Universidad Nacional Autónoma de México

Facultad de Contaduría y Administración

Bitcoin y las Transacciones Financieras en México

Tesis

Ricardo Antonio Rubio Aguilar

Cd. Mx **2019**

Universidad Nacional Autónoma de México

Facultad de Contaduría y Administración

Bitcoin y las Transacciones Financieras en México

Tesis

Que para obtener el título de:
Licenciado en Informática

Presenta:
Ricardo Antonio Rubio Aguilar

Asesor:
Dra. Lucía Patricia Carrillo Velázquez

Cd. Mx **2019**

A mis padres, Joel y Sandra.

A mi abuelita Guadalupe.

Este logro también en suyo.

Agradecimientos

Nada de esto hubiera sido posible sin el apoyo de mi familia, mi papá y mi mamá, quienes me educaron y me dieron la oportunidad de estudiar una carrera universitaria, siempre me han apoyado incondicionalmente; mi abuelita Guadalupe y mi bisabuelita Antonia, que han sido mi apoyo también durante toda mi vida; mi tía Belinda, que me ha brindado su apoyo siempre que lo he necesitado.

También quiero agradecer a mi hermano, quien me proporcionó el software necesario para poder escribir este trabajo.

Quiero dar un agradecimiento especial a mi asesor de tesis, la Dra. Paty, por su paciencia y apoyo incondicional durante todo el desarrollo de este trabajo, su guía y consejos fueron fundamentales para la realización y conclusión de esta obra. También quiero agradecerle por haber creído en mi y en este proyecto de tesis.

Por supuesto agradezco a mi *alma mater*, la UNAM, que me abrió sus puertas para poder estudiar no sólo la universidad sino también la preparatoria. Esta gran universidad ha transformado radicalmente mi vida y mi forma de pensar, es un privilegio y un orgullo ser parte de la máxima casa de estudios.

También agradezco a mi facultad, la FCA, por permitirme ser uno de sus estudiantes, por haberme formado como profesional durante mis años de estudio. A los maestros que fueron parte de mi formación como profesionista; a las autoridades de la facultad que me proporcionaron las opciones para poder concluir mis estudios cuando el plan de estudios en el que me encontraba inscrito finalizó y entró en vigor uno nuevo.

A todos ustedes, mi más sincero agradecimiento.

Contenido

Introducción

El presente trabajo de investigación es relevante para todos aquellos que estén interesados en conocer cómo funciona el sistema financiero mexicano desde una perspectiva sistémica de alto nivel, con lo que se abstraen muchos de los intricados detalles de su funcionamiento, por lo que no se requiere un conocimiento técnico en el área informática para la compresión de esta sección del trabajo, aunque si será necesario conocer algunos sencillos conceptos en materia contable y financiera. Aunado a lo anterior, también se presentan brevemente algunos de los problemas actuales y futuros a los cuales se enfrenta el sistema financiero mexicano, el lavado de dinero, la inclusión en el sector financiero de los inmigrantes mexicanos que radican en Estados Unidos y la inseguridad en materia informática.

Sin embargo, este trabajo está dirigido principalmente a quién desee conocer el funcionamiento tanto del sistema Bitcoin como de *blockchain*, comprender qué son realmente, pero más importante, dilucidar su verdadero significado, su impacto, así como la factibilidad de estos para ser utilizados dentro del sistema financiero mexicano como una mejora en la eficiencia y seguridad de las transacciones financieras en México.

El problema central que se trata de resolver en el presente trabajo es determinar qué tan factible sería la adopción de la tecnología utilizada en el sistema Bitcoin, por parte de los sistemas informáticos que actualmente conforman el sistema financiero mexicano, principalmente el sistema SPEI, como una mejora en términos de eficiencia y seguridad informática para el sistema financiero mexicano.

Esta investigación atiende el creciente interés por parte de los sectores financiero, económico e informático, en las monedas digitales, también llamadas criptomonedas, como el bitcoin, a través de las cuales, ya sea por medio de las propias monedas o del desarrollo de tecnologías basadas en su funcionamiento, se busca el mejoramiento de los servicios financieros actuales, por esta razón, el presente documento busca esclarecer el panorama en torno a estos temas, concretamente, con respecto al sistema Bitcoin. También, responde a los recientes ataques informáticos realizados contra las instituciones financieras, así como el aumento de los crímenes en materia informática derivados de la inevitable digitalización de las actividades humanas, principalmente, de los servicios financieros.

Este trabajo también es una respuesta a la creciente especulación que actualmente se lleva a cabo en diversos ámbitos, como el financiero, el económico y el informático, acerca del concepto que se ha denominado *blockchain*, y sobre el cual se ha depositado una creencia casi religiosa, que propone el *blockchain* como una solución *ad hoc* para todo tipo problemas que aquejan y aquejarán a la humanidad, lo que es, por decir lo menos, inverosímil.

Se pretende demostrar que el sistema financiero mexicano actual se enfrenta a grandes retos en términos de seguridad informática, de inclusión financiera y de lavado de dinero. Se busca presentar qué es y cómo funciona Bitcoin, tanto el sistema como la moneda. Se pretende desmitificar el término *blockchain*, qué es en realidad y cuál es realmente su alcance. También se pretende mostrar una "vista de pájaro" del sistema financiero mexicano en términos de los sistemas informáticos involucrados, esto es, una vista de alto nivel, omitiendo detalles específicos acerca de cómo funciona y concentrándose en la interacción de los sistemas informáticos que conforman el sistema financiero mexicano, entendiendo estos como una suerte de cajas negras en las que sólo nos interesan las entradas y salidas por parte del sistema, omitiendo los detalles internos de su funcionamiento.

Identificar y reconocer que el sistema financiero mexicano actualmente se encuentra frente a una gran amenaza de seguridad informática es importante porque permite tomar decisiones que impacten en la mitigación y prevención de presentes y futuros ciberataques. De la misma forma, comprender qué es y cómo funciona Bitcoin, el sistema y la moneda, así como del concepto *blockchain*, permite determinar su verdadera factibilidad para ser utilizados dentro del sector financiero para mejorar la seguridad, eficiencia y accesibilidad de este, pero también es un acercamiento sobre el verdadero alcance práctico que pueden tener estas tecnologías como soluciones a otros problemas reales.

La presente investigación se llevó a cabo por medio de la recopilación de información documental, impresa y digital, realizando un análisis de esta, fundamentado en la teoría de la disciplina informática, así mismo, se analizó directamente el objeto de estudio, el sistema Bitcoin, concretamente, se analizó la información real almacenada y procesada por este sistema para comprender su funcionamiento, para lo cual se desarrolló un pequeño programa informático que permite interpretar los datos almacenados y procesados por el sistema Bitcoin. También se utilizaron conceptos probabilísticos para la obtención de resultados derivados del análisis llevado a cabo sobre el sistema Bitcoin. Finalmente, el autor de la presente obra ha utilizado su experiencia adquirida al trabajar en el sector financiero desarrollando software.

Capítulo I. La Situación Actual de las Transacciones Financieras en México

1.1 Lavado de dinero

México es el tercer país con mayor flujo de capitales de procedencia ilícita en el mundo, con cantidades oscilantes entre 50 y 60 mil millones de dólares al año, representan el 5.1% del PIB mexicano, de acuerdo con un estudio realizado por GFI (*Global Financial Integrity*). México se posiciona como uno de los principales actores en el lavado de dinero sólo detrás de China y Rusia (Leyva, 2017), actividad que permite a diferentes grupos mexicanos del crimen organizado financiar sus actividades delictivas, derivando en acciones como el soborno a distintos actores del estado mexicano, así como la compra de los diversos recursos materiales y humanos, a través de los cuales pueden continuar operando un negocio multimillonario (Nájar, 2017).

En base a declaraciones en el juicio contra el narcotraficante "el Chapo", tan sólo en 2005 y 2006 el ex secretario de seguridad pública Genaro García Luna, fue sobornado con al menos 56 millones de dólares (Brooks, 2018).

Se puede comenzar a dimensionar la seriedad del problema, así como la ineficiencia ante el combate contra el lavado de dinero por parte de las autoridades mexicanas en el siguiente texto periodístico (Gutiérrez, 2017):

> *De septiembre del 2016 a junio del 2017, el gobierno mexicano recuperó 543.2 millones de pesos y 11.4 millones de dólares como resultado de su lucha contra el lavado de dinero; sin embargo, para expertos estas cifras son irrisorias si se comparan con la dimensión total del problema.*

Podemos encontrar una comparativa del equivalente en pesos mexicanos que representa el lavado de dinero por parte del narcotráfico, en la publicación de la revista Contra Línea (Flores, 2016):

> *En su informe 2015 National money laundering risk assessment, el Tesoro indica que "el negocio del tráfico de drogas genera alrededor de 64 mil millones de dólares en efectivo cada año [que equivale a 1 billón 164 mil 800 millones de pesos]".*

Por estas razones, el lavado de dinero toma vital importancia en el ámbito nacional mexicano como uno de los principales factores que permiten y perpetúan la existencia de las organizaciones criminales en México, convirtiendo a las instituciones financieras en uno de los ejes principales para el combate contra el crimen

organizado, a través de la eliminación de los flujos de capitales ilícitos por medio de los cuales las organizaciones criminales, específicamente los carteles del narcotráfico, obtienen su inmenso poder.

La enorme capacidad económica que poseen los carteles del narcotráfico les proporciona el acceso a todo tipo de recursos lícitos e ilícitos con los cuales concentran un poder que va más allá de lo económico, corrompiendo y coaccionando el ámbito político, social, e incluso las fuerzas policiacas y militares, quienes deben estar al servicio del estado mexicano y de sus ciudadanos. Esto representa una amenaza no sólo hacia la libertad, tranquilidad y seguridad de los mexicanos, representa una amenaza directa contra el propio estado mexicano, porque el hecho de albergar una fuerza enemiga de gran calibre dentro del propio territorio nacional atenta contra sus intereses y el de sus ciudadanos, pero también y más importante, por el gran peligro que representa la infiltración del crimen organizado en el estado mexicano, corrompiéndolo y dejándolo al servicio de la delincuencia organizada.

Pero a pesar del importante papel que desempeñan las instituciones financieras, tales instituciones, tanto nacionales como extranjeras que operan en el territorio mexicano, han sucumbido ante los grupos criminales (Flores, 2016), ya sea por ineficiencia, falta de mecanismos de control o detección, corrupción o coacción, es innegable el factor humano como principal fuente de error en la batalla contra el lavado de dinero.

En 2016 las autoridades mexicanas admitieron públicamente que los siete grandes bancos (BBVA Bancomer, Santander, Citibanamex, HSBC, Scotiabank, Banorte e Inbursa) que operan en el país, han posibilitado a grupos criminales lavar miles de millones de pesos en el sistema financiero (Flores, 2016), una afirmación impactante por parte del gobierno mexicano, pero que se queda bastante corta al decir que se lavan miles de millones de pesos, cuando se estiman cifras en miles de millones de dólares que equivalen a billones de pesos mexicanos, si tomamos en cuenta las publicaciones de Flores y Leyva.

La incapacidad de inhibir el flujo de capitales ilícitos por parte de las instituciones financieras es una señal de que el sistema financiero mexicano no está siendo capaz de combatir esta actividad ilícita utilizando los mecanismo tradicionales existentes, en el cual, las personas desempeñan el factor determinante ante la lucha contra el lavado de dinero, tal y como se demostró durante el caso rampante de blanqueamiento de capitales cometido por el banco HSBC, documentado en las publicaciones realizadas por diversos medios de comunicación, entre ellos, el artículo publicado por Esquivel en la revista Proceso (Esquivel, 2012) y el artículo publicado por Taibbi en la revista Rolling Stone (Taibbi, 2012).

Se podría pensar que es un fallo esperado, como se puede espera de cualquier sistema complejo, el problema no sólo es el enorme fallo en sí mismo, sino la ausencia de acciones reales que mitiguen ese tipo de acontecimientos, o lo que podría ser peor, falta de medidas efectivas que castiguen acciones relacionadas con el lavado de dinero, esto queda de manifiesto en las medidas tomadas por el departamento de justicia de los Estados Unidos para castigar el crimen cometido por HSBC, donde únicamente se aplicó una sanción económica a dicho banco, así lo señala Taibbi:

> *Even more shocking, the Justice Department's response to learning about all of this was to do exactly the same thing that the HSBC executives did in the first place to get themselves in trouble – they took the money to look the other way.*

Entonces, no es posible realmente combatir el lavado de dinero sólo a través de los mecanismos tradicionales existentes, es necesario realizar un cambio de paradigma en la forma en la que operan las instituciones financieras, es más que evidente que no puede recaer en el factor humano toda la responsabilidad de la detección y eliminación de transacciones financieras que están involucradas con actividades de procedencia ilícita, es imprescindible hacer uso de las nuevas tecnologías e innovaciones emergentes en el siglo XXI para la creación de un nuevo sistema financiero que sea capaz de mitigar de forma más contundente, precisa y automática, el flujo de capitales ilícitos, basándose en procesos algorítmicos, los cuales no son susceptibles a los errores humanos una vez que estos han sido diseñados correctamente.

1.2 Remesas

Las remesas son el envío de dinero por parte de personas que viven en un país distinto a su nación de origen, hacia personas, generalmente familiares, que viven en el país del cual emigraron (CONDUSEF, 2014), en este rubro, el envío de dinero por parte de los mexicanos que trabajan en Estados Unidos hacia el territorio mexicano rompió récord durante el año 2018 al llegar a la cantidad de 33 mil millones de dólares, 10.5 % más que en 2017, de acuerdo a las cifras publicadas por el Banco de México (La Jornada, 2019).

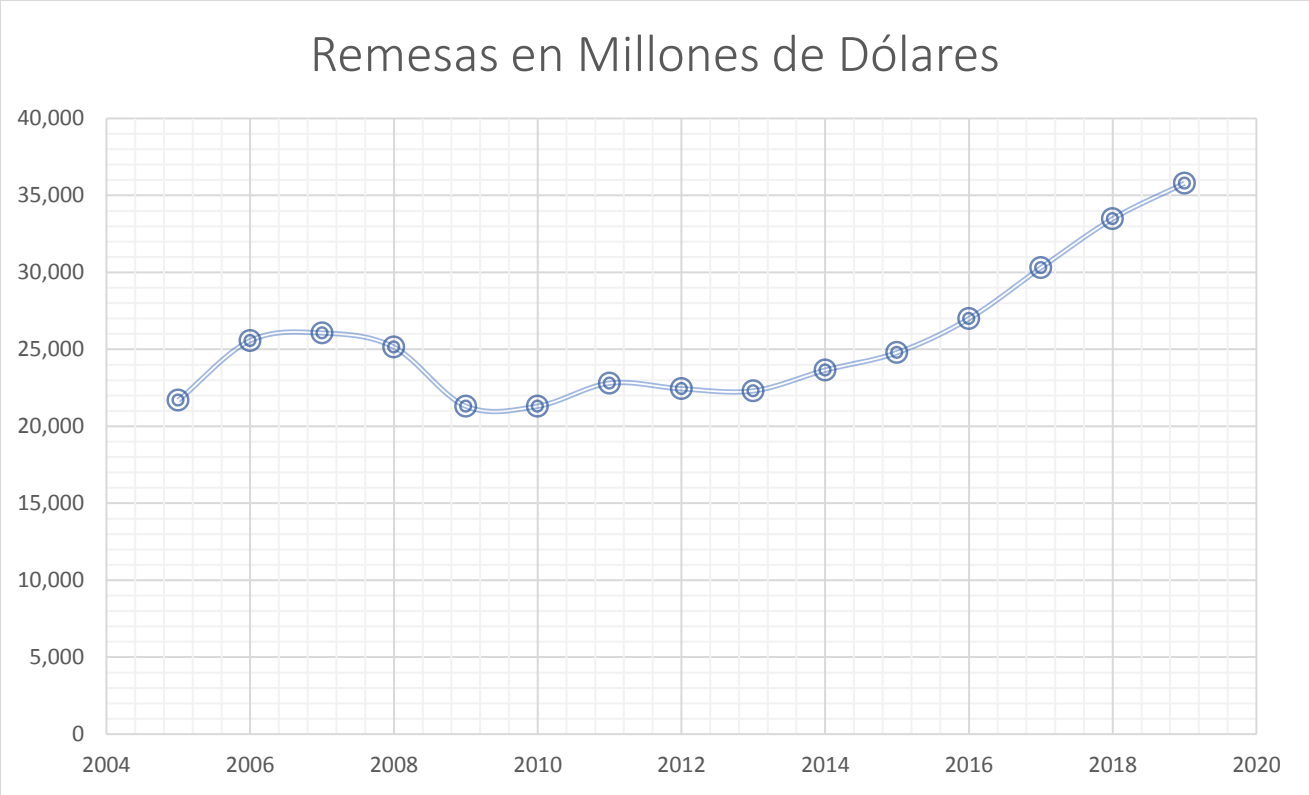

Figura 1. Remesas Familiares enviadas a México, en millones de dólares, 2005 – 2019. Adaptada de Ingreso récord de remesas por 33 mil 481 mdd en 2018: La Jornada. 2019.

La gran cantidad de remesas enviadas cada año desde el país vecino no puede ser demeritada, siguiendo los últimos datos de envío de remesas por parte de mexicanos residentes en los Estados Unidos correspondientes al año 2018, la cantidad total de dinero enviada durante ese año es de 33 mil millones de dólares, esto equivale a poco más del 2% del PIB mexicano, y de acuerdo con el artículo publicado por el periódico El Financiero (Usla, 2018), cerca de 1.7 millones de familias mexicanas se benefician de las remesas, donde el dinero recibido por parte de algunas familias representa su único ingreso.

La forma de envío de remesas se distribuye de la siguiente manera (Usla, 2018):

> *La modalidad de pago más utilizada es el efectivo, pues alrededor del 95 por ciento de las remesas se envían de esta forma. Dichas transacciones se suelen realizar en un 41.3 por ciento en empresas de remesas, el 33.2 por ciento en tiendas, supermercados y farmacias y el 20.6 por ciento en bancos.*

De acuerdo con la CONDUSEF[1], el impacto que tienen las remesas sobre el país mexicano radica en la generación de actividades benéficas, como el incremento en la capacidad de compra, mejora en las condiciones de la educación, la creación de empleos y la producción de nuevos bienes de consumo (CONDUSEF, 2014).

1.3 Sistemas de Pago

Los sistemas de pago son sistemas financieros informáticos diseñados para llevar a cabo operaciones y procedimientos financieros comunes que cumplen con un conjunto de reglas definidas por la autoridad financiera y legal correspondiente. Tienen la capacidad de proveer de servicios financieros a través de una red informática, esto servicios pueden ser accedidos únicamente por instituciones financieras establecidas y autorizadas por la autoridad financiera correspondiente (Bank of England, 2019).

En México la autoridad encargada de definir los procedimientos y reglas para los sistemas financieros es el Banco de México (Banxico), adicionalmente Banxico se encarga del diseño, administración y operación de algunos de los sistemas financieros más importantes en México.

La importancia de los sistemas de pago radica en la problemática asociada con la interacción, esto es, la transaccionalidad entre diferentes instituciones financieras, donde cada una de ellas mantiene sus propios activos, así como su propio libro contable sobre las transacciones financieras que son llevadas a cabo. Esto genera los siguientes problemas, primero, no hay una forma de mantener los libros contables coherentemente sincronizados cuando se llevan a cabo transacciones entre diferentes instituciones financieras, ya que cada una de ellas lleva por su parte su propio libro contable, segundo, no hay garantía de que los derechos y las obligaciones serán cumplidos entre las instituciones financieras que llevan a cabo una o más transacciones financieras, sobre todo cuando una de las partes no puede cumplir con sus obligaciones, por lo tanto, conlleva un alto riesgo llevar a cabo operaciones financieras entre diferentes partes cuando no existe un método que garantice tanto el cumplimiento de derechos y obligaciones como el registro uniforme de la contabilidad derivada de las transacciones.

[1] Comisión Nacional para la Protección y Defensa de los Usuarios de Servicios Financieros (CONDUSEF), es una institución gubernamental mexicana, encargada de supervisar y regular las instituciones financieras para garantizar que los servicios ofrecidos no contravienen los derechos de los usuarios (CONDUSEF, s.f.).

De forma general, se puede decir que se necesita un intermediario o autoridad central que regule la transaccionalidad financiera entre las diferentes partes que llevan a cabo operaciones financieras, de este modo el intermediario o autoridad central garantiza el cumplimiento de los derechos y obligaciones de cada una de las partes y al mismo tiempo puede mantener un único registro contable en el cual se asientan todas las transacciones realizadas, esto último simplifica y facilita la contabilidad y la auditoría.

Los sistemas de pago pueden clasificarse de diferentes formas, los criterios utilizados en el gremio financiero para la clasificación de los sistemas de pago son los siguientes, por la velocidad de liquidación, por el método de liquidación y por el importe de las transferencias o transacciones (Banxico, s.f.).

1.3.1 Sistemas de Pago por Velocidad de Liquidación

Los sistemas de pago se clasifican bajo este rubro dependiendo de qué tan rápido son liquidadas por el sistema las transacciones financieras, la velocidad de la transacción está determinada de la siguiente forma, si la liquidación de la transacción es llevada a cabo inmediatamente, en tiempo real, o si es liquidada de forma diferida. Los sistemas que procesan las transacciones de forma diferida lo hacen de forma programada de acuerdo con un intervalo de tiempo predeterminado en el sistema de pagos, por ejemplo, liquidar las transacciones al día siguiente. Por otro lado, los sistemas de pago que realizan la liquidación de las transacciones inmediatamente, es decir, en cuanto llegan al sistema, son clasificados como sistemas de pago en tiempo real, así mismo, los sistemas de pago que realizan varias liquidaciones de transacciones durante un periodo de tiempo relativamente corto, unos segundos o minutos, durante el mismo día, son sistemas que se aproxima al procesamiento en tiempo real, por lo tanto, el sistema es considerado como un sistema en tiempo casi real (Banxico, s.f.).

Cuadro comparativo de las ventajas y desventajas entre los sistemas de tiempo real y tiempo diferido (Cortina Morfín & Álvarez Toca, 2014):

Sistemas de pago	***Ventajas***	***Desventajas***
Sistemas de liquidación bruta en tiempo real (Real Time Gross Settlement)	La liquidación es inmediata si la institución financiera cuenta con liquidez dentro del sistema. Extinción de obligaciones.	Se requiere una gran cantidad de liquidez dentro del sistema por parte de las instituciones financieras para poder cumplir con las obligaciones.

	Se evita la acumulación de derechos y obligaciones no liquidadas.	
Sistemas de liquidación neta diferida (Deferred Net Settlement)	El proceso de compensación reduce la cantidad de liquidez necesaria por parte de las instituciones financieras dentro del sistema.	Si una de las instituciones financieras no puede cumplir sus obligaciones la liquidación no puede llevarse a cabo. Los problemas no se observan sino hasta el cierre de los mercados financieros. La liquidación de la transacción se lleva a cabo mucho tiempo después con respecto a la generación de la transacción.

Tabla 1. Ventajas y desventajas de los sistemas de pago en tiempo real y diferido.

1.3.2 Sistemas de Pago por Método de Liquidación

Los sistemas de pagos financieros pueden llevar a cabo la liquidación de dos formas, liquidación bruta o liquidación neta, la primera corresponde con la liquidación de cada una de las transacciones financieras de forma individual, siempre y cuando existan fondos en la cuenta de la institución liquidadora. Por otro lado, una liquidación neta consiste en la liquidación conjunta y simultanea de varias transacciones financieras de forma que se compensan derechos y obligaciones a través una sola liquidación (Banxico, s.f.).

La liquidación neta también es llamada compensación porque compensa derechos y obligaciones, esta se puede llevar a cabo de dos formas, de manera bilateral o multilateral, en la compensación bilateral el sistema compensa entre cada par de instituciones financieras los derechos y obligaciones de todas las transacciones financieras llevadas a cabo entre ellas, quedando una única obligación de una de las instituciones financieras frente a la otra. Mientras que en la compensación multilateral se sustituyen todas los derechos y obligaciones de todas las transacciones financieras llevadas a cabo por múltiples instituciones financieras, quedando únicamente un solo derecho u obligación de cada una de las instituciones financieras (Banxico, s.f.).

1.3.3 Sistemas de Pago por el Importe de las Transacciones

En esta clasificación ordenamos los sistemas de pago de acuerdo con los importes de las transacciones procesadas por el sistema, en esta clasificación los sistemas se dividen en sistemas de altor valor y sistemas de bajo valor, los primeros corresponden a sistemas que tienen importes en millones de pesos por las transacciones procesadas. En los sistemas de bajo valor, los importes corresponden a cientos de miles de pesos por las transacciones procesadas (Banxico, s.f.).

1.3.4 Riesgos en los Sistemas de Pago

Los sistemas de pago, por el tipo de transacciones que procesan, conllevan un riesgo inevitable, sin embargo, también se obtienen grandes beneficios con el uso de estos sistemas informáticos, como eficiencia, seguridad y rapidez en las transacciones financieras. Los principales riesgos asociados a los sistemas de pago son, riesgo sistémico, riesgo operativo, riesgo legal y riesgo financiero.

Los riesgos asociados a los sistemas de pago se condensan en el siguiente cuadro (Cortina Morfín & Álvarez Toca, 2014):

Riesgo	**Descripción**
Sistémico	Riesgo generado por un evento dañino para el sistema que se propaga a diversos o todos los agentes en el sistema provocando el colapso de este. La falta de cumplimiento de las obligaciones por alguna de las instituciones financieras podría dar lugar a un efecto domino en el que las demás instituciones no pueden cumplir con sus obligaciones, generando inestabilidad en el sistema.
Operativo	Riesgos asociados a los sistemas informáticos, redes informáticas y personal a cargo de la operación y administración de los sistemas.

	Errores en el diseño o implementación de los sistemas o redes informáticas, errores humanos o incumplimiento de procedimientos, así como descomposturas en el hardware de los sistemas o redes informáticas.
Legal	Riesgo por una incorrecta implementación de los lineamientos dentro del sistema, marcos legales deficientes o erróneos, o defectos en el sistema que produzca resultados o interpretaciones incorrectas de los resultados. La implementación incorrecta de los procedimientos y reglas en el sistema, así como la falta o la incorrecta legislación pueden producir pérdidas económicas.
Financiero	Riesgo generado por parte de una institución financiera que sea incapaz de cumplir con sus obligaciones (insolvencia) o la falta de suficientes recursos para cumplir sus obligaciones en tiempo y forma.

Tabla 2. Riesgos asociados a los sistemas de pago.

1.3.5 Los Sistemas de Pago en México

Sistemas de pago de alto valor en México:

a) SIAC (Sistema de Atención a Cuentahabientes de Banco de México)
b) SPEI (Sistema de Pagos Electrónicos Interbancarios)
c) DALI (Sistema de Depósito, Administración y Liquidación de Valores)
d) SPID (Sistema de Pagos Interbancarios en Dólares)

Sistemas de pago de bajo valor en México:

a) SICAM (Sistema de Cámaras)
b) TEF (Transferencia Electrónica de Fondos)
c) Directo a México
d) Domiciliación de Recibos
e) Compensación de Documentos

1.3.6 Sistema de Atención a Cuentahabientes (SIAC)

Es un sistema informático de liquidación bruta en tiempo real operado por el Banco de México desde el año 1990, se encarga de administrar las cuentas corrientes que por ley tienen las instituciones financieras y las instituciones del sector público en Banxico, la principal función del SIAC es administrar las cuentas corrientes, así como proveer de liquidez a sus participantes, por esta razón el SIAC juega un papel fundamental en el sistema financiero mexicano (Banxico, 2016).

El sistema SIAC permite la transferencia de fondos entre las cuentas de los participantes sin restricción en el monto, sin embargo, el esquema de funcionamiento recomendado y alentado por Banxico consiste en hacer uso del sistema SPEI para realizar transferencias de fondos monetarios no sólo por ser un sistema mucho más moderno, sino porque este sistema fue diseñado precisamente para llevar a cabo transferencias de alto valor en tiempo casi real (Banxico, 2016).

El SIAC mantiene conexiones en tiempo real con los sistemas SPEI y DALI con la finalidad de poder realizar transacciones entre ellos, permitiendo realizar transferencias entre las cuentas de las instituciones financieras, adicionalmente el sistema DALI provee de liquidez a las cuentas de las instituciones financieras en el SIAC (Banxico, 2016).

1.3.7 Sistema de Pagos Electrónicos Interbancarios (SPEI)

SPEI es un sistema informático de liquidación bruta en tiempo casi real desarrollado por el Banco de México desde el año 2000, inició sus operaciones en el año 2004, a través de SPEI es posible realizar transferencias electrónicas de dinero entre cuentas de depósito de las distintas instituciones financieras conectadas al sistema, de tal forma que es necesario que las instituciones financieras que hacen uso del sistema SPEI tengan una cuenta en dicho sistema con fondos suficientes para poder cubrir las obligaciones suscitadas durante la operación por parte del sistema, tener suficiente liquidez por parte de las instituciones financieras participantes en SPEI es muy importante ya que el sistema SPEI no ofrece crédito (Banxico, s.f.).

Las instituciones financieras que pueden conectarse al sistema SPEI son, administradoras de fondos para el retiro, casas de bolsa, casas de cambio, instituciones de crédito, instituciones de seguros, sociedades distribuidoras de acciones de sociedades de inversión, sociedades financieras de objeto limitado y sociedades operadoras de sociedades de inversión (Banxico, s.f.).

SPEI lleva a cabo su proceso de liquidación de transacciones en ciclos de aproximadamente tres segundos o cuando se genera una acumulación de 300 transacciones, cualquiera de los dos escenarios que ocurra primero, una vez liquidadas las operaciones los cambios se aplican inmediatamente en las cuentas de las instituciones financieras correspondientes, una vez aplicada la liquidación de la transacción esta es final e irrevocable. Además, es importante señalar que el sistema SPEI utiliza un sistema de liquidación híbrida, es decir, combina el método de la liquidación bruta con la compensación, el algoritmo del sistema SPEI determina de acuerdo a los ciclos de ejecución si la liquidación de la transacción puede llevarse a cabo con el saldo disponible en las cuentas de las instituciones financieras correspondientes, de manera que al finalizar cada ciclo de procesamiento el saldo final de las cuentas de las instituciones financieras es positivo o cero. De igual forma, durante el ciclo de liquidación, todas aquellas transacciones que no pudieron ser liquidadas son encoladas para poder liquidadas en el siguiente ciclo. Todas aquellas transacciones que queden pendientes durante el cierre de operación se cancelan (Banxico, 2016).

El sistema SPEI funciona continuamente de forma ininterrumpida las 24 horas del día, los 365 días del año, sin embargo, cuenta con un horario diario de operación de 19:00 horas (día anterior) a 17:35 horas, esto parece extraño pero se debe a la siguiente razón, el cierre de la operación no quiere decir que el sistema SPEI deja de funcionar o se apaga, en el periodo de cierre de operación del día, el sistema SPEI transfiere los saldos resultantes de toda las transacciones realizadas durante el día de operación a las respectivas cuentas en el SIAC de cada institución financiera. Por otra parte, durante el inicio del día de operación de SPEI las instituciones financieras participantes deben transferir fondos de su cuenta en el SIAC al sistema SPEI con la finalidad de poder tener liquidez para cumplir con sus obligaciones derivadas de las transacciones que serán llevadas a cabo durante el día de operación (Banxico, s.f.) (Banxico, 2016).

La comunicación entre el sistema SPEI y las instituciones financieras que se conectan con este sistema se lleva cabo por un canal cifrado, adicionalmente las transacciones, así como las notificaciones realizadas por SPEI son firmadas digitalmente (Banxico, 2016), lo que proporciona confidencialidad e integridad de la información.

El sistema SPEI realizaba 3,904 operaciones al mes por un importe de 388,762 millones de pesos en el momento que empezó a operar por primera vez en 2004, al cierre del año 2017 las operaciones mensuales de SPEI eran más de 48 millones por un importe de 23.6 billones de pesos (Juárez, 2018).

En términos de operaciones, si distribuimos las operaciones de manera uniforme, el sistema SPEI pasó de realizar aproximadamente 5.4 operaciones por hora, equivalentes 0.09 operaciones por minuto durante 2004, a realizar aproximadamente 66,660 operaciones por hora, equivalentes 1,111 operaciones por minuto durante 2017, esto es, las operaciones mensuales de 2004 a 2017 aumentaron más de 12 mil veces, mientras que los

importes aumentaron en más de 23 billones de pesos, lo que muestra un aumento exponencial en la adopción y uso de los sistemas informáticos para llevar a cabo transacciones financieras por parte de los clientes, ya sean personas físicas o morales, así como de las propias instituciones financieras.

Proceso general llevado a cabo por SPEI para procesar una transacción[2] (Bank For International Settlements, 2016):

1) Envío de solicitud de transferencia de fondos por parte de un cliente A (persona física o moral) a través de los servicios de la institución financiera A, hacia un cliente B (persona física o moral) quien utiliza los servicios financieros de una institución financiera B.
2) La institución financiera A verifica que el cliente A sea un cliente suyo, verifica la solicitud, que la cuenta del cliente A se encuentre activa y que tenga fondos suficientes para poder llevar a cabo la transacción. Si todo es correcto la institución financiera A aplicará un cargo sobre la cuenta del cliente A.
3) La institución financiera A envía una solicitud de transferencia de su cuenta hacia la cuenta de la institución financiera del receptor de la transferencia, en este caso, la institución financiera B. Esta solicitud la realiza la institución financiera A hacia el sistema SPEI.
4) El sistema de pagos SPEI realiza el proceso de *clearing*[3] *and settlement*[4], estos procesos corresponden a la validación y verificación, tanto de la transacción recibida como de los fondos en la cuenta SPEI de la institución financiera solicitante. Si todo es correcto SPEI aplica el cargo y el abono en las cuentas de las instituciones financieras, finalmente realiza la liquidación de la transacción.
5) SPEI notifica de la liquidación de la transacción tanto el emisor como al receptor de la transferencia de fondos.
6) La institución financiera B recibe la notificación del sistema SPEI, la institución financiera aplica un abono sobre la cuenta del cliente B.
7) Finalmente, la institución financiera B notifica al cliente B del abono aplicado sobre su cuenta. En este punto la transacción ha finalizado y los fondos han sido transferidos exitosamente del cliente A hacia el cliente B.

[2] El proceso de transacción en el sistema SPEI aquí presentado, es una representación de alto nivel, no incluye todos los detalles correspondientes a una transacción real ni todos los posibles escenarios que pudieran surgir durante el procesamiento real, sin embargo, proporciona una visión general del funcionamiento de SPEI.

[3] *Clearing* es el proceso de transmisión, reconciliación y en algunos casos, confirmación de las transacciones antes de llevar a cabo la liquidación, potencialmente incluye el *netting* y el establecimiento de posiciones finales para la liquidación (Bank For International Settlements, 2016).

[4] *Settlement* es la liquidación de la transacción, una vez que la transacción se ha liquidado esta ha finalizado (Bank For International Settlements, 2016).

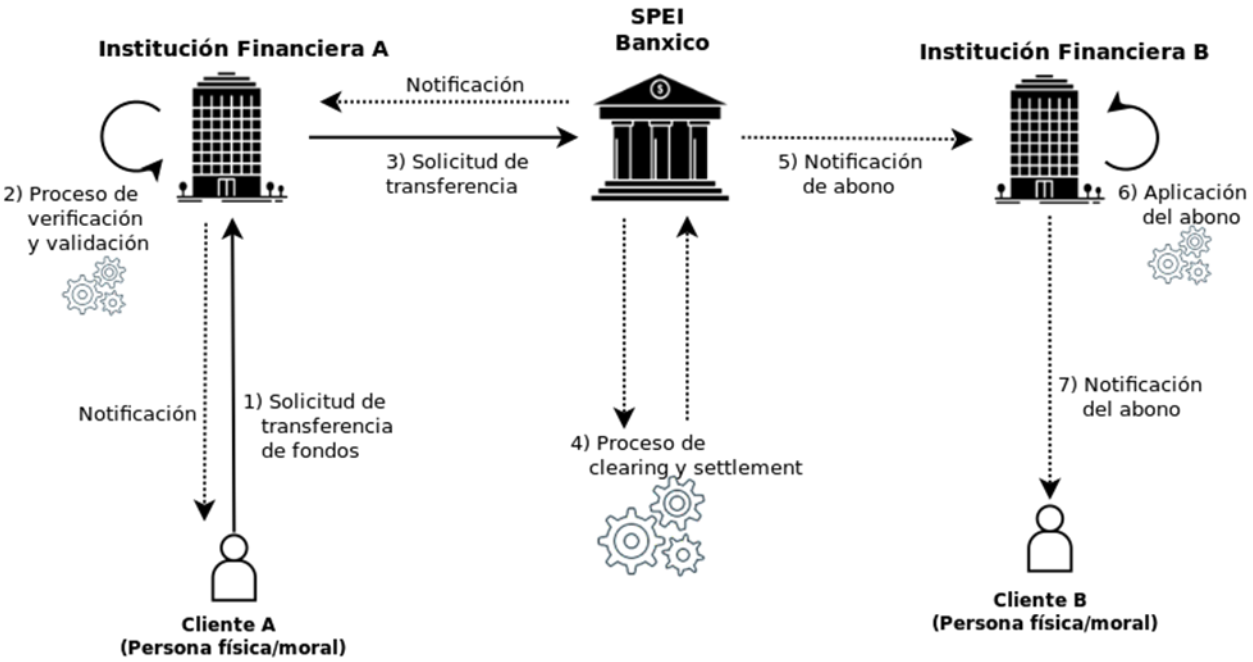

Figura 2. Proceso de transferencia de fondos a través del sistema de pagos SPEI.

Por sus puesto que si durante el proceso de transferencia de fondos a través del sistema SPEI ocurre algún error, la transacción es cancelada, por ejemplo, un error durante la verificación y validación de la solicitud de transferencia de fondos en la institución financiera A cancelará la transacción, ni siquiera se realizará el envío de la transacción hacia el sistema SPEI, este error podría ser que el cliente que realiza la solicitud no cuenta con los fondos suficientes para poder llevar a cabo la transacción. De la misma forma, un error durante el proceso de *clearing* por parte del sistema SPEI resultará en la cancelación de la transacción, esto podría ser debido a la falta de fondos por parte de la institución financiera a la cual se le tiene que aplicar el cargo.

La transacción de transferencia de fondos por medio de SPEI también podría ser cancelada debido a una solicitud inválida, es decir, una transacción financiera apócrifa generada por un delincuente informático, tanto en el lado de la institución financiera como en el sistema SPEI.

Es importante señalar que los clientes de las instituciones financieras, ya sean personas físicas o morales, no pueden interactuar directamente con el sistema SPEI, los clientes de las instituciones financieras siempre llevan a cabo la interacción, esto es, sus transacciones financieras, a través de la institución financiera con la cual tienen un contrato que les suscribe el derecho de una cuenta, que dependiendo del tipo de cuenta y servicios contratados, le permitirá al cliente guardar fondos en la cuenta, disponer de ellos o contar con una línea de

crédito. Para los clientes es irrelevante el funcionamiento de la transacción financiera, la comunicación con los sistemas de pagos o los procesos involucrados, para el cliente toda la transacción es transparente.

El funcionamiento de las transacciones financieras es relevante para los clientes cuando este proceso no es seguro, porque esto pone en riesgo los activos que el cliente puso en custodia de la institución financiera con la cual contrajo un contrato. Cuando los procesos en la transacción financiera son demasiado complejos, difíciles de administrar o involucran la comunicación con demasiados sistemas intermedios, se traduce en un alto costo asociado a la transaccionalidad financiera, este costo inevitablemente se transfiere al cliente. En este sentido, el funcionamiento y los procesos asociados con la transacción se tornan relevantes para los clientes si los procesos les proporciona un beneficio o por el contrario un perjuicio, de otra forma, para el cliente este proceso es trivial, ya que su interacción se lleva a cabo con un intermediario, la institución financiera.

1.3.8 Sistema DALI

DALI es un sistema informático en tiempo casi real que realiza la liquidación de sus transacciones en ciclos que se ejecutan aproximadamente cada dos minutos, es operado por el INDEVAL (Institución para el Depósito de Valores, S.A. de C.V.), el sistema inicio operaciones en el año 2008, este sistema realiza transacciones financieras en las cuales se registran y liquidan los títulos de deuda, acciones emitidas, operaciones de compraventa en directo, reportos[5] y prestamos de valores que sus depositantes realizan en el mercado financiero. Los participantes del sistema DALI pueden ser instituciones financieras como casas de bolsa y bancos, nacionales y extranjeros, estos tienen la posibilidad de realizar transferencias de su cuenta en el sistema DALI hacia los sistemas SPEI y SIAC en cualquier momento.

El sistema DALI tiene un horario de operación de las 7:46 horas a 16:15 horas, a partir de las 16:20 horas el sistema realiza la transferencia de los saldos generados durante el día de la operación de las instituciones financieras en el DALI hacia sus respectivas cuentas en el sistema SPEI.

[5] "Operación de recompra en la que una entidad financiera vende a un inversor un activo con el compromiso de comprarlo en una fecha determinada a un precio determinado" (BBVA, 2015).

1.3.9 Sistema de Pagos Interbancarios en Dólares (SPID)

Sistema informático que permite realizar transacciones financieras para la transferencia electrónica de dólares, el sistema es administrado por el Banco de México, inicio su operación en el año 2016, los participantes de este sistema son instituciones financieras de banca múltiple y de desarrollo, dichas instituciones únicamente pueden ofrecer a sus clientes el servicio de transferencia de dólares si estos son personas morales cuyo domicilio sea el territorio mexicano (Banxico, 2016).

De igual forma que con el sistema SPEI, las instituciones financieras participantes del sistema SPID deben transferir fondos de sus respectivas cuentas en dólares en el sistema SIAC hacia el SPID con el objetivo de tener liquidez y poder cumplir con las obligaciones producidas por las transacciones financieras durante los días de operación (Banxico, 2016).

En su forma más esencial, el proceso de transferencia de dólares entre instituciones financieras a través del sistema SPID es similar al proceso que se lleva a cabo en el sistema SPEI, a continuación, se describe de manera general el proceso de transferencia de dólares a través del sistema SPID:

1) El cliente A (persona moral) solicita a la institución financiera A la transferencia de dólares de su cuenta hacia la cuenta del cliente B (persona moral).
2) La institución financiera valida la solicitud el cliente A, verifica su cuenta y que cuenta con fondos suficientes para llevar a cabo la transacción. Si todo es correcto se aplica un cargo sobre la cuenta del cliente A.
3) La institución financiera realiza una transacción hacia el sistema SPID solicitando una transferencia de fondos de su propia cuenta en el sistema SPID hacia la cuenta de la institución financiera B en el SPID.
4) El sistema SPID realiza el proceso de *clearing and settlement*, es decir, valida la transacción, verifica que existan fondos suficientes en la cuenta del solicitante, adicionalmente aplica procesos de detección de lavado de dinero. Si todo es correcto aplica el cargo y el abono sobre las cuentas de las instituciones financieras participantes en la transacción y liquida la transacción.
5) El sistema SPID envía una notificación hacia la institución financiera B informándole del abono que se aplicó sobre su cuenta en el SPID.
6) La institución financiera B aplica un abono sobre la cuenta del cliente B.
7) La institución financiera B notifica a su cliente B que aplicó un abono sobre su cuenta.

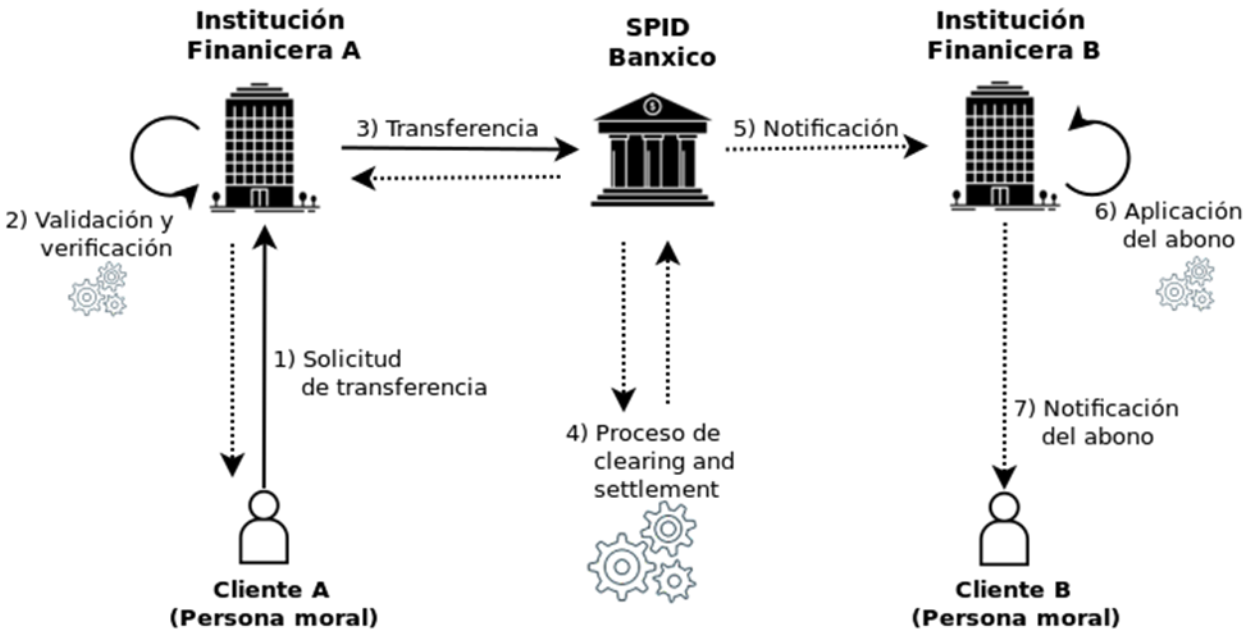

Figura 3. Proceso de transferencia de fondos a través del sistema SPID. Adaptado de Información del Sistema de Pagos Interbancarios en Dólares (SPID), por Banxico.

1.3.10 Sistema Directo a México

Directo a México es un sistema informático desarrollado por el Banco de México que permite realizar transferencias electrónicas de dinero desde Estados Unidos hacia personas físicas residentes en México, esto es, enviar dólares de Estados Unidos hacia México, haciendo uso de la conexión entre los sistemas de pagos del Banco de México y de los Bancos de la Reserva Federal de los Estados Unidos, establecida a finales del año 2003 (IME, 2016).

Los operadores del sistema Directo a México son el Banco de México y la Reserva Federal de los Estados Unidos, en el año 2003 conectaron sus sistemas de pago con la finalidad de permitir el envío de pago a los ciudadanos estadounidenses pensionados por el gobierno de Estados Unidos que radican en México. Durante el año 2004 se permitió la transferencia de fondos desde cualquier institución financiera estadounidense conectada al sistema Directo a México hacia cualquier cuenta bancaria en México (Banxico, 2016).

Directo a México comenzó sus operaciones en 2004 permitiendo a los inmigrantes mexicanos en Estados Unidos realizar el envío de remesas hacia México a través de una cuenta bancaria en algún banco estadounidense suscrito al sistema Directo a México, por parte del emisor de dólares residente en Estados Unidos hacia la cuenta bancaria del beneficiario en territorio mexicano, esto de acuerdo a la información publicada por el Instituto de los Mexicanos en el Exterior (IME, 2016).

Al hacer uso del sistema Directo a México se debe tomar en cuenta no sólo el hecho de que ambas partes deben tener una cuenta con una institución financiera que tenga una conexión hacia el sistema mexicano Directo a México, también se debe considerar la comisión aplicada por transferencia aplicada por el banco estadounidense elegido para realizar la transferencia, ya que dependiendo del banco, las comisiones pueden ir desde un dólar hasta tres y medio dólares, la comisión sólo aplica para el emisor de dólares ya que al receptor no se le aplica ninguna comisión por la transacción financiera. Adicionalmente se debe destacar el hecho de que el dinero recibido por parte del beneficiario es en pesos mexicanos aplicando un tipo de cambio determinado por el sistema Directo a México, se utiliza el FIX[6] [7] de Banxico.

De manera similar al proceso de transferencia llevado a cabo por el sistema SPEI, el sistema de pagos Directo a México sirve como un intermediario entre las instituciones financieras para que estas puedan proveer de un servicio a sus clientes.

Proceso general de transferencia de fondos a través del sistema de pagos Directo a México (Banxico, 2016):

1) Un cliente A (persona física) residente en los Estados Unidos realiza una solicitud de transferencia de fondos de su cuenta bancaria en la institución financiera A, hacia la cuenta bancaria de un cliente B (persona física) residente en México.
2) La institución financiera A realiza la validación y verificación de la solicitud, verifica al cliente, su cuenta y que cuente con los fondos necesarios para poder realizar la transferencia. Si todo es correcto, la institución financiera A realiza un cargo en la cuenta del cliente A.
3) La institución financiera A realiza el mismo día de la solicitud por parte del cliente A, la solicitud de transferencia hacia el *Federal Reserve Bank*.
4) El *Federal Reserve Bank* lleva a cabo el proceso de *clearing and settlement*, es decir, la validación y verificación de la transacción, la verificación de fondos por parte de la institución financiera A. Si todo es correcto, se aplica un cargo sobre la cuenta de la institución financiera A.
5) Un día después de haber recibido la solicitud de transferencia (t + 1), el *Federal Reserve Bank* envía la transferencia hacia el sistema Directo a México.
6) El sistema Directo a México realiza el proceso de *clearing* sobre la transacción recibida, verifica y valida la transacción. Si todo es correcto realiza el tipo de cambio monetario correspondiente.

[6] Es el tipo de cambio en moneda extranjera determinado por el Banco de México.

[7] Financial Information Exchange (FIX), protocolo neutral de comunicación para el intercambio en tiempo real de transacciones de valores (Scott, 2019).

7) El sistema Directo a México envía la transacción hacia el sistema de pagos SPEI, para que este pueda liquidar la transacción.
8) El sistema SPEI realiza el proceso de *clearing and settlement*, verifica y valida la transacción, si todo es correcto, liquida la transacción y aplica un abono en la cuenta de la institución financiera B, que es la institución financiera en la cual el cliente B tiene contratados servicios financieros.
9) Una vez liquidada la transacción el sistema SPEI notifica a la institución financiera B del abono que se acaba de aplicar a su cuenta.
10) La institución financiera B recibe la notificación del sistema SPEI, aplica un abono en la cuenta del cliente B.
11) La institución financiera B notifica a su cliente B del abono que se aplicó sobre su cuenta.

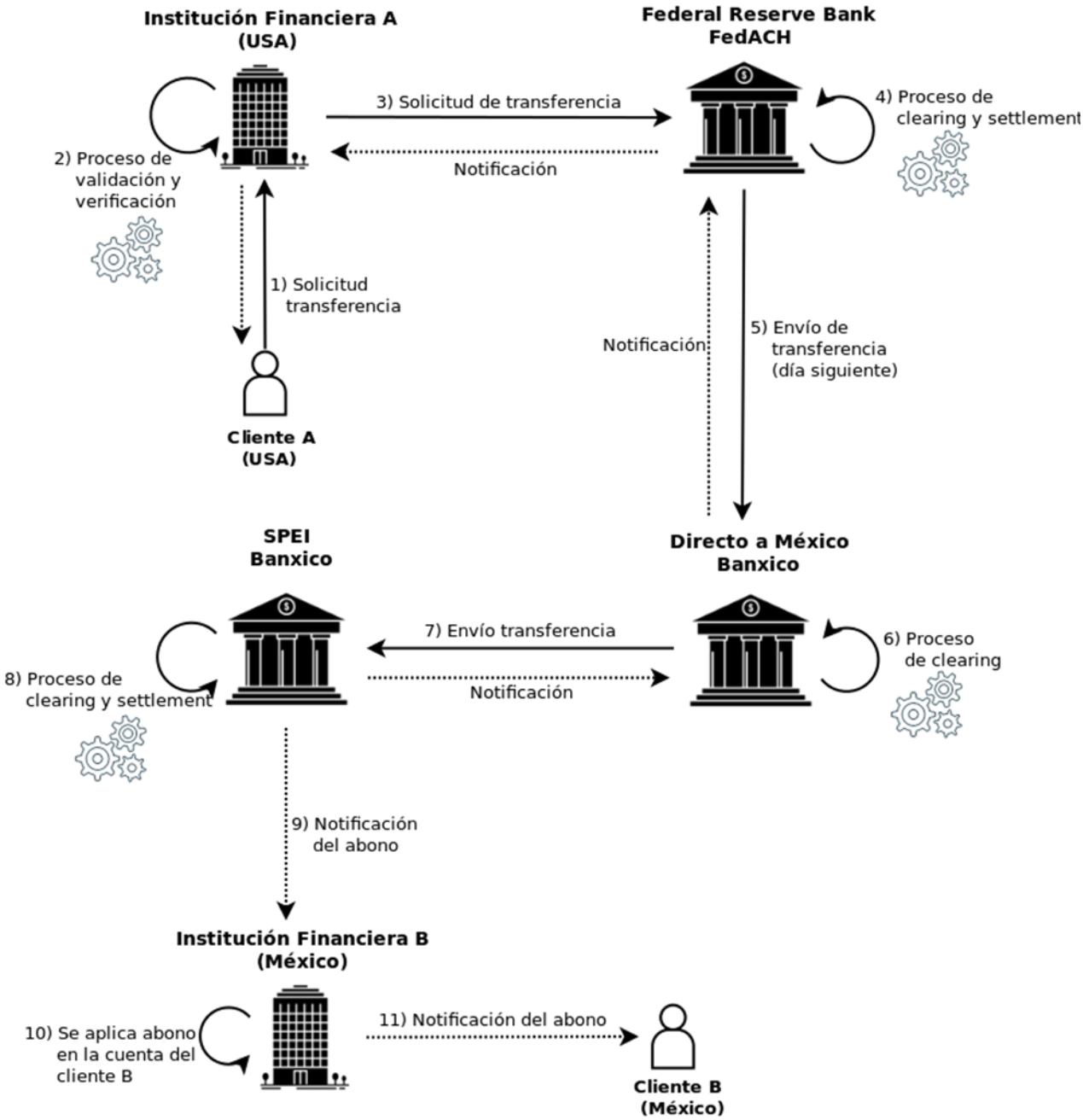

Figura 4. Proceso de transferencia de fondos a través del sistema Directo a México.

Cualquier error durante las validaciones de la institución financiera o en el *Federal Reserve Bank* resultará en la cancelación de la transacción, por ejemplo, el cliente es inválido o no cuenta con los fondos suficientes

Es indudable que el sistema Directo a México representa un gran avance financiero y tecnológico por parte del gobierno mexicano para ofrecer una forma rápida, segura y más económica de enviar dólares desde Estados Unidos hacia México, sin embargo, en una encuesta publicada (CONDUSEF, 2014) sobre las causas más comunes por las cuales los migrantes no hacen uso de los servicios financieros, muestra que los mexicano

residentes en Estados Unidos no saben cómo usar los servicios financieros, no saben hablar inglés, tienen desconfianza o simplemente temen por su situación migratoria ya que para abrir una cuenta bancaría en los Estados Unidos se requiere de una identificación oficial.

A pesar de que los inmigrantes mexicanos residentes en Estados Unidos pueden solicitar y utilizar la matrícula consular como documento de identificación para abrir una cuenta bancaria, esto no cambia su estado migratorio ni los acredita como residentes legales en los Estados Unidos, por lo que podrían correr el riesgo de ser deportados o simplemente les genera miedo y desconfianza el hecho de acudir a una institución financiera y presentar una identificación que no los acredita como ciudadanos legales o que los identifica como inmigrantes.

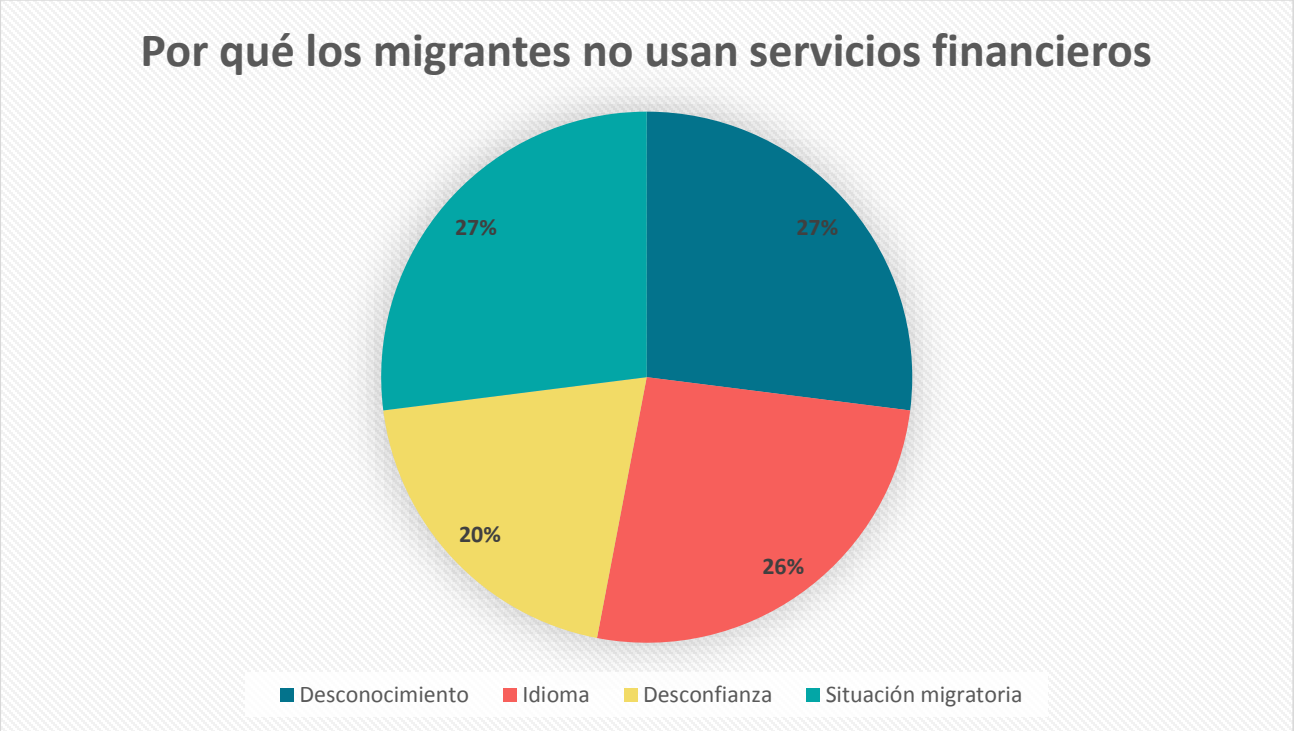

Figura 5. Razones de los migrantes mexicanos para no usar servicios financieros. Adaptado de REMESAS ¡No te dejes sorprender, haz rendir tus envíos! 2014 por CONDUSEF.

1.3.11 Otros Sistemas de Pago en México

Durante estas secciones se ha descrito de manera general el funcionamiento de los sistema de pago de alto valor que operan en México, así como el sistema de bajo valor Directo a México, juntos, estos sistemas forman parte de los principales componentes en el sistema financiero, sin embargo, no son todos los sistemas involucrados, además de los ya mencionados, el sistema financiero mexicano cuenta con sistemas de bajo valor

dedicados a la liquidación de documentos y valores financieros como acciones, cheques o transacciones con tarjetas de crédito y débito. La explicación de cada uno de estos sistemas, así como la descripción de estos en detalle va más allá de los alcances de esta obra, sin embargo, a continuación, se dará un panorama general del sistema financiero mexicano, específicamente de los sistemas informáticos que lo soportan, a través de un diagrama de alto nivel, es decir, el diagrama muestra los elementos más esenciales por lo que omite muchos de los detalles y complejidades del sistema financiero mexicano.

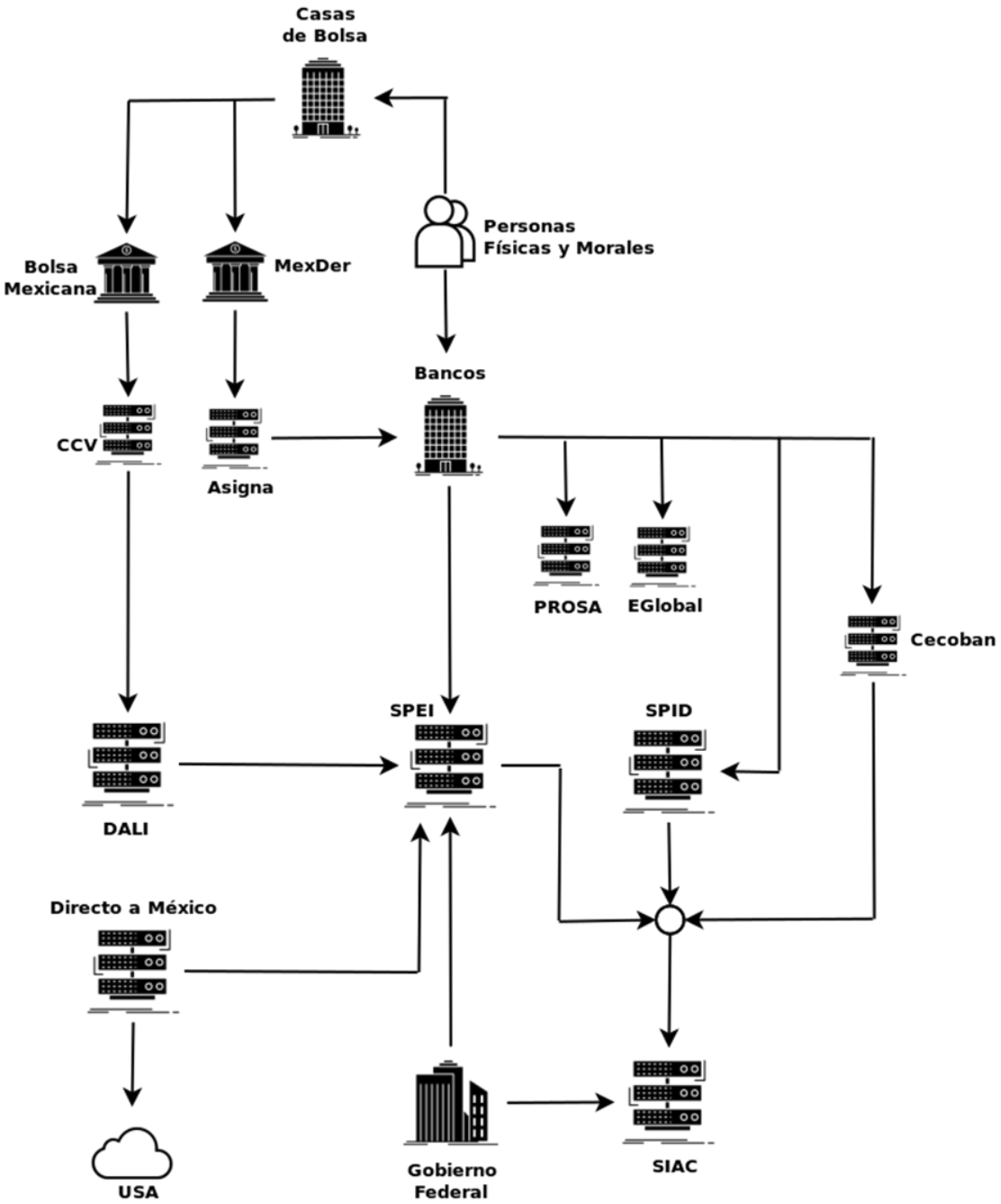

Figura 6. Estructura del sistema financiero mexicano. Adaptado de Política y funciones del Banco de México respecto a las infraestructuras de los mercados financieros. 2016, Banxico.

En la figura 6 se puede apreciar de forma abstracta la estructura del sistema financiero mexicano, tanto las personas físicas como morales tienen acceso a los servicios financieros ya sea a través de los bancos o de las casas de bolsa. Por otro lado, aunque todos los sistemas son importantes y cada uno desempeña una función clave dentro de todo el complejo sistema financiero, dos sistemas financieros sobresalen, SPEI y SIAC. El sistema SPEI funge como sistema central de pagos para llevar a cabo la mayor parte de transferencias de fondos de alto valor en el sistema financiero mexicano, mientras que el sistema SIAC se encarga de concentrar las cuentas de todos los actores que tiene acceso a los sistemas de pago, es decir, las instituciones financieras y el gobierno federal, el SIAC también se conecta con muchos de los sistemas de pago ya que es en el SIAC donde se transfieren los saldos de los sistemas de pago al cierre de la operación diaria.

Aunque el sistema financiero cuenta con diversos sistemas de pago, así como sistemas de compensación, la mayoría de ellos termina enviando las transacciones al sistema SPEI para que este las liquide, de esta forma el sistema SPEI puede aplicar los cargos y los abonos en las respectivas cuentas de las instituciones financieras que se encuentran en el sistema SPEI, manteniendo un sistema financiero centralizado que proporciona control, seguridad y garantía en la ejecución de las transacciones financieras. A su vez, las instituciones financieras podrán notificar del cargo o el abono en las respectivas cuentas de sus clientes en caso de que la transacción financiera involucre a los clientes de las instituciones financieras.

Una mención especial para los sistemas que aparecen en la figura 6, pero los cuales no han sido mencionados hasta este momento, tanto CCV (Contraparte Central de Valores) como Asigna son sistemas de compensación de valores que se encargan de compensar y liquidar activos manejados en la Bolsa Mexicana de Valores y por el Mercado Mexicano de Derivados respectivamente. El sistema Cecoban es también un sistema de compensación que se encarga de procesar los cheques, las transferencias diferidas y las domiciliaciones, el Cecoban se encarga de compensar y liquidar estas transacciones. Finalmente, los sistemas Prosa y EGlobal son los sistemas (*switches*) de las tarjetas de crédito y débito, estos sistemas se encargan de procesar todas las transacciones involucradas con tarjetas bancarias.

1.4 Hackers y Ciberdelincuentes

No es posible hablar de aquello que tanto se ve, oye y lee en los diversos medios de comunicación tradicionales y digitales, ya sean series televisivas, películas o noticias, me refiero a los famosos "*hackeos*" o ciberataques, sin antes definir uno de los términos informáticos más abusados, malinterpretados y sobrecargados de significados, la mayoría de las veces erróneos, me refiero a la palabra *hacker*.

Un *hacker*[8] [9] es aquella persona con una gran habilidad en la informática, tiene gran capacidad y creatividad para la resolución de problemas asociados con la informática, el término era usado durante 1960 para hacer referencia a los individuos con la capacidad de remover o "*hackear*" el exceso de código de un programa de computadora en un época donde las habilidades informáticas eran escasas (Rouse, Sjoholm, & Rosencrance, 2017).

El término *hacker* ha sido ampliamente utilizado por los medios de comunicación para referirse a los delincuentes informáticos, aquellas personas que cometen delitos haciendo uso de herramientas o habilidades informáticas, sin embargo, un hacker es aquella persona que, sin importar sus intenciones, cuenta con una gran habilidad y conocimiento en la informática, en cualquiera o todas las áreas, ya sea software, hardware o redes. Estas habilidades son utilizadas usualmente para acceder a sistemas y redes informáticas saltando todos los mecanismos de seguridad y acceso en ellos, por sí mismo el acto del hacking no es ilegal a menos que se lleve a cabo sin el permiso del dueño o dueños de los sistemas (Symantec, s.f.).

Ya que un *hacker per se* no es bueno o malo, es decir, un *hacker* no necesariamente lleva a cabo actividades ilegales, los *hackers* son clasificados usando una metáfora que consiste en determinar su "moralidad" de acuerdo con el sombrero que usan, dentro de esta clasificación hay tres grandes grupos, aunque podría haber más categorías. Tradicionalmente los *hackers* son clasificados como *white hat*, *black hat* y *grey hat*.

Los *hackers white hat* (sombrero blanco), también llamados *ethical hackers* (*hackers* éticos), son aquellos que utilizan sus habilidades informáticas para ayudar a las organizaciones a encontrar vulnerabilidades en sus sistemas o redes informáticas, así como solucionarlas. Estos *hackers* utilizan sus conocimientos para mejorar la seguridad informática en las instituciones ya sean públicas o privadas (Rouse, Sjoholm, & Rosencrance, 2017) (Symantec, s.f.).

Por otro lado, los *hackers black hat* (sombrero negro), son *hackers* que utilizan sus conocimientos para cometer actos ilícitos, como robar, borrar o cifrar información confidencial, esto último generalmente para pedir una determinada cantidad de dinero a cambio de descifrar la información. Entrar en sistemas o redes informáticas de forma no autorizada, así como el bloqueo de servicios proporcionado por sistemas informáticos. Todos estos actos son realizados con el fin de causar daño a una organización o persona, ya sea por motivos personales, económicos (obtener dinero) o políticos (Symantec, s.f.) (Rouse, Sjoholm, & Rosencrance, 2017).

[8] John Draper (*Capitan Crunch*), uno de los hackers más famosos, define la palabra *hacker* como: "A person who loves exploring technology and the challenge of figuring their way around security obstacles to beat the system" (Levy, 2014).

[9] En palabras de Kevin Mitnick, otra gran y famoso hacker: "A hacker is someone who figures out how to get around things that get in their way" (Levy, 2014).

Por último, los *hackers gray hat* (sombrero gris), son *hackers* que utilizan sus habilidades para acceder a sistemas o redes informáticas de forma no autorizada, así como encontrar vulnerabilidades en estos. A diferencia de los hackers *black hat*, los *gray hat* no tienen la intención de causar un perjuicio, en cambio sus motivos pueden ir desde la simple diversión al reto intelectual, a veces lo hacen para buscar dinero, cuando esto último ocurre generalmente el *hacker* o *hackers* notifican a los dueños del sistema informático que han encontrado una o más vulnerabilidades y que están dispuestos a revelar cuales son estas, a menudo también como solucionarlas, a cambio de una pequeña compensación económica. Los *hackers gray hat* son un intermedio o mezcla entre *white* y *black hat* (Symantec, s.f.) (Rouse, Sjoholm, & Rosencrance, 2017).

Hasta el momento se han proporcionado definiciones acerca de qué es un *hacker*, sin embargo, hago un pequeño paréntesis para hacer una aclaración sobre el término. Todas estas definiciones de hacker acotan el significado al área informática o computacional, sin embargo, existe también una concepción más amplia del término *hacker*, en la cual, un hacker es aquella persona que tiene una gran habilidad, conocimiento, creatividad y pasión, para crear soluciones a problemas y encontrar en estas soluciones una apreciación estética. En este sentido el término *hacker* no está limitado a ningún área del conocimiento y puede ser perfectamente aplicado a una amplia variedad de personas de distintas disciplinas (hack.org, 2014).

Ahora que ya se definió el término hacker, la pregunta pertinente es, cómo llamar a quienes cometen delitos por medio de los conocimientos o tecnologías informáticas. Podemos utilizar el término *hacker black hat*, también podemos usar los términos, delincuente informático, ciberdelincuente, criminal informático o cibercriminal. Adicionalmente podemos utilizar el término *cracker*, este último es una denominación usada para referirse a los *hackers black hat*.

1.5 Ciberataques al Sistema Financiero Mexicano

A finales de abril de 2018 se cometió un ataque cibernético que afectó el sistema mexicano SPEI (López, 2018), tras el cual se calcula que se robaron entre 300 y 400 millones de pesos, el ataque fue dirigido a un subconjunto de instituciones financieras que se conectan con el sistema de pagos SPEI, a través del cual los delincuentes informáticos crearon solicitudes apócrifas de transferencia de fondos desde cuentas bancarias falsas, para posteriormente poder llevar a cabo el retiro de los fondos en efectivo.

A pesar de que durante el ataque de abril de 2018 no se vieron comprometidas las cuentas personales de los usuarios de las instituciones financieras, ni tampoco sus recursos monetarios, el ataque representa una señal

de alerta sobre las instituciones financieras, así como de sus posibles vulnerabilidades ante ataques cibernéticos tanto del nivel registrado, como posibles amenazas de mayor magnitud, ya que la tendencia de las diversas industrias económicas así como de las áreas de conocimiento, es integrarse cada vez más con la informática, no sólo a nivel tecnológico sino también a nivel intelectual creando nuevas áreas de conocimiento, de manera que la sociedad se tiene que adaptar a los nuevos cambios tecnológicos, esto incluye a los sectores de la sociedad que se dedican a llevar a cabo actividades ilícitas, estos segmentos de nuestra sociedad simplemente se están y se seguirán adaptando tal y como cualquier organismo vivo se adapta a un nuevo entorno para poder sobrevivir, por esta razón los ataques informáticos contra cualquier institución pública o privada no deben ser demeritados.

Con respecto del hackeo cometido contra el sistema mexicano SPEI, la consultaría en seguridad informática Tekium, reveló que los ciberdelincuentes que llevaron a cabo el ataque se encontraban infiltrados dentro de los sistemas comprometidos desde un periodo de más de un año antes de realizar el ataque cibernético, es decir, los delincuentes informáticos tenían acceso de forma anticipada a los sistemas financieros de las entidades bancarias que fueron afectadas, ya que el ataque no se realizó dentro del sistema SPEI sino a través de cinco instituciones financieras que se conectan a este sistema para llevar a cabo las transferencias financieras de forma electrónica, hasta el momento no hay ningún detenido, ni se conoce a los responsables (Notimex, 2019).

Estar durante tanto tiempo dentro de los sistemas de las instituciones financieras comprometidas les permitió a los ciberdelincuentes poder estudiar con detenimiento no sólo del funcionamiento del sistema SPEI para poder llevar a cabo su ataque, sino también el funcionamiento de estas entidades financieras (Notimex, 2019).

La consultora de seguridad informática Tekium señala con respecto a la seguridad informática existente en las redes y sistemas informáticos utilizados por las instituciones financieras mexicanas lo siguiente (Notimex, 2019):

> *Del uno al 10, yo le sigo dando una calificación 5.5 al sistema financiero mexicano. En el ciberataque estaba en 4.0-4.5, se han hecho avances, pero el problema es que te estás enfrentando a gente sumamente sofisticada y esos niveles de sofisticación no los veo dentro de los bancos y falta mucha conciencia todavía.*

Durante la prestigiosa conferencia de seguridad informática "RSA *security conference*" celebrada en San Francisco, Estados Unidos, el experto en seguridad informática Josu Loza señala con respecto al hackeo que sufrió el sistema mexicano SPEI, que a pesar de que el ataque fue perpetrado por expertos informáticos y llevado a cabo con una planeación de varios meses incluso años, el factor más importante que permitió el

ciberataque fueron fallas en la arquitectura y seguridad de la red informática de las instituciones financieras mexicanas, así como la falta de control y medidas de seguridad por parte del sistema SPEI (Newman, 2019).

Los ciberdelincuentes fueron capaces de penetrar los sistemas de las instituciones financieras mexicanas a través de conexiones públicas de Internet debido a graves fallas en la seguridad de las redes informáticas de las instituciones financieras, la falta de puntos de control aunado a débiles medidas de seguridad y carencia de segmentación en la red informática, fueron factores claves que permitieron llevar a cabo el hackeo (Newman, 2019).

Uno de los grandes problemas en el combate de ataques informáticos es la falta de transparencia por parte de las instituciones afectadas, el miedo a sanciones económicas y al escarnio público, así como el riesgo de daño en la reputación empresarial son algunas de las causas por las cuales las organizaciones mantienen en secreto que han sido víctimas de un ataque informático, sin embargo, una de las claves para combatir los ataques cibernéticos estriba en la publicación de información de forma transparente, de manera que se propicie un proceso de cooperación por parte de organizaciones no sólo del mismo gremio sino también de otras instituciones que hayan sido sufrido de ataques informáticos, así como de instituciones públicas y privadas dedicadas a la seguridad informática, incluyendo universidades, dando lugar a una fuente de conocimiento compartida que coadyuve en la mejora continua la seguridad informática en las organizaciones, tanto privadas como públicas.

Josu Loza expone la falta de transparencia por parte de las instituciones financieras mexicanas con respecto a los ataques informáticos de la siguiente forma (Newman, 2019):

> *“Mexican people need to start to work together. All the institutions need to cooperate more," Loza says. "The main problem on cybersecurity is that we don’t share knowledge and information or talk about attacks enough. People don't want to make details about incidents public.”*

En 2017 en una nota publicada por el periódico Excelsior en su versión digital (Hernández & Lara, 2017), se informa de un *hackeo* cometido contra 9 instituciones financieras mexicanas, entre ellas la CNBV (Comisión Nacional Bancaria y de Valores) durante el año 2016, de acuerdo a informes realizados por la empresa informática BAE Systems (BAE Systems Applied Intelligence, 2017), se determina que México forma parte de uno de los objetivos de una campaña global de ataques cibernéticos planeados por un grupo de delincuentes informáticos, esto derivado de una lista de 255 direcciones IP pertenecientes a diversas compañías alrededor

del mundo, principalmente instituciones financieras públicas y privadas, los ataques se atribuyen al grupo de *crackers* llamado Lazarus.

A continuación, se muestra el mapa de los países objetivo por parte del grupo de *crackers* Lazarus, construido a partir de la lista de direcciones IP:

Figura 7. Mapa de los países objetivo para ser atacador por el grupo de hackers Lazarus. Obtenida de Lazarus Watering Hole Attacks: THREAT RESEARCH BLOG. 2017 por BAE Systems Applied Intelligence.

En enero del año 2018 la institución financiera BANCOMEXT[10] fue víctima de un ataque informático por parte de un grupo de *crackers* coreanos identificados por la empresa estadounidense en seguridad informática *FireEye* como APT38, un grupo posiblemente vinculado a Lazarus, el ciberataque llevado a cabo por el grupo de *crackers* APT38 fue detectado a tiempo por parte de BANCOMEXT, sin embargo, el ataque pudo haber dejado pérdidas por 110 millones de dólares (Agence France-Presse, 2018).

Los recientes ciberataques perpetrados en contra de las instituciones financieras mexicanas no son una casualidad, una excepción o algo que no se repetirá, los ciberataques no sólo suceden en México, ocurren a diario a nivel mundial, son una tendencia actual que lejos de disminuir comenzará a crecer y acentuarse más conforme las instituciones públicas y privadas adopten cada vez más y de forma inevitable, las tecnologías informáticas. Los ciberataques no se restringen a un sector económico o institución, todas las instituciones públicas y privadas, así como los ciudadanos, se encuentran expuestos a los ciberataques.

[10] Banco Nacional del Comercio Exterior (BANCOMEXT), es un banco del estado mexicano que provee financiamiento al comercio exterior mexicano, con el objetivo de mejorar el desarrollo del país (BANCOMEXT, 2018).

La falta de una estrategia pública de seguridad informática por parte del gobierno deja expuesta a la nación frente a grandes amenazas informáticas, un claro ejemplo de esto son los recientes ataques informáticos que han sido cometidos contra instituciones financieras, así lo señala el director en México de *The Software Alliance*, Kiyoshi Tsuru, al afirmar que es urgente que la actual administración en México se encargue de crear una agencia de ciberseguridad nacional que consolide estrategias y regulaciones para garantizar la seguridad de los ciudadanos, así como de las instituciones públicas y privadas (Bnamericas, 2019).

En una encuesta realizada por *Global State of Information Security* en 2018, se estima que el 78.6% de las empresas ha sufrido al menos un ataque informático durante los últimos 12 meses. Emmanuel Ruiz, *country manager* de *Check Point Software Technologies*, señala que las instituciones financieras así como las instituciones públicas son las que están más expuestas para convertirse en blancos de ataques informáticos, debido al tipo de información que procesan y almacenan, es decir, información confidencial y de valores, también advierte que los ataques informáticos lejos de terminar o disminuir seguirán ocurriendo no sólo con mayor frecuencia sino cada vez con mayor fuerza y complejidad (Hernández Armenta, 2019).

La empresa internacional de administración de riesgos Willis Towers Watson realizó una encuesta a nivel mundial sobre los riesgos asociados con ataques informáticos, los resultados revelan que México ocupa el primer lugar en América Latina en ataques informáticos, de acuerdo al estudio, entre 2017 y 2018, el 83% de empresas mexicanas fueron víctimas de al menos un ataque cibernético, posicionándolo a nivel mundial entre las 10 naciones con más ataques informáticos (González, 2019).

En términos económicos, el costo total por ataques informáticos durante el año 2019 es cercano a los dos billones de dólares, donde sólo el 30% de las organizaciones cuenta con mecanismos de protección ante estos ataques. Aunado a esto, el 59% de los empleados en las organizaciones no cuenta con los conocimientos suficientes para navegar de forma segura en el ciberespacio, lo que representa un riesgo para la empresa, ya que estos empleados se convierten en una vulnerabilidad dentro de la organización (González, 2019).

Eduardo Zamora, *country Manager* de *Fortinet* México, una empresa dedicada al desarrollo de tecnologías para la seguridad informática apunta que las redes y sistemas informáticos del sector público no utilizan las tecnologías de seguridad más innovadoras, ya que mucha de la infraestructura tecnológica con la cuentan es obsoleta, esto deja expuestas a instituciones públicas a los inminentes ataques informáticos. Además, señala el gran riesgo en el que se encuentra el país, ya que México no sólo se encuentra en el primer lugar en ciberataques en América Latina, también es el número seis a nivel mundial, peor aún, los ataques informáticos seguirán aumentando no sólo en número sino en fuerza (Forbes, 2019).

Para Zamora uno de los grandes obstáculos que tiene México para enfrentarse a los nuevos peligros digitales es la falta de cultura sobre seguridad y la indiferencia por parte del actual gobierno para dar prioridad al tema de la seguridad informática, ya que esto traerá grandes consecuencias futuras sino se cambia el rumbo del país para adoptar una estrategia nacional de ciberseguridad, esto es importante porque no sólo se trata de pérdidas económicas, también de una pérdida de imagen y reputación que después es muy difícil de recuperar. Zamora apunta que la riqueza de la economía mexicana es una de las razones por las cuales México es un país atractivo para los delincuentes informáticos, aunado a esto, señala una falta de transparencia por parte del gobierno federal para informar con datos precisos acerca de lo que está sucediendo actualmente en términos de ciberataques en México (Forbes, 2019).

Debido al actual panorama en ciberseguridad en el que se vive, que además se vislumbra aún más catastrófico, es necesario que las instituciones financieras públicas y privadas adopten las tecnologías más recientes e innovadoras que les permitan mejorar la seguridad en el procesamiento, almacenamiento y transmisión de la información que manejan, esto no solo dará tranquilidad y estabilidad en el sistema financiero, permitirá contar con mejores sistemas informáticos, lo que se traduce en mejores servicios, no sólo más seguros, más eficientes y posiblemente más sencillos y económicos si se busca que el sector financiero se encuentre en la vanguardia tecnológica.

En este sentido, las instituciones financieras no deben concentrar sus esfuerzos únicamente en la compra de nuevas tecnologías, en cambio, deben incentivar, promover y llevar a cabo investigación científica y tecnológica dentro de estas instituciones, así como llevarla a cabo de manera conjunta con otras instituciones públicas y privadas, incluidas universidades, lo que permitirá a las instituciones financieras crear mecanismos y metodologías para la mejorar de la seguridad y eficiencia en los sistemas financieros, desencadenando un proceso continuo de mejora y vanguardia dentro del sector financiero, proceso del cual se benefician todos los actores involucrados, incluida la sociedad mexicana.

Es evidente por los hechos anteriormente expuestos, que los sistemas financieros actuales están muy lejos de ser perfectos, más grave aún los sistemas financieros actuales tanto los nacionales como los internacionales se han puesto a prueba por distintos grupos de ciberdelincuentes alrededor del mundo, demostrando que los sistemas financieros con los que contamos no son lo suficientemente seguros por lo que es necesario mejorar la seguridad de los sistemas informáticos en las instituciones financieras.

1.6 Dinero Digital

En la actualidad, a pesar de los riesgos informáticos existentes, de los diversos servicios ofrecidos por las instituciones financieras, aquellos que permiten transaccionar con dinero, son una de las grandes innovaciones de la economía moderna, la capacidad de adquirir bienes y servicios en tan sólo unos segundos mediante la posesión de una cuenta bancaria. Ya sea proporcionando una pequeña tarjeta de plástico a la persona que nos está cobrando, para que pueda deslizarla por una terminal, introduciendo los datos impresos de la tarjeta dentro de un formulario Web para realizar una compra en línea, o utilizando un *smartphone* que cuente con tecnología NFC[11] y tenga además instalada una aplicación de "cartera digital" a través de la cual se vincula el *smartphone* con una cuenta bancaria, permitiendo a la persona llevar a cabo transacciones financieras con su teléfono como si se tratase de una tarjeta de crédito.

De cualquier manera, ya sea utilizando físicamente el plástico asociado a nuestra cuenta bancaria, o proporcionando los datos de la tarjeta a una computadora localizada en alguna parte del mundo a través de nuestro navegador Web, la facilidad y comodidad proporcionada por los sistemas financieros, es innegable.

Tales beneficios no pueden ser sino el resultado de un complejo sistema a través del cual es posible llegar a un acuerdo sobre el envío de "dinero digital" de un punto a otro, de una cuenta bancaria a otra, de una entidad financiera a otra. Esto podría parecer simple cuando se trata de cuentas bancarias pertenecientes a la misma institución financiera, ciertamente lo es, debido a que la transacción financiera ya sea que implique un aumento o disminución de dinero, esto es, un cargo o abono, de uno o más clientes, se lleva a cabo de forma interna, es decir, todo el flujo digital de la transacción transcurre dentro de la red informática interna de la institución financiera, sin la necesidad de comunicarse con otra entidad financiera, por lo que el proceso es en este sentido menos complicada y mucho más rápida, los cambios son inmediatos.

El dinero digital es un cúmulo de registros en una base de datos donde se indica la cantidad de dinero asociada a una cuenta bancaria específica, un montón de unos y ceros ordenados de acuerdo con una codificación específica, por ejemplo, UTF-8[12], para poder representar cantidades numéricas decimales que son actualizadas a través del tiempo de acuerdo con las reglas que gobiernan el sistema informático que procesa y almacena el

[11] Near Field Communication (NFC), es una tecnología de comunicación inalámbrica de corta distancia para la transmisión de información entre dispositivos electrónicos sin la necesidad de establecer contacto físico entre ellos, puede ser utilizado por dispositivos como celulares o tabletas (NearFieldCommunication.org, 2017).

[12] Unicode Transformation Format 8 bits (UTF-8), es un sistema de codificación de longitud variable capaz de representar cualquiera de los caracteres del conjunto de caracteres Unicode, diseñado para ser compatible con la codificación ASCII y además evitar los inconvenientes asociados con las codificaciones UTF-16 y UTF-32 (UTF-8, 2019).

dinero digital o virtual. Cada vez que una persona con una cuenta bancaria realiza algún "movimiento" en su cuenta, esto es, cada vez que realiza la compra de algún bien o servicio, o recibe una entrada de dinero por parte de otra persona o entidad, por ejemplo, el depósito de su salario cada quincena por parte de la empresa en la que trabaja, el sistema se encarga de incrementar o disminuir el conjunto de números que representa la cantidad de dinero asociada a la cuenta bancaria.

El dinero digital en su forma más elemental no es diferente al dinero físico, el dinero digital de la misma forma que los billetes, las monedas, los metales preciosos o las gemas, son objetos a los cuales un conjunto de personas que comparten distintas características, una sociedad, le asigna un valor específico utilizando algún mecanismo de consenso con el cual todos los integrantes o la mayoría de ellos están de acuerdo con el valor que se le ha asignado a dicho objeto, de esta forma es posible intercambiar bienes o servicios de distinta naturaleza con personas desconocidas por medio de un objeto común, con el cual todos los actores del grupo concuerden con el valor que tiene, evitando los problemas y ambigüedades del comercio sobre cuánto de un producto o servicio "x" debo pagar o cobrar por otro producto o servicio distinto "z".

El dinero desde su invención actúa como intermediario en la transacción financiera, de la misma forma, el dinero digital busca ese cometido, la asignación de valor a un objeto que por sí mismo no tiene valor, tal y como los billetes y las monedas actuales no tienen un valor intrínseco, pero representan un determinado valor el cual es reconocido por todos los miembros de la sociedad.

A diferencia del dinero usado durante siglos por diversas sociedades a lo largo de la historia, ya sea en forma de objetos marinos, semillas, metales o papel (Harari, 2014), el dinero digital es la última abstracción de la humanidad para guardar valor, con la cual es posible llevar a cabo transacciones económicas sin la necesidad de almacenar, transportar y proteger objetos físicos que funcionan como dinero[13], desplazarse físicamente o de utilizar la misma "moneda".

Como ya se mencionó, las transacciones financieras llevadas a cabo por una misma institución financiera son, en su proporción, sencillas, sin embargo, cuando se trata de operaciones que involucran instituciones financieras diferentes, el intercambio de información entre ellas comienza a adquirir un nivel mayor de complejidad, no sólo por el hecho de que diferentes instituciones financieras se encuentran en ubicaciones físicas distintas, sino porque ninguna de ellas utiliza el mismo sistema informático para operar sobre esos números que representan el dinero digital, a pesar de que existen lineamientos en la forma en la que operan las instituciones financieras, este intercambio de información entre entidades heterogéneas conlleva riesgos de seguridad informática, como

[13] Sin embargo, estas tareas de almacenamiento, transporte y protección del dinero digital quedan a cargo de las instituciones financieras.

el robo de información, manipulación de la misma, así como interrupción o bloqueo de la comunicación entre los sistemas informáticos que intercambian información, la mitigación, prevención y administración de estos riesgos aumenta la complejidad y los costos asociados a las transacciones financieras.

Es entonces, una tarea no trivial llevar a cabo una transacción financiera, esto es, poder aumentar o disminuir el saldo de una cuenta bancaria cuando se trata de cuentas bancarias correspondientes a distintas instituciones financieras, la situación se complica aún más cuando las diferentes instituciones financieras involucradas en una transacción financiera se encuentran en países distintos, ya que no sólo se tienen que enfrentar los problemas inherentes a la transacción entre organizaciones y naciones diferentes, ahora se añade una complicación más, el tipo de moneda que utilizan ambas instituciones financieras seguramente es distinto, incluso aunque se trate de dinero digital, este debe mantener las mismas características del dinero físico, esto incluye el tipo de cambio entre diferentes monedas.

Pero incluso dentro de los problemas derivados de las operaciones interbancarias, nacionales e internacionales, existe uno que es de vital importancia y que sobresale de entre todos, la seguridad, tal y como ya expuso en la sección anterior.

La seguridad es materia primordial cuando se trata de transacciones financieras, porque al igual que con el dinero físico, el dinero digital es susceptible de delitos, aunque los delitos que se pueden cometer en contra del dinero digital tienen similitud con los cometidos contra el dinero físico, las diferencias son más que sutiles.

El dinero digital opera en una dimensión distinta al dinero que está dentro de nuestras carteras o bolsillos, de acuerdo a lo ya expuesto, los ciberdelitos asociados al robo de dinero virtual se lleva a cabo de forma diferente a como tradicionalmente se realizaban los robos contra instituciones financieras por medio de hombres armados y encapuchados, ahora los delincuentes son llamados ciberdelincuentes, ya que hacen uso de conocimientos o tecnologías informáticas para obtener acceso a los sistemas financieros de forma no autorizada, lo que les permite llevar a cabo transferencias monetarias hacia cuentas bancarias y posteriormente poder disponer del dinero en forma de efectivo.

1.7 Bitcoin, mucho más que una moneda digital

El sistema electrónico de [dinero en] efectivo entre pares[14], es un sistema informático desarrollado por un grupo de personas o persona, bajo el seudónimo de Satoshi Nakamoto[15], quién el 31 de octubre del año 2008 publicó[16] un documento con la descripción y especificación[17] de Bitcoin (Quentson, 2016). Un sistema informático distribuido entre pares que representa una alternativa a los sistemas financieros existentes, con la finalidad de poder llevar a cabo transacciones financieras entre personas de forma directa sin la necesidad de un intermediario, es decir, llevar a cabo intercambio de dinero de forma directa sin la intervención de una institución financiera, todo esto a través de la red mundial Internet, utilizando una moneda virtual creada y administrada por el propio sistema (Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008).

El sistema desarrollado por Nakamoto no es cualquier sistema informático, es un sistema basado en criptografía asimétrica, funciones criptográficas *hash*, sistemas distribuidos *peer-to-peer*, estadística, teoría de juegos, y una fórmula matemática que determina el número máximo de bitcoins[18] que serán producidos por el sistema.

Cada uno de los elementos utilizados para crear Bitcoin, de forma individual, son conceptos que existen desde hace tiempo y han sido ampliamente utilizados en diversos campos del conocimiento, las funciones hash y los sistemas *peer-to-peer*, son relativamente nuevos, ambos son resultado de la era computacional. Una de las primeras funciones hash es MD2[19], desarrollada en 1989 por Ronald Rivest como un mecanismo para comprobar la integridad de la información (md5hashing, s.f.). Por su parte, uno de los primeros sistemas distribuidos *peer to peer* es USENET, desarrollado en 1979 por Jim Ellis y Tom Truscott en la universidad de Duke como un sistema capaz crear una red privada de computadoras, permitiendo intercambiar información de

[14] A Peer-to-Peer Electronic Cash System (Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008).

[15] Desde la aparición de Bitcoin se han hecho muchas especulaciones sobre la verdadera identidad de Satoshi Nakamoto, sin embargo, hasta este momento se desconoce su verdadera identidad.

[16] La primera publicación del documento de Nakamoto fue a través de una lista de correo electrónico criptográfica, sólo para aquellos que se encontraban dentro de dicha lista.

[17] Aunque el documento publicado por Nakamoto contiene la especificación general del funcionamiento de Bitcoin, no se detallan muchos de sus elementos, aunado a ello, Bitcoin ha evolucionado constantemente desde su concepción en 2008 al ser un proyecto *open source*.

[18] Satoshi Nakamoto nunca especifica en su documento el nombre de las monedas que son creadas por su sistema, sin embargo, el sugerente nombre del sistema (Bitcoin) y su posterior uso hicieron bastante natural y obvio llamar a la moneda digital, bitcoin.

[19] La función hash MD2 ya no es considerada como una función criptográficamente segura, por lo que su eso está totalmente desaconsejado.

una computadora a otra sin la necesidad de un sistema de control central, fue concebido como una alternativa a la red ARPANET[20], cuyo acceso estaba restringido a ciertas universidades y centros de investigación (Oram, 2001) (Bonnet, 2010).

Los orígenes de la teoría de juegos son ligeramente previos a la era computacional, comienzan con la aparición de las publicaciones de John Von Neumann[21] [22], *On the Theory of Games of Strategy* publicada en el año 1928 en idioma alemán, fue la primera publicación de Neumann sobre la teoría de juegos, y posteriormente *Theory of Games and Economic Behaviour* publicada en coautoría con Oskar Morgenstern en el año 1944, ambas, forman parte de las obras más importantes e influyentes en la teoría de juegos, fundaron las bases teóricas para una disciplina inexistente durante el tiempo en que se llevaron a cabo las publicaciones. Neumann y Morgenstern establecen conceptos que permiten modelar matemáticamente las decisiones de agentes racionales bajo un contexto de interacción interdependiente entre cada uno de los agentes, uno de los principales conceptos derivados de esto es el juego de suma cero, en el cual la utilidad derivada de la victoria o perdida de los jugadores (agentes) está proporcionalmente balanceada con las pérdidas o ganancias de los otros participantes (Ferguson T. , s.f.) (Beebe, 2019) (Library of Economics and Liberty, s.f.).

Por otra parte, la criptografía es mucho más antigua, el emperador Julio Cesar (100 a.C. – 44 a.C.) utilizaba un algoritmo criptográfico[23] para intercambiar información de forma segura, se trata de un algoritmo de cifrado por sustitución, en donde se reemplaza cada una de las letras del abecedario por otra letra del abecedario de acuerdo a un número determinado de posiciones adelante con respecto a esa letra, por ejemplo, una sustitución usando el número 3 como llave, significa que cada letra del abecedario debe ser reemplazada por la siguiente letra 3 posiciones adelante, es decir, la letra A sería reemplazada por la letra D, la letra B por la letra E y así sucesivamente, esto se repite hasta que se han reemplazado todas las letras de la información que se quiere ocultar, el algoritmo de descifrado funciona de forma inversa para la información que se encuentra cifrada (Paar & Pelzl, 2010).

[20] ARPANET es considerada la primera red de computadoras por conmutación de paquetes, se le puede considerar como el origen de Internet, fue creada por la agencia estadounidense de investigación científica y tecnológica ARPA (ahora DARPA).

[21] Considerado el padre de la teoría de juegos y también uno de los padres de la computadora electrónica digital. La esencia de la arquitectura de computadoras Von Neumann todavía se encuentra en las computadoras modernas.

[22] *"There are two kinds of people in the world: Johnny von Neumann and the rest of us"* (Library of Economics and Liberty, s.f.).

[23] El algoritmo criptográfico utilizado por Julio Cesar actualmente se le conoce como cifrado Caesar, uno de los cifrados por substitución más simples y uno de los primeros que se enseñan en los cursos introductorios de criptografía.

Lo novedoso en el sistema de Nakamoto es la forma en la que se combinan los diferentes elementos para dar lugar a un tipo de sistema informático que no se había visto antes, un sistema criptográficamente seguro, verificable y distribuido que funciona únicamente bajo las instrucciones de los algoritmos que lo conforman, sin una red o sistema central que lo dirija, liberándolo de las vulnerabilidades de las cuales somos objeto los seres humanos, ya que el algoritmo se ejecuta siempre igual, obteniendo resultados predecibles y esperados, en cambio no se puede afirmar lo mismo de los seres humanos, quienes podemos ser inducidos para llevar a cabo una alteración o manipulación del sistema o de la información almacenada y procesada por este, o simplemente equivocarnos.

Una de las principales características de Bitcoin es la capacidad de proteger la información de ser manipulada de forma maliciosa, mientras se mantiene un registro histórico inmutable de todos los cambios llevados a cabo con la información que procesa el sistema.

La novedad radica en la transparencia e inmutabilidad de la información una vez que esta ha sido procesada por Bitcoin, proveer de transparencia e inmutabilidad a la información es una característica fundamental para muchos procesos financieros, económicos y legales, por ejemplo, mantener el saldo de una cuenta bancaria de forma inmutable una vez que está ha sido actualizada por un cargo o un abono, mantener la transparencia e inmutabilidad de documentos legales como el acta de nacimiento, el título de una propiedad, o de la información sobre el origen los materiales de los cuales está compuesto un dispositivo electrónico, de forma que sea posible verificar que los materiales fueron obtenidos de fuentes libres de conflicto[24] [25].

Antes de Bitcoin no existía un sistema capaz de mantener la inmutabilidad de la información mientras se lleva al mismo tiempo un registro público transparente que permite verificar la información por cualquier persona o entidad que requiera hacerlo, pero con la restricción de que sólo Bitcoin puede llevar a cabo el procesamiento de la información, siempre y cuando la transacción sea una transacción válida, es decir, una transacción que cumple con las lineamientos establecidos por el sistema, con lo cual no es posible alterar la información de Bitcoin de forma deliberada.

[24] El continente africano cuenta con la mayor reserva en el mundo de diversos metales, algunos de ellos como el coltán, son elementos necesarios para la fabricación de dispositivos electrónicos como celulares. Las minas africanas de donde se extraen estos metales se encuentran bajo el control de grupos paramilitares subversivos, que obligan a los habitantes de las aldeas aledañas, incluidos niños, a trabajar bajo condiciones deplorables en la extracción de metales (Love, 2017) (Buss, 2018).

[25] Países como China, Perú, Colombia, Bolivia, Brasil, Indonesia y Myanmar se suman a la lista de países en los cuales se lleva a cabo la extracción ilegal de metales bajo el mando de grupos militares (Jamasmie, 2017).

Bitcoin se protege utilizando algoritmos criptográficos para asegurarse que cada una de las transacciones son públicas, transparentes y autorizadas, además de evitar la modificación maliciosa de la información y verificar su integridad. Bitcoin también se apoya en la estadística para asegurar que, si varios nodos de la red han sido subvertidos para tratar de engañar al sistema, no sean capaces de lograr su cometido, y una red distribuida de nodos (pares) que garantizan la accesibilidad de la información y la preservación de esta, juntos estos elementos, proveen seguridad, inmutabilidad, transparencia y redundancia de la información procesada, son estas características las que hacen de Bitcoin un sistema único en su tipo.

Desde la concepción de Bitcoin en 2008, su publicación en 2009, y su creciente adopción en esta primera década, aunado a la gran efervescencia que hay a su alrededor, ha llevado a varias instituciones financieras no sólo a considerar en los últimos años el uso de bitcoin como una nueva moneda digital, sino en la creación de su propia moneda digital basada en el funcionamiento de Bitcoin.

Una de las primeras instituciones financieras que ha decido realizar un cambio de paradigma en el funcionamiento tradicional de las transacciones financieras es precisamente una de las empresa más grande de tarjetas de crédito en el mundo, Visa, quién en 2016 lanzó un sistema prototipo para realizar transferencias financieras nacionales e internacionales, dicho sistema se encuentra basado en el sistema Bitcoin, la finalidad del nuevo sistema informático creado por Visa es facilitar las transacciones financieras, reduciendo los costos, acelerando el tiempo en el que son llevadas a cabo, así como facilitar la operación asociada a estas transacciones financieras (Arnold, 2016).

Las acciones de Visa suponen un gran avance en la adopción de nuevas tecnologías informáticas para el mejoramiento y actualización del actual y viejo sistema financiero, sin embargo, supone también un peligro para la institución financiera internacional SWIFT (*Society for Worldwide Interbank Financial Telecommunication*), encargada de procesar la mayor parte de transacciones financieras alrededor del mundo (Arnold, 2016).

Santander es otra institución financiera que se ha unido a la fiebre Bitcoin, ya que en 2016 se unió con distintas instituciones financieras y una empresa emergente para llevar a cabo investigación sobre el funcionamiento del sistema informático Bitcoin, con la finalidad de poder aplicar el conocimiento para el mejoramiento de los procesos financieros automatizados a través de la creación de una moneda digital llamada *Utility Settlement Coin* (Agencia EFE, 2016) (EP, 2016).

La nueva tecnología que planea desarrollar Santander le permitirá llevar a cabo pagos y liquidaciones entre instituciones financieras de una forma más fácil y segura, esta nueva iniciativa sumada a los esfuerzos de la empresa Visa por renovar los sistemas informáticos encargados de llevar a cabo las transferencias financieras

nacionales e internacionales, es una señal de que las grandes instituciones financieras internacionales ven un gran potencial en la aplicación de los conceptos con los cuales funciona el sistema informático Bitcoin, pero más importante, es un reconocimiento por parte del sector financiero internacional de que hay problemas con los sistema financieros actuales que deben ser solucionados, así como una clara señal de que los métodos tradicionales de transferencias financieras existentes en la actualidad no son suficientes para cubrir las nuevas demandas que requiere el nuevo sector económico y financiero.

Lo cierto es que el mundo digital opera de forma distinta al mundo AFK[26], en el mundo digital, es posible realizar una cantidad indefinida de copias de un mismo objeto, un ejemplo clásico es el siguiente, suponiendo que tenemos un archivo correspondiente a una canción, podríamos copiar miles de veces la canción y cada una de las canciones sería la misma que la original en el sentido de cada una contiene la misma información, esto es, no habría ninguna diferencia relevante o significativa entre todas las copias y la canción original.

Esto supone una ventaja increíble, sobre todo para cierto tipo de situaciones, por ejemplo, compartir información alrededor del mundo, sin embargo, cuando se trata de representar dinero, se convierte en un problema, si tomamos en cuenta que una de las características más importantes del dinero físico es la protección natural que tiene contra ataques de *double spending*.

1.8 El Problema del *Double Spending*

El *double spending* (Pérez, Delgado, Navarro, & Herrera) consiste en poder utilizar el mismo "objeto" que ha sido seleccionado para representar el dinero, más de una vez, sin la correspondiente pérdida (literalmente) de dicho objeto una vez que este ha sido "gastado". En el mundo AFK, el problema del *double spending* es trivial porque una vez que utilizamos un billete o una moneda para comprar un bien o servicio, el dinero se ha "gastado", es decir, el billete o la moneda es entregada a quien nos ha provisto del bien por el que hemos pagado, por lo que no hay manera de que podamos gastarlo nuevamente por qué ya no poseemos ese objeto que representa dinero.

Por esta razón, los ataques que pueden ser cometidos en contra del dinero digital consisten en manipular la información que representa la cantidad de dinero correspondiente a una cuenta bancaria, por ejemplo, incrementar de forma indiscriminada la cantidad de dinero digital asociada a una o más cuentas bancarias, ya

[26] *Away From Keyboard*, literalmente "lejos del teclado" en español, es una abreviación utilizada en algunos contextos informales de la informática para hacer referencia a no estar delante a o estar lejos de una computadora.

sea directamente sobre las cuentas o creando transacciones “artificiales” que transfieran dinero digital de una cuenta a otra, lo que no necesariamente implicaría la disminución de dinero digital de una cuenta sobre otra cuenta bancaria, porque como ya se expuso, los “objetos digitales” pueden copiarse indefinidamente.

Por lo cual, un sistema que se encargue de operar con dinero digital deberá asegurarse que una vez que se ha “gastado” el dinero, la disminución correspondiente para la entidad que gastó el dinero, se vea reflejada para todos los actores del sistema económico, de manera que no sea posible utilizar de forma indefinida el dinero, así mismo, el correspondiente abono de dinero deberá ser reflejado en la entidad correspondiente, manteniendo el balance de entradas y salidas.

Los sistemas financieros actuales resuelven los problemas de *double spending* utilizando dos métodos, el primero consiste en utilizar una autoridad intermediaría que valide que las transacciones son correctas, es decir, que los cargos y abonos son válidos, estos intermediarios pueden ser otras instituciones financieras o instituciones gubernamentales.

El segundo método consiste en mantener sus sistemas privados, es decir, dichos sistemas mantienen la información financiera almacenada en bases de datos cuyo acceso es exclusivo para la propia institución financiera, esto podría parecer más que obvio, las instituciones financieras no deberían compartir la información correspondiente a las transacciones financieras con entidades no autorizadas, sin embargo, el flujo de dinero, las transacciones y la economía no son procesos locales ni exclusivos de una institución financiera específica o de una región geográfica aislada.

Los procesos financieros y económicos son flujos, muchas veces mundiales, transversales e interconectados, en los cuales convergen diferentes actores, como instituciones financieras, regulatorias y gubernamentales, así como un conjunto de diferentes entidades que hacen uso de los servicios financieros, a su vez estas entidades que hacen uso de los servicios financieros utilizan diferentes instituciones financieras para llevar a cabo sus transacciones, por otro lado, diferentes instituciones financieras se pueden encontrar en diferentes ubicaciones geográficas, por lo que son objeto de fiscalización por diferentes instituciones regulatorias y gubernamentales, todo esto da como resultado un conjunto entramado de conexiones financieras que no son aisladas, por lo que se requiere una base datos común que mantenga la coherencia de las transacciones financieras, permitiendo reflejar fielmente para todos los actores y entidades del complejo sistema económico financiero, los cambios llevados a cabo por cada una de las transacciones financieras que han ocurrido.

Sin embargo, hasta antes de la aparición de Bitcoin no se conocía un método efectivo que permitiera mantener de forma distribuida la coherencia de la información, en una base de datos global en la que todas las instituciones

financieras pudieran tener acceso y al mismo tiempo registrar los movimientos financieros realizados por sus clientes. La solución actual de las instituciones financieras es crear sistemas informáticos con bases de datos y redes informáticas privados y centralizados, con el fin de proteger los sistemas financieros contra el problema de *double spending*, así como mantener el control, seguridad y coherencia, administrativa y contable de las transacciones financieras.

1.9 La SWIFT y sus Ciberataques

SWIFT (*Society for Worldwide Interbank Financial Telecommunication*), es una red informática privada creada para proporcionar un sistema de comunicación rápido y seguro que permita el intercambio de transacciones financieras internacionales. La red de SWIFT está compuesta de nodos especializados cuya única función es llevar a cabo el intercambio de mensajes entre los distintos nodos en la red siguiendo un protocolo que garantiza la seguridad de los mensajes intercambiados, asegura que sólo los nodos de SWIFT pueden enviar y recibir mensajes, manteniendo la secrecía sobre los mensajes enviados, así como evitando la alteración de tales mensajes.

SWIFT es también un corporativo internacional fundado en 1973 compuesto por miembros de diferentes naciones que provee un servicio digital de mensajería financiera internacional, este servicio de mensajería utiliza estándares de comunicación segura para facilitar el intercambio de información financiera alrededor del mundo. SWIFT conecta a más de 11 mil instituciones financieras en más de 200 países a través de una plataforma tecnológica privada exclusivamente dedicada a llevar a cabo transacciones financieras, aproximadamente 24 millones de transacciones por día. Cabe resaltar que SWIFT no mantiene ni administra cuentas o balances, únicamente provee de un medio para la transmisión de la información por el cual cobra una cuota a sus miembros por cada mensaje enviado, el costo se determina en función del tipo y longitud del mensaje, así como del volumen total de transacciones que realice la institución financiera (SWIFT, s.f.) (Kagan, Society for Worldwide Interbank Financial Telecommunications (SWIFT), 2018) (Seth, 2019).

Considerando lo anterior, SWIFT juega un papel fundamental en el proceso de transacciones financieras a nivel mundial, ya que a través de la red de SWIFT es posible realizar millones de transacciones, resolviendo el problema de comunicación entre distintas instituciones financieras alrededor del mundo, aunque no se encarga del problema del *double spending*, ya que SWIFT sólo es un sistema de mensajería, por lo que le corresponderá a cada institución financiera mantener el correcto estado del dinero digital que administra.

De esta forma podría parecer que todo está resuelto y no queda nada por hacer, sin embargo, lo que podemos observar en los últimos años son distintos ataques cibernéticos exitosos cometidos contra la red SWIFT, exponiendo de esta manera la inseguridad de la red de comunicación para transacciones financieras.

La sede de SWIFT se encuentra en Bélgica, pero a pesar de esto y de que SWIFT asegura ser una organización neutral, esta organización internacional ha llevada a cabo acciones de censura contra los miembros de SWIFT, impidiéndoles realizar transacciones financieras con Irán debido a sanciones impuestas por los Estados Unidos hacia dicho país (Comfort, 2018) (RT News, 2018), dejando en duda si realmente SWIFT es un organismo neutral como afirma serlo.

Aunado a la polémica de si SWIFT debería limitar o censurar el envío de transacciones con ciertos países por conflictos políticos o militares entre naciones, se suman los ataques cibernéticos cometidos en los últimos años hacia la red SWIFT para robar dinero, estos ataques lejos de disminuir o estar controlados van en aumento, incluso las compañías de seguridad informática vislumbran un panorama no muy alentador debido a que los ataques cometidos por hackers son llevados a cabo con herramientas y métodos cada vez más sofisticados.

1.10 Ciberataque al Banco de Bangladés

Uno de los más recientes y grandes ataques cometidos contra SWIFT fue el que se llevó a cabo en febrero de 2016, en el cual el grupo de *crackers* que perpetuó el ataque fue capaz de llevarse la modesta cantidad de 81 millones de dólares, infectando las computadoras de los empleados del Banco de Bangladés con un *malware*[27], lo que permitió a los hackers robar las claves de acceso con las cuales el Banco de Bangladesh se comunica con la red SWIFT, una vez que obtuvieron las claves de acceso tuvieron la posibilidad de realizar solicitudes "legítimas" hacia SWIFT para solicitar una transferencia de 100 millones de dólares desde el Federal Reserve Bank of New York hacia el Banco de Bangladés y posteriormente hacia diferentes cuentas bancarias en Filipinas, Sir Lanka y otras partes de Asia. Cuatro cuentas bancarias en Filipinas, habían sido abiertas una año antes y sólo tenían un saldo de 500 dólares hasta que recibieron la inyección con millones de dólares el día del ciberataque, llegando a un total de 81 millones de dólares entre los depósitos en las cuatro cuentas bancarias,

[27] Programa informático diseñado para causar daño, por ejemplo, obtener acceso no autorizado, obtener o manipular la información de forma fraudulenta o denegar un servicio informático. El *malware* es software creado intencionalmente o no para hacer daño. El software creado de forma deficiente y que provoca agujeros de seguridad es también un *malware*. El *malware* es clasificado en diferentes categorías dependiendo de su funcionamiento y objetivo (Dulaney & Easttom, 2018).

los millones restantes que el grupo de *crackers* intentó robar pudo ser recuperado por las instituciones financieras una vez que detectaron que existían transacciones financieras sospechosas (Zetter, 2016).

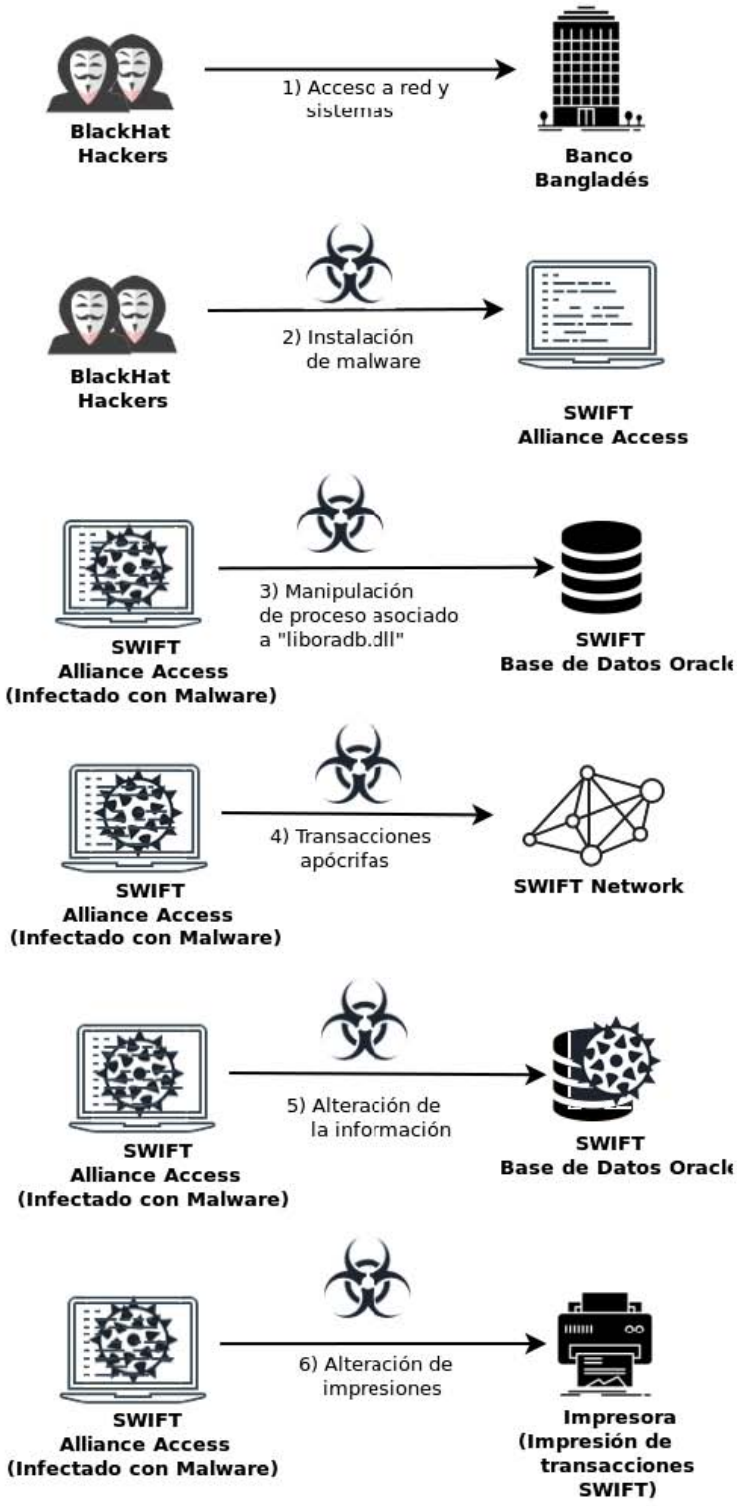

Figura 8. Ciberataque al banco de Bangladesh. Adaptado de Two Bytes To $951M: THREAT RESEARCH BLOG. 2016 por Shevchenko, S.

En la figura 8 se muestra el procedimiento llevado a cabo por el grupo de ciberdelincuentes que planeaban robar 951 millones de dólares del banco de Bangladés, pero sólo pudieron sustraer 81 millones de dólares. El ciberataque comenzó con los ciberdelincuentes obteniendo acceso a los sistemas informáticos de la institución financiera, una vez dentro de los sistemas los ciberdelincuentes infectaron el *Alliance Access*[28] de SWIFT con un *malware* que les permitió realizar transacciones financieras a través de la red SWIFT usurpando la identidad del banco de Bangladés. El *malware* también permitió a los cibercriminales alterar uno de los procesos de la base de datos Oracle, la base de datos utilizada por el *Alliance Access* de SWIFT para registrar las transacciones, con esta alteración en la base de datos fue posible ejecutar instrucciones en la base de datos para borrar las transacciones fraudulentas realizadas por el *malware*. El *malware* fue utilizado también para alterar la información que era enviada a imprimir automáticamente, ya que todas las transacciones llevadas a cabo por SWIFT son enviadas a hacia una impresora, esto podría haber alertado del ciberataque (Shevchenko, 2016).

1.11 Ciberataque al Banco del Austro

En enero de 2015 el Banco del Austro en Ecuador sufrió un ciberataque que dejo pérdidas millonarias para el banco ecuatoriano, los ciberdelincuentes lograron acceder remotamente a la institución financiera para apoderarse de las credenciales de acceso en la red SWIFT de uno de sus empleados. Los ciberdelincuentes usaron las credenciales de acceso del empleado para realizar transacciones desde el Banco del Austro solicitando que la institución financiera Wells Fargo realizará transferencias monetarias hacia cuentas bancarias en Hong Kong, estas transferencias se llevaron a cabo durante 10 días generando un total de 12 millones de dólares transferidos de Wells Fargo en San Francisco a Hong Kong mediante la usurpación de identidad del Banco del Austro por parte de los ciberdelincuentes, sin que ninguna de las dos instituciones financieras o la propia red de SWIFT se percataran de ello (Bergin & Layne, 2016).

Por el momento se desconoce la identidad de los ciberdelincuentes, sin embargo, el Banco del Austro demando a Wells Fargo por las pérdidas millonarias que sufrió argumentando que la institución financiera estadounidense debió percatarse que las transacciones eran sospechosas. En su defensa Wells Fargo alegó que la culpa fue de la institución financiera ecuatoriana ya que a través de las credenciales de uno de sus empleados fue posible que los ciberdelincuentes solicitaran las transacciones a través de la red de SWIFT (Bergin & Layne, 2016).

[28] *Alliance Access* es un componente de software que forma parte de la red SWIFT, es utilizado como una interfaz para realizar transacciones financieras sobre esta red.

Para Bergin y Layne (Bergin & Layne, 2016), el verdadero problema es la falta de transparencia por parte de las instituciones financieras debido a que ninguna de ellas comparte información sobre ataques cibernéticos, y peor aún, ni siquiera informan que han sido víctimas de uno, dejando en la oscuridad la realidad sobre la inseguridad del sistema financiero lo que proporciona una falsa sensación de seguridad en el sistema financiero internacional, especialmente en SWIFT.

1.12 Ciberataque a más de 100 instituciones financieras

Un grupo de ciberdelincuentes han robado una cifrada estimada en al menos un billón de dólares por medio de un *malware* de tipo backdoor[29] llamado Carbanak, con el cual han infectado por lo menos a 100 instituciones bancarias alrededor del mundo, este *malware* es propagado a través de archivos adjuntos en correos electrónicos enviados a empleados de las instituciones financieras, correos tanto institucionales como personales. Una vez que la computadora ha sido infectada por el *malware*, este *malware* además de funcionar como puerta trasera permite vigilar lo que hace el usuario de la computadora, extraer información y controlar el equipo de forma remota. Con el control del equipo de forma remota los ciberdelincuentes comenzaron a infectar aquellas computadoras que se encontraban en la red hasta alcanzar una computadora que fuera de su interés, por ejemplo, la computadora de un usuario con permisos de administrador (GReAT, 2015).

El malware Carbanak además de darles acceso a los equipos de forma remota les permitió a los ciberdelincuentes comenzaron a grabar videos con todas las actividades llevadas a cabo por el usuario de la computadora infectada, así como grabar todo lo que el usuario de la computadora escribía en el teclado usando un *keylogger*[30], esto les dio a los cibercriminales el conocimiento de cómo operaba la institución financiera específica que estaban atacando en ese momento para posteriormente llevar a cabo el robo realizando transacciones monetarias por medio de la red SWIFT hacia diferentes cuentas bancarias falsas creadas por los cibercriminales, alterando las bases de datos. Posteriormente los cibercriminales utilizaron personas para recolectar el dinero depositado en las cuentas bancarias falsas, adicionalmente los cibercriminales utilizaron

[29] El término *backdoor* (puerta trasera) tiene dos significados, el primero hace referencia a un acceso sin autenticación ni autorización dentro un sistema o red informático creado de forma intencional durante la fase de desarrollo con fines de pruebas y depuración del sistema. El otro tipo de backdoor se refiere a un ataque en el cual se obtiene acceso a una red o sistema informático utilizando un *malware* que proporciona la entrada al sistema o red (Dulaney & Easttom, 2018).

[30] Programa informático que registra la escritura realizada con el teclado de una computadora en un archivo de texto para posteriormente ser examinado en busca de información confidencial, por ejemplo, contraseñas. El archivo generado por el keylogger es cifrado para dificultar su acceso, el atacante que instaló el *malware* es el único que tiene acceso al archivo (Dulaney & Easttom, 2018).

malware para enviar instrucciones remotas a los ATM[31] para que "escupieran" dinero que sería recolectado por personas, estas personas son conocidas con el término mulas (GReAT, 2015).

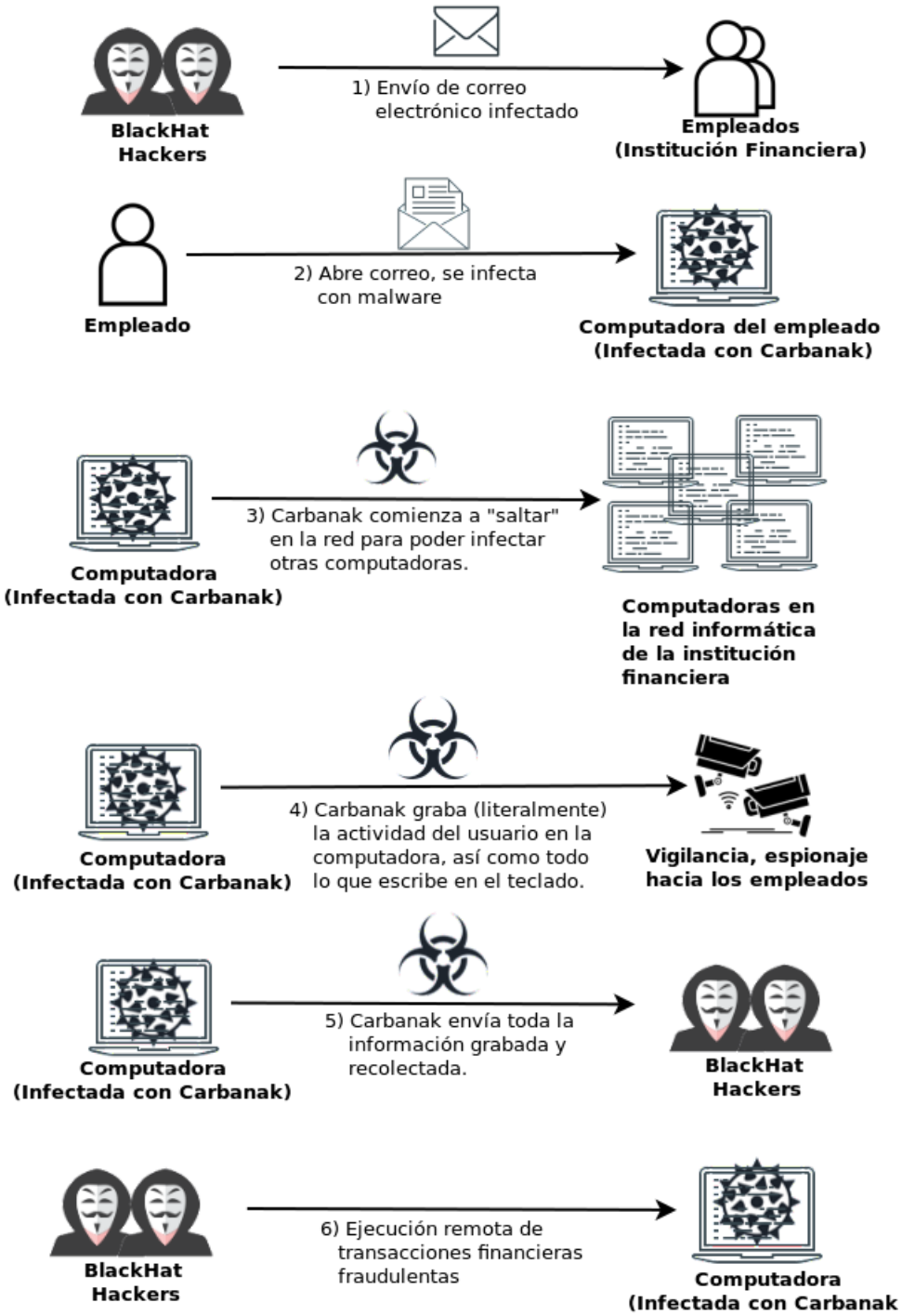

Figura 9. Ciberataque con Carbanak. Adaptado de *The Great Bank Robbery The Carbanak APT*: Kaspersky. 2015 por GReAT.

[31] *Automated Teller Machine*, traducido al español como "cajero automático", es una maquina electrónica perteneciente a una institución financiera, a través de la cual es posible disponer de dinero en efectivo utilizando una tarjeta de crédito o débito (Kagan, Automated Teller Machine (ATM), 2019).

Capítulo II. Fiscalización, Transparencia y Auditoria en Materia Informática

Si los sistemas informáticos utilizados para llevar a cabo las transacciones financieras no son suficientemente seguros como ya se expuso en el capítulo anterior, es necesario llevar a cabo un cambio en los sistemas financieros, pero este cambio no puede realizarse de forma aislada sólo por unas cuantas organizaciones, debe ser un cambio incluyente, que tome en cuenta a todos los actores en el sistema financiero. Para llevar a cabo un cambio de tales dimensiones se requiere establecer lineamientos para la creación de mejores sistemas financieros, principalmente sistemas financieros más seguros. Adicionalmente, será fundamental establecer un proceso continuo de transparencia por parte de las instituciones financieras a través del cual se obtenga información relevante para la mejora en la seguridad de las instituciones financieras.

Pero no son suficientes los lineamientos o normas, es necesario asegurarse que los lineamientos serán acatados por todos los actores involucrados, por que como reza el adagio de la seguridad informática “el sistema es tan seguro como su eslabón más débil”, tal y como sucedió en el caso de *hackeo* al sistema SPEI, no fue un ataque directamente contra SPEI, fue un ataque cibernético contra ciertas entidades financieras que se comunicaban con el sistema SPEI lo que derivó en el robo de millones de pesos a través del sistema SPEI (véase la sección 1.5).

También será necesario que las organizaciones financieras cuenten con la capacidad de proveer de información accesible, confiable y útil, que permita determinar si sus sistemas son confiables, seguros y eficientes respecto a un conjunto de normas neutrales y estándar, que, con el tiempo, llevarán a un proceso continuo de mejora en las instituciones financieras.

2.1 Fiscalización

Para Priego, Ramírez y García, la fiscalización es un proceso de inspección y evaluación de la gestión pública, para comprobar si los recursos públicos son administrados y utilizados con forme a derecho. El proceso mediante el cual la autoridad vigila el uso de los recursos públicos, ya sean humanos, financieros o materiales. Así mismo la fiscalización debe contar con mecanismos de control que le permitan llevar a cabo su cometido, entre estos mecanismos se encuentra la auditoria (Priego Hernández, Ramírez Martínez, & García Rodríguez, 2018).

Partiendo de la definición anterior se deriva la siguiente definición de fiscalización más amplia y universal ya que no es exclusiva de las instituciones o recursos de carácter público.

La fiscalización es un proceso de inspección, revisión, evaluación y seguimiento para garantizar el cumplimiento de normas y procedimientos que son establecidos para la ejecución y administración de procesos y actividades específicas. Esto incluye el uso eficiente de los recursos, independientemente de su tipo u origen, ya sean asociados o derivados de los procesos y actividades sujetos de fiscalización. En este sentido la fiscalización busca establecer el control de instituciones públicas o privadas con el fin de proporcionar certeza en el correcto funcionamiento de estas, por ello la fiscalización consiste en el control y supervisión de los procesos y actividades que las instituciones llevan a cabo. Este proceso de control aplicado de forma correcta asegura y mantiene instituciones que utilizan de la mejor manera posible sus recursos, además de proporcionar servicios o productos de alta calidad, otorgando un beneficio que se extiende también a las sociedades en las cuales se encuentran inmersas las instituciones.

Debido a que la fiscalización controla, supervisa y evalúa, esta debe permanecer y ser ejecutada de forma autónoma e imparcial, con forme a derecho y con plena autoridad para llevar a cabo acciones mediante las cuales se garantice el cumplimiento de las normas por parte de las instituciones. Manteniendo en todo momento, gran cautela en este tipo de acciones correctivas, ya que podrían conducir a nefastos problemas, por ejemplo, falta de transparencia o cohecho por parte de las instituciones con el fin de evadir el control, lo que menoscabaría e impediría el objetivo que se busca a través de la fiscalización.

Es importante señalar que debido a que la fiscalización implica control y supervisión, la fiscalización puede aplicarse a cualquier entidad que requiera ser objeto de control, supervisión y evaluación de normas y lineamientos, de lo que se deduce una aplicación de un orden más global por parte de la fiscalización, que abarca distintas áreas del conocimiento, no limitándose al área contable o administrativa.

De acuerdo a lo anterior, la fiscalización juega un papel principal en el quehacer de cualquier organización, ya que a través de la fiscalización es posible establecer un marco de referencia común para las organizaciones ya sean lucrativas o no, para garantizar que las actividades llevadas a cabo por una organización cumplen estándares mínimos de calidad, que están apegadas a la legislación vigente y cumplen con las normas establecidas, esto proporciona un margen de acción limitado exclusivamente al uso eficiente de sus recursos, así como la entrega de productos y servicios de calidad por parte de las organizaciones.

2.2 Transparencia

La transparencia corporativa es la característica de las organizaciones para proveer de información relevante y confiable acerca de acontecimientos importantes que son de interés, como el rendimiento de la empresa, su posición financiera, las oportunidades de inversión, entre otras. Así mismo la transparencia corporativa puede ser medida de acuerdo con la calidad de la información, los métodos utilizados para obtener la información, la credibilidad y la distribución o diseminación de la información (Bushman & Smith, 2003).

La transparencia es la cualidad de una entidad a través de la cual se obtiene información clara y precisa, por lo que la información está libre de ambigüedades, de esta forma la transparencia incrementa y mejora la capacidad de compresión, vigilancia y comunicación (Naessens, 2010).

De las definiciones anteriores se toman los elementos centrales para construir una definición más genérica sobre transparencia, que pueda ser aplicada en un ámbito mucho más amplio, lo que permitirá aplicar esa definición en la informática, específicamente en los sistemas informáticos financieros.

La transparencia es la característica fundamental por medio de la cual una entidad proporciona información relevante, consistente, verificable, entendible y de fácil acceso acerca de los diferentes elementos o características concernientes con la entidad. Esta información debe permitir a quién la obtenga, entender qué es la entidad, cuál su objetivo, qué procesos realiza y cómo los lleva a cabo. Además de estos elementos, la transparencia debe proveer de información sobre el estado actual y a lo largo del tiempo, de la entidad y de sus procesos, así como cualquier información relevante que mejore la compresión de la entidad y permita fiscalizarla.

Con la definición proporcionada sobre transparencia, podemos aplicarla a un amplio espectro de áreas del conocimiento. Al aplicar la definición de transparencia en las instituciones sin importar si estas son públicas o privadas, es posible decir que una institución es transparente si proporciona información relevante, consistente, verificable, entendible y de fácil acceso acerca de sus procesos, entendiendo los procesos como los procedimientos que lleva a cabo la institución para crear los bienes o servicios que proporciona a sus clientes. Esta información incluye el estado actual y a través del tiempo de los procesos, lo que permite determinar la condición o estado de la institución en cualquier punto del tiempo. Es pertinente recalcar que bajo esta definición quedan abiertas las preguntas sobre los procesos específicos que deberían transparentarse, así como el nivel de detalle de la información, ya que todo esto dependerá de los objetivos concretos que se busquen obtener con la transparencia.

En la actualidad es casi imposible encontrar un proceso organizacional que no se lleve a cabo por medio del uso de alguna tecnología informática, la cualidad casi omnipresente de los sistemas informáticos en las

organizaciones hace relevante poner especial atención sobre dichos sistemas, ya que a través de estos sistemas informáticos se llevan a cabo muchos, si no que la mayoría o todos los procesos críticos de muchas organizaciones, tales procesos pueden ir desde realizar millones de transacciones financieras alrededor del mundo hasta controlar complejos sistemas industriales o aeronáuticos.

La gran responsabilidad que se ha delegado a los sistemas informáticos, aunado a la importancia y criticidad de las tareas que desempeñan, vuelve evidente, incluso obligatoria, la asignación de la transparencia como cualidad fundamental de cualquier sistema informático de manera que sea posible establecer los elementos necesarios para que pueda ser aplicada la fiscalización sobre dicha tecnología, ya que la transparencia es el elemento que posibilita la aplicación de control y normas sobre cualquier entidad.

Sin la asignación de transparencia sobre los sistemas informáticos es imposible tratar de llevar a cabo mecanismos de control y supervisión sobre estos, de manera que no se puede garantizar el correcto funcionamiento de tales sistemas, la calidad en la ejecución de sus tareas y mucho menos su cumplimiento ante cualquier norma aplicable.

2.3 Auditoría

Es un proceso sistematizado que genera evidencia y hallazgos a través del análisis y escrutinio de la información correspondiente a las actividades y procesos de una organización, para poder generar resultados útiles que permitan la toma de decisiones, el mejoramiento de procesos y la aplicación de la normatividad vigente (Sotomayor, 2008).

De la definición anterior se deduce que la autoría es el proceso que permite determinar a través del escrutinio de la información y evidencia recolectada, de la ausencia de esta, de las inconsistencias y falta de congruencia, si el proceso o entidad que se está auditando es coherente con las normas, estándares, legislación aplicable y por supuesto con la realidad, en un periodo determinado.

Adicionalmente, el proceso de auditoría debe cumplir con las características de independencia y competencia, es decir, la auditoría debe llevarse a cabo de forma independiente por profesionales calificados, de tal forma que los resultados derivados del proceso de auditoría sean imparciales, verídicos y en apego a la normatividad vigente (Muñóz Razo, 2002).

La auditoría es de suma importancia, ya que es la materia que se encarga de garantizar la veracidad de los procesos o entidades que examina, es decir, valida la congruencia de la realidad con respecto a la evidencia recopilada así como a las normas aplicables, identificando posibles anomalías, deficiencias en los procesos o

incumplimiento de las normas, que dependiendo de los resultados, podrían llevar a la aplicación de la sanción correspondiente por no cumplir la normatividad, pero más importante aún, deben llevar a un cambio de paradigma en la ejecución y conducción de los procesos o entidades auditados, derivando en una mejora significativa de estos.

Debido a que la auditoría se encarga del escrutinio de procesos o entidades, la auditoría puede ser aplicada a una amplia variedad de áreas del conocimiento, bajo esta idea, la auditoría aplicada en la informática sigue los mismos principios de análisis y escrutinio a través de la recolección de información y evidencia, pero enfocado estos esfuerzos de manera exclusiva para determinar si las redes o sistemas informáticos cumplen con las normas, estándares y legislación aplicable. Aunado a lo anterior la auditoría en informática debe evaluar si las redes o sistemas informáticos son eficientes, seguros y confiables con respecto a la información procesada y almacenada, a los recursos utilizados y a los procesos que ejecutan.

Por lo que, la auditoría informática, no sólo permitirá determinar si una red o sistema informático está cumpliendo con la normatividad correspondiente, también será posible a través de este escrutinio, conocer cuáles son los aspectos y áreas en las cuales las redes o sistemas informáticos deben llevar a cabo una reingeniería para mejorar sus procesos y de esta manera ser más eficientes, seguros y confiables. De esta forma, la auditoría como examinadora y recopiladora de hechos y evidencias, se convierte en el vehículo principal a través del cual es posible llevar a cabo la fiscalización no sólo de las redes o sistemas informáticos, sino de cualquier entidad sujeta a ser fiscalizada.

Al ser la auditoría la forma a través de la cual es posible lleva a cabo la fiscalización, se vuelve indispensable y obligatoria la realización de auditorías sobre cualquier entidad sujeta a cumplir normas, de esta forma siempre se podrá tener la certeza de que en todo momento la entidad auditada se encuentra al margen de la normatividad aplicable.

2.4 Complementariedad Conceptual entre Fiscalización, Transparencia y Auditoria

A partir de los conceptos expuestos en la sección anterior es posible hablar de la fiscalización informática como el proceso de inspección, evaluación y control de las redes y sistemas informáticos con la finalidad de asegurar que estos cumplen con estándares de calidad, eficiencia y seguridad, pero también que cumplen con la legislación y normas aplicables.

Para poder establecer control y evaluación sobre las redes y sistemas informáticos, es decir, para poder convertir las redes y sistemas informáticos en sujetos de la fiscalización, será imprescindible que estos cuenten con características de transparencia a través de las cuales sea posible auditarlos, ya que, sin la transparencia y la auditoría, no es posible llevar a cabo el proceso de fiscalización.

A través de la fiscalización, la transparencia y la auditoria, es posible garantizar que:

1) Una red o sistema informático cumple con estándares de calidad y seguridad.
2) Una red o sistema informático cumple con la legislación o normas vigentes.
3) Una red o sistema informático puede ser sujeto de escrutinio.
4) Una red o sistema informático puede ser mejorado.
5) Una red o sistema informático puede ser verificable.
6) Una red o sistema informático es confiable.

La transparencia brinda al sistema o red informática la cualidad de poder ser fiscalizado y auditado de forma fácil y eficiente, ya que una red o sistema informático transparente ha sido diseñado y construido para proveer de forma sencilla, incluso de forma automática, información veraz, relevante, consistente y verificable concerniente al propio sistema tanto en su propia creación, en su funcionamiento interno, así como en sus entradas y salidas, lo que facilita y agiliza el proceso de inspección para cumplir con los procesos de auditoría y fiscalización, así como en el proceso de mejora continua para la actualización y creación de redes o sistemas informáticos eficientes, seguros y confiables.

Así mismo, el proceso de evaluación y escrutinio por parte de la fiscalización es aplicado a través de la auditoría, siendo la auditoría la autoridad que recopila la información necesaria y genera la evidencia a través de la cual se determinará si una red o sistema informático cumple con los lineamientos correspondientes, por lo que resulta innegable la necesidad de contar con información accesible, consistente y verificable. Precisamente este tipo de información es producto de la transparencia, de esta forma, la fiscalización, la auditoría y la transparencia, convergen en una sinergia armoniosa en la que se encuentran irremediablemente unidos.

Con base en lo anterior se desprende la importancia de los conceptos de fiscalización, auditoría y transparencia para las redes y sistemas informáticos los cuales son en la actualidad el núcleo en los procesos de diversas áreas del conocimiento, por lo que se vuelve sumamente relevante la creación de redes y sistemas informáticos transparentes, eficientes y seguros, ya que estás redes o sistemas informáticos llevan a cabo procesos críticos dentro de las organizaciones.

Capítulo III. El Bitcoin Como un Recurso Tecnológico Para la Seguridad en las Transacciones Financieras

En los últimos años, Bitcoin comenzó a obtener una gran popularidad fuera de los círculos informáticos, académicos y de entusiastas. Ahora grandes actores del sector financiero y económico observan con asombro, estupefacción, incluso recelo y miedo, no sólo a Bitcoin como sistema y como moneda digital, también a toda la ola de nuevas monedas digitales y sistemas de reciente creación basados en el funcionamiento de Bitcoin o, mejor dicho, inspirados en una interpretación difusa de lo que es Bitcoin.

Bajo este panorama de novedad, muchos entusiastas, la mayoría de las veces sin tener un entendimiento demasiado profundo del funcionamiento de Bitcoin, han hecho demasiadas suposiciones[32] acerca de cómo se puede aplicar el "funcionamiento" de Bitcoin fuera de este contexto, es decir, aplicar el funcionamiento de Bitcoin a otras áreas ajenas a la financiera, exagerando sobre manera el potencial de esta tecnología. Siguiendo este argumento, muchos de los sistemas creados recientemente, "basados" en Bitcoin, no están realmente utilizando de la misma forma la tecnología tal y como la concibió Nakamoto. Esto significa, primero, que no se puede asegurar que esos sistemas compartan las mismas características que el sistema Bitcoin proporciona, seguridad, transparencia, inmutabilidad y distribución *peer-to-peer*, que son precisamente los atributos que proporcionan a Bitcoin una gran ventaja sobre los sistemas tradicionales. Segundo, bajo este precepto se habla ya no de Bitcoin, sino de su tecnología, "*blockchain*", esto ha derivado en discusiones en las que se habla de *blockchain* como una solución utópica[33] para resolver muchos de los problemas actuales y futuros que enfrenta la humanidad, más allá de esto, "*blockchain*" se ha convertido en una mera campaña publicitaria.

De ninguna manera se niega la innovación tecnológica de Bitcoin, todo lo contrario, la creación de Nakamoto proporciona características que ningún sistema informático había demostrado tener hasta antes de su creación. También se reconoce la utilidad de la tecnología usada por el sistema Bitcoin, así como su posible aplicación en áreas distintas a la financiera, sin embargo, es necesario establecer cómo es que funciona y a partir de ese punto de referencia determinar cuáles son sus limitaciones. Esto nos permitirá conocer si realmente es factible

[32] Se han hecho demasiadas suposiciones sobre cómo funciona la "tecnología" bajo la cual está construido el sistema Bitcoin y como está puede ser utilizada, como ejemplo de estas interpretaciones confusas, véase el material audiovisual publicado por la revista WIRED en Internet (Warburg, 2017).

[33] Un claro ejemplo de la exageración que se ha hecho sobre la "tecnología *blockchain*" se encuentra en la obra, La Revolución Blockchain, en la que se hacen vaticinios sobre como esta "tecnología" resolverá los problemas de la humanidad, véase (Tapscott & Tapscott, 2018).

su aplicación fuera del sistema Bitcoin, si esto es así, entonces se necesita determinar bajo qué circunstancias o qué adecuaciones son necesarias para llevar a cabo este cometido.

3.1 Qué es *blockchain*

Blockchain es uno de los elementos que forman parte del funcionamiento en el sistema Bitcoin, es una estructura de datos que permite mantener un orden lógico sobre un conjunto de datos. Realmente, *blockchain* es más una variación o adaptación de una sencilla estructura de datos ampliamente conocida y usada dentro de la informática, las listas enlazadas. A diferencia de una lista enlazada tradicional, en el *blockchain* utilizado por Bitcoin, la conexión entre los elementos o nodos de la lista se lleva a cabo mediante cadenas hash o mensajes de digestión.

Es importante recalcar que por sí mismo, *blockchain* no proporciona todo su valor al sistema Bitcoin, ni mucho menos todas sus características y ventajas. Los beneficios proporcionados por el sistema Bitcoin están determinados precisamente por todo el sistema en su conjunto, no por un componente específico de este. Como ya se mencionó en el capítulo uno (véase la sección 1.7), el sistema Bitcoin proporciona características de seguridad sobre las transacciones financieras que ningún otro sistema había tenido, por esta razón, se postula el conjunto de tecnologías por medio de las cuales funciona el sistema Bitcoin, como una alternativa de mejora para los sistemas financieros, con la finalidad de proveer mejor seguridad y transparencia en las transacciones financieras del sistema financiero mexicano actual.

En las siguientes secciones se disecciona el funcionamiento del sistema Bitcoin con el objetivo de comprender cuales son elementos que proporcionan a este sistema sus características principales, concretamente, la seguridad en las transacciones financieras, para así poder determinar su viabilidad para poder utilizar estos elementos tecnológicos en el sistema financiero mexicano como una mejora para la seguridad de las transacciones financieras mexicanas.

3.2 Criptografía

Una de las características principales de cualquier sistema informático es la seguridad, sin embargo, la seguridad toma un papel aún más importante cuando se trata de sistemas que llevan a cabo transacciones financieras. Es necesario garantizar que las transacciones financieras cuentan con los tres componentes

básicos de seguridad en la información, estos son, confidencialidad, integridad y disponibilidad, mejor conocidos en el idioma inglés como CIA (*Confidentiality Integrity and Availability*) *triad*.

Al respecto, uno de los principales elementos que forman parte de la seguridad informática es la criptografía, una rama de las matemáticas y de la informática, cuyo principal objetivo es proporcionar confidencialidad de la información transmitida y almacenada, a través de un proceso reversible que transforme la información legible en un conjunto de datos ilegible. Es importante señalar que la criptografía no sólo proporciona confidencialidad, también provee de otras características de seguridad como integridad y no repudio.

La criptografía proporciona confidencialidad al proveer de algoritmos matemáticos que permiten ocultar la información de personas no autorizadas, principalmente un atacante o ciberdelincuente, pero posibilita tener acceso a esta información a quienes tiene derecho o están autorizados.

Específicamente, la criptografía proporciona confidencialidad a través del cifrado, esto es, la aplicación de algoritmos matemáticos que permiten ocultar la información, estos algoritmos hacen uso de un conjunto de datos generados de manera aleatoria, pertenecientes a un domino específico determinado por el algoritmo de cifrado. Este conjunto de datos se conoce como llave, la llave es usada junto con la información que se quiere ocultar aplicando una serie de operaciones definidas por el algoritmo, el resultado de este cálculo es la información cifrada, es decir, es la información que se encuentra "oculta". Un algoritmo de cifrado lo suficientemente seguro, garantiza que la única manera de obtener la información "original" a partir de la información cifrada, es aplicar el algoritmo correspondiente de descifrado utilizando la llave que uso para cifrar la información en primer lugar, el descifrado es la operación opuesta del cifrado.

3.3 Cifrado

El cifrado[34] es el proceso a través del cual se transforma un conjunto de datos conocido como texto plano, en otro conjunto de datos completamente diferente que es ininteligible y carece de significado respecto al conjunto de datos original, de forma que no sea posible deducir o encontrar una relación entre el conjunto de datos ilegible y los datos originales, este texto ininteligible se le conoce como texto cifrado.

[34] La palabra encriptar es un anglicismo que se ha derivado de la palabra *encrypt*, es usada indiscriminadamente para referirse al proceso de cifrado, de este anglicismo también se han derivado incorrectamente los verbos encriptar y desencriptar. No hay ninguna razón para utilizar un anglicismo cuando existe una palabra en el idioma español que define correctamente este proceso, esta palabra es cifrar.

El proceso de cifrado se lleva a cabo haciendo uso de un conjunto de datos adicionales, generados de forma aleatoria, conocidos como llave o llave de cifrado, la llave de cifrado no sólo permitirá aplicar el proceso de cifrado para volver ininteligible los datos que se quieren ocultar, sino también permitirá aplicar el proceso inverso, llamado descifrado, el proceso de descifrado permitirá, haciendo uso de la misma llave con la cual se cifraron los datos, obtener los datos originales.

Este proceso de ocultamiento de información permite garantizar que sólo quién posee la llave de cifrado es capaz de obtener la información original, de esta manera es posible almacenar o transmitir por medios de comunicación inseguros, cualquier información o conjunto de datos que requieran confidencialidad.

Los algoritmos de cifrado pueden ser clasificados de acuerdo con el funcionamiento general que utilizan para cifrar la información, esto es, de forma simétrica o asimétrica, de manera que los algoritmos de cifrado pueden ser clasificados en algoritmos de cifrado simétrico y algoritmos de cifrado asimétrico.

3.4 Cifrado Simétrico

El cifrado simétrico comprende aquellas funciones criptográficas que requieren una sola llave criptográfica para poder llevar a cabo el proceso de cifrado, el proceso de descifrado requiere de la misma llave para poder efectuarse.

Las funciones de cifrado simétrico generalmente tienen tiempos de ejecución cortos por lo que consumen pocos recursos de procesamiento y memoria[35]. Aunque depende del tipo de algoritmo aplicado y la cantidad de información a cifrar, por lo general se pueden aplicar sin ningún problema sobre grandes cantidades de información sin comprometer la eficiencia, esto es, el tiempo y capacidad de procesamiento, así como la cantidad de memoria requerida durante el proceso de cifrado o descifrado.

Cuando el algoritmo de cifrado sea diseñado de forma correcta, así como probado matemáticamente por distintos grupos de expertos matemáticos y criptógrafos, y que además la implementación del algoritmo en el lenguaje de programación específico sea correcta, sólo entonces, la seguridad del algoritmo de cifrado recaerá únicamente en la llave de cifrado generada.

[35] La mayoría de los procesadores modernos de arquitectura CISC están optimizados con instrucciones a nivel de hardware para realizar de manera mucho más rápida y eficiente el cifrado simétrico AES (*Advanced Encryption Standard*). Cabe señalar que, como su nombre lo indica, AES es un estándar, el algoritmo de cifrado utilizado para esta especificación es *Rijndael*.

Por esta razón, la llave de cifrado debe ser secreta y sólo debe ser proporcionada a quienes estén autorizados para conocer la información que se ha cifrado. Adicionalmente, la llave de cifrado debe contar con una longitud de bits apropiada de acuerdo con el algoritmo utilizado, la llave también debe ser generada de forma aleatoria, esto es, utilizando alguna función que genere un conjunto de datos pseudo aleatorios criptográficamente seguros.

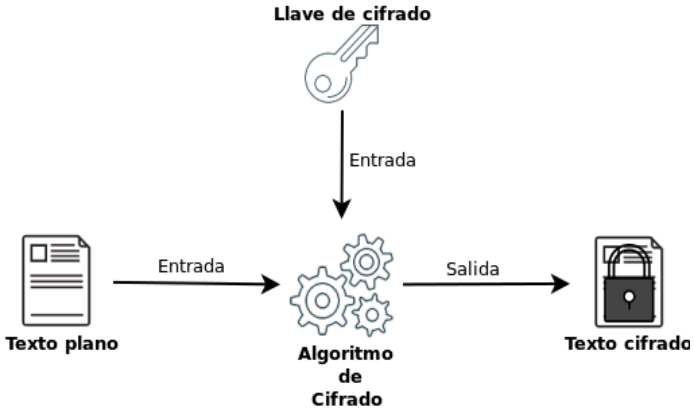

Figura 10. Funcionamiento del cifrado simétrico.

Figura 11. Funcionamiento del descifrado simétrico.

En las figuras 10 y 11 se ejemplifica de forma abstracta el funcionamiento de los algoritmos de cifrado simétrico, como se observa en ambas imágenes, para llevar a cabo tanto el cifrado como el descifrado se hace uso de la misma llave criptográfica. Esto tiene un inconveniente, si la información cifrada se necesita transmitir a través de una red informática, es necesario compartir la llave con aquellos que van a recibir la información de tal forma que puedan descifrar la información, el problema de esto es que ahora el dilema es como se transmite la llave

por la red informática evitando que esta sea interceptada y robada durante la transmisión por la red. Estos inconvenientes sobre transmisión de llaves a través de redes informáticas inseguras se resuelven por medio del cifrado asimétrico.

A pesar de los inconvenientes que tiene la distribución de llaves para cifrado simétrico, las ventajas del cifrado simétrico son mayores y hacen que sea ampliamente usado. La principal ventaja que tiene es la velocidad de procesamiento tanto para el cifrado como el descifrado. La otra ventaja que tienen los algoritmos de cifrado simétrico es la capacidad de poder cifrar grandes cantidades de información sin ningún problema, de hecho, tanto el cifrado simétrico como el asimétrico se usan a menudo de forma conjunta para compensar las desventajas de cada uno.

3.5 Cifrado Asimétrico

El cifrado asimétrico es aquel que requiere de dos llaves de cifrado para llevar a cabo el proceso criptográfico, estas llaves se encuentran ligadas por alguna función matemática, de manera que existe una relación de simetría entre las llaves, cada una de las llaves cumple un propósito especifico, dependiendo de la operación que se quiera llevar a cabo. Cuando se realiza el proceso de cifrado con una de las llaves, el proceso de descifrado se llevará a cabo con la llave opuesta.

Este tipo de cifrado también tiene el nombre de cifrado de llave pública, ya que una de las llaves corresponde a la llave pública, mientras que la otra es la llave privada, tal y como denota su nombre, la llave pública puede ser del conocimiento público, es decir, no tiene por qué ser secreta y de hecho es necesario compartirla cuando se lleva a cabo un esquema de cifrado asimétrico, por otra parte, la llave privada es como su nombre lo indica, privada, y no debe ser compartida, ya que la seguridad en este tipo de cifrado recae en mantener la secrecía de la llave privada, por supuesto, al igual que con el cifrado simétrico, la seguridad del cifrado también dependerá del tipo de algoritmo utilizado, la implementación del algoritmo y la correcta generación de llaves, esto incluye la aleatoriedad y la longitud de la llave criptográfica.

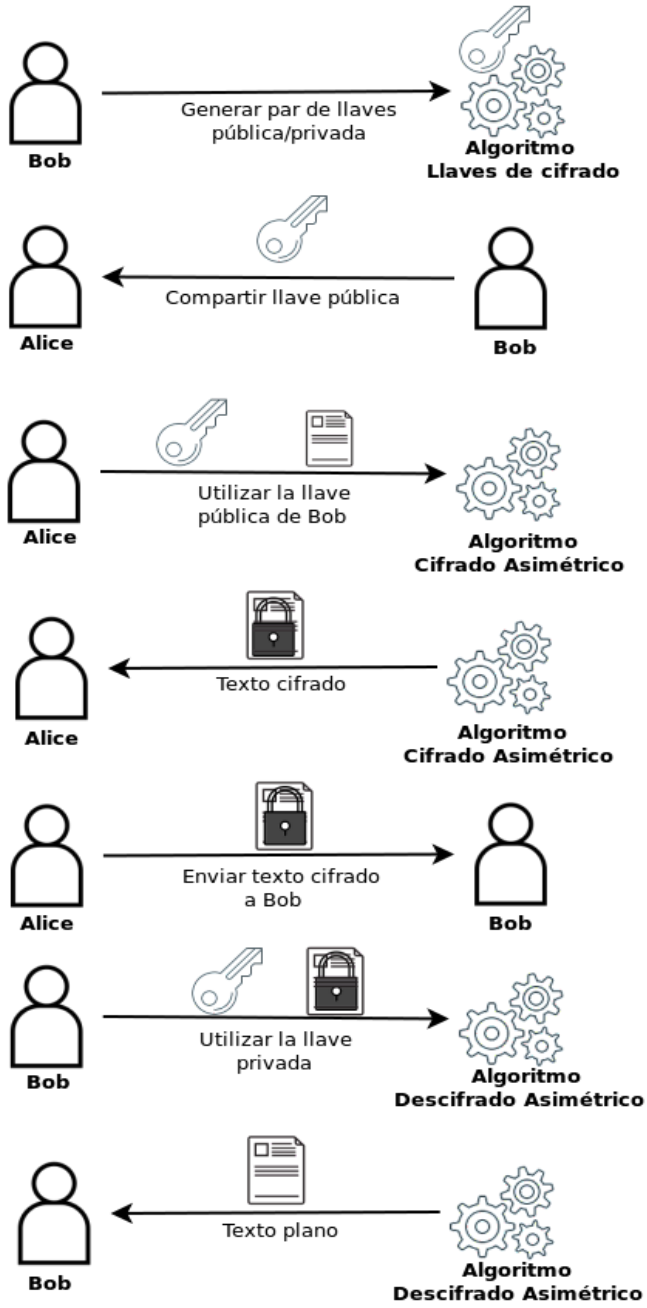

Figura 12. Funcionamiento del cifrado asimétrico.

En la figura 12 se representa un escenario típico de cifrado asimétrico, en el cual dos partes quieren compartir información, pero quieren mantener confidencialidad sobre la información, en este caso se trata de dos personajes ficticios, Alice y Bob[36]. Alice quiere intercambiar información con Bob, para llevar a cabo la transmisión de información a través de un medio de transmisión inseguro, por ejemplo, una red informática, Bob

[36] En criptografía es una convención utilizar personajes ficticios para representar su uso, específicamente se utilizan dos personajes, Alice y Bob, aunque existen más personajes que son usados dependiendo del contexto que se está describiendo.

crea un par de llaves criptográficas asimétricas, es decir, una llave pública y una llave privada, ambas llaves se encuentran ligadas matemáticamente. El algoritmo encargado de crear las llaves criptográficas crea primero la llave privada, posteriormente a partir de la llave privada se deriva una llave pública, la llave pública se puede obtener a partir de la llave privada pero no al contrario, en este sentido las llaves funcionan, al igual que todo el esquema de criptografía asimétrica, utilizando *trapdoors*, esto es, una función cuyo resultado puede ser calculado fácilmente, pero calcular los argumentos de la función a partir del resultado es casi imposible o simplemente tomaría una cantidad de tiempo tan grande realizar el cálculo se vuelve impráctico.

Una vez que Bob tiene su par de llaves, debe compartir únicamente su llave pública con la finalidad de que Alice o cualquier otra parte se comunique con Bob de manera confidencial. La distribución de la llave pública puede llevarse a cabo por un medio inseguro sin importar si esta es interceptada por algún atacante, la razón de esto se debe al funcionamiento matemático de la criptografía asimétrica, coma ya se explicó, obtener la llave privada a partir de la llave pública es prácticamente una tarea imposible, además, toda la información que sea cifrada con llave pública sólo puede ser descifrada con la llave opuesta, es decir, la llave privada y debido a que Bob resguarda la llave privada y no la comparte con nadie más, es el único que puede descifrar y obtener la información cifrada con la llave pública que el algoritmo para crear llaves derivo a partir de su llave privada.

Alice recibe la llave pública de Bob, con esta llave Alice puede cifrar la información que quiere compartir con Bob de manera confidencial, una restricción es que por supuesto tanto Alice como Bob deben acordar el mismo par de algoritmos de cifrado/descifrado. Después de haber cifrado la información con la llave pública de Bob, Alice envía la información cifrada a Bob por medio de un canal inseguro, una red informática. Bob recibe la información de parte de Alice, debido a que la información se encuentra cifrada, Bob debe descifrarla para poder conocer cuál es la información que Alice le envió, para hacer esto, Bob utiliza su llave privada. Con la llave privada Bob puede descifrar la información cifrada por Alice y puede tener acceso a dicha información.

En el escenario anterior entre Alice y Bob, Alice puede comunicarse con Bob de manera confidencial ya que el único que puede descifrar la información es Bob con su llave privada, de la misma manera, Alice puede generar su propio par de llaves pública y privada, compartir su llave pública con Bob y ahora ambos pueden intercambiar información de forma confidencial sin importar si algún atacante intercepta sus mensajes ya que al carecer de las llaves privadas, el atacante no será capaz de descifrar la información, incluso si obtiene las llaves públicas no podrá descifrar los mensajes ni mucho menos calcular la llave privada a partir de la llave pública, esto asegura la confidencialidad entre Alice y Bob.

3.6 Firma Digital

Las firmas digitales son un método a través del cual es posible verificar la autoría de la información que ha sido firmada, con lo cual se puede ofrecer la garantía de que la información firmada sólo pudo ser generar por la entidad a la cual pertenece la firma, esto proporciona autenticación por parte de quién firma la información, es decir, por medio de la firma digital se puede tener la certeza de que la información pertenece al emisor de la firma, con esta característica, la entidad que firma no puede rechazar la autoría de la información, esto se conoce técnicamente como no repudio. La firma digital no sólo proporciona autenticación y no repudio, también proporciona integridad[37], esto último se logra a través de funciones Hash que calculan un valor único de digestión por cada entrada única de información. Sin embargo, las firmas digitales no proveen de confidencialidad, de tal forma que la información firmada digitalmente puede ser accedida por cualquier entidad no autorizada.

El funcionamiento de las firmas digitales está íntimamente relacionado con el cifrado asimétrico debido a que a través del cifrado asimétrico se implementan las firmas digitales, a diferencia del cifrado, en el cual se utiliza la llave pública para cifrar la información y se descifra con la llave privada, en las firmas digitales se utiliza el procedimiento opuesto, se utiliza la llave privada para aplicar una función criptográfica de cifrado y se utiliza la llave pública para descifrar, en términos simples, la firma digital consiste en cifrar y la información con la llave privada, específicamente, cifrar con la llave privada el hash resultante de aplicar una función hash sobre la información que se firma.

La razón por la cual utilizar de forma "inversa" el cifrado asimétrico proporciona autenticación y no repudio sobre la información, se debe a qué la llave privada sólo es conocida por el dueño de dicha llave y nunca es compartida, por lo que, si la información es cifrada con llave privada en lugar de llave pública, la única manera de descifrar dicha información es con la llave opuesta, que es la llave pública y debido a que la llave pública se comparte con aquellas partes con las cuales se quiere establecer un canal de comunicación cifrado, es decir, un canal de comunicación confidencial, todos aquellos que conozcan la llave pública podrán descifrar aquello que se cifró con la correspondiente llave privada y con esto validar que la información fue generada por el dueño de la llave privada, lo que garantiza que ninguna otra entidad pudo cifrar con la llave privada, es decir, firmar la información, ya que la llave privada sólo es conocida por la entidad a la cual pertenece la correspondiente llave pública. Esto proporciona un mecanismo de auditoría, es decir, por medio de la firma digital es posible auditar la información

[37] En realidad, la firma digital a través de cifrado asimétrico proporciona integridad por sí misma, sin la necesidad de utilizar funciones hash, el problema con esto es la ineficiencia por parte del cifrado asimétrico. El cifrado asimétrico es costoso en términos de tiempo de procesamiento y memoria, aunado a esto, no siempre es posible cifrar grandes cantidades de información cuando se utilizan algoritmos de cifrado asimétrico, estos inconvenientes requieren el uso de un mecanismo adicional para proporcionar integridad, por esta razón se utilizan funciones hash en la firma digital.

y comprobar a través de un procedimiento determinístico la autenticidad de la información y su integridad de forma sencilla y estándar.

Para clarificar el funcionamiento de la firma digital utilizaremos un ejemplo usando nuevamente a nuestros personajes ficticios Alice y Bob, pero además ahora se incluye a un tercer personaje Eve, a diferencia de Alice y Bob que son amigos, Eve es un atacante, un *hacker black hat*, por lo que Alice y Bob deberán utilizar la seguridad informática para proteger su comunicación.

Para este ejemplo partiremos del hecho de que tanto Alice como Bob previamente generaron cada uno su par de llaves pública y privada, ambos compartieron sus respectivas llaves públicas entre ellos, sin embargo, durante la transmisión de llaves públicas sobre una red informática, Eve intercepto la comunicación y se apoderó de ambas llaves públicas, la de Alice y la de Bob. A pesar de que Eve no es capaz de calcular la llave privada a partir de la llave pública y por lo tanto no es capaz descifrar los mensajes, Eve puede llevar a cabo un ataque que consiste en manipular la información, esto es, alterar la integridad de la información intercambiada entre Alice y Bob.

Para llevar a cabo el ataque de integridad sobre la información, Eve utiliza las llaves públicas que previamente interceptó, con estas llaves puede usurpar la comunicación creando mensajes cifrados con estas llaves públicas y enviarlos a Alice o Bob. Peor aún, Eve podría interceptar cada mensaje enviado entre Alice y Bob, y reemplazarlo por un mensaje de su autoría cifrado con la llave pública de Alice o Bob según corresponda. En este escenario la confidencialidad entre Alice y Bob no ha sido alterada, bajo estas circunstancia Alice y Bob son los únicos que pueden descifrar los mensajes cifrados con sus correspondientes llaves públicas, sin embargo, ninguno de los dos tiene la certeza de quién está enviando los mensajes, incluso sin el atacante Eve, Alice o Bob podrían enviar mensajes entre ellos y posteriormente negar la autoría de tales mensajes y debido a que la llave pública, como su nombre lo indica, es pública, esta pudo ser utilizada por alguien más, por lo que no se puede comprobar quién está enviando los mensajes.

Para garantizar la integridad de la información, así como el no repudio de la misma, se necesita de un mecanismo que permite comprobar ambos elementos, este mecanismo es la firma digital, con la firma digital podemos obtener integridad, autenticación y no repudio de la información.

Para firmar la información de los mensajes intercambiados entre Alice y Bob, ambos tendrán que firmar digitalmente sus mensajes antes de cifrarlos, una vez calculada la firma digital, podrán proceder a cifrar la información y posteriormente adjuntar la firma junto con la información cifrada. Cuando Alice o Bob reciban un mensaje antes de descifrarlo deberán validar la firma digital, por ejemplo, si Alice recibe un mensaje

presuntamente de la autoría de Bob, primero deberá cerciorarse de que el mensaje contiene una firma digital, si el mensaje recibido por Alice no está firmado digitalmente no hay garantía de conocer quién envío el mensaje, por lo cual podría descartar el mensaje. Por otra parte, si el mensaje está firmado digitalmente, Alice tendrá que validar que la firma digital corresponda, en este caso, con la firma de Bob, para llevar a cabo esta validación, Alice utilizará la llave pública de Bob para proceder a descifrar la firma digital del mensaje, si la llave pública de Bob descifra correctamente la firma digital, significa que la firma efectivamente corresponde a Bob, esto comprueba que Bob envió el mensaje, sin embargo, debido a que la firma digital y la información cifrada *per se* son conjuntos ordenados de bytes diferentes, no se tiene la certeza acerca de la integridad de la información, en este punto, se comprobó que la firma digital corresponde a Bob, es decir, se autentico que el mensaje fue enviado por Bob y también se estableció el no repudio por parte de Bob, pero no aún no se comprueba que el mensaje enviado por Bob permanezca íntegro, esto es, que no haya sido alterado.

Para comprobar la integridad de la información, las firmas digitales utilizan funciones Hash, a través del uso de estas funciones sobre la información que se quiere proteger, es posible establecer un mecanismo por el cual se puede comprobar la integridad de la información, es decir, que la información no ha sido alterada desde su creación hasta la entrega en su punto de destino.

3.7 Funciones Hash

Las funciones hash son algoritmos matemáticos que reciben una cantidad arbitraria y finita de datos, el algoritmo procesa los datos aplicando una serie de funciones matemáticas que arrogan un único resultado de una longitud fija. La longitud del resultado final al aplicar una función hash depende siempre del tipo de función hash utilizada, el resultado de aplicar una función hash sobre un conjunto arbitrario de datos siempre genera un conjunto de datos de la misma longitud determinada por el algoritmo. Utilizar el mismo conjunto ordenado de datos sobre una función hash siempre resulta en el mismo conjunto ordenado de datos de salida por parte de la función. Es decir, si se proporciona un conjunto ordenado de datos A para una función hash h, la función h regresará siempre el mismo conjunto ordenado de datos B cuando la entrada sea siempre el conjunto ordenado A. De la misma forma, un conjunto ordenado de datos diferente, proporcionado como entrada en la función hash, devolverá siempre un conjunto ordenado de datos distinto

.

En base a la descripción anterior podemos definir lo siguiente:

$$Sea\ A = \{(a, b, c, d, e)\}$$
$$Sea\ h\ una\ función\ hash\ h(x)$$
$$Si\ h(A) = B, entonces\ B = \{(w, v, \dots)\}$$
$$Donde\ N(B)\ es\ siempre\ constante$$
$$A \neq B, h(A) \neq h(B)$$

La característica descrita de las funciones hash permite que estas sean usadas para comprobar la integridad de la información, ya que cualquier cambio en la información, por mínimo que sea, dará como resultado una salida de datos diferente por parte de la función hash, en cambio, si los datos permanecen inalterables, el resultado de la función hash será siempre el mismo, lo que permite comprobar la integridad de la información.

Las funciones hash al igual que el cifrado asimétrico funcionan como una *trapdoor*, debido a que es fácil calcular el resultado, esto es, es fácil llevar cabo el cálculo en una sola dirección, sin embargo, es imposible calcular en la dirección opuesta, es decir, una vez obtenido el resultado de una función hash, no es posible determinar cuál es el conjunto ordenados de datos utilizado como entrada en la función hash.

El cálculo de funciones hash es relativamente eficiente, no requiere demasiado poder de procesamiento, por lo que es bastante rápido su cálculo en cualquier procesador promedio actual, adicionalmente existen una gran variedad de implementaciones gratuitas y de código abierto en diversos lenguajes de programación, lo que convierte a las funciones hash en un mecanismo accesible y sencillo de utilizar.

Aunque existe una gran variedad de funciones hash a continuación se listan algunas de las más comunes:

a) MD5
b) SHA1
c) SHA [256 | 384 | 512]
d) RIPEMD [128 | 160 | 256 | 320]
e) Whirlpol
f) Tiger [128 | 160 | 192]

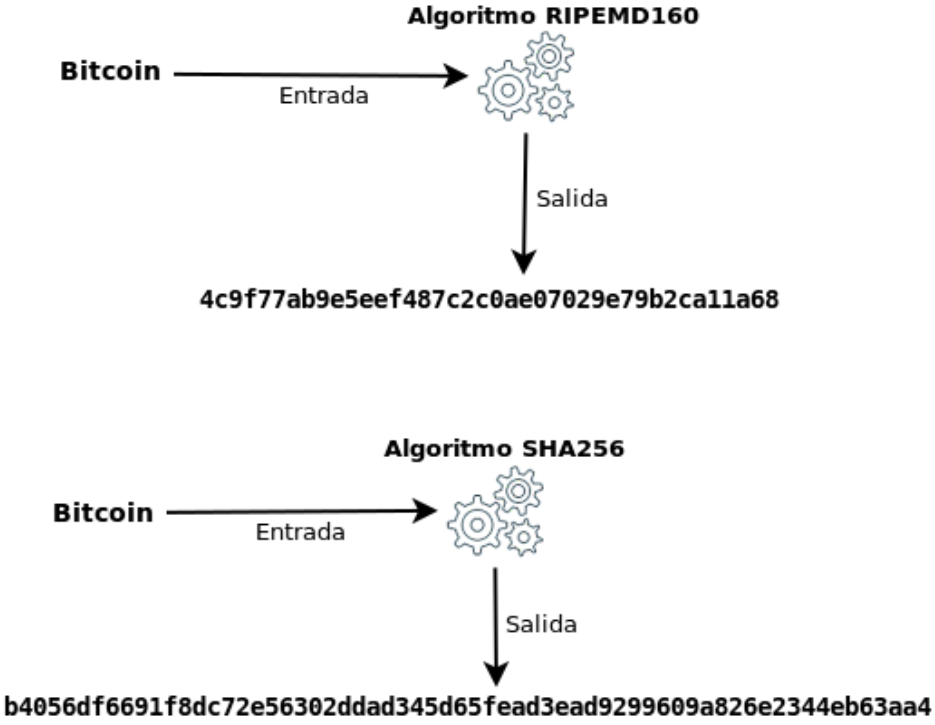

Figura 13. Funcionamiento de algoritmo hash.

En la figura 13 se ejemplifica de forma abstracta el funcionamiento de una función hash, específicamente de las funciones hash SHA256[38] y RIPEMD160, en ambos casos se pasa como argumento de la función, la cadena de caracteres "Bitcoin" como conjunto ordenado de datos, la salida en ambos algoritmos es un conjunto ordenado de datos de longitud constante. Para el caso del algoritmo SHA256, como su nombre lo indica, la longitud de salida es siempre de 256 bits, mientras que para el algoritmo RIPEMD160, la salida siempre tiene una longitud de 160 bits.

Es de suma importancia mencionar que la salida o resultado de las funciones hash son un conjunto ordenado de bits que identifican de manera univoca al conjunto ordenado de datos utilizados como entrada o argumento, sin embargo, los bits de salida son bits "crudos" sin ningún formato, para poder ser representados como caracteres legibles es necesario aplicar una codificación. Debido a que el resultado de una función hash puede ser de varios cientos de bits, es conveniente utilizar una codificación que proporcione un cierto grado de simplificación o compresión, por esta razón generalmente se representa el resultado de una función hash utilizando una codificación hexadecimal. Utilizando la codificación hexadecimal se puede representar hasta 4 bits por cada digito hexadecimal, esto simplifica el número de caracteres utilizados para representar la salida de

[38] SHA256 es una función hash de la familia SHA2 (*Secure Hash Algorithm* 2) desarrollada por la agencia de inteligencia y seguridad estadounidense NSA (Ferguson, Schneier, & Kohno, 2010).

la función hash, en el caso de la función hash SHA256, al codificar el resultado con el sistema hexadecimal se obtiene un resultado hexadecimal de 64 caracteres, mientras que en el algoritmo RIPEMD160 la salida se simplifica hasta 40 caracteres hexadecimales.

Nótese que proporcionar siempre la misma cadena de caracteres “Bitcoin” sobre el mismo algoritmo hash, regresará siempre el mismo resultado, es decir, aplicar el algoritmo SHA256 sobre la cadena de caracteres “Bitcoin” siempre dará el mismo resultado, de la misma forma realizar cualquier cambio sobre la entrada del algoritmo, cambiará siempre la salida, por ejemplo, utilizar la cadena “bitcoin” o “Bitcoin “, producirá un mensaje de digestión distinto, esto es, un resultado diferente. Este comportamiento es justamente lo que se requiere para poder comprobar la integridad de la información, debido a esto, las funciones hash son usadas junto con las firmas digitales para proporcionar integridad sobre la información que se firma digitalmente.

El resultado de aplicar una función hash es determinístico, es decir, se puede predecir el resultado siempre que se proporcione la misma entrada de datos, adicionalmente se puede decir que generalmente las funciones hash en lo posible producen resultados uniformemente distribuidos. A pesar de que no es posible deducir la entrada de datos a partir del resultado de la función hash y que la salida siempre es la misma cuando se proporciona la misma entrada de datos, la funciones hash tratan de proveer de resultados aleatorios como parte de la seguridad en el diseño de estas funciones, por lo que se pude ver el resultado de un algoritmo hash como una distribución uniforme de variables discretas aleatorias.

3.8 Redes *Peer to Peer*

Las redes *peer-to-peer* son un tipo de arquitectura de red en cual los nodos[39] de la red se comportan al mismo tiempo como clientes y servidores, en contraposición con un modelo centralizado cliente/servidor donde los nodos que actúan como clientes sólo consumen los recursos proporcionados por un nodo central que actúa como proveedor de recursos, es decir, actúa como servidor en la red (Park, Ratzin, & van der Schaar).

Un ejemplo de arquitectura de red centralizado es el utilizado en las aplicaciones Web, en estas aplicaciones existe un nodo que actúa como servidor, generalmente es una computadora con grandes capacidades de cómputo, cuenta con varios procesadores, así como una gran cantidad de memoria y unidad de almacenamiento. En esta arquitectura de red, los clientes únicamente realizan solicitudes hacia el servidor para

[39] Un nodo es un elemento conectado en una red informática, el nodo es una computadora o dispositivo capaz de enviar o recibir información, y dependiendo de la red, el nodo podrá ser capaz de procesar y almacenar información.

pedir un recurso, el servidor se encarga de llevar a cabo todo el procesamiento, en pocas palabras, se trata de un modelo centralizado.

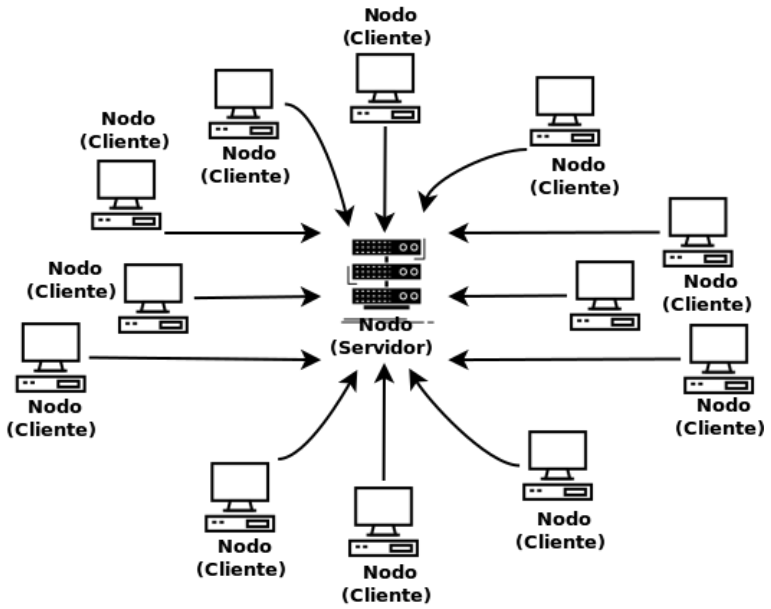

Figura 14. Arquitectura de red centralizada.

En las redes *peer-to-peer*, la arquitectura de red funciona diferente, no existe un nodo central encargado de proveer de recursos, en esta arquitectura todos los nodos actúan como servidores y clientes, de manera que no es necesario que cuenten con gran capacidad de computo, todos los nodos deben contar con el software necesario para llevar a cabo ambas funcionalidades, cliente y servidor, esta funcionalidad dual se conoce con el término *servent,* este término se deriva precisamente de las palabras en inglés, *server* (*serv*) y *client* (*ent*) (Schollmeier, 2001).

Aunque el concepto de redes *peer-to-peer* implica que los nodos son clientes y servidores al mismo tiempo, existen diferentes enfoques para crear redes *peer-to-peer*, por lo que algunas arquitecturas *peer-to-peer* utilizan un nodo central que provee una funcionalidad parcial o inicial cuando un nodo se conecta a la red por primera vez o realiza una funcionalidad específica, aun así, los demás nodos de la red siguen comportándose con la doble funcionalidad, cliente/servidor (Johnsen, Karlsen, & Birkeland, 2005).

Las redes *peer-to-peer* son redes informáticas en las cuales los nodos de la red se encuentran distribuidos de forma estructurada o no estructurada (Park, Ratzin, & van der Schaar), estos nodos se conectan entre si directa o indirectamente, la mayoría de las veces sin la necesidad de un nodo central, de manera que se crea una red distribuida entre todos los nodos existentes. En una red totalmente *peer-to-peer*, cada nodo está conectado con todos y cada uno de los demás nodos de la red (Johnsen, Karlsen, & Birkeland, 2005). Los nodos en una red

peer-to-peer, comparten sus propios recursos computacionales con todos los demás nodos de la red (Schollmeier, 2001). Utilizando esta arquitectura se puede construir un sistema informático distribuido que reside en todos los nodos de la red, de esta forma no se requiere que cada nodo sea una computadora con capacidades de cómputo tan poderosas en comparación con un nodo central de procesamiento.

Bajo este modelo distribuido se obtiene ventajas como la tolerancia a fallas, debido a que todos los nodos actúan bajo una funcionalidad doble, no importa que alguno de los nodos deje de funcionar o se desconecte de la red, ya que todos los demás nodos en la red realizan la misma funcionalidad el sistema sigue funcionando normalmente, a diferencia de una arquitectura centralizada en la cual la falla del nodo central deja el sistema inservible, ya que todos los demás nodos dependen del nodo central.

Este tipo de redes distribuidas también proporcionan una gran escalabilidad, esto quiere decir que el sistema en su conjunto, como un todo, ejecutado por cada nodo de la red, puede aumentar fácilmente su capacidad de procesamiento, así como su alcance, ya que lo único que se requiere es agregar más nodos en la red y debido a que estos nodos generalmente no están compuestos por hardware tan potente, se vuelve relativamente sencillo y económico escalar el sistema. En cambio, en un sistema centralizado, escalar requiere más nodos centrales con capacidades similares de cómputo, así como mecanismos adicionales para sincronizar el procesamiento con los demás nodos centrales, esto aumenta los costos no sólo por la capacidad de cómputo requerida por los nodos centrales, también lo hace por las adecuaciones requeridas para la distribución de los nodos centrales adicionales añadidos en una arquitectura de red que no está diseñada para ser distribuida[40].

Otra ventaja que se puede obtener de las redes distribuidas es un mejor rendimiento en el procesamiento por parte del sistema, debido a que la capacidad de procesamiento está divida en todos los nodos de la red, se obtiene mayor eficiencia en el sistema en oposición a una arquitectura centralizada donde todo el procesamiento es llevado a cabo por uno o pocos nodos centrales que tiene que soportar y procesar la carga de las solicitudes de todos los clientes de la red. Esto también satura la red ya que todo el tráfico se dirige hacia uno o pocos nodos, mientras que en una arquitectura *peer-to-peer* el tráfico en la red se distribuye de manera mucho más uniforme reduciendo cuellos de botella en la red. Esto también puede derivar en mejores tiempos de respuesta, ya que la respuesta de una solicitud de procesamiento en una red distribuida se realiza por el nodo más cercano de la red, mientras que en una arquitectura centralizada el nodo central es el único que puede procesar la solicitud y este nunca es el nodo más cercano para la mayoría de los nodos.

[40] Por supuesto, existen redes híbridas cuyo diseño específico busca aprovechar las ventajas de las redes distribuidas y las que no lo son. No todo es blanco o negro, la mayoría de las redes no son totalmente distribuidas o centralizadas, son una combinación de una u otra.

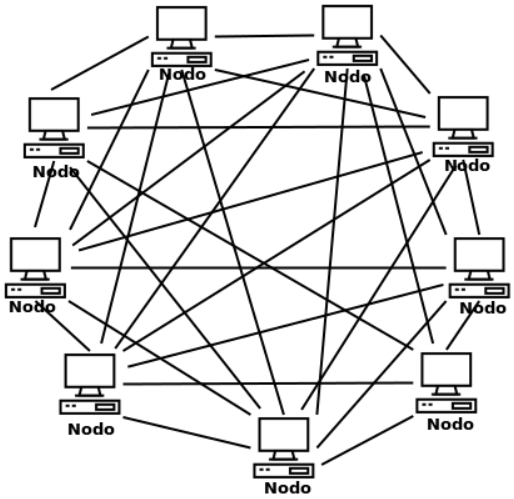

Figura 15. Arquitectura de red distribuida.

En las secciones anteriores se ha descrito de forma general el funcionamiento de los componentes que forman la base del sistema Bitcoin, las firmas digitales, las funciones hash y las redes distribuidas peer-to-peer. Con la visión general obtenida sobre estos componentes, ahora procederemos a explicar el funcionamiento de Bitcoin, explicando los elementos centrales que conforman este sistema, las transacciones, los bloques, así como su "encadenamiento" y finalmente la *proof-of-work*.

3.9 Transacciones (*Transactions*)

A diferencia de lo que se podría pensar, en el sistema Bitcoin no existen de forma explícita las monedas virtuales o digitales, en este caso, los bitcoins. Dentro de la estructura de datos utilizada por el sistema Bitcoin no hay ningún dato o estructura específica llamada bitcoin, en su lugar, Nakamoto creo una estructura abstracta que denomina *transaction* (Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008). Estas transacciones son una estructura de datos que representa el flujo de entradas y salidas de la cantidad de valor transferido entre los distintos usuarios de la red Bitcoin. En esta estructura llamada transacción, se definen *inputs* (entradas) y *outputs* (salidas), las entradas corresponden a salidas de una transacción anterior, es decir, en una determinada transacción, sus entradas son las salidas de una transacción previa, mientras que las salidas de la transacción se convierten en entradas para una nueva transacción[41].

[41] Bajo este funcionamiento, donde las salidas son entradas y viceversa, se genera un problema con respecto a la primera transacción en el sistema, ya que al ser la primera no hay una transacción previa con una salida que funcione como entrada, esto genera un dilema de tipo, ¿qué es primero, el huevo o la gallina? Para solventar esto, la primera transacción

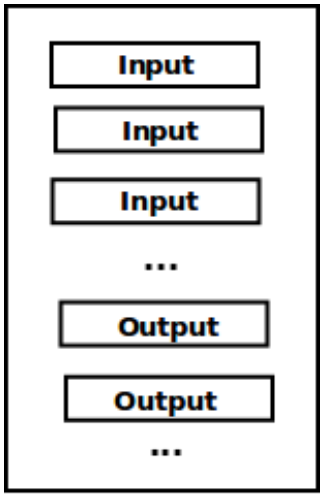

Figura 16. Representación abstracta de una transacción en Bitcoin.

Al igual que con los bitcoins, en el sistema Bitcoin no se registra de forma explícita ningún balance sobre los cargos y los abonos, tampoco se encuentra definido el saldo total de un usuario que realiza transacciones en el sistema. Todos los datos anteriormente mencionados, son una reconstrucción de información que las aplicaciones de uso general de Bitcoin llevan a cabo para los usuarios finales. Un ejemplo de estas aplicaciones son las carteras digitales (*digital wallets*), estas aplicaciones se crean para la comodidad, conveniencia y facilidad de los usuarios en el sistema Bitcoin (Antonopoulos, 2017). Por medio de estas aplicaciones de cartera digital los usuarios pueden llevar a cabo transacciones en el sistema Bitcoin, además de contar con una interfaz que proporciona información ordenada y fácil de comprender sobre todas las transacciones realizadas.

Los únicos datos registrados dentro una transacción son, el conjunto entradas, el conjunto de salidas, y algunos metadatos asociadas a esa transacción específica, como el número total de entradas y salidas, la versión utilizada para generar la transacción, un atributo llamado *locktime*, y unos cuantos atributos más (Antonopoulos, 2017).

en Bitcoin fue creada por Nakamoto sin ninguna transacción previa, esta transacción, concretamente el bloque (véase la sección 3.13) de esta transacción se conoce como el bloque génesis, ya que es el primero en toda la historia de transacciones en Bitcoin.

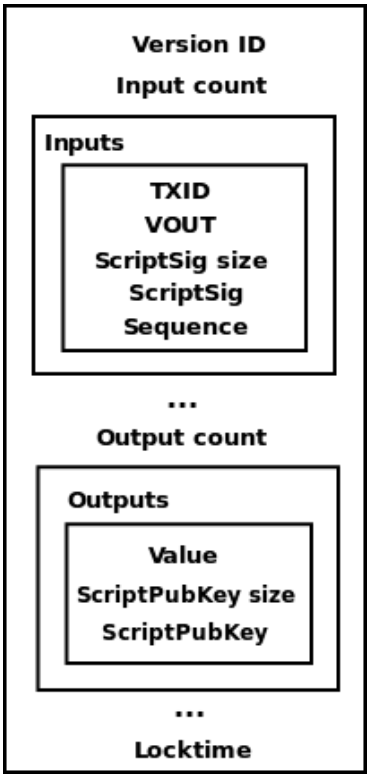

Figura 17. Anatomía de una transacción en Bitcoin.

En la figura 17 se muestran los elementos que componen una transacción en Bitcoin, primero se encuentra el atributo *version*, que representa la versión de la transacción, esta versión de la transacción corresponde con la estructura de datos que componen la transacción, esto indica a los nodos de la red Bitcoin como deben interpretar la transacción, actualmente la versión es la uno. El *input count* se refiere al número total de entradas en la transacción, mientras que el elemento *Inputs* se corresponde con las propias entradas, que a su vez se componen de los elementos TXID, VOUT, scriptSig y sequence (Walker, Transaction Data, 2015) (Antonopoulos, 2017).

Dentro del atributo *inputs*, el TXID corresponde al identificador de una transacción en el pasado que será utilizada como entrada en la transacción actual, es decir, corresponde al hash que identifica una transacción realizada previamente[42], esto significa que esa transacción fue en algún momento una salida y ahora se utilizará como entrada. El elemento VOUT corresponde con el número específico de salida que será utilizado, debido a que una transacción puede tener varias salidas, es necesario especificar no sólo el identificador de la transacción

[42] Para utilizar una transacción como entrada esta debe haber sido confirmada previamente como una transacción legitima por el sistema Bitcoin. Adicionalmente el usuario que pretenda usar esa entrada debe tener las llaves criptográficas que lo acrediten como dueño de la salida en la transacción.

que será utilizado como entrada sino también el número de la salida. El s*criptSig size* se refiere al tamaño en bytes del atributo *scriptSig*, este corresponde a un conjunto ordenado de datos que satisfacen una serie de condiciones previamente definidas en el atributo scriptPubKey de la salida que ahora se utiliza como entrada. Especificar los datos que satisfacen el scriptPubKey permite que la transacción referenciada como entrada a través de los atributos TXID y VOUT, pueda ser utilizada o, mejor dicho, pueda ser gastada. Finalmente, el atributo *sequence* se refiere a un elemento que permite deshabilitar el atributo *locktime*, así como habilitar el uso de una funcionalidad a través de la cual se puede establecer el momento a partir del cual se puede utilizar una entrada de acuerdo con un bloque o tiempo Unix específicos.

Siguiendo con la descripción de la transacción, tenemos el atributo *output count,* el cual indica la cantidad total de salidas contenidas en la transacción, por otra parte, el atributo *outputs* hace referencia propiamente a cada una de las salidas, estas están constituidas por los atributos *value* y *scriptPubKey* (Antonopoulos, 2017) (Walker, Transaction Data, 2015).

Dentro del elemento *ouputs,* se encuentra el atributo *value*, este corresponde al valor total de salida de la transacción expresado *satoshis*[43]. El *scriptPubKey size* se refiere al tamaño en bytes del *scriptPubKey*, este a su vez, corresponde al script que define las condiciones bajo las cuales una salida específica en la actual transacción podrá ser gastada cuando esta salida se utilice posteriormente como una entrada. Cuando la salida sea utilizada como entrada, el *scriptPubKey* deberá ser resuelto utilizando como argumentos los datos especificados en el *scriptSig* de la transacción con el objetivo de poder gastar la entrada.

El último elemento de una transacción es el *timelock*, este atributo permite definir una condición de tiempo determinada para poder utilizar la transacción en la cual fue definida, esta condición debe ser una estampa de tiempo Unix, por ejemplo, con este atributo se puede definir que una transacción especifica sólo sea procesada por el sistema Bitcoin cuando se cumpla una fecha y hora determinada, se puede pensar en este atributo como una característica que crea una transacción programada, que será aplicada en algún punto del tiempo.

A continuación, se muestra una tabla con el tamaño en bytes de cada uno de los atributos que componen una transacción en Bitcoin (Walker, Transaction Data, 2015):

[43] Un *satoshi* es la unidad monetaria mínima dentro del sistema Bitcoin, se puede pensar en los *satoshis* como si fueran centavos. A diferencia del dinero convencional donde una moneda puede ser dividida hasta en 100 pequeñas unidades de centavos como máximo, un bitcoin puede ser dividido hasta en 100 millones de *satoshis*, es decir, 100 millones de satoshis equivalen a un bitcoin (Antonopoulos, 2017).

Campo	Tamaño	Orden
Version	4 bytes	Little endian
Input count (numInput)	Variable de 1-9 bytes	Big endian
TXID	32 bytes	Little endian
VOUT (txOutIndex)	4 bytes	Little endian
ScriptSig size	Variable 1-9 bytes	Big endian
ScriptSig	Variable	Big endian
Sequence	4 bytes	Little endian
Output count (numOutput)	Variable 1-9 bytes	Big endian
Value (satoshis)	8 bytes	Little endian
ScriptPubKey size	Variable 1-9 bytes	Big endian
ScriptPubKey	Variable	Big endian
Locktime	4 bytes	Little endian

Tabla 3. Componentes de una transacción Bitcoin.

Para comprender mejor la estructura de las transacciones, se muestra a continuación, la primera transacción en la historia de Bitcoin, tanto su representación "cruda" en bytes con un formato hexadecimal, así como una versión a la cual se le aplicó un proceso *parse*[44] y posteriormente se codifico como una estructura JSON.

```
01000000010000000000000000000000000000000000000000000000000000000000000000ffffffff4d04ffff001d0
104455468652054696d65732030332f4a616e2f32303039204368616e63656c6c6f72206f6e206272696e6b206f6620
7365636f6e64206261696c6f757420666f722062616e6b73ffffffff0100f2052a01000000434104678afdb0fe55482
71967f1a67130b7105cd6a828e03909a67962e0ea1f61deb649f6bc3f4cef38c4f35504e51ec112de5c384df7ba0b8d
578a4c702b6bf11d5fac00000000
```

La cadena hexadecimal presentada arriba, muestra el aspecto real de una transacción, así es como las transacciones son almacenadas, procesadas y propagadas por el sistema Bitcoin, aunque como se verá más adelante, las transacciones no son procesadas de forma aislada, sino que se procesan en conjuntos de transacciones a través de una estructura llamada bloque (véase la sección 3.13).

[44] Con el término *parse* o *parsing*, se hace referencia a un proceso de análisis sintáctico sobre un conjunto ordenado de datos, a través de este proceso de análisis sintáctico se busca interpretar los datos para poder representarlos en algún formato legible o para poder llevar a cabo algún tipo de proceso sobre los mismos.

Las computadoras procesan de forma eficientes los datos binarios y hexadecimales, sin embargo, los seres humanos no podemos hacerlo, para poder ver y comprender el contenido de una transacción es necesario aplicar un proceso llamado *parse* y posteriormente alguna codificación o aplicar algún formato sobre los datos para poder representarlos de forma legible para los humanos[45], a continuación, se muestra una representación legible de la primera transacción en la historia de Bitcoin.


```
{
  "version": "01000000",
  "numInputs": 1,
  "inputs": [{
   "txId": "0000000000000000000000000000000000000000000000000000000000000000",
   "txoutIndex": "ffffffff",
   "unlockingScript":
"04ffff001d0104455468652054696d65732030332f4a616e2f3230303920436861696e63656c6c6f72206f
6e206272696e6b206f66207365636f6e64206261696c6f757420666f722062616e6b73",
   "sequence": "ffffffff"
  }],
 "numOutputs": 1,
 "outputs": [{
   "satoshis": "00f2052a01000000",
   "lockingScript":
"4104678afdb0fe5548271967f1a67130b7105cd6a828e03909a67962e0ea1f61deb649f6bc3f4cef38c4
f35504e51ec112de5c384df7ba0b8d578a4c702b6bf11d5fac"
  }],
 "lockTime": "00000000"
}
```


La representación mostrada en la parte superior, corresponde a una transacción en un formato de datos llamado JSON[46], este formato se ha utilizado con finalidad de poder visualizar los elementos de una transacción de forma sencilla, nótese que los datos hexadecimales no han sido alterados, únicamente fueron ordenados dentro de la estructura JSON para poder visualizarlos fácilmente, de hecho pueden identificarse visualmente los datos hexadecimales de la transacción cuando se encuentran en su forma natural con respecto a la representación JSON. En la estructura JSON se pueden apreciar todos los atributos de una transacción, tal y como se explicó hace unos momentos en esta sección, algunos atributos contienen nombres distintos con respecto a la figura 17, debido a que estos suelen nombrarse a veces de forma diferente por distintos autores e implementaciones diferentes de código, sin embargo, tienen el mismo significado y conservan el mismo orden de aparición dentro de la transacción.

[45] El apéndice B de esta obra contiene el código necesario para llevar a cabo el proceso *parse* sobre transacciones Bitcoin.

[46] Javascript Object Notation es un formato ligero de intercambio de datos, es un formato que puede ser entendido fácilmente tanto por humanos como por máquinas (json.org, s.f.).

Observe el lector que la representación JSON simplemente cumple un mero propósito de comodidad para los seres humanos, el sistema Bitcoin no almacena ni procesa las transacciones de esta forma.

En este punto una pregunta válida sería por qué una transacción puede tener muchas salidas y entradas, para entender mejor esto utilicemos un ejemplo con los personajes ficticios Alice y Bob, tal y como se hizo en la sección de criptografía. Supongamos que tanto Alice como Bob utilizan Bitcoin, ambos cuentan con bitcoins para poder realizar entradas y salidas, es decir, tienen bitcoins para gastarlos, ya sea comprando algún bien o servicio que acepte como pago bitcoins o simplemente transfiriendo fondos hacia otro usuario dentro del sistema Bitcoin.

Continuando con el ejemplo, Alice tiene 8 bitcoin exactos, esto es, en el registro de transacciones del sistema Bitcoin hay una transacción que contiene una salida destinada hacia Alice, esta salida tiene un valor de 8 bitcoins u 800 millones de satoshis. Alice decide comprar un producto que Bob le ha vendido en 2 bitcoins, para realizar esto, Alice utiliza su cartera digital y transfiere el dinero a Bob, ya que Alice cuenta con 8 bitcoins exactos, la transacción constará de una entrada de 8 bitcoins, sin embargo, tendrá dos salidas, una destinada a Bob con un valor de 2 bitcoins y otra salida de 6 bitcoins hacia la propia Alice, es decir, esta última salida corresponde al cambio que Alice recibirá una vez realizada la transacción. Lo anterior se debe a que en el sistema Bitcoin no es posible fragmentar una entrada de forma deliberada, la única forma permitida de dividir una entrada de n bitcoins o satoshis, es realizar una transacción y llevar a cabo múltiples salidas, esto parece extraño, pero no lo es, es la misma mecánica que se lleva a cabo con el dinero físico. Cuando se utilizan monedas o billetes, la única forma de fragmentar el valor de los billetes o mendas es llevar a cabo una transacción y obtener el “cambio” de la transacción, ya que no es posible romper un billete o moneda en n partes para decir que se tiene una fracción del valor que representa la totalidad del billete o la moneda.

En el pequeño ejemplo proporcionado, se ejemplifica un caso donde existe una entrada y dos salidas, sin embargo, podría haber más entradas o salidas, por ejemplo, un caso donde no se tiene los bitcoins o satoshis exactos para pagar, esto es, producir el valor de la salida requerida en la transacción. En este caso, se tendrían que utilizar varios satoshis o bitcoins para completar la transacción.

En este punto debe quedar claro que en el sistema Bitcoin no existen explícitamente los bitcoins, se trata simplemente de entradas y salidas contenidas en cada transacción llevada a cabo en Bitcoin. De acuerdo con Nakamoto, una moneda electrónica dentro del sistema Bitcoin, es una cadena de firmas digitales (Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008).

3.10 Llaves y Direcciones

La forma en la que se establece la propiedad, es decir, quien es el dueño de una entrada o salida en una transacción determinada, es por medio de firmas digitales y llaves criptográficas de cifrado asimétrico. Bitcoin utiliza el algoritmo de la curva elíptica para el uso de firmas digitales y llaves criptográficas, específicamente utiliza el algoritmo ECDSA (Antonopoulos, 2017).

ECDSA (*Elliptic Curve Digital Signature Algorithm*)[47], es un algoritmo de cifrado asimétrico utilizado para generar firmas digitales, la seguridad de este algoritmo se basa en el problema de logaritmos discretos en una curva elíptica definida por una ecuación matemática (Stallings, 2014). Como ya se expuso, la firma digital se implementa a través del cifrado asimétrico utilizando un par de llaves, una pública y una privada, en el caso del algoritmo ECDSA, el funcionamiento es el mismo. Sin embargo, la diferencia entre diferentes algoritmos de cifrado asimétrico estriba en el problema matemático en el cual están basados, por ejemplo, uno de los algoritmos asimétricos más famosos y dominantes durante décadas ha sido RSA (Rivest-Shamir-Adleman), que se encuentra basado en el problema matemático de la factorización de números primos de gran tamaño (Ferguson, Schneier, & Kohno, 2010).

El problema con el algoritmo RSA es que debido al gran poder de procesamiento que tienen las computadoras actuales, se requieren llaves de cifrado con tamaños cada vez más grandes en orden de mantener la seguridad de la información, actualmente se requieren llaves con longitudes iguales o superiores a los 2048 bits para proporcionar suficiente seguridad. En comparación, el algoritmo ECDSA proporciona la misma o mayor seguridad utilizando llaves de 160 hasta 512 bits. Para dimensionar la diferencia en la eficiencia de los algoritmos, veamos la equivalencia de seguridad de acuerdo con el tamaño de las llaves de cifrado, por ejemplo, para una llave RSA de 1024 bits, se puede obtener el mismo grado de seguridad con una llave ECDSA de 160 bits; para una llave RSA de 2048 bits, se obtiene el mismo nivel de seguridad con una llave ECDSA de 224 bits; con una llave RSA de 3072 bits, se obtiene el equivalente en seguridad con una llave ECDSA de 256 bits; y para una llave RSA de 7680 bits, se requeriría una llave ECDSA de tan sólo 384 bits para obtener el mismo nivel de seguridad (Stallings, 2014). Como se puede apreciar, la diferencia en el tamaño de llave requerida para el cifrado es enorme, este es el motivo por la cual Bitcoin utiliza el algoritmo ECDSA para las firmas digitales.

Ahora bien, para determinar quién es dueño de que transacciones o, mejor dicho, quien puede hacer uso de salidas específicas registradas en Bitcoin, se utilizan llaves criptográficas asimétricas generadas con un

[47] La descripción matemática y criptográfica acerca del funcionamiento de la curva elíptica, así como del algoritmo ECDSA, queda completamente fuera del alcance del presente documento. La obra de Stallings (Stallings, 2014) proporciona un acercamiento al funcionamiento de la curva elíptica.

algoritmo de curva elíptica que proporciona un par de llaves, una pública y una privada. Estas llaves son generadas y administradas de manera local por la aplicación de cartera digital o de otra aplicación informática programada para realizar transacciones en el sistema Bitcoin, sólo por medio de un par de llaves pública/privada es posible realizar transacciones en el sistema Bitcoin.

Dentro de cada entrada en una transacción, se encuentra la llave pública que permite[48] utilizar o gastar la salida referenciada en esa entrada, pero la llave pública no es agregada tal cual, dentro de la entrada, sino que es necesario aplicar un proceso adicional sobre la llave pública, este proceso consta de los siguientes pasos. Primero se obtiene un mensaje de digestión de la llave pública utilizando el algoritmo SHA256, el resultado se utiliza para obtener otro mensaje de digestión a partir del primero, pero ahora se utiliza el algoritmo RIPEMD160, por último, se codifica el hash resultante como una cadena hexadecimal. El resultado final de aplicar las funciones hash sobre la llave, se conoce *hash* de llave pública (*Public Key Hash*), mientras que el procedimiento de aplicar dos veces una función *hash* sobre un conjunto de datos se conoce como *double hash* (doble *hash*). Específicamente, el tipo de doble *hash* aplicado sobre la llave pública se llama hash160 (Antonopoulos, 2017). Cabe señalar que se aplica el doble *hash* sobre los bytes de la llave pública, es decir, sobre los datos "crudos" de la llave pública, adicionalmente el resultado final se codifica en hexadecimal.

Aunque dentro del sistema Bitcoin se utilizan los *hashes* de llave pública, estos *hashes* se encuentran codificados en hexadecimal tal y como se explicó en el párrafo anterior, por ello, los hashes de llave pública son sometidos a un tercer proceso de transformación para producir una cadena de caracteres más corta y fácil de utilizar para los usuarios, estas cadenas reciben el nombre de direcciones (*address*), estas direcciones únicamente son generadas para la conveniencia de los usuarios, son la representación a través de la cual las aplicaciones de cartera digital muestran a los usuarios las llaves públicas con las que cuentan, también se utilizan direcciones para representar las llaves públicas hacia las cuales se transfieren fondos, es decir, las llaves públicas de otros usuarios en Bitcoin.

Las direcciones son generadas a partir de un *hash* de llave pública, el procedimiento para obtener una dirección Bitcoin consiste en el siguiente procedimiento. Se genera un doble *hash* utilizando únicamente el algoritmo SHA256 sobre los bytes del hash160 obtenido a partir de la llave pública, este procedimiento se conoce como *checksum*. Posteriormente se extraen los primeros cuatro bytes del doble hash SHA256, para finalmente aplicar una codificación Base58 sobre el hash160 más los cuatro bytes extraídos, el resultado de la codificación sobre

[48] La llave pública no es el único elemento necesario para poder gastar una salida, es necesario hacer uso de la llave privada para generar una firma digital, estos elementos se utilizan para satisfacer una serie de condiciones preestablecidas bajo las cuales se autoriza el uso de la entrada (véase la sección 3.11).

los *hashes* concatenados genera una cadena de caracteres alfanumérica mucho más de fácil de identificar a simple vista, en contraposición con una cadena numérica hexadecimal.

A continuación, se muestra un ejemplo de llave pública ECDSA, así como la dirección Bitcoin derivada a partir de la llave pública[49].

Public Key
043059301306072a8648ce3d020106082a8648ce3d03010703420004a111701047695a85202cb48e8cdb0d2122c830efcf4e7fd154e811b22b3431e07120bc76525254c3d530a187570914f1f358d52204e4d272021c780e5207dfcc
Double Hash
004f85ec0c2005c671bf5d8ab4e75f8094c9d8695c
Chacksum (únicamente los últimos 4 bytes)
40801d4a
Bitcoin Address
18FUpDuMxJupCyiWFJfFbZkwfqMXmUcRyK

Tabla 4. Ejemplo de llave ECDSA y dirección Bitcoin.

Nótese la diferencia en longitud entre la llave pública y la dirección Bitcoin derivada a partir de esta, así como la facilidad visual para identificar la dirección Bitcoin con respecto a la llave pública ECDSA.

Estas direcciones pueden verse como el equivalente de los números de cuenta bancaria que utilizan las instituciones financieras. A diferencia del sistema financiero tradicional donde un cliente cuenta generalmente con una única cuenta bancaria para llevar a cabo transacciones financieras, en Bitcoin se pueden utilizar muchas direcciones, esto es, se pueden tener muchos pares de llaves publica/privada. Como ya se explicó, estas direcciones son en realidad hashes de llaves públicas a las cuales se les aplicó un procedimiento para producir una cadena de caracteres corta y legible para los seres humanos, por lo que estas direcciones no son utilizadas en ningún momento dentro del sistema Bitcoin. Debe quedar claro que Bitcoin sólo utiliza llaves criptográficas y no direcciones.

[49] El apéndice B de esta obra contiene el código necesario para crear pares de llaves pública/privada ECDSA, así como la generación de direcciones Bitcoin a partir de la llave pública.

3.11 Bloqueo de Entradas y Salidas

De acuerdo con lo explicado en la sección de Transacciones, una transacción consta principalmente de entradas y salidas, cada salida contiene un *scriptPubKey* que contiene las condiciones bajo las cuales se puede gastar esa salida, estas condiciones, son un conjunto de instrucciones escritas en un lenguaje de programación *script* que no es completamente Turing.

Este lenguaje llamado Script, permite ejecutar instrucciones siguiendo una pila de ejecución, esto es, todas las instrucciones son insertadas en una pila y posteriormente son ejecutadas conforme las instrucciones se extraen de la pila. Al ser un lenguaje que no es completamente Turing, carece de la mayoría de las funcionalidades de un lenguaje de programación moderno, esto se ha hecho deliberadamente, la falta de capacidad en el lenguaje Script de Bitcoin es una característica de seguridad que previene diversos ataques cibernéticos sobre el sistema, ya que cada transacción contiene un *script* con instrucciones que son ejecutadas por cada nodo de la red Bitcoin para validar la transacción, una transacción con un script malicioso sería un gran peligro para el sistema (Antonopoulos, 2017).

La única forma de poder utilizar una salida como entrada, esto es, la única forma de poder gastar una salida es proveer en la entrada de una transacción, los datos por medio del *scriptSig* que satisfagan las instrucciones codificadas con el lenguaje Script dentro del elemento *scriptPubKey* de la salida. Es decir, la unca forma de poder gastar un bitcoin es contar con los datos que satisfacen una condición previamente establecida en una salida, generalmente, estos datos corresponden a una firma digital. Esto tiene una gran implicación sobre el sistema Bitcoin, ya que a través de esta característica se puede programar bajo qué condiciones es posible gastar las salidas o de forma abstracta, bajo qué condiciones es posible gastar un bitcoin.

Antonopoulos llama a esta capacidad de poder determinar de forma programática como se gastan los bitcoins como, “dinero programable” (Antonopoulos, 2017). Esto significa que se pueden crear diversas condiciones tanto simples como complejas, acerca de cómo se pueden utilizar las salidas, por ejemplo, con el dinero programable es posible crear condiciones que restrinjan el uso de salidas en base al cumplimiento de una fecha y hora determinada, que las salidas requieran la autorización de múltiples usuarios, una combinación de los ejemplos anteriores o cualquier condición que pueda ser programada utilizando el lenguaje Script.

La mayoría de las transacciones en Bitcoin se ejecutan por medio de una condición o *locking script*[50], que consiste en la validación de una llave pública y una firma digital, es decir, este tipo de *lock script* establece que

[50] Un *locking script* es un código de programación escrito en el lenguaje *Script* de Bitcoin por medio de cual se establece un “acertijo” (*puzzle*) que tiene que tiene que resolverse para poder utilizar la salida como entrada. Para resolver el

para poder utilizar una salida se requiere proporcionar una firma digital correspondiente con las entradas o salidas de la transacción a la cual pertenece la salida, así como la llave pública que permite verificar la firma.

A continuación, se muestra un ejemplo de un script de tipo P2PKH[51] (Antonopoulos, 2017):

```
Locking script
DUP HASH160 <PublicKeyHash> EQUALVERIFY CHECKSIG

Unlocking script
<signature><PublicKey>
```

En el código del ejemplo anterior se tiene un *locking script*, así como su correspondiente *unlocking* script que se ejecutan de forma separada, primero se ejecuta el *unlocking script* y posteriormente se ejecuta el *locking* script que utiliza como argumentos los datos de la pila generada por el *unlocking script* (Antonopoulos, 2017). Si y sólo si, la ejecución de ambos *scripts* es correcta, es decir, no hay ningún error durante la ejecución de ambos *scripts* y el *unlocking script* es capaz de satisfacer con éxito las condiciones del *locking script*, se podrá utilizar la salida como entrada, en caso contrario, la transacción es cancelada. Estas validaciones son llevadas a cabo por todos o la mayoría de los nodos en la red Bitcoin, sólo aquellas transacciones que son válidas son registradas de forma permanente en el registro de Bitcoin.

Siguiendo la ejecución de código del ejemplo, primero se ejecuta el *unlocking script*, esto es, se agrega la firma digital en la pila, después se agrega la llave pública en la pila. El siguiente paso consiste en copiar la pila generada por el *unlocking* script en la pila de ejecución del *locking script* con la finalidad de poder utilizar esos datos. Una vez copiada la pila, la primera instrucción sería DUP, que significa que se toma el primer elemento de la pila, en este caso se toma la llave pública, posteriormente se copia o duplica la llave y se agrega en la cima de la pila, con lo cual se tiene la misma llave pública dos veces dentro de la pila de ejecución. El siguiente comando es HASH160, el cual consiste en tomar el primer elemento de la pila, que es otra vez la llave pública, y aplicar el doble hash160, el resultado se agrega en la cima de la pila. El siguiente paso en el *script* no es un comando, es un dato, es el hash160 de la llave pública correspondiente al usuario hacia el cual está dirigida la salida o, visto de otra forma, es la llave pública que corresponde al usuario hacia el cual se transfirió cierta cantidad de bitcoins, este hash160 de llave pública esta codificado dentro del *locking script,* de manera que no es proporcionado por el *unlocking script*, esto quiere decir que ese hash fue embebido dentro del script cuando

acertijo es necesario proporcionar un *unlocking script*, que es el conjunto de datos que satisfacen el *lock script* y por lo tanto permiten el uso de bitcoins (Antonopoulos, 2017).

[51] *Pay to Publick Key Hash* (P2PKH), es el *locking script* más común en el sistema Bitcoin, aunque no es el único, pero si uno de los más sencillos, establece como condición la validación de la llave pública y de una firma digital (Antonopoulos, 2017).

se creó la salida, por lo tanto, no pude ser modificado y sólo el verdadero receptor de la salida contará con las llaves correspondientes para utilizar la salida.

Regresando con la explicación[52] de la ejecución del *script*, el hash160 contenido en el *locking script* se agrega en la cima de la pila de ejecución, nótese que ahora hay en la pila de ejecución dos *hashes* de tipo hash160, uno calculado con los datos proporcionados por el *unlocking script* y el otro obtenido del *locking script*. La siguiente instrucción es EQUALVERIFY, este comando significa que se toman los dos primeros elementos de la pila de ejecución, en este caso son los dos *hashes*, posteriormente se comparan para determinar su igualdad, si ambos *hashes* son iguales la operación de comparación retorna verdadero, en caso contrario retorna falso, en cuyo caso la validación del *locking script* es falsa y la transacción se cancela. Si la validación anterior es verdadera, se procede a ejecutar la última instrucción disponible en el script, el comando CHECKSIG, este comando toma los dos primeros datos de la pila, en este caso, los últimos datos contenidos en la pila de ejecución, la llave pública y la firma digital, con estos datos determina si la firma digital es válida, es decir, utiliza la llave pública parar validar la firma digital, si todo es correcto, esta última operación devuelve verdadero y la transacción es validada, esto es, se puede utilizar o gastar la salida como entrada. Si la validación de la firma digital falla la operación devuelve falso y la transacción es cancelada.

El ejemplo anterior es sólo una de las formas en las cuales se establece cómo se puede gastar una salida, como ya se mencionó, es posible establecer condiciones para que la salida pueda ser gastada con la aprobación de múltiples partes, esto es, mediante la validación de múltiples firmas digitales, en esta situación, se asegura que los fondos no puedan ser utilizados deliberadamente por una sola de las partes, ya que se requeriría el conceso de todos los usuarios definidos. De igual manera, es posible establecer condiciones de tiempo, por ejemplo, que una salida no pueda ser utilizada sino hasta 15 días después de que la transacción que contiene la salida ha sido procesada y aceptada por el sistema Bitcoin, también se podría especificar una hora exacta y combinarlo con la firma digital múltiple. Todos estos beneficios son posibles gracias a la capacidad única de, dinero programable, que proporciona el sistema Bitcoin a través de la programación con el lenguaje *Script*, que permite definir el bloqueo y desbloqueo de transacciones.

[52] La explicación del ejemplo presentado sobre el funcionamiento del lenguaje Scritp de Bitcoin para llevar a cabo el *locking* y *unlocking*, está basado en la obra de Antonopoulos, para obtener una descripción completa de este procedimiento véase (Antonopoulos, 2017).

3.12 Encadenamiento de Transacciones

En las secciones anteriores se revisaron las transacciones, así como la forma en la que estas contralan la forma en la pueden ser utilizadas a través de *scripts* programables, con esta concepción adquirida sobre las transacciones, ahora se proporcionara una visión acerca de cómo las transacciones se enlazan unas con otras, en un flujo lineal determinado por todas las transacciones previas registradas en el sistema Bitcoin.

Tal y como lo explica Nakamoto, no existen explícitamente las monedas dentro de Bitcoin, en vez de esto, se puede ver a cada una de las transacciones enlazadas como una moneda electrónica, es decir, como bitcoins, aunque esto último es solamente una concepción abstracta.

Cada entrada en una transacción cuenta con una referencia hacia una transacción que ocurrió en el pasado, que es válida y por lo tanto se encuentra registrada dentro de la historia de transacciones del sistema Bitcoin, esta referencia que existe entre las entradas y salidas de una transacción forma una cadena que representa una suerte de línea de tiempo histórica, del flujo de todos y cada uno de los bitcoins que son transaccionados en todo el sistema, de manera que es posible rastrear como ha sido gastado cada bitcoin, así como quién lo ha gastado, concretamente, que llaves públicas están asociadas a cada transacción.

Debido a que el sistema Bitcoin no registra ningún dato correspondiente con un balance sobre la cuenta de un usuario o su saldo, estos datos son reconstruidos por las aplicaciones de cartera digital obteniendo los registros de todas las transacciones asociadas con las llaves públicas de un usuario específico, una vez obtenidas las transacciones, la cartera digital llevan a cabo los cálculos de acuerdo a todas las entradas y salidas en las transacciones que pertenecen al usuario, es decir, que corresponden con sus llaves públicas.

Esta referencia que existe entre las transacciones se puede interpretar como un encadenamiento de transacciones, esto además de permitir que las aplicaciones de cartera digital proporcionen datos útiles a los usuarios, también hace posible en todo momento, la verificación de forma transparente y confiable, de todas y cada una de las transacciones llevadas a cabo. Esto se debe a que toda la historia de las transacciones es pública, la historia de todas las transacciones se encuentra distribuida en todos los nodos de la red, es decir, todos los nodos de la red Bitcoin comparten una misma historia, un único registro distribuido de información. Aunado a esto, debido a que todo el encadenamiento y bloqueo de transacciones se realiza de forma criptográfica, el proceso además de asegurar que sólo el dueño de la llave privada puede gastar la salida, también proporciona una forma de auditar fácilmente las transacciones por medio de las funciones hash y las firmas digitales.

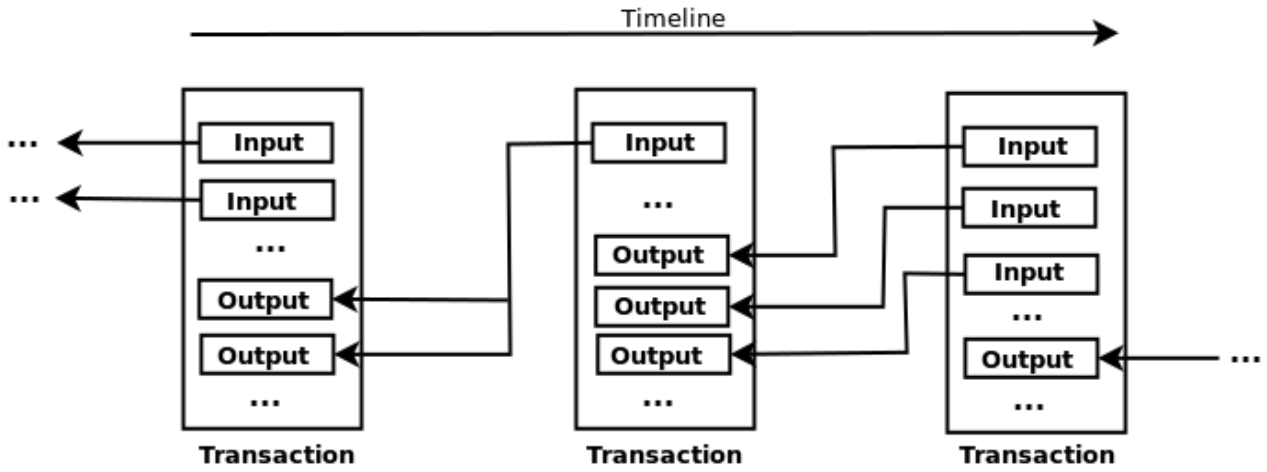

Figura 18. Encadenamiento de transacciones.

En la figura 18 se muestra de forma gráfica el encadenamiento de transacciones, como se puede apreciar, cada entrada es una referencia a una salida previa, a su vez, estas salidas son el resultado de una entrada, esto es, todas las entradas son el flujo de transferencia de valor entre todos los usuarios del sistema Bitcoin a través del tiempo. Es importante señalar que una entrada puede estar compuesta por muchas o una sola salida, de igual forma las salidas pueden estar compuestas de muchas entradas o una sola entrada, además, se debe observar que las salidas de una entrada no necesariamente corresponden con la última transacción en Bitcoin, esto es, una transacción puede estar encadena con diferentes transacciones en diferentes puntos del tiempo, ya que el enlace entre las transacciones es por medio del identificador único de la transacción conocido con el nombre de TXID, que corresponde con el hash de una transacción validada y registrada en el registro de transacciones de Bitcoin.

3.13 Bloques

Los bloques (*blocks*) en Bitcoin son una estructura lógica de datos que permite agrupar un conjunto de transacciones para que estas puedan ser procesadas y validadas por los nodos de la red Bitcoin, cada bloque está compuesto por una cantidad finita de transacciones, el límite del bloque está dado en bytes no en transacciones, por lo que un bloque podrá contener hasta el máximo de bytes definido en el sistema Bitcoin, actualmente el límite definido en el sistema es 1 MiB[53].

[53] Nótese la diferencia entre MiB (Mebibyte) y MB (Megabyte), la segunda es incorrecta. MiB es la abreviación de Mebibyte, la unidad de medida utilizada para hacer referencia a un conjunto de 1,048,576 bytes, esto es, 2 elevado a la

Un bloque es la estructura más grande dentro del sistema Bitcoin, estos bloques son organizados de manera secuencial inversa, cada uno de los bloques se encuentra enlazado con el bloque anterior, este enlace se repite hasta que se alcanza el primer bloque creado en toda la historia de Bitcoin, el bloque cero, conocido también como el bloque génesis. Los bloques se almacenan en cada uno de los nodos de la red Bitcoin como datos binarios con orden *little endian*[54], en archivos que tienen el siguiente nombre. Un prefijo con las letras "blk", como abreviatura de *block* (bloque), seguido de un número de hasta 5 bits que representa el número del archivo, y una extensión ".dat" (Walker, blk.dat, 2017).

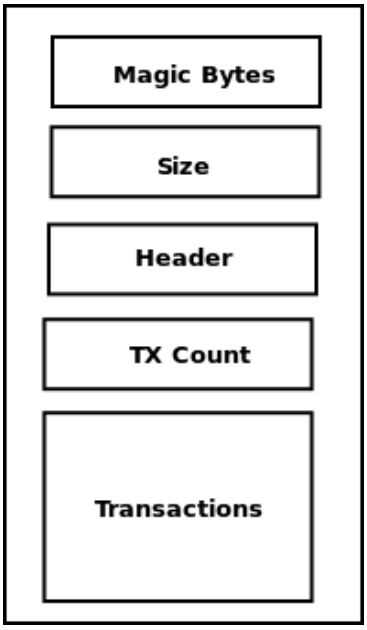

Figura 19. Anatomía de un bloque.

En la figura 19 se muestra la estructura de un bloque Bitcoin, el primero de los elementos contenidos en el bloque son los *magic* bytes, este conjunto de bytes es estático e identifica el tipo de datos, en este caso, identifica que los datos corresponden a un bloque de Bitcoin. El siguiente de los atributos es *size*, este conjunto de bytes corresponde al tamaño total del bloque, es decir, cuantos bytes ocupa todo el bloque. El atributo *header*

potencia 20 (véase el apéndice A). La razón de esto se debe a que las computadoras utilizan el sistema binario y no el sistema decimal, por lo que es incorrecto aplicar unidades de medida decimales. Cuando se utilizan prefijos del sistema internacional de medidas existen una serie de prefijos específicos para el sistema binario, son análogos al del sistema decimal, pero se calculan con una base binaria. Hacer uso de los prefijos decimales en el sistema binario no sólo es un error técnico, representa una pérdida importante en la precisión de bytes de la cual se está hablando, esto último no es trivial.

[54] Dependiendo del orden en el que los bytes son almacenados se dice que son de orden *little* o *big* endian. Cuando los bytes son almacenados con orden *big endian* el bit más significativo se encuentra en la posición numérica más alta, mientras que en el orden *little endian* el bit menos significativo se encuentra en la posición numérica más alta.

corresponde con el encabezado del bloque, este elemento es otra estructura de datos que contiene metadatos acerca del bloque. El atributo *tx count* identifica el número de transacciones contenidas dentro del bloque. Finalmente, el atributo *transactions* corresponde a todas las transacciones contenidas en el bloque.

A continuación, se muestra una tabla con el tamaño en bytes de cada uno de los atributos que componen un bloque en Bitcoin (Walker, blk.dat, 2017):

Campo	Tamaño	Orden
Magic bytes	4 bytes	Constante: 0xD9B4BEF9
Size	4 bytes	Little endian
Header	80 bytes	Véase la tabla 6
Tx count	Variable de 1-9 bytes	Big endian
Transactions	Variable	Véase la tabla 3

Tabla 5. Componentes de un bloque Bitcoin.

Como se mencionó, el atributo *header* de un bloque está compuesto por otros elementos que proporcionan información esencial sobre el bloque, es a partir del *header* de un bloque que se obtiene el hash que identifica de manera única a cada uno de los bloques en el sistema Bitcoin.

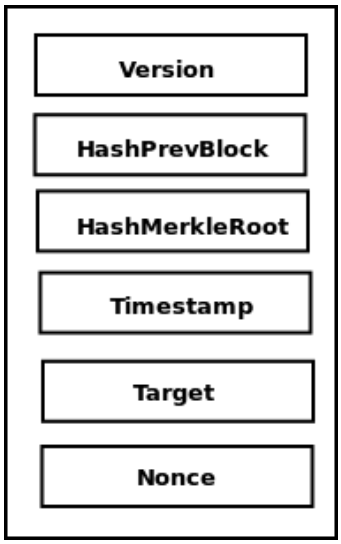

Figura 20. Anatomía del encabezado(*header*) de un bloque.

En la figura 20 se muestran todos los elementos que conforman el encabezado (*header*) de un bloque, el primero de estos atributos es la versión (*version*) que indica precisamente cual es la versión que se está utilizando del sistema Bitcoin. El siguiente atributo es hashPrevBlock, este indica el *hash* del bloque anterior, es decir, es el identificador del bloque antecesor. Después se encuentra el hashMerkleRoot, que corresponde a otro *hash*, esta vez se refiere a un hash que identifica de manera única todas las transacciones contenidas dentro del bloque.

El *timestamp* es un atributo que identifica el momento en el cual se creó el bloque, corresponde a una estampa de tiempo Unix. El *target*[55] corresponde con el punto de referencia establecido para determinar el *hash* válido que se debe obtener a partir del *hash* de un bloque dado cuando se lleva a cabo la *proof-of-work* (véase la sección 3.15). Por último, el *nonce* corresponde con los bytes utilizados para generar el hash de un bloque y que prueban la *proof-of-work* (véase la sección *3.15*).

A continuación, se muestra una tabla con el tamaño de cada uno de los atributos contenidos en el encabezado de un bloque Bitcoin (Walker, Block Header, 2016):

Campo	Tamaño	Orden
Version	4 bytes	Little endian
HashPrevBlock	32 bytes	Big endian
HashMerkleRoot	32 bytes	Big endian
Timestamp	4 bytes	Little endian
Target (Bits)	4 bytes	Little endian
Nonce	4 bytes	Little endian

Tabla 6. Componentes del *header* de un bloque Bitcoin.

En la parte inferior se muestra un bloque Bitcoin "crudo" en formato hexadecimal, este bloque corresponde al bloque génesis, se muestra su representación natural, esto es, la forma en la que el sistema Bitcoin procesa, almacena y propaga los bloques a través de la red Bitcoin.

```
f9beb4d91d0100000100000000000000000000000000000000000000000000000000000000000000000000003ba3edf
d7a7b12b27ac72c3e67768f617fc81bc3888a51323a9fb8aa4b1e5e4a29ab5f49ffff001d1dac2b7c01010000000100
0000000000000000000000000000000000000000000000000000000000ffffffff4d04ffff001d0104455468652
054696d65732030332f4a616e2f32303039204368616e63656c6c6f72206f6e206272696e6b206f66207365636f6e64
206261696c6f757420666f722062616e6b73ffffffff0100f2052a01000000434104678afdb0fe5548271967f1a6713
0b7105cd6a828e03909a67962e0ea1f61deb649f6bc3f4cef38c4f35504e51ec112de5c384df7ba0b8d578a4c702b6b
f11d5fac00000000
```

Para entender cada uno de los elementos del bloque presentado en la parte superior, se muestra a continuación, la representación de ese bloque en formato JSON, esta representación se logra por medio de un *parse* sobre los datos "crudos" hexadecimales de los bloques Bitcoin[56].

[55] El campo *target* dentro del *header* del bloque también es conocido con el nombre de bits.

[56] El apéndice B contiene el código de programación necesario para realizar un *parse* sobre bloques Bitcoin.

```
{
   "magicNumber": "d9b4bef9",
   "blockSize": 285,
   "blockHeader": {
    "version": "01000000",
    "hashPrevBlock":
"0000000000000000000000000000000000000000000000000000000000000000",
    "hashMerkleRoot":
"3ba3edfd7a7b12b27ac72c3e67768f617fc81bc3888a51323a9fb8aa4b1e5e4a",
    "timestamp": "29ab5f49",
    "targetDifficulty": "ffff001d",
    "nonce": "1dac2b7c"
   },
   "numTransactions": 1,
   "transactions": [
   {
    "version": "01000000",
    "numInputs": 1,
    "inputs": [{
     "txId": "0000000000000000000000000000000000000000000000000000000000000000",
     "txoutIndex": "ffffffff",
     "unlockingScript":
"04ffff001d0104455468652054696d65732030332f4a616e2f32303039204368616e63656c6c6f72206f
6e206272696e6b206f66207365636f6e64206261696c6f757420666f722062616e6b73",
     "sequence": "ffffffff"
    }],
   "numOutputs": 1,
   "outputs": [
   {
    "satoshis": "00f2052a01000000",
    "lockingScript":
"4104678afdb0fe5548271967f1a67130b7105cd6a828e03909a67962e0ea1f61deb649f6bc3f4cef38c4
f35504e51ec112de5c384df7ba0b8d578a4c702b6bf11d5fac"
   }],
   "lockingScript": "00000000"
  }]
}
```


Note el lector que la representación mostrada sobre un bloque Bitcoin en la parte superior corresponde a una versión “embellecida” del bloque tras aplicar un formato JSON, esta representación tiene el único propósito de facilitar a los humanos la identificación de cada uno de los atributos contenidos dentro de un bloque Bitcoin, sin embargo, debe quedar claro que los bloques son procesados por el sistema Bitcoin como un conjunto de datos binarios, tal y como se puede apreciar en la representación hexadecimal del bloque mostrada en esta sección.

3.14 Encadenamiento de Bloques (*Block Chain*)[57]

Aunque las transacciones son la estructura más importante en el sistema Bitcoin, estas no son procesadas ni almacenadas de forma individual, en su lugar, las transacciones son agrupadas dentro de un bloque, cada bloque contiene una cantidad n de transacciones. Para poder ser incluidas dentro de un bloque, todas las transacciones deben de ser sujetas de un proceso de validación por parte del nodo que está creando un bloque particular, ninguna transacción inválida puede ser incluida dentro un bloque. Cuando un bloque es propagado en la red bitcoin para ser incluido dentro de toda la historia de transacciones en Bitcoin, todos o la mayoría de los nodos en la red Bitcoin deben validar el bloque, esto es, deben validar que toda la información contenida en el bloque sea correcta, incluida cada una de las transacciones dentro del bloque, sólo aquellos bloques que han sido validados por la mayoría de nodos en la red Bitcoin son registrados de forma definitiva dentro de la historia global de transacciones en todos los nodos pertenecientes al sistema Bitcoin.

Ya que las transacciones son embebidas dentro de bloques para su procesamiento y posterior almacenamiento de forma permanente, los bloques forman la estructura genérica dentro del sistema Bitcoin, es decir, son la estructura de datos más grande que se puede procesar, transportar y almacenar. Por esta razón, los bloques también son enlazados o "encadenados" unos con otros siguiendo un orden lógico en el que un bloque b es encadenado con el siguiente bloque b+1, el bloque b+1 se encadena con el bloque b+2, etc. Esto se repite de forma indefinida, desde el primer bloque hasta el último bloque generado en un momento determinado en el tiempo.

Para poder establecer el encadenamiento entre los diferentes bloques, se utiliza un *hash* que se calcula a partir de toda la información contenida dentro del *header* de un bloque, para obtener el *hash* que identifica de manera única cada bloque se realiza el siguiente procedimiento. Todos los atributos dentro del *header* deben encontrarse en orden *little endian*, los que no lo son deben ser convertidos, en este caso, únicamente se trata de los dos *hashes* correspondientes al Merkle *root* y al *hash* del bloque anterior. Posteriormente se concatenan todos los elementos del *header* en el mismo orden en el que aparecen en la figura 20, primero el atributo *version*, luego el *hash* del bloque anterior, etc. Finalmente se realiza un doble *hash* con el algoritmo SHA256 sobre los bytes "crudos", el resultado se codifica en hexadecimal, la cadena resultante en hexadecimal corresponde al

[57] En la especificación de Bitcoin nunca se define o se habla del término "*blockchain*". Nakamoto habla sobre bloques, como una estructura contenedora de transacciones, y de los bloques que son encadenados unos con otros por medio de un *hash* que identifica cada bloque de manera única. En este sentido, se habla del encadenamiento de bloques o *block chain* (nótese la diferencia entre *blockchain* y *block chain*) simplemente como una característica más dentro del sistema.

hash de todo el bloque, es el identificador único que se utilizará tanto para localizar el bloque como para encadenarlo con el bloque siguiente[58].

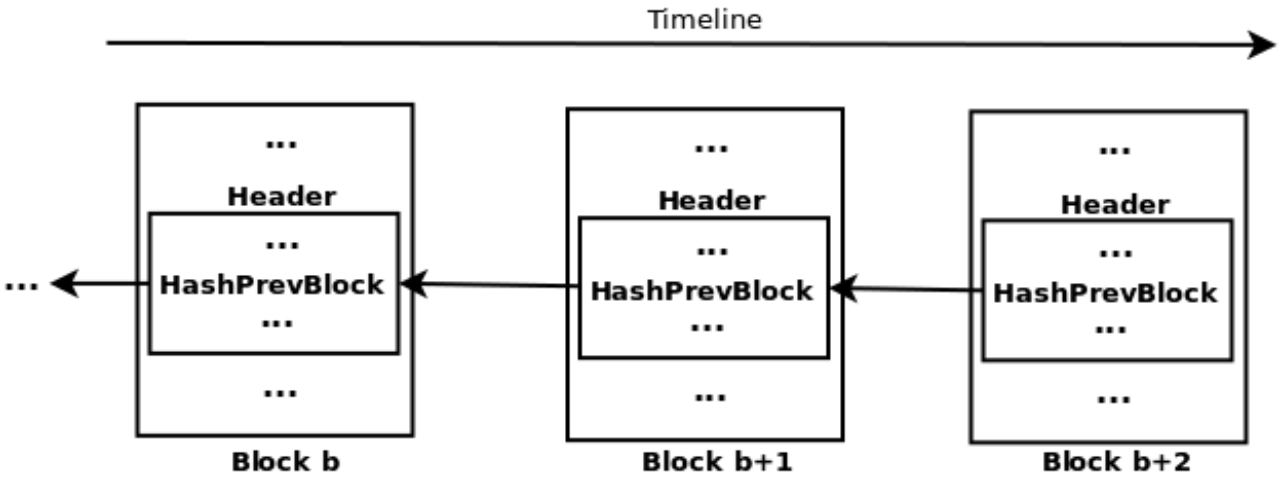

Figura 21. Encadenamiento de bloques (*block chain*).

En la figura 21 se puede observar el encadenamiento de bloques, como se puede apreciar, los bloques son encadenados con el bloque anterior, esto es, los bloques se encadenan hacia atrás, de forma inversa, ya que sólo es posible encadenar un bloque con uno que ya existe, por lo tanto, sólo es posible realizar el encadenamiento con los bloques previamente creados y validados en el pasado. También debe observarse que en realidad los bloques son enlazados por medio del *header* de cada bloque, si bien la referencia explícita entre los bloques se lleva por medio del *hash* del bloque, se debe tener en cuenta que el hash del bloque es creado a partir de toda la información contenida dentro del *header* del bloque. Por último, debe observarse que el *hash* que identifica cada bloque se almacena en bloque que lo referencia, es decir, un bloque nunca almacena su propio *hash,* sino que almacena el *hash* de un bloque anterior al que hace referencia.

3.15 *Proof of Work*

En las secciones anteriores se revisaron las estructuras de datos utilizadas por el sistema Bitcoin para almacenar y procesar la información, así como la forma en la que estas estructuras se enlazan para mantener un orden

[58] Una aclaración importante sobre el *hash* que identifica cada bloque de manera única es que este se genera a partir de los elementos contenidos en el *header* del bloque como ya se expuso, sin embargo, uno de estos atributos dentro del *header* es el *nonce*, el cálculo de este se realiza con base en el procedimiento de la *proof of work* (véase la sección 3.15).

lógico de cada una de las transacciones a través del tiempo. En estas estructuras se encuentra la información acerca de cómo se ha transferido valor a través del tiempo por medio de las entradas y salidas de cada transacción entre todos los usuarios del sistema Bitcoin, desde el bloque génesis hasta el más reciente en un punto determinado del tiempo, sin embargo, la pregunta debemos hacernos es la siguiente, ¿cómo garantiza el sistema que cualquier nodo de la red no altere la historia de las transacciones modificando alguno de los bloques para gastar o utilizar más de una vez una salida, *double spending*, o elimine deliberadamente las salidas de cualquier usuario?.

Debido a que todos los bloques son propagados por la red Bitcoin a todos los nodos y a su vez estos bloques son públicos, es decir, cualquier entidad puede convertirse en un nodo de la red Bitcoin y examinar los bloques, se requiere un mecanismo a través del cual se impida la modificación no autorizada de los bloques, esto es, se requiere un método que proporcione inmutabilidad en los bloques, de forma que no sea posible llevar a cabo el *double spending*, o cualquier alteración en el registro de transacciones del sistema que comprometa la integridad de la información.

También se necesita un mecanismo por medio del cual se establezca que bloques son aceptados de forma unánime por todos los nodos en la red Bitcoin, ya que cualquier nodo en la red Bitcoin puede validar transacciones y crear bloques para posteriormente propagarlos en la red para que ese bloque sea incluido en la historia global de registros de Bitcoin, esto es, que el bloque sea aceptado por todos los nodos de la red Bitcoin para posteriormente almacenarlo de forma permanente, por lo que se necesita un mecanismo de consenso que garantice que sólo uno de los bloques propagados en la red será aceptado por todos los nodos de la red.

La solución propuesta por Nakamoto es la *proof of work*, la *proof of work* es un mecanismo a través del cual se solicita una prueba que demuestre que se ha llevado a cabo una cierta cantidad de trabajo computacional en un intervalo de tiempo específico (Jakobsson & Juels, 1999).

Nakamoto propone el uso del algoritmo SHA256 para realizar la *proof of work*, concretamente, especifica el uso de una función *hash* para generar un valor que contenga una cantidad determinada de bits establecidos como ceros al inicio del valor hash (Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008). La razón para todo esto, se justifica en el hecho de que cada nodo que intente crear un bloque deberá calcular un hash con el algoritmo SHA256 que contenga al inicio una cantidad determinada de bits establecidos como ceros, esta será la restricción para cada nodo que intente propagar un bloque en la red para que sea aceptado por todos los demás nodos, comprobando que se ha realizado un trabajo computacional. Es precisamente este trabajo computacional el que representa una garantía de que el nodo no intenta engañar al sistema manipulando de

forma deliberada las transacciones del bloque, pero esta prueba de trabajo computacional no sólo funge como mecanismo de consenso para la aceptación de los bloques en la red Bitcoin, también establece un importante mecanismo de seguridad que mantiene la inmutabilidad de la información contenida en cada uno de los bloques del sistema.

La *proof of work* es la piedra angular en todo el mecanismo de seguridad del sistema Bitcoin, es precisamente la prueba de trabajo computacional la que mantiene la seguridad, la inmutabilidad y el consenso dentro de todo el sistema Bitcoin, a diferencia de lo que se podría pensar, el encadenamiento de bloques no proporciona seguridad, mucho menos utilidad alguna si se descarta la *proof of work*. Sin la prueba de trabajo computacional no hay manera de asegurar que la información contenida en los bloques no sea manipulada por un atacante, que las transacciones no sean gastadas más de una vez, *double spending*, o llegar a un consenso entre todos los nodos de la red para determinar que bloque es el que se incluye de forma permanente en el sistema Bitcoin.

Para comprender porque la *proof of work* es la pieza clave en el sistema Bitcoin, debemos entender cómo se lleva a cabo el proceso de la prueba de trabajo computacional en el sistema Bitcoin, ya que, a través de la comprensión de su funcionamiento, quedará claro su importancia y porque ninguno de los sistemas actuales que tratan de basar su funcionamiento en el de Bitcoin, pueden ofrecer las ventajas de este. Cuando estos novedosos sistemas eliminan, demeritan o no utilizan una prueba de trabajo lo suficientemente segura, terminan dejando la seguridad en manos de un concepto nebuloso y difuso como lo es el "*blockchain*".

En la sección de funciones *hash* (véase la sección 3.7) se estableció sobre estas funciones que su resultado puede ser interpretado como una distribución uniforme de variables discretas aleatorias, debido a que el resultado final de una función hash es aleatorio y que sus valores son discretos, esto es, ya que las computadoras procesan y representan la información con el sistema binario, el resultado de una función siempre es un conjunto de unos y ceros, es decir, un conjunto de datos discretos, y debido a que no se puede determinar de antemano cuál será la salida de una función hash para un conjunto ordenado de datos, se puede decir que el resultado es un conjunto ordenado de datos aleatorios. Bajo esta concepción se deduce que obtener un resultado de una función hash, esto es, un mensaje de digestión, cuyos primeros bits sean una secuencia determinada de unos o ceros, sólo puede ser generado por medio de un proceso de fuerza bruta[59], donde la capacidad de obtener un hash que cuente con una determinada secuencia binaria está determinada por la probabilidad.

[59] La "fuerza bruta" es un método por medio del cual se realiza una búsqueda secuencial a través de todos los posibles valores para obtener un resultado deseado. Por ejemplo, en criptografía, un ataque de fuerza bruta (Dulaney & Easttom, 2018) es aquel en el cual se prueban todas las llaves posibles a través de cada posible combinación de datos que pueden generar una llave válida con la finalidad de descifrar la información cifrada.

En Bitcoin, la prueba de trabajo computacional que se debe llevar a cabo para cada bloque generado por un nodo especifico en la red, para que este sea aceptado por todos los demás nodos, consiste en generar un *hash* que cumpla con ciertas características especificadas por el sistema Bitcoin. Este *hash* se obtiene con el algoritmo SHA256 sobre el *header* del bloque, es decir, sobre el conjunto de datos que componen el *header* del bloque (véase la sección 3.13), dentro de este conjunto de datos se encuentra el atributo *nonce*, este atributo numérico se modifica de forma específica durante la generación de cada bloque para obtener un *hash* de bloque que cumple con las especificaciones del sistema.

El sistema Bitcoin establece que la prueba de trabajo necesaria para que un bloque sea aceptado como válido, corresponde con la generación de un *hash* a partir de la información en el *header* del bloque, cuyo valor hexadecimal se encuentre por debajo de un número hexadecimal concreto, esto quiere decir que un nodo en la red Bitcoin que quiera agregar un nuevo bloque en la historia de registros de Bitcoin, deberá generar un *hash* cuya secuencia de unos y ceros, correspondan a un valor numérico inferior al número objetivo (*target*) establecido por el sistema, para esto, los nodos utilizaran el *nonce* como variable para modular la salida de la función *hash*, generalmente, el *nonce* es un número que se va incrementando secuencialmente hasta que se encuentra un *nonce* que combinado con todos los demás atributos en el *header* del bloque, generan un *hash* cuya secuencia de bits cumplen con la secuencia establecida por Bitcoin[60]. Si un nodo en la red Bitcoin genera el *hash* con la secuencia requerida antes que ningún otro nodo en la red, entonces el bloque generado por ese nodo es el bloque que se almacena de forma permanente en la historia de transacciones de Bitcoin.

Por supuesto que, una vez que un nodo propaga un bloque nuevo en la red para que este sea incluido de forma permanente en la historia del sistema, el bloque es validado por todos los nodos antes de que estos decidan registrar el bloque de forma permanente, no sólo el contenido del bloque debe ser correcto, es decir, que las transacciones son válidas, el bloque también debe contar con la prueba de trabajo, esta corresponde al atributo *nonce* (véase la sección 3.13). Al incluir el *nonce* dentro del bloque, todos los demás nodos en la red pueden fácilmente y con poco esfuerzo computacional, validar que el nodo que está propagando el bloque por la red, realmente llevó a cabo un esfuerzo computacional. La validación de la prueba de trabajo es tan sencilla como calcular el hash del bloque, es decir, tomar todos los datos dentro del *header* del bloque, concatenarlos y aplicar la función hash SHA256, el resultado deberá ser un número hexadecimal menor al objetivo (*target*) establecido por el sistema Bitcoin, si esto es así, quiere decir que el nodo llevó a cabo el trabajo computacional durante un

[60] Es importante recalcar que debido a la dificultad actual para generar la *proof of work,* el conjunto de valores que pueden ser generados por medio del *nonce,* cuyo tamaño es de 4 bytes, no son suficientes para satisfacer las condiciones establecidas por el *target,* por esta razón se utilizan adicionalmente los bytes disponibles en el *scriptPubKey* de la transacción *coinbase* (véase la sección 3.17), ya que la transacción *coinbase* no utiliza el atributo del *scriptPubKey* (Antonopoulos, 2017).

lapso determinado de tiempo hasta encontrar el *nonce* adecuado, en caso contrario, el bloque es descartado por los nodos.

De acuerdo con lo expuesto, podemos definir la probabilidad de generar un hash cuya secuencia de bits o valor hexadecimal sea inferior al *target* establecido por el sistema. Una función hash SHA256 genera un resultado de 256 bits, esto quiere decir que existen hasta dos elevado a la potencia 256 de combinaciones posibles de unos y ceros, también podemos interpretar esto con una codificación hexadecimal donde se pueden representar hasta cuatro bits por cada dígito, esto es, un byte por cada dos dígitos hexadecimales, por lo que utilizando el sistema hexadecimal tendríamos 16 elevado a la potencia 64, esto correspondería a nuestro espacio muestral, ya que es el valor más grande que podemos representar en un hash de 256 bits.

$$Combinación\ máxima\ de\ bits\ para\ SHA256 = 2^{256}$$
$$1\ digito\ hexadecimal = 4\ bits$$
$$^{256}/_{4} = 64$$
$$Por\ lo\ tanto, 16^{64} = 2^{256}$$

De lo que se deduce, de acuerdo la teoría de la probabilidad:

$$Sea\ S = \{16^{64}\}\ el\ espacio\ muestral$$
$$Donde\ h = hash, t = target$$
$$Sea\ E = \{h < t\}\ un\ evento\ en\ S$$
$$P(E) = {}^{E}/_{S}$$

Con la fórmula descrita podemos calcular cuál es la probabilidad de que un nodo en la red Bitcoin sea capaz de generar un *hash* cuyo valor hexadecimal sea menor al *target* establecido, es decir, que tan probable es que un nodo pueda crear un bloque válido para que sea aceptado por los demás nodos en el sistema. Dada la fórmula de probabilidad de un evento sobre el espacio muestral, si por ejemplo, el target definido es el valor máximo posible, 16 elevado a la 64, el máximo valor que podrá tener el hash del bloque será de 16 elevado a la 64 menos uno, ya que nuestro evento consiste en que el hash generado sea menor al target, por lo tanto, si definimos el target como el máximo valor del espacio muestral, entonces el máximo valor que podrá alcanzar el hash para cumplir la condición del evento será el máximo valor en el espacio muestral menos uno. Entonces, al dividir el evento sobre nuestro espacio muestral que es 16 elevado a la 64 obtendremos un valor cercano a uno, 0.9999999[…], es decir, la probabilidad de obtener un hash con un valor inferior a un *target* cuyo valor es el máximo valor en el espacio muestral, es de más del 99% o cercana al 100%.

En este sentido, obtener un *hash* cuyo valor es menor al *target* cuando este toma el máximo valor posible en el espacio muestral, no representa ninguna dificultad, ya que la probabilidad de obtener el hash dado ese evento, que el hash sea menor al target, es con certeza, seguro. Sin embargo, para garantizar que sólo uno de los nodos de la red puede generar un bloque válido a la vez, es necesario disminuir el valor del *target*, ya que mientras más pequeño sea el *target*, la probabilidad del evento, encontrar un hash con un valor inferior al target, es menor. Esto no sólo garantiza el conceso entre los nodos en la red Bitcoin, también establece un importante y fundamental mecanismo de seguridad en todo el sistema. Debido a que todos los bloques se encuentran encadenados unos con otros a través del *hash* del bloque, en orden de alterar la integridad de la información contenida en cada uno de los bloques, será necesario generar un nuevo *hash* del bloque (véase las secciones, 3.7 y 3.13), esto es, será necesario calcular nuevamente un *hash* cuyo valor este por debajo del target, esto significa que un atacante que quiera modificar la historia de Bitcoin tiene que realizar el trabajo computacional de todos los bloques que quiera alterar, pero como estos están encadenados, también tendrá regenerar el *hash* de todos los bloques encadenados posteriores a cada uno de los bloques alterados, incluso cuando estos bloques encadenados no se hayan alterado, es decir, un atacante tendría regenerar el *hash* del bloque o bloques que quiere modificar más todos los bloques encadenados subsecuentes a estos.

Entre más antiguo sea el bloque, más trabajo computacional tendrá que llevar a cabo un atacante para regenerar toda la cadena de bloques, ya que un cambio en un bloque rompe el encadenamiento, porque el bloque siguiente en la cadena tiene una referencia hacia el bloque anterior, entre más “profundo” sea el bloque más trabajo computacional se requiere para alterarlo, esto vuelve prácticamente imposible alterar la información contenida en los bloques de Bitcoin una vez que estos han alcanzado cierta antigüedad o mejor dicho, que se encuentran cierta cantidad de bloques atrás con respecto al más reciente.

Para tener una referencia de que tan difícil, así como qué tantos recursos computacionales se requieren para encontrar un hash que cumple con los criterios del evento, es necesario proporcionar un ejemplo. La primera versión de Bitcoin tenía un *target* de 0x1d00ffff, ya que fue el primer *target* del sistema Bitcoin, ningún *target* posterior puede ser mayor a este, es decir, los *targets* posteriores deben ser menores para incrementar la dificultad de encontrar un *hash* que cumpla con las condiciones establecidas por el evento, debido a que se trata del primer target en la historia de Bitcoin, la dificultad de este primer target se estableció como dificultad uno, esto es, la dificultad para encontrar un hash durante el inicio del sistema Bitcoin era de uno.

En la sección 3.13 se revisaron los atributos contenidos dentro de un bloque, entre estos atributos se encuentra precisamente el *target*, que como ya se explicó, corresponde con el valor de referencia utilizado para determinar si el *hash* del bloque calculado es válido con respecto al trabajo computacional requerido. Sin embargo, el

atributo *target* no contiene directamente el valor que será usado como *target*, en cambio contiene una serie de bits que representan el *target*, esto es, a partir del contenido del target se calcula el valor real de referencia que será utilizado como *target* en el cálculo de la prueba de trabajo computacional.

Fórmula para el cálculo del valor *target* a partir del campo bits (*target*) en el *header* del bloque:

$$e = primeros\ 2\ nibbles\ del\ target$$
$$b = 6\ últimos\ nibbles\ del\ target$$
$$t = \ b * 2^{(8 * (e - 3))}$$

Utilizando la fórmula de la probabilidad de que ocurra el evento, en este caso, de encontrar un hash con un valor hexadecimal menor al *target*, dado el *target* en el bloque génesis, se calcula la probabilidad de este evento a continuación.

$$target\ bits = \ 0x1d00ffff$$
$$t = 0x00ffff * 2^{(8 * (0x1d - 3))}$$
$$t = \ 415029 * 2^{(8 * (41 - 3))}$$
$$t = 1.3526864067554E + 97$$
$$P(E) = {}^{t}/_{16^{64}}$$
$$P(E) = 2.3282709094019E - 10$$
$$P(E) \cong 0.000000023282709094019084053206\%$$

Como se puede observar en el cálculo anterior, las probabilidades de encontrar un *hash* cuyo valor sea menor al establecido en el atributo *target* son increíblemente bajas, esto significa que para encontrar dicho *hash*, se requerirá hacer millones de intentos antes de poder obtener un *hash* que cumpla con la restricciones del evento, por lo que un nodo en la red Bitcoin que quiera agregar un nuevo bloque en el sistema, tendrá que llevar a cabo un gran trabajo computacional en orden de obtener un bloque válido que sea aceptado por todos los nodos de la red Bitcoin.

Aunque los resultados de los cálculos anteriores muestran una probabilidad ínfima para obtener un *hash* válido, son incluso mucho más bajas las probabilidades actuales de generar el *hash* de un bloque dado el *target* que es utilizado por el sistema Bitcoin en el momento de escribir este documento. Para comprender la magnitud de la dificultad actual de generar un bloque válido en el sistema Bitcoin, es necesario realizar los cálculos correspondientes, al momento de escribir estas líneas, el *target* en el sistema Bitcoin es 0x171a213e.

$$target\ bits = 0x171a213e$$

$$t = 0x1a213e * 2^{(8*(0x17-3))}$$

$$t = 24192070 * 2^{(8*(35-3))}$$

$$t = 2.5027426328408E + 54$$

$$P(E) = {}^{t}/_{16^{64}}$$

$$P(E) = 2.1614107227234E - 23$$

$$P(E) \cong 0.00000000000000000000216141072272336 1\%$$

Como se puede apreciar, las probabilidades de obtener un hash que cumpla con las restricciones del evento definido por el sistema Bitcoin son por decir lo menos, minúsculas, por lo tanto, en orden de lograr este cometido, se requieren ya no calcular millones de *hashes* sino trillones[61] de *hashes* para poder encontrar un *hash* que satisfaga el evento, específicamente, se necesitan calcular trillones de *hashes* por segundo. En promedio, los nodos de Bitcoin son capaces de encontrar un *hash* que cumpla las condiciones del evento en aproximadamente diez minutos, de hecho, el sistema Bitcoin ajusta el *target* cada determinado tiempo para asegurar que los nodos en la red Bitcoin serán capaces de resolver el problema de encontrar el *hash* del bloque en un intervalo de tiempo de aproximadamente diez minutos.

Bitcoin ajusta el tamaño del *target* cada 2016 bloques, el sistema ajusta el *target* dependiendo de que tanto tiempo le toma a los nodos en la red encontrar el hash de un bloque, esta estimación de tiempo se hace con base a una referencia de dos semanas, cuando se alcanzan 2016 bloques el sistema Bitcoin determina si esos bloques fueron generados a una tasa aproximada de 10 minutos durante un periodo de dos semanas, si esta cantidad de bloques fueron generados en un lapso de tiempo mayor, el target es ajustado a un valor mucho más alto, es decir, se disminuye la dificultad de generar el evento, por otro lado, si los bloques fueron generados por los nodos en un periodo de tiempo inferior a las dos semanas, entonces el target es ajustado a un valor menor, es decir, se aumenta la dificultad para generar el evento. De esta manera se mantiene la estabilidad en el sistema, lo que garantiza una generación estable y predecible de bloques, en contraposición con una generación descontrolada de bloques que podría agotar demasiado pronto los suministros disponibles de bitcoin (véase la sección 3.17).

Para poner en contexto la dificultad de generar cierta cantidad de hashes por segundo, se utilizará la siguiente comparación, una computadora promedio con una CPU que cuente con una capacidad de procesamiento de

[61] Aquí se hace referencia a las cantidades numéricas de acuerdo con el sistema numérico de escala larga, por lo que un trillón significa un millón de billones (véase el apéndice A). Esto es importante porque en países de habla inglesa se utiliza un sistema numérico de escala corta, en ese sistema, el trillón del que hablamos corresponde a un quintillón.

digamos 2.4 Ghz, con dos núcleos, podría procesar aproximadamente 2MH/s, es decir, 2 Mega hashes por segundo o lo que es lo mismo, 2 millones de *hashes* por segundo, por lo que utilizando una CPU con mayor capacidad de procesamiento, incluso utilizando una computadora convencional, sería posible generar un *hash* con un valor menor al *target* establecido en el bloque génesis, o mejor dicho, en los primeros 2016 bloques en la historia de Bitcoin en un periodo de tiempo de diez minutos. Sin embargo, es imposible encontrar un *hash* con un valor inferior al *target* que se utiliza actualmente en el sistema Bitcoin haciendo uso de una computadora convencional, que sólo es capaz de generar millones de hashes por segundo, cuando el promedio actual de hashes por segundo requeridos para generar un hash por debajo del target es del orden de EH/s, es decir, de trillones de *hashes* por segundo. Incluso si se intentara generar dicho *hash* utilizando el *target* de algunos años posteriores al inicio de Bitcoin sería una tarea imposible si se utilizara una computadora convencional, se necesitarían años, y, aun así, jamás encontrar el *hash*.

Por estas razones, el sistema Bitcoin ha demostrado hasta el día de hoy, a casi una década desde su lanzamiento, ser un sistema impenetrable, la cantidad de esfuerzo computacional requerido para alterar la información contenida en este sistema, es colosal, no existe actualmente una forma en la cual los bloques de información de Bitcoin puedan ser alterados, debido a esto, se considera que la información contenida dentro del sistema Bitcoin es inmutable, ya que no puede ser modificada una vez que esta ha sido aceptada por todos los nodos en la red Bitcoin.

Debido a que los bloques que contienen las transacciones se encuentran enlazados y que cada bloque se encuentra respaldado por una prueba de trabajo computacional, entre más antiguo sea un bloque, más seguro es, esto quiere decir que entre más bloques se van encadenando sobre un bloque previamente generado y validado, las transacciones contenidas en ese bloque pueden ser consideradas inmutables y finales, no hay forma de que puedan ser revertidas.

Tal y como se explicó en las secciones anteriores, el sistema Bitcoin provee integridad sobre la información que procesa y almacena por medio de las funciones hash, autenticación y no repudio por medio de las firmas digitales, y a pesar de esto, estas características no pueden garantizarse si la información es fácilmente alterada de forma deliberada por un atacante, es por ello que la característica más importante en el sistema Bitcoin es la inmutabilidad de la información, a través de la inmutabilidad se garantizan todas las demás características del sistema, integridad, no repudio y autenticación. Además, la inmutabilidad de la información proporciona a Bitcoin la propiedad única de poder hacer pública toda la información contenida en los bloques, proporcionando transparencia de toda la información procesada por el sistema, lo que deriva en la posibilidad de poder auditar de forma sencilla las transacciones llevadas a cabo en el sistema.

3.16 Consumo de Energía

En la sección anterior se estudió el proceso a través del cual el sistema Bitcoin proporciona seguridad y consenso dentro de la red de nodos distribuidos que procesan las transacciones, este proceso llamado prueba de trabajo computacional, requiere actualmente el cálculo de trillones de *hashes* por segundo por parte de toda la red Bitcoin. Para poder alcanzar un nivel de procesamiento del orden de Exa *hashes* por segundo (EH/s), se necesitan miles de nodos especializados dedicados únicamente a validar transacciones, estos nodos especializados son dispositivos electrónicos diseñados con la única de finalidad de calcular la mayor cantidad de *hashes* en el menor tiempo posible (Martindale, 2018). Debido a que se requiere una gran cantidad de estos equipos especializados para poder calcular suficientes *hashes* por segundo, estos nodos son ahora operados por compañías dedicadas exclusivamente al procesamiento de transacciones (Genesis Mining, 2018). La tasa de *hashes* por segundo requerida en los últimos años en la red Bitcoin ha desplazado el uso individual de nodos especializados para procesar transacciones, ya no es una opción económicamente rentable para un solo individuo o incluso para un pequeño grupo de decenas de personas, esto ha derivado en la creación de compañías con gran capital, capaces de costear miles de equipos especializados para generar *hashes*, estos costos no sólo se generan por la compra de los equipos *per se*, también son consecuencia de la gran cantidad de energía eléctrica que se necesita para alimentarlos (Cambridge Centre for Alternative Finance, 2019), la refrigeración para mantenerlos a una temperatura adecuada y grandes extensiones de tierra necesarias para construir las instalaciones necesarias para operar estos nodos procesadores de transacciones Bitcoin (Motherboard, 2015) (CBS News, 2018).

Por estas razones, actualmente los nodos que procesan transacciones Bitcoin se encuentran en grandes instalaciones con todos los requerimientos necesarios para hacer uso de este hardware especializado, uno de estos requerimientos es la refrigeración, para evitar que estos dispositivos se derritan, literalmente. También requieren una instalación eléctrica de tipo industrial para poder satisfacer el apetito voraz de los miles de nodos Bitcoin, así como una gran cantidad de espacio que permita contenerlos. A este tipo de instalaciones de nodos procesadores de transacciones Bitcoin se les conoce con el nombre de granjas (*farms*).

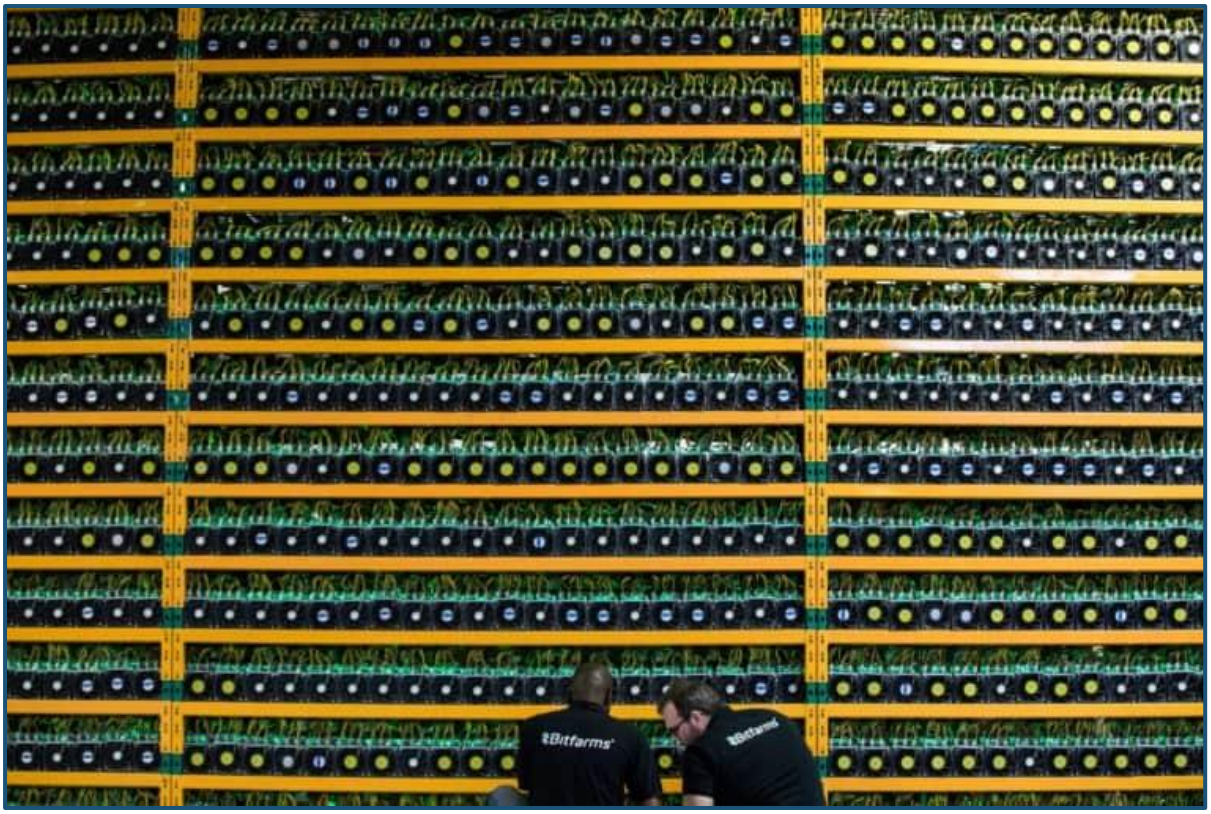

Figura 22. Granja de nodos Bitcoin. Recuperada de *What happened to the cryptocurrency craze?*. 2018 por CoinNews Telegraph.

Para dimensionar la cantidad de energía eléctrica necesaria para satisfacer las granjas de nodos procesadores de transacciones Bitcoin, es necesario establecer algún punto de referencia, por ello, el punto de referencia que será utilizado es el consumo eléctrico de algunas de las principales naciones alrededor del mundo. El consumo de la energía eléctrica se mide en Watts, esta unidad de medida representa la cantidad de energía eléctrica consumida o generada. El consumo o generación de energía eléctrica se mide sobre la unidad de tiempo, hora, esto es, la electricidad se mide en Watts hora, la cantidad de energía eléctrica consumida o generada en el lapso de una hora expresada en Watts (U.S. Energy Information Administration, s.f.). Adicionalmente, como en cualquier unidad de medida, los Watts son expresados utilizando los prefijos internacionales para las unidades de medida, por ejemplo, Kilo, Mega o Giga (véase el apéndice A), con la finalidad de representar unidades del orden de miles, millones o miles de millones, por medio de una nomenclatura estándar.

Durante el año 2018 el sistema Bitcoin en su totalidad, esto es, todos los nodos en la red Bitcoin que realizan la prueba de trabajo computacional en orden de crear nuevos bloques y obtener una recompensa en Bitcoins (véase la sección 3.17), consumieron un aproximado de más de 22 TWh por año, que corresponde a unos 22 billones de Watt hora por año, un consumo eléctrico superior al de toda la nación irlandesa durante el mismo

periodo de tiempo (G.F., 2018). Mientras que durante el año 2019 la red Bitcoin está consumiendo aproximadamente, más de 65 TWh, esto representa un consumo de electricidad superior al de naciones como Israel, Portugal, Grecia o Suiza (Vincent, 2019) (Cambridge Centre for Alternative Finance, 2019).

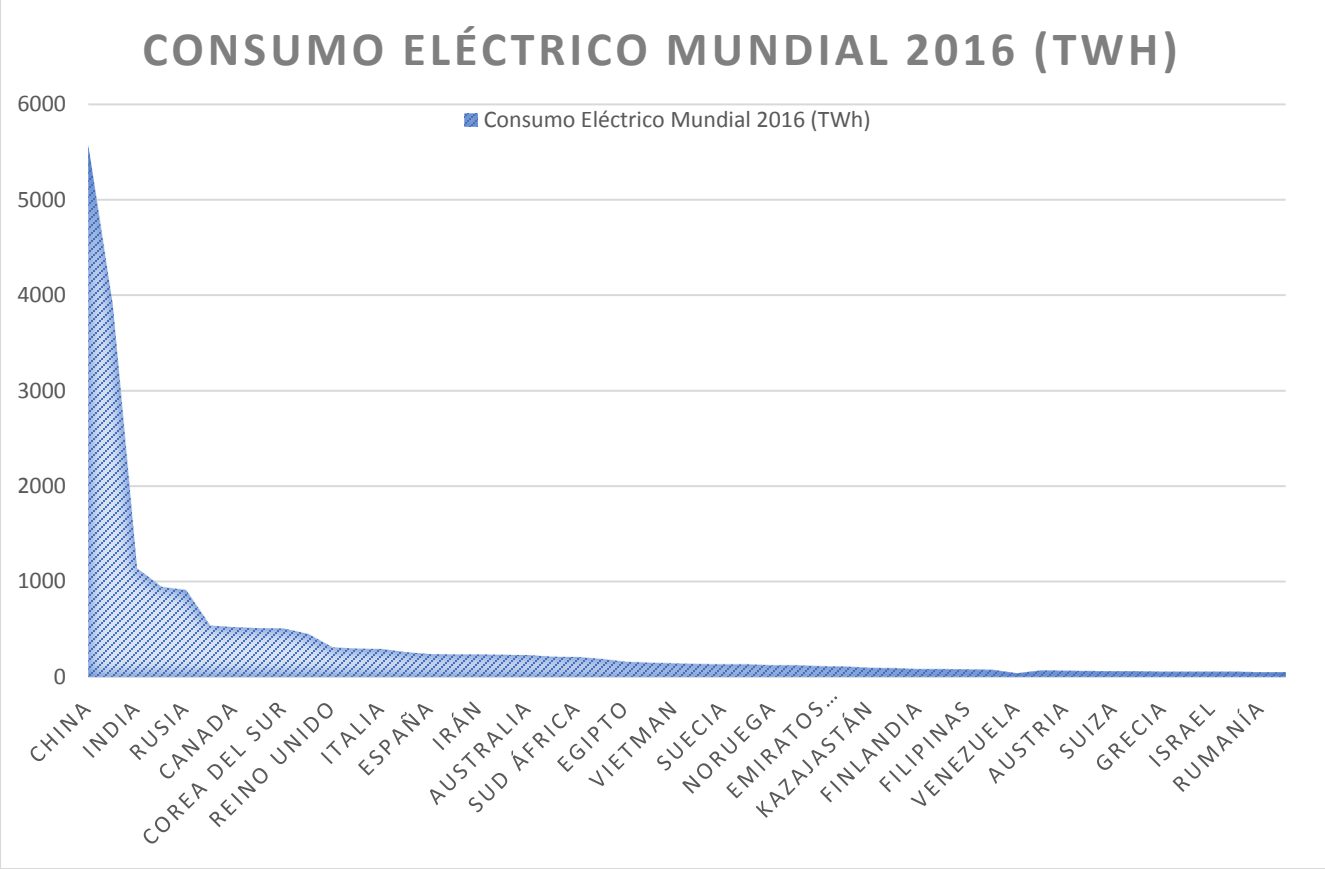

Figura 23. Consumo de electricidad a nivel mundial durante el año 2016. Datos obtenidos de *The World Fact Book* (CIA, 2016).

Los mayores consumidores de electricidad en el mundo son China, Estados Unidos y la India, en ese orden, cada uno de ellos consume electricidad en el orden de los miles de Tera Watt hora, esto es, billones de Watts hora por año. Este consumo eléctrico está muy por encima del consumo eléctrico de la red Bitcoin, sin embargo, el consumo eléctrico de este sistema está muy lejos de ser ínfimo, como ya se expuso en el párrafo anterior, el consumo eléctrico del sistema Bitcoin está por encima de varias naciones, de hecho, su consumo eléctrico sólo está por debajo de las primeras 40 naciones que más electricidad consumen en el mundo.

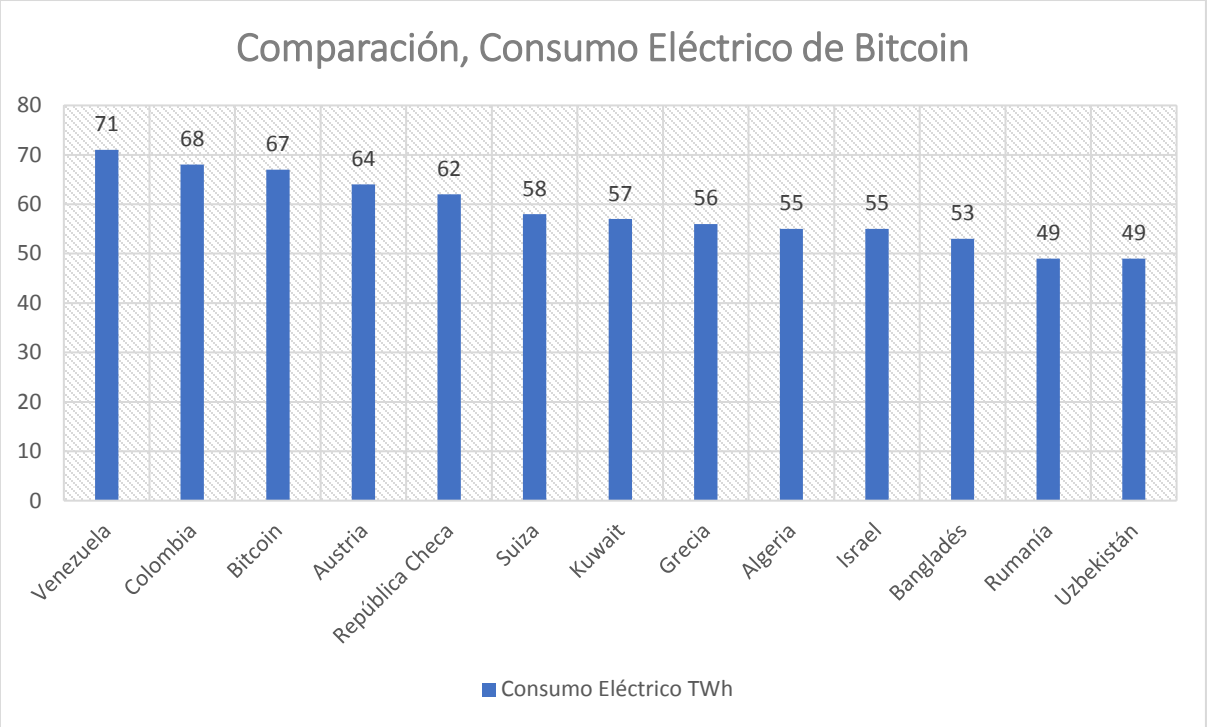

Figura 24. Comparación del consumo eléctrico de Bitcoin con respecto a varias naciones. Datos obtenidos de *The World Fact Book* (CIA, 2016) y de *Cambridge Bitcoin Electricity Consumption Index* (Cambridge Centre for Alternative Finance, 2019).

En la figura 24 se muestra una comparación del consumo de electricidad en TWh de Bitcoin con respecto a diferentes países cuyo consumo eléctrico es similar, esta comparación se realiza haciendo una extrapolación del consumo promedio de los países durante el año 2016, contra el consumo promedio del año 2019 del sistema Bitcoin. Como se aprecia en la imagen, el consumo eléctrico por parte del sistema Bitcoin no es trivial, consume una cantidad de energía superior al de varios países, y se encuentra sólo por debajo de Venezuela y Colombia.

3.17 Bitcoins

En esta última sección se abordan los últimos componentes que forman parte del sistema Bitcoin. Ya se expuso en secciones anteriores cómo el sistema Bitcoin procesa las transacciones de todos los usuarios, pero la cuestión que no se ha planteado hasta ahora es de dónde provienen los bitcoins, es decir, las monedas electrónicas que se utilizan para transferir valor de forma distribuida entre pares. Como ya se explicó anteriormente, los bitcoins no existen explícitamente, más bien se trata del flujo de entradas y salidas, es precisamente en las salidas dónde se generan los bitcoins, esto sucede cada vez que un nodo en la red genera

el *hash* de un bloque válido, es decir, cuando un nodo es capaz de calcular el *hash* de un nuevo bloque cuyo valor se encuentra por debajo del *target*, el nodo es recompensado por el trabajo computacional que realizó en orden de encontrar el *hash*. Este proceso puede verse de forma análoga al procedimiento llevado a cabo por los bancos centrales en una nación, cada banco central es el encargado de emitir el dinero, es decir, de crear y distribuir las monedas y billetes, en el caso del sistema Bitcoin, es el propio sistema el que genera nuevos bitcoins como recompensa al nodo que logre generar un bloque válido. Debido a que todo el valor en Bitcoin se representa por medio de entradas y salidas a través de transacciones, la recompensa otorgada a un nodo cuando este encuentra un hash válido para un bloque es por medio de una transacción, concretamente, es el propio nodo quien especifica por adelantado como primera transacción de cada bloque nuevo, una transacción correspondiente con la cantidad de bitcoins que en ese momento el sistema está otorgando a los nodos como recompensa por generar nuevos bloques válidos.

Las transacciones que generan nuevos bitcoins son transacciones especiales que se conoce con el nombre de *coinbase* (Antonopoulos, 2017), estas transacciones siempre deben ser las primeras dentro de un bloque, sólo se permite una transacción *coinbase* por bloque, esta es la única transacción en el sistema que no cuenta con una entrada que justifique la transferencia de valor, porque se trata de la creación de nuevos bitcoins, así que no hay una entrada previa, pero si una salida para transacción, la salida corresponderá con la llave pública del nodo que generó el bloque.

A menudo se piensa en la falta de valor de los bitcoins, sobre todo porque no tienen un respaldo económico como en el caso del dinero fíat, que se encuentra respaldado por algún metal precioso como el oro o por más dinero fíat, como pueden ser los dólares. Sin embargo, el valor del bitcoin se respalda en toda la energía eléctrica y trabajo computacional que se ha gastado para encontrar un nuevo *hash* de bloque válido, los recursos computacionales invertidos en la generación de un nuevo bloque que contiene transacciones son el respaldo económico sobre la moneda electrónica bitcoin.

Un nodo en la red Bitcoin recibe una cantidad determinada de bitcoins cada vez que genera un bloque válido, por supuesto, como ya se explicó, para que el bloque sea válido además de que todas las transacciones contenidas deben ser válidas, el bloque debe contener dentro de su *header* el atributo *nonce* como evidencia de la prueba de trabajo. La cantidad de bitcoins inicial que proporcionaba el sistema como recompensa eran 50 bitcoins por cada bloque válido generado. El sistema ajusta la cantidad de bitcoins generados por la creación de cada nuevo bloque válido en un intervalo correspondiente a 210 mil bloques con un 50% del valor actual (Hughes, 2017), es decir, después de los primeros 210 mil bloques en la historia del sistema, la recompensa por generar un nuevo bloque en Bitcoin se redujo a la mitad, esto es, a 25 bitcoins por bloque. Actualmente la

recompensa por generar un nuevo bloque es de 12.5 bitcoins, cuando se alcancen nuevamente 210 mil bloques, se volverá a reducir hasta la mitad la cantidad de bitcoins que son creados como recompensa por cada nuevo bloque generado.

De esta forma, la primera vez que se generó un bloque en el sistema Bitcoin, el bloque génesis, se crearon los primeros 50 bitcoins en la historia, el nodo que generó este bloque fue Nakamoto. Este nivel de recompensa de 50 bitcoins permaneció constante hasta que se superaron los 210 mil bloques. Como se puede observar, cada vez que se agrega un bloque válido en el sistema Bitcoin, se crea una nueva cantidad específica de monedas electrónicas o bitcoins que pueden ser utilizados, por esta razón, no hay forma de crear de forma deliberada más bitcoins de la "nada", el sistema es el único que puede generar más "dinero", pero sólo cuando se ha realizado una cierta cantidad de trabajo computacional (*proof of work*), esto es, cuando se han consumido cierta cantidad de recursos. Estas reglas acerca del control de suministro de bitcoins, forman parte de todo el sistema, esto es, se encuentran programadas dentro del sistema de forma permanente, así mismo, existe un límite de bitcoins que pueden ser generados, este suministro máximo de monedas electrónicas que el sistema puede generar es de 21 millones de bitcoins (Hughes, 2017), una vez alcanzado el máximo número de bitcoins en circulación, no será posible crear más bitcoins, aunque esto podría suponer que el sistema dejará de ser inservible, esto no es así, ya que a pesar de que no se suministrarán más bitcoins cuando se alcance el límite, los 21 millones de bitcoins que se encuentren en circulación podrán ser utilizados de forma indefinida en el sistema.

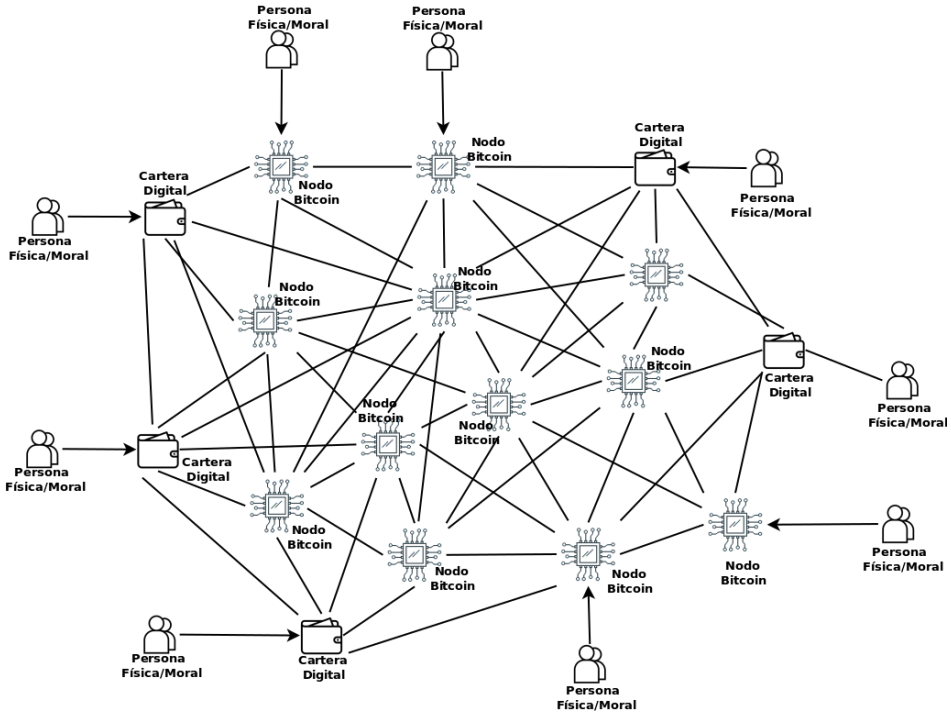

Figura 25. Red Bitcoin.

En la figura 25 se muestra la estructura de la red Bitcoin, como se aprecia en la imagen, los usuarios del sistema pueden ser tanto personas físicas como morales, los usuarios generalmente utilizan el sistema Bitcoin a través de una aplicación de cartera digital (*digital wallet*), aunque también es posible hacer uso del sistema a través de un nodo Bitcoin que procesa transacciones.

Es importante aclarar que cualquier dispositivo o computadora que se conecte en la red Bitcoin es considerado un nodo, en este sentido, las carteras digitales también son nodos dentro del sistema Bitcoin, sin embargo, estas aplicaciones de cartera digital generalmente se ejecutan en un dispositivo móvil, razón por la cual no son capaces de llevar a cabo el procesamiento de las transacciones, no cuenta no la suficiente capacidad de almacenamiento y procesamiento, aun así, pueden crear transacciones para que el dueño de la cartera transfiera valor entre los distintos usuario del sistema. Adicionalmente, la cartera digital puede verificar que las transacciones y los bloques sean válidos de forma parcial, pero no tiene la capacidad de cómputo para generar un bloque y llevar a cabo la prueba de trabajo computacional, puede verificarla pero no crearla, por lo que las carteras están limitadas a validar las transacciones del usuario y presentar a este la información de las transacciones de forma amigable, es decir, mostrar al usuario un balance de sus transacciones así como su saldo, aunque como ya se explicó en secciones anteriores, ninguno de estos datos existe dentro las transacciones procesadas por el sistema.

Con el sistema Bitcoin es posible llevar a cabo transferencias de fondos entre personas físicas o morales prácticamente desde cualquier parte del mundo, sólo se necesitan una aplicación de cartera digital y una conexión a Internet, como ya se explicó, la transacción se hace efectiva aproximadamente en diez minutos, de esta forma, por ejemplo, sería posible enviar bitcoins entre una persona que trabaja en Estados Unidos hacia una persona en México en tan sólo diez minutos con una cuota muy baja por llevar a cabo la transacción. Así mismo, una persona en México que no cuenta con una cuenta bancaría podría fácilmente descargar en su teléfono móvil una aplicación de cartera digital y hacer uso del sistema Bitcoin de forma inmediata, por supuesto, necesitaría que alguien le transfiriera algunos bitcoins o satoshis para que pudiera realizar alguna transacción.

Es importante recalcar que las transacciones en Bitcoin son inmediatas, sin embargo, estas no son finales hasta el momento en que un nodo en la red Bitcoin crea un bloque que incluye la transacción y genera el *hash* con un valor por debajo del *target*, como ya detalló, este proceso toma aproximadamente diez minutos, por lo que se podría decir que una transacción toma aproximadamente diez minutos para considerarse válida, aun así, un usuario en la red Bitcoin puede que tenga que esperar todavía más tiempo para confirmar que su transacción no será rechazada, por ejemplo, digamos que dos nodos llegaran a encontrar un hash de bloque válido al mismo tiempo, en este escenario, dos nodos diferentes propagarían un bloque válido pero diferente por la red,

posiblemente con transacciones diferentes, esto generaría una bifurcación temporal en la historia de Bitcoin, es decir, se crearían de manera temporal dos "cadenas" distintas de bloques con un solo bloque diferente, el último. Bajo este escenario, en los siguientes minutos del procesamiento natural del sistema Bitcoin, más bloques se seguirán generando y cada uno de ellos sería agregado a una versión diferente de la historia de Bitcoin dependiendo de qué nodo generó el bloque. Debido a que existe una bifurcación en la historia de Bitcoin, una parte de los nodos en la red utilizaría una de las cadenas de bloques, llamémosla cadena de bloques A, y la otra parte de nodos utilizaría la cadena de bloques B, cómo decidieron qué cadena, simplemente no lo hacen, utilizan la cadena de bloques más cercana, esto es, los nodos vecinos con cuales tienen conexiones, son los que determinan con cuál de las cadenas se queda cada nodo.

Continuando con el ejemplo de dos nodos generando un bloque válido al mismo tiempo en diferentes regiones de la red, en algún punto del tiempo, digamos, después de tres o cuatro transacciones, esto es, unos 30 o 40 minutos después, la cadena de bloques de Bitcoin, es decir, toda la historia de transacciones se unificaría nuevamente. Esto sucede porque eventualmente una de las dos cadenas de bloques, la cadena A o la cadena B, sería más grande que la otra, en este punto, todos los nodos de la red detectarían que existe una cadena de bloques más larga, por lo que cambiarían y utilizarían la cadena más larga, cualquiera que está fuera, la cadena A o la cadena B. Aunque todo esto parece un error dentro del sistema, no lo es, ya que el sistema es capaz de unificar de manera automática una bifurcación en la historia de las transacciones, además, debido a que se trata de una red *peer to peer*, cuando un nodo propaga una transacción de un usuario, esta es recibida por la mayoría de los nodos, también es mantenida por los nodos en una región temporal de memoria hasta que es procesada. Si por alguna razón durante la bifurcación o posterior unificación de la cadena, alguna transacción no fue incluida, esta transacción sería incluida en el siguiente bloque, aunque lo más probable es que la transacción haya sido incluida en alguno de los bloques de ambas cadenas durante la bifurcación, por lo que no importa cuál de las dos cadenas resulte al final la ganadora, las transacciones se encuentran en ambas cadenas de bloques.

Debido a lo anterior, dependiendo de la cantidad de valor que se transfiere en el sistema Bitcoin se puede esperar más o menos bloques como un mecanismo de confirmación antes de dar como final la transacción, por ejemplo, en una transacción de gran valor, digamos, de algunos bitcoins, sería prudente esperar la generación de varios bloques posteriores al bloque que contiene la transacción realizada, es decir, esperar la confirmación de varios bloques antes de estar seguros que la transacción no será cancelada. Se dice que una confirmación de bloques después de realizada una transacción se puede considerar definitivamente como final e inmutable sólo después de una confirmación de 100 bloques, de igual forma un nodo que ha sido recompensado con n

bitcoins por generar un nuevo bloque sólo podrá utilizar sus bitcoins después de 100 bloques (Antonopoulos, 2017).

El último elemento por discutir del sistema Bitcoin son las cuotas (*fees*) que son aplicadas a cada transacción (Antonopoulos, 2017), es decir, el costo asociado por llevar a cabo una transacción sobre la red Bitcoin. A primera vista esto se puede percibir como una desventaja, sin embargo, es una característica fundamental de cualquier sistema de transferencia de valor, incluso los sistemas tradicionales de pago estudiados en el capítulo uno son sujetos de costos operativos y administrativos que son transferidos inevitablemente hacia los usuarios. La diferencia con el sistema Bitcoin estriba en el pago único de cuota por transacción, esto es, todas las transacciones pueden tener la misma cuota sin importar si trata de una transferencia de fondos de alto o bajo valor, más importante aún, la cuotas no están definidas por la localización geográfica en la que ocurre la transacción, ni mucho menos el origen y destino, con lo cual es posible llevar a cabo transferencias de alto valor entre dos usuarios situados en dos puntos geográficos completamente opuestos de la tierra, por ejemplo, realizar una transferencia de fondos entre una entidad en México y la contraparte que recibe la transferencia situada en algún país de Europa o Asia, sin ningún cargo adicional por llevar a cabo una transacción adicional o algún retraso de días o incluso semanas por tratarse de un pago que tiene que pasar por una gran cantidad de sistemas de pago intermedios y redes privadas que no sólo retrasan la transacción sino que también aumentan su costo.

Estas cuotas no son obligatorias y su valor puede ser definido de forma personalizada por cada usuario especifico que realiza una transacción en el sistema Bitcoin, sin embargo, en orden de asegurarse que la transacción sea realmente procesada por los nodos de la red Bitcoin es necesario incluir una cuota dentro de cada una de las transacciones que se llevan cabo. Recuerde el lector (véase la sección 3.15) que los nodos tienen que llevar a cabo trabajo computacional para poder generar un nuevo bloque que contenga nuevas transacciones, este trabajo computacional incurre en costos asociados no sólo a los equipos de cómputo especializados[62] para llevar a cabo la *proof-of-work*, sino también por toda la electricidad utilizada (véase la sección 3.16) para alimentar todos estos equipos de alto rendimiento que computan, cada uno, decenas de billones de hashes por segundo. Adicionalmente se requieren grandes espacios para almacenar todos estos equipos, no porque tengan un gran tamaño, más bien por la gran cantidad que de estos equipos se requiere

[62] Véase la publicación: "*What is an ASIC miner?*" (Martindale, 2018).

para acercarse a la tasa actual requerida de cálculo de hash de bloque, que corresponde al orden de trillones de hashes por segundo[63].

Es importante aclarar que la cuota corre a cuenta de quien realiza la transacción, es decir, de quien lleva a cabo la transferencia de valor, el beneficiario de la transacción queda exento de cualquier pago. Esta cuota provee de un incentivo adicional para los nodos que se encargan de procesar las transacciones en el sistema Bitcoin. Esto también asegura que los usuarios dentro del sistema realizarán únicamente transacciones fidedignas, es decir, esta cuota asociada a cada transacción previene ataques DDoS[64] dentro del sistema contra transacciones generadas por algún atacante que trata de saturar el sistema. Cada usuario que desee participar en el sistema Bitcoin debe incurrir en un costo, ya sea para realizar una transacción o para crear bloques con transacciones válidas, esto asegura de cierta forma que todos los participantes en el sistema interactúan siguiendo las reglas del sistema.

Hasta este punto se han revisado los componentes principales del sistema Bitcoin, así como cuál es su funcionamiento, con esto es posible tener un panorama general de como este sistema proporciona seguridad y valor a sus usuarios permitiendo la transferencia de fondos de forma segura y programable, desde casi cualquier parte del mundo, en una fracción de tiempo muy pequeña en comparación con los sistemas de pago tradicionales, a través de una red distribuida de nodos que validan las transacciones a cambio de una recompensa económica expresada en la moneda virtual expedida por este sistema, bitcoins. También se estudió porque este sistema es tan seguro, el punto central de esta seguridad es la prueba de trabajo computacional llevada a cabo por cada uno de los nodos que compiten por crear un nuevo bloque válido para poder obtener su recompensa económica.

[63] De acuerdo con las estadísticas más recientes sobre el trabajo computacional requerido para genera un nuevo bloque en Bitcoin, véase las estadísticas publicadas por Saint Bitts (Saint Bitts LLC, 2019).

[64] *Distributed Denial of Service* es un ataque informático que consiste en realizar un inmenso número de peticiones hacia un sistema o servicio dentro de una red informática, saturando al sistema por la gran cantidad de solicitudes, lo que deja al sistema inutilizable de manera temporal. Este ataque es una variante del ataque DoS (*Denial of Service*), la diferencia estriba en que DDoS hace uso de un gran número de dispositivos o computadoras distribuidos en una red informática, generalmente estos equipos se encuentran comprometidos por algún malware que los obliga a realizar el ataque (Dulaney & Easttom, 2018).

Capítulo IV. Metodología de Estudio

En este brevísimo capítulo, se presentan los mecanismos metodológicos que fueron utilizados para llevar a cabo la presente obra.

El presente trabajo es en su mayoría un trabajo de investigación documental, porque se concentra en la recopilación de información documental de diferentes tipos, tanto impresos como digitales, que son presentados como sustento, evidencia del fenómeno estudiado, así como fuente de información para su posterior análisis e interpretación. También es un trabajo de orden descriptivo, ya que a través de la narración y de imágenes se presenta el funcionamiento del objeto de estudio, así como sus características. Finalmente, este trabajo hace uso de un enfoque principalmente cualitativo, ya que la investigación se fundamenta en el estudio y análisis de la información documental recopilada, realizando una interpretación y descripción de los hechos a partir de esta información, tomando en cuenta lo anterior, es importante señalar que dentro del trabajo se incluyen algunos datos de orden cuantitativo, como datos estadísticos (véase las secciones 1.3.10 y 3.16), así como datos probabilísticos derivados del análisis realizado sobre la información documental, estos datos pueden ser verificados con sus correspondientes fórmulas matemáticas (véase la sección 3.15), de igual manera se presenta información extraída directamente del objeto de estudio, estos datos de igual manera pueden ser verificados junto con el procedimiento por el cual fueron obtenidos (véase las secciones 3.9, 3.13 y el apéndice B).

Capítulo V. Resultados

Del análisis e interpretación de la información recolectada y presentada en el presente trabajo de investigación, se obtienen los siguientes resultados. Bitcoin es un sistema informático capaz de proveer de inmutabilidad sobre la información que es procesada por este, esta capacidad de inmutabilidad sobre la información es la característica más sobresaliente del sistema, ya que a partir de este elemento se desprenden las ventajas del sistema Bitcoin, por ejemplo, la integridad de la información, es decir, la capacidad de mantener la información en su estado original sin que esta pueda ser alterada de forma accidental o deliberada por un atacante informático, esta característica se basa sobre la inmutabilidad porque a través de esta última existe certeza de que la información permanece inalterable, adicionalmente, esta integridad sobre la información es fácilmente verificable por medio de los mensajes de digestión que se utilizan para representar de manera única la información dentro del sistema (véase las secciones 3.9 y 3.13).

Otra característica del sistema Bitcoin fundamentada sobre la inmutabilidad de la información es la transparencia, debido a que la información una vez procesada y validada (véase la secciones 3.13 y 3.15) no puede ser alterada, esta puede publicarse de forma abierta para ser accedida por cualquier entidad que requiera auditar de forma independiente la veracidad y validez de las transacciones llevadas a cabo, con lo cual se facilita la aplicación de la fiscalización sobre transacciones financieras al ser estas públicas.

El mecanismo por medio del cual se provee de inmutabilidad a toda la información almacenada por el sistema Bitcoin es la prueba de trabajo computacional (véase la sección 3.15), este mecanismo representa el proceso medular sobre el cual se cimienta toda la seguridad del sistema, es decir, a través de este proceso el sistema proporciona su característica más importante, la inmutabilidad. Si se elimina la prueba de trabajo computacional de la ecuación, el sistema Bitcoin no posee ninguna ventaja o característica sobresaliente con respecto a otros sistemas de pago existentes, de hecho, se vuelve trivial el uso de un sistema público y distribuido para realizar transacciones financieras si este no puede garantizar seguridad sobre las transacciones. Sin la prueba de trabajo computacional, cualquier nodo o atacante dentro de la red Bitcoin puede alterar de forma indiscriminada la información, lo que derivaría en problemas de *double spending* (véase la sección 1.8), alteración en las transacciones o simplemente la eliminación de la información.

La siguiente característica fundamental en el sistema Bitcoin es su arquitectura distribuida (véase la sección 3.8), ya que a través de esta es posible llevar a cabo transacciones alrededor del mundo a un costo menor[65] y

[65] Costo relativamente menor (hasta el momento) para los usuarios que realizan las transacciones, no para aquellos que poseen las granjas de nodos Bitcoin (véase la sección 3.16).

mucho más rápido, con respecto a los sistemas tradicionales de pago, concretamente en pagos internacionales, ya que en las transacciones internacionales las instituciones financieras hacen uso de una gran cantidad de sistemas intermedios, así como redes privadas (véase las secciones 1.3.9 y 1.9) que incrementan los costos y tiempos en la liquidación de las transacciones financieras. Mientras que la red de Bitcoin hace uso de la red mundial Internet para la transmisión de sus transacciones, así como de nodos que se conectan en la red desde diferentes localizaciones geográficas. Esta característica de distribución por parte del sistema Bitcoin, sólo puede implementarse si se garantiza la inmutabilidad y un mecanismo de consenso por parte del sistema, estos dos elementos se logran a través de la prueba de trabajo computacional.

La arquitectura distribuida de Bitcoin requiere de un mecanismo de consenso por parte de todos los nodos en la red, de manera que exista congruencia y uniformidad en la información almacenada y procesada, esto se logra también por medio de la prueba de trabajo computacional, sin esta, no existe una forma de garantizar un consenso uniforme y no sesgado, ya que este proceso se basa en la probabilidad (véase la sección 3.15), por lo que no hay forma de que un nodo o una parcela de la red tenga el control[66] en el procesamiento y validación de las transacciones.

De acuerdo con lo anterior, se deduce que el componente tecnológico central en el sistema Bitcoin es la prueba de trabajo computacional (véase la sección 3.15), y a partir de este proceso, el sistema puede obtener las características únicas que lo diferencian y que proporcionan una ventaja sobre los sistemas financieros convencionales. Dicho esto, tal y como se revisó en las secciones 3.13 y 3.14, los bloques y su encadenamiento, simplemente son estructuras lógicas que permiten empaquetar y organizar la información para facilitar su almacenamiento y procesamiento, sin embargo, estos elementos no proporcionan ningún mecanismo de seguridad, por lo que el uso de estas estructuras de datos no ofrece ninguna utilidad relevante si se desprenden de la prueba de trabajo computacional. Por lo que no hay razón alguna para atribuir características inexistentes sobre esta simple estructura de datos comúnmente llamada "*blockchain*".

La prueba de trabajo computacional, es decir, el componente medular del sistema Bitcoin, es realizado a cambio de un consumo descomunal de energía eléctrica cada año (véase la sección 3.16) por parte del sistema, superando a naciones enteras alrededor del mundo en cuanto a consumo eléctrico, esto deriva en altos costos para quienes operan estas granjas de nodos Bitcoin que se encargan de validar las transacciones. Este consumo es producto del mecanismo a través del cual el sistema garantiza la seguridad en las transacciones (véase la

[66] La prueba computacional en el sistema Bitcoin se basa en la probabilidad, sin embargo, contar con un mayor número de nodos que calculen hashes, con respecto a toda la red Bitcoin, proporciona al dueño de estos nodos mayor probabilidad de encontrar el hash que cumple con el evento (véase la sección 3.15) y de esta manera generar un nuevo bloque en el sistema y al mismo tiempo quedarse con la recompensa (véase la sección 3.17).

sección 3.15), esto obliga a la red Bitcoin a utilizar cada vez más nodos con mayor poder computacional en orden de poder encontrar un hash de bloque válido, cuya dificultad para encontrarlo, se incrementa cada vez más con el tiempo, lo que significa que constantemente se requiere incrementar el número de nodos en la red. A su vez, el aumento de nodos requiere no sólo de más electricidad que los alimente, también se requiere de energía eléctrica y hardware adicional que mantenga estos equipos especializados refrigerados, lo que implica un costo adicional, sin contar la gran cantidad de espacio físico requerido para alojar todos esos equipos computacionales especializados.

Características del Sistema Bitcoin	
Consumo eléctrico (variable)	≈ 67 TWh
Bloques máximos procesados	1 bloque cada 10 minutos
Tiempo de liquidación de transacción	≈ 10 minutos
Tamaño máximo por bloque	1 MiB
Tamaño de transacción (variable)	≈ 256 bytes
Transacciones máximas por bloque (variable)	≈ 4096
Cantidad máxima de bitcoins por transacción	36,893,488,147
Tipo de cambio BTC / USD (variable)	10,278 USD
Unidad monetaria	bitcoin (BTC)
Unidad monetaria mínima	satoshi
Equivalencia bitcoin / satoshi	1 BTC = 100,000,000 satoshis
Cuota por byte en la transacción (variable)	≈ 81 satoshis
Cuota por transacción en satoshis (variable)	≈ 20,736 satoshis
Cuota por transacción en USD (variable)	≈ 2.131246 USD
Restricción geográfica	Sin restricción, transacciones internacionales
Saldo mínimo	Sin saldo mínimo
Crédito	No existe el crédito dentro del sistema
Costo por usar el sistema	Sin costo, excepto por la cuota de las transacciones
Uso	Personas físicas y morales
Propiedad	Público, el sistema es de libre acceso, es mantenido por la red de nodos que validan las transacciones, pero no existe un individuo o corporación que sea dueño del sistema.

Tabla 7. Características del sistema Bitcoin.

En la tabla de la parte superior se muestran algunas de las características relevantes del sistema Bitcoin, es importante aclarar algunos de los valores presentados en la tabla, en el caso del tiempo de liquidación, se especifica como variable debido a que aproximadamente cada 10 minutos se validan bloques, sin embargo, no hay garantía de que una transacción específica sea incluida inmediatamente en el siguiente bloque, sobre todo si la cuota asignada en la transacción es muy baja. Por otro lado, como ya se revisó en la sección 3.17, la transacción aún podría ser revocada si existe una bifurcación (algo poco probable) en la cadena de bloques, aunado a esto, dependiendo del valor transferido se requerirá mayor o menor certeza de la inmutabilidad de la transacción, lo que implica un mayor tiempo de espera (véase la sección 3.17).

El tamaño de la transacción en bytes presentado en la tabla corresponde al tamaño promedio del tipo de transacción más usada en Bitcoin, esto es, transacciones de tipo P2PKH (véase la sección 3.11), que representan el 89% de todas las transacciones por bloque (TradeBlock, 2015). Tomando en cuenta lo anterior, el número de transacciones por bloque se calculó asumiendo que el 100% de las transacciones tienen un tamaño de 256 bytes, por lo que el valor presentado de transacciones máximas por bloque es sólo una aproximación.

En el caso del tipo de cambio, este es variable, al momento de escribir estas líneas el tipo de cambio es el que se presenta en la tabla, con este dato de tipo de cambio a más de 10 mil dólares por bitcoin, se calcularon los costos en dólares por realizar una transacción, tomando en cuenta una tasa de 81 satoshis (Earn bitcoin, 2019) por byte de transacción y usando un tamaño estimado de transacción de 256 bytes. Es importante recalcar que estos últimos datos son los de mayor variabilidad y dependen del tipo de cambio, así como de la cuota asignada de manera particular por cada usuario a cada transacción realizada, por lo que mayores o menores cuotas se traducen en mayor o menor costo por realizar una transacción.

Conclusión

Durante el desarrollo del presente trabajo fue posible establecer qué es y cómo funciona el sistema Bitcoin, concretamente se examinaron los principales componentes tecnológicos por medio de los cuales el sistema Bitcoin provee sus características más importantes, lo que permitió determinar la factibilidad de la aplicación de estos componentes tecnológicos sobre los sistemas informáticos que conforman el sistema financiero mexicano, esto con la finalidad de mejorar principalmente la seguridad en las transacciones financieras mexicanas. De acuerdo a lo expuesto en el capítulo número cinco, la aplicación de la tecnología utilizada por el sistema Bitcoin dentro de los sistemas financieros mexicanos no es una alternativa viable, ni mucho menos práctica, en términos económicos e informáticos, los costos asociados en la implementación de un mecanismo similar al sistema Bitcoin serían exorbitantes, por otro lado, la eficiencia de los sistemas financieros mexicanos se vería mermada dramáticamente y finalmente requeriría un enorme esfuerzo técnico, de colaboración y coordinación entre todos las instituciones financieras que participan en el sistema financiero mexicano, por estas razones no hay manera de que este tipo de tecnología, tal y como se encuentra en este momento, pueda ser implementada dentro del sistema financiero mexicano, los costos y el esfuerzo rebasan por mucho los beneficios que podría traer la implementación de un sistema distribuido basado en una red *peer-to-peer*, con un registro global de transacciones cuyo único mecanismo real de seguridad es la prueba de trabajo computacional basada en el cálculo de trillones de *hashes* por segundo.

Los esfuerzos deberían estar enfocados en cómo mejorar el sistema financiero mexicano contemplando no sólo una visión a largo plazo, sino con una que contemple también el corto plazo, esto es, el plazo inmediato, lo que deriva en mejorar de forma gradual y aprovechar de mejor manera la infraestructura y los recursos de los que ya se disponen actualmente. Por lo que los esfuerzos deben centrarse en mejorar el sistema financiero mexicano, pero desde una perspectiva no sólo teórica sino también pragmática en la que se contemple la realidad del sistema financiero mexicano.

En todo caso, si lo que se busca es la implementación de la tecnología Bitcoin porque se considera como una verdadera solución práctica a un problema específico y bien delimitado, el enfoque debería concentrarse en determinar si realmente es posible desarrollar una prueba de trabajo computacional eficiente, escalable y de bajo de costo, o un mecanismo que sustituya la forma en la que el sistema Bitcoin provee de seguridad a sus transacciones, pero manteniendo el mismo nivel de seguridad o mayor, sin perder de vista la practicidad de dicha solución, de otra manera sólo se trata de un ejercicio intelectual, teórico o simplemente especulativo.

Tal y como se planteó en el capítulo número dos, los tres pilares que pueden realmente mejorar y llevar al sistema financiero mexicano, o cualquier institución, hacia el mejoramiento continuo tanto de sus sistemas informáticos como de cualquier de sus procesos o actividades, es la aplicación de la transparencia, la fiscalización y la auditoría. Por su puesto no se trata de una fórmula mágica, pero estos tres conceptos fundamentales aplicados dentro de cualquier institución pública o privada aseguran mucho mejores resultados en comparación con los resultados que se pueden obtener en la ausencia de estos elementos.

La creación de estándares abiertos y neutrales enfocados en la seguridad, la eficiencia y la experiencia del usuario, por parte de las instituciones financieras mexicanas sería un buen comienzo para el mejoramiento de los servicios financieros, incluidas las transacciones financieras. Aunque existen estándares internacionales sobre diversos elementos como la seguridad informática o ciertos aspectos de los servicios financieros, sería de gran ayuda que las propias instituciones financieras mexicanas crearan sus propios estándares sobre cómo los sistemas informáticos financieros funcionan e interactúan entre ellos, de manera que exista uniformidad y consenso de cómo deben llevarse a cabo los diferentes procesos involucrados en las transacciones financieras. Por supuesto, estos estándares mexicanos deberán aplicar y tomar elementos existentes en los estándares internacionales y en los casos correspondientes aplicar estos últimos, sin embargo, la creación de los propios atendería las necesidades del sistema financiero mexicano, con sus particularidades, esto es, tomando en cuenta el contexto mexicano.

Al final, este trabajo representa una guía rápida de por qué los esfuerzos no deberían de enfocarse totalmente y de forma desmesurada en el *blockchain*, aún es posible mejorar los sistemas financieros mexicanos actuales de forma dramática utilizando la tecnología existente, mucha de ella se encuentra en forma de software gratuito y de código abierto, que ha demostrado ser eficiente, confiable, robusto y de grado empresarial como lo demuestran las exitosas compañías, muchas de ellas tecnológicas, cuyos sistemas informáticos se basan en estas tecnologías abiertas que son fácilmente susceptibles de la mejora continua, la fiscalización y la auditoria, ya que por defecto, tienen la propiedad de la transparencia al ser estos recursos tecnológicos de código abierto.

Todavía no se ha exprimido el máximo potencial de los recursos tecnológicos actuales por parte de las instituciones financieras mexicanas, por lo que es mucho más práctico, eficiente y rentable llevar a cabo reingeniería sobre los sistemas y procesos existentes en el sistema financiero mexicano para mejorar su eficiencia y seguridad. Durante los años en los que trabajé dentro del ámbito financiero mexicano observé que algunas instituciones financieras mexicanas todavía ocupan tecnología de hace una década o más, muchos de los sistemas informáticos usados por estas instituciones contienen código que se programó hace ya demasiado tiempo y pocas veces ha sido actualizado, si es que esto último se ha realizado, por lo que en cierto sentido

parte de todo el sistema financiero mexicano es un sistema antiguo en el que aún no se aprovecha todo el potencial de muchos avances tecnológicos que se han hecho y que en muchos casos no representan un cambio de paradigma tan abrupto y dramático como lo es la adopción de la tecnología Bitcoin, en muchas ocasiones simplemente estos avances son actualizaciones, mejoras en el rendimiento y la seguridad de la tecnología específica. Incluso la refactorización y reingeniería de ciertas partes en el código existente puede llevar a una mejora significativa en el rendimiento y la seguridad, muchas veces la simple actualización de un software específico junto con su correcta configuración puede hacer una enorme diferencia.

En este sentido, el Banco de México ha respondido recientemente a las necesidades actuales financieras con el lanzamiento de su nueva plataforma de pagos electrónicos de baja denominación, con lo que se busca incluir a un mayor número de mexicanos dentro del sistema financiero, disminuir el uso de efectivo y combatir el lavado de dinero. Esta nueva plataforma llamada CoDi[67] (Cobro Digital), no es más que una extensión de los servicios e infraestructura tecnológica actual del sistema mexicano, por lo que es un ejemplo de cómo se puede seguir mejorando los servicios y transacciones financieras mediante la optimización de la tecnología existente.

[67] El lanzamiento de esta plataforma ha sido posterior a la finalización de la presente obra, por esta razón no se ha incluido dentro de los sistemas que conforman el sistema financiero mexicano.

Apéndice A

Prefijos binarios equivalentes con el Sistema Internacional

Prefijo	Base 2	Valor
Kibi (Ki)	2^{10}	1024
Mebi (Mi)	2^{20}	1048576
Gibi (Gi)	2^{30}	1073741824
Tebi (Ti)	2^{40}	1099511627776
Pebi (Pi)	2^{50}	1125899906842624
Exbi (Ei)	2^{60}	1152921504606846976
Zebi (Zi)	2^{70}	1180591620717411303424
Yobi (Yi)	2^{80}	1208925819614629174706176

Tabla 8. Prefijos binarios.

Prefijos del Sistema Internacional

Prefijo	Base 10	Valor
Kilo	10^{3}	1000
Mega	10^{6}	1000000
Giga	10^{9}	1000000000
Tera	10^{12}	1000000000000
Peta	10^{15}	1000000000000000
Exa	10^{18}	1000000000000000000
Zetta	10^{21}	1000000000000000000000
Yotta	10^{24}	1000000000000000000000000

Tabla 9. Prefijos del sistema internacional.

Sistema métrico de escala larga

Unidad	Base 10	Valor
Mil	10^3	1000
Diez mil	10^4	10000
Cien mil	10^5	100000
Millón	10^6	1000000
Mil millones	10^9	1000000000
Billón	10^{12}	1000000000000
Mil billones	10^{15}	1000000000000000
Trillón	10^{18}	1000000000000000000
Mil trillones	10^{21}	10000000000000000000000
Cuatrillón	10^{24}	1000000000000000000000000
Mil cuatrillones	10^{27}	10000000000000000000000000000
Quintillón	10^{30}	1000000000000000000000000000000
Mil quintillones	10^{33}	10000000000000000000000000000000000
Sextillón	10^{36}	100000000000000000000000000000000000000

Tabla 10. Sistema métrico de escala larga.

Apéndice B

Código PHP

```php
<?php
namespace rrubioa\bitcoin\address;

/**
 * @author Ricardo Rubio <r_ricardo@outlook.com>
 * @copyright 2019 Ricardo Rubio
 * @license MIT https://opensource.org/licenses/MIT
 *
 * Funcionalidad básica para llaves de cifrado asimétrico ECDSA
 * y direcciones Bitcoin Base58check.
 */
interface BitcoinAddressInterface
{
    /**
     * Crea un par de llaves pública/privada para ser utilizadas
     * con el algoritmo ECDSA.
     *
     * <b>IMPORTANTE:</b>
     * <p>
     * Proceda con precaución con las llaves generadas por este método,
     * para más información sobre la seguridad criptográfica de las
     * llaves generadas, refiérase directamente a la implementación:
     * https://packagist.org/packages/mdanter/ecc
     * </p>
     *
     * @return void
     */
    public function createPairKeysEcdsa(): void;

    /**
     * Devuelve la llave pública con el estándar DER (Distinguished Encoding Rules)
     * y con una representación hexadecimal.
     *
     * @return string
     */
    public function getPublicKey(): string;

    /**
     * Devuelve la llave privada con el estándar DER (Distinguished Encoding Rules)
     * y con una representación hexadecimal.
     *
     * @return string
     */
    public function getPrivateKey(): string;

    /**
     * Devuelve el hash de la llave pública codificada como una cadena
     * de caracteres hexadecimal (véase Bitcoin KeyHash).
     *
     * @return string
     */
```

```
    public function getKeyHash(): string;

    /**
     * Devuelve una dirección de tipo Bitcoin derivada de la llave
     * pública generada por un objeto de esta clase. Con una codificación
     * Base58.
     *
     * @return string Dirección Bitcoin Base58check
     */
    public function getBitcoinAddress(): string;
}
```

```
<?php
namespace rrubioa\bitcoin\address;

use Mdanter\Ecc\EccFactory;
use Mdanter\Ecc\Serializer\PublicKey\DerPublicKeySerializer;
use Mdanter\Ecc\Serializer\PrivateKey\DerPrivateKeySerializer;
use Tuupola\Base58;

/**
 * @author Ricardo Rubio <r_ricardo@outlook.com>
 * @copyright 2019 Ricardo Rubio
 * @license MIT https://opensource.org/licenses/MIT
 *
 * @inheritDoc
 */
class BitcoinAddress implements BitcoinAddressInterface
{
    private $privateKey;
    private $publicKey;
    private $keyHash;

    /**
     * @inheritDoc
     */
    public function createPairKeysEcdsa(): void
    {
        $this->publicKey    = null;
        $this->keyHash      = null;
        $this->privateKey = EccFactory::getNistCurves()->generator256()->createPrivateKey();
    }

    /**
     * @inheritDoc
     */
    public function getPublicKey(): string
    {
        if($this->publicKey !== null)
            return $this->publicKey;

        $serializer         = new DerPublicKeySerializer(EccFactory::getAdapter());
        $publicKey          = $serializer->serialize($this->privateKey->getPublicKey());
        $this->publicKey    = bin2hex($publicKey);

        return $this->publicKey;
```

```
    }

    /**
     * @inheritDoc
     */
    public function getPrivateKey(): string
    {
        $serializer = new DerPrivateKeySerializer(EccFactory::getAdapter());
        return bin2hex($serializer->serialize($this->privateKey));
    }

    /**
     * @inheritDoc
     */
    public function getKeyHash(): string
    {
        if($this->keyHash !== null)
            return $this->keyHash;

        $binKey         = hex2bin('04'.$this->getPublicKey());
        $keySha256      = hash("sha256", $binKey, true);
        $this->keyHash  = ('00'.hash("ripemd160", $keySha256));

        return $this->keyHash;
    }

    /**
     * @inheritDoc
     */
    public function getBitcoinAddress(): string
    {
        $checkSum   = hash("sha256", hash("sha256", hex2bin($this->getKeyHash()), true));
        $checkSum   = substr($checkSum, 0, 8);
        $address    = $this->getKeyHash().$checkSum;
        $base58     = new Base58(['characters' => Base58::BITCOIN]);

        return $base58->encode(hex2bin($address));
    }
}
```

```
<?php
namespace rrubioa\bitcoin\blocks;

/**
 * @author Ricardo Rubio <r_ricardo@outlook.com>
 * @copyright 2019 Ricardo Rubio
 * @license MIT https://opensource.org/licenses/MIT
 *
 * Funcionalidad para bloques Bitcoin
 */
interface BlockInterface
{
    public const MAGIC_NUMBER       = "magicNumber";
    public const BLOCK_SIZE         = "blockSize";
    public const BLOCK_HEADER       = "blockHeader";
```

```
    public const NUM_TRANSACTIONS    = "numTransactions";
    public const TRANSACTIONS        = "transactions";
    public const VERSION             = "version";
    public const HASH_PREV_BLOCK     = "hashPrevBlock";
    public const HASH_MERKLE_ROOT    = "hashMerkleRoot";
    public const TIMESTAMP           = "timestamp";
    public const TARGET              = "targetDifficulty";
    public const NONCE               = "nonce";

    public const MAGIC_NUMBER_BYTES     = 4;
    public const BLOCK_SIZE_BYTES       = 4;
    public const BLOCK_HEADER_BYTES     = 80;
    public const VERSION_BYTES          = 4;
    public const HASH_PREV_BLOCK_BYTES  = 32;
    public const HASH_MERKLE_ROOT_BYTES = 32;
    public const TIMESTAMP_BYTES        = 4;
    public const TARGET_BYTES           = 4;
    public const NONCE_BYTES            = 4;

    /**
     * Realiza un análisis sintáctico (parse) profundo sobre un bloque
     * Bitcoin codificado como una cadena de caracteres hexadecimal.
     *
     * @param string $hexBlock El bloque Bitcon codificado en hexadecimal.
     * @return array
     */
    public function parseHex(string $hexBlock) : array;

    /**
     * Realiza un análisis sintáctico (parsing) profundo sobre un bloque
     * Bitcoin almacenado en un archivo binario blk<>.dat
     * (Véase Bitcoin file .dat).
     *
     * <b>IMPORTANTE:</b>
     * <p>
     * Este método sólo analiza un bloque a la vez desde la posición actual del puntero
     * del archivo. Para realizar un parse sobre todos los bloques en el archivo será
     * necesario crear un ciclo y llamar este método hasta alcanzar el final del archivo
     * binario.
     * </p>
     *
     * @param [resource] $fileHandler El handler de un archivo binario (sólo lectura).
     * @return array
     */
    public function parseBin(&$fileHandler): array;

    /**
     * Realiza un análisis sintáctico (parsing) parcial sobre un bloque
     * Bitcoin codificado como una cadena de caracteres hexadecimal.
     *
     * @param string $hexBlock El bloque Bitcoin codificado en hexadecimal.
     * @return array
     */
    public function shallowParseHex(string $hexBlock) : array;

    /**
     * Realiza un análisis sintáctico (parsing) parcial sobre un archivo
```

```
     * binario que contenga bloques Bitcoin (Véase Bitcion file blk<>.dat).
     *
     * <b>IMPORTANTE:</b>
     * <p>
     * Este método sólo analiza un bloque a la vez desde la posición actual del puntero
     * del archivo. Para realizar un parse sobre todos los bloques en el archivo será
     * necesario crear un ciclo y llamar este método hasta alcanzar el final del archivo
     * binario.
     * </p>
     *
     * @param [resource] $fileHandler El handler de un archivo binario en modo sólo lectura.
     * @return array
     */
    public function shallowParseBin(&$fileHandler): array;

    /**
     * Realiza un análisis sintáctico (parsing) sobre el encabezado
     * de un bloque Bitcoin codificado como cadena de caracteres
     * hexadecimal.
     *
     * @param string $hexHeader El encabezado del bloque codificado en hexadecimal.
     * @return array
     */
    public function parseHeaderHex(string $hexHeader) : array;
}
```

```
<?php
namespace rrubioa\bitcoin\blocks;

use rrubioa\bitcoin\transactions\TransactionInterface;

/**
 * @author Ricardo Rubio <r_ricardo@outlook.com>
 * @copyright 2019 Ricardo Rubio
 * @license MIT https://opensource.org/licenses/MIT
 *
 * @inheritDoc
 */
class Block implements BlockInterface
{
    private $hexTool;
    private $transaction;

    /**
     * Constructor, inyección de dependencias.
     *
     * <p>
     * <b>IMPORTANTE:</b> La inyección del objeto TransactinInterface es necesaria
     * si se ocupa el análisis sintáctico (parsing) profundo sobre un bloque Bitcoin.
     * </p>
     *
     * @param HexToolsInterface $hexTool Objeto para procesamiento hexadecimal.
     * @param TransactionInterface $transaction Objeto para procesamiento de transacciones.
     */
    public function __construct(
            HexToolsInterface $hexTool,
```

```
            TransactionInterface $transaction = null
            )
    {
        $this->hexTool      = $hexTool;
        $this->transaction  = $transaction;
    }

    /**
     * Inyecta la dependencia para satisfacer la necesidad de un objeto de tipo
     * transacción. Esta inyección es necesaria si utiliza el análisis sintáctico
     * (parse) profundo sobre un bloque Bitcoin.
     *
     * @param TransactionInterface $transaction
     * @return void
     */
    public function setTransaction(TransactionInterface $transaction): void
    {
        $this->transaction = $transaction;
    }

    /**
     * @inheritDoc
     */
    public function shallowParseHex(string $hexBlock): array
    {
        $hexArray   = str_split($hexBlock, 2);
        $block      = array();
        $index      = 0;

        $block[BlockInterface::BLOCK_HEADER] = $this->hexTool->getHexBytes(
            BlockInterface::BLOCK_HEADER_BYTES, $hexArray, $index
        );

        $block[BlockInterface::NUM_TRANSACTIONS] =
                    $this->hexTool->sizeVarInt($hexArray, $index);
        $block[BlockInterface::TRANSACTIONS] = $this->hexTool->getHexBytes(
            (count($hexArray) - $index), $hexArray, $index
        );

        return $block;
    }

    /**
     * @inheritDoc
     */
    public function shallowParseBin(&$fileHandler): array
    {
        $block = array();

        $block[BlockInterface::MAGIC_NUMBER] = $this->hexTool->littleToBig(
            bin2hex(fread($fileHandler, BlockInterface::MAGIC_NUMBER_BYTES))
        );

        $size = bin2hex(fread($fileHandler, BlockInterface::BLOCK_SIZE_BYTES));
        $block[BlockInterface::BLOCK_SIZE] = hexdec($this->hexTool->littleToBig($size));

        $index      = 0;
```

```
        $hexArray   =
            str_split(bin2hex(fread($fileHandler, $block[BlockInterface::BLOCK_SIZE])), 2);

        $block[BlockInterface::BLOCK_HEADER] = $this->hexTool->getHexBytes(
            BlockInterface::BLOCK_HEADER_BYTES, $hexArray, $index
        );

        $block[BlockInterface::NUM_TRANSACTIONS] =
                    $this->hexTool->sizeVarInt($hexArray, $index);
        $block[BlockInterface::TRANSACTIONS] = $this->hexTool->getHexBytes(
            (count($hexArray) - $index), $hexArray, $index
        );

        return $block;

    }

    /**
     * @inheritDoc
     */
    public function parseHex(string $hexBlock): array
    {
        $hexArray   = str_split($hexBlock, 2);
        $block      = array();
        $index      = 0;

        $block[BlockInterface::MAGIC_NUMBER] = $this->hexTool->getHexBytes(
            BlockInterface::MAGIC_NUMBER_BYTES, $hexArray, $index
        );
        $block[BlockInterface::BLOCK_SIZE] = $this->hexTool->getHexBytes(
            BlockInterface::BLOCK_SIZE_BYTES, $hexArray, $index
        );

        $hexHeader = $this->hexTool->getHexBytes(
                    BlockInterface::BLOCK_HEADER_BYTES, $hexArray, $index
             );
        $block[BlockInterface::BLOCK_HEADER] = $this->parseHeaderHex($hexHeader);

        $block[BlockInterface::NUM_TRANSACTIONS] =
                    $this->hexTool->sizeVarInt($hexArray, $index);

        $transactionsBytes = hexdec(
                    $this->hexTool->littleToBig($block[BlockInterface::BLOCK_SIZE])
             );
        $hexTransactions =
            $this->hexTool->getHexBytes($transactionsBytes, $hexArray, $index);

        $block[BlockInterface::TRANSACTIONS] = array();

        for($count = 0; $count < $block[BlockInterface::NUM_TRANSACTIONS]; $count++)
            $block[BlockInterface::TRANSACTIONS][$count] =
                        $this->transaction->parseHexArray($hexArray, $index);

        return $block;
    }

    public function parseBin(&$fileHandler): array
```

```
    {
        $block = array();

        $block[BlockInterface::MAGIC_NUMBER] = $this->hexTool->littleToBig(
            bin2hex(fread($fileHandler, BlockInterface::MAGIC_NUMBER_BYTES))
        );

        $size = bin2hex(fread($fileHandler, BlockInterface::BLOCK_SIZE_BYTES));
        $block[BlockInterface::BLOCK_SIZE] = hexdec($this->hexTool->littleToBig($size));

        $index      = 0;
        $hexArray   =
            str_split(bin2hex(fread($fileHandler, $block[BlockInterface::BLOCK_SIZE])), 2);

        $block[BlockInterface::BLOCK_HEADER] = $this->hexTool->getHexBytes(
            BlockInterface::BLOCK_HEADER_BYTES, $hexArray, $index
        );

        $block[BlockInterface::NUM_TRANSACTIONS] =
            $this->hexTool->sizeVarInt($hexArray, $index);

        $block[BlockInterface::TRANSACTIONS] = array();

        for($count = 0; $count < $block[BlockInterface::NUM_TRANSACTIONS]; $count++)
            $block[BlockInterface::TRANSACTIONS][$count] =
                        $this->transaction->parseHexArray($hexArray, $index);

        return $block;
    }

    /**
     * @inheritDoc
     */
    public function parseHeaderHex(string $hexHeader): array
    {
        $hexArray   = str_split($hexHeader, 2);
        $header     = array();
        $index      = 0;

        $header[BlockInterface::VERSION] = $this->hexTool->getHexBytes(
            BlockInterface::VERSION_BYTES, $hexArray, $index
        );
        $header[BlockInterface::HASH_PREV_BLOCK] = $this->hexTool->getHexBytes(
            BlockInterface::HASH_PREV_BLOCK_BYTES, $hexArray, $index
        );
        $header[BlockInterface::HASH_MERKLE_ROOT] = $this->hexTool->getHexBytes(
            BlockInterface::HASH_MERKLE_ROOT_BYTES, $hexArray, $index
        );
        $header[BlockInterface::TIMESTAMP] = $this->hexTool->getHexBytes(
            BlockInterface::TIMESTAMP_BYTES, $hexArray, $index
        );
        $header[BlockInterface::TARGET] = $this->hexTool->getHexBytes(
            BlockInterface::TARGET_BYTES, $hexArray, $index
        );
        $header[BlockInterface::NONCE] = $this->hexTool->getHexBytes(
            BlockInterface::NONCE_BYTES, $hexArray, $index
        );
```

```
        return $header;
    }
}
```

```php
<?php
namespace rrubioa\bitcoin\transactions;

/**
 * @author Ricardo Rubio <r_ricardo@outlook.com>
 * @copyright 2019 Ricardo Rubio
 * @license MIT https://opensource.org/licenses/MIT
 *
 * Funcionalidad para transacciones Bitcoin.
 */
interface TransactionInterface
{
    public const VERSION            = "version";
    public const NUM_INPUTS         = "numInputs";
    public const INTPUTS            = "inputs";
    public const TXID               = "txId";
    public const TXOUT_INDEX        = "txoutIndex";
    public const UNLOCKING_SCRIPT   = "unlockingScript";
    public const SEQUENCE           = "sequence";
    public const NUM_OUTPUTS        = "numOutputs";
    public const OUTPUTS            = "outputs";
    public const SATOSHIS           = "satoshis";
    public const LOCKING_SCRIPT     = "lockingScript";
    public const LOCKTIME           = "locktime";

    /**
     * Realiza un análisis sintáctico (parsing) sobre una transacción Bitcoin
     * codificada como una cadena de caracteres hexadecimal.
     *
     * @param string $hexTransaction La transacción codificada en hexadecimal.
     * @return array
     */
    public function parseHex(string $hexTransaction): array;

    /**
     * Realiza un análisis sintáctico (parsing) sobre una transacción Bitcoin
     * codificada como un arreglo de caracteres hexadecimales agrupados por
     * bytes.
     *
     * <p>
     * <b>IMPORTANTE:</b> El arreglo debe contener caracteres hexadecimales
     * agrupados por bytes, esto es, cada elemento del arreglo debe contener
     * exactamente 2 caracteres hexadecimales.
     * </p>
     *
     * @param array $hexArray Transacción codificada como un arreglo hexadecimal
     *                        agrupado por bytes.
     * @param integer $index Un índice o puntero que será usado para recorrer el
     *                       el arreglo de bytes hexadecimales.
     * @return array
     */
```

```
    public function parseHexArray(array &$hexArray, int &$index): array;
}
```

```
<?php
namespace rrubioa\bitcoin\transactions;

use rrubioa\bitcoin\utils\HexToolsInterface;

/**
 * @author Ricardo Rubio <r_ricardo@outlook.com>
 * @copyright 2019 Ricardo Rubio
 * @license MIT https://opensource.org/licenses/MIT
 *
 * @inheritDoc
 */
class Transaction implements TransactionInterface
{
    private $hexTools;

    /**
     * Constructor, inyección de dependencias.
     *
     * @param HexToolsInterface $hexTools Objeto con las funcionalidades necesarias
     *                                    para manipular bytes hexadecimales.
     */
    public function __construct(HexToolsInterface $hexTools)
    {
        $this->hexTools = $hexTools;
    }

    /**
     * Lleva a cabo el análisis sintáctico (parsing) de una transacción representada como
     * un arreglo de caracteres hexadecimales agrupados en bytes, es decir, cada elmento
     * del arreglo debe contener exactamente 2 caracteres hexadecimales.
     *
     * @param array $hexArray Arreglo hexadecimal de bytes.
     * @param integer $index El índice que será usado para recorrer el arreglo.
     * @return array
     */
    private function parseTransaction(array &$hexArray, int &$index): array
    {
        $transaction = array();

        $transaction[TransactionInterface::VERSION] =
                    $this->hexTools->getHexBytes(4, $hexArray, $index);
        $transaction[TransactionInterface::NUM_INPUTS] =
                    $this->hexTools->sizeVarInt($hexArray, $index);
        $transaction[TransactionInterface::INTPUTS] = array();

        // Inputs
        for($count = 0;
            $count < $transaction[TransactionInterface::NUM_INPUTS];
            $count++) {

            $transaction[TransactionInterface::INTPUTS][$count] = array();
```

```
            $transaction
                [TransactionInterface::INTPUTS]
                [$count]
                [TransactionInterface::TXID] =
                    $this->hexTools->getHexBytes(32, $hexArray, $index);

            $transaction
                    [TransactionInterface::INTPUTS]
                    [$count]
                    [TransactionInterface::TXOUT_INDEX] =
                        $this->hexTools->getHexBytes(4, $hexArray, $index);

            $scriptSize = $this->hexTools->sizeVarInt($hexArray, $index);

            $transaction
                    [TransactionInterface::INTPUTS]
                    [$count]
                    [TransactionInterface::UNLOCKING_SCRIPT] =
                        $this->hexTools->getHexBytes($scriptSize, $hexArray, $index);

            $transaction
                    [TransactionInterface::INTPUTS]
                    [$count]
                    [TransactionInterface::SEQUENCE] =
                        $this->hexTools->getHexBytes(4, $hexArray, $index);
        }

        $transaction[TransactionInterface::NUM_OUTPUTS] =
                    $this->hexTools->sizeVarInt($hexArray, $index);
        $transaction[TransactionInterface::OUTPUTS] = array();

        // Outputs
        for($count = 0;
            $count < $transaction[TransactionInterface::NUM_OUTPUTS];
            $count++) {

            $transaction[TransactionInterface::OUTPUTS][$count] = array();

            $transaction
                [TransactionInterface::OUTPUTS]
                [$count]
                [TransactionInterface::SATOSHIS] =
                    $this->hexTools->getHexBytes(8, $hexArray, $index);

            $scriptSize = $this->hexTools->sizeVarInt($hexArray, $index);

            $transaction
                [TransactionInterface::OUTPUTS]
                [$count]
                [TransactionInterface::LOCKING_SCRIPT] =
                    $this->hexTools->getHexBytes($scriptSize, $hexArray, $index);
        }

        $transaction[TransactionInterface::LOCKTIME] =
            $this->hexTools->getHexBytes(4, $hexArray, $index);
        return $transaction;
    }
```

```
    /**
     * @inheritDoc
     */
    public function parseHex(string $hexTransaction): array
    {
        $hexArray   = str_split($hexTransaction, 2);
        $index      = 0;
        return $this->parseTransaction($hexArray, $index);
    }
    /**
     * @inheritDoc
     */
    public function parseHexArray(array &$hexArray, int &$index): array
    {
        return $this->parseTransaction($hexArray, $index);
    }
}
```

```
<?php
namespace rrubioa\bitcoin\utils;

/**
 * @author Ricardo Rubio <r_ricardo@outlook.com>
 * @copyright 2019 Ricardo Rubio
 * @license MIT https://opensource.org/licenses/MIT
 *
 * Herramientas útiles para el manejo de bytes codificados en hexadecimal.
 */
interface HexToolsInterface
{

    /**
     * Convierte cadenas de caracteres hexadecimales en orden little endian
     * a cadenas hexadecimales con orden big endian.
     *
     * @param string $hex Una cadena hexadecimal en orden little endian
     * @return string
     */
    public function littleToBig(string $hex): string;

    /**
     * Obtiene el valor entero decimal de un tipo de dato VarInt (Vease: Bitcoin VarInt).
     *
     * <p>
     * Si el byte codificado en hexadecimal NO contiene uno de los tres prefijos,
     * entonces se toma el valor de ese byte para obtener VarInt.
     * </p>
     *
     * <ul>
     *  <li>0xfd Significa que se toman los siguientes 2 bytes
     *      para poder conocer varint</li>
     *
     *  <li>0xfe Significa que se toman los siguientes 4 bytes
     *      para poder conocer varint</li>
     *
```

```
     *  <li>0xff Significa que se toman los siguientes 8 bytes
     *      para poder conocer varint</li>
     * </ul>
     *
     * <p>
     * <b>IMPORTANTE:</b> Cuando se obtiene VarInt utilzando uno de los prefijos, este valor
     * hexadecimal se debe convertir a orden big endian ya que por defecto se encuentra en
     * orden little endian.
     * </p>
     *
     * @param array $hexArray   Un arreglo de bytes codificados en hexadecimal.
     * @param integer $index    Un índice correspondiente al arreglo pasado como argumento.
     *                          Un índice utilizado como puntero dentro del arreglo.
     * @return integer
     */
    public function sizeVarInt(array &$hexArray, int &$index) : int;

    /**
     * Obtiene el valor entero decimal de un tipo de dato VarInt (Vease: Bitcoin VarInt)
     * a partir de un archivo binario blk<>.dat contenedor de bloques Bitcoin.
     *
     * <p>
     * Si el byte codificado en hexadecimal NO contiene uno de los tres prefijos,
     * entonces se toma el valor de ese byte para obtener VarInt.
     * </p>
     *
     * <ul>
     *  <li>0xfd Significa que se toman los siguientes 2 bytes
     *      para poder conocer varint</li>
     *
     *  <li>0xfe Significa que se toman los siguientes 4 bytes
     *      para poder conocer varint</li>
     *
     *  <li>0xff Significa que se toman los siguientes 8 bytes
     *      para poder conocer varint</li>
     * </ul>
     *
     * <p>
     * <b>IMPORTANTE:</b> Cuando se obtiene VarInt utilzando uno de los prefijos, este valor
     * hexadecimal se debe convertir a orden big endian ya que por defecto se encuentra en
     * orden little endian.
     * </p>
     *
     * @param resource $fileHandler El handler de un archivo binario en modo de sólo
     *                              lectura.
     *
     * @return integer
     */
    public function sizeVarIntFile(&$fileHandler): int;

    /**
     * Obtiene el número de bytes especificados de un arreglo de bytes codificados en
     * hexadecimal.
     *
     * @param integer $numBytes El número de bytes requeridos del arreglo.
     * @param array $hexArray   El arreglo de bytes codificados en hexadecimal del
     *                          cual se obtendrán los bytes especificados.
```

```
     * @param integer $index    Un índice utilizado como puntero del arreglo para obtener
     *                          los datos.
     * @return string
     */
    public function getHexBytes(int $numBytes, array $hexArray, int &$index) : string;
}
```

```php
<?php
namespace rrubioa\bitcoin\utils;

/**
 * @author Ricardo Rubio <r_ricardo@outlook.com>
 * @copyright 2019 Ricardo Rubio
 * @license MIT https://opensource.org/licenses/MIT
 *
 * @inheritDoc
 */
class HexTools implements HexToolsInterface
{
    /**
     * @inheritDoc
     */
    public function littleToBig(string $hex): string
    {
        $hexArray = str_split($hex, 2);
        $hexArray = array_reverse($hexArray);
        return implode("", $hexArray);
    }

    /**
     * @inheritDoc
     */
    public function sizeVarInt(array &$hexArray, int &$index): int
    {
        $varint = $hexArray[$index++];
        $size   = 0;

        switch($varint) {
            case "fd":
                $size = $this->getHexBytes(2, $hexArray, $index);
                break;
            case "fe":
                $size = $this->getHexBytes(4, $hexArray, $index);
                break;
            case "ff":
                $size = $this->getHexBytes(8, $hexArray, $index);
                break;
            default:
                return hexdec($varint);
        }

        return hexdec($this->littleToBig($size));
    }

    /**
     * @inheritDoc
```

```
     */
    public function sizeVarIntFile(&$fileHandler): int
    {
        $varint = bin2hex(fread($fileHandler, 1));
        $size   = 0;

        switch($varint) {
            case "fd":
                $size = bin2hex(fread($fileHandler, 2));
                break;
            case "fe":
                $size = bin2hex(fread($fileHandler, 4));
                break;
            case "ff":
                $size = bin2hex(fread($fileHandler, 8));
                break;
            default:
                return hexdec($varint);
        }

        return hexdec($this->littleToBig($size));
    }

    /**
     * @inheritDoc
     */
    public function getHexBytes(int $numBytes, array $hexArray, int &$index): string
    {
        $available = (count($hexArray) - $index);

        if($available < $numBytes)
            throw new \Exception("Sin suficientes datos en \$hexArray. Se solicitaron".
            " {$numBytes} bytes, pero sólo hay {$available} bytes disponibles");

        $hex = "";
        for($count = 0; $count < $numBytes; $count++)
            $hex .= $hexArray[$index++];

        return $hex;
    }
}
```

Composer JSON

```
{
	"name" : "rrubioa/bitcoin",
	"description" : "Basic Bitcoin API implementation in PHP",
	"type" : "library",
	"license" : "https://opensource.org/licenses/MIT",
	"authors" : [{
			"name" : "Ricardo Rubio",
			"email" : "r_ricardo@outlook.com"
		}
	],
	"require" : {
		"php" : "7.2.22",
		"mdanter/ecc" : "^0.5.2",
		"tuupola/base58" : "^2.0"
	},
	"autoload" : {
		"psr-4" : {
			"bitcoin\\" : "src/"
		}
	}
}
```

Referencias

Agence France-Presse. (03 de 10 de 2018). El hackeo a Bancomext vino desde Corea del Norte, según expertos. *Milenio*. Obtenido de https://expansion.mx/empresas/2018/10/03/el-hackeo-a-bancomext-vino-desde-corea-del-norte-segun-expertos

Agencia EFE. (24 de 08 de 2016). *Santander y otros tres bancos experimentan con la tecnología de pagos del bitcoin*. Obtenido de Agencia EFE: https://www.efe.com/efe/espana/efeempresas/santander-y-otros-tres-bancos-experimentan-con-la-tecnologia-de-pagos-del-bitcoin/50000908-3021462

Antonopoulos, A. (2017). *Mastering Bitcoin* (Segunda ed.). California, Estados Unidos: O'reilly.

Arnold, M. (15 de 09 de 2016). La banca acelera en el uso del 'blockchain'. *Expansión*. Obtenido de https://www.expansion.com/economia-digital/innovacion/2016/09/15/57d190a322601d456d8b45cb.html

BAE Systems Applied Intelligence. (12 de 02 de 2017). *THREAT RESEARCH BLOG*. Obtenido de https://baesystemsai.blogspot.com/2017/02/lazarus-watering-hole-attacks.html

BANCOMEXT. (22 de 10 de 2018). *¿Quiénes somos?* Obtenido de BANCOMEXT: https://www.bancomext.com/conoce-bancomext/quienes-somos

Bank For International Settlements. (11 de 2016). *Fast payments – Enhancing the speed and availability of retail payments.* Obtenido de Bank For International Settlements: https://www.bis.org/cpmi/publ/d154.pdf

Bank of England. (01 de 10 de 2019). *Payment and settlements.* Obtenido de Bank of England: https://www.bankofengland.co.uk/payment-and-settlement

Banxico. (2016). *Política y funciones del Banco de México respecto a las infraestructuras de los mercados financieros.* Obtenido de Banco de México: http://www.banxico.org.mx/sistemas-de-pago/d/%7B9ACA4DC8-2B96-8EB3-6FF3-F58DDFA3FE51%7D.pdf

Banxico. (s.f.). *Características del Sistema de Pagos Electrónicos Interbancarios (SPEI)*. Obtenido de Banco de México: http://www.banxico.org.mx/servicios/spei_-transferencias-banco-me.html

Banxico. (s.f.). *Cronología de los principales cambios en el sistema de pagos*. Obtenido de Banco de México: http://www.banxico.org.mx/sistemas-de-pago/cronologia-sistemas-pago-tran.html

Banxico. (s.f.). *Información del Sistema de Pagos Interbancarios en Dólares (SPID)*. Obtenido de Banco de México: http://www.banxico.org.mx/servicios/sistema-pagos-interbancarios-.html

Banxico. (s.f.). *Información del SPEI para el público en general*. Obtenido de Banco de México: http://www.banxico.org.mx/servicios/sistema-pagos-electronicos-in.html

Banxico. (s.f.). *Sistemas de Pago*. Obtenido de Banco de México: http://www.anterior.banxico.org.mx/divulgacion/sistemas-de-pago/sistemas-pago.html

BBVA. (23 de 12 de 2015). *¿Qué es un repo?* Obtenido de BBVA Educación Financiera: https://www.bbva.com/es/que-es-un-repo/

Beebe, N. (29 de 07 de 2019). *A Complete Bibliography of Publications of John*. Obtenido de netlib: http://www.netlib.org/bibnet/authors/v/von-neumann-john.pdf

Bergin, T., & Layne, N. (20 de 05 de 2016). *Special Report - Cyber thieves exploit banks' faith in SWIFT transfer network*. Obtenido de Reuters: https://uk.reuters.com/article/uk-cyber-heist-swift-specialreport/special-report-cyber-thieves-exploit-banks-faith-in-swift-transfer-network-idUKKCN0YB0DO

Bnamericas. (14 de 08 de 2019). *Mexico tech revolution may be sitting on cybersecurity bomb*. Obtenido de Bnamericas: https://www.bnamericas.com/en/news/mexico-tech-revolution-may-be-sitting-on-cybersecurity-bomb

Bonnet, C. (17 de 05 de 2010). *A Piece of Internet History*. Obtenido de Duke Today: https://today.duke.edu/2010/05/usenet.html

Brooks, D. (20 de 11 de 2018). García Luna y otros funcionarios, sobornados por el narco: Zambada. *La Jornada*. Obtenido de https://www.jornada.com.mx/ultimas/2018/11/20/garcia-luna-y-otros-funcionarios-sobornados-por-el-narco-zambada-9345.html

Bushman, R., & Smith, A. (04 de 2003). Transparency, Financial Accounting Information, and Corporate Governance. *Economic Policy Review, 9*(1). Obtenido de https://ssrn.com/abstract=795547

Buss, D. (18 de 12 de 2018). *Conflict Minerals and Sexual Violence in Central Africa: Troubling Research* . Obtenido de Oxford Academic: https://academic.oup.com/sp/article/25/4/545/5251792

Cambridge Centre for Alternative Finance. (2019). *Cambridge Bitcoin Electricity Consumption Index*. Obtenido de University of Cambridge: https://www.cbeci.org/

CBS News. (26 de 08 de 2018). *Cryptocurrency: Virtual money, real power and the fight for a small town's future*. Obtenido de YouTube: https://www.youtube.com/watch?v=S00MWI3YeP4

CIA. (2016). *CIA (Central Intelligence Agency)*. Obtenido de https://www.cia.gov/library/publications/resources/the-world-factbook/fields/253rank.html

Comfort, N. (09 de 11 de 2018). *If Iranian Banks Are Blocked, Is Russia Next? Germany Wants to Know*. Obtenido de Bloomberg: https://www.bloomberg.com/news/articles/2018-11-09/if-swift-blocks-iranian-banks-is-russia-next-german-lobby-asks

CONDUSEF. (s.f.). *¿Qué hacemos?* Obtenido de CONDUSEF: https://www.gob.mx/condusef/que-hacemos

CONDUSEF. (2014). *REMESAS. ¡No te dejes sorprender, haz rendir tus envíos!* Folleto, Secretaria de Hacienda y Crédito Público, CONDUSEF, Ciudad de México. Obtenido de https://www.gob.mx/cms/uploads/attachment/file/95777/CUADERNOSYVIDEOS-REMESAS.pdf

Cortina Morfín, J., & Álvarez Toca, C. (2014). *El Mercado de Valores Gubernamentales en México.* Ciudad de México: Banco de México. Obtenido de http://www.anterior.banxico.org.mx/elib/mercado-valores-gub/OEBPS/Text/default.html

Dulaney, E., & Easttom, C. (2018). *CompTIA Security+ Study Guide* (Séptima ed.). Indianapolis, Indiana: SIBEX.

Earn bitcoin. (2019). *PREDICTING BITCOIN FEES FOR TRANSACTIONS*. Obtenido de https://bitcoinfees.earn.com/

EP. (26 de 08 de 2016). Santander se une a cinco entidades para impulsar el dinero digital. *El País*. Obtenido de https://elpais.com/economia/2016/08/24/actualidad/1472029845_649313.html

Esquivel, J. (17 de 07 de 2012). HSBC acepta que permitió el lavado de dinero en México, Irán y Siria. *Proceso*. Obtenido de https://www.proceso.com.mx/314328/hsbc-acepta-que-permitio-el-lavado-de-dinero-en-mexico-iran-y-siria

Ferguson, N., Schneier, B., & Kohno, T. (2010). *Cryptography Engineering.* Indianapolis, Indiana: Wiley.

Ferguson, T. (s.f.). *GAME THEORY PART II. Two-Person Zero-sum Games*. Obtenido de UCLA : https://www.math.ucla.edu/~tom/Game_Theory/mat.pdf

Flores, N. (23 de 10 de 2016). Bancos han lavado más de 150 mil millones de pesos. *ContraLínea*. Obtenido de https://www.contralinea.com.mx/archivo-revista/2016/10/23/bancos-han-lavado-mas-de-150-mil-millones-de-pesos/

Forbes. (06 de 05 de 2019). *Sector público, con alto riesgo de ataques cibernéticos: especialista*. Obtenido de Forbes: https://www.forbes.com.mx/sector-publico-con-alto-riesgo-de-ataques-ciberneticos-especialista/

G.F. (09 de 07 de 2018). *Why bitcoin uses so much energy*. Obtenido de The Economist: https://www.economist.com/the-economist-explains/2018/07/09/why-bitcoin-uses-so-much-energy

Genesis Mining. (25 de 01 de 2018). *Marco Streng, CEO of Genesis Mining, recaps 2017 and gives an outlook on 2018*. Obtenido de YouTube: https://www.youtube.com/watch?v=W43Vl8FeIfA

González, S. (10 de 06 de 2019). México, primer lugar en América Latina en ciberataques: estudio. *La jornada*. Obtenido de https://www.jornada.com.mx/ultimas/economia/2019/06/10/mexico-primer-lugar-en-america-latina-en-ciberataques-estudio-1562.html

GReAT. (16 de 02 de 2015). *The Great Bank Robbery: the Carbanak APT*. Obtenido de Kaspersky: https://securelist.com/the-great-bank-robbery-the-carbanak-apt/68732/

Gutiérrez, F. (06 de 09 de 2017). México con cifras irrisorias en lucha antilavado 2017. *El economista*. Obtenido de https://www.eleconomista.com.mx/sectorfinanciero/Mexico-con-cifras-irrisorias-en-lucha-antilavado-20170906-0199.html

hack.org. (27 de 02 de 2014). *What is a hacker?* Obtenido de hack: https://hack.org/faq-hacker.html

Harari, Y. N. (2014). *Sapiens De animales a dioses.* Barcelona: Penguin Random House.

Hernández Armenta, M. (16 de 08 de 2019). *Check Point prevé más ciberataques a financieras y a gobiernos*. Obtenido de Forbes: https://www.forbes.com.mx/veremos-mas-ciberataques-y-no-solo-a-financieras-sino-a-gobiernos-check-point/

Hernández, A., & Lara, P. (17 de 02 de 2017). CNBV y nueve bancos han sido hackeados. *Excelsior*. Obtenido de https://www.excelsior.com.mx/hacker/2017/02/17/1146899

Hughes, J. (27 de 09 de 2017). *Bitcoin Controlled supply*. Obtenido de Steemit: https://steemit.com/bitcoin/@urbandwellers/bitcoin-controlled-supply

IME. (18 de 08 de 2016). *Directo a México*. Obtenido de Instituto de los Mexicanos en el Exterior: https://www.gob.mx/ime/acciones-y-programas/directo-a-mexico-59159

Jakobsson, M., & Juels, A. (1999). *Proofs of Work and Bread Pudding Protocols.* Obtenido de Springer: https://link.springer.com/content/pdf/10.1007%2F978-0-387-35568-9_18.pdf

Jamasmie, C. (06 de 04 de 2017). *'Conflict minerals' entering tech supply chains from countries beyond Africa — report*. Obtenido de Mining Dot Com: https://www.mining.com/conflict-minerals-entering-tech-supply-chains-from-countries-beyond-africa-report/

Johnsen, J. A., Karlsen, L. E., & Birkeland, S. S. (12 de 2005). *Peer-to-peer networking with BitTorrent.* Obtenido de UCLA - Department of Telematics: http://web.cs.ucla.edu/classes/cs217/05BitTorrent.pdf

json.org. (s.f.). *Introducing JSON*. Obtenido de https://www.json.org

Juárez, E. (03 de 05 de 2018). ¿Cuántas operaciones se realizan en la actualidad en SPEI? *El economista*. Obtenido de https://www.eleconomista.com.mx/sectorfinanciero/Cuantas-operaciones-se-realizan-en-la-actualidad-en-SPEI-20180503-0150.html

Kagan, J. (18 de 02 de 2018). *Society for Worldwide Interbank Financial Telecommunications (SWIFT)*. Obtenido de https://www.investopedia.com/terms/s/swift.asp

Kagan, J. (07 de 10 de 2019). *Automated Teller Machine (ATM)*. Obtenido de Investopedia: https://www.investopedia.com/terms/a/atm.asp

La Jornada. (02 de 02 de 2019). Ingreso récord de remesas por 33 mil 481 mdd en 2018. *La Jornada*.

Levy, S. (21 de 11 de 2014). *What is a hacker?* Obtenido de Wired: https://www.wired.com/2014/11/what-is-a-hacker/

Leyva, J. (11 de 08 de 2017). Tanto cuidan el 'lavado de dinero' en México que sigue creciendo. *El Financiero*. Obtenido de https://www.elfinanciero.com.mx/economia/tanto-cuidan-el-lavado-de-dinero-en-mexico-que-sigue-creciendo

Library of Economics and Liberty. (s.f.). *John von Neumann*. Obtenido de Library of Economics and Liberty: https://www.econlib.org/library/Enc/bios/Neumann.html

López, I. (15 de 05 de 2018). Esto es lo que se sabe del robo millonario tras el hackeo a los bancos. *Forbes*. Obtenido de https://www.forbes.com.mx/esto-es-lo-que-se-sabe-del-robo-con-el-ciberataque-a-los-bancos/

Love, D. (25 de 09 de 2017). *The Mining of Coltan: Chances are Your Smartphone Was Manufactured With African Blood*. Obtenido de Atanta Black Star: https://atlantablackstar.com/2017/09/25/mining-coltan-chances-smartphone-manufactured-african-blood/

Martindale, J. (12 de 04 de 2018). *What is an ASIC miner?* Obtenido de Digital Trends: https://www.digitaltrends.com/computing/what-is-an-asic-miner/

md5hashing. (s.f.). *Md2*. Obtenido de md5hashing: https://md5hashing.net/hash/md2

Motherboard. (06 de 02 de 2015). *Life Inside a Secret Chinese Bitcoin Mine*. Obtenido de YouTube: https://www.youtube.com/watch?v=K8kua5B5K3I

Muñóz Razo, C. (2002). *Auditoría en sistemas computacionales.* México: Pearson Educación.

Naessens, H. (09 de 2010). *Ética pública y transparencia. XIV Encuentro de Latinoamericanistas Españoles.* Obtenido de HAL - Sciences de l'Homme et de la Société: https://halshs.archives-ouvertes.fr/halshs-00531532/document

Nájar, A. (27 de 02 de 2017). Por qué los carteles del narcotráfico en México se parecen a las grandes multinacionales financieras. *BBC News*. Obtenido de https://www.bbc.com/mundo/noticias-america-latina-39035737

Nakamoto, S. (31 de 10 de 2008). *Bitcoin: A Peer-to-Peer Electronic Cash System.* Obtenido de bitcoin: https://www.bitcoin.org/bitcoin.pdf

Nakamoto, S. (2009). Base58.h. *bitcoin-0.1.5*.

NearFieldCommunication.org. (2017). *About Near Field Communication*. Obtenido de NearFieldCommunication.org: http://nearfieldcommunication.org/about-nfc.html

Newman, L. (15 de 03 de 2019). *HOW HACKERS PULLED OFF A $20 MILLION MEXICAN BANK HEIST*. Obtenido de WIRED: https://www.wired.com/story/mexico-bank-hack/

Notimex. (07 de 04 de 2019). Piratas informáticos tardaron 18 meses en atacar bancos. *La Jornada*. Obtenido de https://www.jornada.com.mx/ultimas/economia/2019/04/07/piratas-informaticos-tardaron-18-meses-en-atacar-bancos-tekium-790.html

Oram, A. (2001). *Peer-to-Peer: Harnesing the power of Disruptive Technologies.* California, United States: O'Reilly Media.

Paar, C., & Pelzl, J. (2010). *Understanding Cryptography.* New York: Springer.

Park, H., Ratzin, R. I., & van der Schaar, M. (s.f.). *Peer-to-Peer Networks –Protocols, Cooperation and Competition.* Obtenido de vanderschaar-lab: http://www.vanderschaar-lab.com/papers/chapter_P2P_hpark.pdf

Pérez, C., Delgado, S., Navarro, G., & Herrera, J. (s.f.). *Double-spending Prevention.* Obtenido de Universidad Autónoma de Barcelona - Department of Information Engineering and Communications: https://eprint.iacr.org/2017/394.pdf

Priego Hernández, O., Ramírez Martínez, M., & García Rodríguez, J. (07 de 2018). Fiscalización, Transparencia y Rendición de Cuentas en México. *Horizontes de la Contaduría en las Ciencias Sociales*(9). Obtenido de https://www.uv.mx/icp/files/2018/12/Num09-Art13-102.pdf

Quentson, A. (31 de 10 de 2016). *WAS THE BITCOIN WHITE PAPER INTENTIONALLY PUBLISHED ON HALLOWEEN?* Obtenido de ccn: https://www.ccn.com/bitcoin-white-paper-intentionally-published-halloween/

Rouse, M., Sjoholm, H., & Rosencrance, L. (08 de 2017). *Hacker*. Obtenido de TechTarget: https://searchsecurity.techtarget.com/definition/hacker

RT News. (01 de 10 de 2018). *Iran considers SWIFT payment system alternative to bypass US sanctions*. Obtenido de RT News: https://www.rt.com/business/440017-iran-swift-analogue-sanctions/

Saint Bitts LLC. (2019). *Hash Rate*. Recuperado el 09 de 2019, de charts.bitcoin.com: https://charts.bitcoin.com/btc/chart/hash-rate#5moc

Schollmeier, R. (2001). *A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications.* Obtenido de researchgate.net: https://www.researchgate.net/publication/3940901_A_Definition_of_Peer-to-Peer_Networking_for_the_Classification_of_Peer-to-Peer_Architectures_and_Applications

Scott, G. (24 de 05 de 2019). *Financial Information Exchange*. Obtenido de Investopedia: https://www.investopedia.com/terms/f/financial-information-exchange.asp

Seth, S. (29 de 05 de 2019). *How the SWIFT System Works*. Recuperado el 03 de 02 de 2019, de Investopedia: https://www.investopedia.com/articles/personal-finance/050515/how-swift-system-works.asp

Shevchenko, S. (25 de 04 de 2016). *Two Bytes To $951M*. Obtenido de THREAT RESEARCH BLOG: https://baesystemsai.blogspot.com/2016/04/two-bytes-to-951m.html

Sotomayor, A. (2008). *Auditoría Administrativa Proceso y Aplicación.* México: McGrawHill.

Stallings, W. (2014). *Cryptography and Network Security Principles and Practice* (Sexta ed.). Estados Unidos: Pearson.

SWIFT. (s.f.). *Introduction to SWIFT*. Obtenido de SWIFT: https://www.swift.com/about-us/discover-swift

Symantec. (s.f.). *What is the Difference Between Black, White and Grey Hat Hackers?* Obtenido de Symantec: https://us.norton.com/internetsecurity-emerging-threats-what-is-the-difference-between-black-white-and-grey-hat-hackers.html

Taibbi, M. (13 de 12 de 2012). Outrageous HSBC Settlement Proves the Drug War Is a Joke. *RollingStone*. Obtenido de https://www.rollingstone.com/politics/politics-news/outrageous-hsbc-settlement-proves-the-drug-war-is-a-joke-230696/

Tapscott, D., & Tapscott, A. (2018). *La Revolución Blockchain.* Ciudad de México: Paidos.

TradeBlock. (15 de 10 de 2015). *Analysis of Bitcoin Transaction Size Trends*. Obtenido de TradeBlock: https://tradeblock.com/blog/analysis-of-bitcoin-transaction-size-trends

U.S. Energy Information Administration. (s.f.). *Electricity explained*. Obtenido de U.S. Energy Information Administration: https://www.eia.gov/energyexplained/electricity/measuring-electricity.php

Usla, H. (08 de 05 de 2018). Remesas a México superaran los 30 mil mdd en 2018 estiman expertos. *El financiero*. Obtenido de https://www.elfinanciero.com.mx/economia/remesas-a-mexico-superaran-los-30-mil-mdd-en-2018-estiman-expertos

UTF-8. (07 de 02 de 2019). *UTF-8 and Unicode*. Obtenido de UTF-8: http://www.utf-8.com

Vincent, J. (04 de 07 de 2019). *Bitcoin consumes more energy than Switzerland, according to new estimate*. Obtenido de The Verge: https://www.theverge.com/2019/7/4/20682109/bitcoin-energy-consumption-annual-calculation-cambridge-index-cbeci-country-comparison

Walker, G. (18 de 08 de 2015). *Transaction Data*. Obtenido de learn me a bitcoin: https://learnmeabitcoin.com/glossary/transaction-data

Walker, G. (23 de 03 de 2016). *Block Header*. Obtenido de learn me a bitcoin: https://learnmeabitcoin.com/glossary/block-header

Walker, G. (09 de 01 de 2017). *blk.dat*. Obtenido de learn me a bitcoin: https://learnmeabitcoin.com/glossary/blkdat

Warburg, B. (28 de 11 de 2017). *Blockchain Expert Explains One Concept in 5 Levels of Difficulty*. Obtenido de YouTube: https://www.youtube.com/watch?v=hYip_Vuv8J0

Zetter, K. (17 de 05 de 2016). That insane, $81M Bangladesh Bank Heist? Here's What We Know. *Wired*. Obtenido de https://www.wired.com/2016/05/insane-81m-bangladesh-bank-heist-heres-know/

Zhao, W. (05 de 08 de 2019). *Bitcoin's Computing Power Sets Record as Over 100K New Miners Go Online*. Obtenido de Coindesk: https://www.coindesk.com/bitcoins-computing-power-sets-new-record-as-over-100k-miners-go-online