Visualización gráfica de datos de la blockchain de Bitcoin

Federico Laport Rivas
Máster en Seguridad de las Tecnologías de la Información y las Comunicaciones
Seguridad en redes y aplicaciones distribuidas

Jordi Herrera Joancomartí
Guillermo Navarro Arrivas

08/01/2018

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Visualización gráfica de datos de la blockchain de bitinio*
Nombre del autor:	*Federico Laport Rivas*
Nombre del consultor/a:	*Jordi Herrera Joancomartí*
Nombre del PRA:	*Guillermo Navarro Arrivas*
Fecha de entrega (mm/aaaa):	01/2018
Titulación::	*Máster en Seguridad de las Tecnologías de la Información y las Comunicaciones*
Área del Trabajo Final:	*Seguridad en redes y aplicaciones distribuidas*
Idioma del trabajo:	***Castellano***
Palabras clave	*Blockchain, blocksci, bitcoin*

Resumen del Trabajo:

Bitcoin se basa en una nueva tecnología llamada blockchain. Cada transacción confirmada en la red se almacena en un bloque de la blockchain. Los bloques son unidades de información ordenadas temporalmente mediante mecanismos criptográficos. Recientemente, han aparecido nuevas herramientas de análisis de código abierto que permiten extraer información interesante de la blockchain. Esta información puede darnos nuevas perspectivas sobre el comportamiento histórico del ecosistema Bitcoin. Este trabajo presenta una herramienta que analiza diez parámetros diferentes de la blockchain, materializada en una serie de scripts y una aplicación web que muestra los datos en dos tipos de visualización gráfica. La aplicación web está activa en [15].

Abstract:

Bitcoin was built over a new technology called blockchain. Every transaction settled in the network is recorded in a block of the blockchain. Blocks are information pieces temporary ordered by cryptographic constrains. Recently, new open source analysis tools were developed that can be used to extract interesting information from the blockchain. This information can be used to have new insights into the historical behavior of the Bitcoin ecosystem. This dissertation introduces a tool that analyzes ten different parameters from the blockchain, materialized in a set of scripts and a web application that displays the data in two different graphical visualizations. The web application is live in [15].

Índice

Lista de figuras

1. Introducción

1.1 Contexto y justificación del Trabajo

La blockchain de bitcoin contiene gran cantidad de información. Existe interés en realizar análisis sobre los datos existentes y han ido apareciendo herramientas que facilitan esta tarea. Por otro lado, la blockchain es un conjunto de datos que crece constantemente y es interesante disponer de herramientas visuales que permitan percibir de forma global y actualizada ciertas características o parámetros extraíbles de los datos.

Existen proyectos interesantes sobre visualización de datos de la blockchain, la mayoría de ellas con actualización en tiempo real de los datos. Este trabajo pretende aportar una herramienta práctica y extensible para la visualización de datos de análisis de la blockchain, basándose en dos tipos de visualización estandarizados.

1.2 Objetivos del Trabajo

Los objetivos iniciales del trabajo eran los siguientes:

- Definir un grupo de parámetros de análisis de la blockchain.
- Definir sistemas de visualización gráfica de los parámetros.
- Evaluar los posibles métodos de análisis.
- Diseñar y escribir scripts para obtener los datos empleando el método más apropiado.
- Diseñar y programar una aplicación web que muestre las gráficas a partir de los datos obtenidos con los scripts.
- Diseñar y programar un worker para el backend que actualice los datos en tiempo real.

1.3 Enfoque y método seguido

Se han buscado parámetros prácticos y con relevancia a la hora de ser visualizados, de forma que las gráficas pudiesen ofrecernos cierta información visual y no una simple nube de datos aparentemente aleatorios.

En cuanto al sistema de análisis, se han evaluado varias posibilidades siendo finalmente la aplicación *BlockSci* la más ventajosa.

La aplicación web se ha montado en un *framework* Python por ser el lenguaje de referencia del autor del trabajo.

Se ha intentado seguir una metodología dinámica de trabajo intentando acabar el producto cuanto antes para ir añadiendo las funcionalidades después. Esto se puede ver reflejado en la planificación (orientativa).

1.4 Planificación del Trabajo

Se realizó una planificación orientativa inicial marcando fechas para algunas entregas intermedias de trabajo. La planificación se adaptó a medida que surgieron necesidades o problemas imprevistos. En la siguiente tabla se muestran los hitos fundamentales:

Fecha	Entrega
9/10/17	PEC1 – Planning
30/10/17	Draft for definition of parameters
20/11/17	Front-end first release
22/12/17	PEC3 – Working website
08/01/18	Project + Source Code + Final Report
19/01/18	Presentation

Tabla 1: Hitos fundamentales de la planificación.

1.5 Breve sumario de productos obtenidos

Se han obtenido varios **scripts** Python para análisis de la blockchain, y una **aplicación web** de visualización.

1.6 Breve descripción de los otros capítulos de la memoria

El capítulo 2 expone los aspectos básicos de Bitcoin para el lector no experto. Se explica de forma somera sin entrar en detalles ni en aspectos irrelevantes para el objeto de este trabajo.

En el capítulo 3 se trata el análisis de la Blockchain, las herramientas existentes, los casos conocidos de visualizaciones gráficas de datos, y se expone el enfoque y metodología empleados en este trabajo. Se describen los parámetros analizados y los tipos de visualización que se generaron.

El capítulo 4 describe las herramientas desarrolladas y explica cómo ponerlas en funcionamiento.

El capítulo 5 comenta brevemente las gráficas resultantes, explicando hitos y eventos en la historia de Bitcoin que pueden observarse en algunas de las gráficas.

Por último, en el capítulo 6 se incluyen las conclusiones, una serie de aspectos a mejorar y las futuras líneas de trabajo propuestas.

El capítulo 7 contiene un glosario con términos mencionados en el trabajo y el 8 la bibliografía.

2. Básicos de Bitcoin

Bitcoin engloba un conjunto de tecnologías y conceptos que forman un sistema de almacenamiento y transferencia de valor, un sistema de moneda digital basada en criptografía. O simplemente criptodivisa. La red Bitcoin es una red peer-2-peer de nodos que se comunican a través del protocolo Bitcoin. Los nodos almacenan el historial de transacciones, propagan las nuevas transacciones y las verifican. También se llama Bitcoin a la unidad de moneda básica que usa esta red, equivalente a 100 millones de *satoshis*, la unidad indivisible de Bitcoin.

Los usuarios de la red pueden hacer transferencias de Bitcoin a otros usuarios. Estas transferencias se codifican en un formato compacto y se transmiten por la red, eventualmente siendo añadidas a un nuevo bloque del histórico de transacciones de Bitcoin conocido como *blockchain*. Las transferencias no transportan monedas, no existen monedas digitales de por sí. Las monedas existen como anotaciones en la *blockchain*. Al estar escrito todo el histórico de transacciones, podemos llegar a obtener el estado actual de distribución de bitcoins entre los usuarios de la red, pero ninguno de estos usuarios ostenta un ente físico o digital representativo de la moneda.

Lo que sí deben almacenar los usuarios son las clavas privadas asociadas a las direcciones bitcoin. Únicamente con estas claves privadas, un usuario puede firmar una transacción y así demostrar que puede gastar determinados bitcoins sin revelar su clave. No es necesario nada más. Es habitual que los usuarios almacenan colecciones de claves y direcciones en carteras digitales pero se puede incluso anotar en un papel o imprimir una clave (*paper wallets*). Las direcciones se derivan de las claves públicas. En la figura 1 se muestra un ejemplo de dirección bitcoin y clave privada, que se podría imprimir y usar como *paper wallet*.

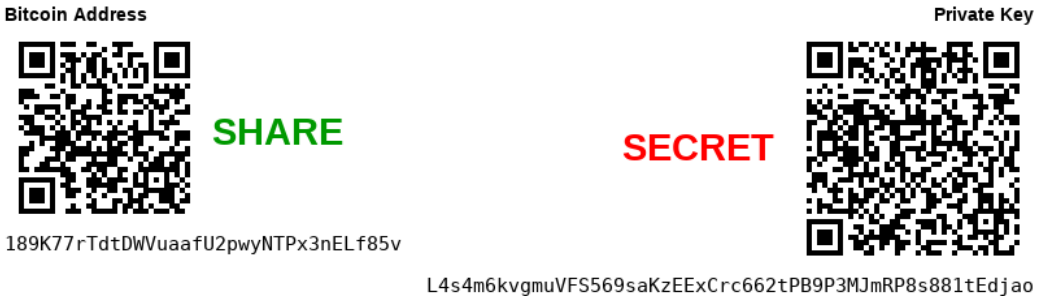

Figura 1: Dirección Bitcoin y su clave privada generadas con una conocida herramienta.

Nótese que en la figura 1 se dan la clave privada y la dirección bitcoin codificadas en Base58, un alfabeto usado específicamente en Bitcoin.

El proceso por el cual se generan nuevos bloques y se añaden a la blockchain se conoce como minería. Los nodos mineros de la red participan en una competición por encontrar la solución a un problema matemático que implicaría la creación de un nuevo bloque de la blockchain con sus respectivas transacciones validadas. Además, el minero que encuentra un bloque tiene derecho a una recompensa en bitcoins nuevos, mas las comisiones por transferencia que hayan asignado los usuarios. Es así como se crean los bitcoins por primera vez. El proceso de minado consiste en calcular el hash del encabezado del bloque de forma iterativa. En cada iteración se intenta obtener un hash con suficiente dificultad, alterando un contador para así obtener un hash distinto en cada intento.

En teoría, un nuevo bloque debería aparecer cada 10 minutos en promedio. La realidad es que la media es ligeramente más corta debido al mecanismo por el cual se autorregula la red. Mecanismo que se basa en un parámetro conocido como la dificultad. La dificultad se va ajustando dinámicamente con las marcas de tiempo de los últimos bloques para garantizar que se tiende a los 10 minutos de promedio independientemente de la cantidad de mineros. En la práctica, la capacidad minera no ha parado de aumentar exponencialmente así que normalmente el ajuste se queda corto y la media es inferior a los 10 minutos.

Cuando un minero encuentra un bloque lo propaga por la red y el resto de nodos lo validan. Una vez validado, los mineros se ponen a trabajar en un nuevo bloque. La competencia es muy fuerte y actualmente los actores mineros tienen grandes granjas de dispositivos ASIC fabricados específicamente para minería bitcoin. En la figura 2 vemos las principales pools de minería bitcoin.

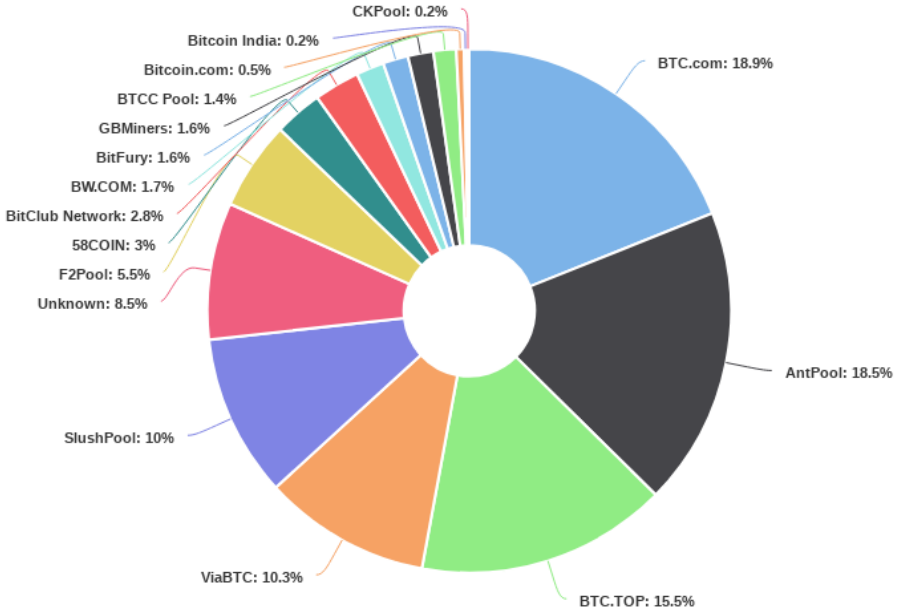

Figura 2: Distribución de poder de minado (Fuente: blockchain.info [6])

Este sistema de validación, conocido como PoW, *proof-of-work*, permite prescindir de una tercera parte en el sistema, en definitiva prescindir de una autoridad central. De este modo, las reglas se establecen por consenso. El consenso implica, por ejemplo, que la cantidad de bitcoins que se crean en cada bloque se reduce a la mitad cada 4 años (12.5 BTC al tiempo de escritura), o que con el tiempo, aproximadamente en el año 2140, dejarán de crearse nuevos bitcoins y la minería obtendrá ingresos únicamente de las comisiones. La cantidad máxima de bitcoins en circulación será de 21 millones.

2.1. El formato de transacciones en bitcoins

Las transacciones informan a la red de una transferencia de valor de un usuario a otro. Tienen al menos un input y al menos un output, y si los inputs son mayores a los outputs, la diferencia supone una comisión implícita para los mineros. Cuando los mineros incluyen transacciones en bloques estas quedan validadas, pero solo después de 6 bloques se considera una transacción completamente confirmada. Evidentemente los mineros priorizan las transacciones con comisiones más ventajosas. A fecha de diciembre de 2017, las comisiones rondaban los 600 satoshis/Byte.

El tipo de transacción más común es una transferencia de bitcoins de una dirección a otra. Los inputs de la transacción referencian a outputs de transacciones anteriores. Además, la transacción debe contener una prueba de que se poseen esos outputs referenciados. Esta prueba se materializa en un *redeem script*, que normalmente se trata de una firma digital.

Existe una implementación de referencia llamada Bitcoin Core. Es la primera implementación, escrita originalmente por Satoshi Nakamoto y otros. En Bitcoin Core, las transacciones, al igual que los bloques, se indexan por el hash de su representación binaria. Las transacciones se almacenan y transmiten en su formato binario, y se suelen deserializar en JSON para manejarlas. Un ejemplo de transacción deserializada sería el siguiente:


```
{
   "ver":1,
   "inputs":[
      {
         "sequence":4294967295,
          "witness":"02483045022100b5190974721cc9e6d1c581c55119b9
88597236168383f89046377bac2adb2472022008e2f65c6dac7ca01db2c04d74
343e2af34183367088f44419835ff494df0bdd0121036fc20db3515cad44e9ae
c35209d89d6c67032c2e5e1a26f08efab1153b8724e9",
         "prev_out":{
            "spent":true,
            "tx_index":320099057,
            "type":0,
            "addr":"3KtCUDuJAJ8jwnWeea9Ku27efiZeeC26aP",
            "value":1200000000,
            "n":0,
             "script":"a914c78fa4a975612190d0ffb6b6baa3e9142b9fe0
b087"
         },
          "script":"160014c39e5de4331c96140a2596cb97efebe3253dcc3
4"
      }
   ],
   "weight":542,
   "block_height":502530,
   "relayed_by":"0.0.0.0",
   "out":[
      {
         "spent":false,
         "tx_index":320114498,
         "type":0,
         "addr":"13dVQQqmhi4pBQ23dCCFXczvHq3knww6Ux",
         "value":1199830000,
         "n":0,
          "script":"76a9141cd6efb5000376fc411154eac3d2e98a864fcee
288ac"
      }
   ],
   "lock_time":0,
   "size":218,
   "double_spend":false,
   "time":1515068213,
   "tx_index":320114498,
   "vin_sz":1,
    "hash":"362fb31c32abbc467f6576011169a2be979fc9ad4d6c69bc1259d
6e61c09b3af",
   "vout_sz":1
}
```

Como se puede ver en la transacción anterior, tanto en el input como en el output, existe un campo *script.* En las transacciones bitcoin se usa un lenguaje básico de script RPN para bloquear los outputs y desbloquearlos. En un principio, Bitcoin Core solo permitía un tipo de script, el de clave pública. Las transacciones eran del tipo P2PKH (Pay to Public Key Hash). Hoy en día todos los nodos aceptarán transacciones con cualquier tipo de script. Las transacciones P2SH (Pay to Script Hash) abren un gran abanico de posibilidades.

Los outputs quedan bloqueados por un script que consta de una serie de datos y operadores. Los inputs por su parte, referencian a un output anterior y lo desbloquean con otro script formado también por una serie de datos y operadores. Para verificar la transacción, se concatenan los scripts y se ejecutan. Si el resultado es satisfactorio, la transacción es válida. Si no, se descarta. Como referencia pueden consultarse los capítulos 6 y 7 de [1] para una explicación técnica.

A efectos de este trabajo, es relevante destacar que se pueden realizar transacciones con outputs bloqueados de muchas formas distintas. Existen, por lo tanto, muchos tipos de transacciones: multifirma, bloqueos temporales, etc.

Además, existe un operador (OP_RETURN) que permite almacenar 80 bytes de datos arbitrarios en un output. Esto ha posibilitado la aparición de diversas aplicaciones sobre la blockchain.

2.2. El formato de la blockchain

A medida que se emiten transacciones, se van propagando por la red y los nodos las verifican antes de pasarlas a la *mempool*, el almacén temporal de transacciones válidas pero no verificadas. Los nodos mineros construyen un bloque candidato a partir de las transacciones de la mempool que consideran más ventajosas, normalmente las de comisiones más altas, y comienzan el trabajo *PoW* para encontrar un bloque válido.

Un bloque contiene principalmente las transacciones que valida y una cabecera. En la cabecera hay campos como número de versión, hash del bloque previo, un timestamp, y otros. El hash de esta cabecera es el identificador primario del bloque pero también es posible identificar un bloque por su altura en la blockchain. Es común hablar de altura (*heigh*) o profundidad (*depth*) de un bloque, haciendo referencia a la idea de que se apilan unos bloques encima de otros, consolidando cada vez más la posición de un bloque, a medida que más bloques se añaden sobre el mismo. Los bloques más recientes tienen más altura y menos profundidad.

Las transacciones en un bloque tienen un índice que las identifica. La primera, identificada con el índice *0,* es la conocida como *coinbase.* Esta

transacción la crea el nodo minero incluyendo su recompensa de nuevos bitcoins en el output.

El conjunto de todos estos bloques se conoce como *blockchain*. Actualmente hay más de medio millón de bloques, obligando a los *full nodes* a almacenar casi 200 GB.

En los últimos años, gracias a varios avances entre los cuales se incluye el operador OP_RETURN, han aparecido nuevas aplicaciones que operan sobre la blockchain de bitcoin. Colored coins y certificación de existencia de datos son algunas de las aplicaciones más populares.

3. Análisis de la blockchain

Este trabajo se centra en el análisis de la blockchain y la obtención de parámetros útiles y representativos de la actividad en la red a lo largo de su historia. En concreto, se crearon scripts que obtenían datos directamente de un nodo bitcoin mediante llamadas RPC, y calculaban los parámetros para después almacenarlos en ficheros. Algunos de los parámetros eran suficientemente complejos como para hacer que el script tardase meses en recorrer la blockchain. Bitcoin Core no está diseñado para este propósito y mucho menos su interfaz RPC. Estos scripts, que se incluyen en el repositorio, fueron sustituidos por otros que usaban herramientas específicas de análisis.

3.1. BlockSci

BlockSci [2] es una herramienta de analysis de la blockchain de bitcoin y otras blockchain similares. BlockSci está escrita en C++, almacena gran parte de la blockchain en memoria y se complementa con una interesante interfaz para Python. Esta herramienta se empleó para realizar los análisis del trabajo. Se reescribieron los scripts que operaban directamente sobre el nodo para que trabajaran con BlockSci. Como resultado, se obtuvieron scripts más ligeros y notablemente más rápidos. Por otro lado, cabe mencionar que los requisitos de memoria RAM de BlockSci son exigentes, del orden de los 60 Gbytes de memoria.

3.2. Información a extraer

En los siguientes apartados, se exponen uno a uno los parámetros escogidos para analizar, la nomenclatura empleada para identificarlos en los scripts, el tipo empleado para almacenar el dato, y cualquier otra información específica y relevante.

3.2.1. Ratio de gasto de las transacciones coinbase.

- **Descripción**: Ratio de outputs de transacciones coinbase ya gastadas. Este parámetro debería ser dinámico. A medida que aparecen nuevos bloques puede ser necesario revisar el

resultado de los bloques anteriores si se detecta una transferencia gastando un output de coinbase.
- **Tipo de dato**: Se almacena como *float,* un ratio entre 0 y 1. Si se han gastado todos los outputs el valor es 1.
- **Nombre de variable en el código**: spent_coinb_vouts

3.2.2. Tiempo hasta el gasto de las transacciones coinbase.

- **Descripción**: Tiempo promedio que tardan los outputs de una transacción coinbase en gastarse.
- **Tipo de dato**: Se almacena como *integer* en unidades de segundos.
- **Nombre de variable en el código**: tts_coinb_txos

3.2.3. Número de direcciones en output de transacciones coinbase.

- **Descripción**: Simple conteo de direcciones en los outputs de cada transacción coinbase.
- **Tipo de dato**: *Integer*.
- **Nombre de variable en el código**: coinb_addr

3.2.4. Throughput de la red bitcoin.

- **Descripción**: Throughput medio de la red calculado respecto a los últimos 12 bloques. Se introduce esta corrección para evitar el error inducido por la desviación de los timestamps de los bloques respecto a la hora correcta. Como red descentralizada, la red bitcoin permite variaciones considerables en los timestamps que marcan los nodos en los bloques que generan.
- **Tipo de dato**: *Float*, en unidades de *tx/min.*
- **Nombre de variable en el código**: throughput

3.2.5. Numero de transacciones por bloque.

- **Descripción**: Número total de transacciones en cada bloque.
- **Tipo de dato**: *Integer*.
- **Nombre de variable en el código**: tx_count

3.2.6. Número de transacciones con *timelocks* por bloque.

- **Descripción**: Número total de outputs con timelocks en todas las transacciones del bloque. Nótese que, en teoría, este parámetro podría ser mayor que el número de transacciones totales del bloque.
- **Tipo de dato**: *Integer.*
- **Nombre de variable en el código**: timelocked_txos

3.2.7. Número de transacciones con *multisignature* por bloque.

- **Descripción**: Número total de outputs con multisignature en todas las transacciones del bloque. Este parámetro, al igual que el anterior, podría superar al numero total de transacciones del bloque.
- **Tipo de dato**: *Integer*.
- **Nombre de variable en el código**: multisign_txos

3.2.8. Número de transacciones P2SH por bloque.

- **Descripción**: Número total de outputs P2SH, incluyendo transacciones *SegWit*. Este parámetro, al igual que los dos anteriores, podría superar al numero total de transacciones del bloque.
- **Tipo de dato**: *Integer.*
- **Nombre de variable en el código**: p2sh_txs

3.2.9. Número de transacciones usando OP_RETURN por bloque.

- **Descripción**: Número total de outputs usando OP_RETURN en todas las transacciones del bloque.
- **Tipo de dato**: *Integer.*
- **Nombre de variable en el código**: op_return_txos

3.2.10. Número de transacciones de aplicaciones conocidas por bloque.

- **Descripción**: Número total de outputs de transacciones utilizando cada una de las aplicaciones conocidas que usan OP_RETURN para almacenar datos en la blockchain.
- **Tipo de dato**: Lista de *integers*.
- **Nombre de variable en el código**: app_op_return_txos

3.3. Análisis básico del estado del arte en visualizaciones web de datos de la blockchain

Muchos sitios en Internet están mostrando visualizaciones gráficas de datos relativos a bitcoin: la red bitcoin, la blockchain, los mercados, etc.

Desde los primeros años de existencia de bitcoin, fueron apareciendo aplicaciones web que ofrecían interesantes datos de análisis de la blockchain, conocidos como *blockchain explorers*. Blockchain.info [6] es uno de los servicios más usados actualmente. Normalmente, este tipo de sitios muestra gráficas clásicas con datos de métricas básicas de la blockchain. Además, casi todas las *altcoins* tienen algún tipo de *blockchain explorer* gracias a la existencia de exploradores genéricos adaptables open-source, como por ejemplo Iquidus Explorer [7].

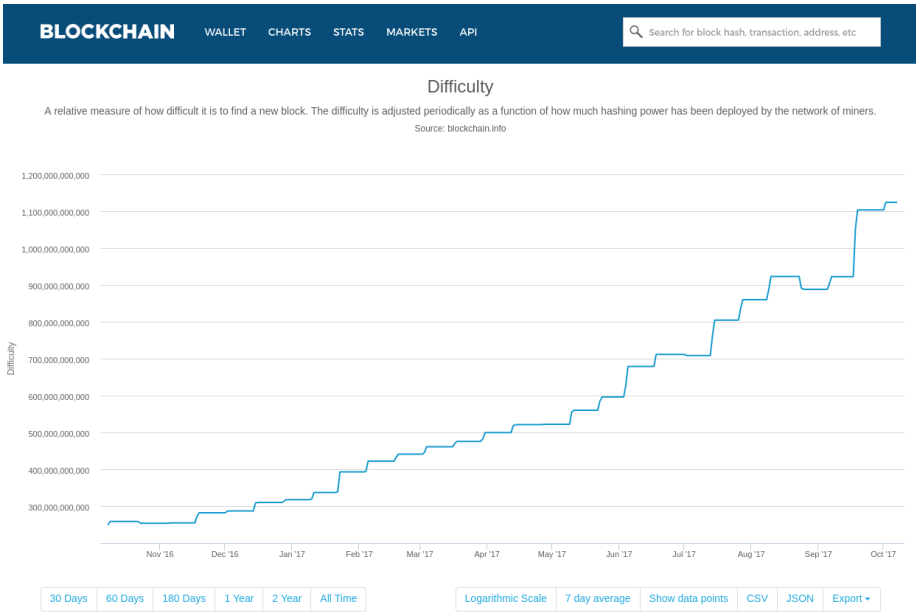

Figura 3: Gráfica de blockchain.info

Por otro lado, se puede extraer información interesante de los mercados. Estos datos provienen de fuera de la red bitcoin. Los *cripto exchanges,* mercados de criptomonedas, manejan gran cantidad de información que a menudo es accesible vía APIs de sus propios servidores. Pueden verse gráficas en vivo de los mercados de criptodivisas en cualquiera de estas plataformas, como por ejemplo Kraken [8].

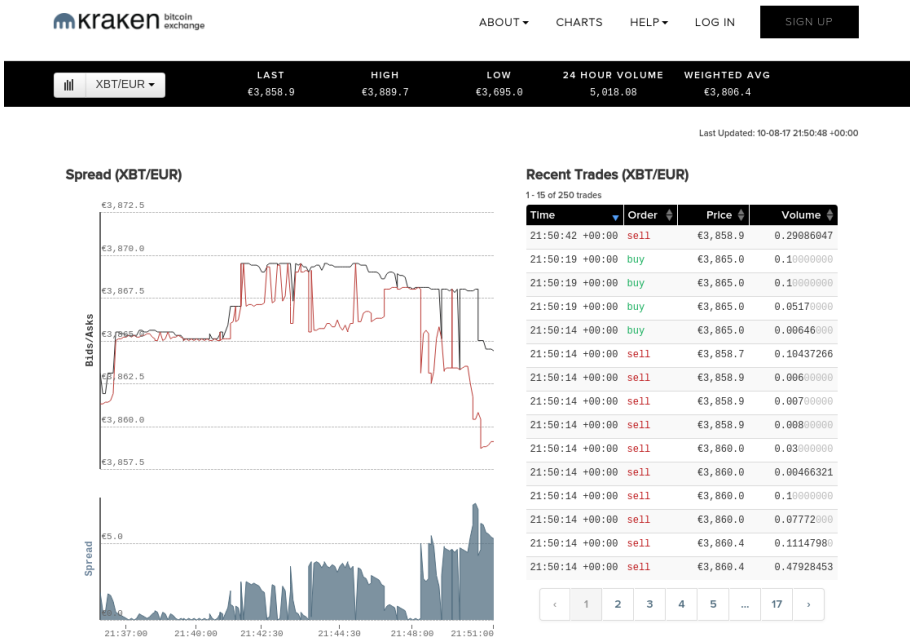

Figura 4: Gráficas de trading en kraken.com

Además de estos dos enfoques ampliamente extendidos (exploradores de blockchain y gráficas de mercados), se pueden encontrar gran variedad de nuevos tipos de visualizaciones mostrando datos de la

blockchain en diferentes formatos. Utxo-stats [9], por ejemplo, es un sitio *one-page* que muestra dos visualizaciones de la blockchain completa. Cada visualización es una matriz de píxels donde cada píxel representa un bloque de la blockchain. Una de las columnas muestra el número de outputs de transacciones no gastados, *UTXOs,* y la otra el valor combinado en bitcoins de dichos outputs.

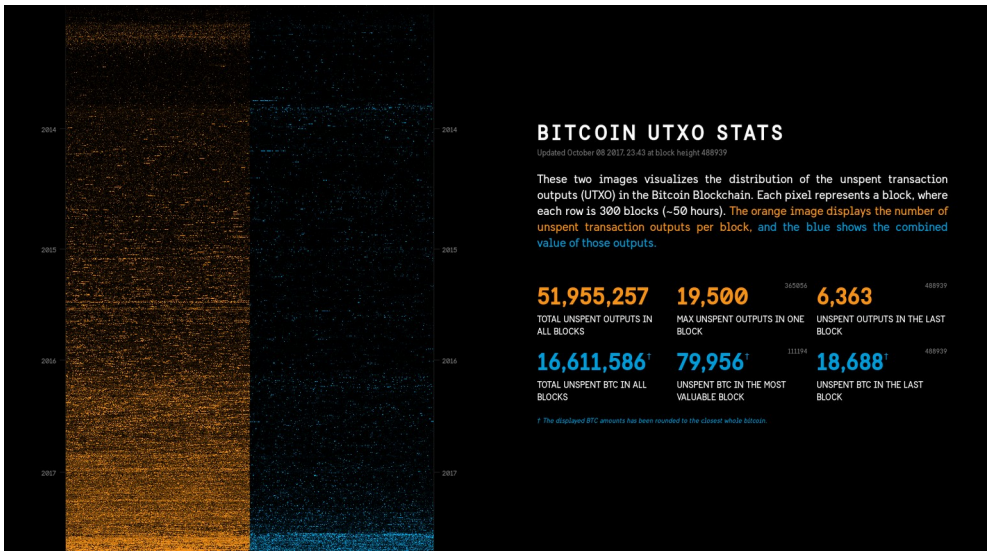

Figura 5: Utxo-stats, sitio de página única.

Existen otros experimentos interesantes como BitBonkers [10]. El sitio muestra bloques y bolas cayendo, representando bloques de la blockchain encontrados y transferencias llegando a la red, respectivamente. El tamaño de las bolas es proporcional al monto total de la transacción representada. Un enfoque similar se observó en [11]. Otros sitios interesantes se pueden encontrar en [12], [13] y [14].

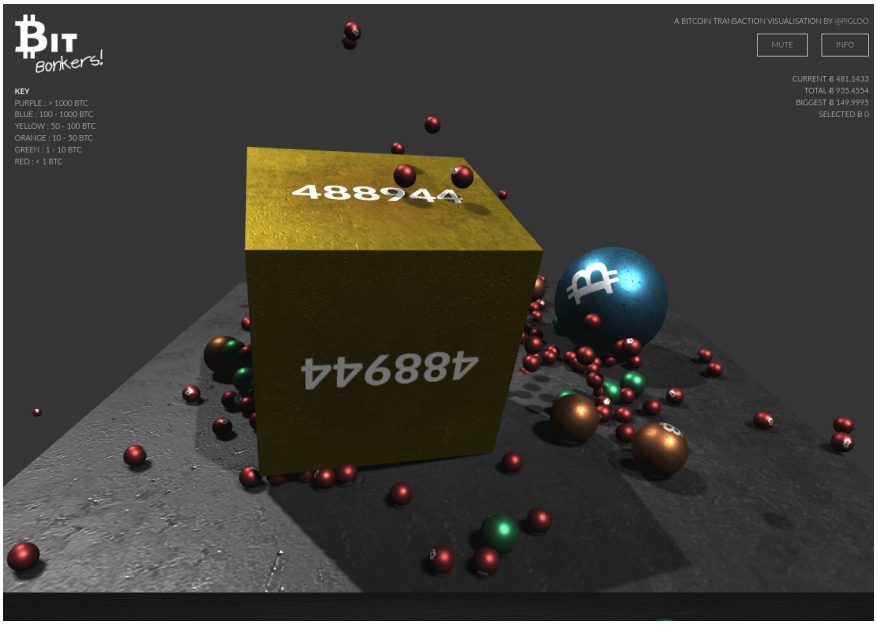

Figura 6: Bitbonques, sitio de página única.

3.4. Visualización de la información

Para visualizar los datos, se optó por dos posibles representaciones gráficas para cada parámetro. La primera es una matriz de píxels y la segunda una gráfica de línea convencional.

La matriz de píxels es una idea tomada de utxo-stats.com. Cada píxel representa un bloque. El color del píxel es función del valor del parámetro representado. En la práctica, se materializa con un canvas HTML de ancho 504 píxels y altura variable. Se ha escogido este ancho porque 1008 píxels se corresponden, en teoría, con 1 semana de blockchain. A diferencia de Utxo-stats, los bloques más recientes están arriba y los más antiguos abajo. Esto concuerda con la noción de profundidad y altura de bloque que existe en la comunidad bitcoin. En algunos de los parámetros representados, se usan distintos colores en los píxels para identificar distintas series de datos. En este caso solamente se muestra la seria de datos cuyo valor es mayor para ese píxel.

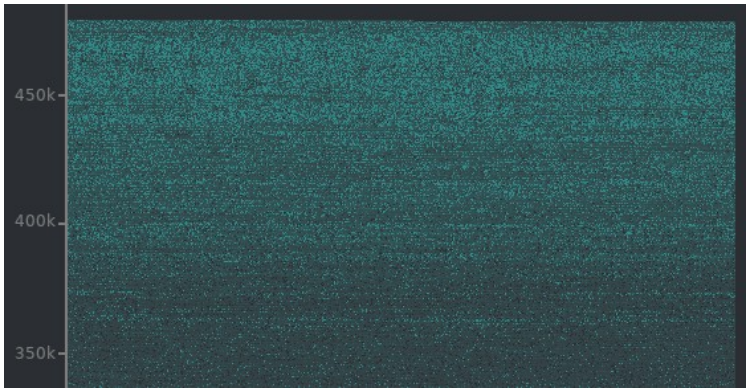

Figura 7: Captura de la visualización basada en matriz de pixels

Por su parte, la segunda representación es una gráfica clásica con ejes invertidos (abscisa en vertical y ordenada en horizontal). De este modo se mantiene la noción de altura de bloque. En este caso, cada punto de la gráfica representa un segmento de blockchain de 1000 bloques. Se calcula la media del valor representado para los mil bloques. También se utilizan distintos colores para las distintas series de datos.

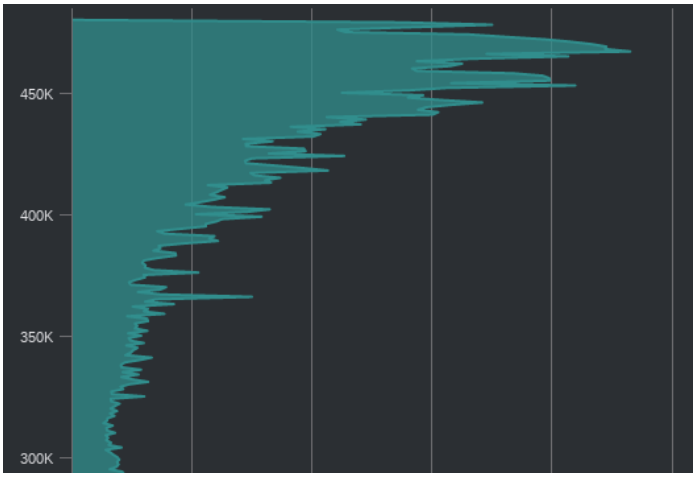

Figura 8: Captura de la visualización de gráfica de línea

4. Herramientas desarrolladas

El producto de este trabajo está materializado en, por un lado, scripts de análisis sobre BlockSci que generan ficheros de datos con resultados, y por otro, una ligera aplicación web para visualizar estos datos. Prácticamente todo el código está escrito en Python, a excepción de unos cientos de líneas en JavaScript.

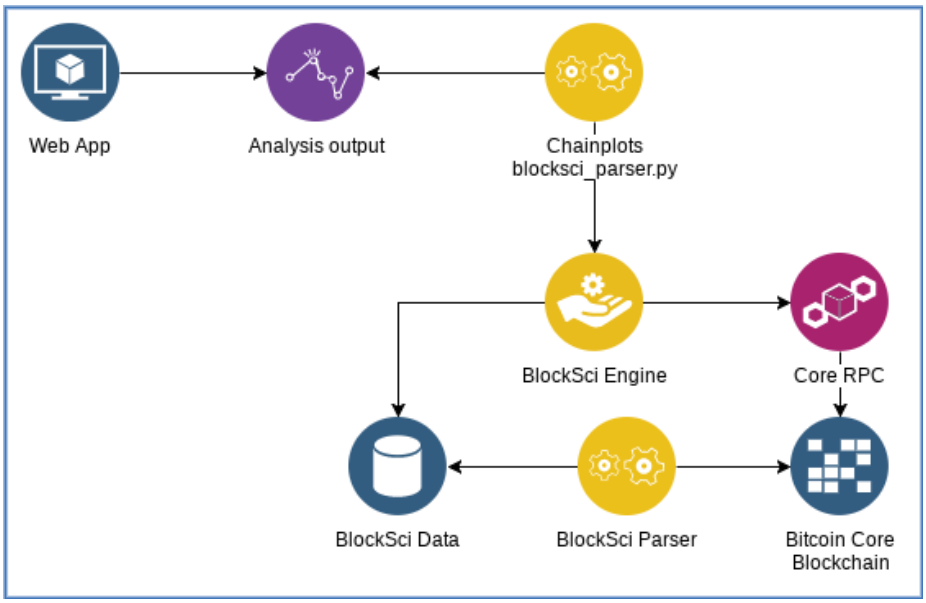

Figura 9: Diagrama de funcionamiento de la aplicación

4.1. Scripts de análisis

Se han escrito cuatro scripts de análisis:

- **random_generator.py**: Genera valores aleatorios para los 10 sets de datos.
- **static_generator.py**: Calcula todos los parámetros de análisis propuestos obteniendo la información a través de la interfaz RPC de Bitcoin Core. Crea los ficheros de datos.
- **blocksci_generator.py**: Calcula todos los parámetros de análisis propuestos con BlockSci (que a su vez funciona sobre Bitcoin Core). Crea los ficheros de datos.
- **thousand_generator.py**: A partir de los ficheros de datos, genera otro nuevo grupo de ficheros calculando la media de cada 1000 valores. Estos ficheros alimentan el código que genera la gráfica de línea en el front-end.

4.2. Aplicación web

La aplicación web usa los datos generados por los scripts para mostrar las dos tipologías de gráfica descritas. Se ha diseñado una barra de navegación horizontal en la parte superior del cuadro. En la misma, se dispone un botón por cada parámetro para facilitar el acceso a cada uno

de ellos de forma rápida. Cada parámetro dispone de una página en la que se puede cambiar de una visualización gráfica a otra (matriz de píxels o gráfica de líneas).

4.2.1. Pila de tecnologías

- **Web Backend Python/Django**: El backend prácticamente carece de responsabilidad. Se limita a funciones de *templating* con los contenidos de las diferentes pestañas para generar cada una de las páginas y servirlas junto a los contenidos estáticos del sitio: el código CSS, JavaScript, y los datos de análisis para proceso en el front-end.

- **Web Frontend con bootstrap, JS con JQuery**: El front-end genera las gráficas al cargar la página. Se emplea JQuery para hacer peticiones AJAX y obtener los datos del análisis. La matriz de píxels se genera punto a punto, con JavaScript en un *canvas* HTML. La gráfica de línea se genera con la librería *highcharts.js* [5].

4.3. Descarga de repositorio y *setup* del nodo

Para ejecutar cualquiera de las herramientas de este trabajo, lo primero es descargarse el repositorio. Podemos descargarlo directamente de [3] o sincronizarlo con la herramienta *git*:

```
$ git clone https://github.com/fedelaport/chainplots.git
```

Para ejecutar los scripts necesitamos un nodo Bitcoin Core corriendo en la máquina. El nodo debería indexar todas las transacciones por lo que hay que editar el archivo de configuración añadiendo las siguientes líneas:

```
# Add this to bitcoin.conf
txindex=1
dbcache=2048
```

Así ya podemos arrancar el nodo:

```
$ bitcoind -daemon -datadir=/PATH/TO/FOLDER/BLOCKCHAIN -rest
```

También se pueden utilizar los scripts incluidos en el repositorio para arrancar y parar el nodo: start.sh y stop.sh. Es necesario actualizar las rutas de los directorios.

4.4. Ejecución de scripts de análisis

Una vez tenemos el nodo corriendo podemos ejecutar el script static_generator.py. Este script es la versión **sin BlockSci**. Se ha mantenido en el repositorio porque las modificaciones suponían una refactorización.

Para ejecutar el script blocksci_generator.py necesitamos instalar BlockSci. Para ello, podemos seguir las indicaciones en [2] o directamente inicializar una instancia en EC2 en Amazon Web Services a partir de la imagen existente. También se explica en [2].

Una vez tenemos acceso a una máquina con Bitcoin Core corriendo y BlockSci preparado, podemos correr el script:

```
$ python3 blocksci_generator.py -h
usage: blocksci_generator.py [-h] [-v] [-s START] [-g GROUPING]
[-j JSONFILE]
                             [-c CONFFILE] [-d]

Blockchain analysis tool. Dataset builder.

optional arguments:
  -h, --help            show this help message and exit
  -v, --verbose
  -s START, --start START
                         Starting block. Heigh of the first
analyzed block. 1
                        or more.
  -g GROUPING, --grouping GROUPING
                        Output grouped by GROUPING elements.
  -j JSONFILE, --jsonfile JSONFILE
                          Output JSON filename. Will be saved in
the script
                        directory.
  -c CONFFILE, --conffile CONFFILE
                        Blockchain directory.
  -d, --dataset         Generate the web static data
```

Esta configurado en forma de herramienta en la línea de comandos. Para generar los datos para la aplicación web, debemos pasar la opción -d. La opción -c nos permite definir la ubicación de los datos del *parser* de BlockSci. Si no la pasamos, escogerá el directorio por defecto de la instancia de Amazon. La opción -v nos da información a medida que avanza el script y la -g nos agrupará el output para reducir la verbosidad. Por ejemplo, podemos generar datos para la aplicación web así:

```
$ python3 blocksci_generator.py -v -g 1000 -d
```

4.5. Puesta en marcha del servidor Django en entorno de desarrollo

Deberíamos haber generado los datos con los scripts anteriores para poder usar la web. Debido a las posibles dificultades que se pueden encontrar para tener una máquina con BlockSci funcionando, el repositorio incluye datos hasta la altura de bloque 480.000 aproximadamente.

Para poder correr el servidor debemos tener instalado python3 y pipenv:

```
$ sudo apt-get install python3
$ pip install pipenv
```

Ahora instalamos todas las dependencias:

```
$ pipenv install
```

Después, ejecutar el servidor de desarrollo de Django se reduce a:

```
$ pipenv run python3 manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).
January 07, 2018 - 22:28:07
Django version 2.0.1, using settings 'chainplots.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

5. Información del análisis realizado de la blockchain

Se pueden realizar diversas observaciones respecto a los resultados del análisis realizado. Podemos realizar las afirmaciones expuestas a continuación, incluyendo algunas apreciaciones subjetivas.

5.1. Ratio de gasto de las transacciones coinbase.

- En primer lugar, puede verse como los últimos bloques no han podido gastar todavía el output. Es una franja muy pequeña ya que se trata de únicamente 100 píxels.
- Hay una cantidad importante de transacciones que no se han gastado entre los primeros 50k a 100k bloques. En ese momento la minería todavía no estaba industrializada. Son fortunas de los pioneros de bitcoin. También es probable que haya bitcoins perdidos en muchas de esas coinbase.
- En general, casi todo el grueso de las transacciones coinbase ha sido gastado, excepto dos pequeñas zonas entorno a la altura de bloque 300k y 350k. Además, ambas franjas suponen un descenso similar, por lo que se puede suponer que se trata de un actor minero con una potencia de la misma proporción.

5.2. Tiempo hasta el gasto de las transacciones coinbase.

- Se aprecian tiempos más largos en los primeros 100k bloques. Hay una bajada brusca, que se entiende que se trata del inicio de la competencia minera.
- Entre los 250k y los 350k los tiempos de gasto eran algo mayores. Puede ser alguna política de pagos habitual en las pools en esa época.

5.3. Número de direcciones en output de transacciones coinbase.

- Se vuelve a apreciar un patrón en los primeros 100k. Una única dirección por transferencia coinbase.
- En general, hay unas 2 direcciones de media exceptuando un período entre los bloques 250k y 350k donde la media ronda las 10 direcciones.

5.4. Throughput de la red bitcoin.

- Se pueden notar dos escalones de crecimiento del throughput de la red. En primer lugar entorno a los 125k, en segundo lugar entorno a los 180k.
- A partir de los 350k o 400k bloques el crecimiento es prácticamente exponencial.
- En la actualidad la red está saturada, continuamente cerca del throughput máximo teórico de la red, unas 7 tx/s.

5.5. Numero de transacciones por bloque.

- Se aprecia un crecimiento continuado. Las conclusiones son equivalentes a las del parámetro anterior. Aunque no son iguales, ya que el tiempo entre bloques se ve influenciado por las variaciones fuertes en el poder de minado total de la red. Cuanto más rápido crece, más incorrecta es la corrección de dificultad que realiza la red, por lo que los bloques salen más rápido. Esto implica que a una misma cantidad de transacciones por bloque, el throughput puede ser mayor.

5.6. Número de transacciones con *timelocks* por bloque.

- Crecen notablemente a partir del bloque 350k.

5.7. Número de transacciones con *multisignature* por bloque.

- Se observa un crecimiento fuerte a partir del bloque 300k y aproximadamente en el 420k deja de utilizarse. Esto tiene que ver con el surgimiento de otros tipos de multifirma que se pueden contener en transacciones P2SH, resultando en transacciones más sencillas y baratas.

5.8. Número de transacciones P2SH por bloque.

- A partir del bloque 350k crece notablemente, de forma similar a los timelocks.

5.9. Número de transacciones usando OP_RETURN por bloque.

- A partir del bloque 300k empiezan a aparecer como habitual el uso de OP_RETURN.
- A partir del bloque 375k el uso de OP_RETURN aumenta notablemente.

5.10. Número de transacciones de aplicaciones conocidas por bloque.

- Coincidiendo con el punto anterior, a partir del bloque 375k aparecen multitud de aplicaciones sobre la blockchain.
- La mayoría de aplicaciones muestran un uso muy dispar, con una frecuencia de aparición aparentemente aleatoria.
- A partir del bloque 440k aumentan notablemente los usos de OP_RETURN que no se corresponden con ninguna de las aplicaciones configuradas en BlockSci.

6. Conclusiones

El presente Trabajo Fin de Máster se ha desarrollado a lo largo de cinco meses de trabajo. En los primeros dos meses, se enfocó el trabajo en investigación y formación sobre la materia. Entender bitcoin y la tecnología que lo sustenta ha permitido que el trabajo se desarrollase con facilidad. En los siguientes tres meses, se organizó el trabajo siguiendo (débilmente) la planificación. Se definieron parámetros de análisis, se intentaron varios métodos de análisis hasta lograr el adecuado, se construyó un front-end web y, por último, un backend.

Sin duda, ha supuesto un gran avance en los conocimientos sobre bitcoin para el autor.

Las complicaciones durante el transcurso del proyecto y la escasez de tiempo no han permitido resolver algunas de las funcionalidades más interesantes. En concreto, se notará en falta la actualización automática de los datos contra un nodo o contra algún servicio web. Por otra parte, cabe mencionar que es intención del autor seguir con el desarrollo de la herramienta, por lo que en [3] y [15] podrán consultarse las últimas versiones.

También se han podido sacar conclusiones interesantes a raíz del análisis de los gráficos. Pueden percibirse cambios en el ecosistema bitcoin que afectaron a la información en la blockchain. El despegue de la competencia minera, las aplicaciones sobre blockchain, la saturación de la red bitcoin. Todo sin hacer referencia a ningún tipo de dato de mercado.

Se plantean las siguientes funcionalidades para seguir trabajando en la aplicación:

- Actualización automática de datos (requiere el diseño de un nuevo worker para el backend).
- Ampliación de las tipologías gráficas.
- Inclusión de detección de gastos SegWit en la gráfica de P2SH.
- Nuevos parámetros de alto nivel, en los que intervengan más variables y den información más interesante.
- Nuevos parámetros con datos de mercado.

Para futuros trabajos independientes de la aplicación web desarrollada:

- Análisis en BlockSci de nuevos parámetros.
- Análisis de parámetros económicos en BlockSci con datos de blockchain y de mercados.
- Análisis en BlockSci de otras blockchain compatibles. Consultar [2].

7. Glosario

- Altcoin: Criptodivisa alternativa al bitcoin. Actualmente el término engloba casi cualquier tecnología basada en blockchain con tokens *tradeables*.
- Bloque: Unidad de información de la blockchain compuesto por un conjunto de transacciones y un header, cuya *solución* ha sido encontrada por un minero.
- Blockchain: Conjunto de bloques ordenados cronológicamente y vinculados ordenadamente por criptografía.
- Blockchain explorer: Aplicación web que permite explorar datos de la blockchain.
- Cartera Digital: Archivo y/o software que almacena direcciones bitcoin y claves privadas.
- Coinbase (transferencia): Primera transferencia en un bloque en la que los mineros transfieren los bitcoins de nueva creación a sus direcciones.
- Dirección Bitcoin: Cadena de bytes derivada de la clave pública asociada a una clave privada. Actualmente se llama dirección a cualquier hash al que se le puede asignar un output.
- Full node: Nodo en la red que realiza todas las tareas posibles, almacenando la blockchain completa entre otras responsabilidades.
- Minería: Competencia en la red bitcoin por encontrar un hash de un bloque con suficiente dificultad como para considerarse válido y así recibir la recompensa.
- Minero: Actor o nodo en la red que compite en la minería.
- RPN (Reverse Polish Notation): Método algebraico de introducción de datos. Se usa en los scripts bitcoin.
- Paper wallet: Pareja de clave privada y dirección bitcoin impresa en papel, habitualmente en Base58 y con códigos QR.
- Pool (mining pool): Servidor que reparte trabajo de minería entre sus clientes y reparte las recompensas mineras proporcionalmente al trabajo.
- PoW: *Proof of Work.*
- Satoshi: Unidad indivisible de la moneda bitcoin. Equivalente a 0.000000001 BTC.
- SegWit (Segregated Witness): Solución implementada en bitcoin a través de un soft fork (retrocompatible) en la que se separan los scripts de desbloqueo de la transacción incluida en los bloques.
- Throughput: Capacidad de una red, en este caso cantidad de transferencias que se pueden procesar por unidad de tiempo.
- UTXO (Unspent Transaction Output): Output de transacción que no ha sido gastado.
- Wallet: ver Cartera Digital.

8. Bibliografía

[1] **Mastering Bitcoin, 2nd Edition**. Andreas Antonopoulos: https://github.com/bitcoinbook/bitcoinbook

[2] **BlockSci: Design and applications of a blockchain analysis platform.** Harry Kalodner et al. : https://arxiv.org/pdf/1709.02489.pdf

[3] **Repositorio del TFM**: https://github.com/fedelaport/chainplots

[4] **Bitcoin: A Peer-to-Peer Electronic Cash System**. Satoshi Nakamoto: https://bitcoin.org/bitcoin.pdf

[5]: https://www.highcharts.com/

[6] **Blockchain.info**: https://blockchain.info/

[7]: https://github.com/iquidus/explorer

[8]: https://www.kraken.com/charts

[9]: **utxo-stats.com**. Timoty E. Johansson: https://utxo-stats.com/

[10]: https://bitbonkers.com/

[11]: http://bitcoin.interaqt.nl/

[12]: https://blocks.wizb.it/

[13]: http://dailyblockchain.github.io/

[14]: https://bitnodes.21.co/nodes/network-map/

[15]: https://chainplots.herokuapp.com/