ilustra la secuencia realizada en el proceso de una transacción financiera, el mismo que está conformado por varios nodos que corresponden a los microservicios ejecutados, iniciando con el microservicio del pago (1), seguido del almacenamiento en IOTA (2), ejecución del smart contract (3) y finalmente el microservicio de notificación (4), tomando un total de 25 segundos por transacción.

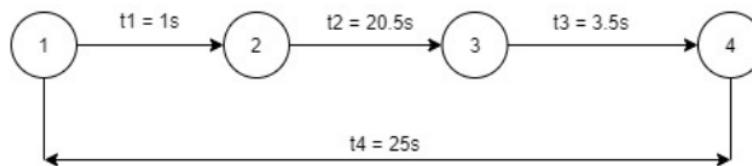


Figura 68: Tiempos de medición de rendimiento y carga

Fuente: Elaboración propia

Tipos	Características	Tiempos (s)
t1	Tiempo promedio de ejecución entre microservicios	1
t2	Tiempo promedio en el almacenamiento de las transacciones en IOTA	20.5
t3	Tiempo promedio de ejecución de los smart contracts	3.5
t4	Tiempo total de ejecución de transacciones de 1000 usuarios simultáneos.	25

Tabla 21: Tiempos de ejecución

Fuente: Elaboración propia

#### 4.2 Relación con trabajos previos.

A continuación, se analizan los resultados obtenidos con trabajos previos de otros autores, entre los cuales se destacan los siguientes:

#### **4.2.1 Seguridad contra ataques man-in-the-middle.**

A pesar de que la seguridad que brindan los DLT es alta, esta solamente abastece la encriptación de la información dentro de los nodos de los DLT pero la seguridad dentro de la red local entre aplicaciones cliente y la utilización de secretos digitales (llaves privadas, contraseñas, tokens, etc) en los microservicios aún queda expuesta a ataques de tipo man in the middle. En especial en aplicaciones Fintech donde existe un constante flujo de información tipo sensible, por tal motivo, esta investigación aporta con la utilización de una encriptación RSA con tamaño de 4096 bits en sus llaves públicas y privadas para brindar protección a los datos que provengan de las aplicaciones clientes hacia los microservicios. Además, dentro de los microservicios se empleó una conexión con la base de datos criptográfica de IOTA Stronghold para obtener, cuando se requiera, los secretos digitales necesarios para el funcionamiento de los endpoints.

Este aporte mejoraría con el trabajo realizado por Ekparinya [148] donde midió el impacto de los ataques man-in-the-middle sobre la red blockchain de Ethereum, pero en sus pruebas realizadas no contempló la utilización de algún algoritmo de encriptación para la conexión con la red, lo que supondría una vulnerabilidad de filtración de las llaves privadas de las billeteras Ethereum utilizadas en dicho experimento. También contribuiría en el trabajo realizado por Riadi [153] donde utilizaron una encriptación de tipo SHA256 en las aplicaciones clientes, pero dentro de sus microservicios para la conexión con base de datos no existe protección, creando una posible vulnerabilidad, caso contrario a esta investigación donde se utilizó IOTA Stronghold para mitigar este problema.

#### **4.2.2 Identidad digital con NFT.**

Aunque actualmente existen varios algoritmos de machine learning para detectar fraudes que realizan las personas con tarjetas de créditos en sus pagos online como se observa en el trabajo realizado por Dornadula [162], el problema de fraudes aún persiste. Por tal motivo, aunque la plataforma Pay2Meta utiliza su propio sistema machine learning antifraude, en esta investigación añade una capa de seguridad adicional al utilizar un algoritmo de verificación biométrica a los usuarios para posteriormente convertirla en un NFT. Obteniendo así una identidad digital dentro del blockchain, aumentando así las probabilidades de reducir los casos de fraudes con apoyo de los algoritmos de machine learning estudiados en el trabajo de Dornadula.

#### **4.2.3 Transacciones con comisiones cero con IOTA.**

Una de las preocupaciones más comunes que poseen los usuarios al utilizar aplicaciones financieras son los costos que estas cobrarán por las transacciones que se realicen. En especial aquellas transacciones que involucren criptomonedas. Sin embargo, el aporte ofrecido en esta investigación al no cobrarse comisiones por la utilización de IOTA cuando se almacenan transacciones financieras en su Tangle, que gracias a su inmutabilidad servirán de apoyo durante el dispute de fraudes de primera persona con las entidades bancarias, los usuarios no perderán su dinero y se brinda un nivel de confianza adicional a la plataforma Fintech, por los pagos realizados de los usuarios. Muchas aplicaciones Fintech utilizan blockchain para el almacenamiento de información, donde los usuarios asumen el pago de comisiones, un ejemplo de esto se observa en el trabajo realizado por Karaivanov [115], que utiliza la red Ethereum pagando comisiones altas, con nuestra solución utilizando IOTA no se añaden estos costos al usuario.

#### **4.2.4 Comisiones bajas de smart contracts con Iotex.**

En nuestra investigación preliminar se obtuvo que Iotex cobra una comisión relativamente más baja que otras blockchain al momento de ejecutar un smart contract, comparado con Zarir [125] que utiliza la red Ethereum para ejecutar los smart contracts, con costos de comisión entre los 3\$ a 10\$ en criptomonedas, nuestra propuesta con Iotex presenta una comisión mucho más baja, de solo 0.35 centavos a 2\$ en criptomonedas.

#### **4.2.5 Aplicación de la metodología ABCDE.**

Las aplicaciones de tipo marketplace con pagos online actuales típicamente utilizan metodologías de desarrollo que no incluyen a los DLT en sus ciclos de vida, un ejemplo de ello aparece en Seitz & otros [172]. En nuestro trabajo se aporta a la comunidad científica, con la propuesta de uso de metodologías ABCDE para este tipo de aplicaciones, resolviendo así la problemática planteada.

### **4.3 Trabajos futuros.**

El desarrollo de este proyecto abre paso a la investigación de varios trabajos futuros que por cuestiones de tiempo, alcance y falta de madurez de algunas tecnologías no se ha implementado aún. A continuación, se indican algunos de ellos.

#### **4.3.1 Smart contracts con IOTA**

Según la web oficial de IOTA, hasta el momento cuando se redactó este trabajo, los contratos inteligentes con IOTA se encuentran disponibles en fase beta, sin soporte de una API para ser utilizado en aplicaciones clientes en un ambiente de producción. Esto resulta ser una total desventaja para las aplicaciones Fintech debido a que como se observó en esta investigación, se tuvo que optar por la utilización de Iotex para la implementación de smart contracts lo que asumía un coste adicional de comisión a los usuarios por cada vez que se las utilizaban, cuestión que no sucedería si se utilizará los smart contracts de IOTA con su coste cero en comisión, lo que hubiera potenciado enormemente esta funcionalidad en aplicaciones Fintech.

#### **4.3.2 NFT con IOTA**

Algo similar ocurre con los NFT de IOTA donde los NFT con IOTA si es posible pero solo en ambiente de pruebas (testnet) debido a que se encuentra en fase beta, lo que imposibilita su implementación en un ambiente de producción. Actualmente IOTA dispone de su propio marketplace de NFT llamado Pylon que aún no abre sus servicios.

#### **4.3.3 Implementación del DLT Radix Tempo**

Cuando se realizó la búsqueda de trabajos relacionados para la elaboración del estado de arte sobre las tecnologías de registros distribuidos, se encontró que aparte del Blockchain y Tangle existe otro tipo de DLT conocido como Tempo, siendo Radix la única tecnología disponible para este tipo de DLT, que goza de muchas ventajas con respecto al blockchain, pero el problema radica en su falta de madurez debido a que esta tecnología fue creada en el año 2020 y hasta el momento se encuentra en fase beta y no dispone de una API completa para su implementación.

#### **4.3.4 Blockchain 4.0**

La blockchain 4.0, según el estado de arte, hace referencia a la combinación de los DLT con inteligencia artificial. La fase de revisión manual de facturas firmadas enviadas por los comercios, podría ser automatizada con la implementación de una IA para optimizar este proceso y exista una transformación real del blockchain 3.0 a la 4.0.

## CONCLUSIONES

La investigación concluyó exitosamente con todos los objetivos planteados comprobando la hipótesis de que si se implementa tecnologías de registros distribuidos (DLT) en una arquitectura de microservicios cloud se incrementa la probabilidad de ganar disputas financieras por casos de estafas y fraudes de primera persona en transacciones financieras online de sistemas DApps fintech.

El uso de los DLT incrementa la seguridad en transacciones financieras online debido a los protocolos de consenso que utilizan, esto sumado a la seguridad ofrecida por Google Cloud en sus microservicios y a las encriptaciones RSA o AES aplicadas en las aplicaciones clientes, todos estos aspectos ayudaron a mitigar casos de estafas y fraudes de primera persona en conjunto con la utilización de smarts contracts e identidad digital con NFT.

La selección de los DLT depende exclusivamente de la naturaleza del proyecto a realizarse por tal motivo es necesario la realización de un SLR para conocer las ventajas y desventajas que estas proveen y elegir las que más se ajusten a las características del proyecto.

La aplicación de la metodología ABCDE resultó ser eficaz para el desarrollo de los sistemas Dapp en la arquitectura de microservicios de Google Cloud utilizada en esta investigación, incluyendo el uso del DLT en sus ciclos de vida.

Con los aportes realizados en nuestro trabajo, se logró construir una Fintech segura, rápida, con tiempos de almacenamiento de apenas 20.5 s en la cloud, costos de comisiones cero o muy bajos, alta confiabilidad y respaldo a los usuarios, gracias a la inmutabilidad de las transacciones.

Quedó demostrado la aceptación de los usuarios por obtener una identidad digital con verificación biométrica y NFT con un resultado del 98% de aceptación y esto sumado a la utilización de códigos PIN, huella dactilar o código de Google Authenticator que ayudaron a la mitigación de fraudes de primera persona al momento de realizar pagos online, ya que todos estos aspectos serían pruebas irrefutables de una persona realizando compras por internet en una aplicación Fintech.

Se obtuvo mayor rentabilidad económica para la plataforma Fintech con un total de ganancias por disputas ganadas de \$2.566,39 frente a los \$559,36 por las disputas

perdidas en el tiempo de pruebas de tres meses. También los usuarios de la plataforma Fintech obtuvieron rentabilidad económica al momento de utilizar smart contract por las bajas comisiones ofrecidas por Iotex blockchain en comparación con otras tecnologías como Ethereum o Bitcoin.

### **RECOMENDACIONES**

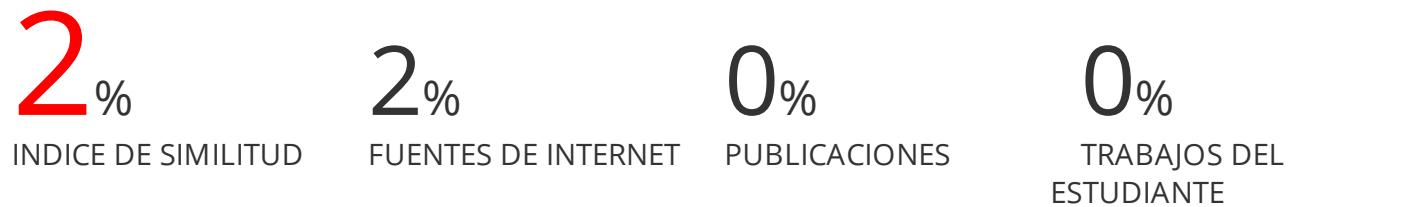
Para trabajar con smart contract o NFT es necesario contar con al menos una billetera de criptomonedas por cuestión de pago de comisiones y cada billetera maneja sus propias claves privadas, que al ser robadas puede provocar el robo del dinero en estas billeteras, por tal motivo se recomienda el uso de IOTA Stronghold para salvaguardar estas llaves privadas y no almacenarlas dentro de archivos del propio proyecto o dentro de base de datos no criptográficas.

Los smart contract están expuestos a vulnerabilidades, por tal motivo se recomienda el uso de alguna herramienta de análisis de seguridad del código generado para los smart contract, en esta investigación se hizo uso de Mythril que resultó ser eficaz para detectar vulnerabilidades antes de proceder con el deploy del mismo.

Las tecnologías de registros distribuidos aseguran seguridad de encriptación en redes WAN pero no en redes LAN, por tal motivo se recomienda la utilización de algoritmos de encriptación como AES o RSA para mitigar estas vulnerabilidades presentes en las aplicaciones clientes.

# Tesis maestria

## INFORME DE ORIGINALIDAD



## FUENTES PRIMARIAS

1	<a href="#">hdl.handle.net</a>	<1 %
2	<a href="#">www.indetec.gob.mx</a>	<1 %
3	<a href="#">documentop.com</a>	<1 %
4	<a href="#">es.slideshare.net</a>	<1 %
5	<a href="#">docs.tatum.io</a>	<1 %
6	<a href="#">doku.pub</a>	<1 %
7	<a href="#">patents.google.com</a>	<1 %
8	<a href="#">www.criptonoticias.com</a>	<1 %
9	<a href="#">www.consumer.es</a>	<1 %

10	Mónica Chillarón Pérez. "Análisis y desarrollo de algoritmos de altas prestaciones para reconstrucción de imagen médica TAC 3D basados en la reducción de dosis.", Universitat Politecnica de Valencia, 2021 Publicación	<1 %
11	<a href="http://desastres.cies.edu.ni">desastres.cies.edu.ni</a> Fuente de Internet	<1 %
12	<a href="http://michoacanimparcial1.wixsite.com">michoacanimparcial1.wixsite.com</a> Fuente de Internet	<1 %
13	<a href="http://www.bsecure.com.mx">www.bsecure.com.mx</a> Fuente de Internet	<1 %
14	<a href="http://www.elconfidencial.com">www.elconfidencial.com</a> Fuente de Internet	<1 %
15	<a href="http://www.macworld.es">www.macworld.es</a> Fuente de Internet	<1 %
16	<a href="http://www.sii.cl">www.sii.cl</a> Fuente de Internet	<1 %
17	80.34.8.41 Fuente de Internet	<1 %
18	<a href="http://elnacional.com.do">elnacional.com.do</a> Fuente de Internet	<1 %
19	<a href="http://manualzz.com">manualzz.com</a> Fuente de Internet	<1 %

20	news.un.org Fuente de Internet	<1 %
21	par.cebas.csic.es Fuente de Internet	<1 %
22	qdoc.tips Fuente de Internet	<1 %
23	repositorio.pucp.edu.pe Fuente de Internet	<1 %
24	www.gestiopolis.com Fuente de Internet	<1 %
25	www.gmu.ayuncordoba.es Fuente de Internet	<1 %
26	documents.mx Fuente de Internet	<1 %
27	pubmed.ncbi.nlm.nih.gov Fuente de Internet	<1 %
28	repositorio.una.ac.cr Fuente de Internet	<1 %
29	somoshalcones.com Fuente de Internet	<1 %
30	transportesynegocios.wordpress.com Fuente de Internet	<1 %
31	www.monografias.com Fuente de Internet	<1 %

32

[www.techtitute.com](http://www.techtitute.com)

Fuente de Internet

<1 %

33

[www.texcoco.gob.mx](http://www.texcoco.gob.mx)

Fuente de Internet

<1 %

34

[empiezoinformatica.wordpress.com](http://empiezoinformatica.wordpress.com)

Fuente de Internet

<1 %

Excluir citas

Activo

Excluir coincidencias Apagado

Excluir bibliografía

Activo