UNIVERSIDADE FEDERAL DA GRANDE DOURADOS

FACULDADE DE CIÊNCIAS EXATAS E TECNOLOGIA

TRABALHO DE CONCLUSÃO DE CURSO

BACHARELADO EM SISTEMAS DE INFORMAÇÃO

BITCOIN

Ricardo Matias de Almeida

Orientadora Profª Drª Janne Y. Y. Oeiras Lachi

DOURADOS – MS

2019

BITCOIN

Trabalho de Conclusão de Curso apresentado à Faculdade de Ciências Exatas e Tecnologia da Universidade Federal da Grande Dourados, como requisito à obtenção do título de Bacharel em Sistemas de Informação.

Orientadora: Profª Drª Janne Y. Y. Oeiras Lachi.

DOURADOS – MS

2019

AGRADECIMENTOS

Agradeço primeiramente a minha família, em especial minha mãe Francisca e meu pai Edesom, que sempre me apoiaram e orientaram para o caminho acadêmico.

Ao meu irmão por dar o exemplo de ótimo estudante tornando-se o Doutor Marcelo e também por todo apoio nos assuntos universitários.

A minha esposa Dayene, pois sem ela eu simplesmente não seria capaz. Sua força e alegria me ajudaram nos momentos mais difíceis da minha vida e me guiaram aos mais felizes também.

Aos pais da minha esposa, Célio e Romilda, que me acolheram como filho em momentos muito difíceis, e também prestaram muito apoio na caminhada ao diploma.

A minha orientadora, Prof. Dra. Janne Y. Y. Oeiras Lachi, pela orientação e incentivo para conclusão deste trabalho e da graduação.

Ao meu amigo Gabriel com quem divido lutas de longa data, pois sempre acreditou que eu era capaz de realizar este trabalho.

A todos eles, meu maior obrigado.

RESUMO

O presente trabalho explana um conceito tecnológico bastante disseminado no mundo da economia: as moedas virtuais. Neste trabalho será abordada a primeira criptomoeda: a bitcoin. Serão apresentados vários conceitos sobre essa moeda e destacadas como e quais mudanças ela oferece, de que maneira este novo conceito pode impactar nossa atual realidade e quais dificuldades as criptomoedas podem enfrentar. Para tanto será realizado um estudo geral sobre o *software* denominado Bitcoin por meio de pesquisa bibliográfica em artigos e monografias sobre o tema e em portais de notícias sobre tecnologia e economia.

Palavras-Chave: Criptomoeda, Bitcoin, *Blockchain*.

Bitcoin

Ricardo Matias de Almeida, ricardoalmeida77@gmail.com, UFGD

Janne Y. Y. Oeiras Lachi, janneoeiras@ufgd.edu.br, UFGD

RESUMO

O presente trabalho explana sobre um conceito tecnológico bastante disseminado no mundo da economia: as moedas virtuais. Neste trabalho será abordada a primeira criptomoeda – a bitcoin. Serão apresentadas várias perspectivas e conceitos dessa moeda e destacadas como e quais mudanças ela oferece, de que maneira este novo conceito pode impactar nossa atual realidade e quais dificuldades as criptomoedas podem enfrentar. Para tanto será realizado um estudo geral sobre o software denominado Bitcoin por meio de pesquisa bibliográfica em artigos e monografias sobre o tema, em portais de notícias sobre tecnologia e economia.

Introdução

Moeda é o meio pelo qual são efetuadas as transações financeiras. Ao efetuarmos um pagamento no Brasil, podemos optar por um dos seguintes tipos de pagamento mais comuns atualmente: eletrônico, via boleto ou cartão de crédito, ou com o uso de cédulas, com um conjunto de notas no valor do pagamento. O que influencia na tomada de decisão é o tipo de transação que queremos. Se a escolha for por um pagamento rápido, sem custo adicional, e que não requer intermediários, nesse caso convém a opção pelas cédulas (o dinheiro físico). Por outro lado, ao escolher pelo pagamento eletrônico, a transação deixa de ser tão rápida, visto que o vendedor só recebe o dinheiro depois de algum tempo, e que na maioria das vezes apresenta um custo maior, seja pela tarifa do boleto, ou pelas taxas da operadora de cartões. Por fim, esse tipo de pagamento ainda precisa de alguns intermediadores, como um banco por exemplo.

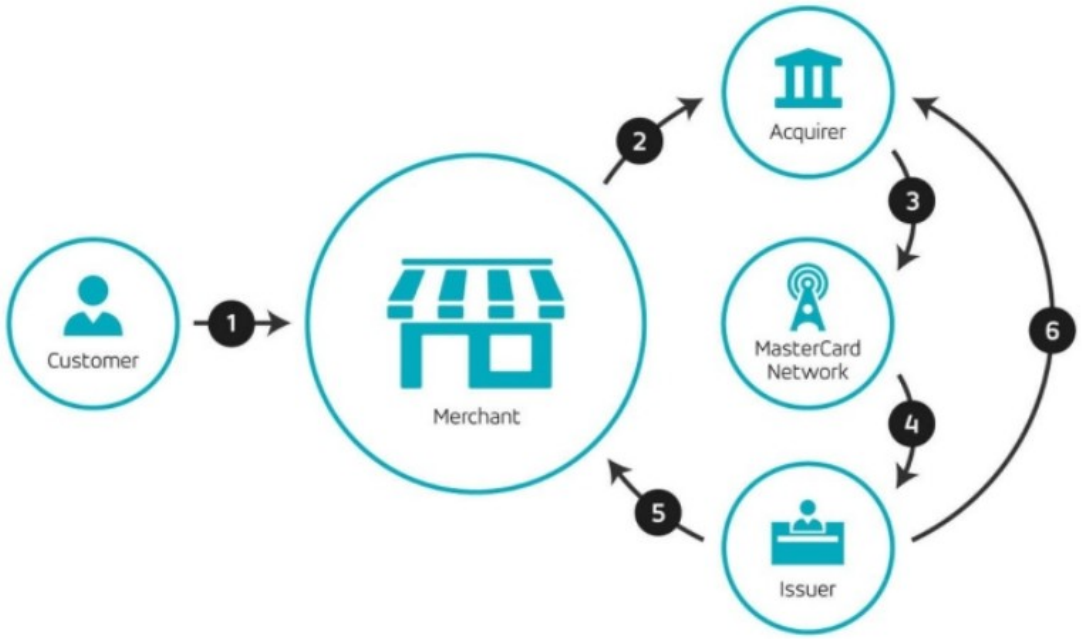
Existem ainda outras formas de pagamento. No comércio é comum o uso de cheques e cartões de débito e crédito em substituição ao dinheiro em espécie, podendo ser uma escolha por motivos de segurança devido ao valor da transação, ou por praticidade entre as partes envolvidas. Nestes métodos é necessária a intermediação de empresas auxiliares para a validação da transação. No caso de pagamento por cheque, geralmente há um banco ou instituição financeira que emite o talão em nome do cliente e faz o débito na sua conta, conforme o Banco Central do Brasil (BACEN) [1]. Porém nesse caso existe um acordo entre comprador e vendedor para aceitação do cheque, pois a compensação só é feita na instituição financeira em que o vendedor possuir conta, e não na loja.

Se o cliente optar pelo pagamento com cartão de débito ou crédito, os procedimentos são ainda maiores, segundo descrito na página da Mastercard, empresa estadunidense, que atua internacionalmente no setor de pagamentos [2].

Conforme a figura 1, a primeira parte é o cliente (*Customer*), que inicia a transação financeira com a intenção de compra ou pagamento. A segunda parte, neste cenário, geralmente é o lojista (*Merchant*), parte recebedora que comercializa o produto ou serviço, objeto principal e motivo da

transação. Para o método de pagamento selecionado tornam-se necessárias mais duas partes, a emissora de cartões de débito e crédito (*Acquierer*), empresa que põe sua bandeira no cartão (*MasterCard Network*) e a empresa que administra a máquina de cartões (*Issuer*), responsável pela aceitação da bandeira do cartão.

Figura 1 – Demonstração de algumas etapas envolvendo o pagamento por cartão de crédito.



Fonte: Mastercard [2].

Existem ainda os bancos que emitem o cartão do cliente e que geralmente recebem os créditos da máquina de cartões. Porém, algo que passa despercebido ao cliente, e está embutido no valor total da compra, são as taxas de operação. Por exemplo, o cliente que possui um cartão de débito paga tarifa de manutenção de sua conta junto ao banco. Quando possui um cartão de crédito, geralmente paga anuidade em suas faturas. O lojista paga uma porcentagem das transações que efetua na máquina de cartões ao emitente da máquina. Dessa forma uma simples transação entre lojista e cliente passa a ter vários intermediários. E ainda, quando se tratar de uma transação internacional, dependendo da bandeira do cartão do cliente, toda intermediação ganha alcance mundial, com algumas taxas adicionais devido à natureza da transação. No caso do Brasil, temos o Imposto sobre Operações Financeiras (IOF), referente às operações realizadas no exterior, quando o cartão de crédito é usado para uma compra de um vendedor fora do país [3]. A intenção de todo esse sistema eletrônico de pagamentos é que um cliente possa ter um cartão de qualquer bandeira, emitido por qualquer banco e fazer seu pagamento eletrônico em qualquer loja com qualquer máquina de cartões.

Este sistema de pagamentos está bem estabelecido no mercado atual, porém para alguns lojistas, em determinadas situações, deixa de ser interessante. Por isso em algumas vendas é oferecido desconto para pagamento com cartão de débito ou em dinheiro. Com a diminuição de intermediários ocorre também a diminuição de taxas, e aumento do lucro. Importante destacar que apesar de o pagamento eletrônico possuir custo neste amplo ciclo de validação operacional, seu uso tem crescido expressivamente no Brasil. Segundo pesquisa do BACEN, somente no ano de 2015 foram realizadas 5,7 bilhões de transações com cartões de crédito [4].

Se nos aprofundarmos nas características das transações, mesmo em um pagamento em espécie existe um intermediário, que é o emitente da cédula, no caso do Brasil o órgão responsável é o BACEN. As taxas de manutenção sobre o dinheiro em espécie estão embutidas nos impostos cobrados pelo governo sobre o sistema financeiro, as instituições financeiras como bancos, administradoras de cartões e máquinas de cartão, e também do lojista. E todo esse custo é repassado no preço final do produto ao comprador, independente do uso de cartão, cheque ou espécie.

Partindo desse ponto de vista, em um cenário onde o pagamento eletrônico é desejável por clientes do ponto de vista de segurança e conforto, porém penoso ao se analisar os custos, surge uma alternativa: se transações forem feitas usando uma moeda que seja naturalmente eletrônica, ou digital, que não é emitida por nenhum governo, seu uso estaria livre de impostos, pois não necessita da intermediação de bancos, emissoras de cartões ou administradoras de máquinas de cartões. Afinal se a transação é eletrônica, por que é preciso usar uma moeda virtualizada por todas as empresas citadas no cenário acima? Não seria mais fácil usar uma moeda nativamente digital? Essa é a alternativa oferecida pela primeira criptomoeda, a bitcoin.

A próxima seção descreve o surgimento da bitcoin. Convém destacar que foram adotadas as denominações das referências utilizadas neste trabalho. “Bitcoin” com inicial maiúscula é o termo usado quando for citado o sistema de pagamento e “bitcoin” com inicial minúscula é usado quando for citada a moeda digital [5].

Surgimento da bitcoin

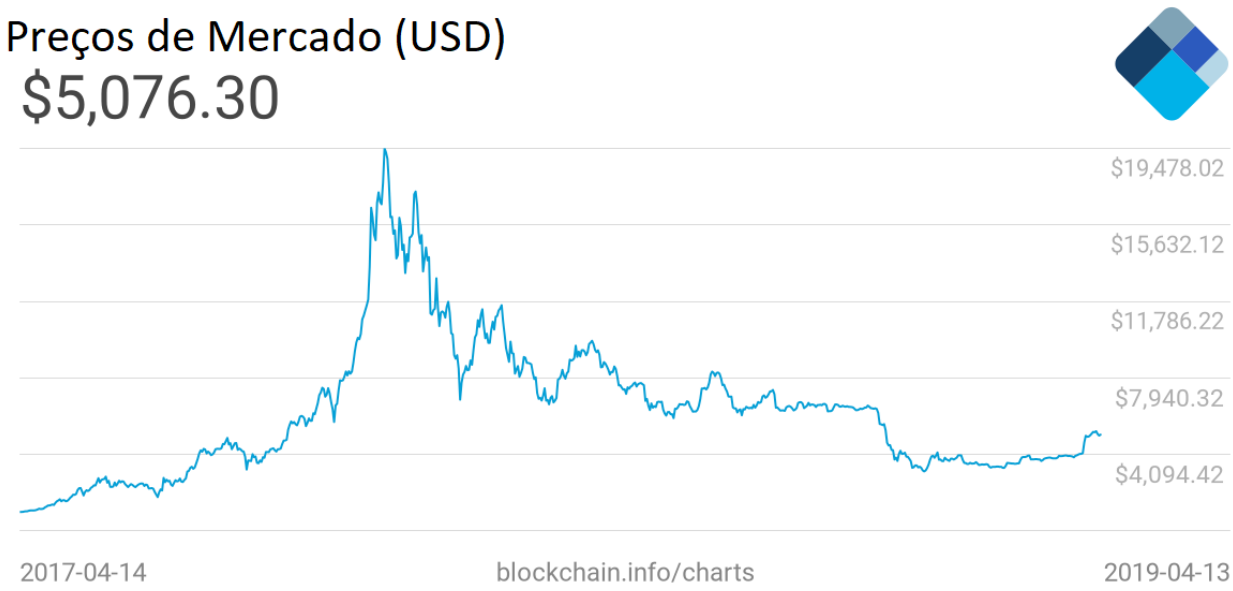
A bitcoin foi apresentada em outubro de 2008 em forma de artigo por Satoshi Nakamoto, que se acredita ser um pseudônimo para mais de uma pessoa. Desde então várias possíveis identidades foram associadas à Satoshi, porém nenhuma com provas definitivas. O empresário australiano Craigh Wright se autodeclarou ser o criador da bitcoin em 2016 e embora tenha recebido apoio em sua revelação na época, também foi desmentido recentemente em 2019 [34]. O empreendedor John McAfee declarou também este ano, 2019, que havia descoberto a identidade de Satoshi, porém não chegou a revelá-la [35]. Muitas teorias foram formuladas sobre a real identidade do criador da moeda, porém até hoje nenhuma foi aceita definitivamente [36].

Antes mesmo do seu lançamento, em agosto de 2008, o domínio bitcoin.org foi registrado junto ao *anonymousspeech.com*, um site que permite o registro de domínios anonimamente. Em novembro de 2008 o projeto Bitcoin foi registrado no Source Forge, um *website* de colaboração de comunidade focado no desenvolvimento de software livre. Esse ato viabilizou o estudo da rede do software Bitcoin. Através deste estudo foi possível entender, alterar e aprimorar a ideia das criptomoedas, o que proporcionou o surgimento das *Alternate Cryptocurrencies* (Altcoins) que, inserindo modificações no código fonte da Bitcoin alteraram algumas características da rede e adicionaram novas funcionalidades. Atualmente existem mais de 700 moedas digitais alternativas à bitcoin.

Apesar de seu lançamento ter sido em 2008, a primeira transação feita com a moeda ocorreu em 12 de janeiro de 2009, entre Satoshi Nakamoto e Hal Finney, programador e ativista criptográfico. A primeira taxa de câmbio estabelecida para bitcoin foi feita pela *New Liberty Standard* em 5 de outubro de 2009. O valor de 1.309,03 BTC (mil trezentos e nove bitcoins) equivalia a 1 USD (um dólar americano) [6].

Desde então o valor da moeda vem sofrendo constantes variações conforme exemplificado pela figura 2 abaixo.

Figura 2 – Cotação da moeda bitcoin no período de abr/2017 a abr/2019.



Fonte:Blockchain – Most Trusted Crypto Company [7].

O mercado da bitcoin nunca foi estável, porém sua variação no ano de 2017 foi a maior e mais significativa até o presente momento. Pelo gráfico podemos observar que o valor de 1 BTC chegou a marca de 19.479,50 USD, sendo este considerado seu maior preço até então. O valor atual da bitcoin está cotado em aproximadamente 5.076,30 USD, em abril de 2019, segundo o site blockchain.com.

O que é bitcoin

A bitcoin é uma criptomoeda ou simplesmente uma forma de pagamento virtual, que não dispõe de uma autoridade central que a regule.

Segundo Ulrich

> "Bitcoin é uma moeda digital *peer-to-peer* (par a par ou, simplesmente, de ponto a ponto), de código aberto, que não depende de uma autoridade central. Entre muitas outras coisas, o que faz o Bitcoin ser único é o fato de ele ser o primeiro sistema de pagamentos global totalmente descentralizado" [8].

A arquitetura de redes *peer-to-peer* funciona sem um servidor central, onde cada nó da rede pode operar como servidor e cliente. De acordo com Regalado

> "A expressão *peer-to-peer* (ponto-a-ponto) é uma arquitetura de redes de computadores onde cada um dos pontos ou nós da rede funciona tanto como cliente quanto como servidor, permitindo compartilhamentos de serviços e dados sem a necessidade de um servidor central" [9].

Criptomoeda é o termo usado para definir moeda virtual, que usa criptografia para impedir que ela seja falsificada. O termo é utilizado não só para a bitcoin, mas para todas as moedas virtuais, tendo em vista que todas se baseiam na mesma tecnologia de criptografia ou em uma de suas alterações.

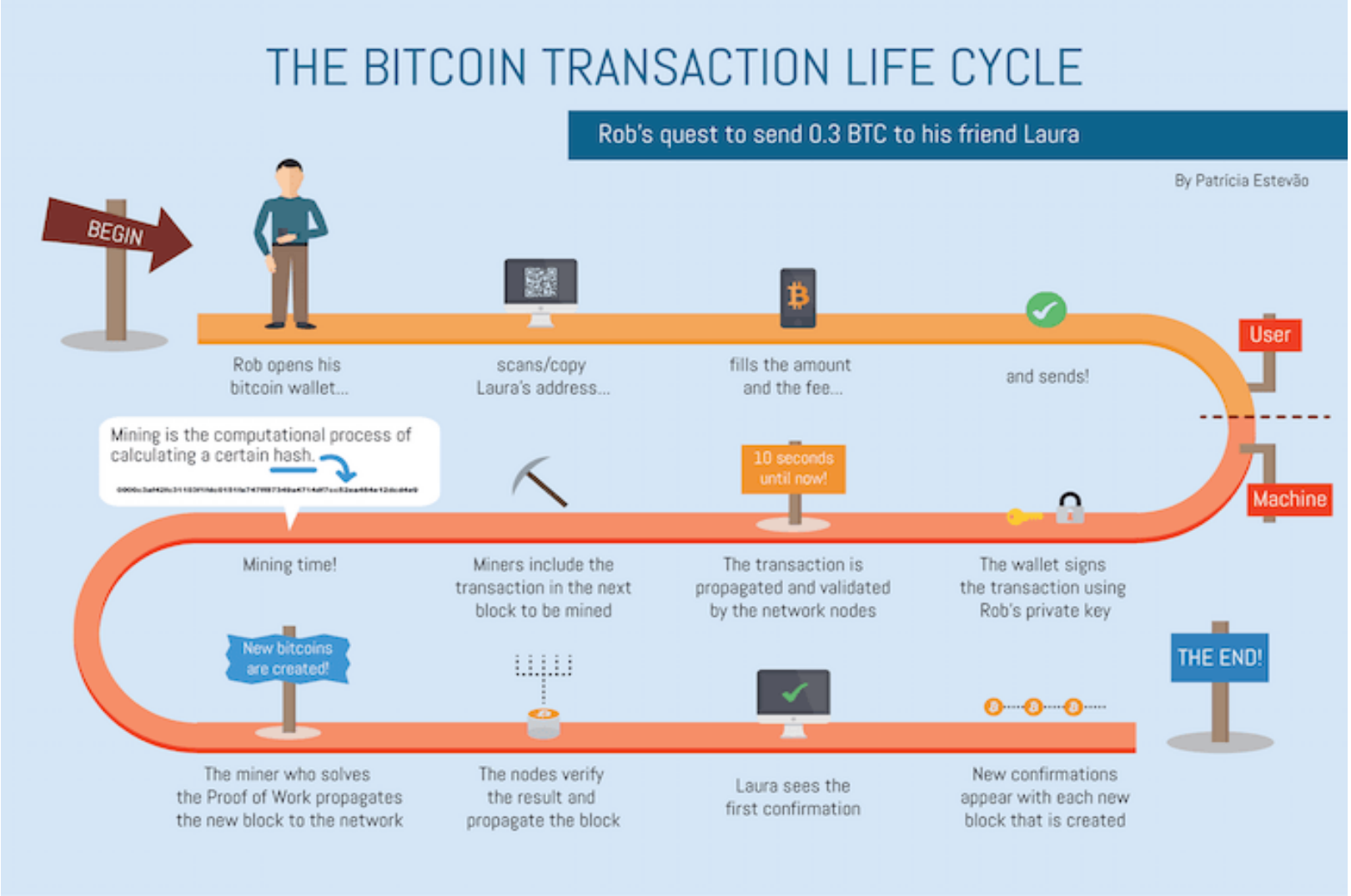
A proposta da bitcoin em ser uma forma de pagamento virtual não é original. Temos serviços como PayPal que executam essa tarefa sem usar o dinheiro físico. Neste serviço o usuário pode ter uma conta para receber ou realizar pagamentos *online* e adicionar saldo ou outras formas de pagamento. Também é possível transferir o saldo disponível na conta para uma conta bancária tradicional e assim efetuar o saque em dinheiro. Como descrito por Ulrich[8], o PayPal exerce o papel de

intermediário registrando as transações entre as partes e impedindo que alguém tente uma transferência sem possuir saldo, ou ainda usar o mesmo saldo para mais de uma transação. Percebe-se que este intermediário fornece a confiabilidade do sistema eletrônico de pagamentos. Se as partes têm segurança de que o valor transferido não será perdido, desviado e nem duplicado, o mesmo torna-se viável. Porém, serviços como PayPal utilizam moedas não virtuais como o dólar e o real e são regulados pelos governos emitentes dessas moedas. Esse tipo de regulação é dispensado pelo sistema Bitcoin, já que a rede é responsável pela validação das transações e exerce o papel de intermediário. Ninguém controla a rede, porém cada usuário do sistema colabora para a autoridade e autenticidade do sistema em si. Desse modo, diferentemente do PayPal, o sistema eletrônico de pagamentos Bitcoin utiliza sua própria moeda. A rede Bitcoin fornece um outro tipo de confiabilidade, baseado em sua estrutura de rede e algoritmos criptográficos, que serão descritos na seção de mineração. A próxima seção descreve o ciclo de vida de uma transação usando a moeda bitcoin.

O ciclo de vida de uma transação com bitcoin

A figura 3 ilustra um exemplo de transação em bitcoin, sendo que cada etapa é executada por alguma funcionalidade do sistema. Essa figura mostra as interações entre as duas ou mais partes envolvidas em uma transação. Nela há um exemplo de uma a pessoa, denominada Rob, enviando 0.3 bitcoin para outra pessoa (Laura).

Figura 3 – Ciclo de vida de uma transação com bitcoin.



Fonte: Medium, Sanjeet Sahay [20].

Observando-se a figura 3, pode-se destacar que o início da transação é de iniciativa do usuário Rob, abrindo sua **carteira bitcoin**, ou *bitcoin wallet*. Uma carteira bitcoin é um software usado para enviar e receber valores em bitcoin. Em seguida nota-se que para enviar o determinado valor é necessário o endereço bitcoin de Laura, que deve ser fornecido à carteira de Rob como um destinatário de um e-mail. Rob então preenche os valores de envio e de taxa da transação. Os valores enviados em bitcoin podem ser fracionados (por exemplo, Rob quer enviar 0.3 bitcoin), enquanto que a taxa da transação pode ter seu valor definido pelo remetente do valor em bitcoin. A taxa pode variar de 0 (zero) a quantas bitcoins o usuário puder pagar, pois é um tipo de "gorjeta" dada a fim de ajudar a transação a ser processada pela rede Bitcoin mais rapidamente.

Definidos os valores de envio e de taxa, o usuário Rob pode então enviar a transação para a rede validar. Isso seria como confirmar uma transação com cartão após digitar a senha na máquina de cartões do lojista. A carteira de Rob prepara a transação e envia para a rede solicitando sua autenticação. Essa transação enviada é sempre propagada pelos usuários da rede, chamados de nós da rede, após ser verificada por cada nó.

Alguns nós da rede são chamados de mineradores. Estes são os responsáveis por adicionar a transação de Rob e Laura permanentemente ao sistema Bitcoin através do processo de mineração. Basicamente os mineradores montam um bloco, cada, contendo transações pendentes de validação que foram propagadas pelos nós da rede, como a transação de Rob e Laura, e executam um processo computacional para calcular um determinado valor conhecido por *hash*. Cada minerador tenta validar o seu bloco, que pode ser diferente dos outros, contendo mais ou menos transações, ou ordená-las em uma ordem diferente. A validação de uma transação acontece quando algum minerador consegue encontrar o *hash* necessário para validar o bloco que contém a transação, porém assim que ela é propagada pela rede não é possível cancelá-la. Sempre que uma transação for enviada para a rede do sistema, ela será validada. O minerador que conseguir terminar de calcular o *hash* primeiro, consegue validar o bloco e recebe como recompensa novas bitcoins.

Após essa validação pelo minerador, o bloco é propagado pela rede para que outros mineradores verifiquem que ele realmente foi validado. Por meio desse processo, Laura recebe uma confirmação de que os valores em bitcoin enviados por Rob foram registrados permanentemente na rede do Sistema. Um bloco é considerado validado quando após o cálculo do *hash* realizado pelo minerador, for propagado pela rede e adicionado permanentemente à *blockchain*, que é o registro de transações do sistema. A partir deste bloco validado, os mineradores constroem o próximo bloco, buscando novas transações pendentes e tentando calcular o *hash* de um novo bloco.

Podemos notar que a interação entre Rob e Laura é semelhante a uma transação com qualquer tipo de moeda. Para um usuário do sistema Bitcoin, o conhecimento necessário do funcionamento é parecido com o de sistemas de transação bancária, uma vez que a interação dos usuários ocorre por meio da carteira bitcoin, que por sua vez esta interage com a rede que se encarrega do restante. É importante ressaltar que uma vez enviada para a rede, uma transação com bitcoins não pode ser cancelada. Se Rob e Laura decidirem desfazer qualquer tipo de acordo comercial, seja uma compra ou prestação de serviços, Laura deve enviar de volta, em uma nova transação, o valor primeiramente enviado por Rob para que ele seja reembolsado.

Transações

Na figura 3 vimos que os softwares de carteiras são responsáveis pela interação com o usuário do sistema bitcoin. Os detalhes do funcionamento das transações e dos *softwares* de carteiras serão descritos nesta seção.

Além dos algoritmos e protocolos usados na *blockchain*, o sistema Bitcoin possui importantes elementos e funcionalidades implementados em suas transações, afinal trata-se de um sistema online de pagamentos, portanto deve haver formas específicas de movimentar o dinheiro usado no sistema. A *blockchain*, como falado anteriormente, registra todas as transações como um livro contábil, ou *ledger*, como é usado em se tratando de Bitcoin. Portanto uma transação é como uma linha deste livro contábil, com valores de entrada e saída, onde pode haver mais de uma entrada e também mais de uma saída. As entradas dizem de onde sairão as bitcoins a serem transferidas, e as saídas designam quem deve recebê-las.

As entradas, chamadas de *inputs*, são saídas de outras transações já processadas pela rede, e a transação descreve qual o valor será destinado para cada uma das novas saídas, ou *outputs*. Simplificando, cada *input* foi um *output* já processado na *blockchain*, dando direito ao seu receptor de enviá-la total ou parcialmente para novos destinatários. Em nosso exemplo de transação na figura 3, Rob cria um *input* com um *output* ainda não gasto que lhe pertence para elaborar a transação, colocando um *output* para Laura.

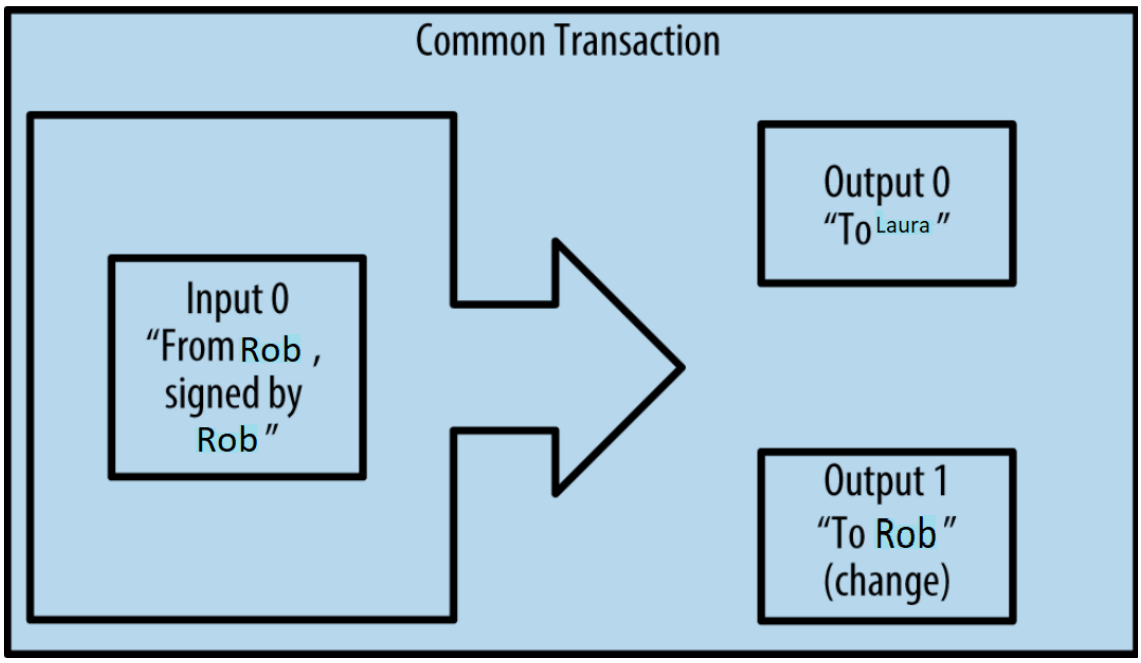
A blockchain é um livro contábil aberto de livre acesso, de modo que é possível rastrear um valor em bitcoin através de todas as contas que o possuíram até a data da sua criação. Importante destacar que esse rastreamento apresenta somente o endereço da carteira bitcoin e nunca a identidade do usuário.

O sistema Bitcoin prevê em sua implementação uma remuneração em moedas bitcoins ao minerador que validar um bloco contendo transações dos usuários. O sistema também prevê um número máximo de moedas bitcoins que serão geradas pelo sistema [15], ou seja, a mineração de moedas novas deve parar um dia. Quando o sistema parar de remunerar os mineradores com moedas novas eles ainda receberão as taxas das transações validadas nos blocos. Por isso é comum que os valores dos *inputs* sejam maiores que dos *outputs.* Essa diferença é a taxa da transação que é a outra forma de remuneração ao minerador que validar o bloco. Quanto maior for esta taxa mais atrativa ela se torna aos mineradores, o que faz com que a transação seja registrada na *blockchain* mais rapidamente, como a gorjeta ao minerador que foi descrito no capítulo anterior. Em nosso exemplo, se Rob quiser enviar 0,03 BTC para Laura e tiver um *output* não gasto em sua carteira no valor de 0,035 BTC, ele pode elaborar a transação para Laura, usando este *output* como *input* da transação, e criando um *output* para Laura no valor de 0,03 BTC, a diferença 0,005 BTC será a gorjeta ao minerador para validar a transação.

Ao falarmos de transações envolvendo a moeda bitcoin é importante destacar que existe mais de um tipo de transação além da tradicional com um remetente e um destinatário. É o caso da transação exemplificada na figura 4, em que temos um *input* e dois *outputs,* sendo um para o destinatário e outro retornando ao remetente, com nome de *change*, o que no caso de dinheiro físico, seria equivalente ao pagamento com retorno de troco. Neste caso temos Rob enviando algum valor para Laura usando um grande *input*, que é

dividido pela transação em dois *outputs* menores, um para Laura e o outro retornando para Rob.

Figura 4 – A forma mais comum de transação.



Fonte: Adaptado de Mastering Bitcoin: unlocking digital cryptocurrencies – Andreas Antonopoulos [15].

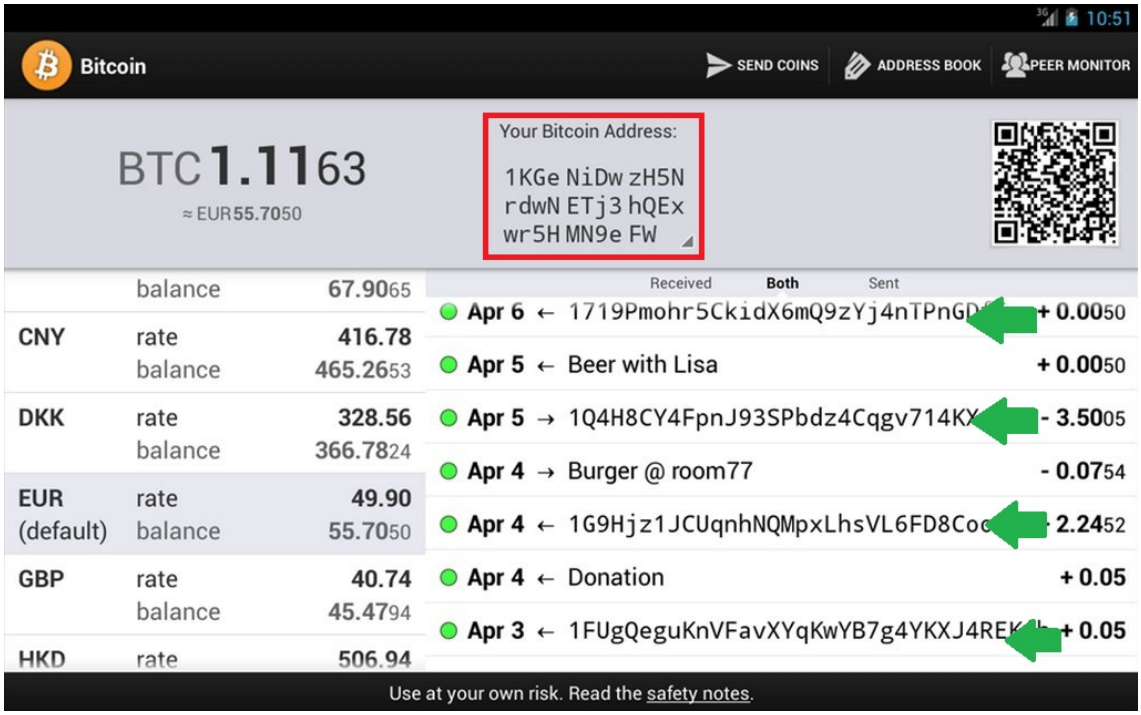
Outro tipo comum de transação é possuir vários *inputs* sendo agregados em um único *output*, que no mundo não virtual seria o equivalente a transformar vários trocados e moedas em uma nota graúda. Este tipo de transação é comum nos aplicativos de carteiras de bitcoins, quando o mesmo agrupa várias entradas para melhor organizar e apresentar o saldo ao usuário [15].

Na figura 3, pode-se notar que Rob utiliza sua carteira para enviar bitcoins para Laura. As carteiras de bitcoins são softwares usados no envio e recebimento de bitcoins, que criam, organizam e enviam as transações para a rede solicitando validação pelos mineradores. Uma carteira tem seu funcionamento muito parecido com os *e-mails*. Para enviar ou receber *e-mails* é necessário um endereço de *e-mail* em algum servidor. O usuário deste servidor fornece seu endereço de *e-mail* para as pessoas, sites ou empresas de quem deseja receber *e-mails*, funcionando como uma chave pública de sua identificação online. Porém, possuir o endereço de *e-mail* de alguém não permite usar este endereço para mandar *e-mails* para outras pessoas, pois são protegidos por senha. Esta senha é a chave privada usada para autenticar o usuário detentor do endereço, e assim assegurando que os *e-mails* enviados deste endereço sejam realmente autorizados pelo usuário. Para enviar *e-mails* para outras pessoas, este usuário precisa dos endereços, ou chaves públicas, dos destinatários.

As carteiras bitcoin funcionam de modo semelhante. Elas geram a chave pública usada para receber bitcoins através de transações e permitem ao usuário cadastrar uma chave privada associada a sua chave pública, permitindo enviar transações para outras carteiras. A figura 5 mostra uma carteira bitcoin com destaque no quadrado vermelho mostrando o endereço da carteira, divulgado para receber saldos, e outros endereços

destacados com as setas verdes, mostrando os recebimentos e envios de saldos para e de outras carteiras bitcoins.

Figura 5 – Exemplo de carteira bitcoin.



Fonte: Malavida, aplicativos para seu Android [30].

A segurança da chave privada de uma carteira é de suma importância, pois as transações em bitcoin não podem ser canceladas. Uma vez enviadas para alguém, as bitcoins só podem ser devolvidas ao remetente se o destinatário assim quiser. Da mesma maneira, quando alguém recebe bitcoins em seu endereço público, essas moedas lhe pertencem e só podem ser repassadas com uso de sua chave privada [16].

Existem vários tipos de carteiras bitcoin, e nenhuma delas armazena as moedas em si, como funciona com as carteiras do mundo real com cédulas e moedas. O que se armazena nelas são as chaves privadas que dão acesso ao envio e consulta de saldos de bitcoins em seu endereço público de bitcoins. As moedas, pode se dizer, são armazenadas na própria rede, nas transações registradas na *blockchain*. Alguns exemplos de carteiras incluem os softwares de carteira, que são instalados no computador pessoal. Essas carteiras dão a segurança do usuário possuir e controlar suas chaves privadas, porém também as expõe aos riscos que todo computador pode correr, como vírus, perda do mesmo ou defeito de *hardware*. Se o computador for corrompido ou perdido e não houver nenhum *backup*, as chaves privadas salvas no software também podem ser perdidas.

A carteira de software original é o bitcoin core, programa original que executa a rede bitcoin. Para executar o bitcoin core como carteira também é necessário baixar toda a *blockchain*, e isso se torna um tanto inviável, pois atualmente, maio de 2019, o livro razão da *blockchain* ocuparia um total de 210 gigabytes. O tipo mais comum de carteira, chamado de carteiras leves, ou SPV (*Simplified Payment Verification*, que significa Verificação Simplificada de Pagamentos) não baixam toda a *blockchain*, apenas

sincronizam com o bloco mais atual. O bloco mais atual representa o “saldo atual” da pessoa, como é visto nas contas-correntes de bancos tradicionais. Não é necessário ter todo o histórico do saldo que se pretende receber ou enviar, apenas sua última posição e valor. Com o endereço do *output* vinculado ao *hash* do seu bloco, é possível ligá-lo até o bloco mais atual da *blockchain* através dos *hashes* dos cabeçalhos que sempre apontam para o bloco pai, comprovando que a transação que a carteira alega pertencer ao seu dono está de fato na *blockchain* e pode ser enviada para outra pessoa.

Existem também carteiras online, baseadas na computação em nuvens, que oferecem mais praticidade, pois podem ser acessadas nas páginas dos servidores de carteira e através de aplicativos em celulares. No entanto, as chaves privadas são guardadas online pelo servidor, e a segurança depende deles, e não do possuidor das bitcoins. Além disso, por ser um serviço online, pode ficar sem funcionamento por problemas no servidor impedindo que as bitcoins sejam gastas. Porém, desde que haja um backup das chaves privadas, mesmo que o serviço online seja terminado, o usuário pode acessar as bitcoins através de outro software de carteira normalmente.

As carteiras *mobile*, são aplicativos de celular usados para controlar as transações, e são muito práticos para transações usando bitcoin. A figura 5 acima é um exemplo de carteira *mobile*. A maioria das carteiras *online* e *desktop* possuem versões *mobile* e algumas foram especialmente desenvolvidas para celulares.

Carteiras de *hardware* são pequenos dispositivos que ficam desconectados da Internet na maior parte do tempo, o que aumenta a segurança já que não estão expostos a vírus ou invasões como outros tipos de *software* que ficam conectados o tempo todo (Figura 6). Essas carteiras são conectadas apenas para enviar e receber transações. Por ser um *hardware* pequeno, corre o risco de ser roubado ou perdido. Nessas situações, as criptomoedas nele armazenadas serão perdidas.

Figura 6 – Exemplo de carteira de *hardware*.

Fonte: Buy Bitcoin Worldwide [31].

Por fim, existem as carteiras de papel, a carteira mais simples existente. São pedaços de papel onde a chave privada e a chave pública são impressas (Figura 7), comumente em QR codes, e são ideais para armazenamento em um longo período de tempo, desde que mantidos em local seguro [17].

Figura 7 – Exemplo de carteira impressa.

Fonte: BitcoinPaperWallet [32].

Diferentemente de e-*mails*, as chaves privadas de uma carteira possuem uma forte relação com as chaves públicas. No caso de e-*mails* geralmente um endereço de *e-mail* de uma pessoa não tem relação com a senha escolhida para protegê-lo. No caso do sistema Bitcoin, as chaves públicas derivam matematicamente da chave privada que é escolhida ao acaso geralmente pelo software da carteira ou pelo usuário. Em bitcoin são usadas multiplicações de curvas elípticas na geração de chaves públicas a partir de chaves privadas escolhidas aleatoriamente. Antonopoulos (2014) define:

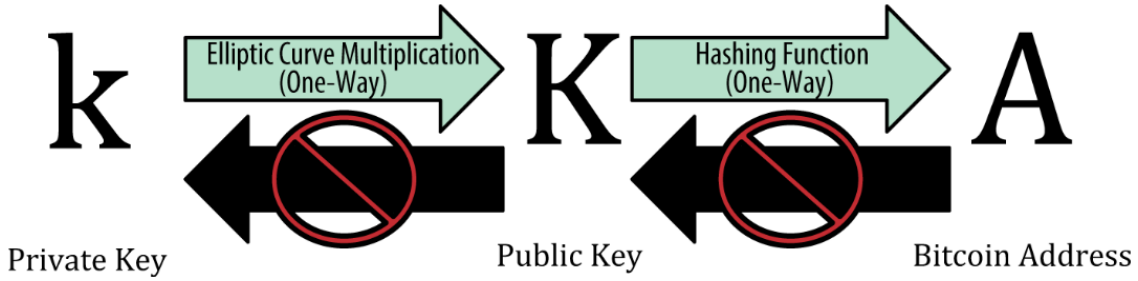
> "A criptografia de curva elíptica é um tipo de criptografia de chave pública ou assimétrica baseada em um problema logarítmico discreto que é expressado pela adição e multiplicação nos pontos de uma curva elíptica. (...) O Bitcoin usa uma curva elíptica específica e um conjunto de constantes matemáticas, como definido em um padrão chamado secp256k1, estabelecido pelo Instituto Nacional de Padronização e Tecnologia (NIST)" [15].

Como existe uma relação matemática entre a chave pública e privada, mesmo não se divulgando a chave privada, é possível comprovar que a mesma foi usada para assinar uma transação a partir daquela chave pública, tornando a transação válida. Portanto, para se gastar as bitcoins, o dono delas apresentará a chave pública que recebeu bitcoins previamente e uma assinatura que será sempre diferente, porém sempre baseada na chave privada.

Quando se estabelece um *output* de transação é informado um endereço bitcoin, que é uma *string* de letras e números que começam com o dígito 1. Isto pode ser observado na figura 5 com o exemplo de carteira, onde os endereços de carteiras mostrados começam com o dígito 1. O endereço de bitcoin se obtém aplicando os algoritmos de *hash* criptográfico SHA256 e depois RIPEMD160 sobre a chave pública gerada previamente. RIPEMD160 é um algoritmo de *hash* criptográfico assim como o SHA256, porém seu resultado é um *hash* de 160 bits. Ao se aplicar a função *hash* RIPEMD160 têm-se como resultado um número de 160 bits, que é o endereço a ser usado na transação bitcoin.

A Figura 8 abaixo mostra a relação entre chave privada, chave pública e endereço bitcoin. Nela há três elementos principais: k representando a chave privada, K representando a chave pública e A que, por fim, é o endereço bitcoin gerado. Após aplicar a multiplicação de curva elíptica na chave privada k, obtém-se a chave pública K. E ao aplicar o algoritmo de *hash* criptográfico RIPEMD160 sobre K, finalmente obtém-se o endereço bitcoin A. Note-se que nem a multiplicação de curva elíptica e nem a função *hash* são reversíveis, implicando na segurança do endereço bitcoin gerado.

Figura 8 – Relação entre chave privada (private key, k), chave pública (public key, K), e endereço bitcoin (bitcoin Address, A).

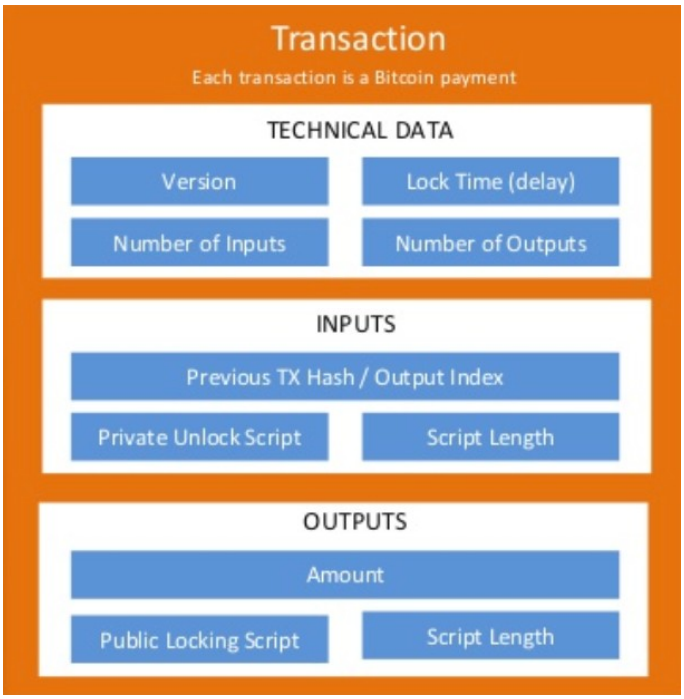


Fonte: Mastering Bitcoin: unlocking digital cryptocurrencies - Andreas Antonopoulos [15].

Como um exemplo, podemos dizer que um usuário do sistema Bitcoin, Rob, possui uma carteira que produziu um endereço válido A e que recebeu uma certa quantia em bitcoin. O sistema permite a fração de até 0,00000001 (10 elevado a -8). Esta fração é chamada de 1 satoshi. Ao receber um valor em seu endereço bitcoin, Rob passa a possuir uma UTXO (*Unspent Transaction Output*), que significa *output* de transação não gasta. Uma UTXO pertencente a Rob só poderá ser gasta usando a chave privada de Rob associada ao endereço bitcoin que recebeu esse valor, isto é, a chave pública de Rob. Porém, Rob pode fornecer essa autorização a outra carteira, de outra pessoa ou dele mesmo, para que essa UTXO seja gasta.

No sistema Bitcoin existem várias UTXO sendo direcionadas para várias outras de diferentes posses. Os *inputs* de uma transação são UTXO devidamente assinadas por seus possuidores. Os *outputs* são novas UTXO que ficam registradas na *blockchain*. Uma UTXO tem seu valor fixo e só pode ser fracionada por meio de seu gasto em transações, por isso a forma mais comum de transação, como explicado anteriormente, é a que gera um troco de volta ao remetente [15]. A figura 9 demonstra a estrutura de uma transação em bitcoin mostrando todos os campos que devem ser preenchidos. Na parte superior da estrutura temos o *Tecnhnical Data*, dados técnicos listando a versão, o *locktime*, que é um bloqueio temporal para a validação da transação e campos para o número de *inputs* e número de *outputs*. É possível montar transações com mais de um *input*, e direcionar os valores para mais de um *output*.

Figura 9 – Estrutura de uma transação bitcoin.



Fonte: Anatomy of a blockchain [33].

A tabela I mostra o detalhamento do campo *output* das transações feitas em bitcoin, que é usado para gerar *inputs* de novas transações e surge novamente a cada transação feita pelas carteiras. O *script* especificado na tabela I é o código que trava essa *output* para um endereço bitcoin especifico. Assim somente o dono deste endereço bitcoin poderá gastar este *output.*

TABELA I – ESTRUTURA DE UM OUTPUT DE TRANSAÇÃO

Tamanho	**Campo**	**Descrição**
8 bytes	Quantia	Quantia em bitcoin designada em satoshis
1-9 bytes	*Locking-Script Size*	Comprimento do *script* em bytes
Variável	*Locking-Script*	Um *script* definindo as condições necessárias para gastar o *output*

Fonte: Antonopoulos, A. M. (2014) [15].

A tabela II mostra a estrutura de um *input* de transação, este campo da transação representa o gasto da moeda de fato, que é o ato de usar uma UTXO para gerar um *output*.

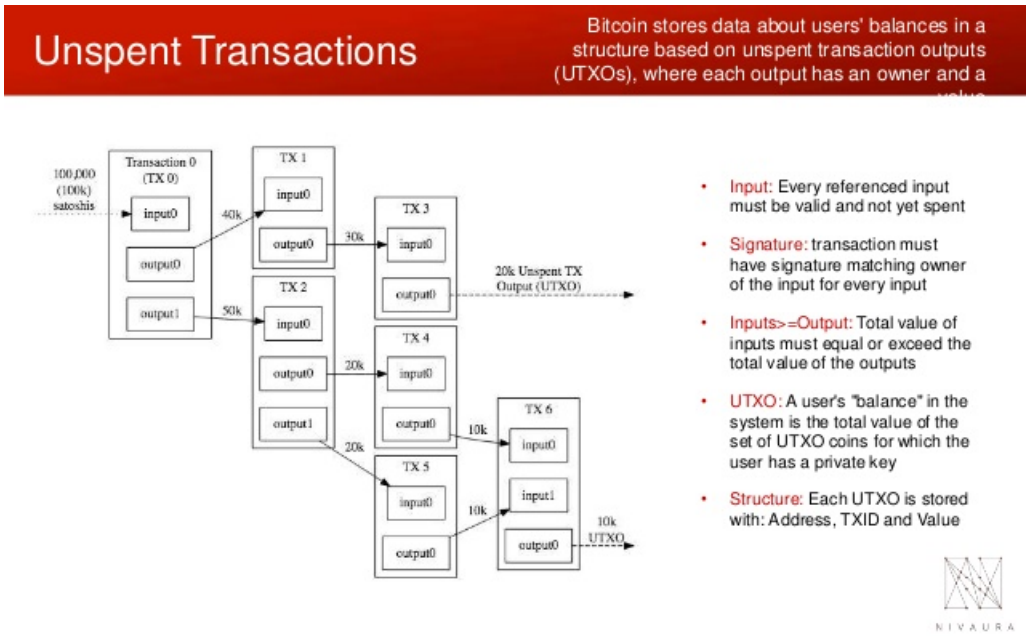
TABELA II – ESTRUTURA DE UM INPUT DE TRANSAÇÃO

Tamanho	**Campo**	**Descrição**
32 bytes	Transaction Hash	Apontador para a transação contendo o UTXO para ser gasto
4 bytes	Output Index	O número índice do UTXO para ser gasto
1-9 bytes	Unlocking-Script Size	Comprimento do script de destravamento em bytes
Variável	Unlocking-Script	Um script que preenche os critérios para o script de travamento do UTXO
4 bytes	Sequence Number	Funcionalidade de substituição de transação, atualmente desabilitada.

Fonte: Antonopoulos, A. M. (2014) [15].

Após a formação da transação, a carteira de Rob, como em nosso exemplo, envia para a rede bitcoin a transação para que seja validada. Cada nó na rede irá propagar para os nós adjacentes esta transação até que chegue em um nó minerador (ver Figura 3). Este, assim como todos os nós que propagaram a transação pela rede, irá também verificar se a transação de Rob é válida. Como os inputs são formados por UTXO e contém um *hash* apontando para a transação que recebeu esse valor a ser gasto, a validação acontece buscando na *blockchain* por esse UTXO e confirmando-se que ele não foi gasto anteriormente, prevenindo assim também o gasto duplo. A figura 10 mostra as ligações entre *input* e *output* que são verificadas pelos nós da rede.

Figura 10 – Exemplo de UTXO em transações bitcoin.



Fonte: Anatomy of a blockchain [33].

Sistema Bitcoin

A figura 3 mostrou que as carteiras se encarregam da interação entre os usuários do sistema Bitcoin que desejam enviar ou receber moedas. Esta seção descreve como esse sistema lida com as transações após as carteiras enviarem pedidos de validação pela rede, ou seja, pedidos de registro na *blockchain*. Também serão detalhadas aqui as estruturas de dados usadas pelo sistema Bitcoin.

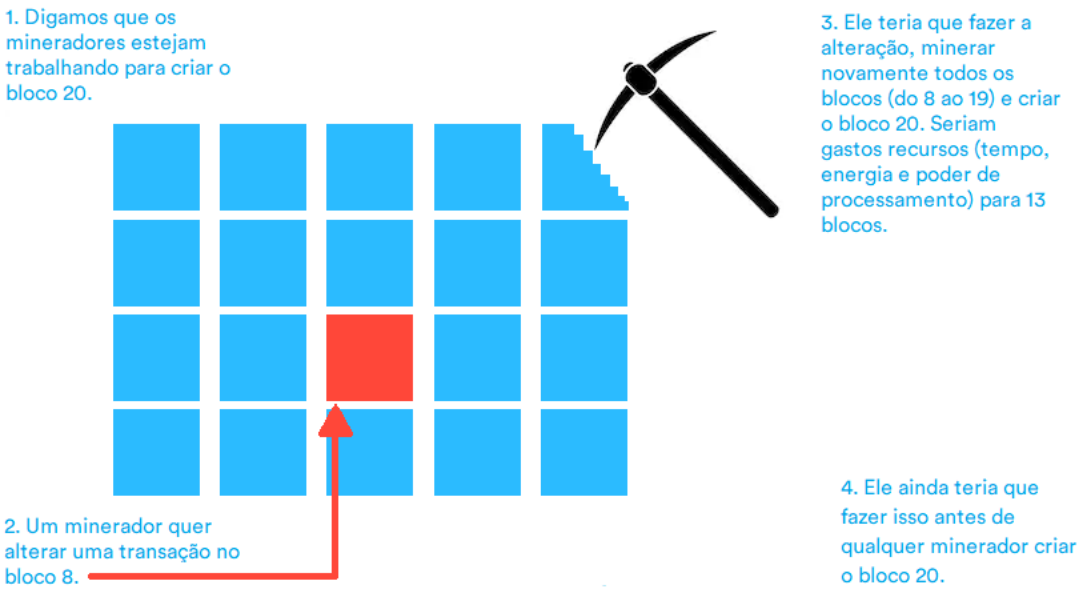
Blockchain, termo em inglês que significa cadeia de blocos, é uma das estruturas principais do sistema Bitcoin, assim como de todas as criptomoedas. Funciona como um histórico de todas as transações feitas no sistema (envio ou recebimento de saldos) e pode ser consultado livremente por qualquer usuário do sistema.

Cada nó da rede que age como minerador, executa algoritmos para a validação do bloco atual e, portanto, das transações que estão sendo armazenadas nele [10]. Uma vez validado, o bloco é adicionado a *blockchain* permanentemente, e espalhado para os outros nós da rede, assim todos os membros do sistema Bitcoin podem possuir uma cópia da *blockchain*, ou seja, dos blocos com as transações validadas pelos mineradores.

Todos os blocos da cadeia tem a mesma estrutura, com o identificador do bloco anterior funcionando como um ponteiro para o bloco pai. Este ponteiro é um *hash* do bloco, e é usado na criação de cada bloco novo, o que garante a propriedade de cadeia contínua [10]. Por exemplo, numa sequência de 3 blocos, o bloco 2 usa o *hash* do bloco 1 em sua criação. E o *hash* do bloco 2 contém o *hash* do bloco 1. Na criação bloco 3 será usado o *hash* do bloco 2, e isso gerará automaticamente uma relação de continuidade com o bloco 1. Para alterar algum bloco na sequência da *blockchain*, seria necessário alterar todos os blocos subsequentes, pois uma simples alteração mudaria totalmente o *hash* do bloco. Como o cálculo do *hash* leva cerca de 10 minutos, para conseguir fraudar um bloco, adicionando uma transação inválida, o fraudador teria que fazer a alteração no

bloco desejado na cadeia, calcular seu *hash*, e calcular todos os *hashes* dos blocos posteriores, como é mostrado na figura 11.

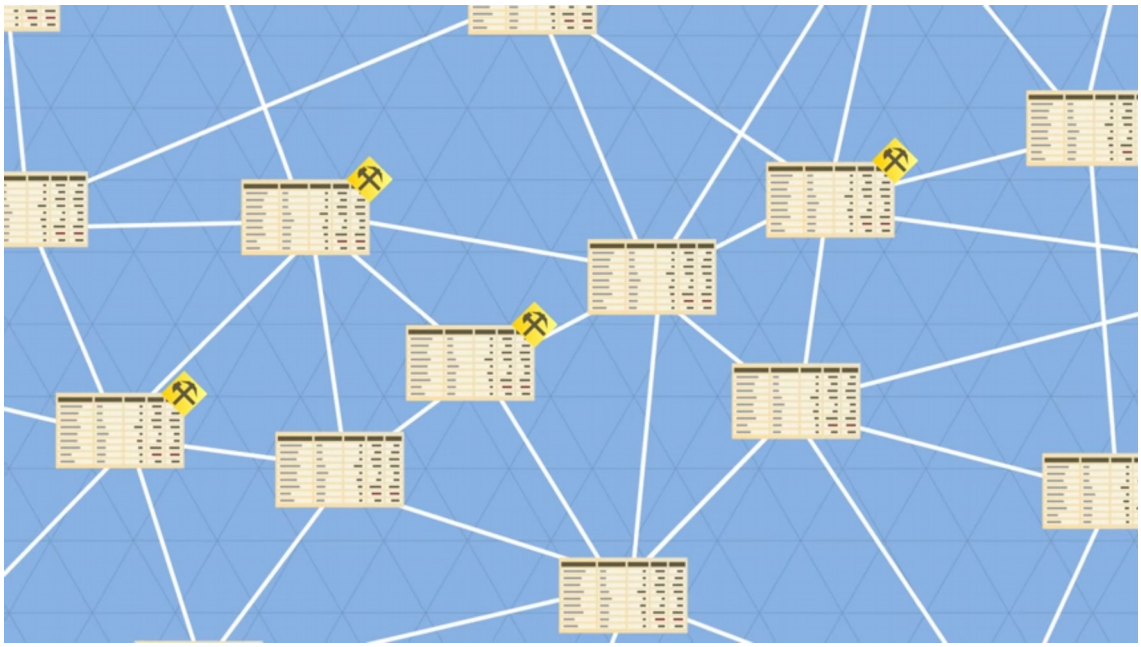
Figura 11 – Segurança da *blockchain*.



Fonte: Modiax – O que é minerar bitcoin [28].

Nesse sistema, existe, portanto, uma estrutura de dados, o bloco, responsável por armazenar os dados das transações feitas com bitcoin, organizados em outra estrutura de dados, a cadeia de blocos, ou *blockchain,* compondo uma das principais partes do sistema Bitcoin. A figura 12 mostra uma representação da *blockchain* como vários nós conectados espalhados pela rede com os registros das transações feitas no sistema Bitcoin. Cada nó representa um usuário do sistema que mantém o registro das transações feitas desde o início até o momento atual da rede. Os nós com o símbolo de picareta representam os mineradores, que estão adicionando novas transações à *blockchain.* Os outros nós são os usuários do sistema que estão enviando transações para validação dos mineradores. Todos eles propagam as atualizações feitas na cadeia.

Figura 12 – Estrutura da *blockchain*.



Fonte: The real value of bitcoin and crypto currency technology – The Blockchain explained [21].

Existe uma aplicação chamada *sidechain*, que se baseia na *blockchain* com o intuito de adicionar outras funcionalidades à cadeia. A ideia central da *sidechain* seria interligar outras *blockchains*, ou outros ativos a cadeia original. Seus usuários seriam capazes de enviar bitcoins através de transações na *sidechain* e convertê-las em outras criptomoedas ou qualquer ativo relacionado a uma outra *blockchain*, e ainda assim prevenir o gasto duplo nas duas cadeias [37]. Apesar de ter sido publicada em artigo, assim como a bitcoin, a *sidechain* está sendo estudada sem ter tido implementações até junho de 2019, porém com estudos focados no aperfeiçoamento da *blockchain* original da bitcoin [38].

Para melhor detalhamento dessas estruturas e, em consequência, melhor entendimento do funcionamento do sistema, a próxima seção apresenta conceitos computacionais utilizados na construção de blocos e da *blockchain*.

O Algoritmo criptográfico da *blockchain*

Dentre os conceitos fundamentais de criptografia está o uso de pares de chaves, sendo uma pública e uma privada, para o envio de mensagens de forma segura. Basicamente a chave pública é divulgada, enquanto que a chave privada é usada pelo destinatário no recebimento da mensagem. O procedimento simplificado baseia-se no remetente usando a chave pública do destinatário para encriptar a mensagem e enviá-la. Somente o destinatário possuidor da chave privada correspondente é capaz de desencriptar a mensagem [10].

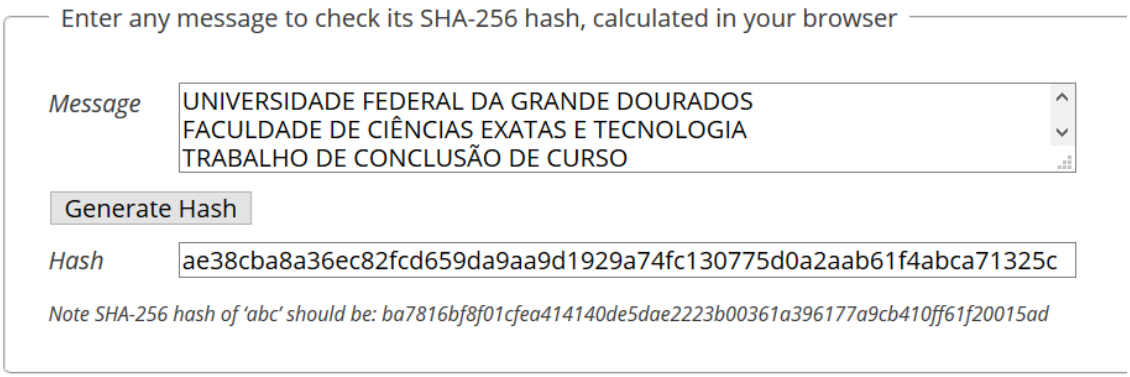
Outro conceito relevante da criptografia moderna é o de *hash*. A função de *hash* recebe uma mensagem de qualquer tamanho e devolve um *hash* criptográfico sempre do mesmo tamanho, como um resumo criptográfico da mensagem original. Os algoritmos de *hash* não possuem uma função de desencriptação, ou seja, não possuem uma chave para decifrar a mensagem. Sua utilidade maior está na função de autenticação da

mensagem, visto que deve ser praticamente impossível gerar o mesmo *hash* para mensagens diferentes, mesmo que tenha se alterado apenas um bit na mensagem original, a função *hash* deverá gerar um resultado totalmente diferente [8].

O algoritmo SHA256, que é um algoritmo de função de *hash* criptográfico, é um dos componentes mais importantes da *blockchain* e, portanto, do sistema Bitcoin inteiro. O Anexo A deste trabalho apresenta o código fonte da implementação desse algoritmo em Javascript. As funções de *hash* se diferem de algoritmos criptográficos comuns, os quais usam chaves para cifrar e decifrar mensagens, pois as funções de *hash* possuem a característica da unidirecionalidade. Uma vez calculado o *hash* de uma palavra ou frase, torna-se praticamente impossível encontrar a mensagem original [10]. Essa característica fornece a propriedade de autenticidade à mensagem, pois, sabendo-se que se apenas um bit for alterado, o algoritmo irá produzir um *hash* totalmente diferente. E mesmo conhecendo a função *hash* usada para cifrar a mensagem, só é possível encontrá-la através de força bruta, testando várias mensagens com a função *hash* e comparando seus resultados com o desejado.

Além disso, a função de cifragem é computacionalmente rápida e produz um *hash* de tamanho fixo, independente do tamanho da mensagem, no caso do SHA256 o tamanho é de 256 bits. A figura 13 abaixo mostra um exemplo de mensagem que foi criptografada pelo algoritmo SHA256. Na caixa *Message* foi inserido todo o texto deste trabalho.

Figura 13 – Exemplo de mensagem criptografada pelo SHA256. A mensagem criptografada neste exemplo é todo o texto deste trabalho.



Fonte: Movable Type Scripts [18].

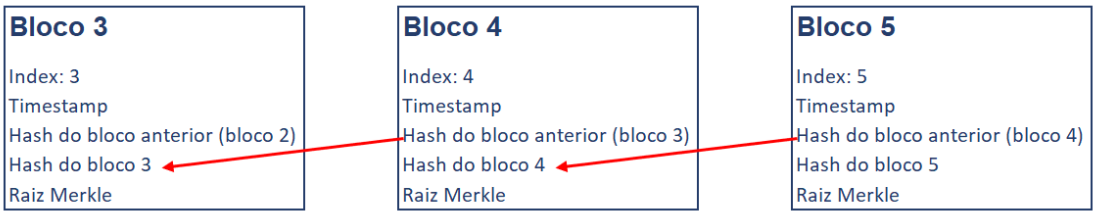
Com estas propriedades o *hash* resultante da função funciona como uma impressão digital de uma pessoa, algo que identifica unicamente a mensagem criptografada e prova assim sua autenticidade.

O funcionamento diferenciado dos algoritmos de dispersão, como também são conhecidos os algoritmos de *hash* criptográfico, são adequados a uma importante característica da *blockchain*, a prova de trabalho, em inglês, *proof of work* [12]. O SHA256 é um algoritmo da família SHA-2, que é sucessora do SHA-1. Em seus procedimentos são usadas funções de compressão, o que garante que uma mensagem grande seja diminuída para o hash do tamanho desejado. Basicamente a mensagem original é dividida em palavras de 32 bits e então se aplicam as funções lógicas para compressão.

Em SHA256 são usadas 6 funções lógicas, que são MAJ, CH, e quatro funções sigma, sendo duas sigma de deslocamento e duas sigma de rotação. As funções MAJ e CH tem 3 bits de entrada e 1 bit de saída, portanto fazem a compressão dos bits. As funções sigma usam operações de rotação e moção (deslocamento) dos bits, para embaralhar uma palavra. Essas funções são executadas sucessivamente até que a mensagem original tenha sido totalmente processada pelo algoritmo, e então temos o seu *hash* criptográfico [13].

Um bloco no sistema Bitcoin é a estrutura básica da *blockchain*. A *blockchain* por si é uma estrutura de dados composta por esses blocos, que contém um ponteiro para o bloco anterior, ou bloco pai, o que cria a característica de cadeia dos blocos. Com exceção do bloco gênese, que é o primeiro bloco da *blockchain,* todos os blocos devem apontar para o seu bloco pai. Este ponteiro, como dito anteriormente, é um *hash* do bloco anterior. Cada bloco é identificado pelo seu *hash*, e seu *hash* é criado a partir de suas informações de identificação localizadas no cabeçalho do bloco, sendo que uma delas é o *hash* do bloco anterior. Os *hashes* dos blocos são o elo entre os blocos que geram a propriedade de corrente. A figura 14 mostra a ligação entre os blocos na *blockchain* através dos seus *hashes*.

Figura 14 – Ligação entre os blocos na *blockchain* através dos *hashes*.



Fonte: Modiax – O que é minerar bitcoin [28].

A tabela III a seguir, apresenta detalhadamente a estrutura de um bloco, com cada campo detalhado e seu tamanho em bytes. A estrutura do bloco em si é simples, porém o campo cabeçalho carrega detalhes importantes, e é onde ocorre o elo com a *blockchain*. Ainda na estrutura do bloco vemos um campo destinado exclusivamente para a listagem das transações que foram validadas nele. Este campo representa o papel do bloco como registro de livro contábil, que pode ser consultado como visto anteriormente.

TABELA III – ESTRUTURA DO BLOCO

Tamanho	Campo	Descrição
4 bytes	Tamanho do bloco	Tamanho do bloco, medido em bytes.
80 bytes	Cabeçalho	Informações relacionadas ao processo de mineração do bloco, permitindo sua

		identificação.
1 a 9 bytes	Contador de transações	Número de transações existentes no bloco.
Variável	Transações	Transações armazenadas no bloco.

Fonte: Rodrigues, C. K. S. (2018) [10]

Cada bloco armazena um certo número de transações, identificador do bloco anterior, seu nível de dificuldade de validação e o instante de criação, que é um carimbo de tempo, ou *timestamp*. Dentre os campos citados na estrutura do bloco na tabela III, o campo cabeçalho merece um importante destaque. A tabela IV apresenta o detalhamento do cabeçalho de um bloco. No cabeçalho do bloco é armazenado o ponteiro para o seu bloco pai no formato de *hash*, que é o elemento que cria o vínculo com a *blockchain*. Outro campo importante no cabeçalho é o resumo das transações que estão armazenadas no bloco. O resumo é feito numa estrutura de dados chamada árvore de *Merkle*, e no cabeçalho é armazenada apenas a raiz da árvore, que também é um *hash*. Finalmente, o campo mais importante do bloco todo também está presente no cabeçalho, o *nonce*. Quando um bloco é criado por um minerador é necessário calcular u*m hash* que satisfaça a dificuldade imposta pela rede. Como o *hash* muda completamente com a alteração de apenas um bit, alterar qualquer informação na estrutura do bloco resultaria em um *hash* totalmente diferente. O campo *nonce* é alterado pelos mineradores até que se encontre um *hash* resultante que satisfaça a dificuldade da rede. Na tabela IV abaixo são descritos todos os campos do cabeçalho de um bloco na *blockchain* [11].

TABELA IV – ESTRUTURA DO CABEÇALHO

Tamanho	Campo	Descrição
4 bytes	Versão	Número de versão e protocolo.
32 bytes	*Hash* do bloco anterior	*Hash* do bloco anterior na cadeia.
32 bytes	Raiz da árvore de Merkle.	*Hash* contendo um resumo das transações existentes no bloco.
4 bytes	Instante de criação	Instante da criação do bloco.

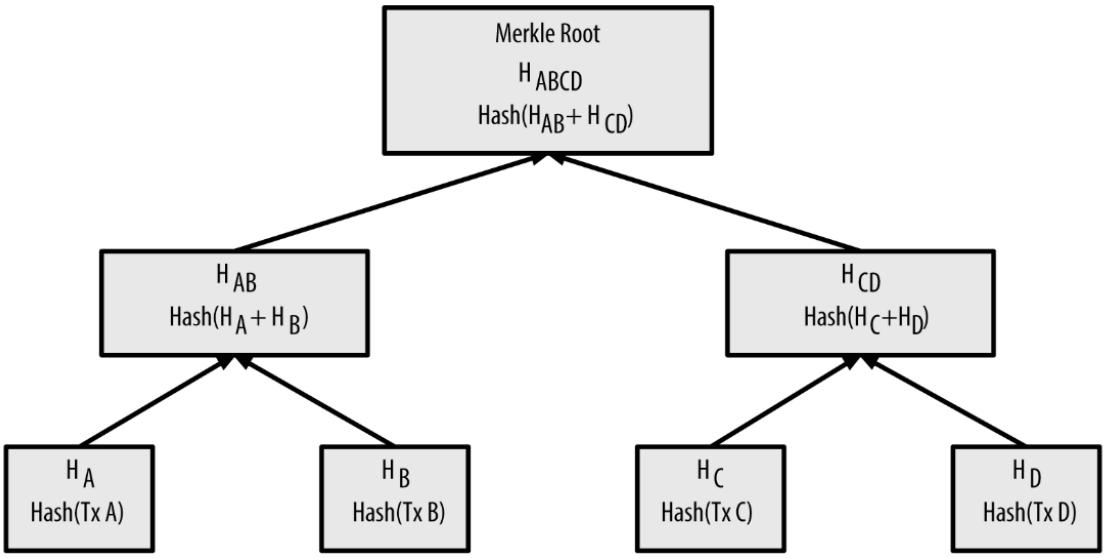
4 bytes	Dificuldade	Grau de dificuldade para mineração. Este valor é ajustado consensualmente para garantir um tempo mínimo de mineração por bloco.
Variável	*Nonce*	Número inteiro encontrado como solução no processo de mineração. Após encontrado, esse valor é denominado *golden nonce*.

Fonte: Rodrigues, C. K. S. (2018) [10]

A terceira linha da Tabela IV mostra que um bloco da cadeia do sistema Bitcoin armazena um resumo das transações nele contidas no campo do cabeçalho, chamado raiz da árvore de Merkle. Uma árvore de Merkle é uma estrutura de dados capaz de armazenar uma grande quantidade de informação e fornecer um método fácil para verificar se determinado dado está armazenado nela. Inicialmente os dados são organizados em ordem e depois se formam pares com eles (podendo ser trios ou maiores agrupamentos, dependendo do tipo de árvore que se deseja). Para cada par é aplicado uma função *hash,* e após isso, os *hashes* são concatenados em pares e aplica-se novamente a função *hash*, até chegar no topo da árvore e se obter um *hash* que representa todos os dados armazenados na árvore. O método usado para demonstrar a presença de um dado na árvore é chamado de Prova de Merkle e usa a estrutura da árvore como um atalho [11].

A árvore de Merkle é a estrutura de dados usada pelo sistema Bitcoin para organizar as transações dentro do bloco. Ela é uma árvore binária que usa o algoritmo SHA256 (Figura 15) em sua construção. Cada nó folha da árvore representa uma transação e seu formato é um *hash* do SHA256 executado sobre a transação duas vezes, sendo também chamado de SHA256-duplo. Caso o bloco contenha um número ímpar de transações, a última transação será duplicada para completar a estrutura de árvore binária. A figura 15 mostra o exemplo das transações A, B, C e D sendo armazenadas em uma árvore de Merkle. Os nós folha são o *hash* duplo de cada transação. E para cada dupla de transações, a estrutura de dados irá concatenar os *hashes* e produzir um novo *hash* duplo. O processo se repetirá até que se chegue à raiz da árvore [15].

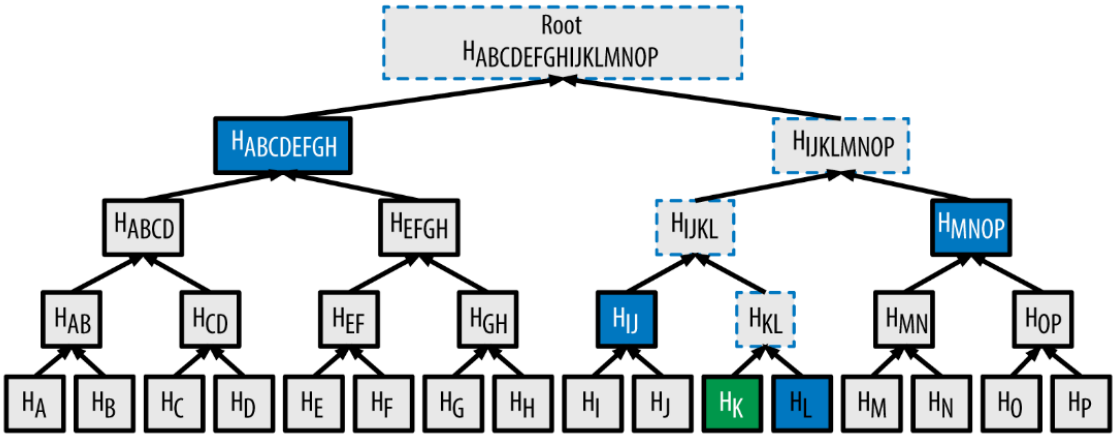
Figura 15 – Exemplo de árvore de Merkle.



Fonte: Mastering Bitcoin: unlocking digital cryptocurrencies - Andreas Antonopoulos [15].

Essa estrutura de árvore garante que cada nó da árvore, assim como a raiz, tenha o mesmo tamanho, apenas 32 bytes. Essa propriedade da árvore de Merkle permite que seja muito prática a verificação de que uma transação pertence a determinado bloco da *blockchain*. Basta fornecer a transação e o caminho de Merkle, que é o caminho usado para percorrer uma árvore de Merkle desde o nó folha, que é a transação, até a raiz, que está no cabeçalho do bloco, assim provando que aquela transação está naquele bloco da cadeia. A Figura 16 abaixo demonstra o caminho de Merkle usado para provar que a transação K pertence ao bloco desta árvore, e os quatro *hashes* identificados em pontilhado azul representam o caminho de Merkle [15].

Figura 16 – Exemplo de caminho de Merkle.



Fonte: Mastering Bitcoin: unlocking digital cryptocurrencies - Andreas Antonopoulos [15].

As árvores de Merkle são muito usadas pelos nós SPV da rede bitcoin, que são as carteiras que fazem Verificações Simplificadas de Pagamentos. Não é preciso fazer o download do bloco completo para provar que uma transação está contida nele, somente o cabeçalho do bloco que contém a raiz da árvore de Merkle e o caminho de Merkle já

provam que o bloco armazena a transação. E o *hash* do cabeçalho do bloco comprova que ele faz parte da *blockchain* [15].

Na próxima seção será descrito a atividade de mineração e o uso do algoritmo SHA256 em seu processo.

Mineração

Na figura 3, que exemplificou o ciclo de vida de uma transação bitcoin, mostrou-se que a mineração é uma etapa importante que acontece no processo de validação de uma transação. Esta seção mostra como ela acontece interagindo com as transações e com a *blockchain*.

Um dos dois propósitos da mineração é a validação das transações pendentes. Durante a mineração serão requisitados a rede do sistema Bitcoin quais transações estão pendentes de validação. Assim, os mineradores vão incluir, além da transação de Rob e Laura exemplificada pela figura 3, todas as outras que estiverem pendentes naquele momento em um bloco. E então tentarão adicionar este bloco a *blockchain.*

O outro propósito da mineração, e o fator motivacional para que um usuário queira participar da rede do sistema Bitcoin e ajudar a validar transações, é a criação de novas moedas bitcoins. As novas moedas serão a recompensa ao usuário que conseguir incluir o bloco na *blockchain* primeiro [20].

Esses usuários do sistema Bitcoin não vão utilizá-los para pagamentos ou recebimentos com a moeda bitcoin, mas vão usar seus computadores em busca de recompensa em bitcoins.

A atividade de mineração se propagou pelo mundo inteiro conforme o valor da moeda foi aumentando e se tornando relevante e lucrativo. Esses usuários que buscam lucro ao tentar produzir novas moedas bitcoins a serem usadas pelo sistema são chamados de **mineradores** [14].

Na rede bitcoin mineradores ficam continuamente trabalhando na validação dos blocos, concorrendo entre si pela recompensa. Aquele que conseguir validar o bloco primeiro, divulga para a rede a prova de que validou corretamente o bloco e começa a tentar validar o próximo bloco.

O processo de mineração de bitcoin está associado ao processo de validação das transações que é o mesmo que adicionar um novo bloco à *blockchain*. Na formação de um novo bloco, o software de mineração pergunta à rede qual o bloco mais atual da cadeia e então adiciona as transações pendentes de validação, que são enviadas à rede bitcoin pelos usuários da moeda, como Rob e Laura em nosso exemplo. Então as transações são agrupadas pelo software do minerador em um bloco, com a estrutura descrita nas tabelas anteriormente. A primeira transação inserida no bloco é onde a geração de novas bitcoins ocorrerá (se ele conseguir validar o bloco), pois ali o minerador declara que ele mesmo será o destinatário de uma transação sem remetente. Uma transação sem remetente significa que bitcoins estão sendo criadas. O destinatário da transação está recebendo bitcoins novas, que não pertenciam a ninguém antes disso.

O valor da recompensa ao minerador é determinado pelo sistema e começou sendo 50 bitcoins, porém é reduzido pela metade a cada 4 anos aproximadamente. Na própria implementação do sistema é programado um decréscimo no valor da recompensa aos mineradores sempre que um certo número de blocos forem adicionados a *blockchain*. Com os dados agrupados no bloco, o minerador passa a calcular o *nonce,* uma sequência de caracteres concatenada ao final do bloco que pode variar de tamanho de acordo com as especificações da estrutura do bloco descrita na tabela 2. O que o minerador precisa para validar o bloco é encontrar um *nonce* que ao ser adicionado ao bloco gera um *hash* menor do que a dificuldade atual que é definida pela rede. A dificuldade é determinada pela quantidade de bits 0 no começo do *hash*. Quanto mais bits precisarem ser 0, maior a dificuldade de se produzir o *hash*, visto que o SHA256 produz *hashes* completamente diferentes mesmo quando se altera apenas um bit [14]. Quando um *nonce* satisfaz a dificuldade exigida, ele é chamado de *golden nonce* e o minerador o divulga com o bloco para a rede.

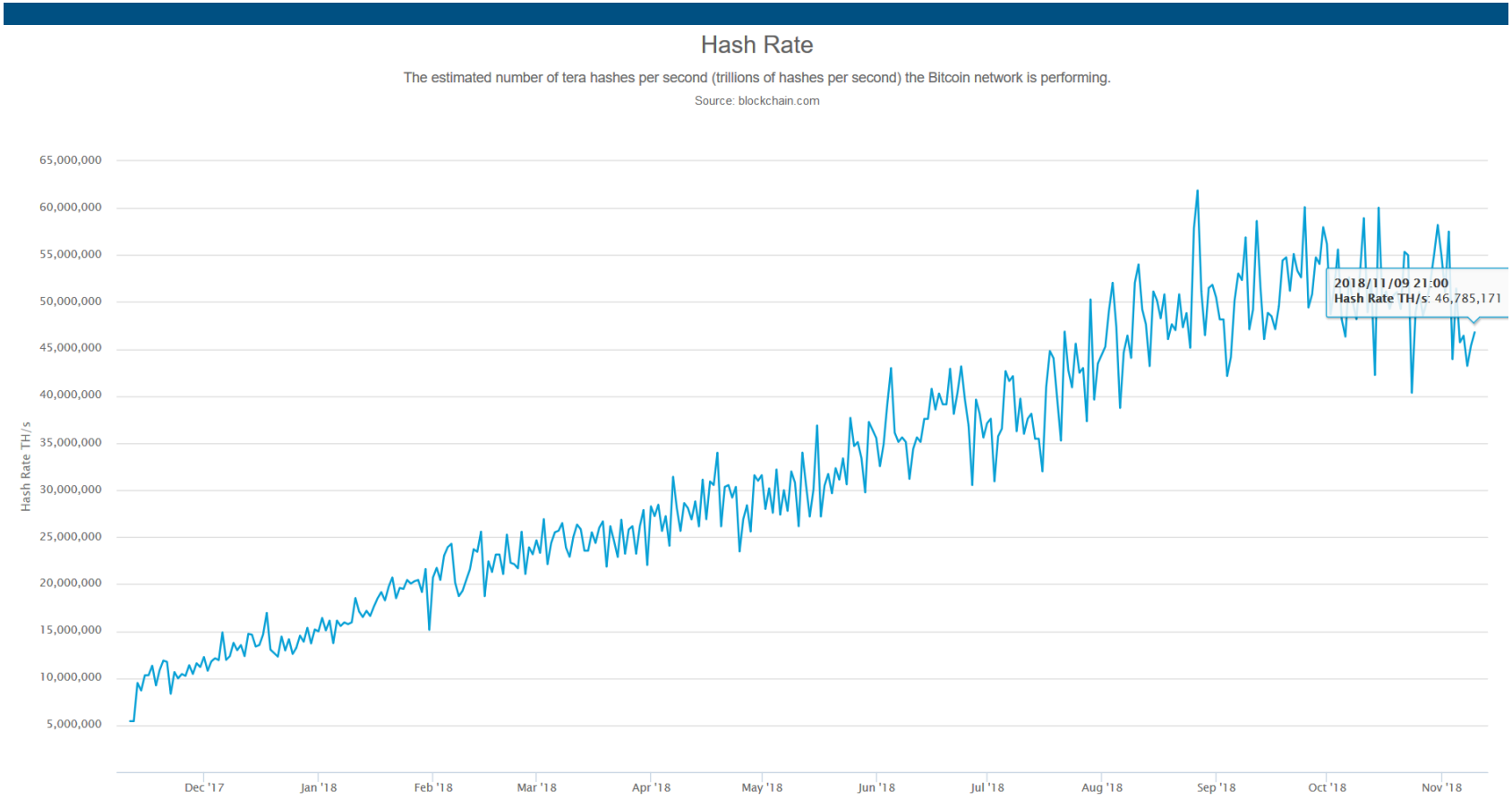
Esta busca pelo *golden nonce* usando o algoritmo SHA256 acontece da seguinte maneira. Após a criação do bloco, o campo *nonce* do cabeçalho fica em branco. Todos os dados do bloco (versão, *hash* do bloco anterior, resumo das transações, *timestamp* da criação do bloco, dificuldade do *nonce* e *nonce*) são processados pelo SHA256, o que gera um *hash* de 256 bits. A dificuldade atual da blockchain define que este *hash* precisa conter uma quantidade de bits 0 no começo do *hash*, ou seja, o *hash* precisa ser menor que determinado valor. A variação do valor do *nonce* é o que os mineradores percorrem para buscar por um *hash* resultante que seja correspondente àquele bloco que estão tentando validar e que seja menor que a dificuldade estabelecida no momento de sua criação. Isso significa executar o algoritmo SHA256 quadrilhões de vezes, dependendo da dificuldade da *blockchain*, o que geralmente leva em torno de 7 a 10 minutos, até encontrar o *golden nonce*. Para isso o minerador gasta hardware e energia elétrica, e por isso é recompensado em bitcoins, o que torna a mineração rentável. E quando o minerador encontra o *golden nonce*, ele publica para a rede que encontrou o mais novo bloco da cadeia, assim os outros mineradores podem verificar e validar este bloco. A verificação de validação feita pelos mineradores é um processo computacional muito rápido, pois eles recebem o *hash* e o bloco, e apenas rodam o algoritmo SHA256, e verificam se o *hash* resultante satisfaz a dificuldade de mineração atual da rede.

Se durante este processo o minerador perceber que outro bloco foi adicionado a *blockchain* antes que ele conseguisse validar seu bloco, ele para a tentativa de validação, cria outro bloco baseado nesse bloco mais novo, requisita a rede as novas transações pendentes, e reinicia o processo de validação em busca do *golden nonce*. Quando isso ocorre, o bloco que o minerador estava tentando validar já não serve mais, pois ele era baseado no bloco anterior. Mesmo que ele consiga validar seu bloco, quando divulga-lo para a rede ele não será aceito pois não usou o bloco mais recente da *blockchain*. Por isso, apesar de ter gasto tempo e energia tentando validar aquele bloco, ele deve abandoná-lo e recomeçar o processo de validação. Este mecanismo faz com que o minerador que dispender mais poder computacional seja remunerado pelo seu esforço. Se o minerador ignorar a existência de um novo bloco adicionado, terá seu bloco descartado pela rede quando conseguir validá-lo, pois sempre que os mineradores começam o processo de validação eles buscam pelo bloco mais recente.

Pelas regras do sistema Bitcoin, as novas moedas geradas pela mineração precisam ter 100 blocos acima de si, ou seja, validados após este bloco, antes de serem gastas [14]. Além disso, todo esse processo de busca pelo *golden nonce*, a partir da dificuldade estabelecida pela rede, faz com que o ato de encontrá-lo, comprove que aquele minerador executou o algoritmo SHA256 sucessivamente até que alcançou o objetivo. Perante isso, quando um minerador apresenta um bloco válido para a rede, todos os procedimentos e o algoritmo de validação fornecem a prova de trabalho, de que este minerador seguiu as regras e gastou poder computacional e energia elétrica para adicionar um novo bloco a cadeia e validar as transações nele contidas. E diante desta prova de trabalho será construído o próximo bloco, e quanto mais blocos estiverem acima dele significa que mais poder computacional foi utilizado, aumentando a autenticidade do bloco, e a sua prova de trabalho [15].

A figura 17 a seguir demonstra a quantia de poder computacional medida em *hashes* por segundos desde a origem do sistema até atualmente, ou seja, uma demonstração da variação de quantas vezes o algoritmo SHA256 é executado em busca do *golden nonce* por todos os computadores operando o sistema Bitcoin, em relação ao início do sistema até atualmente. Podemos notar que conforme o valor da moeda bitcoin foi aumentando e o sistema se tornando relevante, a quantidade de *hashes* por segundo foi aumentando significativamente, isso porque com o aumento no valor da moeda o número de mineradores foi aumentando, e também os computadores usados especificamente para mineração foram sendo aperfeiçoados para conseguir calcular mais *hashes* mais rapidamente.

Figura 17 – Taxa de *hash*. O número estimado de *hashes* por segundo (trilhões de *hashes* por segundo) que a rede Bitcoin está realizando.



Fonte: Blockchain - Most Trusted Crypto Company [19].

A capacidade do sistema Bitcoin de mensurar a quantidade de *hashes* por segundo sendo executados pelos mineradores é o mecanismo que controla a dificuldade de validação citada na seção de mineração. É importante controlar a dificuldade de validação para evitar que muitas moedas sejam adicionadas ao sistema em pouco tempo, pois isso causaria inundação de moedas no sistema econômico da bitcoin levando a sua desvalorização por inflação.

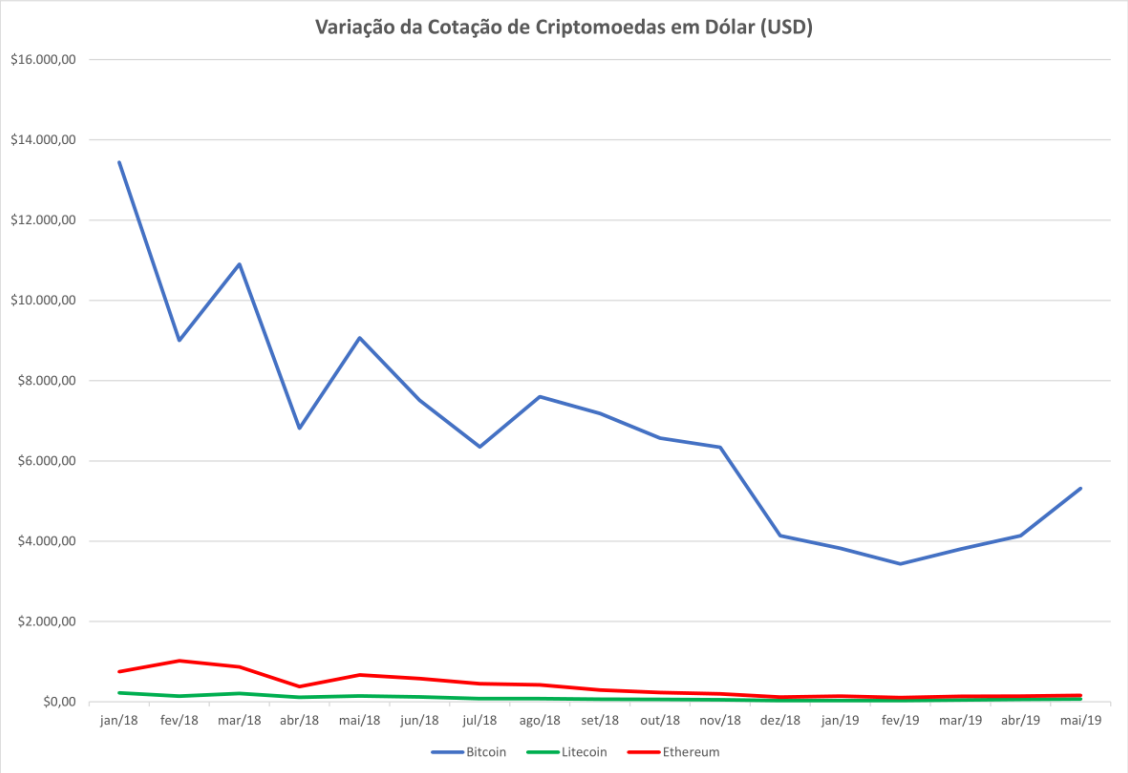
Considerações Finais

A proposta de Satoshi Nakamoto é um software livre para operar um dinheiro livre. Sem autoridades centrais e oferecendo total controle do sistema aos usuários, o sistema Bitcoin trouxe inovações tanto para a tecnologia, como a aplicação de criptografia e redes descentralizadas em conjunto para criar a *blockchain,* quanto para o sistema financeiro, como uma nova moeda de troca e também forma de investimento. Apesar de o seu uso ter sido repreendido em alguns países, por ser visto como uma ameaça ao sistema monetário controlado que utilizam, qualquer pessoa pode usar bitcoin em transações financeiras, e não se paga nada para passar a aceitar essa moeda em pagamentos, não é necessário pagar uma adesão ao sistema Bitcoin, só é necessário ter uma carteira bitcoin, que são gratuitas, e começar a aceitá-las como pagamento. Em geral por não haver intermediários nas transações, as taxas de transação têm valores muito menores, quando comparadas com as taxas de cartão de crédito, máquina de cartões e bancos, e podem ser feitas entre diferentes países sem custos adicionais por cruzar fronteiras.

Apesar de ser taxada como moeda sem lastro, pode-se dizer que a bitcoin tem um tipo de lastro diferente das moedas tradicionais, o lastro matemático. O sistema Bitcoin prevê que novas moedas serão geradas a cerca de cada dez minutos e que o total será de vinte e um milhões de unidades [22]. A atividade de mineração, responsável pela geração das moedas também pode ser facilmente ligada a três elementos que representam valor: tempo, energia elétrica e *hardware*. O algoritmo de mineração garante que levará um tempo para se conseguir criar novas bitcoins, e que será feito uso de *hardwares,* ou seja, equipamentos com valor de mercado e também será consumida energia elétrica pelos equipamentos [23].

Apesar de já haver estabelecimentos que aceitam bitcoin em pagamentos no Brasil, é ainda incomum seu uso para tal devido a alta volatilidade em sua cotação, o que é comum para quase todas as criptomoedas. A figura 18 apresenta a volatilidade de três criptomoedas, representando uma das dificuldades de se aceitá-las como pagamento, porém isto é um atrativo para forma de investimento, mesmo com a desvalorização em relação ao dólar americano no período de janeiro de 2018 a janeiro de 2019 o histórico de volatilidade na cotação das criptomoedas ainda atrai investidores.

Figura 18 – Variação dos valores de bitcoin, litecoin e ethereum.



Fonte: O autor, dados retirados do portal TradingView [25].

A forma mais fácil de investir em bitcoin ou qualquer outra criptomoeda é através de uma *exchange*, que é basicamente uma corretora de criptomoedas muito similar às corretoras de valores do mercado financeiro tradicional, onde se compra e vende ações. Essas *exchanges* são plataformas online que requerem cadastro de seus usuários com endereço de e-mail, dados de identificação e, algumas vezes, imagens de documentos oficiais. O procedimento padrão é depositar dinheiro na conta bancária da *exchange* para então usar este saldo na aquisição de criptoativos [39].

Além da volatilidade da cotação, pode haver outros empecilhos na utilização das moedas virtuais, como no caso da corretora canadense Quadriga, que após a morte de seu fundador perdeu todo o acesso às criptomoedas de seus clientes gerando um prejuízo no valor aproximado de 137 milhões de dólares. A segurança das carteiras, uma carteira fria que armazenava os códigos de acesso para as carteiras dos clientes, era feita somente pelo fundador da corretora e ele não havia deixado a senha para acesso em seu computador pessoal com nenhum outro funcionário de sua empresa [26].

Outro fator negativo para a confiabilidade do uso da criptomoeda no Brasil é a ligação com o crime de lavagem de dinheiro. De acordo com a Polícia Federal, em março de 2018, 300 mil reais foram transformados em bitcoin para impedir o rastreamento do dinheiro público desviado. Devido à falta de regulação, a Receita Federal do Brasil não teria acesso às informações de onde foram compradas as bitcoins com o dinheiro desviado [27].

Apesar dessas dificuldades, no cenário nacional, um shopping em Recife tinha previsão de implementar o recebimento com a criptomoeda em todas as suas lojas em janeiro de

2018, assim como também em algumas lanchonetes, transportadoras, agências de turismo e empresas de softwares em outros lugares do país [24].

Completando dez anos desde a publicação do artigo que originou o sistema, a bitcoin representa uma revolução tecnológica que não se pode ignorar. Criticada e ovacionada por muitos, é um fenômeno que usa conceitos matemáticos, lógica de programação, criptografia e os aplica sobre conceitos econômicos, contábeis e financeiros de maneira muito bem implementada, além de abrir caminhos para novos estudos e novas implementações. A aplicação da *blockchain* já é visada para controlar outros tipos de ativos financeiros que não sejam moedas, mas ações de uma empresa, ou posse de bens imóveis por exemplo. Sua implementação também permite o uso como contratos inteligentes que seriam assinados digitalmente e não poderiam ser alterados posteriormente, podendo ser aplicado no ramo de negócios em que contratos devem ser registrados e assinados. Também é possível o uso para rastreio de mercadorias, sendo possível rastrear lotes específicos e acessar suas informações como quantidade de mercadorias e valor de venda unitário [29]. Certamente a maior colaboração do sistema Bitcoin é a tecnologia da *blockchain*, e esta ainda oferece muitos estudos e muitas aplicações.

Recursos

[1] BANCO CENTRAL DO BRASIL (BACEN). Disponível em <http://www.bcb.gov.br/pre/bc_atende/port/servicos7.asp#a1> Acesso em: 20/07/2018.

[2] MASTERCARD - Serviços de Processamento de Pagamento Mastercard. Disponível em <https://www.mastercard.com.br/pt-br/sobre-mastercard/oque-fazemos/processamento-pagamentos.html> Acesso em: 20/01/2018.

[3] BANCO CENTRAL DO BRASIL (BACEN) - FAQ - Câmbio - Cartão de crédito internacional. Disponível em <http://www.bcb.gov.br/pre/bc_atende/port/cardInt.asp#5> Acesso em: 20/07/2018.

[4] GOVERNO DO BRASIL - Pagamento eletrônico cresce enquanto cai uso de cheque. Disponível em <http://www.brasil.gov.br/economia-e-emprego/2016/07/pagamento-eletronico-cresce-enquanto-cai-uso-de-cheque> Acesso em: 20/07/2018.

[5] DA SILVA RODRIGUES, Carlo Kleber. Sistema Bitcoin: uma análise da segurança das transações. **iSys-Revista Brasileira de Sistemas de Informação**, v. 10, n. 3, p. 5-23, 2017.

[6] BITCOIN HISTORY: The Complete History of Bitcoin [Timeline] - Disponível em <http://historyofbitcoin.org/> Acesso em: 20/07/2018.

[7] BLOCKCHAIN - Most Trusted Crypto Company. Disponível em <https://www.blockchain.com/pt/charts/market-price> Acesso em: 20/07/2018.

[8] ULRICH, Fernando. Bitcoin a moeda na era digital. São Paulo: Instituto Ludwig Von Mises Brasil, 2014, p.18.

[9] REGALADO, João Miguel dos Santos. Determinantes da procura da Bitcoin. 2015, p.5. Tese de Doutorado.

[10] RODRIGUES, Carlo Kleber da Silva. Uma análise simples de eficiência e segurança da Tecnologia Blockchain. Revista de Sistemas e Computação, Salvador, v. 7, n. 2, p. 147-162, jul./dez. 2017.

[11] FERREIRA, Frederico Lage. Blockchain e Ethereum Aplicações e Vulnerabilidades. 2017. Trabalho de Formatura Supervisionado.

[12] SENDIN, Ivan da Silva et al. Funções de hashing criptograficas. 1999.

[13] SHA256, Uma Implementação Literal. Por webassemblycodecom-admin. Abril 29, 2018. Disponível em <http://webassemblycode.com/pt/sha256-uma-implementacao-literal/> Acesso em: 20/10/2018.

[14] ANTONOPOULOS, Andreas. Algoritmos de Consenso, Tecnologia Blockchain e Bitcoin – por Andreas M. Antonopoulos. Disponível em: < https://youtu.be/fw3WkySh_Ho> Acesso em: 20/10/2018.

[15] ANTONOPOULOS, Andreas M. **Mastering Bitcoin: unlocking digital cryptocurrencies**. " O'Reilly Media, Inc.", 2014.

[16] The Cryptoverse. What Is A Bitcoin Wallet? - The Best Explanation EVER. Disponível em: < https://youtu.be/AD-vWx3oA84> Acesso em: 20/10/2018

[17] How to Store Your Bitcoin. Coindesk. Disponível em: < https://www.coindesk.com/information/how-to-store-your-bitcoins> Acesso em: 20/10/2018.

[18] Movable Type Scripts. Disponível em: <https://samifar.in/sha256.html> Acesso em 20/10/2018.

[19] BLOCKCHAIN – Most Trusted Crypto Company. Disponível em: < https://www.blockchain.com/pt/charts/hash-rate?timespan=all> Acesso em 20/10/2018.

[20] Medium – Sanjeet Sahay. This is how i explained bitcoin to a 7 years old kid part 2. Disponível em: < https://medium.com/@sanjeetsahay/this-is-how-i-explained-bitcoins-to-a-7-year-old-kid-part-2-499694f4f8cd> Acesso em 22/04/2019.

[21] The real value of bitcoin and crypto currency technology – The Blockchain explained. Disponível em: < https://www.youtube.com/watch?time_continue=147&v=YIVAluSL9SU> Acesso em 13/05/2019.

[22] A verdade sobre o lastro do Bitcoin - Infomoney. Disponível em: < https://www.infomoney.com.br/blogs/cambio/moeda-na-era-digital/post/3206256/verdade-sobre-lastro-bitcoin> Acesso em 24/05/2019.

[23] Bitcoin Mining. Disponível em: < https://www.bitcoinmining.com/> Acesso em 24/05/2019.

[24] Tudo o que você já consegue comprar com bitcoin no Brasil e no mundo – Época Negócios. Disponível em: < https://epocanegocios.globo.com/Tecnologia/noticia/2017/12/tudo-o-que-voce-ja-consegue-comprar-com-bitcoin-no-brasil-e-no-mundo.html> Acesso em 24/05/2019.

[25] TradingView. Disponível em: < https://br.tradingview.com/> Acesso em 24/05/2019.

[26] Usuários ficam sem acesso a milhões em criptomoedas após morte de dono de corretora. - G1 O portal de notícias da Globo. Disponível em: < https://g1.globo.com/economia/noticia/2019/02/05/usuarios-ficam-sem-acesso-a-milhoes-em-criptomoedas-apos-morte-de-dono-de-corretora.ghtml> Acesso em 24/05/2019.

[27] PF descobre 1º esquema de lavagem de dinheiro envolvendo Bitcoin no Brasil – Tecmundo – Descubra e aprenda sobre tecnologia. Disponível em: < https://www.tecmundo.com.br/mercado/128146-pf-descobre-1-esquema-lavagem-dinheiro-envolvendo-bitcoin-brasil.htm> Acesso em 24/05/2019.

[28] Modiax – O que é minerar bitcoin. Disponível em: < https://www.modiax.com/descubra/educacao/o-que-e-minerar-bitcoin/> Acesso em 28/05/2019.

[29] Entendendo o Blockchain e suas aplicações. Disponível em: < https://youtu.be/44L0-E0MD1c> Acesso em 29/05/2019.

[30] Malavida, Aplicativos para seu android. Disponível em: < https://www.malavida.com/br/soft/bitcoin-wallet/android/#gref> Acesso em 29/05/2019.

[31] Buy Bitcoin Worldwide. Disponível em: < https://www.buybitcoinworldwide.com/pt-br/carteiras-bitcoin/> Acesso em 29/05/2019.

[32] BitcoinPaperWallet. Disponível em: < https://bitcoinpaperwallet.com/> Acesso em 29/05/2019.

[33] Anatomy of a Blockchain – Avtar Sehra. Disponível em: < https://www.slideshare.net/arcatomia/anatomy-of-a-blockchain> Acesso em 29/05/2019.

[34] Por que criador do bitcoin se viu obrigado a revelar identidade secreta – BBC News Brasil. Disponível em: <https://www.bbc.com/portuguese/internacional/2016/05/160502_verdadeiro_criador_bitcoin_fn> Acesso em 10/06/2019.

[35] Criador do Bitcoin era um grupo de indianos, diz John McAfee – CoinTimes. Disponível em: <https://cointimes.com.br/criador-do-bitcoin-era-um-grupo-de-indianos-diz-john-mcafee/> Acesso em 10/06/2019.

[36] Quem é Satoshi Nakamoto? Veja algumas teorias – Foxbit. Disponível em: <https://foxbit.com.br/blog/quem-e-satoshi-nakamoto-veja-algumas-teorias/> Acesso em 10/06/2019.

[37] A simple explanation of bitcoin “Sidechains” - Richard Gendal Brown, Thoughts on the future of finance. Disponível em: <https://gendal.me/2014/10/26/a-simple-explanation-of-bitcoin-sidechains/> Acesso em 10/06/2019.

[38] Medium – Allex Ferreira. Aplicações de Sidechain e Crosschain: RSK, Bytom, Polkadot e Ppk. Disponível em: <https://medium.com/@allexferreira/aplica%C3%A7%C3%B5es-de-sidechain-e-crosschain-rsk-bytom-polkadot-e-ppk-c014062d6b54> Acesso em 10/06/2019.

[39] Blog Mercado Bitcoin – Exchange de criptomoedas: O que é e como escolher? Disponível em: <https://blog.mercadobitcoin.com.br/exchange-de-criptomoedas-o-que-%C3%A9-e-como-escolher-f06d8c41e0b7> Acesso em 10/06/2019.

ANEXO A – CÓDIGO FONTE DA IMPLEMENTAÇÃO EM JAVASCRIPT DO ALGORITMO SHA256 [18].

```
/* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - */
/* SHA-256 (FIPS 180-4) implementation in JavaScript                (c) Chris Veness 2002-2018  */
/*                                                                                  MIT Licence  */
/* www.movable-type.co.uk/scripts/sha256.html                                                    */
/* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - */


/**
 * SHA-256 hash function reference implementation.
 *
 * This is an annotated direct implementation of FIPS 180-4, without any optimisations. It is
 * intended to aid understanding of the algorithm rather than for production use.
 *
 * While it could be used where performance is not critical, I would recommend using the 'Web
 * Cryptography API' (developer.mozilla.org/en-US/docs/Web/API/SubtleCrypto/digest) for the browser,
 * or the 'crypto' library (nodejs.org/api/crypto.html#crypto_class_hash) in Node.js.
 *
 * See csrc.nist.gov/groups/ST/toolkit/secure_hashing.html
 *     csrc.nist.gov/groups/ST/toolkit/examples.html
 */
class Sha256 {


  /**
   * Generates SHA-256 hash of string.
   *
   * @param   {string} msg - (Unicode) string to be hashed.
   * @param   {Object} [options]
   * @param   {string} [options.msgFormat=string] - Message format: 'string' for JavaScript string
   *   (gets converted to UTF-8 for hashing); 'hex-bytes' for string of hex bytes ('616263' ≡ 'abc') .
   * @param   {string} [options.outFormat=hex] - Output format: 'hex' for string of contiguous
   *   hex bytes; 'hex-w' for grouping hex bytes into groups of (4 byte / 8 character) words.
   * @returns {string} Hash of msg as hex character string.
   */
  static hash(msg, options) {
    const defaults = { msgFormat: 'string', outFormat: 'hex' };
    const opt = Object.assign(defaults, options);

    // note use throughout this routine of 'n >>> 0' to coerce Number 'n' to unsigned 32-bit integer

    switch (opt.msgFormat) {
      default: // default is to convert string to UTF-8, as SHA only deals with byte-streams
      case 'string':   msg = utf8Encode(msg);       break;
      case 'hex-bytes':msg = hexBytesToString(msg); break; // mostly for running tests
    }
// constants [§4.2.2]
    const K = [
      0x428a2f98, 0x71374491, 0xb5c0fbcf, 0xe9b5dba5, 0x3956c25b, 0x59f111f1, 0x923f82a4, 0xab1c5ed5,
      0xd807aa98, 0x12835b01, 0x243185be, 0x550c7dc3, 0x72be5d74, 0x80deb1fe, 0x9bdc06a7, 0xc19bf174,
      0xe49b69c1, 0xefbe4786, 0x0fc19dc6, 0x240ca1cc, 0x2de92c6f, 0x4a7484aa, 0x5cb0a9dc, 0x76f988da,
      0x983e5152, 0xa831c66d, 0xb00327c8, 0xbf597fc7, 0xc6e00bf3, 0xd5a79147, 0x06ca6351, 0x14292967,
      0x27b70a85, 0x2e1b2138, 0x4d2c6dfc, 0x53380d13, 0x650a7354, 0x766a0abb, 0x81c2c92e, 0x92722c85,
      0xa2bfe8a1, 0xa81a664b, 0xc24b8b70, 0xc76c51a3, 0xd192e819, 0xd6990624, 0xf40e3585, 0x106aa070,
      0x19a4c116, 0x1e376c08, 0x2748774c, 0x34b0bcb5, 0x391c0cb3, 0x4ed8aa4a, 0x5b9cca4f, 0x682e6ff3,
      0x748f82ee, 0x78a5636f, 0x84c87814, 0x8cc70208, 0x90befffa, 0xa4506ceb, 0xbef9a3f7, 0xc67178f2 ];
```

```
// initial hash value [§5.3.3]
      const H = [
        0x6a09e667, 0xbb67ae85, 0x3c6ef372, 0xa54ff53a, 0x510e527f, 0x9b05688c, 0x1f83d9ab,
0x5be0cd19 ];

      // PREPROCESSING [§6.2.1]

      msg += String.fromCharCode(0x80);  // add trailing '1' bit (+ 0's padding) to string [§5.1.1]

      // convert string msg into 512-bit blocks (array of 16 32-bit integers) [§5.2.1]
      const l = msg.length/4 + 2; // length (in 32-bit integers) of msg + ‘1’ + appended length
      const N = Math.ceil(l/16);  // number of 16-integer (512-bit) blocks required to hold 'l' ints
      const M = new Array(N);     // message M is N×16 array of 32-bit integers

      for (let i=0; i<N; i++) {
        M[i] = new Array(16);
        for (let j=0; j<16; j++) { // encode 4 chars per integer (64 per block), big-endian encoding
          M[i][j] = (msg.charCodeAt(i*64+j*4+0)<<24) | (msg.charCodeAt(i*64+j*4+1)<<16)
                  | (msg.charCodeAt(i*64+j*4+2)<< 8) | (msg.charCodeAt(i*64+j*4+3)<< 0);
        } // note running off the end of msg is ok 'cos bitwise ops on NaN return 0
      }
      // add length (in bits) into final pair of 32-bit integers (big-endian) [§5.1.1]
      // note: most significant word would be (len-1)*8 >>> 32, but since JS converts
      // bitwise-op args to 32 bits, we need to simulate this by arithmetic operators
      const lenHi = ((msg.length-1)*8) / Math.pow(2, 32);
      const lenLo = ((msg.length-1)*8) >>> 0;
      M[N-1][14] = Math.floor(lenHi);
      M[N-1][15] = lenLo;


      // HASH COMPUTATION [§6.2.2]

      for (let i=0; i<N; i++) {
        const W = new Array(64);

        // 1 - prepare message schedule 'W'
        for (let t=0;  t<16; t++) W[t] = M[i][t];
        for (let t=16; t<64; t++) {
          W[t] = (Sha256.σ1(W[t-2]) + W[t-7] + Sha256.σ0(W[t-15]) + W[t-16]) >>> 0;
        }

        // 2 - initialise working variables a, b, c, d, e, f, g, h with previous hash value
        let a = H[0], b = H[1], c = H[2], d = H[3], e = H[4], f = H[5], g = H[6], h = H[7];
// 3 - main loop (note '>>> 0' for 'addition modulo 2^32')
        for (let t=0; t<64; t++) {
          const T1 = h + Sha256.Σ1(e) + Sha256.Ch(e, f, g) + K[t] + W[t];
          const T2 =     Sha256.Σ0(a) + Sha256.Maj(a, b, c);
          h = g;
          g = f;
          f = e;
          e = (d + T1) >>> 0;
          d = c;
          c = b;
          b = a;
          a = (T1 + T2) >>> 0;
        }
```

```
// 4 - compute the new intermediate hash value (note '>>> 0' for 'addition modulo 2^32')
          H[0] = (H[0]+a) >>> 0;
          H[1] = (H[1]+b) >>> 0;
          H[2] = (H[2]+c) >>> 0;
          H[3] = (H[3]+d) >>> 0;
          H[4] = (H[4]+e) >>> 0;
          H[5] = (H[5]+f) >>> 0;
          H[6] = (H[6]+g) >>> 0;
          H[7] = (H[7]+h) >>> 0;
        }

        // convert H0..H7 to hex strings (with leading zeros)
        for (let h=0; h<H.length; h++) H[h] = ('00000000'+H[h].toString(16)).slice(-8);

        // concatenate H0..H7, with separator if required
        const separator = opt.outFormat=='hex-w' ? ' ' : '';

        return H.join(separator);

        /* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - */

        function utf8Encode(str) {
          try {
            return new TextEncoder().encode(str, 'utf-8').reduce((prev, curr) => prev +
String.fromCharCode(curr), '');
          } catch (e) { // no TextEncoder available?
            return unescape(encodeURIComponent(str)); // monsur.hossa.in/2012/07/20/utf-8-in-javascript.html
          }
        }

        function hexBytesToString(hexStr) { // convert string of hex numbers to a string of chars (eg '616263' ->
'abc').
          const str = hexStr.replace(' ', ''); // allow space-separated groups
          return str=='' ? '' : str.match(/.{2}/g).map(byte => String.fromCharCode(parseInt(byte, 16))).join('');
        }
      }

      /**
       * Rotates right (circular right shift) value x by n positions [§3.2.4].
       * @private
       */
      static ROTR(n, x) {
        return (x >>> n) | (x << (32-n));
      } /**
       * Logical functions [§4.1.2].
       * @private
       */
      static Σ0(x) { return Sha256.ROTR(2,  x) ^ Sha256.ROTR(13, x) ^ Sha256.ROTR(22, x); }
      static Σ1(x) { return Sha256.ROTR(6,  x) ^ Sha256.ROTR(11, x) ^ Sha256.ROTR(25, x); }
      static σ0(x) { return Sha256.ROTR(7,  x) ^ Sha256.ROTR(18, x) ^ (x>>>3);  }
      static σ1(x) { return Sha256.ROTR(17, x) ^ Sha256.ROTR(19, x) ^ (x>>>10); }
      static Ch(x, y, z)  { return (x & y) ^ (~x & z); }          // 'choice'
      static Maj(x, y, z) { return (x & y) ^ (x & z) ^ (y & z); } // 'majority'

}

/* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - */

export default Sha256;
```