ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADO EN INNGENIERÍA INFORMÁTICA

Monedero BTC multidireccional para Android
Android BTC wallet with multiaddress support

Realizado por
Ángel Palma Genicio
Tutorizado por
David Santo Orcero
Departamento
Lenguajes y Ciencias de la computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, Junio de 2018

Fecha defensa:
El Secretario del Tribunal

Resumen: Dentro del cambiante paradigma informático que impera en nuestro día a día, hay una nueva tecnología cuyo impacto en la sociedad puede llegar a ser equivalente al que en su día tuvo la World Wide Web.

Esta tecnología es la conocida como Blockchain. Si bien el Blockchain estuvo muchos años relegado hacia un plano más secundario por su concepción de utópico, todo cambió cuando por fin se consiguió sacarlo del plano meramente teórico para conseguir una implementación netamente funcional, el Bitcoin.

El nacimiento del Bitcoin trajo consigo una novedosa visión sobre las monedas electrónicas, lo que desembocó en un nuevo sistema de pago por internet que garantizaba la seguridad manteniendo el anonimato, lo cual sin duda supuso un hito tecnológico.

En este trabajo se parte de esta primera implementación para después hacer un estudio sobre las criptomonedas de modo genérico y del BTC más concretamente. Así mismo se analizará el conjunto de tecnologías impactadas por Blockchain, sus usos actuales, lo que supone para la informática y lo que puede resultar de seguir la misma línea de progresión ascendente.

Como consecuencia de todo este trabajo resultará el desarrollo de un monedero de Bitcoins para Android basado en tecnología Blockchain.

Palabras claves: Blockhain, Bitcoin, Android, Criptomoneda, Aplicación, Monedero, Multidirección, Cartera, Red Bitcoin, Pagos recurrentes

Abstract: According to the computer-changing paradigm that prevails every day, there's a new technology whose impact on society could be the same as the World Wide Web was.

This technology is known as Blockchain. Though many years ago Blockchain was playing a second fiddle due to his utopian conception, everything changed when it finally got out of the purely theoretical plane to become a purely functional implementation, the Bitcoin.

The Bitcoin's birth brought with it a new vision of electronic currencies, which led to a new online payment system that guaranteed security while keeping anonymity, which undoubtedly was a technological milestone.

In this work, we start with this first implementation to make a study about cryptocurrencies in a generic and BTC way more concretely. Likewise, it analyzes the set of impactful technologies by Blockchain, his current uses, what it supposes for the computer science and what can result from the same line of ascending progression.

As a result of all this work resulted in the development of a Bitcoin wallet for Android based on Blockchain technology.

Keywords: Blockhain, Bitcoin, Android, Cryptocurrency, Application, Wallet, Multidirection, Bitcoin Network, Recurring Payments

Índice general

Capítulo 1

Introducción

1.1. Introducción

Actualmente, unos de los tópicos calientes es el de las tecnologías de blockchain; que permiten realizar transacciones verificables mediante mecanismos P2P. Estas tecnologías blockchain, junto con las funciones criptográficas hash, han dado lugar a decenas de criptomonedas, cada una con sus características propias.

Las criptomonedas permiten establecer un mecanismo de intercambio de valor descentralizado, que permite que individuos realicen interacciones a un coste muy inferior al de los servicios bancarios actuales incluso entre países distintos, saltando bloqueos económicos y limitaciones gubernamentales; siendo actualmente los bancos algunos de los entes que más invierten en I+D+i en aplicaciones de la blockchain.

En concreto, el bitcoin es una criptomoneda trazable, con marca de tiempo basada en prueba de trabajo, que permite verificar de forma fuerte y distribuida las transacciones, y que permite transacciones muchos a uno. El precio de las transacciones es muy bajo; y, aunque ganó popularidad por su uso para operaciones ilegales, se utiliza de forma extensiva en muchos mercados on-line de productos y servicios, tanto legales, como de economía grís. El Tribunal de Justicia Europeo falló en 2015 que el bitcoin era otra moneda más a todos los efectos, y en España es completamente legal emitir y recibir factura en bitcoins.

El bitcoin adminte clientes ligeros, mediante el SPV –simplified payment verification–, por lo que podemos tener un monedero en un dispositivo tipo Android, sin tener que descargar la blockchain completa.

1.2. Objetivos

El objetivo de este TFG ha consistido en el desarrollo de un monedero plenamente funcional de Bitcoin para Android. Es decir, ha consistido en una App para Android que se conecta con la red de blockchain de Bitcoin, y es capaz de realizar la siguiente funcionalidad:

- Conectarse a la red Bitcoin.
- Validar fondos en la red Bitcoin.
- Mantener un listado arbitrariamente largo de direcciones de Bitcoin distintas en una misma cartera.
- Generar nuevas carteras criptográficamente válidas para la red de Bitcoin.
- Importar carteras ya existentes.
- Olvidar direcciones de la lista de carteras de la que se dispone.
- Mostrar para cada cartera el dinero en Bitcoin que esa cartera tiene disponible.
- Mostrar el agregado de dinero disponible en Bitcoin en todas las direcciones de una cartera.
- Ordenar una transferencia desde una dirección concreta, hacia otra cartera que sea válida de la red de Bitcoin.
- Ordenar una transferencia desde el grupo de direcciones, hacia otra cartera que sea válida de la red Bitcoin.
- Poner nombre –etiquetar– las transferencias realizadas.
- Validar que una transferencia se ha realizado con éxito y que esa transferencia ha sido aceptada como válida por el consenso de la red de Bitcoin.
- Pago recurrente desde una cartera: se ordena un pago recurrente desde una cartera, que se realiza con una cadencia predeterminada, realizándose solo el pago enésimo si el inmediatamente anterior cuenta con una transferencia aceptada como válida por el consenso de la red de Bitcoin.

Todas estas operaciones se validarán respecto a la red Bitcoin real, y el entregable es una cartera real, funcional, que pueda realizar transferencias monetarias efectivas dentro de la red de Bitcoin. Es decir, el entregable corresponde a un programa que implementa la funcionalidad de las carteras ya existentes para Bitcoin, más la funcionalidad adicional relativa a la gestión multicartera y a los pagos recurrentes; siendo, por lo tanto, superior a las aplicaciones ya existentes para estos menesteres.

Respecto a las fases, la investigación es de 120 horas, dada la complejidad de la tecnología, que es nueva, que está mal y pobremente documentada y que hablamos de una aplicación que se integre en la red Bitcoin real. La elaboración de memoria 60 horas, y la implementación 116.

1.3. Organización de la memoria

Hemos estructurado la memoria de la siguiente forma:

- En el primer capítulo comentamos nuestros objetivos con este trabajo fin de grado.
- En el segundo capítulo, analizamos el estado del arte de las criptomonedas y de la blockchain.
- En el tercer capítulo comentamos las tecnologías empleadas.
- En el cuarto capítulo, describimos el análisis y el diseño del sistema que implementaremos.
- En el quinto capítulo, describimos brevemente las decisiones de diseño tomadas y la implementación final, que se profundiza en el Apéndice A.
- En el sexto capítulo, comentamos las conclusiones a las que hemos llegado tras la realización de este trabajo.
- En el séptimo capítulo, añadimos la bibliografía empleada.
- Incluímos un apéndice con un pequeño manual de usuario de la aplicación desarrollada.

Capítulo 2

Estado del arte

2.1. Introducción

La evolución tecnológica no es tanto un reto para la tecnología en sí, sino que es algo que atañe a la sociedad y a la cultura de la misma. A lo largo de la historia hemos visto cómo los grandes cambios han ido irrumpiendo en nuestro día a día paulatinamente; integrar nuevos elementos en nuestras vidas -pese a que su objetivo sea el de mejorarlas y hacerlas más fácil- nunca ha sido sencillo. Nuestra primera reacción ante cualquier cambio que implique adaptar nuestro estilo vida es el rechazo. El ciudadano medio en general es reticente a aceptar la evolución tecnológica cuando esto le lleva a cambiar sus hábitos.

En este paradigma sociocultural y tecnológico se ha presentado una excepción asociado al sector bancario: el Blockchain. Esta

tecnología ha pasado de estar en una absoluta obsolescencia a acaparar un importante porcentaje de las inversiones hasta llegar a ser, a día de hoy, el mayor foco de innovación de los grandes bancos.

Según datos del portal expansion.com, Santander estima que el uso de las tecnologías blockchain puede suponer para el sector un ahorro de 20.000 millones de dólares para 2022. Y BNP Paribas va más allá y se atreve a comparar este invento con el de la máquina de vapor o el motor de combustión.

Lo que hace al Blockchain tan suculento para inversores y para los bancos es su capacidad de realizar transacciones financieras entre dos participantes de manera segura, confiable e irreversible, sin necesidad de utilizar un intermediario para establecer una relación de confianza entre las partes", como nos explican desde Grant Thornton.

En esta sección comenzaremos introduciendo el concepto que nos atañe; el Blockchain. Empezando por qué es esa tecnología, cómo se utiliza, para qué se utiliza y para qué se podría utilizar. Así mismo haremos un breve repaso por las publicaciones más recientes sobre éste tema, los marcos prácticos donde se está desarrollando y los teóricos donde se podrían desarrollar.

2.2. Blockchain

Naughton (2016) sugiere que la tecnología blockchain podría ser "la invención de TI más importante de nuestra era"; Mougayar (2016) que está .ªl mismo nivel que la World Wide Web en términos de importancia". Según Tapscott, la tecnología que tendrá el mayor impacto en las próximas décadas serán las Cadenas de Bloques (blockchains).

A grandes rasgos, el blockchain no es más que un método para registrar datos, como pudiera ser, verbigracia, un archivo Excel. Sin embargo este archivo está compartido: hay copias por toda la red y en los ordenadores de cada miembro involucrado en la creación y modificación de dicho archivo, cuyos permisos están restringidos, pues el acceso a personas ajenas a él no está permitido.

Así mismo, no es posible el borrado de la información ya creada, sino que solo permite añadir nuevos registros. Esto permite que sea el conjunto de miembros implicados en el archivo el que se encargue de la protección de los datos que contiene, y sean ellos quienes protegen la integridad del archivo y no una autoridad central.Este modo de funcionamiento permite conocer a todos los participantes los movimientos que se realicen en el archivo, así como su autor.

Por otra parte, al basarse en operaciones matemáticas, el blockchain es hasta ahora uno de los métodos más seguros que existen para crear, modificar, compartir y almacenar información, por lo que podría aplicarse a cualquier ámbito que necesitará realizar alguna de esas acciones, sobre todo si en ellas tienen que participan múltiples usuarios.

Pero, ¿qué es exactamente el Blockchain? Según el portal infotechnology.com, el blockchain (o cadena de bloques) es una base de datos compartida que funciona como un libro para el registro de operaciones. En realidad, el blockchain no deja de ser un protocolo criptográfico que se basa en la integración de ficheros relacionados en una matriz por distintos identificadores.

Estos identificadores son generados mediante combinaciones de unos deter-

minados algoritmos, en múltiples ordenadores y de forma idéntica en todos, lo que provoca que dado un número lo suficientemente elevado de usuarios existentes en el sistema, permite la identificación perfecta e irreversible del contenido añadido a esos ficheros.

Todos los bloques que conforman la cadena, tienen una contraseña (hash) que apuntan al bloque previo. Los bloques se ordenan en la cadena por orden cronológico. Además, gracias a ese hash, todos los bloques están referenciados por el bloque que los creo, por lo que solo los bloques que contienen un hash válido son introducidos en la cadena y replicados a todos los nodos. Gracias a este sistema es prácticamente imposible modificar un bloque que ha estado durante la cadena un tiempo determinado.

Los nodos "mineros"se encargan de crear los bloques que forman la cadena, añadiendo a cada uno de ellos el hash correspondiente y todas las nuevas transacciones que se han introducido en la red. De esta manera podemos decir que el blockchain nos permite llevar una “contabilidad” pública de manera totalmente transparente de todas las transacciones de la red, sin casi posibilidad de fraude, congestionamiento, ni pérdida de datos, siendo además totalmente trazable.

El uso de claves criptográficas y el hecho de estar distribuido por distintos ordenadores provoca grandes ventajas en la seguridad contra alteraciones ilícitas: una alteración en una de las copias es completamente inútil, pues cualquier cambio debe ser propagado a todas las copias existentes en la base de datos (que recordemos que es pública).

La primera virtud que se nos viene a la cabeza tras esto es que la pérdida de datos en una transacción es una posibilidad que no se contempla (o al menos no desde el punto de vista matemático, pues los riesgos son despreciables), y por consecuente también obtendrán estas ventajas los objetos asociados a dichas transacciones. Además, esta característica es bidireccional, pues identifica tanto a comprador como a vendedor, así como a objeto comprado y precio, respectivamente.

Cada operación registrada deja una huella imborrable mediante el encadenado inaccesible a los bloque previos. Como dice Emiliano Garayar en su blog "no hace falta que usted persiga al billete, pues, si es su legítimo destinatario, el billete le perseguirá a usted".

Así que, aunque empezamos definiendo las cadenas de bloques como un libro público de contabilidad, ahora podemos decir que es un elemento autenticador, equivalente a un instrumento de registro, con la particularidad de que

esta matriz es indeleble y compartida.

En éste se haya la revolucionaria ventaja del sistema de Blockchain; esa huella que deja en el registro no es falsificable ni manipulable por nadie, lo cual supone un enorme hito respecto al registro clásico. Tampoco se pueden producir valores repetidos ni problemas de venta múltiple, pues las claves criptográficas identifican de manera excelsa a las partes en este tele-registro descentralizado (distributed ledger), formado y sustanciado por la propia cadena de datos que usan claves criptográficas (crypto-keys).

Aunque ya estamos viendo que el sistema blockchain es un sistema férreo y seguro, esto no es lo que lo hace tan especial. Lo verdaderamente destacable de esta tecnología es que el contenido de los registros puede ser cualquier cosa. En el caso de Bitcoin y como veremos más adelante, este contenido son monedas emitidas como instrumentos de cambio, pero, en otros ámbitos, la tecnología que tenemos entre manos puede singularizar cualquier conjunto de datos.

En este momento empezamos a vislumbrar el potencial que tiene esta tecnología, ya que, entre otras utilidades, el blockchain se puede utilizar para probar e identificar de forma individual y segura como posible objeto, cualquier negocio o intercambio de valor. Legalmente la consecuencia inmediata de esto es la trivialidad a la hora de identificar una transacción y el objeto envuelto en ella (propiedad que proporcionan los registros públicos cuya fiabilidad es externa).

En síntesis, blockchain se presenta como un conjunto de tecnologías (P2P, sellado de tiempo, criptografía, etc.) que combinadas hacen posible que computadoras y otros dispositivos puedan gestionar su información compartiendo un registro distribuido, descentralizado y sincronizado entre todos ellos, en vez de utilizar las tradicionales bases de datos. Blockchain supone una nueva revolución tecnológica que va a cambiar, para siempre, nuestra relación con el ecosistema digital. Una tecnología con un enorme potencial para transformar los procesos de la organizaciones y generar nuevas oportunidades de negocio para las empresas.

2.2.0.1. Cómo está implementada la Blockchain

Aunque en el apartado anterior ya hemos descrito el funcionamiento de las cadenas de bloques, queremos hacer énfasis desde el punto de vista técnico para demostrar la veracidad de las afirmaciones respecto a la estructura sólida e infranqueable de esta tecnología.

Como ya hemos dicho, el blockchain no deja de ser una base de datos distribuida que registra bloques de información y los relaciona mediante hash, lo cual facilita la recuperación de dicha información y la verificación de que ésta es veraz. La cadena se va formando con los apuntadores hash que conectan el bloque actual con el anterior y así sucesivamente hasta llegar al bloque génesis.

Esta cadena de bloques se almacena en todos los nodos existentes y sincronizados en la red.

La información que posee cada bloque de la cadena hace referencia a: las transacciones relativas a un lapso de tiempo determinado, y son agrupadas en una estructura denominada Merkle Tree; el hash del bloque anterior y a un número arbitrario único llamado nonce.

La información que contiene cada bloque se registra mediante otro hash criptográfico, lo cual permite una verificación sencilla, haciendo a su vez inviable recrear la transacción de entrada. Bitcoin usa la función hash criptográfica SHA-256 lo que implica que sus apuntadores hash son de un tamaño fijo de 256 bit, como nos explican desde el blog de labaranda.yuku.com.

Las transacciones son incluidas en cada bloque de la cadena en una estructura de criptografía, el árbol Merkle. Éste árbol agrupa los bloques en pares y genera un hash por cada uno de ellos. Luego, los códigos hash generados se vuelven a agrupar en pares y generan un nuevo hash que se vuelve a agrupar con otro y así sucesivamente hacia arriba del árbol hasta llegar al nodo raíz, que se denomina apuntador hash raíz (root hash) y se registra en la dirección del bloque actual (block hash) reduciendo así el espacio ocupado por cada bloque.

Otra ventaja de usar este árbol como estructura de datos es que permite recorrer cualquier punto del árbol para verificar que los datos no han sido alterados, pues, al igual que con la blockchain, si alguien manipula algún bloque en la parte inferior del árbol, hará que el apuntador hash que está un nivel más arriba no coincida.

Incluso, si persistiese en la manipulación de dicho bloque, el cambio eventualmente se propagara a la parte superior del árbol en la que no será capaz de manipular el apuntador hash que fue almacenado por pertenecer a otra estructura (cadena de bloques) en la que también se ha generado un hash utilizando el hash raíz como entrada. Así que, de nuevo, se detectará cualquier intento de manipular cualquier pieza de datos con sólo registrar el apuntador hash en la parte superior.

2.3. Campo de aplicación de Blockchain

Las aplicaciones de Blockchain son soberbiamente amplias. En los últimos años el crecimiento de esta tecnología está siendo exponencial. Si bien muchas de estas ideas se están quedando en el plano teórico, es cuestión de no mucho tiempo el que se tornen en realidad.

El rango de autores que han trabajado en estas aplicaciones no es precisamente corto. Al ser una tecnología relativamente reciente y con un impacto en el mercado bastante considerable, han sido muchos los científicos que han orientado sus investigaciones en torno a este libro de contabilidad distribuido.

Como venimos explicando, la mayor proeza de Bitcoin fue ser el primer sistema que consiguió funcionar exitosamente apoyándose en el sistema blockchain y desechando así los fantasmas sobre lo utópico de esta tecnología. A partir de esto, se .ªbrió la veda".

La aplicación más inmediata que se nos puede ocurrir tras esta irrupción tiene que ver con el sistema financiero, pues al final y al cabo el blockchain ha nacido de la mano de Bitcoin, y está atenta principalmente contra el mercado económico.

Haciendo un repaso por las principales consecuencias que se intuye se darán en los próximos años debido al fenómeno blockchain, podemos poner como eje central el Internet del Valor (Ismael Santiago Moreno, 2016) basándose principalmente en la tecnología blockchain y las criptomonedas, que en conjunto permitirían el desarrollo de un nuevo paradigma económico.

En la actualidad estamos asistiendo a una disyuntiva en el sistema capitalista, en la cual el beneficio (y eje principal de este sistema económico), podría acabar desapareciendo. Esto supondría un nuevo paradigma económico en el que tecnologías como las cadenas de bloques (blockchains), el Big Data y el Internet de las Cosas (IoT) tomarán mucho más protagonismo. [Ismael Santiago Moreno, "La revolución de la tecnología de las cadenas de bloques y su impacto en los sectores económicos].

El Internet del Valor llegaría en conjunto con la blockchain; cualquier tipo de activo, empezando por el dinero hasta la música, podrá almacenarse, moverse, tramitarse, intercambiarse y manejarse, sin intermediarios poderosos. Estas cadenas de bloques podrán contener cualquier documento legal, además de contratos inteligentes, que podrían cambiar la manera en la que las empresas y las personas actúan y trabajan en la actualidad. [Ismael Santiago Moreno, 2016].

Por otra parte, el IoT al que antes se hacía referencia adquiere un potencial mayor si se involucra con la blockchain.

No obstante, en la introducción al fenómeno blockchain hablamos de que sus usos podrían ser tan amplios como nuestra imaginación lo permitiese, pues una tecnología así no debería restringirse al campo financiero. Sin ir más lejos, la compañía NTT DATA hizo un estudio de cómo podría aplicarse dicha tecnología, por ejemplo, al ámbito judicial. [Javier Ibáñez Jiménez, "Blockchain, ¿el nuevo notario?"].

Concretamente se preguntan cuál sería el papel del notario en un sistema automático en el cual todo está verificado y en el que no hay lugar para falsificación. En ese supuesto el reto reside fundamentalmente en centralizar la información que contiene la cadena de bloques, y teniendo en cuenta que blockchain es, por definición, un sistema descentralizado, no sería un reto fácil de superar. En cualquier caso, habría que hacer un esfuerzo legislador potente para poder compatibilizar ambos sistemas.

En este contexto la fe pública no sólo sería redundante por la digitalización contractual, es que quedaría relegada a un plano en el cual su importancia es relativamente nula. De esta manera en un plano en el que no exista comparecencia de las partes, y por tanto, no se precise estrictamente levantar acta o dar fe de conocimiento; o bien, donde por diversos motivos se repiten plenamente aseguradas ya la identidad y capacidad de las partes, el sentido de esta fe pública se diluiría en gran parte [Javier Ibáñez Jiménez, 2016].

Empezamos a ver cómo el impacto de blockchain es mayúsculo sea cual sea el sector implicado. Posiblemente esto sea un arma de doble filo si tenemos en cuenta las históricas reticencias de las sociedades en general a aceptar grandes cambios, como ya comentábamos anteriormente.

Otro prometedor uso de la cadena de bloques es el que hace referencia a los sistemas de voto electrónico. Sin duda debería ser un campo en el que el cuerpo político debería hacer mayor incisión si realmente la intención es que la tasa de votación crezca. En este sentido, y tomando como ejemplo los sistemas ya implementados de votación electrónica centralizados, se plantea el uso de la tecnología blockchain para proponer una solución que dé respuesta a los requisitos de voto electrónico, y que represente una mejora respecto a los sistemas actuales basados en tecnologías centralizadas de almacenamiento y procesamiento de las transacciones. [Antonio Marín Bermúdez, "Blockchain e-voting"].

En esta comparativa de los sistemas actuales de voto electrónico se hace es-

pecial hincapié en que el hecho de que todas las soluciones propuestas hasta ahora cojean de la misma pata; necesitan una autoridad central que la gestiones, el cual posiblemente sería administrado por alguno de los participantes de la contienda electoral.

La aplicación de la tecnología blockchain en un campo así se antoja evidente. La solución que proponen en el mencionado trabajo supone un sistema seguro, descentralizado y transparente. Más allá de la comodidad para el electorado de poder votar por esta vía, habría que tener en cuenta el imponente ahorro que supondría eliminar -o reducir- el sistema de voto tradicional. La implantación de una solución así no supondría un sobrecoste considerable. Además, permitiría a cualquier interesado añadir su propio nodo a la red, formando parte del sistema a la vez que auditando el mismo. [Antonio Marín Bermúdez, 2016].

Bien es cierto que comercialmente hablando los dos ejemplos anteriores son más llamativos de cara a un ciudadano corriente que, por ejemplo, las aplicaciones de blockchain a ámbitos estrictamente tecnológicos y cuya repercusión queda limitada al campo de la investigación. Pero no por ello no se han hecho estudios en esta dirección.

Un ejemplo de ello es la utilización de blockchain para crear juegos. En este sentido sería posible la implementación de algunos juegos a partir de la tarea de los mineros en el proceso de validación de bloques antes de añadirlos a la blockchain. [Aggelos Kiayias, "Blockchain Mining Games"].

A partir de un estudio de las consideraciones estratégicas de los mineros que participan en el protocolo de bitcoin, se puede formular un juego estocástico que subyace a partir de éstas consideraciones estratégicas. Los mineros colectivamente construyen un árbol de bloques, y se les paga cuando crean un nodo (que terminará en el camino del árbol que es adoptado por todos).

Dado que los mineros pueden esconder nuevos nodos minados, juegan a una especie de juego con información incompleta. Aquí se pueden considerar dos formas simplificadas de este juego en las cuales los mineros tienen información completa: en el juego más simple, los mineros liberan inmediatamente todos los bloques minados, pero siguen una estrategia para decidir qué bloques minan

En el segundo juego más complicado, cuando se extrae un bloque, se anuncia de inmediato, pero no puede ser liberado para que otros mineros no puedan seguir minando de él. Un minero no sólo decide qué bloques de mina, sino también cuándo liberar bloques a otros mineros. En ambos juegos, se puede observar que cuando la potencia computacional de cada minero es relativamente peque-

ña, su mejor respuesta coincide con el comportamiento esperado del diseñador de bitcoin. Sin embargo, cuando el poder computacional de un minero es grande, se desvía del comportamiento esperado y surgen otros equilibrios. [Aggelos Kiayias, "Blockchain Mining Games", 2016].

Sin embargo este juego tampoco es el uso más raro que se le puede dar a esta polivalente tecnología. La capacidad creativa de ciertos investigadores siempre va un paso más allá. Tal es así que podemos ver cómo la tecnología de la cadena de bloques puede ser enfocada incluso

Regresando parcialmente al sector legal, hay otro uso de la tecnología de cadena de bloques cuyo valor en un futuro puede ser alto. Éste no es otro que el tema de las patentes. Concretamente, las patentes en la industria musical. Si tenemos en cuenta que desde la irrupción de Napster este mercado se ha encontrado en una controversia constante, el hecho de que llegase una tecnología así a acabar con los históricos problemas debería ser una prioridad para los artistas envueltos en dichas polémicas. [Middlesex University London, "Music on the Blockchain"].

Tanto en los medios de comunicación como en los eventos de la industria, ha habido una explosión de interés en los últimos meses en el impacto potencial de la tecnología blockchain en las industrias de la música, en particular las asociadas con la música grabada.

El grupo Blockchain para Creative Industries de la Universidad de Middlesex destaca varias áreas en las que blockchain parece tener un potencial transformador para la música grabada, como por ejemplo una base de datos en red para la información de copyright de música, o facilitar los pagos de regalías rápidos, ofrecer transparencia a través de la cadena de valor... [Middlesex University London, 2016].

Teniendo en cuenta que las industrias de la música tienen un valor estimado de 45.000 millones de dólares a nivel mundial, de los cuales la industria discográfica (a efectos prácticos lo podemos equiparar con la música grabada) es responsable de aproximadamente 15.000 millones de dólares (Rethink Music 2015), pensar en un sistema que pueda solucionar el problema de las patentes para un claro acierto.

A pesar de que la era de las descargas legales y, más recientemente, el streaming puede parecer en algunos aspectos una mejora en la era inicial de la piratería del consumo digital (Watson 2015), el cambio hacia un modelo basado en el acceso más que en la propiedad trae consigo diversos desafíos. La reciente

explosión de interés en la tecnología de cadenas de bloque es vista por algunos como una solución a los desafíos que enfrenta la industria discográfica. [Middlesex University London, "Music on the Blockchain"].

Para finalizar, es conveniente señalar que, pese a ser una tecnología puntera, depurada y con un futuro realmente esperanzador, pueden surgir ciertos problemas de carácter técnico asociados al funcionamiento de blockchain. Esto sucede debido a que establecemos como verdaderos ciertos supuestos que, en principio, no tienen porqué ser ciertos. [Christopher Natoli, "The Blockchain Anomaly"].

Las soluciones de bloques de bloques más populares, como Bitcoin, se basan en pruebas de trabajo, garantizando que la salida es un consenso acordado con alta probabilidad.

Sin embargo, esta probabilidad depende de la entrega de mensajes y de que la potencia computacional del sistema está suficientemente dispersa entre las agrupaciones de nodos de la red, de modo que ninguna piscina pueda extraer más bloques más rápido que la multitud.

Nuevos enfoques, como Ethereum, generalizan el enfoque de prueba de trabajo al permitir que los individuos despliegan su propia cadena de bloqueo privada con alto rendimiento de transacciones.

A medida que las empresas están empezando a desplegar cadenas privadas, se ha vuelto crucial para comprender mejor las garantías que ofrecen blockchains en un entorno tan pequeño y controlado. [Christopher Natoli, "The Blockchain Anomaly"].

Así nace la llamada anomalía Blockchain (Christopher Natoli, 2016), una ejecución que se experimenta al construir una cadena privada en NICTA / Data61. A pesar de que esta anomalía nunca se ha reconocido antes, puede traducirse en consecuencias dramáticas para el usuario de blockchains.

Nombrado después de la infame anomalía de Paxos, esta anomalía hace que las transacciones dependientes, como "Bob envía dinero a Carole después de que recibió dinero de Aliceïmposible. Esta anomalía se basa en el hecho de que las cadenas de bloque existentes no garantizan la seguridad del consenso de forma determinista: no hay manera para que Bob se asegure de que Alice le envió realmente monedas sin Bob usando un mecanismo externo, como convertir estas monedas en una moneda fiduciaria que le permite retirar. [Christopher Natoli, 2016].

En definitiva, el uso de blockchain debería suponer un gran avance no sólo para el sistema financiero, si no para otros ámbitos como el notarial o el sistema

de patentes musicales. Del mismo modo otros usos subyacentes como podrían ser los smart contracts o el e-voting deberían ir implantando poco a poco. Sin duda el tiempo será el que dicte sentencia sobre si todo esto quedará como una utopía o se impondrá la razón.

2.4. Criptomonedas

El nacimiento de blockchain trajo consigo el nacimiento de un nuevo fenómeno; las criptomonedas. Y lo hizo mediante Bitcoin como veremos más adelante. En este apartado simplemente trataremos de explicar a grandes rasgos qué es una criptomoneda o moneda virtual.

El concepto de "Moneda Criptográfica"fue descrito por primera vez en 1998 por Wei Dai en la lista de correo electrónico Çypherpunks1", donde propuso la idea de un nuevo tipo de dinero que emplearía la criptografía para controlar la creación y las transacciones, en lugar de una autoridad centralizada. Cypherpunks fue el predecesor de Satoshi Nakamoto, quién anunciaría al mundo el Bitcoin.

Según el Banco central Europeo (BCE), una moneda virtual es ün tipo de dinero no regulado, digital, que se emite y por lo general controlado por sus desarrolladores, y utilizado y aceptado entre los miembros de una comunidad virtual específica". Esto en primera instancia nos da una aproximación al concepto, aunque también nos invita a pensar acerca de la legalidad de ésta, a lo cual le dedicaremos un apartado más adelante.

Por otra parte, Nakamoto (2008) define el concepto de moneda electrónica como una cadena de firmas digitales. Una moneda digital, sin embargo, es una forma de moneda virtual que se crea y se almacena electrónicamente, lo cual nos da paso a la definición de criptomoneda.

Las criptomonedas son un específico tipo de monedas digitales que operan de manera independiente a un banco central, basadas en la criptografía y que son usadas para asegurar los pagos que se intercambian. En su origen estas divisas utilizan una red distribuida para permitir el pago P2P (entre pares) y garantizar la seguridad. Este sistema no es otro que el blockchain, que como ya hemos explicado garantiza que una transacción sea legítima.

2.5. Bitcoins

Teniendo en cuenta que el trabajo que fundamentalmente nos ocupa es el blockchain y que su máximo exponente es el Bitcoin, es importante estudiar en profundidad todos los aspectos de éste, ya que el resto de criptomonedas toman como base a esta pionera divisa virtual.

Bitcoin pertenece al grupo de criptomonedas anteriormente descrito. Es la moneda virtual más popular y solamente está disponible en internet. Lanzada en 2009 por "Satoshi Nakamoto", está construida sobre un diseño descentralizado y protegido por criptografía potente (Caetano, 2015).

Según publicita su propia página web, “Bitcoin es una innovadora red de pagos y una nueva clase de dinero", basada en una moneda virtual e intangible, que üsa tecnología peer-to-peer o entre pares para operar sin una autoridad central o Bancos”; esto es, "la gestión de las transacciones y la emisión de bitcoins es llevada a cabo de forma colectiva por la red".

Como cualquier moneda, Bitcoin proporciona un sistema de pago -en este caso electrónico- que además es anónimo, pues se persigue con celo el mantener la privacidad de los usuarios.

En el aspecto monetario, hay que destacar que esta divisa tiene valor real, esto es, fuera de internet, pues es posible el cambio a moneda de curso legal. Su valor es directamente proporcional a la confianza que tienen las personas en dicha moneda.

El hecho de que no exista una autoridad central que regule el sistema hace que Bitcoin tenga un comportamiento económicamente auto-reglamentado. Esto no sería posible sin usar el famoso libro de contabilidad distribuido, o, dicho de otra manera, sin blockchain.

Por tanto, podemos deducir que el proceso que sigue es similar al explicado en blockchain a partir del cifrado hash que se obtiene mediante un método que relaciona dicha transacción con el registro anterior.

El proceso de generar un bloque de cada transacción se denomina prueba de trabajo (PoW) y es realizado por los validadores o mineros. Cada vez que se guardan nuevos bloques, estos tienen mayor complejidad para su procesamiento y guardado.

Teniendo en cuenta que a día de hoy el bitcoin es la criptodivisa más importante de todas, es necesario mencionar que se considera como una especie de bien propio o dinero privado, lo cual será tratado de diferentes maneras aten-

diendo a la regulación vigente de cada país.

Otro aspecto reseñable es el que destaca la firma Juniper Research, que predice que las transacciones Bitcoin se triplicarán en 2017. Para llevar a cabo esta deducción parte de factores como la fragilidad de la economía china, las consecuencias del Brexit en Europa, además de otros problemas como el desempleo y políticas económicas desacertadas.

Por último, según Bhagwan Chowdhry, el bitcoin tiene muchas ventajas sobre el papel físico y las monedas: es seguro, porque se basa en un código criptográfico casi inviolable y rompe el circuito de los gobiernos, bancos centrales, los intermediarios financieros, eliminando retrasos y los costes de las transacciones (BBVA Innovation Center, 2016).

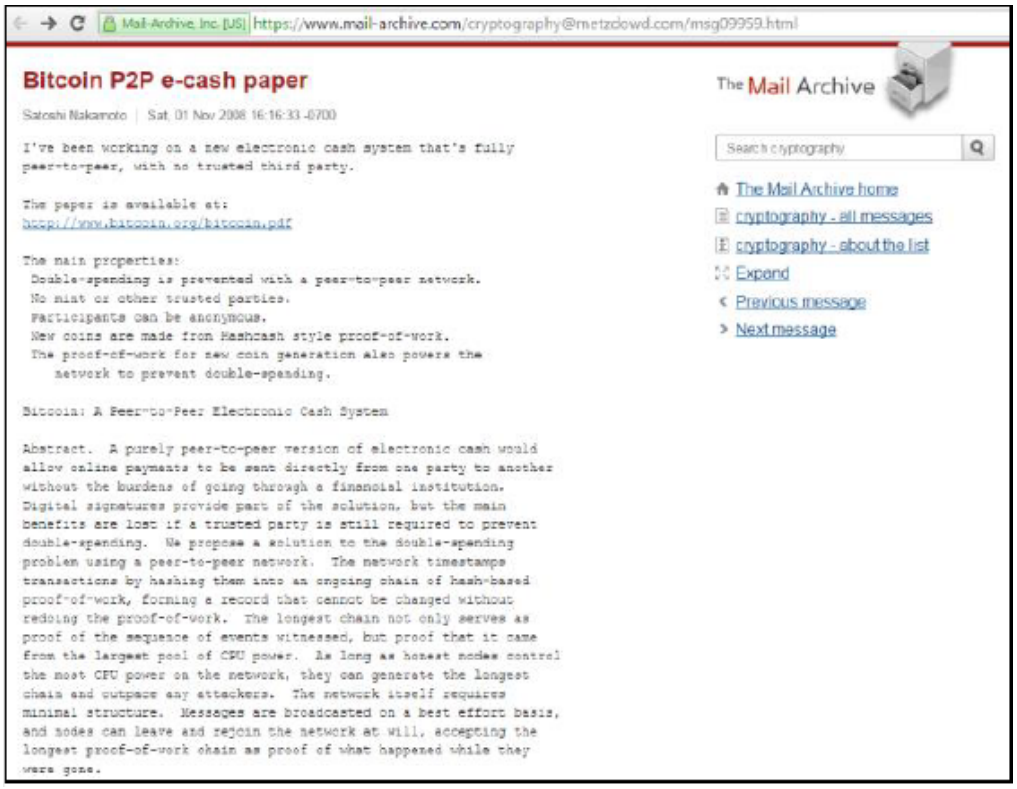
2.5.1. Cómo surge Bitcoin

Bitcoin es la implementación de lo que se conocía como "b-money.º çriptomoneda", un concepto sacado de las conversaciones que mantenían por mail su creador, Wei Dai, con sus amigos informáticos, allá por 1998.

Este concepto de moneda criptográfica sugería la creación de un sistema anónimo y distribuido de dinero electrónico (.ªnonymous, distributed electronic cash system") que pudiera operar sin necesidad de los agentes centrales del sistema financiero tradicional y de los típicos intermediarios en las transacciones.

Algunas de las ideas que tenía Wei Dai se traducen en lo siguiente: .En una criptoanarquía, el gobierno no se destruye de forma temporal sino que permanentemente está prohibido y es innecesario". En esta frase sin saberlo ya estaba introduciendo el concepto de blockchain y su extensión a todos los ámbitos de la vida que necesitasen una tercera parte confiable.

Sabía que el establecimiento de este sistema supondría un cambio histórico a nivel político y social: "Hasta ahora no está claro, incluso teóricamente, cómo podría funcionar tal comunidad. Una comunidad se define por la cooperación de sus participantes, y la cooperación eficiente requiere un medio de cambio (dinero) y una forma de hacer cumplir los contratos. Tradicionalmente, estos servicios han sido prestados por el gobierno o las instituciones patrocinadas por el gobierno, y sólo a entidades legales. En este artículo describo un protocolo por el que estos servicios pueden ser prestados a entidades no rastreables". [Wei Dai, "b-money"http://www.weidai.com/bmoney.txt, 1998].



Bitcoin P2P e-cash paper

Satoshi Nakamoto | Sat, 01 Nov 2008 16:16:33 -0700

I've been working on a new electronic cash system that's fully peer-to-peer, with no trusted third party.

The paper is available at:
http://www.bitcoin.org/bitcoin.pdf

The main properties:
Double-spending is prevented with a peer-to-peer network.
No mint or other trusted parties.
Participants can be anonymous.
New coins are made from Hashcash style proof-of-work.
The proof-of-work for new coin generation also powers the network to prevent double-spending.

Bitcoin: A Peer-to-Peer Electronic Cash System

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without the burdens of going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as honest nodes control the most CPU power on the network, they can generate the longest chain and outpace any attackers. The network itself requires minimal structure. Messages are broadcasted on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

Figura 2.1: Figura:Email enviado por Nakamoto en 1998

La tarea de la creación y gestión de un sistema monetario históricamente ha recaído en los gobiernos y autoridades centrales. El problema de esto es que al final dependemos del sentido de la ética y el sentimiento de empatía que los gobernantes y encargados de estos menesteres puedan tener sobre el pueblo. Por tanto, antes que confiar en la buena fe de unos pocos, lo que Wei Dai sugería en la anterior declaración es un sistema en el cual ningún organismo central pueda ejercer el control, sino que dicho poder resida en el conjunto de todos los usuarios participantes en el sistema (que en este caso sería distribuido).

Además, el sistema descrito tendría la ventaja del anonimato, ya que cada participante tiene una clave de manera tal que las transacciones en todo momento sean asociadas a un monedero pero sin necesidad de que exista una forma de relacionar cada monedero con una persona física o jurídica.

Wei Dai en su correo electrónico habla de dos protocolos para llevar a cabo la creación de este sistema monetario virtual. En el caso del primero el mismo autor lo da por imposible por inviable. No obstante, será éste protocolo el que impulse el segundo propuesto.

En el caso del primero cada usuario que participa en el sistema registra en

una base de datos la cantidad de dinero que pertenece a cada participante bajo su seudónimo, y cada vez que se realiza una transferencia se actualiza la base de datos, emulando el comportamiento de cualquier base de datos. El 'quid' de la cuestión estará en cómo se actualiza esta base de datos.[Wei Dai "b-money"]. Si se pretende eliminar la autoridad central teóricamente necesaria, debe existir acuerdo entre las partes. Y en este protocolo cualquier persona puede crear dinero resolviendo una serie de problemas computacionales.

El segundo protocolo modifica al primero de manera que, en lugar de ser el conjunto de todos los usuarios los que intervienen en la actualización de la base de datos, sólo es un subconjunto de los participantes -a los que se refiere como servidores- los que se encargan de validar las transacciones. Transacciones que, tan pronto sean aprobadas, deben ser propagadas al resto de la red de usuarios. La intención primordial es la de generar confianza, por lo que el requisito principal es el de algún mecanismo que haga a los servidores comportarse de manera honesta, en beneficio de la red.

Wei Dai en su propuesta ofrece una serie de posibles soluciones a estos requisitos, las cuales en primera instancia se basaban en hacer depositar a cada usuario una cierta cantidad de dinero en una cuenta especial. De este modo, en caso de incumplimiento de normas o en caso de conducta inapropiada, este fondo común podría ser utilizado para multar al usuario implicado.

Así mismo, cada servidor se comprometería a hacer público periódicamente sus bases de datos actuales de creación de dinero y la propiedad del mismo. Además, cada usuario partícipe en el sistema debería certificar que sus propios saldos de cuentas son correctos y que la suma de los saldos de las cuentas no es mayor que la cantidad total de dinero creado. [Wei Dai, 1998]

Como primera toma de contacto con lo que serán las criptomonedas y el sistema monetario virtual, estuvo correcto. No obstante, el modelo queda lejos de estar libre de fallos. En cualquier caso, la intención del mail de Wei Dai era la de plantear el concepto a modo de boceto, de modo que supusiera el punto de partida para el desarrollo del idealizado sistema.

Acorde con las propias palabras de su creador, .[el] protocolo propuesto en este artículo permite a entidades con seudónimos imposibles de rastrear cooperar entre ellas de manera más eficiente, proporcionándoles un medio de intercambio y una manera de hacer cumplir los contratos. El protocolo probablemente se pueda hacer más eficiente y seguro, pero espero que esto sea un paso para hacer de la cripto-anarquía una posibilidad tanto práctica como teórica."[Wei Dai, "b-

money", 1998]

Esta teoría constituye la base de la creación de bitcoin y blockchain en los años posteriores. Tan importante como hacer mención a los futuros usos que le podremos dar a la cadena de bloques es hacer mención a este mail de Wei Dai, quien sembró la semilla de lo que posiblemente sea el sistema que revolucionará todo el sistema financiero y político tal y como lo conocemos, pues esa idea de Wei Dai de eliminar a las autoridades centrales y pese a ello generar confianza, ya es realidad.

2.5.2. Funcionamiento

A continuación explicaremos paso a paso cómo funciona Bitcoin de cara a un usuario común:

1. Acto después de registrarse, cada usuario nuevo deberá elegir un monedero y a continuación instalarlo en su correspondiente dispositivo. Éste wallet tiene una clave única (guid) que se usa para la verificación de la identidad del usuario mediante la firma digital con dicha clave.

2. El primer paso da lugar a una dirección de bitcoin (que debería usarse una sola vez), que será enviada a otros usuarios con los que se quiera establecer algún tipo de transacción.

3. Estas transacciones son verificadas gracias al registro de contabilidad, es decir, al blockchain, que ofrece el conjunto de transferencias confirmadas y corrobora que el usuario que va a realizar la transferencia tiene al menos tantas monedas como pretende gastar.

4. Mediante la minería (proceso que estudiaremos en el siguiente apartado), las transacciones son transmitidas y confirmadas para su posterior inclusión en la cadena de bloques. Como veremos a continuación, es este proceso el que se asegura de que la cadena sigue un orden cronológico y protege la neutralidad de la red.

2.5.3. Minería

El proceso de minería mencionado anteriormente y mediante el cual se crean los bitcoins, se basa en hallar soluciones a un problema de carácter matemático

en el cual se procesan transacciones bitcoin. Cada 10 minutos según estadística se consigue verificar y registrar transacciones, recompensando al minero con nuevos bitcoins.

Para registrar exitosamente un bloque de transacciones, blockchain se ajusta dinámicamente de forma que, en promedio, algún nodo será exitoso cada 10 minutos independientemente de cuántos mineros hayan trabajado en la tarea. El valor límite es de 21 millones de monedas. Se prevé que el número de bitcoins en circulación alcance su límite en el año 2140 en cual llegaría a 20,999,999.9769 BTC, debido a que su circulación sigue una curva predecible.

2.5.4. Proof of work (PoW), valor computacional y forks

Proof of work es el principal componente de Bitcoin. Su principal función es la de garantizar que la red se comporta de manera legítima. Computacionalmente hablando, el PoW es un dato especialmente costoso de conseguir aunque fácil de corroborar a partir de ciertos requisitos.

Es calculado por una función *hascash*, que es una función para la ejecución de la prueba de trabajo que se basa en el costo del uso de la CPU y que devuelve un token (Back, 2002). Se utiliza para garantizar la autenticación de la fuente de la transacción y para guardar datos en el Blockchain.

La función Hascash trabaja con cualquier algoritmo criptográfico. La combinación de la criptografía y la tecnología Blockchain evita la duplicación del registro de transacciones. Para obtener una prueba PoW es necesario la participación de un grupo de mineros que tienen los recursos computacionales adecuados. La moneda del contexto utiliza algunas recompensas para "pagar" el esfuerzo realizado por esta validación (Ramzan, 2013).

El uso de Hashcash sirve para la prevención del spam de correo electrónico, requiriendo una prueba de trabajo en el contenido del email (incluyendo la dirección) de cada correo. Los correos electrónicos legítimos serán capaces de hacer el trabajo para generar la prueba con facilidad (no se requiere mucho trabajo para un solo correo electrónico), pero los que envían correos de spam masa tendrán dificultades para generar las pruebas necesarias (lo que requeriría enormes recursos computacionales).

Por otra parte, es necesario analizar dos cuestiones subyacentes al esquema de minería que usamos en el protocolo Bitcoin. La primera de ellas es la referida al poder computacional. Pese a que no es un factor que normalmente se tenga en

cuenta en los análisis de Bitcoin, indirectamente afecta mucho al funcionamiento de ésta.

En el sistema monetario tradicional cualquier individuo tiene la posibilidad de acudir al cajero automático y extraer dinero para sus operaciones habituales. En dicho proceso el banco como autoridad central pagaría por la energía consumida por mantener el cajero operativo y por la regulación de las extracciones de dinero.

En Bitcoin también se precisa de cierto poder computacional que mantenga la red funcional en todo momento. Emulando a la red de cajeros del sistema tradicional, en Bitcoin están los mineros, que también se necesita que estén validando transacciones y añadiendoles a la blockchain tras crear el pertinente bloque. Es mediante este sistema de prueba de trabajo como se consigue la referida actividad computacional.

Como incentivo para que la red de mineros esté siempre funcional, éstos reciben recompensas en función del número de transacciones operadas. Veamos a continuación qué significa doble gasto y cómo lo gestiona Bitcoin.

Nuevamente vamos a hacer un paralelismo con el sistema monetario tradicional. En un intercambio de efectivo entre un cliente y una tienda, el primero provee el dinero y el segundo entrega a cambio el producto requerido. Por tanto, el dinero utilizado solo puede existir en un sólo lugar.

En una extracción de efectivo en un banco, la entidad bancaria hace una comprobación sobre el saldo de la cuenta antes de aprobar la retirada. En cuanto es aprobada, se modifica el estado de la cuenta para evitar un doble gasto (usar el mismo dinero dos veces).

En la red Bitcoin ya hemos hablado de que no existe esta autoridad central que haga esos menesteres. Lo que posibilita que un usuario no registre dos transacciones distintas con el mismo dinero es el trabajo y el consenso de los mineros; aunque llegase una bifurcación con dos transacciones basadas en el mismo dinero, sólo una de ellas podría ser aprobada.

Aprovechando la tesitura de esta situación de doble gasto, procedemos a analizar qué ocurre con estas bifurcaciones. Las bifurcaciones (fork) no solo pueden ser producidas por problemas de doble gasto, sino que podría darse la situación de que dos mineros obtienen la solución de un bloque relativamente al mismo tiempo.

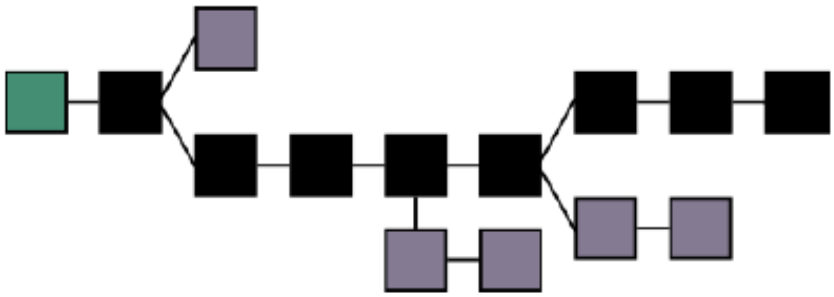


Figura 2.2: Figura: Arquitectura fork

Al enviarse a la red, el resto de nodos deben tomar la decisión de qué cadena procesar. En este contexto, el poder de procesamiento equivaldría a votos, puesto que la cadena de mayor longitud, esto es, mayor esfuerzo computacional, se impondría.

Por último, es conveniente resaltar que las transacciones que se encuentran en bloques de la cadena más corta, se transforman en transacciones pendientes para su inclusión en un bloque de la cadena principal.

A grande rasgos, podemos ver cómo funciona el protocolo de BTC a la hora de realizar un pago en la siguiente imagen:

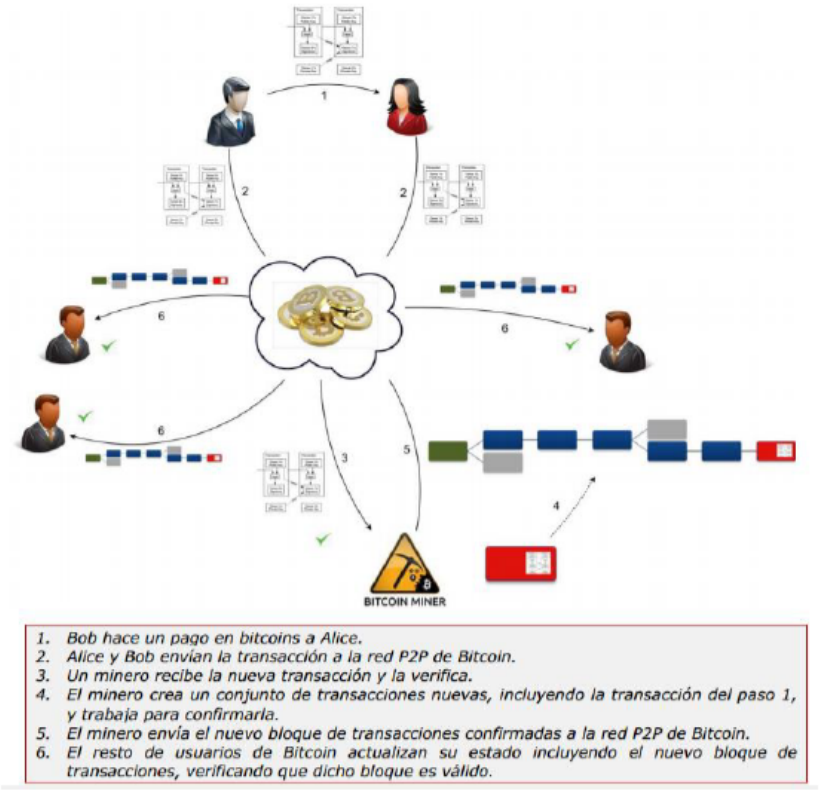


Figura 2.3: Figura: Protocolo de Bitcoin

2.5.5. Legalidad Bitcoin

¿Bitcoin es legal? Esta pregunta es una de las más recurrentes en torno a esta criptomoneda, especialmente para los nuevos usuarios generalmente escépticos para con estas tecnologías.

Aunque en rigor la respuesta sería un rotundo sí, bien es cierto que es un tema que ocupa el centro de toda la polémica. No sólo Bitcoin, sino también blockchain. Es una idea tan buena que lleva el anonimato al límite, lo que posibilita la transferencia anónima de dinero ilícito.

Según la propia web de Bitcoin, la moneda no se ha hecho ilegal por ley en la mayoría de territorios. Sin embargo, algunos territorios (como Argentina o Rusia) restringen o prohíben monedas extranjeras de manera severa. Otros territorios (como Tailandia) pueden limitar la concesión de licencias a ciertas entidades como son las casas de cambio de Bitcoins.

Reguladores de varios territorios están tomando medidas para proveer a individuos y negocios con reglas acerca de cómo integrar esta nueva tecnología al

regulado y convencional sistema financiero. Por ejemplo, la Red de Protección de Crímenes Financieros (FinCEN), una oficina del Ministerio de Hacienda de Estados Unidos, emitió una guía sobre cómo caracterizan ciertas actividades que involucran monedas virtuales.

Por otra parte, en los últimos meses hemos visto cómo en nuestro país hacienda volvía a levantar la liebre del BTC. Según El Confidencial, la agencia tributaria española lleva tiempo trabajando en controlar eso del 'blockchain' y, .[es]pecialmente, las criptomonedas", de tal manera que en abril de este mismo año, la Oficina Nacional de Investigación contra el Fraude (ONIF) 'disparaba' esta pasada semana un total de 60 peticiones de información a entidades bancarias, intermediarios, casas de cambio y, sobre todo, empresas que aceptan en internet cobros en esta divisa digital.

La polémica estaba servida, pues como decían fuentes de la investigación "disfrazan esto de tema impositivo pero en realidad con lo que andan mosqueados es con el blanqueo. Quieren meter mano y establecer mayores controles pero no están muy seguros de cómo lo tienen que hacer".

Fuentes de la Agencia Tributaria señalan que el objetivo de estos requerimientos masivos a decenas de empresas es doble: por un lado, analizar los flujos de ingresos realizados con bitcoins por encima de ciertas cantidades (hasta varios millones de euros) y, por otro, identificar el origen de los pagos realizados con criptomonedas para detectar posibles blanqueos de capitales.

Lo cierto es que como confiesa uno de los investigados, en el sector "hay cierta inquietud. Los usuarios muchas veces no tienen claro cómo tributar. Esto no ayuda a normalizar la situación en torno a las criptomonedas y genera nervios. Cuando se enteran de este tipo de colaboraciones muchos sueltan la puya de que no cuidas la privacidad de tus clientes y tal...".

"La propia naturaleza del bitcoin dificulta muchísimo que se puedan rastrear operaciones. Por eso señalan a casas de cambio o intentan localizar a la gente que opera con ellos a través de sus compras", opina otra de las plataformas contactadas, que también prefiere mantener en el anonimato y que facilita una copia de la comunicación del fisco. .[En] las conversaciones que he tenido con la Agencia Tributaria, les he visto muy verdes y lejos de poder llevar una tutela efectiva", concluye.

Como vemos, el tema fluctúa en función de la importancia de la moneda, pero lo único cierto es que a día de hoy las autoridades aún están muy lejos de poder prohibir mundialmente la criptomoneda.

2.5.6. Problemas de seguridad

El protocolo Bitcoin fue introducido originalmente en [Nakamoto, 2008] y fue construido sobre la base de las ideas de [Back, 1997] y [Dai, 1998]. Después de la creación de Bitcoin otras varias monedas criptográficas siguieron la ruta marcada por la pionera. Éstas son conocidas como altcoins (http://altcoins.com/), por ejemplo, litecoin, Primecoin, etc.

Una gran mayoría de trabajos anteriormente introducidos, examinan posibles tipos de ataques contra el protocolo Bitcoin y sugiere adaptaciones del protocolo para garantizar su seguridad. Brevemente mencionamos algunos de estos trabajos aquí.

Los ataques exitosos relacionados con la minería de piscinas se discuten en [Rosenfeld, 2011] y [Courtois y Bahack, 2014]. En [Eyal, 2014], el autor considera que los ataques se realizan entre distintas piscinas donde los usuarios son enviados a filtrar un pool competitivo dando raise a un juego de billar.

[Kroll et al., 2013] considera un ataque que puede ser realizado por personas que sólo están interesadas en destruir Bitcoin, a diferencia de otros ataques realizados por usuarios que intentan aumentar su recompensa esperada. Esto se llama el ataque de Gold nger. [Heilman et al., 2015] se centra en la red peer-to-peer y examina los ataques de eclipse donde el atacante (a) aísla a un nodo / usuario de la red y lo obliga a perder su poder computacional participando así en un ataque Sin siquiera ser conscientes.

Ciertos ataques de desanonimización también han sido observados recientemente [Meiklejohn et al., 2013] analizando el gráfico transaccional (ver también [Fergal Reid, 2012] y [Ron y Shamir, 2013]).

En [Sompolinsky y Zohar, 2015] se describe un método alternativo de consenso, llamado Greedy Heaviest-Observed Sub-Tree o GHOST. Una variante de GHOST ha sido adoptada por Ethereum, una plataforma de aplicaciones distribuidas que se construye en la parte superior de las cadenas de bloque. [Eyal et al., 2014] intenta superar los problemas de escalabilidad que surgen en Bitcoin (tamaño de bloques e intervalo vs. latencia y estabilidad) al proponer un nuevo protocolo de bloqueo escalable. [Poon y Dryja, 2015] proporciona otra alternativa para la escalabilidad que utiliza o? Mientras se utiliza el ledger distribuido para mantener los contratos entre las partes que participan en el o? Transacciones en cadena.

En [Garay et al., 2015] los autores analizan el protocolo Bitcoin en profun-

didad. Ellos resumen el núcleo del protocolo que denomina backbone bitcoin y demuestran formalmente sus principales atributos y propiedades, que pueden ser utilizados como bloques de construcción para lograr objetivos que no sean simplemente mantener un libro mayor público. Muestran que cuando el retardo de propagación en la red es relativamente pequeño, una mayoría honesta de usuarios es suficiente para garantizar un funcionamiento sin problemas en un modelo criptográfico donde la racionalidad de los jugadores no se considera.

El libro blanco de Bitcoin proporciona un análisis probabilístico de los ataques de doble gasto algo escueto. No obstante, se puede encontrar un análisis más detallado en [Rosenfeld, 2014]. Los desafíos de diseño e investigación de Bitcoin se discuten en [Bonneau et al., 2015] junto con una presentación de la investigación existente.

En [Tschorsch y Scheuermann, 2015] se puede encontrar una extensa y más introductoria encuesta sobre las criptomonedas distribuidas. Las obras más relevantes al respecto son [Kroll et al., 2013], [Eyal y Sirer, 2014] y [Sapirstein et al., 2016].

En [Kroll et al., 2013], se consideran los equilibrios del sistema Bitcoin. Los autores observan que cualquier estrategia monótona es un equilibrio de Nash (uno de muchos). Su análisis, sin embargo, es bastante restringido: como se muestra, incluso en el caso del sistema de liberación inmediata, Frontier no es la mejor respuesta para valores por encima de 0:455.

En [Eyal y Sirer, 2014], los autores demuestran que una mayoría garantizada de mineros honestos, es condición necesaria pero no suficiente para garantizar la seguridad del protocolo Bitcoin.

En particular, presentan una estrategia específica denominada estrategia de "Sel Sh Minez examinan cuando esta estrategia es benéfica para un grupo de mineros. Parece que una fracción de 1/3 del poder de procesamiento total es siempre suficiente para una piscina de los mineros, en beneficio de la aplicación de la estrategia de Sel Sh Min sin importar las características de propagación del bloque de la red.

Por lo tanto, esto constituye un ataque de tabla pro contra el protocolo Bitcoin. Además, cuando la propagación de bloques está favoreciendo al atacante, el umbral por encima del cual "Sel Sh Mine.[es] beneficioso es cualquier valor distinto de cero.

No es difícil ver que la estrategia de "Sel Sh Mine"no explora completamente la configuración donde la propagación de bloques es más favorable al atacan-

te, lo que se observó en [Garay et al., 2015], donde un ataque óptimo contra la propiedad de la cadena se consideró en su entorno donde la red se considera completamente contradictoria.

En [Sapirstein et al., 2016], los autores consideran un conjunto más amplio de posibles estrategias que incluyen la estrategia de "Sel sh-Minez exploran este espacio computacionalmente.

No es difícil vez como el protocolo Blockchain también presenta ciertos aspectos con vulnerabilidades de distinta índole. Pese a todo lo repasado en esta sección, lo cierto es que a día de hoy su robustez sigue siendo infranqueable y la seguridad que ofrece el comercio electrónico con BTC mejora por mucho a la que ofrecen otros servicios de banca electrónica. Hasta la fecha, ningún ataque a la cadena de bloques ha conseguido ser exitoso, de ahí que todos los trabajos en esta sección mencionados hayan quedado relegados al plano puramente teórico.

2.5.7. Blockchain y Bitcoins

Como ya hemos dicho, el Blockchain es el mecanismo tecnológico utilizado para el funcionamiento de Bitcoin. En este caso estas cadenas de bloques contienen las transacciones de Bitcoins que están validadas en orden cronológico, de tal forma que cada vez que un bloque es confirmado por la red, este pasa a ser parte de la cadena.

El primer registro de la blockchain de Bitcoin se hizo el 3 de enero de 2009. A partir de este momento cuando un nodo de la red consigue crear un nuevo bloque, se añade a ésta cronológicamente y éste lo transmite al resto de nodos para que verifiquen que el bloque es correcto.

A partir de aquí, en caso de que se confirme, se añade a la cadena y se difunde. En este proceso la seguridad está garantizada porque todas las transacciones son públicas y automatizadas por algoritmos matemáticos que funcionan a modo de 'notarios' que certificaron el documento.

De esta manera conseguimos que un usuario no pueda volver a usar monedas que ya utilizó; la red rechazará cualquier transacción ilegítima debido a que éstas transacciones se hacen públicas, imposibilitando así una reutilización. De hecho, en la página de Blockchain, podemos encontrar casos de reutilización de monedas que fueron detectados y bloqueados.

El minero crea el blockchain y envía un bloque de transacciones confirmadas a la red P2P de Bitcoin. El resto de nodos actualizan su estado e incluyen el

nuevo bloque de transacciones verificando que dicho bloque es válido.

2.5.7.1. Bitcoin con blockchain completa y sin blockchain completa

Los monederos de criptomonedas tienen diversos patrones de diversificación. Al que se hace referencia en esta sección es a la operativa que sigue al instalar la aplicación. Y es que el consumo de memoria será mucho menor si la aplicación no necesita la cadena de bloques completa para actuar:

Ciertas aplicaciones precisan que, efectivamente, la cadena sea descargada entera para garantizar los principios de seguridad asociados al protocolo. Aunque eso conlleva mayor fluidez en las transacciones, también conlleva un gasto de memoria potencialmente mayor al que precisan otras aplicaciones que operan con una parte, tomando esta parte como referencia para las transacciones y para el posterior acoplamiento a la cadena completa, que se hará fuera del dispositivo.

Las aplicaciones que operan con la cadena de bloques completa están condenadas a la obsolescencia, pues conllevan un problema: la cadena crece cada día al ritmo de 1MB cada 10 minutos.

Actualmente, la cadena blockchain completa es de aproximadamente 50GB. Si continúa creciendo al mismo ritmo, el almacenamiento de la cadena de bloques en algún momento será completamente inviable.

Para ello se usa la cadena parcial. Esta cadena se basa en la opción de ejecutar un nodo de poda. Dichos nodos de poda están habilitados para transmitir bloques y la cantidad de servidores habilitados para ello es bastante verosímil. Sin embargo, habría que tener algunos nodos con la cadena de bloques completa para asegurar la historia.

Aunque el tiempo de inicio del cliente puede convertirse en un problema en algún momento, los datos de la cadena de bloques no se comprueban en cada inicio, sino que la sincronización sólo se ejecuta una vez para cada bloque, lo cual descarta a medio y a largo plazo problemas de computación dado el salto tecnológico que estamos experimentando.

Por último, en referencia a la sincronización hay que destacar que mejoró drásticamente con el lanzamiento de Bitcoin Core 0.12, que aumenta la velocidad de sincronización en 5x, debido a validación de las firmas que se ejecutan a través de libsecp256k1 , una biblioteca que ha sido codificada específicamente para ejecutar las operaciones de Bitcoin de manera más eficiente. Ahora, solo los

encabezados de los bloques deben procesarse en orden y todos los demás datos de bloques pueden procesarse en cualquier orden. Este y otros ajustes han reducido exponencialmente el tiempo necesario para ponerse al día con la red.

2.5.7.2. Construcción de la cadena

La cadena se construye del modo siguiente: al realizarse una transferencia de una dirección Bitcoin a otra, el propietario de la dirección origen firma una transcripción de la dirección destino, lo cual crea una estructura que contiene tanto claves como otros dato que se encuentran pendientes de confirmar. Estos datos se agrupan en bloques, realizándose las tareas de minería sobre cada bloque. Una vez que se validan y se confirman pasan a formar parte de la cadena que conocemos como blockchain.

El tiempo medio de generación de un bloque es de 8.78 minutos, según Blockchain.info. Por defecto cada nodo debe esperar 6 bloques, lo cual quiere decir que hasta que no se hayan validado 6 bloques desde que comenzó la transacción no se considera efectuado el pago.

Por otra parte, la red trata de crear 6 bloques por hora, y cada 2016 bloques (lo cual lleva un tiempo medio aproximado de dos semanas), todos los clientes comparan el número real creado con este objetivo y modifican el porcentaje que ha variado, aumentando (o disminuyendo si procede) la dificultad de generación de bloques.

2.5.7.3. Proceso de creación de una clave pública Bitcoin

En el caso de blockchain se sigue el esquema clásico de algoritmos híbridos. Una dirección en la red Bitcoin se compone de dos claves, una privada y otra pública. La dirección se identifica con el código hash de la clave pública más un checksum de verificación. Cada operación que se realizan con esa dirección de clave pública debe estar apoyada por la utilización de la clave privada asociada mediante la firma digital, por lo tanto únicamente al usuario puede utilizar los Bitcoins.

2.5.8. Mitos Bitcoin

Para finalizar esta sección, en esta última parte haremos un repaso de la actualidad BTC respecto a los mitos que arrastra esta moneda desde sus inicios y

que a día de hoy aún sostienen algunos indocumentados:

- El precio: Pese a haber descendido más de un 60 % desde su pico, el valor de 1 BTC (sobre unos $7.000 al momento de escribir esta memoria) es disuasorio para muchas personas con intención de entrar al mercado. Aunque el Bitcoin siga en la sección de cabecera de muchos diarios digitales, desde mediados del pasado 2017, la gente en su mayoría desconoce lo que es un BTC y mucho más que puede comprar una fracción de la moneda.

  1 BTC es divisible en 100 millones de satoshis, que constituye la unidad más pequeña de BTc y será la que utilicemos en la app. El mero hecho de que alguien no pueda afrontar el pago de una moneda completa, no es impedimento para comprar una parte; el hecho de que alguien no pueda comprar una empresa no supone que no pueda comprar acciones de dicha empresa. Por tanto esto no deja de ser, efectivamente, un mito.

- La cantidad: Pese a haberse perdido de 3 a 4 millones de BTCs en los primeros años (lo corroboran varios estudios) y basándose en una población mundial cercana a los 7 mil millones de individuos, podemos establecer una cantidad aproximada de 220 - 250.000 satoshis por persona.

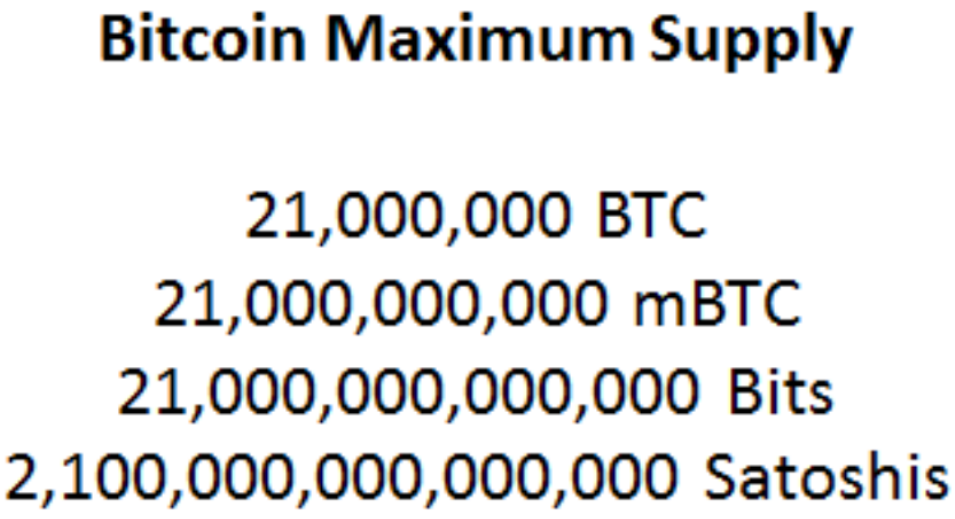


Figura 2.4: Figura: Equivalencia entre las sub monedas de BTC

A finales de 2017, este problema desembocó en una intensa carrera, cuando todas las monedas cuyo valor era inferior a $1 comenzaron a subir de

manera repentina debido al pensamiento general de que eran "baratas". Evidentemente esto era absurdo: cada moneda tiene un suministro distinto, por tanto el precio de una moneda es completamente irrelevante, pues lo que realmente tiene importancia es la capitalización de mercado de la oferta pendiente y si dicha moneda concretamente tiene margen de mejora y expectativas o, de lo contrario, está condenado a desaparecer. Por supuesto es subida fue fugaz y al poco tiempo la mayoría de estas divisas bajaron un 80 %.

En resumen, el precio de 1 BTC en breve no será la métrica más adecuada; lo más acertado será hablar de 1 mBTC (una milésima de Bitcoin) o incluso de un satoshi. A $7.000 el Bitcoin, el precio de 1 satoshi es de unos 0,007 centavos de dólar, precio asequible para cualquiera.

- La volatilidad: No cabe discusión posible en lo relativo a la volatilidad del precio del Bitcoin, pero sí sobre sus motivos. La novedad y el hito histórico que supone una moneda electrónica, segura y descentralizada y a su vez, no respaldada ni fiscalizada por ninguna autoridad central, supone que la estabilidad del BTC sea totalmente inviable a día de hoy. En un futuro no es descartable que esta volatilidad disminuye hasta la desaparición a medida que la capitalización de bitcoins se vuelva equiparable a la de otros activos del mercado con los que se mide.

- El control: Otra de las grandes leyendas es la que dice que el 40 % de BTC pertenecen a sólo unas 1.000 personas, un dato que no ha sido probado y por tanto se restringe al plano puramente especulativo. El único dato cierto al respecto es que a día de hoy hay aproximadamente 24 millones de wallets Bitcoin. No obstante, una sola persona puede ser poseedor de varios wallets, por lo que se puede tener BTC que pertenecen a miles de usuarios, lo que claramente imposibilita un posible análisis sobre la concentración de la riqueza bitcoin.

  Las dos carteras que condensan más BTC son las billeteras frías de Bittrex y Bitfinex. No obstante lo único que estos datos nos permiten inferir es que los dueños de sendas billeteras son multimillonarios, mientras que los BTC en estas billeteras pertenecen a millones de clientes de los referidos intercambios. Solo Coinbase dice tener más de 10M de usuarios. Un intercambio mediante Bitcoins no genera una billetera específica para ti en

Blockchain, sino que asigna algunos de los Bitcoins de un usuario a otro.

Por otra parte, la mayoría de los wallets generan una nueva dirección por cada transacción entrante. Esto supone que un usuario que hubiese recibido 5 veces 0,2 BTC dispondrá de 1 BTC distribuido en 5 direcciones diferente, por lo que es imposible conocer que dichas direcciones en realidad pertenecen a la misma persona. Por lo tanto queda demostrado que la concentración de riqueza BTC es un debate que no se puede probar, y por tanto, estéril.

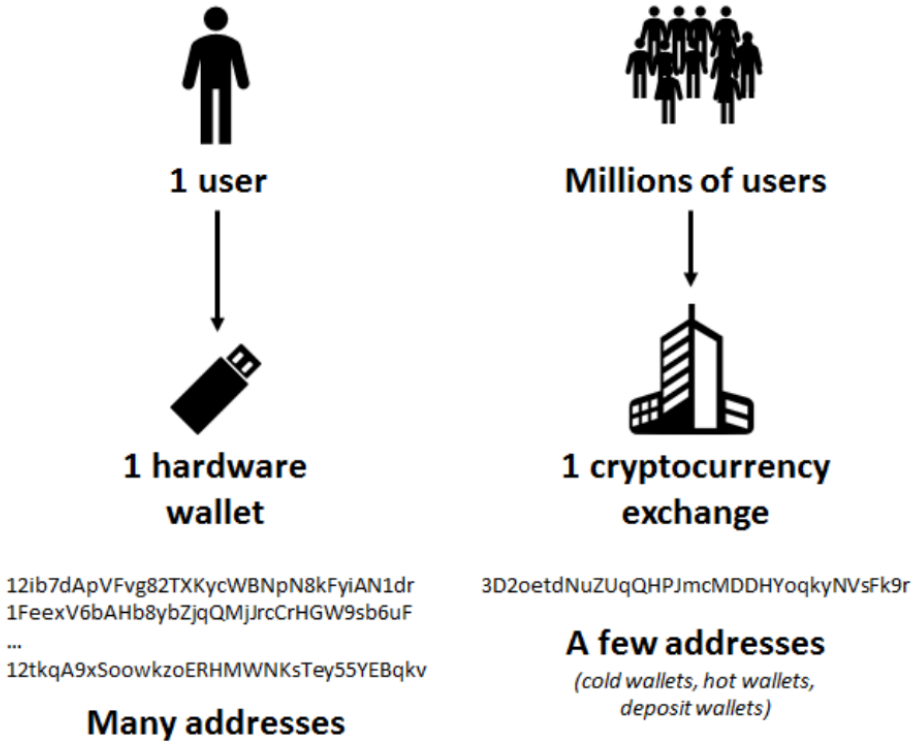


Figura 2.5: Figura: Mito del 40 %

- Los fines ilícitos: Como ya sabemos las transacciones de BTC son públicas, lo cual no constituye el escenario ideal para una actividad ilegal. Hace poco fueron publicados dos informes alegando que solamente un porcentaje menos a un 2 % del total de transacciones BTC fueron empleadas para el blanqueamiento de dinero, aunque es evidente que no es un debate cerrado y la falta de consenso es notable.

  La realidad es que las criptomonedas no suponen una ventaja para los delincuentes. Quizás no sean la panacea en este sentido, pero teniendo en

cuenta que el sistema tradicional está lejos de funcionar en ese respecto, sería injusto tachar a las monedas criptográficas de fraude por el mismo motivo. De hecho, y a diferencia de las monedas fiduciarias, BTC sí implementa mecanismos contra el blanqueamiento de dinero y otros; procedimientos como los de Conozca a su cliente (KYC) y Anti lavado de dinero (ALD) garantizan que no puedas hacer una conversión de BTC a dinero físico de forma anónima, con lo cual las agencias puedes controlar este flujo de manera mucho más fácil. Por tanto, es plausible que siga siendo el dinero físico el protagonista en estas transacciones ilícitas.

- El coste de las transacciones: Desde la implementación de la bifurcación blanda SegWit, teóricamente el número máximo de transacciones por segundo subió a 1.7 millones por día aproximadamente. Huelga decir que dicha cifra es muy lejana de lo que se asumen que tendría que ser para competir con los sistemas de pago tradicionales. No obstante, jamás fue el propósito de Blockchain plasmar cada transacción unitaria. Bastantes transacciones de menos envergadura podrían registrarse fuera de la cadena, lo que constituye la esencia de lo que la próxima Lightning Network llevará a cabo.

  La red completa de Bitcoin fue desarrollada en base a los incentivos. Las tarifas son un mal necesario para prevenir ataques de spam en la red. De no tenerlas, cualquier atacante podría, por ejemplo, transferir millones de pequeños envíos solo para ocupar los bloques y bloquear el sistema. Con tarifas se garantiza que las transacciones de mayor envergadura se procesan primero.

En general, una amplia mayoría de personas desconoce qué es el Bitcoin y su funcionamiento. Pasó algo parecido hace más de 20 años cuando Internet se hizo popular; muchas personas ni siquiera veían utilidad al hecho de tener una dirección de correo electrónico, pues era un sistema poco extendido. BTC y blockchain se encuentran en la misma tesitura. La adopción del Bitcoin sigue progresando, incluso en medio de un mercado a la baja que padeció que el precio de 1 mBTC cayese de $20 a $6. La volatilidad de la que hablábamos ha hecho que el BTC haya tenido tantos auges como caídas en sus 9 años de existencia, pero lo que hace que el Bitcoin sea diferente de otras burbujas es que siempre vuelve.

2.6. Carteras existentes de bitcoin

Como cualquier otra moneda los bitcoins tienen un lugar donde guardarse. A diferencia del dinero crematístico, que son los bancos quien lo almacena y controla, el sistema bitcoin tiene unas aplicaciones llamados monederos o wallets.

Los wallets Bitcoin son carteras digitales en las cuales se guardan los bitcoins. Es por esto que desde el primer momento se debe poseer un monedero que te permita almacenar tu dinero digital. Haciendo un paralelismo con el sistema financiero tradicional, esta billetera emula el comportamiento de una cuenta bancaria.

Las carteras Bitcoin se pueden obtener mediante un software instalable en un ordenador privado o acudiendo a un proveedor online que proporcione dicho servicio. Por consiguiente, éste último posee la ventaja de la facilidad de acceso, como el cualquier servicio online.

Hay una amplia gama de opciones a la hora de decantarse por un wallet. En este apartado analizaremos algunas de las más populares para destacar las ventajas y desventajas de todas ellas.

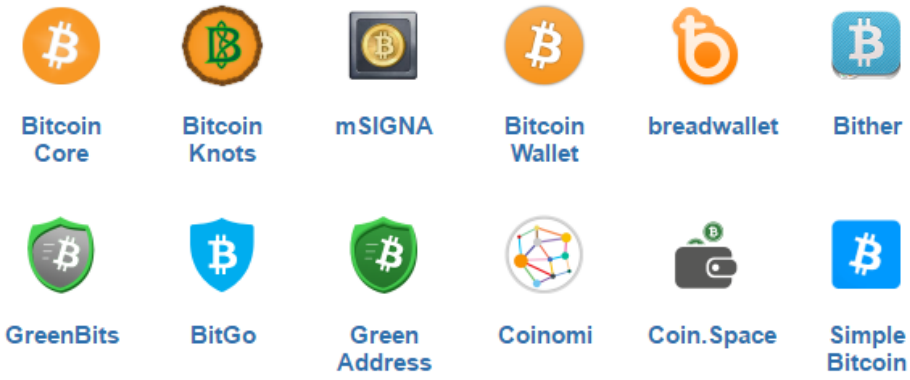


Figura 2.6: Figura:Conjuntos de carteras existentes de BTC

Tras probar los distintos clientes de Bitcoin, Atlas (quien saltaría a la fama por ser expulsado del foro de Bitcoin) llegó a las siguientes conclusiones:

"Hay muchas maneras de usar Bitcoin, pero sólo unos pocos servicios de cartera Bitcoin se destacan por la interfaz, las funciones y la seguridad que ofrecen. Otros, como Bitcoin-Qt, obligan al usuario a esperar durante horas mientras la cadena de bloques es descargada y verificada [N. de la R.: sólo la prime-

ra vez que se instala], sin ofrecer funciones tales como el soporte de múltiples carteras..[Es]to último es algo común a todos los monederos que analizamos a continuación.

2.6.1. Mycelium

Mycelium es el wallet más extendido a día de hoy en el mundo Android. Con soporte para la red Tor, Mycelium constituye una aplicación con un diseño cuidado y con funcionalidades un poco más extendidas destinadas a usuarios más avanzados.

Megion Research & Development GmbH diseñan Mycelium con código abierto en el marco de un conjunto de aplicaciones del paradigma Bitcoin:Entropy, Mycelium Gear,Bitcoin Card, ATMs, Tabtrader y LocalTrader. Este wallet destaca por su rapidez y por sus características de cold wallet (esto es, almacenamiento offline) que supone un nivel de seguridad aún mayor a las monedas que se mantienen en dicho estado. Entre otras funciones, destacan las siguientes:

- Generación dirección pública nueva cada vez que se recibe un pago.
- Control absoluto de las claves privadas.
- Lightweight wallet: No es necesario descargar la cadena de bloques entera para usar el wallet, lo que como ya se ha mencionado, mejora sustancialmente la fluidez.
- Compatibilidad con Tor (en el acceso a los nodos vía web) y con Trezor (para poder guardar los bitcoins en dispositivos hardware).
- En el aspecto de seguridad, incorpora la posibilidad de añadir un código PIN.

Figura 2.7: Figura: Logo de Mycelium

Además, como mejoras adicionales que se implementarán en breve, añade las siguientes (fuente: https://wallet.mycelium.com):

1. Cuentas de Fiat: de pleno derecho, basadas en blockchain.
2. Remesas baratas: los corredores más populares.
3. Tarjetas de débito. Cartera: vinculada y emitida en monedero.
4. Finanzas personales: manejo conveniente de facturas y albaranes.
5. Inversiones: carteras eficientes y cobertura.
6. Transacciones y transacciones comerciales protegidas con custodia.
7. Creación e intercambio de activos criptográficos.

Como complemento diremos que Mycelium sí utiliza la tecnología HD, que se explicará más adelante y que proporciona más seguridad.

2.6.2. Bitcoin Core

Bitcoin Core fue el primer monedero Bitcoin y se considera uno de los más relevantes de toda la red de wallets Bitcoin. Fue desarrollado en sus inicios por Satoshi Nakamoto (creador de Bitcoin), aunque actualmente lo mantienen los desarrolladores de Bitcoin.

Debido a esto, es uno de los monederos que más desarrollo y pruebas ha tenido, lo cual es sin duda su mayor punto a favor. Ofrece alta seguridad, privacidad y estabilidad, lo que le convierte en uno de los wallets más valorado por los usuarios. Además, está disponible para la mayoría de plataformas; Windows, Linux, Mac...

Por contra, su escueto y poco atractivo diseño, junto con la necesidad de mucho espacio en disco y memoria (necesita descargar toda la blockchain) se destacan como sus mayores desventajas.

2.6.3. Multibit

MultiBit es un monedero sencillo de usar y cuya mayor virtud es la rapidez. La instalación y sincronización con la red se hace en pocos minutos. Esto es posible gracias a que funciona sin descargar la cadena de bloques completa.

MultiBit es un software multi idioma (traducido en casi 40 idiomas), por lo que es utilizado por todo el mundo. Además, pese a estar diseñado para usuarios novatos, comparte muchas de las opciones que ofrecen clientes más profesionales. Otra característica destacable es la sencillez y eficacia con la que permite gestionar varias carteras en paralelo.

Disponible para las tres plataformas mayoritarias, otro aspecto positivo es su interfaz, que es considerada por muchos como la mejor de todas las carteras Bitcoin instalables.

2.6.4. Electrum

Electrum es una de las mejores carteras Bitcoin para tener instalado en un ordenador personal. Monedero fácil de utilizar y rápido, se caracteriza especialmente por su sencillez. Además, como MultiBit, tampoco necesita descargar la blockchain entera para realizar transacciones.

La principal característica de este monedero es el uso de servidores para la ejecución de las tareas con mayor carga computacional. Un ejemplo de estas tareas es la creación del código "semilla", que es un código que permitiría la recuperación del monedero en caso de haber perdido la clave privada. Con esta semilla podríamos restaurar o hacer "backups"sin necesidad de ningún tipo de información adicional

Electrum ofrece al usuario la posibilidad de disponer de varios monederos

en una sola semilla. También es posible firmar y crear transacciones en modo offline. Electrum está disponible para Windows, Max, Linux y Android.

2.6.5. Breadwallet

Pensado y diseñado para el sistema operativo de Apple,iOS, este monedero de naturaleza opensource se está posicionando cada vez más arriba en el ranking de los usuarios. Breadwallet usa AES como sistema de encriptación, así como keychain.

Como hacía Electrum con la semilla, Breadwallet también permite la recuperación del wallet mediante una frase encriptada previamente creada. Los usuarios, además, tienen un control total de su llave privada.

2.6.6. Hive OS X

Este monedero se caracteriza por su facilidad de uso y rapidez, puesto que incorpora un sistema que acelera el tiempo de arranque. Entre otras características cabe destacar que ofrece un servicio de mensajería instantánea traducido a varios idiomas para los usuarios del sistema.

Además es uno de los wallets que más se preocupa por el anonimato de sus clientes, ya que se apoya en la red Tor, así que no revela las direcciones IP. Está disponible en Mac OS X, pero también llegará a Android.

2.6.7. Armory

Hablar de Armory es hablar de uno de los wallets más seguros y completos que existen. Gracias a una interfaz perfectamente diseñada y a su variedad de gadgets de seguridad, está considerada por muchos como la mejor cartera Bitcoin.

Tiene tres modos distintos de funcionamiento: Standard, Advanced y Developer. Cada uno de ellos está parametrizado de forma que cada tipo de usuario pueda sacarle un mayor rendimiento en función de sus necesidades y de su nivel de conocimiento.

Entre las opciones de seguridad de Armory cabe destacar un teclado gráfico para inhibir los keyloggers, transacciones offline, además de opciones de almacenamiento en frío. Por otra parte, también es muy interesante la posibilidad de crear carteras fuera de línea.

Por contra, el tiempo de espera, como pasaba en Bitcoin-Qt, es su mayor desventaja, aunque como hemos visto no es significativa respecto a sus grandes ventajas, pues es uno de los monederos que más provecho saca al mundo Bitcoin.

Está disponible para Windows, Mac y Linux y para poder utilizarlo se necesita tener descargado Bitcoin-qt o Bitcoind.

2.6.8. Blockchain.info

Llegamos al que posiblemente sea el mejor wallet del momento. Sus grandes ventajas son que requiere instalación y que la encriptación se produce en el navegador del usuario, no en los servidores de Blockchain.info.

Tiene una interfaz bastante atractiva e intuitiva, incluso para nuevos usuarios. Como cualquier servicio web, se puede acceder desde cualquier dispositivo con internet.

En resumen Blockchain.info satisface todos los requisitos de un usuario más rápido que cualquier cliente de escritorio, sin perder por ello demasiada seguridad respecto a éstos.

Tanto el futuro de Bitcoin, como el de las finanzas pasará por el navegador, y parece que esto es algo que los responsables de Blockchain.info tienen en cuenta.

Por ser la mejor opción, se elige para la realización de este trabajo. Y es que la aplicación a la que da lugar esta memoria, MyWapplet!, se basa en la API de blockchain y se aprovecha de sus múltiples ventajas y de su robustez, y la mejora en el sentido que amplía ciertas funcionalidades y la hace portable por tratarse de una aplicación Android.

2.7. Evolución del Bitcoin

La viabilidad de las bitcoins hoy en día no es discutida prácticamente. Es un fenómenos que ha ido dispersando todas las dudas que sugería en sus inicios. Sin embargo esto se ha ido produciendo con el tiempo. Cierto es que al principio no sólo se desconocía el brutal impacto que significaba la implementación de la cadena de bloques, sino que realmente había dudas de las propia sostenibilidad en el tiempo de Bitcoin. [Jaime Sánchez de Diego Martínez-Cabrera, "Bitcoins, ¿revolución o historia?"].

Entre otras reticencias que levantó Bitcoin en sus orígenes las que más preocupaban al sector eran las referidas al precio de mercado, pues parecía que sería un valor muy volátil y de cierta inestabilidad. Sin embargo esa fluctuación en la cotización podría explicarse por la entrada masiva de inversores y por las decisiones de los gobiernos respecto a Bitcoin.

Por otra parte, la mayor ventaja de Bitcoin es la subyacente a utilizar blockchain: no se precisa de una autoridad central. Sin embargo, la especialización de los mineros en un pequeño número de piscinas podría desembocar en un aumento considerable del poder computacional en la red, lo cual plantearía en el riesgo de una piscina obteniendo más de la mitad de la red y, por consiguiente, hacerse con el control de la misma. [Sánchez de Diego, 2014].

No obstante, y pese a vaticinar erróneamente el fracaso de bitcoin como moneda y restringiendo su uso a una simple plataforma de pago online, de estas lecturas podemos darnos cuenta de que por aquel entonces nadie reparaba en que el verdadero potencial de bitcoin era la tecnología subyacente, como ya venimos explicando.

Sin embargo como hemos visto estos años Bitcoin es una criptomoneda que cada vez aúna más poder. Son varios los autores que han desmontado los mitos en torno a esta moneda electrónica y han explicado cómo hemos avanzado desde el crédito al (Crypto) Cash. [Arvind Narayanan, "Bitcoin and Cryptocurrency Technologies"].

Otro tema recurrente ha sido el relacionado con los aspectos legales. Ciertos autores enfocaron sus investigaciones en la relación de la moneda criptográfica con las instituciones, gobiernos centrales y otros organismos con carácter arbitrario y en ubicarlo dentro de un marco legal, además de analizar la situación del sistema monetario/bancario en la actual web 2.0 para realizar una comparación entre ambos y establecer unas conclusiones. [Luis Antonio Alejo, "Documentos electrónicos para el intercambio de bienes y servicios].

2.7.0.1. El futuro del bitcoin

Tras analizar en profundidad la tecnología blockchain y la evolución del bitcoin, es conveniente culminar con lo que podría ser el futuro de este paradigma tecnológico.

No podemos ignorar que a principios del año pasado, uno de los desarrolladores más importantes del universo bitcoin, Mike Hearn, afirmó que el bitcoin,

tal y como lo conocemos, podría colapsar debido a la congestión y a la falta de capacidad para operar a más de tres transacciones por segundo. [Mª Nieves Pacheco Jiménez, Çriptodivisas: del Bitcoin al MUFG"].

Hearn hizo referencia a que el continuo crecimiento del ecosistema y del número de usuarios necesitaba la aplicación de ciertos ajustes técnicos al protocolo original, pero fue imposible lograr un consenso que permitiese realizarlos, lo que puso a la red de bitcoin al borde del colapso técnico, ya que algunas transacciones llegaron a tardar más de 43 minutos en poder ser confirmadas (incluso algunas se quedaron sin confirmación) cuando lo habitual son 10 minutos. Lógicamente, siendo una moneda con precio volátil, 43 minutos de espera complican su aceptación como método de pago.

Realmente no es más que una lucha interna de la misma red: por un lado, la comunidad de desarrolladores encargados de mantener y evolucionar el código original de Bitcoin -conocidos como "Core"-, por otro, la comunidad rival que ha sacado su propia versión del código donde se aumenta el tamaño del minado de bloques -conocidos como "Classic". [Pacheco, 2016].

Por otra parte, tenemos que los problemas de legalidad depende de la legislación de cada país y de las intenciones de los mismos de aprovechar estas tecnologías o no. El Gobierno de Japón fue pionero en la regulación de las criptodivisas, aprobando en 2014 el primer marco normativo del bitcoin.

De esta manera, pasó a considerar a los bitcoins como una mercancía a efectos prácticos parecida a los metales preciosos, y no como una moneda. En consecuencia los beneficios derivados de las transacciones online y las rentabilidades obtenidas mediante esa moneda estarán sujetas a impuestos, y el lavado de dinero a través de dicha moneda será tipificado como delito. [Mª Nieves Pacheco Jiménez, Çriptodivisas: del Bitcoin al MUFG"].

En este contexto no resulta raro que Japón vuelva a situarse en la vanguardia con su objetivo de emitir su propia divisa electrónica. Por ello, Mitsubishi Tokyo-UFJ, el banco más grande de Japón, está desarrollando actualmente una divisa digital, denominada provisionalmente "MUFG"(procedente de las siglas de la entidad), que permita realizar compras, transferencias bancarias o cambiar divisas extranjeras a menor coste que con monedas corrientes.

El funcionamiento del MUFG está basado, al igual que el bitcoin, en la blockchain. El objetivo es que funcione como una tarjeta-monedero con la salvedad que la nueva moneda incluirá la posibilidad de intercambiarse entre sus usuarios y la gestión podrá realizarse a través del ordenador o de los smartphones,

incluso en un futuro habrán disponibles una nueva generación de cajeros automáticos que posibiliten operaciones con la criptomoneda y con smartphones. [Pacheco, 2016].

2.8. Otras criptomonedas: Altcoins

Bitcoin fue la primera, pero no única. En rigor, todas las demás -que se denominan Altcoins- han evolucionado de Bitcoin, fuente de inspiración del resto, pero no por ello carecen de importancia en el mercado actual. Para Antonopoulo (2014), las Altcoins son monedas digitales implementadas utilizando el mismo patrón de diseño que bitcoin, pero con una cadena de bloques y una red completamente independiente.

Muchos son los estudios que se han hecho sobre las diferentes criptodivisas creadas a partir de la idea de Bitcoin, aunque todas con sus respectivas modificaciones, como hemos visto en el apartado anterior. Por esta razón, focalizar en el bitcoin como pilar es el primer paso, ya que, tomando como referencia ésta moneda criptográfica se pueden realizar comparativas a muchos niveles.

Hay algunas comparativas que pretenden centrar las características y propiedades de las monedas que trabajen bajo el mantra de las pruebas de trabajo o PoW como sistema de minado, y a su vez, las que estén basadas en la tecnología de cadenas de bloques. A partir de esto se busca estudiar las siguientes propiedades de dichas monedas criptográficas:

1. Sistema utilizado para incluir bloques en la cadena, es decir distinguirlas en función del PoW que se utiliza (SHA-256, Scrypt, etc...).

2. Sistema de distribución de la moneda con respecto al volumen de monedas que se van generando.

3. Características de los bloques.

4. Rendimiento de las transacciones por unidad de tiempo,

5. Plataformas disponibles en la que los usuarios puedan hacer diferentes funciones como la minería y la seguridad de bloques transaccionales. [Medinarey, 2016].

Hay otros factores ajenos a la implementación de la blockchain pero que deben tenerse cuenta pues son implícitos al uso de criptomonedas. Éstos no son otros que los factores humanos, como las emociones y la confianza. que conviven en los proveedores, los consumidores y los administradores de las monedas, que deben reducirse para hacer fácil y natural su adopción.

[Paulo N Carrillo, "Joincoin: Un entorno de trabajo común para criptomonedas"].

En este caso la solución pasaría por el diseño de un entorno móvil común definido como un ecosistema sostenible para impulsar negocios apoyados en redes sociales virtuales que permitan que los acuerdos firmados entre pares sean validados por otros pares, utilizando la tecnología Blockchain para confirmar y asegurar las transacciones. Para esta comprobación utilizan como base la moneda virtual Eurakos, desarrollada en Girona. [Paulo N Carrillo, 2016]

A continuación hacemos un repaso y una breve descripción de algunas de ellas para finalizar con algunos papers sobre dichas divisas.

2.8.1. Ethereum

Ethereum es el proyecto que comenzó a principios de 2014 con el fin de realizar todo tipo de acciones relacionadas con los intercambios financieros; crowdfunding), propiedad intelectual, etc... Además añadía la posibilidad de que cualquier desarrollador creará y publicará aplicaciones, utilizando esta plataforma orientada al código abierto, dando lugar a uno de los usos ya mencionados de BLockchain: los contratos inteligentes.

Lo que diferencia a este proyecto de otros semejantes es que no se trata solo de la creación de una criptomoneda, sino que contempla toda una red cuyo objetivo es la realización las actividades anteriormente mencionadas (siempre relacionadas con la filosofía open source).

El propósito inicial del proyecto Ethereum fue el de descentralizar la web mediante la introducción de cuatro componentes como parte de los objetivos de su Web 3.0:

1. Publicación de contenido estático.
2. Mensajes dinámicos.
3. Transacciones confiables.

4. Interfaz de usuario integrada y funcional.

Estos componentes fueron diseñados para reemplazar algunos aspectos de la experiencia web que se dan por sentado actualmente, solo que siguiendo el estilo blockchain, esto es, de una manera descentralizada y anónima.

La moneda que desarrollaron se llama Ether. Cada una de las actividades, transacciones -incluso los contratos inteligentes- se realizan mediante dicha moneda. Al estar basada en tecnología blockchain, permite a todos los usuarios controlar en tiempo real el valor actual de las transacciones procesadas.

Así mismo ofrece un sistema de minería que a grandes rasgos recompensa a los usuarios que cedan los recursos hardware de sus equipos para realizar las operaciones llevadas a cabo dentro de la red.

El proyecto en sus comienzos fue bien acogido; en las primeras 20 horas tras su lanzamiento, los primeros inversores compraron divisas de esta moneda por valor de 2,6 millones de dólares. De hecho, hoy en día se consolida como un competidor directo del Bitcoin.

2.8.2. Primecoin

De modo anecdótico, mencionamos esta Altcoin por su relación con el mundo matemático. Primecoin es una divisa que trata de hallar nuevos números primos, aportando de esta manera conocimiento a la ciencia de las matemáticas. La cadena de bloques de Primecoin contiene todos los números primos descubiertos hasta la fecha, produciendo de este modo un registro público de los descubrimientos científicos, en paralelo con el libro de contabilidad público de las transacciones.

2.8.3. Ripple

Ripple es un proyecto fundamentado en un pequeño software libre cuyo objetivo es el desarrollo de un sistema de crédito basado en el sistema de par a par. Siguiendo uno de los principios de Bitcoin, Ripple usa una arquitectura P2P: todos los nodos de Ripple funcionan como un sistema de cambio local, formando de manera global una especie de banco mutualista descentralizado.

Ripple es un producto de la empresa RippleLabs de EE.UU. RippleLabs es manejada por algunas de las mentes más brillantes del Silicon Valley, y financia-

do por los mejores fondos de los Estados Unidos (Google Ventures, Andreesen-Horowitz, etc...).

Llevado al extremo, la red Ripple es un servicio de red social distribuido basado en el honor y en la confianza entre las personas existentes en las redes sociales del mundo real. De esta manera, el capital financiero se sustenta en el capital social. Una versión reducida de la red ripple consistiría en una extensión del sistema bancario jerárquico existente, en el cual existirían rutas de pago alternativas que no pasarían por un banco central.

Haciendo un paralelismo con el sistema de enrutamiento de paquetes de datos de internet a través de los ordenadores, Ripple se propone hacer lo mismo con los pagos, apoyándose en una red de ordenadores arbitraria y abierta.

Con este objetivo Ripple se propone ser como una variante del sistema jerárquico bancario actual; los bancos son intermediarios entre sus cliente mientras que los bancos centrales son intermediarios de dichos bancos. En este caso el proyecto Ripple ofrece un sistema ajeno a instituciones centrales jerárquicas:o todos los participantes de forma democrática decidirán la política monetaria.

La moneda a la que dio lugar este proyecto es el XRP, cuyas características son las siguientes:

1. Los XRP funcionan como "puentes.[en]tre las monedas crediticias y las digitales. Complementan al Bitcoin en cuanto a la facilidad de intercambio de dinero crematístico por criptomonedas, o por otro tipo de monedas.

2. Al igual que Bitcoin, Ripple usa una red descentralizada, con la salvedad de que Ripple no precisa hacer uso de su criptodivisa para usar dicha red.

3. Respecto a los costes de transacción, para un usuario estándar son mínimos. Sin embargo para prevenir el abuso de transacciones se retiene una pequeña cantidad de XRP.

4. Otro cambio respecto a los Bitcoins es que las transacciones no funcionan a través de un método de proof-of-work, sino que Ripple funciona con el método de proceso de consenso iterativo el cual no posibilita la minería.

Aunque el plan inicial era hacer una distribución amplia y equitativa, los creadores se quedaron con una parte como ya hiciesen los de Bitcoin.

2.8.4. Litecoin

Lanzada en octubre de 2011 por Charles Lee, Litecoin es una moneda de Internet de tipo punto a punto que permite realizar pagos instantáneos y de costo casi cero a cualquier parte del mundo. Litecoin es una red de pagos global y de código abierto que es completamente descentralizada y sin autoridades centrales, según indican ellos mismos en su web litecoin.org/es/.

Se desarrolló como un proyecto similar a Bitcoin; criptomoneda descentralizada, apoyada en una red P2P, transferencias a través del sistema proof-of-work y de código abierto, pero con 3 disimilitudes frente al Bitcoin:

1. Rapidez: el tiempo de transacción de bloques en Litecoin es de 2,5 minutos frente a los 10 minutos de media de Bitcoin. En principio esto supone una confirmación más rápida de las transacciones, pero por otra parte las hace menos seguras.

2. Límites: En este caso los límites de emisión son también mayores que los de Bitcoin. El máximo de emisión de criptomonedas para Bitcoin es de 21 millones frente a los 84 millones de Litecoin.

3. Algoritmo Scrypt: El algoritmo que usan para calcular el hash es diferente.

Más allá de estas sutiles diferencias y del cambio de interfaz de uno respecto al otro, la principal diferencia radica en la minería. Al cambiar el algoritmo que resuelve las transacciones, y al ser éstas más rápidas con Litecoin, consigue que equipos más limitados a nivel de recursos puedan realizar esta actividad.

Aunque a simple vista esto parecía una ventaja de esta criptodivisa frente a la hegemónica Bitcoin, con el tiempo se vio que en realidad esto no era así, pues la competencia tecnológica es lo que ha supuesto mayor crecimiento para Bitcoin que para sus competidores, ya que es un mercado que ha copado las mayores inversiones de los distintos competidores .

2.8.5. Dogecoin

Dogecoin es otro proyecto cuyas bases vienen de Bitcoin. Esta criptodivisa se apoya en los mismos mecanismos de Bitcoin pero a su vez cambia determinados parámetros que la han hecho posicionarse como una de las Altcoins más importantes.

Dogecoin es una moneda digital peer-to-peer y de código abierto que derivada de Litecoin (que como ya explicamos deriva a su vez de Bitcoin). Fue creada por Billy Markus como una moneda experimental a partir de otra criptomoneda denominada Bells. El logo y el nombre de Dogecoin hacen referencia al "meme"que tiene como imagen un perro de raza ShibaInude de origen japonés. Esto la ha convertido en la primera criptodivisa con una religión paródica.

Dogecoin creció merced a la mayor implicación de sus desarrolladores, que observaron cómo su proyecto ganaba popularidad en las redes sociales, motivo por el cual decidieron lanzarlo pensando en que podían ganar usuarios respecto a Bitcoin.

Según su propia web, hay varias maneras de conseguir Dogecoins (DOGE):

1. Participar activamente en la comunidad Dogecoin: mediante el uso de la red social Reddit, los usuarios pueden dar recompensas a otros usuarios nuevos que realicen alguna "buena acción.[en] Internet, promoviendo así una participación de usuarios más o menos altruista.

2. Registrarse en un "faucet": los faucet son sitios web que te dan una pequeña cantidad de Dogecoin gratuitamente para que los usuarios se inicien en el uso de la moneda.

3. Comprar directamente Dogecoins por dinero corriente en cualquiera de sus mercados. Los mercadores que se recomiendan a través de su página web oficial son:

   a) WellSellDoge

   b) ANXPRO

   c) Celery

4. La minería: Está enfocado a usuario avanzados, aunque como pasaba en Litecoin, no requiere el uso de muchos recursos computacionales. Actualmente ya se han minado más del 80 del total de Dogecoin (100 mil millones).

La principal diferencia respecto a Litecoin es su dominio y popularidad en las redes sociales, tales como Twitter y Reddit (en esta última es la segunda criptomoneda con más seguidores, solo por detrás del Bitcoin). Su espíritu altruista junto con esto supone que su expansión continúe y que su comunidad de usuarios se postule como una comunidad sólida y de ayuda mutua.

2.8.6. MaidSafeCoin

Nacido en 2006 y desarrollado por David Irvine, su apuesta es la de un internet descentralizado, seguridad, privacidad y libertad para todo el mundo.

Para éste propósito se crea la red SAFE Network, cuyos fundamentos explicamos a continuación:

1. Acceder y asegurar la información personal. Una de las mayores batallas del proyecto MaidSafe es conseguir un internet anónimo. Por ello, proponen que, aunque la información del usuario encriptada se encuentra en la red, sólo podrá ser visualizada con la clave privada del usuario.

2. Datos que se encriptan mutuamente. Los datos se envían a la red y se fragmentan. Cada fragmento se encripta y se distribuye por toda la red.

3. Hay disponible una red distribuida que permite el almacenamiento en caché cerca del usuario que la solicita.

4. Disponibilidad y redundancia de datos: se realizan copias de cada dato y se distribuye por la red P2P, cambiando la localización de los datos constantemente, haciéndolos prácticamente inmunes a ataques informáticos.

Bajo estas directrices crean la criptomoneda denominada Safecoin. Estas safecoins se usan como recompensa a los miembros que colaboren, por lo que conforma el sustento mayoritario de la red. También es posible conseguir las monedas mediante la compra tradicional.

Como novedad, ofrecen otro método para adquirirlas, el farming. Los farmers -usuarios que se dedican a esto- son como los mineros en Bitcoin, solo que éstos ganan safecoins proporcionando "proof of resource". Básicamente los farmers ejecutan un nodo de red en su ordenador.

La red SAFE es capaz constantemente de medir y verificar estos recursos de manera inmediata. Esto es el proof of resource. Para ganar Safecoin los farmers, al tratarse a la vez de usuarios, deben proporcionar más recursos de los que consumen y cuantos más recursos proporcionan, más obtendrán, pero el sistema funciona también de forma aleatoria ya que la información se almacena aleatoriamente en todos los nodos de la red.

Hay otra criptodivisa que persigue el anonimato de los usuarios y que merece que le hagamos mención. Darkcoin busca ser una moneda totalmente anóni-

ma, utilizando un protocolo de remezcla de todas las transacciones denominado DarkSend. Pretende parecerse lo más posible al dinero en efectivo.

En definitiva, el futuro de las criptomonedas supondrá que muchas actividades humanas que anteriormente requerían instituciones u organizaciones centralizadas para funcionar ahora pueden descentralizarse. La invención de la cadena de bloques reducirá de manera considerable el coste de organización y coordinación de los sistemas a gran escala. Además, trae consigo otra serie de ventajas implícitas como son la eliminación de la concentración de poder, la corrupción, y la captura del regulador.

Capítulo 3

Herramientas utilizadas

En esta sección se presentarán formalmente las herramientas empleadas para el desarrollo del proyecto a nivel técnico. Pasaremos por encima en tecnologías tan conocidas ya como Java o Android Studio (paradigma sobre el que se desarrolla esta aplicación) para centrarnos y profundizar en las tecnologías propias del trabajo, es decir, la API usada como base de todo el proyecto, la de blockchain.info.

3.1. Java

En primer lugar vamos a hablar de Java. Como ya todos sabemos a estas alturas,Java es un lenguaje de programación además de una plataforma informática, la cual fue comercializada por vez primera en 1995 por Sun Microsystems. Multitud de aplicaciones y sitios web funcionan a base de java. pues Java es veloz, seguro y extremadamente fiable.

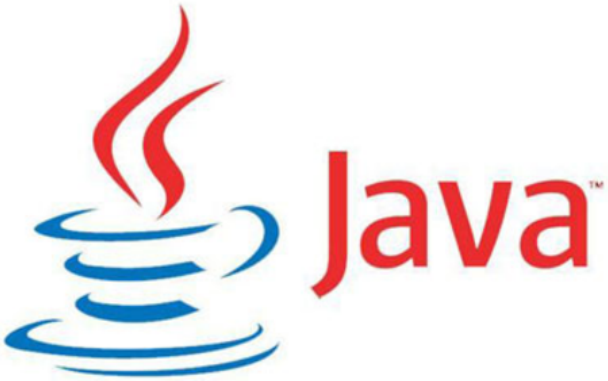


Figura 3.1: Figura: Java símbolo

Entre otras características, se destacan las siguientes:

Java es el paradigma tecnológico que sustenta el desarrollo de aplicaciones y que proporcionan a la Web componentes que le proporcionan atractivo y utilidad. Es importante hacer distinción entre Java y javascript. Ésta última se trata de una tecnología trivial utilizada en la creación de páginas web y solamente se ejecuta en el navegador..

Java ofrece elementos que posibilitan ejecutar ciertos juegos, cargar fotografías, chatear en línea, realizar visitas virtuales y utilizar servicios como, verbigracia, cursos online, servicios de banca online o mapas interactivos. Otro aspecto importante es que, por defecto, Java avisa si hay nuevas actualizaciones listas para instalarse.

Así pues, Java se destaca por ser la base para prácticamente todos los tipos de aplicaciones en red, además de ser el estándar global para desarrollar y distribuir aplicaciones móviles y embebidas, juegos, contenido basado en web y software de empresa.

Hoy en día se antoja complicado imaginarse un mundo informático o tecnológico sin java. Es más, la consecuencia inmediata de lo anterior es que directamente el paradigma económico actual es inconcebible sin Java. Más allá de una opinión, los datos que avalan esta teoría son lo siguientes (fuente: java.com):

- El 97 % de los escritorios empresariales ejecutan Java.
- El 89 % de los escritorios (o computadoras) en Estados Unidos ejecutan

Java.

- 9 millones de desarrolladores de Java en todo el mundo.
- La primera opción para los desarrolladores.
- La primera plataforma de desarrollo.
- 3 mil millones de teléfonos móviles ejecutan Java.
- El 100 % de los reproductores de Blu-ray incluyen Java.
- 5 mil millones de Java Cards en uso.
- 125 millones de dispositivos de televisión ejecutan Java.
- 5 de los 5 principales fabricantes de equipos originales utilizan Java ME.

JavaFX es un conjunto de productos tecnológicos de Oracle destinado a la creación de Rich Internet Applications (RIAs), que son aplicaciones web cuyas características y capacidades de aplicaciones son equivalentes a las de escritorio. JavaFX incluye tecnologías como JavaFX Script y JavaFX Mobile, aunque hay más productos JavaFX que llegarán.

JavaFX está basado en Java. Esto supone que la comunidad de desarrolladores puedan usar las RIAs para la creación de aplicaciones con comportamiento idéntico en distintas plataformas.

JavaFX incrementa la potencia de Java posibilitando a los programadores utilizar bibliotecas nativas de Java en aplicaciones que implementan JavaFX, lo cual les permite combinar las capacidades en Java y la tecnología JavaFX para crear aplicaciones que resulten muy atractivas visualmente.

3.1.0.1. ¿Por qué los desarrolladores de software eligen Java?

Java ha sido fuertemente testeado por una extensa comunidad de desarrolladores, ingenieros y entusiastas de la informática en general y de Java en particular. Java está implementado de tal manera que permite el desarrollo de aplicaciones portátiles cuyo rendimiento es bastante alto independientemente de la plataforma en la que se ejecute. En resumen, Java se ha convertido en un elemento cuyo valor es extremadamente alto para los desarrolladores, ya que les permite:

Desarrollar código en una determinada plataforma y ejecutar virtualmente en otra.

Crear programas ejecutables vía navegador y acceder a diversos servicios web.

Diseñar aplicaciones de servidor.

Crear combinaciones de aplicaciones o servicios que utilizan el lenguaje Java.

Escribir aplicaciones potentes y eficaces para teléfonos móviles, procesadores remotos, microcontroladores, módulos inalámbricos, sensores, gateways, productos de consumo y prácticamente cualquier otro dispositivo electrónico

3.1.0.2. ¿Qué es JavaFX?

JavaFX es un conjunto de productos tecnológicos de Oracle destinado a la creación de Rich Internet Applications (RIAs), que son aplicaciones web cuyas características y capacidades de aplicaciones son equivalentes a las de escritorio. JavaFX incluye tecnologías como JavaFX Script y JavaFX Mobile, aunque hay más productos JavaFX que llegarán.

Figura 3.2: Figura: Qué es JavaFx

JavaFX está basado en Java. Esto supone que la comunidad de desarrolladores puedan usar las RIAs para la creación de aplicaciones con comportamiento idéntico en distintas plataformas.

JavaFX incrementa la potencia de Java posibilitando a los programadores utilizar bibliotecas nativas de Java en aplicaciones que implementan JavaFX, lo cual les permite combinar las capacidades en Java y la tecnología JavaFX para crear aplicaciones que resulten muy atractivas visualmente.

3.2. Android

Android es un sistema operativo basado en el Kernel Linux que inicialmente estaba pensado para el uso en dispositivos móviles como pueden ser Symbian, Blackberry OS, iOS o Windows Mobile, aunque en los últimos años se han expandido a otros soportes como relojes, tablets, ordenadores, netbooks o televisores.

Android fue desarrollado en sus inicios por Android Inc., una pequeña compañía de Palo Alto, California fundada en 2003, que posteriormente sería adquirida por Google en Julio del 2005. En 2007 se fundó la Open Handset Alliance (OHA), una alianza comercial de actualmente 84 compañías dedicada al desarrollo de estándares abiertos para dispositivos móviles, la cual es fue liderada por Google. Con la creación de esta alianza se presentó Android al mundo, y el 12 de noviembre de ese mismo año publicaron una beta del SDK basado en una licencia de código abierto.

3.2.1. Versión actual

3.2.1.1. Android 8.0 – Oreo

El día 21 de marzo de 2017, Google anunció el sucesor de Android Nougat, Android Oreo (cuyo alias en el anuncio fue Android .ºº"). Android Oreo es la versión actual de Android. Su nombre se relevó el mismo día del eclipse total de Sol en USA. Se lanzó una vista previa de calidad alfa para los móviles de Google (Nexus y Pixel). La segunda revelación se considera la calidad beta, siendo la tercera revelación la que completa la API.

Repasando los avances de esta versión, es notable la limitación de procesos en segundo plano, lo cual consigue una optimización de los recursos mayor que redunda en un ahorro mayúsculo de la batería.Respecto a las notificaciones, se mejoran principalmente mediante dos aspectos: para empezar es posible categorizarlas y agruparlas con cierto orden arbitrario, así como establecer la frecuencia con la que se muestran (deslizas la aplicación hacia la izquierda y manipular mediante un menú de ajustes que dispone de un reloj para ajustar dicha frecuencia).

Por otra parte, las "Notifications Dots", que añaden un número junto a los iconos que representan el conjuntos de notificaciones asociadas a esa categoría.

Respecto al diseño, una mejora sustancial se puede apreciar en el menú de

Çonfiguración", que transiciona hacia un tema blanco y una distinta categorización más amplia de los diversos ajustes. Se incorpora además un nuevo patrón unificador de diseño para los iconos: el icono se basa en dos capas, la del icono y la de fondo, que interactúan y se acoplan formando animaciones cuando se hace uso de ellas. En lo sucesivo se deberán incluir cuatro tipos de iconos para que el usuario elija el que usará el launcher.

Otras característica visuales destacables son las relativas al Android TV, que incluye un nuevo launcher, soporte integrado para modos picture-in-picture, la simplificación de los sonidos de timbre, alarma y notificación, además de nuevos emojis agregados al estándar Unicode 10.

A nivel de plataforma añade soporte para redes inalámbricas, gamas de color amplias en aplicaciones, multiprocesos, una API para autofillers con el soporte de navegación segura nativo de Google para WebViews, otra API que permite al sistema aplicaciones de VoIP, y actividades de lanzamiento en pantallas remotas.

Además, cuenta con Android Runtime (ART), que ofrece mejora cualitativamente el rendimiento. Android O limita de manera transparente al usuario la actividad de las aplicaciones en segundo plano, mejorando así la duración de la batería.

La arquitectura en la que se apoya Android está siendo sometida a revisión de tal manera que el código de más bajo nivel propio del proveedor soporte la arquitectura física de un dispositivo sea independiente del sistema operativo en sí, basándose en una capa hardware de abstracción llamada ïnterfaz de proveedor". Para ello se requiere que las interfaces de dichos proveedores no sean incompatibles con las próximas versiones de Android. Estos cambios suponen que los fabricantes de equipos originales necesiten realizar las actualizaciones necesarias en el ecosistema asociado al sistema operativo, manteniendo así inmutable su interfaz.

3.2.1.2. Android Go

Android incorpora con Android Go una distribución específica hecha a medida para dispositivos de gama baja, que se usa en cualquier dispositivo que posea menos de 1 GB de RAM. Estos dispositivos son provistos con mejoras de la plataforma diseñadas para reducir el uso de datos móviles y un conjunto especial de servicios Google diseñados para reducir el uso de recursos y ancho de banda. Google Play Store (que cambia su diseño hacia una escala más modular

que reduce el uso de memoria) también destacó aplicaciones ligeras adecuadas para estos dispositivos.

A su vez, Google Play Store resalta las aplicaciones ligeras cuyo diseño se adapte a este patrón. Así mismo la interfaz del sistema operativo se cambia mediante el panel de configuración rápida que proporciona mayor precisión a la información sobre la batería y el límite de datos móviles. El menú de aplicaciones recientes implementan un diseño actualizado que se limita a cuatro aplicaciones (reduciendo así consumo de RAM), y una API que permite a los operadores móviles desarrollar el seguimiento de datos y recargas bajo el menú de configuraciones.

3.2.2. Características Android

- Adaptable a múltiples pantallas y resoluciones.
- Almacenamiento de datos en bases de datos SQLite.
- Conectividad: actualmente soporta, entre otras, las siguientes tecnologías: GSM, EDGE, GPRS, UMTS, Bluetooth, Wifi, LTE, HSDPA,HSDPA+, NFC.
- Mensajería SMS y MMS
- Navegador Web basado en WebKit
- Soporte de Java a través de una máquina virtual Dalvik. Esta máquina virtual es específicamente diseñada para Android y está optimizada para tener un consumo de recursos limitado.
- Soporte para hardware adicional: GPS, brújula, acelerómetro, cámara o pantalla táctil, etc.
- Entorno de desarrollo: Inicialmente se utilizaba el IDE Eclipse con el plugin ADT, hasta que surgió el IDE oficial para Android, denominado Android Studio.
- Google Play: Catálogo de aplicaciones gratuitas y de pago que pueden ser descargadas e instaladas sin necesidad de un pc.
- Soporte para pantallas multi-táctiles.
- Multitarea

- Búsqueda a través de voz.
- Tethering: permite al dispositivo Android actuar como punto de acceso inalámbrico, compartiendo su conexión de datos.

3.2.3. Arquitectura

A continuación se describe la arquitectura del sistema operativo Android. Este se compone de cinco bloques divididos en cuatro niveles o capas. Cada una de estas capas utiliza servicios ofrecidos por las anteriores, y ofrece a su vez los suyos propios a las capas de niveles superiores, tal como muestra la siguiente figura:

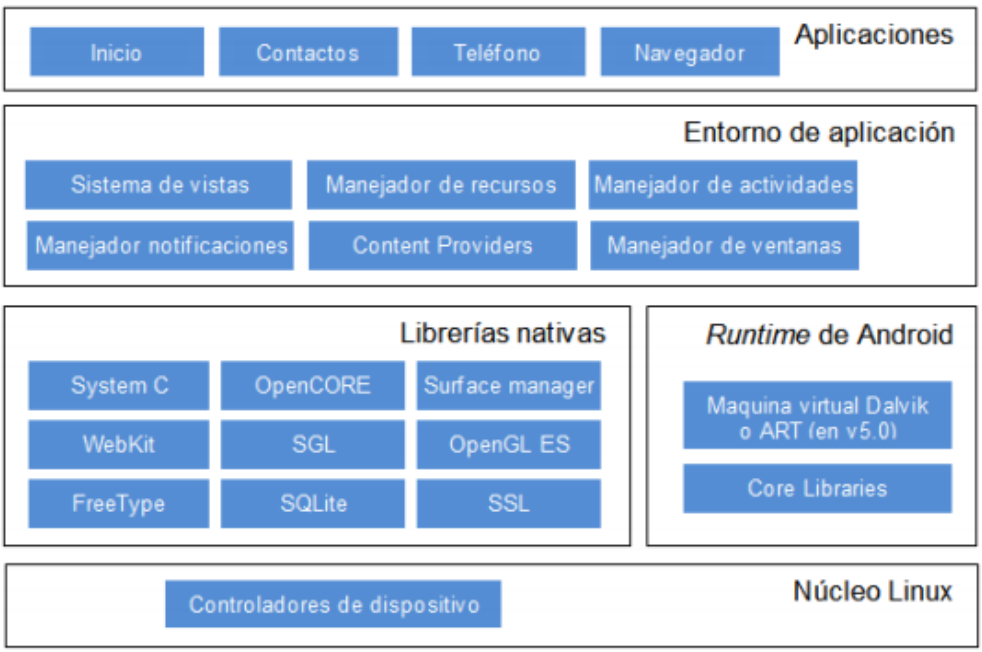


Figura 3.3: Figura: Arquitectura del SO Android

3.2.3.1. El núcleo Linux

El núcleo de Android está formado por el sistema operativo Linux versión 2.6. Esta capa proporciona servicios como la seguridad, el manejo de la memoria, el multiproceso, la pila de protocolos y el soporte de drivers para dispositivos. Esta capa del modelo actúa como capa de abstracción entre el hardware y el resto de la pila. Por lo tanto, es la única que es dependiente del hardware.

3.2.3.2. Runtime de Android

Está basado en el concepto de máquina virtual utilizado en Java. Dado las limitaciones de los dispositivos donde ha de correr Android (poca memoria y procesador limitado) no fue posible utilizar una máquina virtual Java estándar. Google tomó la decisión de crear una nueva, la máquina virtual Dalvik, que respondiera mejor a estas limitaciones. Algunas características de la máquina virtual Dalvik que facilitan esta optimización de recursos son: que ejecuta ficheros Dalvik ejecutables (.dex) –formato optimizado para ahorrar memoria. Además, está basada en registros.

Cada aplicación corre en su propio proceso Linux con su propia instancia de la máquina virtual Dalvik. Delega al kernel de Linux algunas funciones como threading y el manejo de la memoria a bajo nivel. A partir de Android 5.0 se reemplaza Dalvik por ART. Esta nueva máquina virtual consigue reducir el tiempo de ejecución del código Java hasta en un 33 %.También se incluye el "core libraries" con la mayoría de las librerías disponibles en el lenguaje Java.

3.2.3.3. Librerías nativas

Incluye un conjunto de librerías en C/C++ usadas en varios componentes de Android. Están compiladas en código nativo del procesador. Muchas de las librerías utilizan proyectos de código abierto. Algunas de estas librerías son:

- System C library: una derivación de la librería BSD de C estándar (libc), adaptada para dispositivos embebidos basados en Linux. - Media Framework: librería basada en PacketVideo's OpenCORE; soporta codecs de reproducción y grabación de multitud de formatos de audio, vídeo e imágenes.
- Surface Manager: maneja el acceso al subsistema de representación gráfica en 2D y 3D.
- WebKit: soporta un moderno navegador Web utilizado en el navegador Android y en la vista Webview. Se trata de la misma librería que utiliza Google Chrome y Safari de Apple.
- SGL: motor de gráficos 2D.

- Librerías 3D: implementación basada en OpenGL ES 1.0 API. Las librerías utilizan el acelerador hardware 3D si está disponible, o el software altamente optimizado de proyección 3D.
- FreeType: fuentes en bitmap y renderizado vectorial.
- SQLite: potente y ligero motor de bases de datos relacionales disponible para todas las

  aplicaciones.
- SSL: proporciona servicios de encriptación Secure Socket Layer (capa de conexión segura).

3.2.3.4. Entorno de aplicación

Proporciona una plataforma de desarrollo libre para aplicaciones con gran riqueza e innovaciones (sensores, localización, servicios, barra de notificaciones, etc.).

Esta capa ha sido diseñada para simplificar la reutilización de componentes. Las aplicaciones pueden publicar sus capacidades y otras pueden hacer uso de ellas (sujetas a las restricciones de seguridad). Este mismo mecanismo permite a los usuarios reemplazar componentes. Los servicios más importantes que incluye son:

1. Views: extenso conjunto de vistas, (parte visual de los componentes).
2. Resource Manager: proporciona acceso a recursos que no son en código.
3. Activity Manager: maneja el ciclo de vida de las aplicaciones y proporciona un sistema de navegación entre ellas.
4. Notification Manager: permite a las aplicaciones mostrar alertas personalizadas en la barra de estado.
5. Content Providers: mecanismo sencillo para acceder a datos de otras aplicaciones (como los contactos).

3.2.3.5. Aplicaciones

Este nivel está formado por el conjunto de aplicaciones instaladas en una máquina Android. Todas las aplicaciones han de correr en la máquina virtual Dalvik para garantizar la seguridad del sistema.

Normalmente las aplicaciones Android están escritas en Java. Para desarrollar aplicaciones en Java podemos utilizar el Android SDK. Existe otra opción consistente en desarrollar las aplicaciones utilizando C/C++. Para esta opción podemos utilizar el Android NDK (Native Development Kit).

3.2.4. Componentes

Dentro de una aplicación Android se pueden encontrar cinco componentes principales: Service, Content Provider, Broadcast Receiver, Intent y Activity. Todas las aplicaciones estarán formadas por la combinación de estos elementos.

- Service: Los servicios (service) son componentes sin interfaz gráfica que se ejecutan en segundo plano. En concepto, son similares a los servicios presentes en cualquier otro sistema operativo.
- Content Provider: Es el mecanismo que se ha definido en Android para compartir datos entre aplicaciones. Mediante estos componentes es posible compartir determinados datos de nuestra aplicación sin mostrar detalles sobre su almacenamiento interno, su estructura, o su implementación.
- Broadcast Receiver: Un broadcast receiver es un componente destinado a detectar y reaccionar ante determinados mensajes o eventos globales generados por el sistema (por ejemplo: "Batería baja", "SMS recibido", "Tarjeta SD insertada", ...) o por otras aplicaciones.
- Intent: Un intent es el elemento básico de comunicación entre los distintos componentes Android que hemos descrito anteriormente. Se pueden entender como los mensajes o peticiones que son enviados entre los distintos componentes de una aplicación o entre distintas aplicaciones.
- Activity: Las actividades (activities) representan el componente principal de la interfaz gráfica de una aplicación Android. Se puede pensar en una actividad como el elemento análogo a una ventana o pantalla en cualquier otro lenguaje visual. Su función principal es la creación del interfaz de

usuario. Una aplicación suele necesitar varias actividades para crear el interfaz de usuario. Toda actividad ha de pertenecer a una clase descendiente de Activity. Este componente tiene un ciclo de vida y podemos controlar este ciclo dependiendo de la situación en la cual se encuentre. En la siguiente imagen se pueden observar el ciclo de vida de un activity:

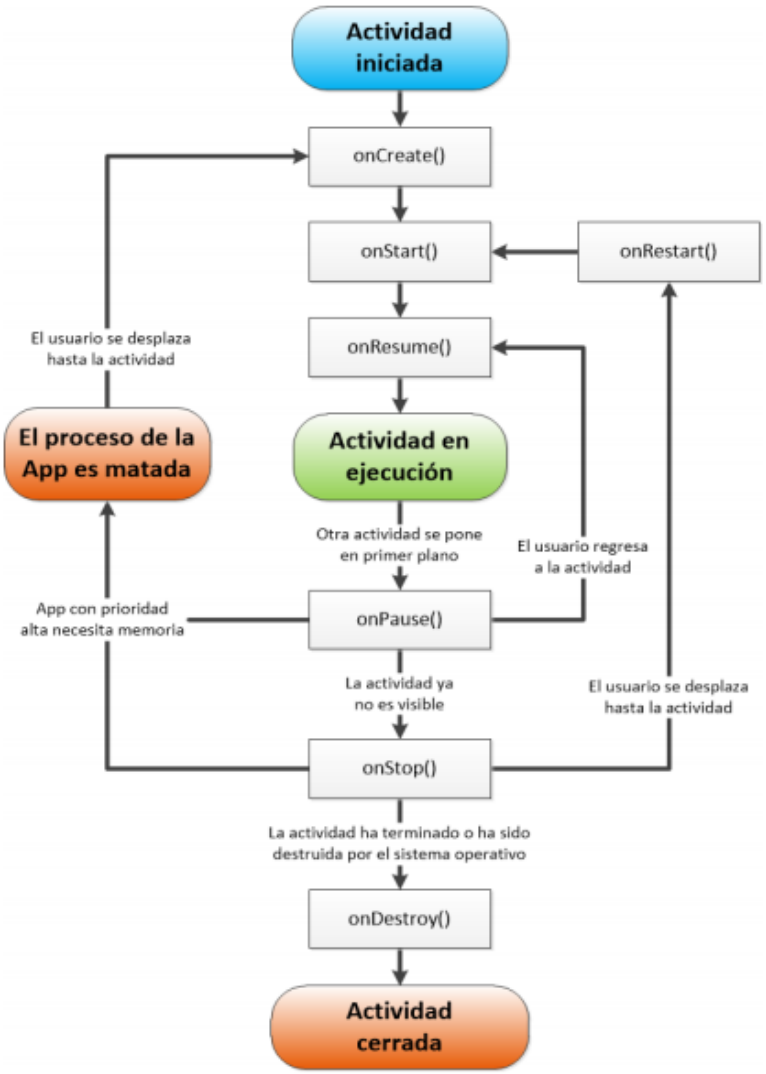


Figura 3.4: Figura: Ciclo de vida de una Activity

3.2.5. Android Studio

El software elegido para la implementación de este proyecto es Android Studio, ya que contiene todas las funcionalidades que se necesitan para generar una

aplicación a la par de ser la aplicación más extendida y más documentada. Android Studio es un entorno de desarrollo integrado (IDE). Basándose en IntelliJ IDEA, es el entorno oficial para el desarrollo de aplicaciones Android como la que motiva este trabajo. Lo que le distingue por encima de su escasa competencia, es tanto la potencia del editor de códigos como las herramientas para desarrolladores de IntelliJ, que proporciona bibliotecas de código muy útiles y enfocadas absolutamente al desarrollo de aplicaciones Android. Está disponible para distintas plataformas; Windows, Mac OS X y GNU/Linux. Básicamente está diseñado para el desarrollo de Android.

Android Studio se basa en cierta medida en el software IntelliJ IDEA de JetBrains, lo que proporciona a nuestra herramienta un soporte mejor para los HiDPI, es decir, una mejora en la parte visual del software en pantallas de ordenador. En resumen, esto supone conseguir una interfaz mejorada y pensada para que el usuario tenga una mejor experiencia cuando se disponga a programar.

Hay varias propiedades a destacar en este software: la integración de ProGuard, que permite eliminar todas las clases que no se utilizan del APK, la inclusión de herramientas para firmar aplicaciones y el renderizado en tiempo real son las más llamativas e importantes.

Además hay otras muchos cuyo valor cualitativo es muy positivo, como puede ser la interfaz gráfica, la manera de refactorizar de manera única para Android, así como el editor de diseño enriquecido (lo cual era especialmente demandado por el conjunto de desarrolladores de Android Studio desde sus inicios), que añade controles más eficientes para arrastrar y soltar.

No obstante sus ventajas no acaban aquí. Existen diferentes paquetes de herramientas incluidos (por ejemplo las Lint para detectar problemas de rendimiento) y soporte para diferentes plataformas como Android Wear o Google Cloud Platform, la que permite la integración con Google Cloud Messaging y App Engine.

No obstante eso no es todo. Posiblemente su mayor fortaleza sea la capacidad de evolución y reciclaje contínua, así como de implementación de aspectos innovadores, pues en cada versión se suelen incluir funcionalidades novedosas y mejoras en las antiguas.

Una de las mejoras más significativas de las últimas betas es el nuevo sistema para compilar y desplegar la aplicación, lo que reduce significativamente el tiempo de espera para ver el resultado de la aplicación.

Otra mejora destacable es la concerniente al despliegue de medios; las librerías y otros códigos son visualizados de forma más ágil, lo cual acelera el trabajo de los desarrolladores.

Como ya hemos dicho la preocupación de Google por sus desarrolladores es máxima, pues su prioridad es que la creación de aplicaciones para su tienda sea lo menos engorrosa posible, lo cual se constituye como el principal motivo para que Android Studio evoluciona constantemente y se adapte a los requerimientos de la comunidad de desarrolladores.

De esta forma, el IDE de Android, presenta novedades como Instant Run, una herramienta que posibilita revisar todos los cambios que sufra código casi al instante en un emulador, pudiendo editar código y ver las consecuencias inmediatamente. Además, es esta última actualización ha conseguido que sea más rápido escribir código en él, (de 2 a 2.5 veces más rápido), y actualizar el código podría llegar a ser incluso 50 veces más veloz. Así mismo se ha añadido soporte para las plataformas de Android Auto y Android Wear.

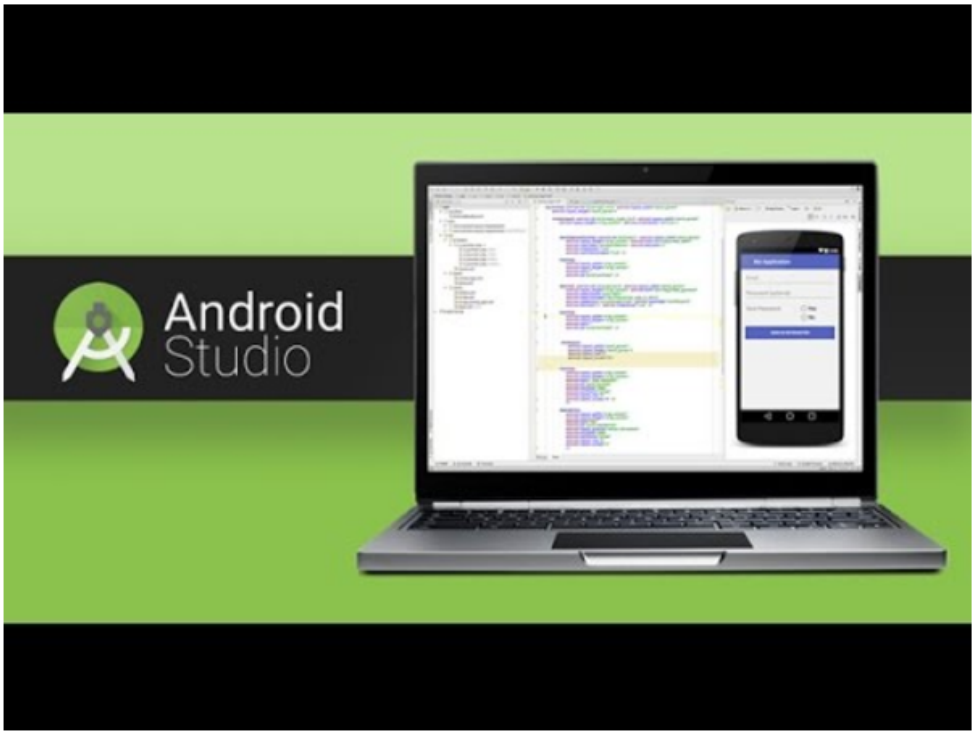


Figura 3.5: Figura: Android Studio: Interfaz

Además, el software ofrece aún más funcionalidades diseñadas para aumentar la productividad durante la compilación de aplicaciones Android, entre las que destacan las siguientes:

1. Sistema de compilación con base en Gradle.
2. Un emulador eficaz con distintas funcionalidades de gran utilidad.
3. Entorno unificado que posibilita un desarrollo común para todos los dispositivos Android.
4. Instant Run para aplicar cambios de manera concurrente sin que haya que compilar nuevamente.
5. Integración de plantillas y repositorios de GitHub, lo cual ha sido una gran ventaja de cara a este proyecto.
6. Multitud de herramientas y frameworks de prueba.
7. Biblioteca Lint para detectar problemas de rendimiento y monitorizar la aplicación.

8. Compatibilidad con C++ y NDK

3.3. Node.js

Como entorno de ejecución para el servidor se ha utilizado Node.js. Este entorno desarrollado para JavaScript está construido con el motor de JavaScript V8 de Chrome. Node.js se basa en un modelo de operaciones de entrada salida orientado a eventos y sin bloqueo, lo cual lo hace bastante robusto y eficaz. El conjunto de paquetes de Node.js, npm, conforma el paradigma de librerías de código abierto más grande a nivel mundial.

Figura 3.6: Figura: Node Js

Entre otros aspectos técnicos, cabe destacar los siguientes:

1. Concurrencia: Sin lugar a dudas es el aspecto de Node.js más importante de cara a un servidor. El funcionamiento es análogo al de un modelo de evaluación de un único hilo de ejecución; usa entradas y salidas capaces de ejecutarse concurrentemente sin penalizar en tiempos relacionados con el cambio de contexto. Aunque de cara a nuestra aplicación este diseño es excesivamente potente, responde a la necesidad de otras aplicaciones altamente concurrentes que requieren que cualquier operación que incluya entradas y salidas tenga una función de callback.

El único inconveniente que tiene este enfoque es que requiere módulos complementarios para escalar la aplicación con la cantidad de núcleos de procesamiento de la computadora en la que está corriendo, pero es algo que afecta a más alto nivel.

2. V8: El entorno de ejecución para Js desarrollado para Google Chrome es V8. Es un software libre escrito en C++ que compila el código source de JavaScript en código de máquina en lugar de interpretarlo en tiempo real, lo cual supone un gran avance.

   Para el manejo de eventos asíncronos, Node.js alberga libuv. Libuv es una biblioteca de soporte multiplataforma con enfoque a entradas/salidas asincrónicas con funcionalidades de redes y sistemas de archivo en sistemas Windows y sistemas basados en POSIX.

3. Módulos: Node.js contiene ciertos "módulos básicosçomo el módulo de red que se compilan en el propio binario y proporciona una capa para programación de red asíncrona.Además proporciona otros módulos de carácter fundamental, como por ejemplo FileSystem, Path, o Stream (con propósito más genérico).

   Otra característica destacable es la posibilidad de utilizar módulos ajenos desarrollados externamente mediante archivos ".node"precompilados, o como archivos en javascript plano. Estos módulos pueden ampliar la funcionalidad de node.js o directamente agregar un nivel nuevo de abstracción mediante la implementación de varias utilidades middleware que faciliten el uso web. Un ejemplo de ellos son los frameworks connect y express. Aunque los módulos pueden ser instalados como archivos simples, lo recomendable es instalarlos utilizando el Node Package Manager (npm) que facilita la compilación, instalación y actualización de módulos así como la gestión de las dependencias.

4. Desarrollo homogéneo entre cliente y servidor: Node puede combinarse con bases de datos documentales y con JSON, lo cual permite programar en un entorno de desarrollo JavaScript unificado. Por otra parte facilita la reutilización de código del mismo modelo de interfaz entre el lado del cliente y el lado del servidor.

5. Bucle de eventos: Node se complementa con el sistema operativo de tal

modo que cada vez que un cliente realiza una petición, se ejecuta un callback. No es necesario crear un hilo de ejecución para que cada conexión reciba una asignación de memoria, pues se hace de forma dinámica dentro del entorno de ejecución.

El sistema de gestión de eventos de Node no se llama de forma explícita sino que se activa al final de cada ejecución de una función callback y finaliza cuando ya no quedan eventos pendientes, lo que supone una ventaja frente a otros servidores accionados por eventos.

3.4. SQLite

La necesidad de acoplar la aplicación con una base de datos relacional se antoja indispensable a poco que creció el proyecto. Si bien es cierto que el servidor emula en muchos procesos internos a la base de datos y por tanto disimula en ciertos aspectos la necesidad de incorporar dicho componente, hay ciertos aspectos claves que evidencian la necesidad de que sí la haya.

Tal es el caso de la autenticación. El servidor no proporciona mecanismos para la autenticación, aunque sí que lo haga de forma implícita al operar con un wallet. Por todas estas condiciones se decidió que la herramienta ideal para realizar este acoplamiento era la API SQL Lite, disponibles en el paquete android.database.sqlite.

Creado por D. Richard Hipp, SQLite es una base de datos Open Source compatible con ACID. Usa como base una biblioteca desarrollada en C bastante pequeña (se puede comprobar ya que es de dominio público). Con los años, debido a su gran potencial y su facilidad de uso, ha ganado mucha popularidad en los pequeños dispositivos como Android.

Las ventajas que supone usar SQLite son diversas: no requiere configuración (necesitamos una clase que herede de SQLiteOpenHelper), no tiene un proceso ejecutándose secundariamente y gastando memoria para mantener el servidor de base de datos y su uso es bastante sencillo.

La principal diversidad respecto a los SGBD tradicionales con arquitectura cliente-servidor es la que supone esa segunda ventaja; el motor nativo de SQLite no es un hilo autónomo al que llama el programa principal, sino que es la propia biblioteca SQLite la que sirve de enlace con el programa pasando a ser parte de él.

La funcionalidad de SQLite se explota vía llamadas a sus funciones, lo que supone una reducción de la latencia en el acceso a la base de datos. El conjunto de elementos relativos a la base de datos se almacenan como un único fichero estandarizado en la máquina origen. Este trivial diseño se lleva a cabo mediante el bloqueo de todo el fichero de base de datos al inicio de cada transacción.

En su versión 3, SQLite permite bases de datos de hasta 2 Terabytes de tamaño, y también permite la inclusión de campos tipo BLOB.

La clase SQLiteOpenHelper requiere que se implementen dos métodos obligatoriamente onCreate y onUpgrade,y dos constructores que tienen parámetros.

3.5. API Blockchain.info

Para este proyecto se ha congregado la tecnología blockchain con la implementación de un monedero de bitcoins. Para tal fin, se ha usado una API pública para desarrolladores de BTC, la de blockchain.info. En este apartado haremos un resumen de cómo funciona esta API y de cuál ha sido su uso.

Figura 3.7: Figura: Logo de Blockchain.info

Esta API ofrece repositorios para múltiples funcionalidades relacionadas con los servicios habituales de las criptomonedas. Estos son algunos de ellos:

- Procesar pagos: Proporciona un método sencillo para recibir pagos en Bitcoin desde un sitio web. A partir de una clave pública extendida (xPub), te generan una dirección única, no utilizada, para que sus clientes envíen los pagos. Así mismo, notifican los pagos a esa dirección al instante usando una URL de devolución de llamada configurable.
- Cartera de bloques: Es la API para enviar y recibir pagos de carteras de blockchain. Es la que utilizaremos y por tanto se explica en profundidad en la siguiente sección.
- Transacciones y datos de bloques: Proporciona métodos para el tratamiento de bloques y transacciones. Para ello, ofrece tres APIs distintas:

- API de datos de Blockchain: Consulta datos de JSON sobre bloques y transacciones, lo cual proporciona mucha versatilidad, pues casi todos los elementos web pueden transformarse a formato JSON.
- API de consultas simple: API de texto plano para consultar datos de blockchain.info.
- Websockets: Socket channel de transmisión de baja latencia que proporciona datos sobre bloques y transacciones nuevas. Puedes suscribirte a la notificaciones de bloques y transacciones de tal manera que recibirás objetos descriptivos provocados por eventos en formato JSON.

- Datos del Mercado: Ofrece datos monetarios de los mercados mayoritarios de BTC.

3.5.1. Blockchain Wallet API - Java

El API Wallet de Blockchain aporta una interfaz sencilla para ser usada en la interacción mediante programación con una billetera. Para usar esta interfaz, es necesario instalar un servidor local que acepte las peticiones que gestionan el Wallet Blockchain. La aplicación interactúa con el servidor localmente mediante llamadas HTTP.

3.5.1.1. Servidor Blockchain: blockchain-wallet-service

Ya se ha introducido esta parte en la sección de Node.js, que hacía referencia más a qué tecnología implementa internamente. Como información complementaria, vamos a hablar un poco del funcionamiento del servidor de cara al usuario.

Al aceptar peticiones HTTP, cualquier lenguaje debidamente acoplado puede hacer uso del servidor. En este caso se ha utilizado en lenguaje Java, pero no sería complejo adaptar otros lenguajes de programación como Python, C, Ruby, PHP o Node.

No hay que olvidar que el servidor está diseñado para ejecutarse localmente en la misma máquina que la aplicación y, por consiguiente, sólo aceptará conexiones de localhost. Podría modificarse el servicio para aceptar conexiones externas, pero entonces habría que asegurarse de que las reglas del firewall son correctas.

Aunque para hacer un uso básico y a pequeña escala no es necesario un código API, sí que se requiere para la creación de una billetera con límites de solicitud más altos. En nuestro caso se ha solicitado como ejercicio complementario, aunque no era necesario.

A continuación vemos una captura del servidor operativo:

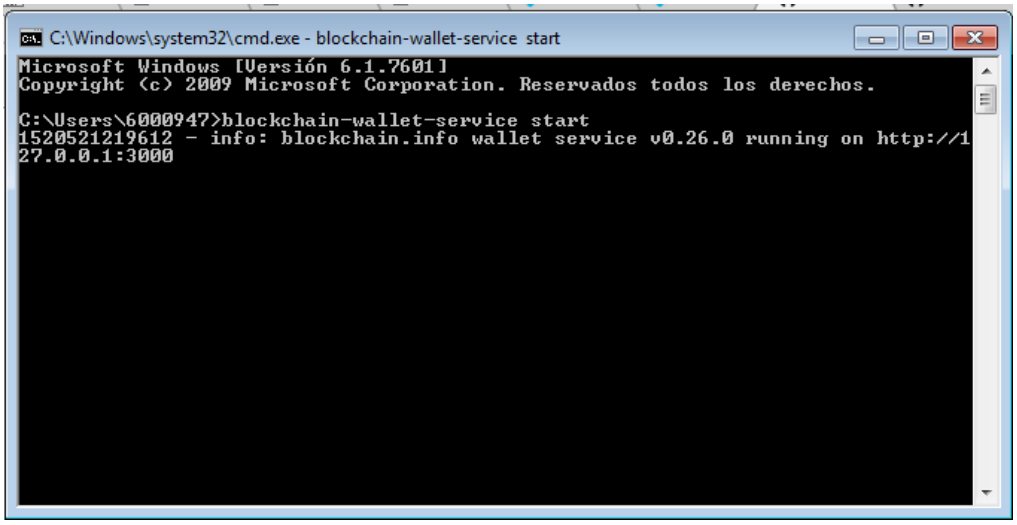


Figura 3.8: Figura: Servidor corriendo en el puerto 3000

3.5.1.2. Métodos de la API

La API proporciona métodos para realizar las actividades usuales en el ámbito de un monedero electrónico. Las principales las usamos en la aplicación del proyecto, aunque el modo "sandbox"para BTC no está muy testeado y falla bastante.

En general, esta interfaz proporciona métodos para crear un Wallet y luego operar con él. Será imprescindible como premisa crear un Wallet o tener la dirección guid de uno ya creado para poder hacer uso del resto de operaciones, pues lógicamente no es posible hacer operaciones con un monedero si no tienes monedero.

Hay que diferenciar entre los métodos que proporciona la API oficial, la API utilizada (un subconjunto de la primera), y el servidor. Aunque todo hace referencia a un mismo paradigma e, idealmente, debería funcionar indistintamente, se ha constatado que existe cierta desactualización entre todas; el servidor no siempre acepta los parámetros de entrada que según la API oficial debería aceptar, y tampoco ofrece siempre la respuesta esperada. Por ejemplo, al crear un

nuevo wallet se crea una dirección principal asociada; según la API debería retornar una etiqueta junto al código guid y a la dirección, pero no lo hace.

Esto supuso un problema inicialmente, pues la API espera ese parámetro y al parsearlo provoca un "nullpointer"que haría fallar la aplicación. Tras contactar con los desarrolladores de la API, se constató que realmente no tenía por qué devolver ese parámetro y por tanto era un error de implementación. Esto se podía solucionar .ºbligando.ª meter la etiqueta como parámetro de entrada. Otra opción era volver a implementar internamente el primer método y que generase una etiqueta aleatoriamente en caso de no ser informada, y luego ya llamase al método principal de la API. Se decidió crear una solución intermedia para garantizar el correcto funcionamiento de la aplicación y además el uso de la API, de tal forma que permitiese introducir el campo adicional "labelz no introducirlo e igualmente hacer un funcionamiento correcto.

El segundo problema encontrado fue para con las creaciones de las nuevas direcciones asociadas a un wallet-guid. En este caso, el único parámetro de entrada es la nueva etiqueta asociada a esa nueva dirección. Esto es así porque el resto de parámetros de entrada necesarios (contraseña principal y secundaria si la doble encriptación está activada) ya están implícitas en el wallet al iniciar la sesión. El inconveniente surge a la hora de añadir esta nueva dirección al conjunto de direcciones del wallet. Por defecto, todas las direcciones se generan, lógicamente, con una cantidad inicial de 0 BTC tanto en los campos de "balancez de "totalreceived". Sin embargo, se descubrió que en primera instancia y erróneamente estos dos campos asociados a la dirección se crean con un valor de "nullz no de 0, como se indica en código. Este error se subsana solo al hacer alguna actualización interna del servidor, pero si intentas acceder a esa dirección en ese periodo en el que aún está a null, fallará.

Tras esta breve introducción, enumeramos ahora los métodos de la API, sus parámetros y funcionamiento, así como el uso que se le ha dado internamente en la aplicación móvil:

1. Create Wallet API: El método create wallet se puede usar para crear una nueva billetera bitcoin del tipo blockchain.info. El endpoint asociado acepta peticiones tanto POST como GET. Los parámetros que acepta son los siguientes: password, api code, priv, label y email, siendo solamente los dos primeros obligatorios.

   La contraseña -password-, será la contraseña asociada al nuevo wallet. De

carácter alfanumérico, debe tener al menos 10 caracteres de longitud. Es un campo externo, pues es algo que el usuario elige al crear un nuevo wallet y que usará siempre para entrar.

El código de API es el parámetro del cual hablamos anteriormente en la sección del servidor. Es un parámetro que, para un uso local limitado, nos podemos inventar, aunque lo recomendado es solicitar uno que será oficial y no estará capado. En cualquier caso, el código debe tener permisos para crear wallet (por defecto lo tiene). El código usado en la aplicación es proporcionado por la plataforma blockchain.info exclusivamente para este proyecto.

El campo priv, es una clave privada opcional para agregar a la billetera. Este campo será obligatorio a no mucho tardar, pues la encriptación de un monedero se antoja imprescindible y con una segunda contraseña se blinda la seguridad mucho más.

Del parámetro label también hemos hablado anteriormente. Es recomendable usarlo siempre porque constituye una identificación más amigable para el usuario de cara a las direcciones. Es simplemente una descripción y por tantos no afecta a la funcionalidad, pero de cara a la usabilidad ayuda mucho.

Adicionalmente y emulando al anterior, el email representa un identificador amigable para con el wallet. Análogamente al label, es opcional pero también es altamente recomendable su uso.

Se realizará una llamada al servidor cuyo endpoint será análogo al siguiente: http://localhost:3000/api/v2/create, y, si todos los parámetros son correctos y va todo bien, el servidor proporcionará una respuesta en formato JSON con la dirección guid del nuevo wallet en un formato semejante al IBAN de los bancos pero combinando letras y números, el código de la dirección principal que sea crea asociada a ese wallet, así como la etiqueta asociada a dicha dirección (la introducida o una generada aleatoriamente si no sucedió lo primero).

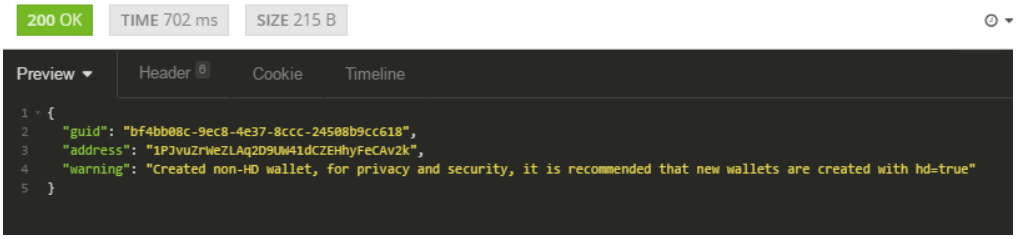


Figura 3.9: Figura: Android Studio interfaz

Este método lo usaremos en principio una sola vez para crear un wallet, ya que el resto de operaciones se pueden hacer dentro del wallet. En nuestra interfaz, localizamos el acceso a este método en la pantalla principal, la que te permite acceder a tu wallet o crear uno si no lo tienes.

2. Making Outgoing Payments: Envía bitcoin desde tu billetera a otra dirección de bitcoin. Todas las transacciones incluyen una tarifa de 0.0001 BTC para los mineros y

   todos los valores de bitcoins están en Satoshi, es decir, se dividen por 100000000 para obtener la cantidad en BTC.

   Esta parte de la interfaz supone la acción principal que se puede hacer con un wallet, es decir, hacer pagos (pues recibir pagos es algo intrínseco y externo). De cara a esta funcionalidad, necesitaremos las siguientes variables:

   - main password: contraseña principal, una vez autenticado no será necesario volver a introducirla.
   - second password : si y solo sí la doble encriptación está habilitada y de manera análoga a la contraseña principal, esto es, no sería necesario volver a introducirla.
   - to: la dirección a la que vas a enviar el dinero. Puede ser de tu propio wallet o de otro. Esta dirección no es la del posible email asociado a un wallet, es el identificador de una dirección dentro de un wallet.
   - amount: la cantidad a enviar en satoshi.
   - from: la dirección de origen dentro de tu wallet. Es un parámetro opcional; si no seleccionas se cogerá la principal.
   - fee: Otro parámetro opcional. Por defecto se pone el porcentaje ya mencionado, así que si lo informas debe ser mayor a ese porcentaje. Es la tasa por la transacción en satoshi.

En este caso será la primera opción que tengamos en la interfaz principal de la aplicación. La aplicación hará las validaciones de formato de dirección y de cantidad, y el servidor de si es posible realizar esa transacción, es decir, si tienes dinero suficiente para ello. Obviamente para esto hay que usar un modo Sandbox que como ya se ha adelantado está un poco cogido con pinzas en blockchain.info.

La llamada al servidor tendrá como destino el endpoint /merchant/guid/payment, en el cual se usará el guid obtenido en la creación del wallet para hacer la llamada. Si todo va bien, el servidor dará una respuesta de 200 OK y un mensaje en JSON con un mensaje informativo de respuesta (por ejemplo,"Sent 0.1 BTC to 1A8JiWcwvpY7tAopUkSnGuEYHmzGYfZPiq"), un código hash asociado a la transacción y un mensaje adicional si lo hubiere (por ejemplo, "Some funds are pending confirmation and cannot be spent yet (Value 0.001 BTC)").

3. Send Many Transactions: Envía una transacción a múltiples destinatarios en la misma transacción.

   Respecto al apartado anterior, el único cambio es que en este caso no hay un campo para la dirección y otro para el dinero, sino que se unifica en un único campo que contiene un número indeterminado de pares direccion-dinero, que almacenaremos en un mapa para hacer tantos envíos como se requieran en la misma transacción.

   De cara a la parte técnica, esta funcionalidad se implementa con un botón que te abriese un cuadro de diálogo en el cual informaban los dos campos, dirección y cantidad y luego previsualizar el búfer entero con la cadena concatenada de todos los pagos a realizar, pudiendo limpiarse y volver a construirse.

Figura 3.10: Figura: Respuesta del servidor ante un envío múltiple

Como observamos en la figura, la respuesta del servidor en este caso no es exactamente como en el anterior; en este caso ofrece dos respuestas, una muy semejante de confirmación de transacción como en el caso del pago

único y otra de manera adicional de las direcciones introducidas y las cantidades enviadas en una lista de pares (siempre que la transacción se haya completado exitosamente).

El ejemplo enviaría 1 BTC a 1JzSZFs2DQke2B3S4pBxaNaMzzVZaG4Cqh, 15 BTC a 12Cf6nCcRtKERh9cQm3Z29c9MWvQuFSxvT y 2 BTC a 1dice6YgEVBf88erBFra en la misma transacción.

4. Fetching the wallet balance: Obtiene el saldo de una billetera. Esto debe usarse solo como una estimación e incluirá transacciones no confirmadas y posiblemente gastos dobles.

   A partir de la dirección de guid y la contraseña principal, el endpoint balance te ofrece una respuesta con un campo descripción ("Wallet Balance in Satoshi") y otro con la cantidad. Para ello hace un cálculo internamente del saldo de todas las direcciones asociadas a ese Wallet y las suma antes de enviarlas en la respuesta. En la aplicación, hay una opción destinada a eso que te ofrecerá esa información en un mensaje de cuadro de diálogo.

5. Listing Addresses: Enumera todas las direcciones activas en una billetera. También incluye un saldo de confirmación de 0 que debe usarse sólo como un estimado e incluirá transacciones no confirmadas y posiblemente gastos dobles.

   En este caso la respuesta del servidor son varios objetos encapsulados en un objeto principal .ªddresses"que muestran la dirección, el saldo, la etiqueta y el total recibido en dicha dirección. En la aplicación hacemos el listado mediante campos dinámicos, pues no sabemos cuántas direcciones hay en un momento dado. Además, añadimos el botón de archivar todas en conjunción con la funcionalidad de archivar dirección.

6. Getting the balance of an address: Recuperar el saldo de una dirección bitcoin. La consulta del saldo de una dirección por etiqueta se depreca. Ofrece la misma información que la opción anterior con la salvedad de que se omite la etiqueta, pero al fin y al cabo la llamada y respuesta es muy semejante.

7. Generating a new address: En este caso el único parámetro necesario es la etiqueta con la que se creará la nueva dirección, pues la contraseñas y el guid como ya hemos dichas están en el scope y no hace falta introducirlas para todas las operaciones. La etiqueta según el servidor y la API no es

obligatoria, pero por razones de legibilidad y como seguridad adicional, se decide que en la aplicación sí que sea obligatorio. El endpoint ofrecerá una respuesta con la nueva dirección generada asociada a la etiqueta introducida, así como los campos tota_received y balance a 0 (que como hemos indicado anteriormente tiene un pequeño fallo).

8. Archiving an address: Para mejorar el rendimiento del monedero, las direcciones que no se han utilizado recientemente se deben mover a un estado archivado. Se mantendrán en la billetera, pero ya no se incluirán en las llamadas "lista.º "listas de transacciones".

   Por ejemplo, si se genera una factura para un usuario una vez que se paga la factura, la dirección debe archivarse.

   O bien, si se genera una dirección bitcoin única para cada usuario, los usuarios que no hayan iniciado sesión recientemente (30 días) sus direcciones se deben archivar.

9. Unarchive an address: Desarchivar una dirección. También restaura las direcciones consolidadas.

3.5.2. Funcionalidad HD

En el transcurso del trabajo se constata que la API explicada en el apartado anterior es totalmente funcional, segura y está operativa. Además, al llevar mucho tiempo operativa, está perfectamente testeada, lo que nos hace intuir su robustez. No obstante, al hacer uso de los endpoints arriba indicados, el servidor complementa la respuesta con un mensaje adicional: "This endpoint has been deprecated, for the best safety and security, use the HD API instead: https://github.com/blockchain/service-my-wallet-v3#enable-hd-functionality"

Investigando se documenta que esta nueva funcionalidad que opera sobre la misma base que nuestra API, es un complemento de seguridad como valor añadido, por lo que se depreca el anterior. El cambio sustancial de esta versión es que actualizará una billetera a una billetera HD (Hierarchical Deterministic):

Las Billeteras HD"son wallets que se pueden compartir parcial o totalmente con diferentes sistemas, cada uno con o sin la capacidad de gastar monedas. La especificación está destinada a establecer un estándar para billeteras deterministas que pueden intercambiarse entre diferentes clientes.

La especificación consta de dos partes. En una primera parte, se presenta un sistema para derivar un árbol de pares de llaves de una sola semilla. La segunda parte demuestra cómo construir una estructura de billetera encima de dicho árbol.

Estos wallets tienen muchas características, aunque no todas son requeridas por todos los clientes de soporte.

Los endpoints no son los mismos, aunque los parámetros para llamarlos son prácticamente iguales con ligeras discrepancias en algunas casos. Con la respuesta de este nuevo endpoint pasa de forma análoga, prácticamente es la misma.

3.5.2.1. Necesidad de interfaz HD

El cliente habitual de Bitcoin usa claves generadas aleatoriamente. Para evitar la necesidad de una copia de seguridad después de cada transacción, (de manera predeterminada) 100 claves se almacenan en la memoria caché en un pool destinado para tal fin. Aún así, estas billeteras no están diseñadas para ser compartidas y utilizadas en varios sistemas simultáneamente. Implícitamente se acepta ocultar las claves privadas mediante el uso de la función de cifrado de monedero aunque conlleve no compartir la contraseña, pero esto supone que estos wallets estén çastradosz pierdan el poder de generar claves públicas.

Los monederos deterministas (HD) no requieren tales backups contínuo, y la función matemática de curva elíptica permite crear esquemas donde es posible calcular las claves públicas sin revelar las claves privadas. Esto permite, por ejemplo, que un negocio de tienda online genere nuevas direcciones (claves públicas) para cada pedido o para cada cliente, sin dar acceso al servidor web a las claves privadas correspondientes (que son necesarias para gastar los fondos recibidos).

Sin embargo, los Wallets HD generalmente consisten en una única çadena"de pares de llaves. El hecho de que haya una sola cadena significa que el compartir una billetera se produce sobre una base de todo o nada. Sin embargo, en algunos casos uno solo quiere que algunas claves (públicas) sean compartidas y recuperables. En el ejemplo de la tienda online, el servidor web no necesita acceso a todas las claves públicas del wallet del comerciante; sólo a aquellas direcciones que se utilizan para recibir pagos de clientes, y no por ejemplo las direcciones de cambio que se generan cuando el comerciante gasta dinero. Las billeteras de-

terministas jerárquicas permiten un intercambio selectivo al admitir múltiples cadenas de pares de claves, derivadas de una única raíz.

No obstante y además de ser una tecnología que está relativamente nueva y por tanto en fase beta, esta seguridad no es necesaria de cara al proyecto a pequeña escala que aquí se desarrolla, motivo por el que se decir seguir usando la API anterior.

Capítulo 4

Análisis y diseño

El desarrollo de la aplicación de gestión de un monedero multidireccional Android conlleva la superación de varios hitos que debemos definir de forma clara antes de comenzar el diseño. Para ello se van a identificar los principales obstáculos que se deben alcanzar de manera individual.

Antes de nada vamos a explicar la composición típica de un monedero electrónico para poder entender mejor el diseño a realizar. Como se ha explicado en el apartado de la API, un monedero consta de un identificador de con un formato como el del tradicional IBAN pero incluyendo caracteres alfanuméricos y no sólo números (necesario para todas las operaciones), un conjunto de direcciones con identificadores alfanuméricos y sus identificadores internos amigables (etiquetas) y los campos de total recibido y cantidad en ese instante.

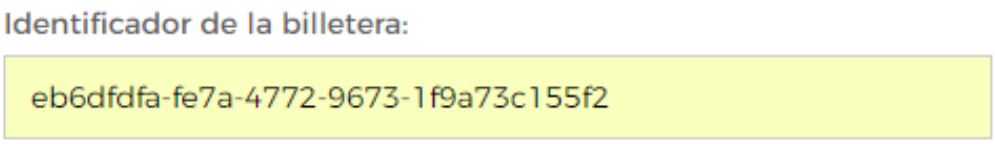


Figura 4.1: Figura: Ejemplo de identificador guid de una billetera

Por otra parte, el sistema debe realizar la función de wallet propiamente dicha. Esta función, básicamente, consiste en realizar transacciones monetarias entre direcciones internas dentro del mismo wallet y externas entre distintos wallets, dejando siempre coherentes los valores totales (un principio básico de BTC es que el número total de BTC es finito e inalterable).

Para ello habrá que tener una interfaz a través de una aplicación móvil desarrollada en Android que gestione toda esa información y permita realizar las

operaciones clásicas de un monedero y las operaciones asociadas a un monedero electrónico (todas ellas se explicaron en la sección de API Blockchain).

Para poder almacenar esta información se debe crear una estructura de datos global la cual contenga la información relativa a todos los wallets creados mediante la aplicación, o, dicho de otra manera, la información que gestiona el servidor: identificadores de wallets, direcciones, cantidades...

La comunicación entre servidor y aplicación se tiene que desarrollar de forma intrínsecamente segura, lo cual controlaremos de forma local, pues no hay interacción con agentes externos.

4.1. Casos de uso

Hay que plantear los casos de uso de la herramienta para gestión de un Wallet BTC. Los casos de uso no son más que la representación de la interacción y comunicación de los usuarios de la aplicación con la misma. Mediante el listado de los casos de uso se persigue alcanzar un análisis de requisitos lo más completo posible:

Cuadro 4.1: Acceder a la aplicación

Casos de uso	Autenticarse (Authentication)
Resumen	Acceder a la aplicación mediante id/contraseña
Actores	Usuario
Precondición	El usuario dispone de un guid y una contraseña asociada en la base de datos
Postcondición	El sistema garantiza la validez de los datos y el usuario accede
Curso Normal	1.- El usuario abre la aplicación. 2.- El usuario rellena los datos de guid y password correctamente. 3.- El usuario pulsa el botón .Entrar". 4.- La aplicación llama a la base de datos para corroborar los datos y después transmite dicha validación al servidor. 5.- El servidor realiza una segunda validación interna y recupera los datos del wallet. 6.- La aplicación deja acceder al usuario para que gestione su wallet.
Curso Alternativo	1.- La base de datos informa de que ese usuario no existe. 2.- La aplicación informa de que la contraseña es errónea.
Observaciones	1.- Para todo el proceso será necesario una base de datos de apoyo.

Cuadro 4.2: Enviar dinero

Caso de uso	Enviar dinero (Making Outgoing Payments)
Resumen	Envía bitcoins desde tu billetera a otra dirección de bitcoins
Actores	Usuario
Precondición	El usuario tiene un wallet con tanto dinero o más del que pretenden enviar. La dirección de envío existe.
Postcondición	La cantidad de dinero transferida se resta de la dirección de origen y se suma a la de destino.
Curso Normal	1.- El usuario hace login y accede al menú principal. 2.- El usuario selecciona la opción .enviar dinero". 3.- El usuario rellena los campos dirección de destino y cantidad. 4.- El usuario pulsa el botón GO. 5.- El servidor procesa la información e informa del éxito de la operación y el usuario vuelve a la pantalla principal.
Curso Alternativo	1.- La aplicación informa al usuario de que la dirección de destino no existe o es inválida y debe cambiarla. 2.- La aplicación informa al usuario de que no tiene suficientes fondos para realizar esa transacción.
Observaciones	El servidor debe estar operativo durante todo el proceso.

Cuadro 4.3: Pagos múltiples

Casos de uso	Envío múltiple (Send Many Transactions)
Resumen	Envía un cantidad a múltiples destinatarios en la misma transacción.
Actores	Usuario
Precondición	- El usuario se logea en un wallet en el que tiene tanto o más dinero como pretende enviar. - Las direcciones introducidas para el envío existen.
Postcondición	La cantidad de dinero transferida se resta del wallet y se suma a su correspondiente dirección de destino.
Curso Normal	1.- El usuario hace login y accede al Wallet. 2.- El usuario selecciona la opción "hacer transacciones múltiples." 3.- El usuario pulsa el botón añadir para abrir el cuadro de diálogo. 3.- El usuario genera la cadena de pares dirección-cantidad a partir de los datos introducidos en el paso anterior. 4.- El usuario pulsa el botón GO. 5.- El servidor procesa la información e informa del éxito de la operación.
Curso Alternativo	1.- El usuario pulsa el botón limpiar que destruye la cadena de pares generada. 2.- La aplicación informa al usuario de que alguna de las direcciones introducidas no es correcta. 3.- La aplicación informa al usuario de que no dispone de suficientes fondos para realizar la transacción.
Observaciones	

Cuadro 4.4: Crear Wallet

Caso de uso	Crear Wallet (Create Wallet API)
Resumen	Crear una cartera blockchain
Actores	Usuario
Precondición	- La contraseña debe tener al menos 10 caracteres de longitud. - La plataforma blockchain te debe proveer un código API con permisos para crear wallet.
Postcondición	Se crea el Wallet y se otorga un guid identificativo del mismo y se crea una primera dirección asociada con 0 BTC.
Curso Normal	1.- El usuario entra en la aplicación y selecciona çrear cuenta". 2.- El usuario rellena los campos contraseña y API_CODE. 3.- El usuario pulsa el botón GO. 5.- El servidor procesa la información e informa del éxito de la operación, proporciona el guid del wallet creado y el usuario vuelve a la pantalla principal.
Curso Alternativo	1.- La aplicación informa al usuario de que la dirección de que la contraseña no es válida. 2.- La aplicación permite al usuario solicitar un código de API en la página oficial de blockchain.info si éste no dispone de uno.
Observaciones	El servidor debe estar operativo durante todo el proceso.

Cuadro 4.5: Consultar saldo

Casos de uso	Consultar saldo del Wallet (Fetching the wallet balance)
Resumen	Obtenga el saldo de una billetera.
Actores	Usuario
Precondición	El usuario se ha logeado correctamente en la aplicación.
Postcondición	El usuario ve su saldo total (la suma de todas las direcciones pertenecientes al wallet). h
Curso Normal	1.- El usuario hace login y accede al menú principal. 2.- El usuario selecciona la opción çonsultar saldo". 3.- El servidor procesa la información e informa al usuario del saldo.
Curso Alternativo	1.- La aplicación informa al usuario de que la dirección de que la contraseña no es válida. 2.- La aplicación permite al usuario solicitar un código de API en la página oficial de blockchain.info si éste no dispone de uno.
Observaciones	El servidor debe estar operativo durante todo el proceso.

Cuadro 4.6: Consultar direcciones

Casos de uso	Listar direcciones (Listing Addresses)
Resumen	Enumerar y visualizar todas las direcciones activas en una billetera. También incluye un saldo de confirmación debe usarse solo como un estimado e incluirá transacciones no confirmadas y posiblemente gastos dobles.
Actores	Usuario
Precondición	El usuario tiene una o más direcciones activas en el wallet.
Postcondición	Se visualizan las direcciones y sus datos asociados pero no se actualiza ningún campo.
Curso Normal	1.- El usuario hace login y accede al Wallet. 2.- El usuario selecciona la opción "listar direcciones". 3.- El usuario visualiza todas las direcciones, sus respectivas etiquetas, y su saldo. 4.- El usuario pulsa el botón volver y se vuelve al menú principal.
Curso Alternativo	1.- El usuario pulsa el botón .ªrchivar todasz para que se archiven todas las direcciones que se están visualizando. 2.- La aplicación informa de que no hay direcciones activas y te conmina a crear una.
Observaciones	

Cuadro 4.7: Balance de una dirección

Casos de uso	Obtener saldo de una dirección (Getting the balance of an address)
Resumen	Recuperar el saldo de una dirección bitcoin. La consulta del saldo de una dirección por etiqueta se depreca.
Actores	Usuario
Precondición	La dirección introducida para ser consultada existe y está activa.
Postcondición	La aplicación abre un cuadro con la cantidad de la dirección asociada.
Curso Normal	1.- El usuario hace login y accede al Wallet. 2.- El usuario selecciona la opción "Gestionar direcciones" 3.- El usuario selecciona la opción .ºbtener balance de una dirección". 4.- La aplicación abre un cuadro de diálogo solicitando la dirección a consultar. 5.- El usuario introduce la dirección y visualiza el saldo actual y el histórico recibido.
Curso Alternativo	1.- La aplicación informa de que la dirección no existe y vuelve a solicitar que se introduzca la dirección.
Observaciones	

Cuadro 4.8: Generar dirección

Casos de uso	Crear una dirección (Generating a new address)
Resumen	Crear una dirección en el marco de un wallet.
Actores	Usuario
Precondición	No existe una dirección con la misma etiqueta introducida.
Postcondición	La dirección se crea y se asocia al Wallet.
Curso Normal	1.- El usuario hace login y accede al Wallet. 2.- El usuario selecciona la opción "Gestionar direcciones" 3.- El usuario selecciona la opción "Generar una nueva dirección". 4.- La aplicación abre un cuadro de diálogo solicitando una etiqueta para esa dirección. 5.- El usuario introduce la etiqueta y pulsa GO. 6.- El servidor genera automáticamente un identificador de dirección asociada a ese Walley, con la etiqueta introducida y con 0 BTC.
Curso Alternativo	1.- La aplicación informa de que la etiqueta ya existe y conmina a introducir otra.
Observaciones	

4.2. Diagramas de secuencia

En la siguiente sección se representará mediante formato gráfico los diagramas de secuencia asociados a los escenarios descritos en los casos de uso de la sección anterior. De esta manera se ven claramente la relación entre los agentes y la comunicación que se realiza entre ellos en cada acción que el usuario ejecuta en la aplicación.

Se puede apreciar como destacan 4 actores principales: usuario, aplicación, servidor y base de datos.

En este caso la presencia del servidor se antoja como clave, pues dependiendo del escenario hace funciones de validación, de agregado o incluso llega a emular a la base de datos por tener su propio base de datos integrada internamente.

De ahí se extrapola que nuestra base de datos con SQLite tiene un carácter secundario y muchas veces casi presencial, puesto que su función se restringe fun-

damentalmente al almacenamiento de los identificadores y las contraseñas de los Wallets (pese a que la autenticación que se hace aquí no es definitiva puesto que se tendrá que autenticar también en el servidor).

Los otros dos implicados en las funcionalidades descritas son el usuario, que es cualquiera que se instale la aplicación y desee hacer cualquiera de las gestiones pertinentes de un wallet, y la aplicación, la protagonista de esta memoria. La aplicación es una interfaz gráfica que se encarga de gestionar la interacción del usuario y el servidor, haciéndola amigable y descriptiva visualmente. Especialmente importante es que las funcionalidades desarrolladas en el servidor y utilizadas mediante la API, sean gestionadas correctamente y siguiendo el mismo patrón de comportamiento que el servidor; para que la lógica de negocio responda a la infraestructura tecnológica, la aplicación y el alineamiento de datos debe estar normalizado y ser total.

En el primer caso vemos como sería el modelo de autenticación de un escenario en el que el usuario ya conoce su guid y su contraseña. Vemos cómo se realiza una doble validación: una explícita en base de datos y otra implícita en el servidor cuando comprueba si hay datos asociados a ese guid.

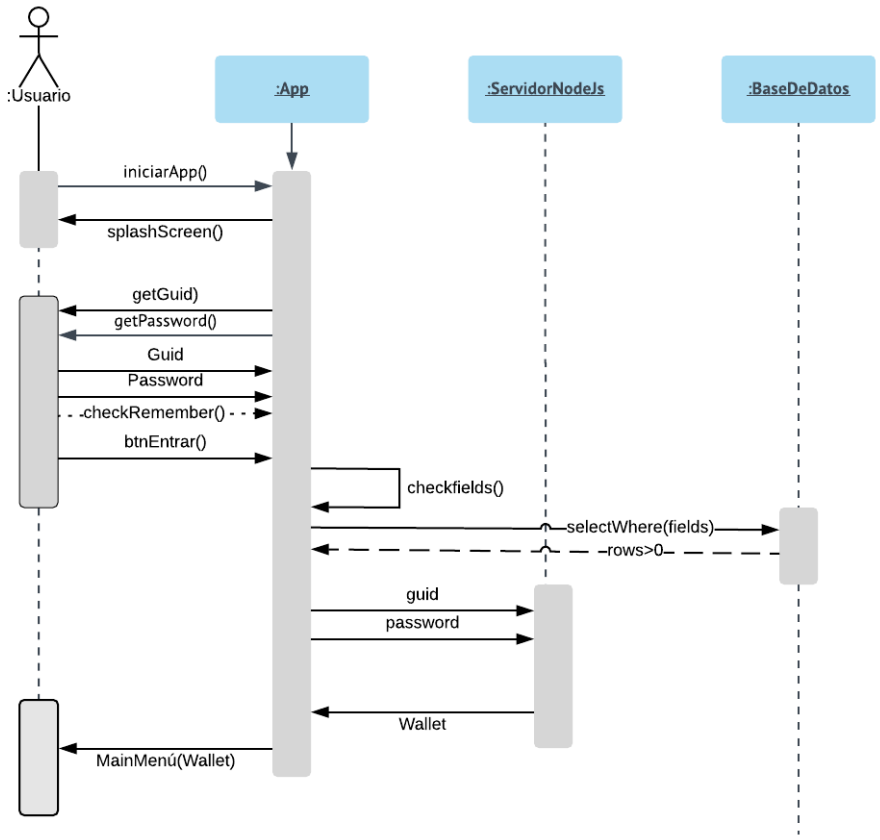


Figura 4.2: Figura: Diagrama de secuencia: Autenticación

En este segundo caso podemos ver cómo se comportaría el sistema ante una petición de consulta de saldo de un wallet, es decir, de la suma de todas las cantidades que hay en las direcciones.

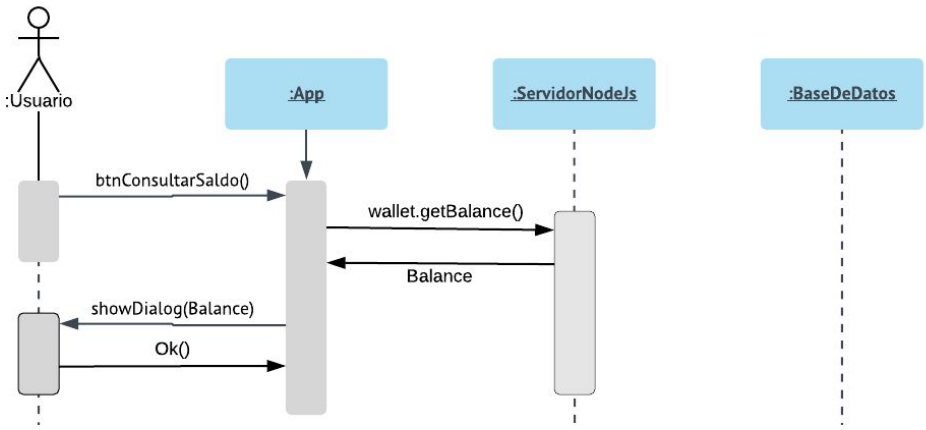


Figura 4.3: Figura: Diagrama de secuencia: Wallet balance

Este es el caso más visual en el que la interacción entre todos los actores es total. Se trata del escenario de crear una nueva cartera. Para ello hay que tener una código API o solicitarlo.

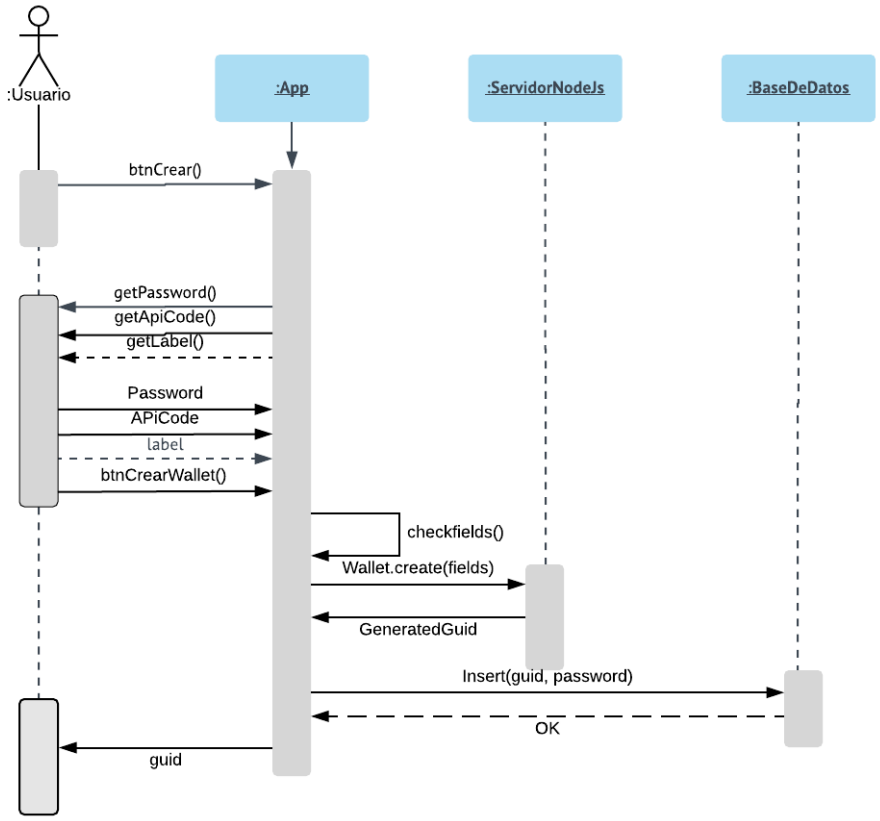


Figura 4.4: Figura: Diagrama de secuencia: Crear Wallet

Los siguientes casos describen lo relativo a la gestión de direcciones: crear, archivar y listar.

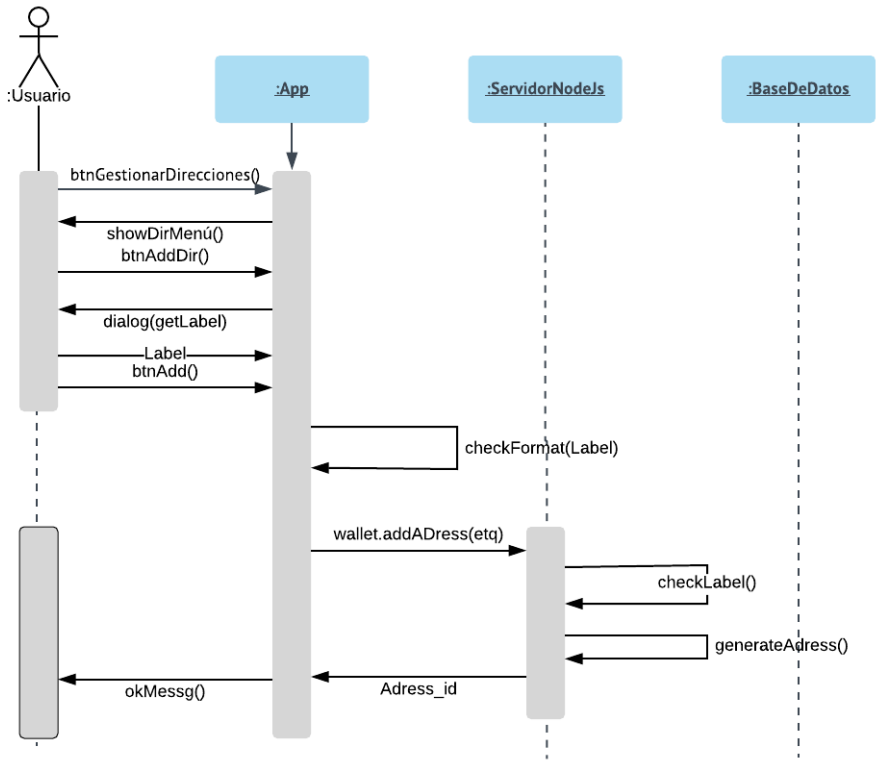


Figura 4.5: Figura: Diagrama de secuencia: Generar dirección

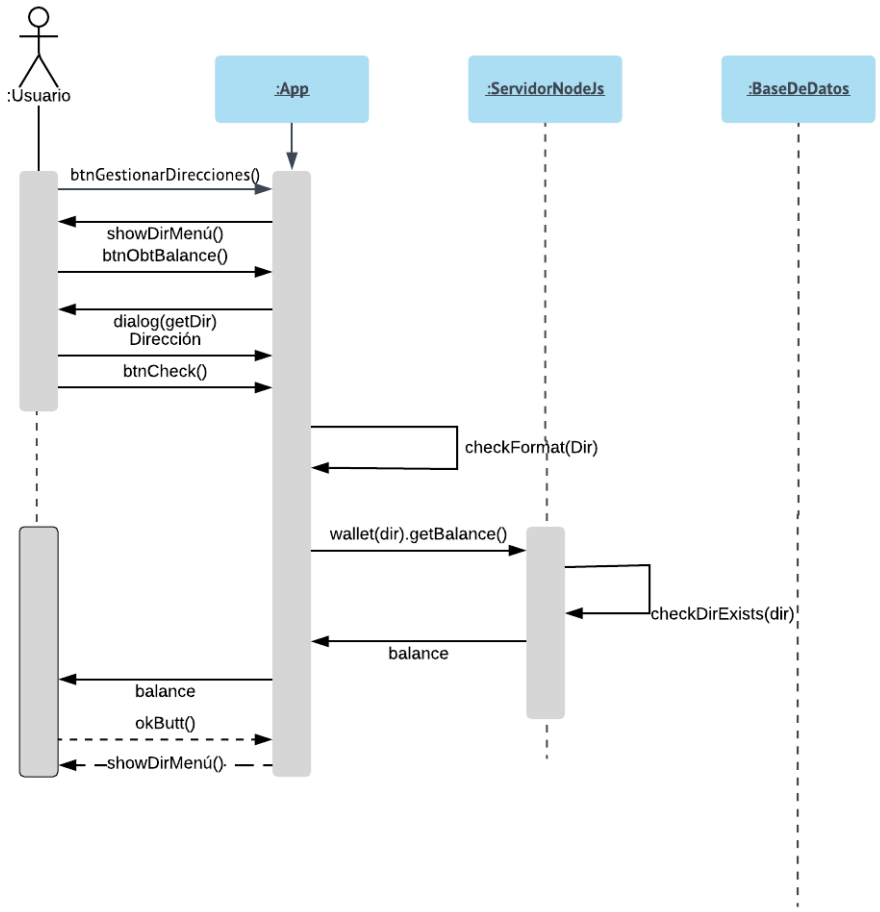


Figura 4.6: Figura: Diagrama de secuencia: Adress balance

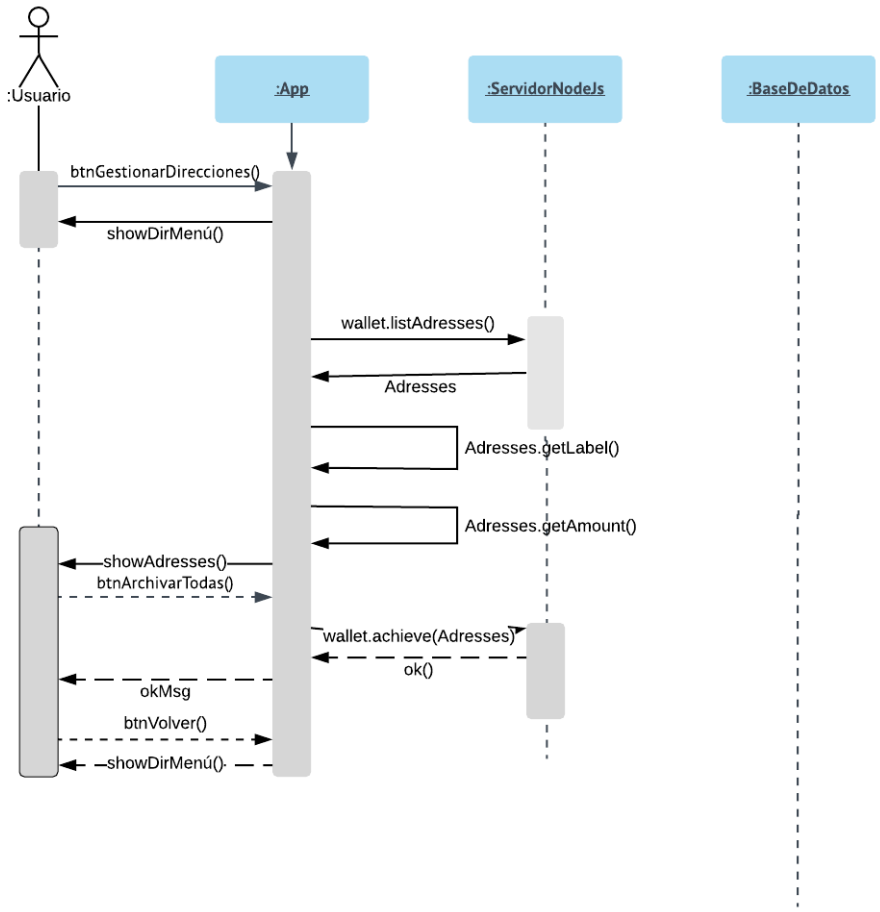


Figura 4.7: Figura: Diagrama de secuencia: Listar direcciones

Por último, en este caso vemos lo relativo a la transferencia de emolumentos, tanto individual como múltiple. Es el caso más importante porque añade comprobación de que la cantidad a transferir esté disponible.

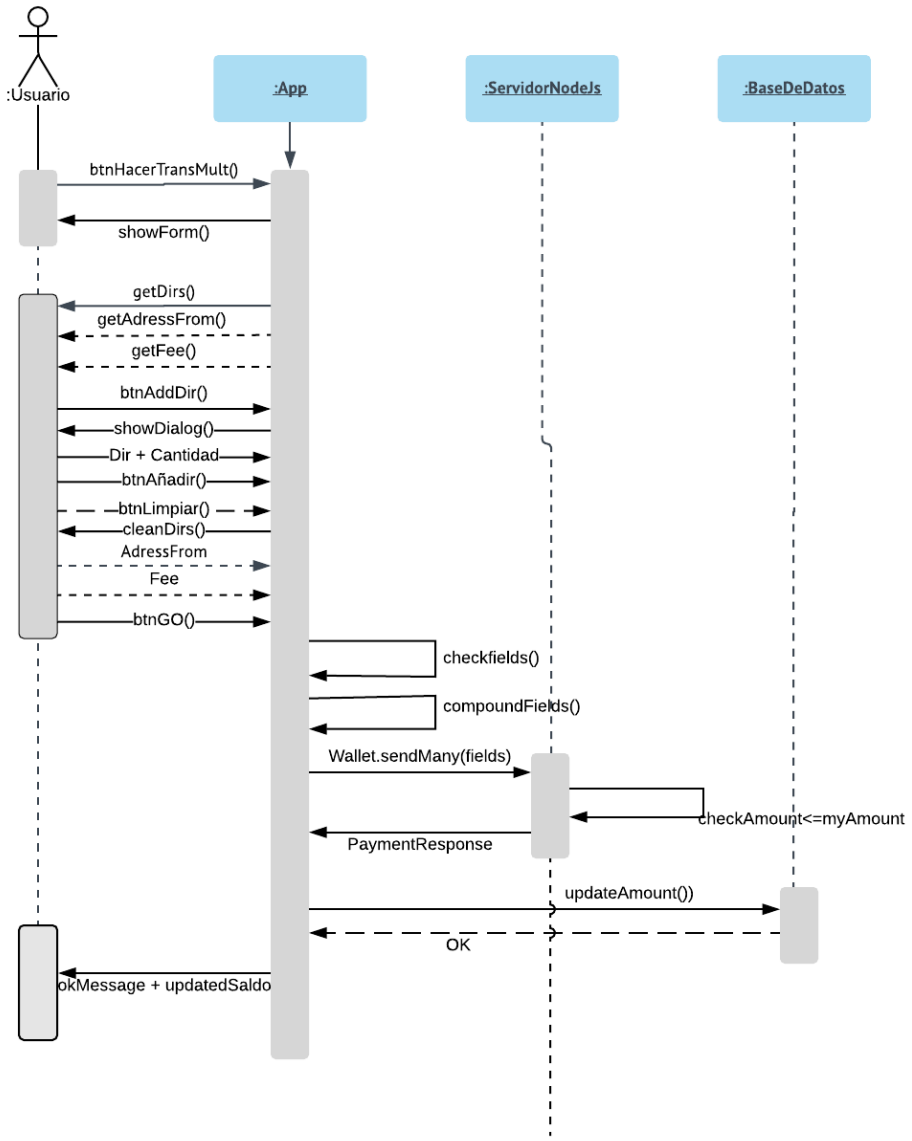


Figura 4.8: Figura: Diagrama de secuencia: Múltiples transferencia

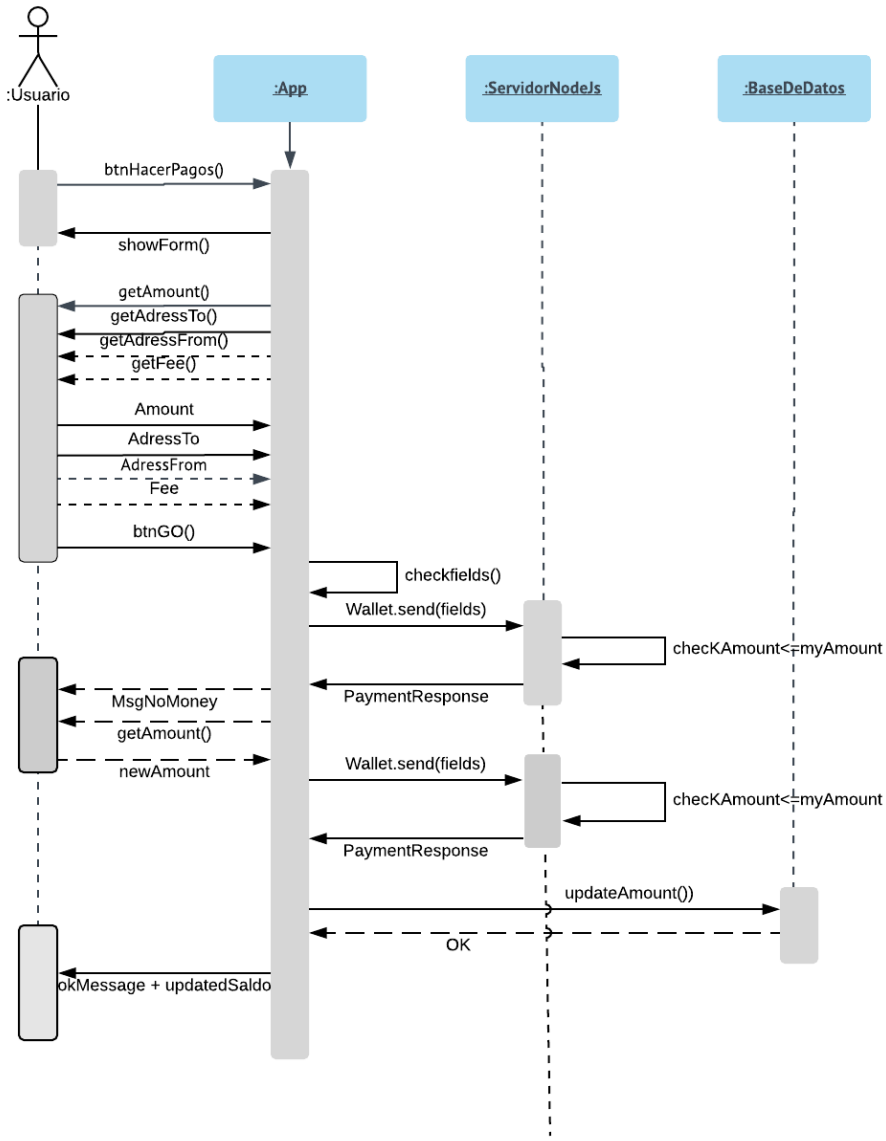


Figura 4.9: Figura: Diagrama de secuencia: Múltiples transferencia

Capítulo 5

Implementación

En este capítulo se tratará de explicar los pasos seguidos, así como las decisiones tomadas en la implementación de los diseños explicados en el capítulo previo. Mediante un recorrido por las capas que componen la arquitectura de la aplicación se mencionaron, con distinto nivel de detalle, los componentes de las clases y las relaciones entre ellas, la estructura definitiva de la base de datos, y los diseños finales de las interfaces, describiendo la funcionalidad de sus componentes.

El esquema básico del proyecto es una comunicación entre un cliente y un servidor. Este servidor es el que tiene realmente acceso a la base de datos y para llevar a cabo la comunicación del cliente es indispensable una conexión a internet, a no ser que nos limitemos a un ámbito local, como es el caso.

5.1. Capa de datos

El patrón de diseño de la Base de Datos utilizada, SQLite, es el patrón estándar que se intuye para entidades de tipo genérico. A grandes rasgos, este contendrá una tabla principal en la que se almacenará los identificadores de las cuentas y su password asociado, posibilitando así la funcionalidad de la multicartera especificada al principio.

El hecho de tener un servidor en NodeJs facilita mucho las operaciones para con la base de datos y la complejidad de esta, pues la mayoría de ellas las asume el servidor, quedando la base de datos como un mero mecanismo de autenticación, ya que las validaciones de las transferencias, así como el conjunto de operaciones es asumido por el servidor.

Adicionalmente la base de datos podría constituir un segundo mecanismo de autenticación añadiendo así una nueva contingencia, aunque en este caso no se hará dada la simplicidad de los procesos. Sí se recomendaría su uso en caso de una posible comercialización.

Las relaciones que se establecen se hacen asumiendo la simbiosis servidor-base de datos:

- Una o más Wallets pertenecen a una persona.
- Una o más direcciones pertenecen a una Wallet.
- Un Wallet, en sentido estricto, son un conjunto de transiciones etiquetadas.

Donde las tablas Wallet, GlobalWallet, Direcciones, quedarán reflejadas en la relación con la base de datos de la forma en que se aprecia en la siguiente figura:

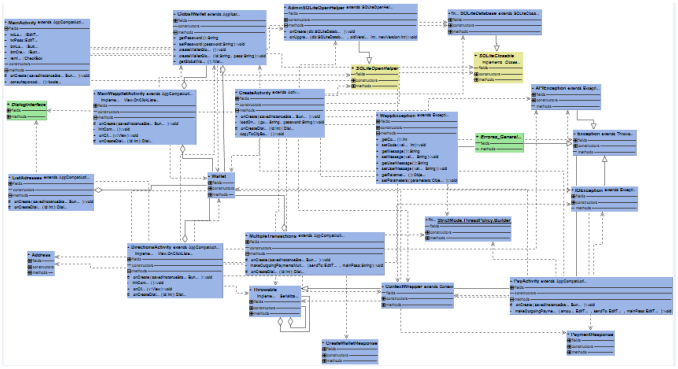


Figura 5.1: Figura: Relación entre clases

El acceso a la base de datos se ha implementado a través de la utilización de las librerías ya mencionadas en los capítulos previos y relativas a SQL Lite, en concreto se usará la clase *AdminSQLiteOpenHelper* que hereda de *SQLiteOpenHelper* e instancia una base de datos a medida.

Las funciones:

- execSQL(string query)
- query (string query)
- delete (string tabla, String PK)
- insert (string tabla, string [params])

- beginTransaction()
- compileStatement

desempeñan papel de "driver.[ej]ecutando en el modelo las consultas con los parámetros recibidos desde el controlador y devolviendo los registros en el objeto deseado. Se encargan además de establecer y cerrar la conexión con la base de datos, de lanzar las consultas y de controlar los errores y elevar y gestionar las excepciones que pueda arrojar la base de datos, descargando a la capa controlador de estas tareas. De esta forma, se podrán modificar los métodos de acceso a datos, o el control de errores, si hay un cambio en la tecnología de almacenamiento utilizada y sin repercusión en la lógica de negocio.

En el diagrama se aprecia cómo se relacionan las clases de la aplicación con la base de datos, con los objetos nativos y con los objetos y clases de la API, que a su vez son los que interaccionan con el servidor. En las siguientes secciones se pasará a explicar los elementos más relevantes de la figura.

5.2. Capa controlador

En el capítulo anterior se han descrito los varios bloques funcionales que contienen la lógica necesaria para el conjunto de operaciones de nuestra aplicación. A continuación, se mostrará el conjunto final de clases y sus componentes con los que se implementa la lógica de negocio de la aplicación.

Principalmente se ha definido una clase global (empleadas para almacenar constantes y variables accesibles al resto de clases), y un subconjunto de clases que implementan la capa del controlador. A continuación detallaremos los aspectos más relevantes de dichas clases.

La clase **MainActivity** se ha creado con el propósito de independizar de la aplicación la base de datos y los campos de sus tablas. Gracias al uso de esta clase se puede gestionar la base de datos sin necesidad de cambiar el código en las clases de la capa de negocio, y sus funciones, bastando modificar las constantes de esta clase.

Para facilitar la búsqueda e identificación, se ha empleado la siguiente nomenclatura en la definición de las variables: Nombres de botones: *btnbuttNombreAcción*. Nombres de campos: *textNombreSimbólico*.

La eficacia del uso de esta clase se ha demostrado en la etapa de desarrollo del código, durante la cual ha sido necesario realizar diversas modificaciones en

los nombres de variables. Gracias al uso de las constantes declaradas en *Strings*, se pudieron hacer esos cambios sin tener que revisar las funciones del resto de clases.

Por su parte, la clase interfaz *Errores_Generales* sirve para centralizar los mensajes de error, permitiendo además su clasificación y especialización y proporciona el soporte para el conjunto de identificadores únicos (wallet_id, dirección..) necesarios durante la operación del sistema. A su vez esta se complementa con, la clase *WappException*, que constituye una clase para construir nuestras propias excepciones, que hereda de exception y que será la que se utilice en los errores en tiempo de ejecución.

5.3. Clases especiales

Para el caso, se va a hacer una pequeña descripción de cómo se ha implementado la APP, haciendo especial hincapié en las clases cuya complejidad es mayor y se han implementado con alguna especialidad respecto a las demás.

5.3.1. Clases de inicio e interacción con la base de datos

Basándonos en lo expuesto anteriormente, partimos de la base de datos como plataforma principal en la que se sustenta la aplicación. La clase **MainActivity**, como ya hemos indicado, se acopla con la base de datos de tal manera que independiza el resto de la aplicación de la base de datos. En este caso, una vez cruzada la autenticación de este login-bdd, tendremos acceso a los datos asociados al Wallet con el guid que se ha autenticado.

Para este primer caso, la clase Main creará un Intent que propagara una vez pasada las validaciones pertinentes. Para dichas validaciones usará un objeto Toast que informe al usuario de si está incumpliendo alguna validación de formato o longitud para hacer de filtro y evitarle trabajo innecesario al servidor y a la base de datos. La validación principal se realizará en base de datos por medio de una función que se apoye en las clases *AdminSQLiteOpenHelper* y *SQLiteDatabase*, para así cruzar el password introducido con el que hay en base de datos a través de una query simple (método *SQLiteDatabase.rawQuery*).

Con las validaciones pasadas, esta clase generará una actividad, cuyo contenido será el Intent anteriormente creado, a la que se le dará inicio para continuar

con el flujo normal de funcionamiento de la aplicación. Se implementará un Listener(*setOnClickListener*) en el botón de login para dar inicio a esta secuencia.

En contraposición, si por el contrario se pretende crear un nuevo wallet, se entraría en la opción crear (Botón crear, listener en botón y nueva actividad), que nos llevará a una nueva pantalla donde se implementarán métodos para al obtención de datos y generación de un nuevo Wallet. Lo más reseñable en este caso en particular será que habrá que imponer una política de hilos permisiva para la correcta concurrencia de las hebras. Para ello usaremos la más permisiva: ***StrictMode.ThreadPolicy.Builder().permitAll()***.

En este supuesto habrá que usar la API explicada en los anteriores capítulos. En este caso en concreto, usaremos la clase *CreateWalletResponse*, la que nos proporcionará una cartera a partir de la dirección del servidor (montado en local, en este caso), una contraseña válida, un código de API y adicionalmente y de manera no obligatoria una segunda contraseña privada para la doble encriptación y un email.

Si el servidor acepta todos los datos obtenidos, se insertará en la base de datos los nuevos datos relativos a la nueva cartera, consiguiendo así el propósito de ser multicartera. Con el intent creado y la cartera incorporada a la base de datos, se abrirá un cuadro de diálogo informando sobre guid generado. Mediante un método de copiado, al aceptar dejaremos copiado en el portapapeles dicho guid, pues se entiende la tediosidad de tener que copiar una cifra de 16 dígitos a mano.

Estas dos clases aquí explicadas comparten la generalidad de ser las dos únicas que interaccionan con la base de datos, como se veía en los diagramas de secuencia. A partir de aquí la aplicación en conjunto con el servidor será la que se encargue de la gestión de todas las operaciones.

y a partir de los datos que se solicitan se crearía una nueva cartera y se proporcionará el nuevo guid para acceder al mismo menú visto anteriormente.

5.3.2. Clase principal y clase global

En cualquier de los escenarios descritos en la sección anterior, el paso inmediato es el que aquí se explica y el que implementa la clase *MainWappletActivity*. Más allá de los Intents, las Activity y los Toast comunes a todas las clases, esta clase es especial por ser la que instancia el objeto GlobalWallet, que contiene una copia de los datos que contiene el servidor y que será el eje común sobre el que se opere hasta que finalice la sesión o se entre con otro Wallet.

Para dicha instanciación, habrá que usar el método nativo de Android *getApplication()* apuntando a la clase *GlobalWallet.java*. Esta clase que hereda de Application, contendrá los mismos datos que almacena el servidor sobre un Wallet; apiCode, idGuid, password... Esta clase principalmente contiene un objeto **Wallet** (heredado de la API) y un método que se trae los datos del servidor e instancia dicha clase Wallet. Adicionalmente contendrá un método para actualizar la contraseña.

Esta clase además contendrá un novedoso mecanismo de la gestión de las operaciones. Y es que se ha desarrollado un control basado en un *onClick* sobre el layout previamente establecido. Este listener genérico ahorra muchos recursos pues solo genera un intent que será identificado mediante un switch para saber qué actividad se inicia. Atendiendo a esto se enviará a la pantalla deseada con toda la información del intent.

En la opción de consultar saldo no sería necesario iniciar ninguna actividad, así que esta parte se optimiza sabiendo que ya tenemos el Wallet instanciado y accesible, se usa un cuadro de diálogo para informar y nuevamente nos apoyamos en métodos de la API que apoya en el servidor para obtener el balance del conjunto de todas las direcciones de dicho Wallet.

Así mismo es remarcable que el GlobalWallet sólo se instancia aquí, en el resto de clases y pantalla se accede a la instancia aquí realizada, ayudando así en la eficiencia y cubriendo posibles dobles gastos.

5.3.3. Clases de gestión de direcciones

La gestión de las direcciones pertenecientes a un Wallet tiene una complejidad elevada. Nuevamente será necesario establecer una política de gestión de hilos permisiva para evitar fallos de concurrencia en tiempo de ejecución. Optamos por la opción más permisiva otra vez:

- ***StrictMode.ThreadPolicy.Builder().permitAll().build();***

En general las funcionalidades implementadas en esta capa no distan mucho del resto de la aplicación, así que entraremos en la particularidad directamente. Se da un caso especial causante de esta: a la hora de listar las direcciones, en tiempo de desarrollo no sabemos cuántas direcciones vamos a tener que listar. Para solventar este problema de la manera más eficiente, se opta por añadir un

segundo layout al principal. Y dentro de este, tantos como direcciones activas haya.

Desde el punto de vista técnico esto ha sido posible mediante la creación de una actividad con su respectivo intent que nos lleve a una nueva pantalla que contenga los respectivos marcos de diseño:

1. El layout principal será un coordinador de tipo *ConstraintLayout*. Este contendrá al segundo, así como a los botones, etiquetas y demás campos necesarios en la pantalla.

2. El segundo será el que contenga dentro una "toolbar". Dicha barra será la plataforma en la que se incorporen el número indeterminado de layouts (el tercero).

3. El tercero estará implementado a parte, contendrá los datos comunes a las cuentas y será incorporado desde el código de manera dinámica y no estática como el resto de la aplicación.

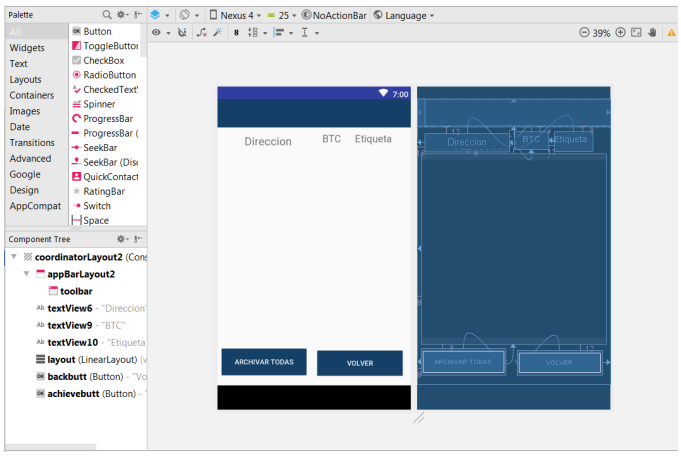


Figura 5.2: Figura: Relación de Layouts para pantalla de listado

Para cohesionar este framework será necesario establecer la misma política de hilos anteriormente mencionada, declarar un layout genérico y un toolbar que identifique el arriba nombrado y establecerlo como ActionBar (*setSupportActionBar(toolbar)*). Después, tan solo se tratará de recuperar las direcciones activas a partir del objeto global y la simbiosis API-servidor. Para cada dirección recuperada se creará una layout de tipo 3 y se le infla al layout plataforma, el de tipo 2,

con los datos recuperados del servidor relativos a dicha dirección; id_dirección, etiqueta y balance.

Adicionalmente se podrán archivar todas las direcciones a través del botón que implementa el listener correspondiente. La App informará de si esta operación ha ido correctamente. Para la concurrencia adecuada y el rendimiento óptimo cada proceso conlleva su propia hebra.

5.4. Interfaz gráfica

Hasta ahora se ha hecho un recorrido por el funcionamiento de la aplicación Android pasando por encima de la interfaz gráfica. En la presente sección vamos a entrar en el diseño de la parte visual de la aplicación.

El lenguaje de diseño de una aplicación Android es el *XML*, ubicándose todos los archivos necesarios para el diseño en la carpeta *res* del proyecto. Dentro de esta carpeta se ubican los siguientes directorios:

- \item .\res\drawable\: En esta ruta se irán guardando de manera local las imágenes utilizables en la aplicación.
- \item .\res \menú \: Directorio para salvar la estructura de la actionBar de cada Activity que implemente la clase *AppCompatActivity*.
- \item .\res \values\: Almacena un conjunto de archivos .xml fundamentales para la aplicación; colors.xml: en la cual se configuran los diversos colores usados en la aplicación; strings.xml: encargada de establecer las cadenas de textos constantes accesibles en cualquier punto de la aplicación; *styles.xml*: cuya misión es definir y configurar un tema o estilo determinado para la aplicación...

  \item .\res\layout\: Para el caso de nuestra aplicación, contendrá un total de 13 archivos XML, 9 referentes a las activities de la aplicación y el resto para los adaptadores personalizados que se han creado y para la pantalla de inicio *(splash)*.

  Para todos los elementos que forman un XML es recomendable especificar un conjunto de propiedades tales como el color, distancia, tamaño, ubicación, propiedades *onClick()*... En adición, cada elemento posee un identificador único para poder referenciar en caso de que se necesite operar con él en tiempo de ejecución. se asocia un objeto.

Esto funciona de tal manera que se asocia un objeto equivalente al elemento del layout -del mismo tipo- a través de dicho identificador, se opera y modifica esta copia heredada, y luego automáticamente se propaga al objeto padre.

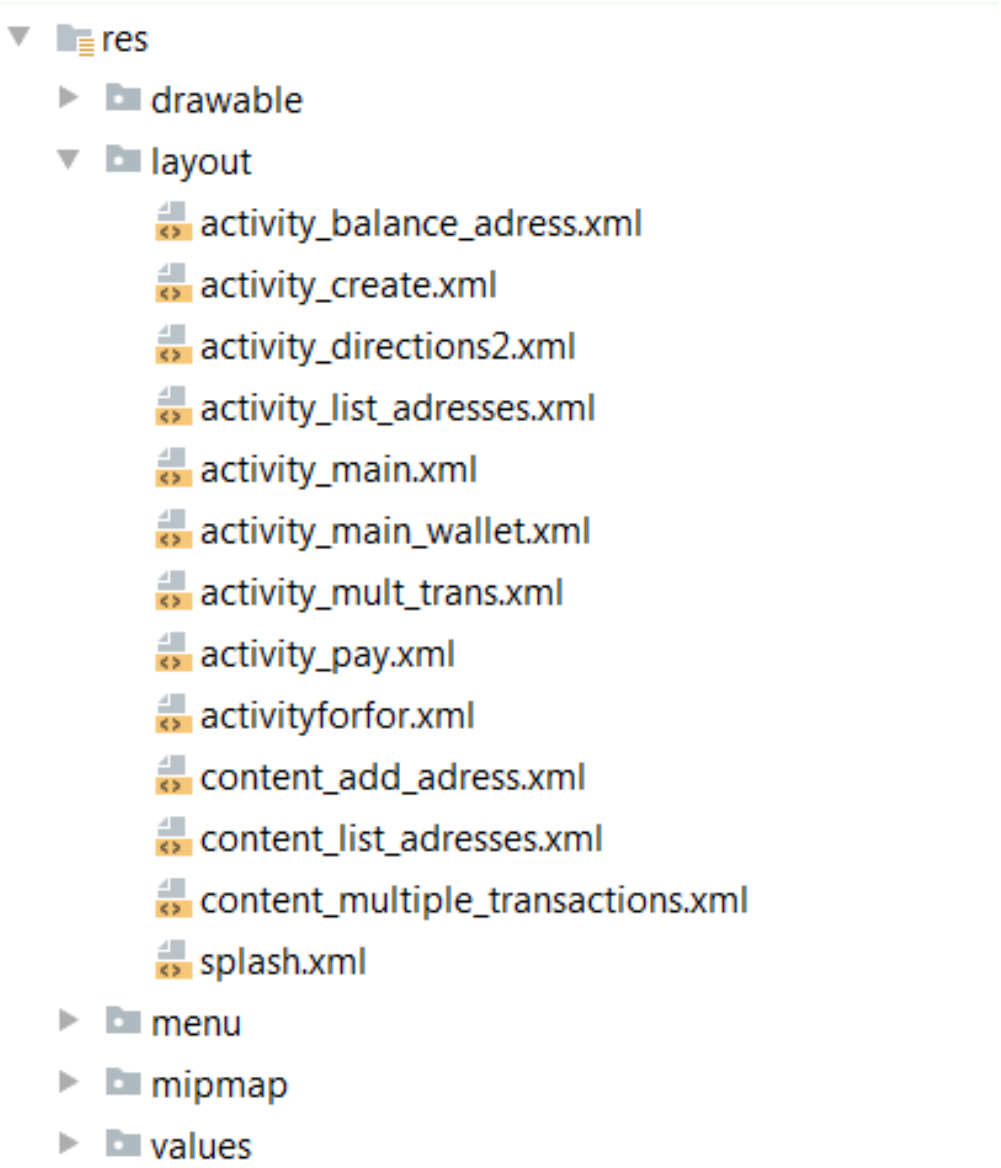


Figura 5.3: Figura: Conjunto de layouts usados en la APP

- \item .\res\menú\: Esta ruta contiene el archivo navigation.xml. Este archivo es el encargado de configurar la actionBar de la actividad.

Entrando en la parte del código se observa que hay tres elementos en el menú (perteneciente al actionBar) con sus respectivas propiedades. Entre dichas pro-

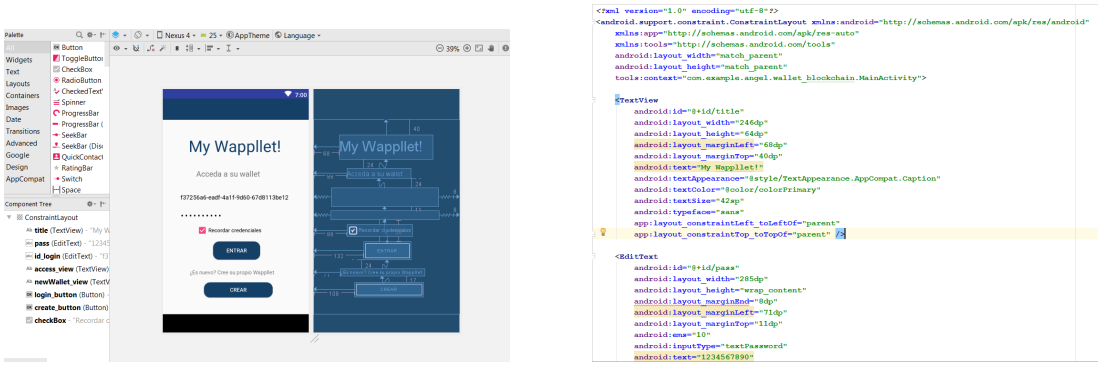
piedades están el identificador, empleado para la identificación al elemento del menú; icon, la cual referencia un archivo del directorio drawable; *showAsAction*, la cual posee una propiedad llamada ifRoom, que evidencia que el elemento será incluido en el actionBar si tuviese cabida; title, que permite mostrar el elemento en lugar del icono; y, finalmente, *layoutwidth y layoutheight*, que parametrizan la anchura-altura del elemento en cuestión, estando definidos como *wrap_content*, que indica que la vista des solo extensible si se va a proceder a extender lo necesario para que se muestre el elemento.

En este caso, el orden definido para los elementos es especialmente relevante, dado que al tener cada uno de los elementos la propiedad showAsAction configurada como ifRoom, si no hubiera suficiente hueco para mostrar todos los elementos, se se seguiría el orden preestablecido, quedándose en este caso la tercera opción como la candidata a no aparecer en la actionBar. En caso de no aparecer en la actionBar, pasaría a aparecer pulsando el botón de menú en el dispositivo Android.

Como ya se ha explicado y se verá en la aplicación, en el directorio:

- .\res\layout\

se encuentran los archivos XML de las activities. Android carga el layout relativo a un activity mediante el método *setContentView(R.layout.Nombre_de_Activity)*, que normalmente se carga al inicio del método onCreate() de cada activity. Como ejemplo inicial y visual se coge el layout activity_main.xml. Se puede observar que hay un *ContraintLayout* con 8 elementos divididos entre campos de entrada y salida:



(a) Aspecto visual

(b) Código

Figura 5.4: Figura: Activity_main.xml

Capítulo 6

Conclusiones y líneas futuras

6.1. Conclusiones

Entramos en este trabajo fin de grado con el interés en criptomonedas, y las posibilidades futuras de estas, apenas para escribir una cartera de la criptomoneda actualmente más popular.

Hemos desarrollado, de hecho, una aplicación para Android que implementa una cartera de la criptomoneda bitcoin, con la funcionalidad que suelen soportar estas.

En el camino, hemos descubierto que la tecnología de blockchain es mucho más que un soporte de criptomonedas; y que abre inmensas posibilidades para gestionar de forma distribuida y concurrente un almacén de información fiable; que actualmente se utiliza para transacciones económicas, pero que con el tiempo tendrá más usos.

6.2. Líneas futuras

Dentro de las líneas futuras, se abren dos grandes ramas; las derivadas de la blockchain, y las derivadas de la cartera desarrollada en sí.

Respecto a la cartera en sí, sería interesante hacerla multimoneda, para que soportará varias criptodivisas.

Respecto a la blockchain, se abre un mundo de posibilidades; desde almacén de históricos en blockchain privadas de organizaciones, para evitar la manipulación de estos, como uso de la blockchain como notario público digital. Es una

tecnología emergente, y hay muchos usos que se le pueden dar más allá del actual como soporte de criptomonedas.

Capítulo 7

Bibliografía

En la presente sección de incluyen los documentos y páginas que han sido consultados para la realización de este documento:

1. http://www.infotechnology.com/online/Que-es-blockchain-la-tecnologia-que-viene-a-revolucionar-las-finanzas-20160810-0001.html

2. http://www.expansion.com/economia-digital/innovacion/2015/12/18/56704eb5e2704

3. Bitcoin and Cryptocurrency Technologies. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder

4. http://www.fundacionctic.org/ctic/articulos-y-otras-publicaciones/que-es-el-blockchain-del-que-todo-el-mundo-habla?gclid=COyeovi47NICFRBmGwod7hkGuA

5. Blockchain Mining Games. Aggelos Kiayias, Elias Koutsoupias, Maria Kyropoulou, Yiannus Tselekounis.

6. "Blockchain, ¿El nuevo notario?"NTT Data Company. Javier Ibáñez Jiménez

7. https://www.xataka.com/empresas-y-economia/para-que-se-esta-usando-blockchain-mas-alla-de-bitcoin Usos de blockchain -.-

8. https://criptonoticias.com/informacion/que-es-una-cadena-de-bloques-blockchain

9. Análisis sobre el origen, comportamiento y crecimiento del mercado del Bitcoin. Victor Manuel Gonzalez Soltero

10. http://blog.bit2me.com/es/que-es-cadena-de-bloques-blockchain/

11. https://blockchain.info/es/wallet/

12. Bitcoin: Bases, comportamiento como moneda e inversión. Guillermo Zaera Vidal

13. https://criptonoticias.com/educacion/universidad-princeton-publica-primer-borrador-gratuito-libro-bitcoin/#axzz4c924pMLU

14. The Blockchain Anomaly. Christopher Natoli, Vincent Gramoli.

15. http://www.isoin.es/blog/

16. La nueva moneda virtual que está revolucionando el mundo de las divisas digitales. Yasmina Asensio Grau

17. http://labaranda.yuku.com/topic/52865/Bitcoin?page=59#.WNVC6Ts1-00

18. Criptodivisas: Del BTC al MUFG. El potencial de la tecnología Blockchain. Mª Nieves Pacheco Jiménez.

19. http://bitcoinbcn.blogspot.com.es

20. Bitcoin y el cambio de divisa. Alfredo Iglesias Colino

21. http://elbitcoin.org/las-tres-mejores-carteras-billeteras-bitcoin/

22. Music On The Blockchain. Blockchain For Creative Industries Research Cluster. Middlesex University

23. http://www.diariobitcoin.com/index.php/2016/06/15/los-5-mejores-monederos-bitcoin-para-usar-en-el-2016/

24. https://elandroidelibre.elespanol.com/2015/12/eclipse-vs-android-studio.html

25. Documentos electrónicos para el intercambio de bienes y servicios. Luis Antonio García Alejo

26. https://elandroidelibre.elespanol.com/2016/02/android-studio-2-beta.html

27. https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki

28. https://developer.android.com/training/basics/data-storage/databases.html?hl=es-419

29. http://www.tutorialesprogramacionya.com/javaya/androidya/androidstudioya/detal

30. https://blog.bit2me.com/es/carteras-moviles-bitcoins/

31. Análisis y comparación de monedas criptográficas basadas en la tecnología blockchain. Maria Fernanda Medina Reyes.

32. https://es.cointelegraph.com/news/6-myths-about-bitcoin-and-how-to-bust-them-expert-take

33. JOINCOIN: Un entorno de trabajo común para criptomonedas. Paulo N Carrillo Peña, Clara Inés Peña de Carrillo, Josep Lluis de la Rosa

Capítulo 8

Apéndice A: Manual de usuario

En la siguiente sección se incluirá un pequeño manual de usuarios que explique todas las funcionalidades ya tratadas en otras partes de la presente memoria pero a través de la aplicación generada a partir de todo ello. La aplicación en general tiene un diseño minimalista enfocado a resaltar al máximo las funcionalidades por encima de los gráficos, de tal modo que a un usuario poco familiarizado con el mundo de las criptomonedas le cueste relativamente poco entender el funcionamiento de la aplicación.

La aplicación comienza con una pantalla de bienvenida en el cual podemos ver el logo de la aplicación -desarrollado en el presente trabajo-, así como un pequeño eslogan. Esta pantalla inicial que supone un breve retraso en el inicio de la app, nos permite visualizar el logo a la par que secundariamente la aplicación inicia los procesos.

Figura 8.1: Figura: Login de la aplicación

Comenzaremos describiendo los dos posible caminos que puede tomar un usuario al acceder a la aplicación; o bien es un usuario existente, es decir, es usuario de blockchain, posee un wallet y por tanto tiene el identificador de este, con lo cual se loguean tal y como vemos en la siguiente imagen:

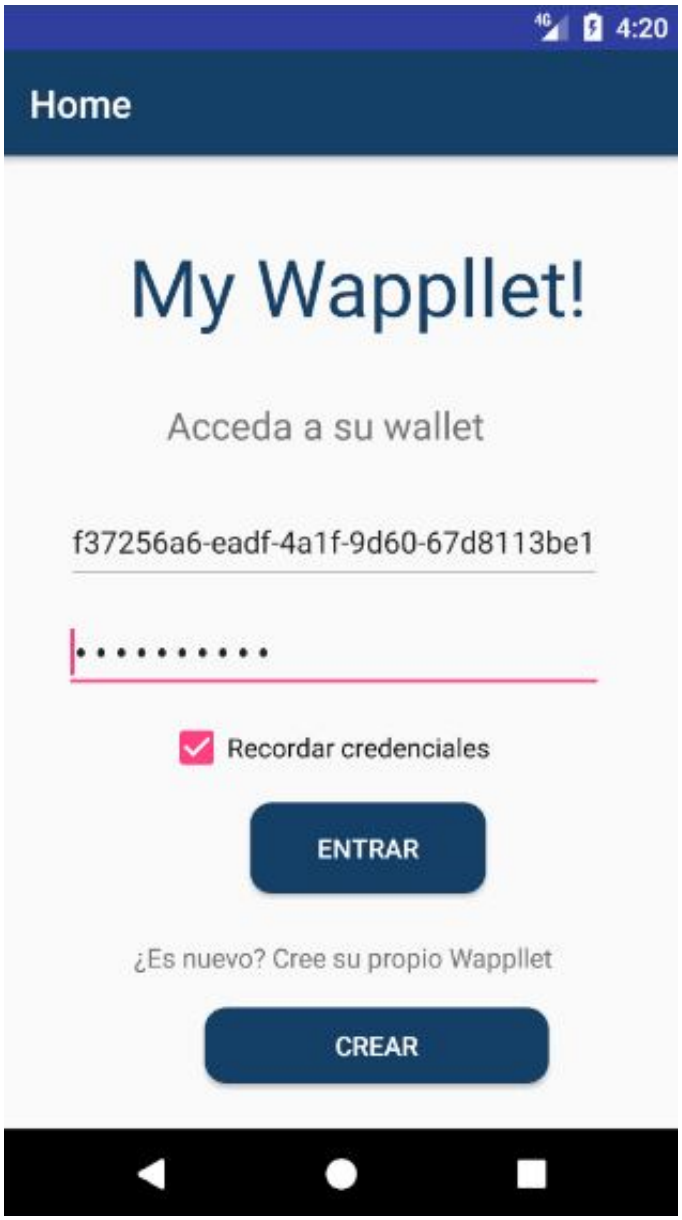
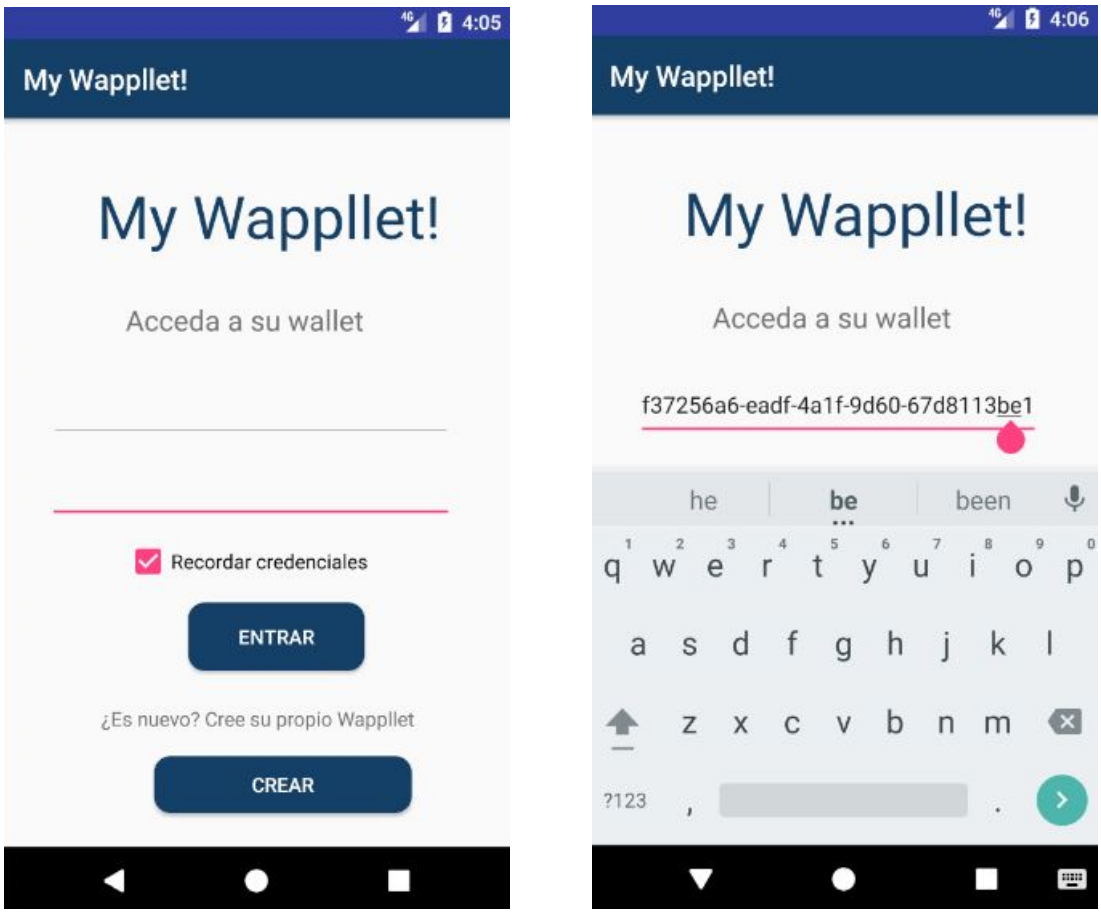


Figura 8.2: Figura: Login de la aplicación

En el caso anterior partimos del escenario en que un usuario ya ha accedido con anterioridad y ha recordado sus datos, con lo cual tanto la dirección como la contraseña aparecen rellenas al acceder a la aplicación. El caso complementario sería el que un usuario accede por primera vez a la aplicación pero no a la red blockchain, por lo que tendría que rellenar los datos por primera vez.

(Nótese que en este caso y dado que "jugamosçon una aplicación a pequeña escala, la única manera de registrarse es crear una cuenta mediante el enlace crear. Con la aplicación totalmente operativa y no corriendo sobre un servidor local, podrías registrate en

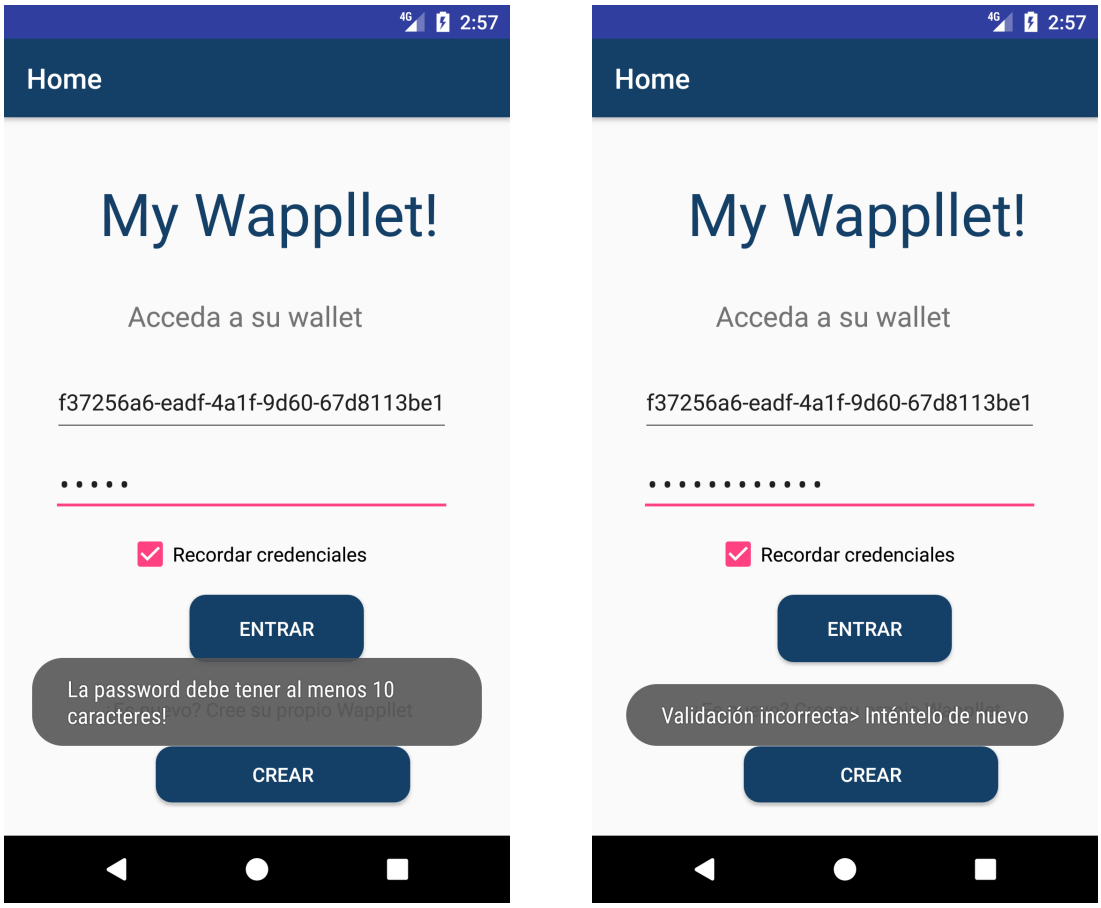
portales como "blokchain.infoz luego acceder a tu billetera mediante la aplicación.)



(a) Campos vacíos (b) Campos rellenándose

Figura 8.3: Figura: Login por primera vez

En el caso anterior (o incluso en el primero, si se hubiese desalineado la base de datos), podría dar un error al entrar a la aplicación; el usuario no existe o bien la contraseña es incorrecta. En ese caso la aplicación nos informará de dicha anomalía. Por otra parte, también nos indicaría si el error se hallase en la longitud mínima de la contraseña.



(a) Longitud de contraseña incorrecta

(b) Usuario no existe o contraseña no coincide

Figura 8.4: Figura: Error al loguearse

Pasados todos estos supuesto, el flujo normal comprende la validación exitosa ya sea por primera vez o por enésima vez tras recordar las credenciales. En ese caso ya se accedería al menú principal de la aplicación en el cuál nos encontraremos con todas las opciones expuestas a lo largo del capítulo anterior:

Figura 8.5: Figura: Login de la aplicación

Sin embargo este no es el único camino de llegar a este punto, a este menú; la otra opción es la creación de un wallet propiamente dicha. Para ello, como podemos observar en las anteriores capturas, tenemos un botón en la pantalla principal para crearnos un monedero de blockchain. Si pulsamos en dicho botón, nos llevará a la siguiente pantalla:

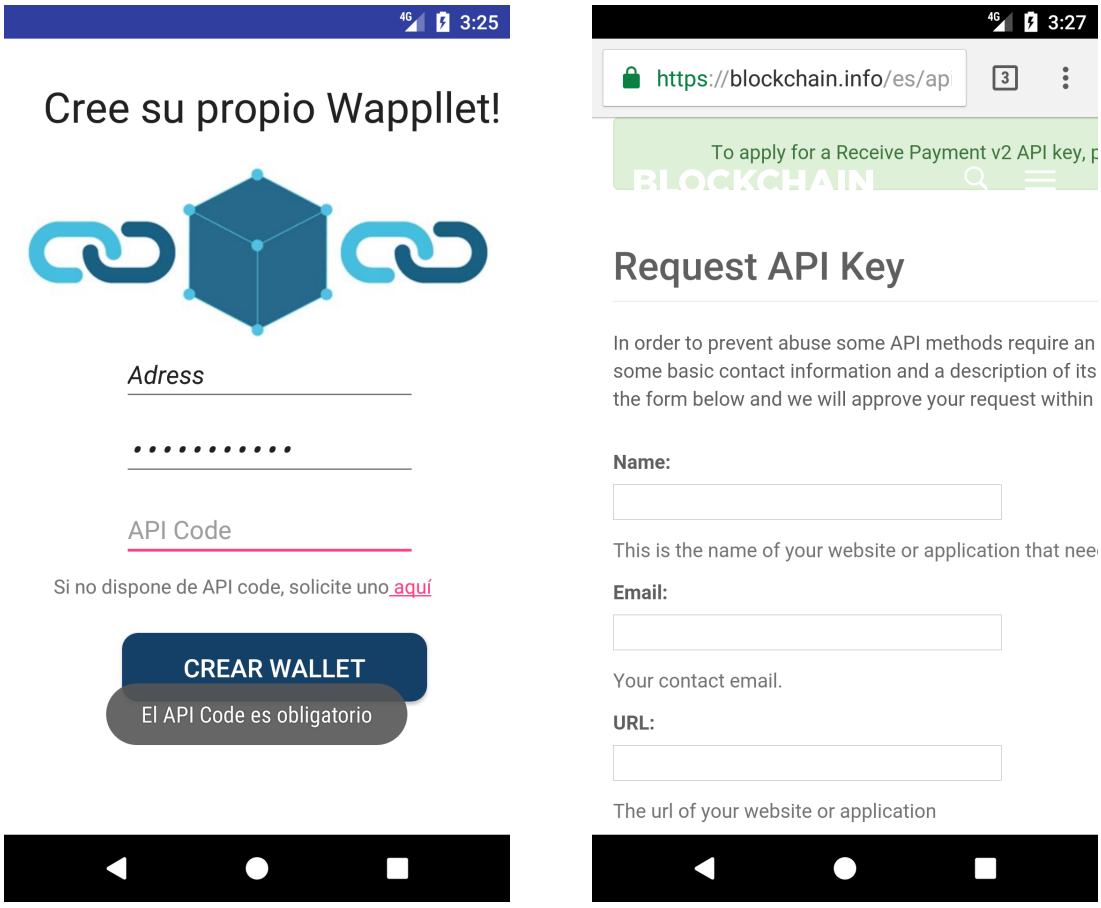
Figura 8.6: Figura: Login de la aplicación

Ahora nos encontramos ante un menú que nos solicita 3 datos para poder crearnos un wallet; una etiqueta, una contraseña y una código API. Aunque se podrían añadir otros muchos patrones adicionales de autenticación, por limpieza de la interfaz y por tratarse de un modelo sólo usable a nivel local, se ha prescindido de parámetros adicionales.

La etiqueta será usada para nombrar a la dirección principal que se crea al crear un wallet. No es obligatoria puesto que si no se pone se generará una automáticamente para asociar a dicha dirección. Todas las direcciones generadas

llevan una etiqueta asociada, se informe explícitamente o no. Por otro lado tenemos un campo contraseña, cuyo único requisito es que su longitud sea mayor de 8 caracteres. En este caso se permiten que estos caracteres sean sólo numéricos. Por último, necesitamos un código de API. Como ya hemos explicado este código debe disponer de permisos para crear un wallet. Para ello se facilita un enlace a la página oficial de blockchain en la cual se pueden solicitar dichos códigos.

Esta página validará que los campos obligatorios se rellenen y además de que sea una API con permisos de crear wallet y una contraseña correcta.

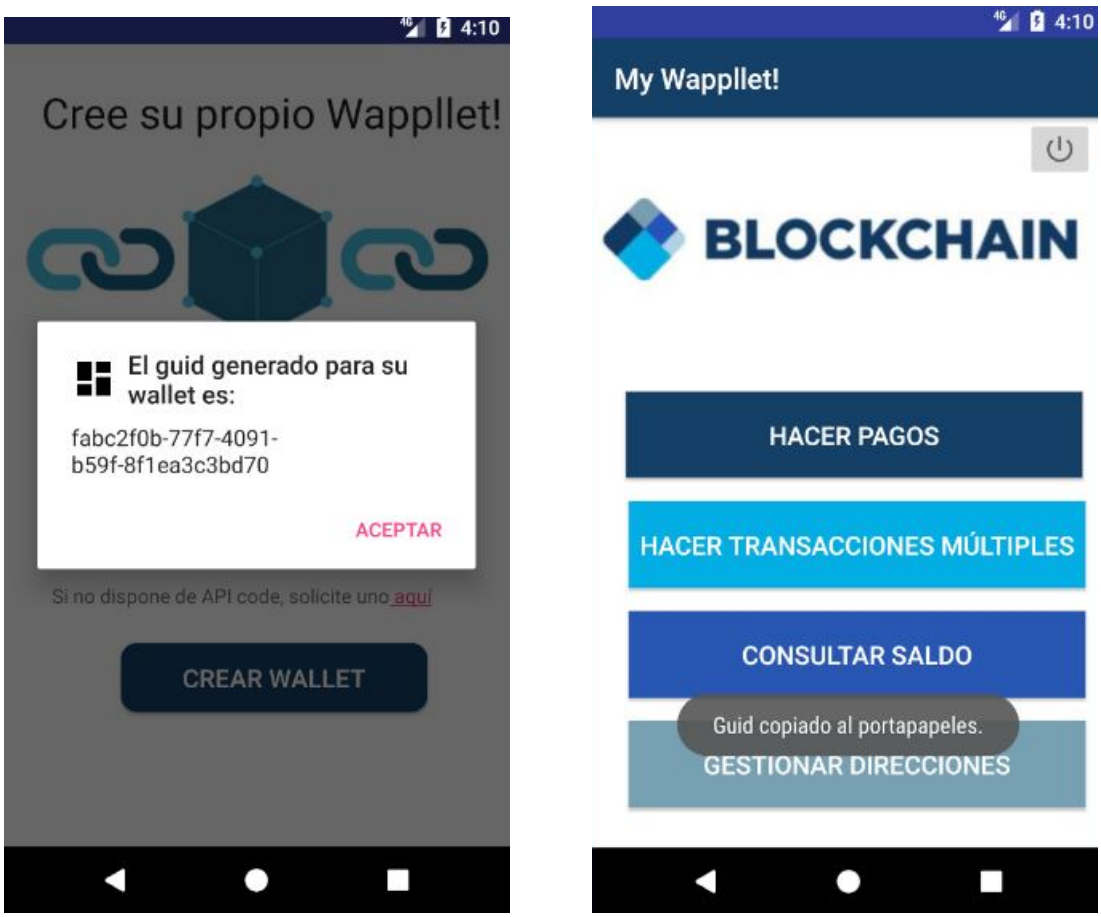


(a) API code obligatorio (b) Obtener API Code

Figura 8.7: Figura: Crear Wallet

Una vez obtenido un código API, podremos crear nuestro Wallet dándole al botón crear. Si todo va bien, la aplicación informará del guid creado y asociado al nuevo wallet creado. Esto se hace mediante un cuadro de diálogo en modo .ªlert". Aunque no es seleccionable este parámetro en este modo de información,

la app gestiona esta disyuntiva copiando dicho guid automáticamente al portapapeles en cuanto se pulse el botón aceptar:



(a) Guid generado (b) Guid copiado al portapapeles

Figura 8.8: Figura: Wallet creado: la app informa el guid

Si todo ha ido bien, al consultar las direcciones veremos como hay una existente con 0 BTC y con una etiqueta asociada llamada .ªdress".

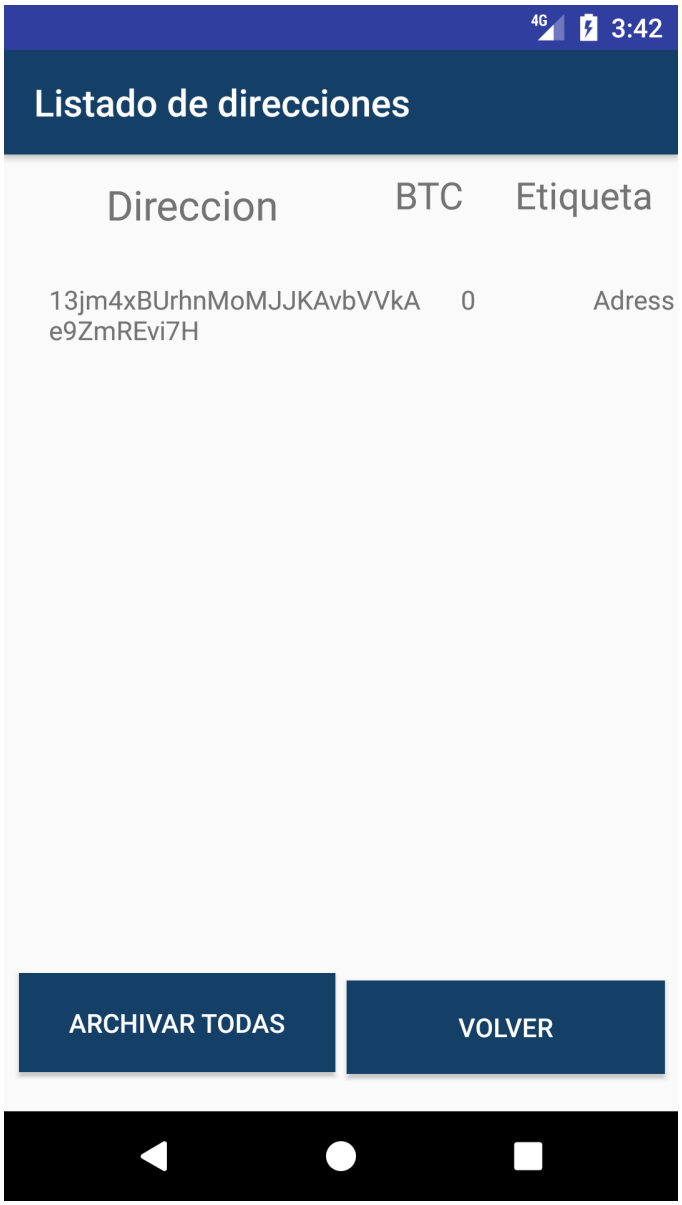


Figura 8.9: Figura: Login de la aplicación

Como podemos apreciar, al seguir el otro camino, el de crear el wallet, hemos llegado al mismo punto en el cual nos encontrábamos antes, es decir, al menú principal donde podemos hacer uso de todas las funcionalidades del wallet.

Las dos primeras opciones hacen referencia a la transacción de BTC. La primera se refiere a una transferencia simple, entre una dirección perteneciente al wallet en el que estamos metidos (podemos elegir cuál o dejárselo a la plataforma) y otra existente en la cadena de bloques. En rigor sólo es necesario informar de la cantidad a transferir y de la dirección de destino y que ésta sea correcta.

Se da por supuesto que la cantidad a transferir más la cuota de los mineros será menos o igual que la cantidad total de BTC de la que dispone el wallet.

Figura 8.10: Figura: Login de la aplicación

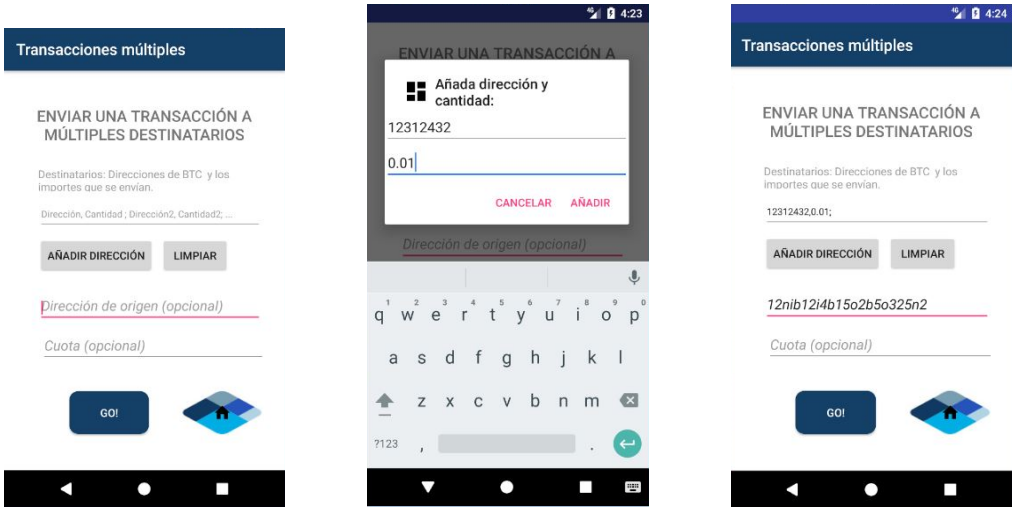
Dada que esta funcionalidad es relativamente simple, no se entrará en mucho más detalle en ella, pues es algo fácilmente comprensible incluso para un usuario inexperto en estas tecnologías.

La otra funcionalidad asociada a la transferencia si tiene un poco más de complejidad. Y es que esta abarca una de las mejores funcionalidades que implementa esta aplicación y que constituye una novedad respecto a las ya existentes en el mercado; la transferencia múltiple. Aunque se resuma en la misma

pantalla y en el mismo botón, el modo de utilizarlo constituye 2 funcionalidades distintas pero complementarias. Para empezar, puedes hacer 'N' transferencias distintas en la misma transacción, es decir, como pulsar 'N' veces el botón de hacer una transferencia simple. Sin embargo estas dos maneras de enviar dinero no funcionan igual internamente.

Este menú por tener un comportamiento distinto y precisar los datos con otro formato, incluye un botón para abrir un menú contextual más amigable en el que introducir los datos. Dichos datos serán visualizados en un formato de separación de cadenas mediantes ';', y será tratado según corresponda internamente y de manera transparente al usuario.

Pulsando el botón .ªñadir dirección.ªccederemos a él y pulsando el botón limpiar borraremos los datos introducidos hasta ese momento. El resto de campos funcionan de forma análoga al envío simple:



(a) Menú transacción múltiple

(b) Añadir dirección y cantidad

(c) Dirección añadida y formateada

Figura 8.11: Figura: Gestión de transacciones

Esta funcionalidad engloba la relativa a los pagos recurrentes desde una cartera: si añades la misma dirección 'N' veces, se ordenará un pago recurrente que se realiza con una cadencia predeterminada (en este caso un día), realizándose sólo el pago enésimo si el inmediatamente anterior cuenta con una transferencia aceptada como válida por el consenso de la red de Bitcoin.

Dejando atrás la dinámica de las transferencias, nos quedan dos botones del menú principal por explicar: el primero no tiene demasiado recorrido teórico.

Una consulta de saldo no es más que la suma de todas las cantidades que poseen las distintas direcciones pertenecientes al wallet logueado.

Figura 8.12: Figura: Comprobar saldo

Por manejabilidad se opta por reflejar todas las cantidades en satoshi (un satoshi se divide por 100000000 para obtener la cantidad en BTC).

Para terminar, nos queda el apartado de la gestión de direcciones, pues como hemos dicho se trata de un monedero multidireccional. El botón gestionar direcciones nos llevará a un nuevo menú enfocado exclusivamente al tratamiento y

creación de las direcciones del Wallet. Podemos verlo en la siguiente pantalla:

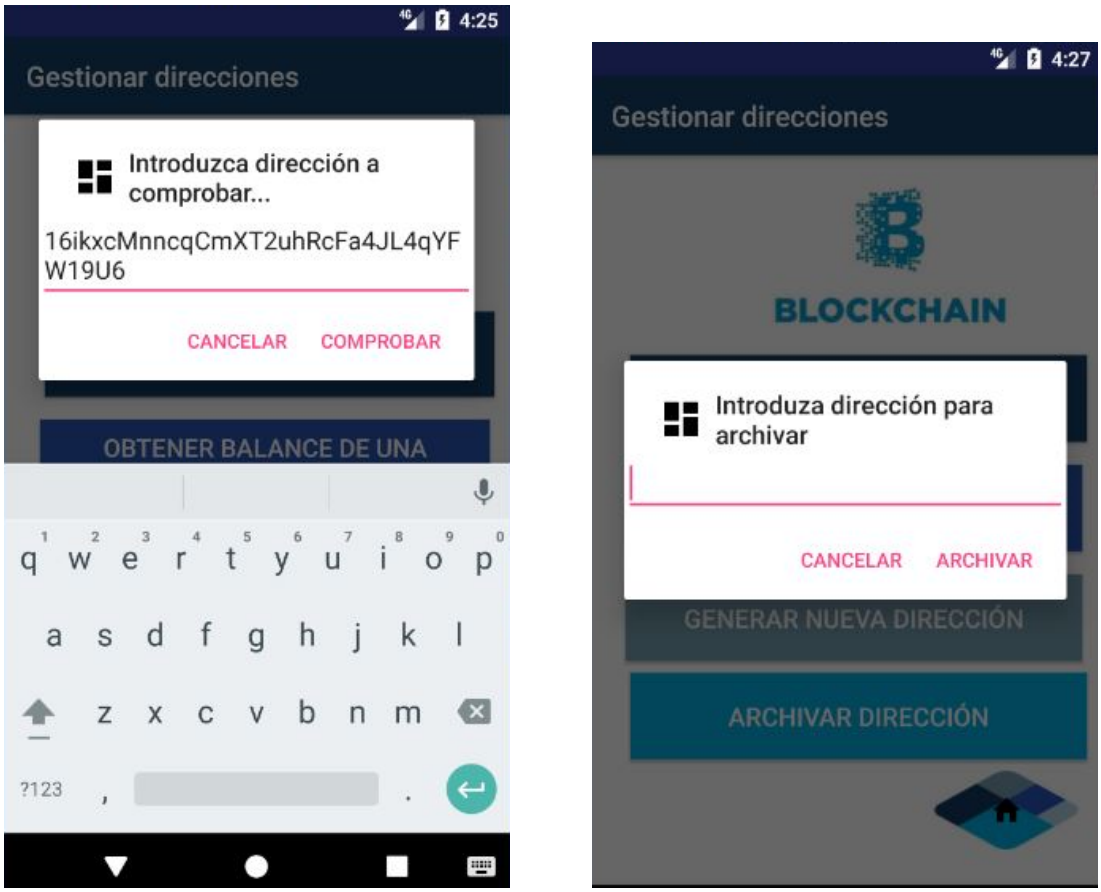
Figura 8.13: Figura: Login de la aplicación

Observamos como la aplicación nos ofrece 4 alternativas además del botón *home*. En orden de arriba abajo, apreciamos que podemos listar las direcciones, obtener el balance de una dirección, generar una nueva dirección o archivar una ya existente. Por ser la funcionalidad de mayor calibre técnico, la opción de "Listar direcciones.ªbarca la funcionalidad de visualizar las direcciones en formato [dirección; amount; label], visualizar la cantidad de todas ellas (explícitamente en el formato recién explicado), y de archivar todas las direcciones mediante un

botón adicional añadido para tal fin. Esta pantalla ya se vio en el primer ejemplo en el que creamos un wallet con la dirección principal, no obstante, vemos como funciona de manera idéntica cuando el número de direcciones es mayor:

Figura 8.14: Figura: Gestión de direcciones

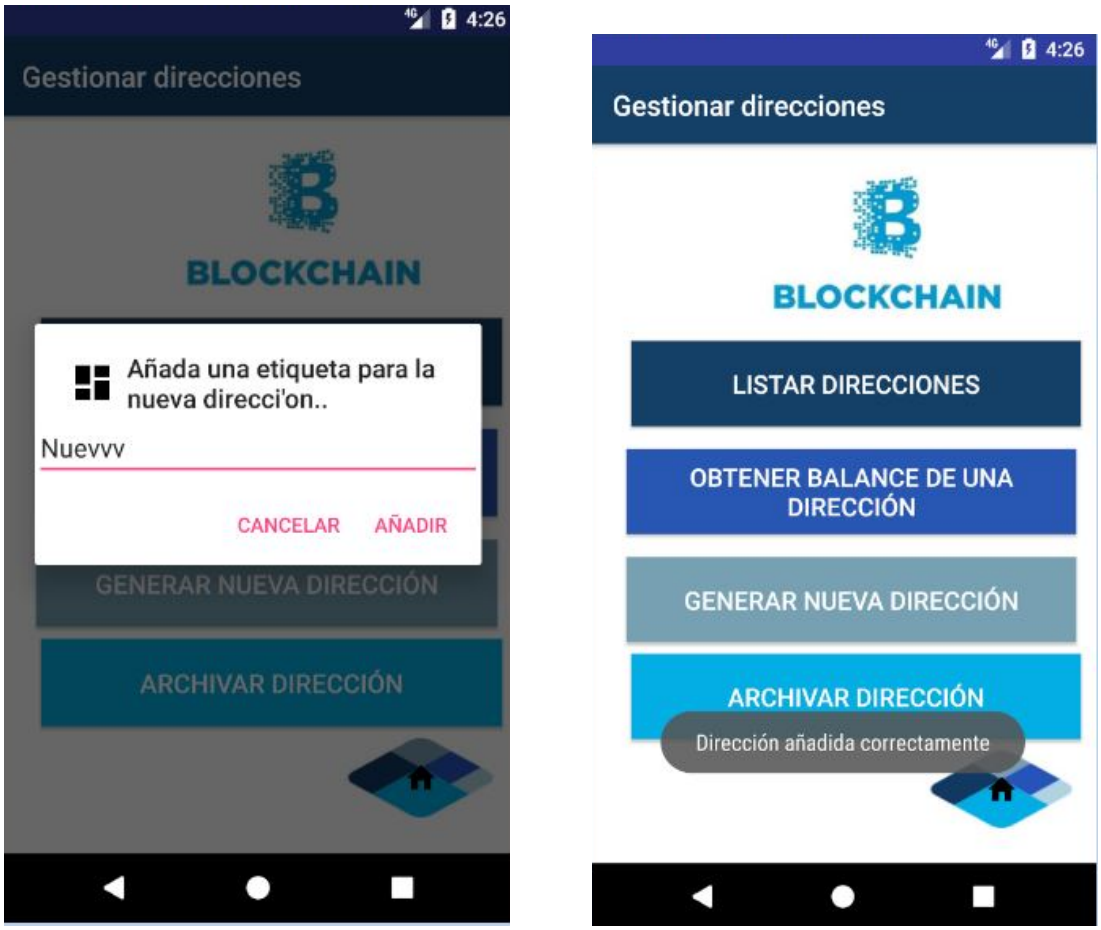
Además, observamos que se permite seleccionar la dirección para copiarla y usarla individualmente en el botón Çonsultar balance de una direcciónz,o, en el de .ªrchivar dirección":



(a) Comprobar balance (b) Archivar

Figura 8.15: Figura: Archivar direcciones

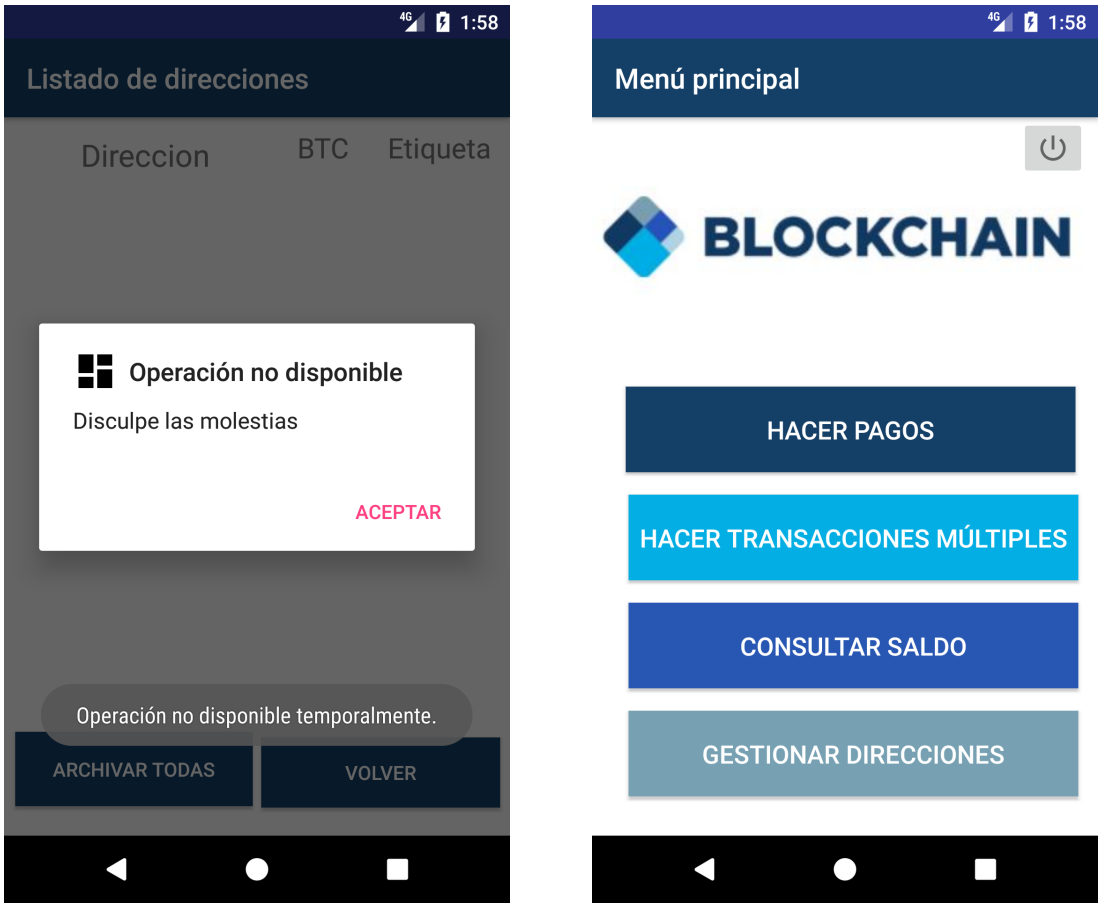
Para finalizar, sólo nos quedaría la opción de generar dirección. Lo ideal es generar una dirección por cada transferencia a realizar y luego archivarla, de ahí que esta distribución de pantalla haya sido elegida así. Como ya se explicó en la sección de la API, por imperativo lógico no sería necesario añadir una etiqueta por cada dirección, pues es un parámetro no obligatorio. Sin embargo, se ha estimado que mejora la utilidad de las transacciones el forzar al usuario a identificar dichas direcciones, por lo que para que generar una nueva dirección, solicitamos al usuario una etiqueta que será asociada a la nueva dirección generada (que obviamente por defecto se genera con balance 0):



(a) Comprobar balance (b) Archivar

Figura 8.16: Figura: Generar nueva dirección

Por último, hay que destacar que al contrario que otras muchas aplicaciones que intentan gestionar un wallet, esta aplicación no asume que el servidor sea infalible, y por tanto proporciona contingencias frente a posibles fallos del servidor. Uno de ellos se descubrió, se documentó y se notificó en el desarrollo de este proyecto, que es un error que no sigue un patrón concreto y se produce internamente en el servidor algunas veces tras añadir una dirección, pues en lugar de crear una dirección con el amount a 0 lo crea con valor null, lo que provoca errores al hacer una lectura de la cantidad de esa dirección. En este caso concreto y en otros en los cuáles es el servidor el que falla, se informará al usuario antes de llevarlo nuevamente a la pantalla principal:



(a) Comprobar balance (b) Archivar

Figura 8.17: Figura: Contingencia frente a errores