GERENCIAMENTO DE NEGOCIAÇÃO COM ARBITRAGEM EM *BLOCKCHAIN*

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS 01/2019

Discente: Cléber Macieski[1]

Orientador: Rafael Vieira Coelho[2]

Instituto Federal de Educação Ciência e Tecnologia do Rio Grande do Sul

Campus Farroupilha

Resumo: Criptomoedas tem se mostrado cada vez mais confiáveis e valiosas. As possibilidades de uso para a tecnologia de criptomoedas ainda está aberta a novas implementações, as quais tornam ainda mais útil esta tecnologia. Este artigo apresenta a idéia que auxilia o usuário de determinada rede de criptomoedas em suas transações, a fim permitir realizá las de maneira arbitrada, por um terceiro elemento bem como explicita as características desejáveis a um software com tal funcionalidade.

Palavras-chave: Criptomoedas. *Blockchain*. Multi-Assinaturas.

Abstract: According to ABNT NBR 6022:2018, an abstract in foreign language is optional.

Keywords: latex. abntex.

Data de submissão e aprovação: elemento obrigatório. Indicar dia, mês e ano

Identificação e disponibilidade: elemento opcional. Pode ser indicado o endereço eletrônico, DOI, suportes e outras informações relativas ao acesso.

1 INTRODUÇÃO

> Di-nhe-iro: Meio de troca, sob a forma de moedas ou notas, usado na aquisição de bens, na compra de serviços, de mão-de-obra, ou noutras transações financeiras, emitido pelo governo de cada país (PRIBEREAM, 2008-2013).

Em dado momento da evolução do ser humano o mesmo tornou-se capaz de dominar animais e plantas, de tal forma que a vida da espécie, anteriormente atrelada a uma rotina ditada pelo ritmo da caça e coleta de alimentos, passou a dispor de uma crescente quantidade de tecnologias voltadas, inicialmente, a produção dos bens essenciais (COSTA, 2009). Juntamente a estas novas técnicas, a espécie se organizou socialmente, e passou a dividir as tarefas, a fim de aumentar a eficiência de tais técnicas. Nesta divisão, indivíduos utilizaram as técnicas em trabalhos diversos e passaram a dominá-las e evoluí-las, de maneira especializada, definindo o início das profissões.

[1] cleber.macieski@gmail.com

[2] rafael.coelho@ifrs.edu.br

Em paralelo, a vontade de obter bens e serviços de outros definiu o comércio. Inicialmente, sistemas de troca eram a solução utilizada para realizar o comércio, porém, devido às evoluções tecnológicas, a espécie humana passou a dispor de um excedente de bens, os quais possibilitaram um aumento populacional, num ciclo que se retro-alimenta (NOGUEIRA, 2018).

Com esse crescente, o comércio passou a ficar cada vez mais complexo e problemas surgiram. Por exemplo, para conseguir obter bens de comerciantes não interessados em nada diretamente produzido por determinado produtor, havia a necessidade da formação de cadeias de relações comerciais que se mostravam ineficientes. Além disso, a troca por vezes se mostrava ineficaz no fato de que o valor estimado de determinado bem em relação a outro se mostrava incorreto ou até falso.

Com base neste tipo de problema delimitou-se a necessidade de padronizar um valor para as coisas, um modelo de referência que servisse de intermediário nas relações de comércio e que inspirasse confiança aos envolvidos. E assim nasceu o conceito de moeda (ENCYCLOPAEDIA BRITANNICA, 2018).

No início de sua utilização, a moeda foi cunhada em metais como prata e ouro, onde identificaram-se outros problemas: (a)transações de grandes quantias se tornavam difíceis; (b)armazenamento de moeda; e (c)se tornou necessário a existência de um intermediário nas transações, que fornecesse a moeda padrão, o que aumentava o custo das transações.A participação dos bancos se consolidou. Eles eram utilizados para guardar o dinheiro dos indivíduos de maneira relativamente segura, auxiliar em transações e prover serviços financeiros, a juros.

Ainda sim, a utilização de metais como moeda se mostrou ineficaz, assim sendo, as sociedades criaram o papel moeda. Tal tipo de moeda não possui valor por si só como os metais anteriormente utilizados, mas é produzida de maneira a ser de difícil cópia e era inicialmente lastreada em referência a um estoque de metal precioso existente no país, medidas que garantiam a escassez e, consequentemente, um valor relativo. Efetivamente, era possível trocar papel moeda por metal precioso nos bancos. Este modelo lastreado causou problemas: (a) Podia não suprir a demanda de dinheiro da sociedade, devido ao aumento da disponibilidade dos metais não estar estritamente relacionado com as necessidades financeiras da economia; (b) Um país talvez não conseguisse isolar sua economia da crise ou inflação do resto do mundo; (c) O processo de ajuste para um país com um deficit poderia ser longo e doloroso.

Assim sendo, a maioria dos países do mundo abandonou o lastro e passou a usar um sistema financeiro no qual o governo busca definir junto a um banco central a quantidade de dinheiro disponível (ROTHBARD, 1995). Esta abordagem traz os problemas atualmente existentes no sistema financeiro:(a) Custos extras na transação: a utilização de intermediários para possibilitar transações ou outras operações financeiras exige que as partes envolvidas paguem por serviços diversos. (b) Liberdade: a necessidade de intermediários limita a liberdade do indivíduo em realizar transações. (c) Inflação: a desvalorização da moeda de um país, causada por um grande disponibilidade de moeda, que quando descontrolada aumenta o custo dos produtos para o consumidor. (d) Deflação: a valorização da moeda, devido a baixa disponibilidade de moeda na economia, que a longo prazo em um cenário de recessão, pode gerar por exemplo, desemprego. (e) Na economia globalizada atual, problemas financeiros das nações provocam reações em cadeia, afetando diversos países.

Estas limitações do sistema monetário atual causam diversos problemas, como instabilidade econômica, pobreza, disfunções sociais e guerras.

1.1 CRIPTOMOEDAS E A TECNOLOGIA *BLOCKCHAIN*

Através das técnicas de criptografia e da computação distribuída, foi desenvolvida a tecnologia *blockchain*, que define um protocolo de comunicação eletrônico, o qual por sua vez

possibilita a implementação das chamadas criptomoedas.

Entre as particularidades da implementação efetiva de uma criptomoeda, está a capacidade de lidar com problema do gasto múltiplo de uma mesma moeda. E a solução de tal problema veio ser ser criada através da criptografia utilizada na tecnologia *blockchain*.

A tecnologia *blockchain*, intimamente ligada as criptomoedas, pode ser considerada um livro-razão, ou no inglês, *ledger*, na qual transações são registradas, e validados como únicas, utilizando de criptografia e de uma rede *peer-to-peer* para tal.

Criptomoedas impactam diretamente no sistema monetário atual:(a) A capacidade de distribuir, entre próprios indivíduos na rede, a possibilidade de realizar o trabalho de intermédio de transações, pode se mostrar, em diversos níveis, uma opção barata e eficaz na resolução do problema. (b) Utilizando a rede mundial de computadores, pode promover a liberdade a povos antes relegados o um sistema econômico restrito. (c) A auto regulação determinada na implementação do sistema pode acabar com os problemas gerados pela ingerência de governantes, atuar na estabilização do sistema econômico mundial. (d) Devido ao sistema ser baseado em um código aberto, a segurança do mesmo pode ser verificada por qualquer um, assim como qualquer pesquisador pode buscar melhorar a implementação do sistema. (e) A tecnologia de livro-razão distribuído possui capacidades ainda pouco exploradas, para auto-processar transações, na qual regras podem ser persistidas dentro *blockchain*, e serem executadas de maneira automática.

1.2 OBJETIVO GERAL

Utilizar da tecnologia *blockchain* para produzir um aplicativo para dispositivos móveis capaz de gerenciar transações, delegando arbitragem a um ou mais participantes extras a transação. A implementação desta solução visa capacitar o usuário a realizar transações, em dispositivos móveis, dando uma maior liberdade financeira aos indivíduos, reduzindo custos e aumentando a competitividade de pequenos negociantes no mercado. A simplificação da utilização de terceiros em arbitragem de conflitos busca tornar o processo menos custoso para as partes, mantendo um nível de segurança adequado para diversos tipos de negociações comuns.

1.3 OBJETIVOS ESPECÍFICOS

- Pesquisar e analisar as dificuldades técnicas em transações de m-n participantens no *blockchain Bitcoin*.
- Implementar solução para criptomoedas para dispositivos móveis.
- Explorar capacidades da tecnologia *blockchain*, principalmente para transferências complexas e criação/execução de contratos inteligentes.
- Realizar transações no *blockchain Bitcoin*.
- Realizar transações complexas (arbitradas) no *blockchain Bitcoin*.
- Utilizar tecnologias de programação para dispositivos móveis.
- Fazer uso de linguagem de programação dotada de vasta gama de recursos embutidos, com disponibilidade de um *framework* completo.
- Utilizar preferencialmente *frameworks*, API's ,softwares e ferramentas *open source*.
- Utilizar interfaces de desenvolvimento integradas, ferramentas de controle de código fonte e métodos ágeis de gerenciamento de projeto.

2 MÉTODO

A fim de atingir os objetivos deste trabalho, foi realizada ampla pesquisa on-line em materiais relacionados ao sistema monetário, criptografia, a tecnologia *blockchain*, a mais conhecida implementação desta, a rede *peer-to-peer* Bitcoin, assim como a plataforma Xamarin, para programação de dispositivos móveis multiplataforma, juntamente a linguagem de programação C# e o *framawork* .NET, utilizados pela plataforma e *framework open source* escolhidos (NBitcoin), além da linguagem de marcação XAML, usada na interface gráfica do aplicativo, e da API's QBit Ninja utilizadas para consulta do *blockchain Bitcoin* em sua rede de teste, a *TestNet*.

Os sistemas monetários atuais se correlacionam com o assunto tendo a pesquisa das teorias econômicas existentes relevância ao projeto. Em se tratando de tecnologias, o entendimento do funcionamento básico da estrutura que sustenta a rede Bitcoin se mostrou útil na melhor compreensão das características e limitações da rede, assim como o estudo da criptografia aplicada a segunrança da informação na rede validou a confiança na tecnologia. Do estudo das tecnologias de desenvolvimento de aplicações móveis, obteve-se a visão do melhor caminho a ser percorrido para o atingimento dos objetivos.

Sistema monetário, segundo Abreu e Coelho (2009 apud RATTI, 2001), é o conjunto das diversas moedas que circulam em um país, guardando entre si relações definidas de valor, de acordo com normas legais estabelecidas pelas autoridades monetárias. Criptografia por sua vez, é o estudo de técnicas matemáticas relacionadas a aspectos da segurança da informação como confidencialidade, integridade de dados, autenticação de entidades e autenticação de origem de dados (KATZ et al., 1996).

Um *blockchain* é análogo a um livro contábil ou livro razão, contendo todas as transações já executadas em determinada rede. Ele está constantemente crescendo ao passo que os chamados mineradores adicionam novos "blocos"a ele (a cada 10 minutos) para registrar as transações mais recentes. Bitcoin é dinheiro digital. É uma moeda digital e um sistema de pagamento online no qual técnicas de encriptação são utilizadas para regular a geração de unidades de moeda e verificar a transferência de fundos, operando de maneira independente de um banco central. A terminologia pode se tornar confusa devido as palavras Bitcoin e *blockchain* serem usada para se referir as três partes do conceito: a tecnologia base *blockchain*, o protocolo/cliente através do qual transações são efetivadas, e a atual criptomoeda (SWAN, 2015).

Xamarin é uma solução para desenvolvimento móvel multi-plataforma. Xamarin oferece ao desenvolvedor um caminho para criar uma aplicação multi-plataforma, sem sacrificar o uso de componentes de interface de usuário nativos ou mesmo performance, enquanto permite a reutilização de código entre as plataformas.Isto é feito através da unificação do desenvolvimento em uma única linguagem de programação, o C#. Ele utilizado o Mono *runtime* que foi desenvolvido para trazer a linguagem C# para os sistemas operacionais Linux e Macintosh (DICKSON, 2013). C# é uma linguagem de programação orientada a objetos, produzida pela Microsoft, visando o desenvolvimento para seu sistema operacional. Ela é uma linguagem independente,porém altamente relacionada ao chamado *framework* .NET. O *framework* .NET define um ambiente que suporta o desenvolvimento e execução de aplicações altamente distribuídas, baseadas em componentes. Ele habilita diferentes linguagens de computador a trabalhar juntas e provê segurança, portabilidade de programas, e um modelo de programação comum para inicialmente voltado a plataforma Windows. Para tal, ele define duas entidades de suma importância, o CLR(*Common Language Runtime*), responsável por gerenciar a execução de programas, e a .NET *Framework Class Library*, uma vasta biblioteca de classes que dispõe de diversos recursos prontos para utilização em um programa (SCHILDT, 2010). Para criação da interface de usuário do aplicativo utilizou-se de XAML (*eXtensible Aplication Markup Language*), linguagem de marcação baseada em XML, criada pela Microsoft como uma alternativa a código de programação para instanciação e inicialização de objetos, assim como organização dos mesmos em hierarquias

pais-filhos. XAML tem sido adaptado para diversas tecnologias dentro do .NET *framework*, mas tem seu principal uso na definição de interfaces de usuário (MICROSOFT, 2019).

NBitcoin é a mais completa biblioteca Bitcoin para o *framework* .NET. Ela implementa todas as mais relevantes BIP's(*Bitcoin Improvement Proposals*)e também provê acesso de baixo nível as primitivas Bitcoin, possibilitando e facilitando a construção de aplicações Bitcoin baseadas nela (DORIER, 2019). QBit Ninja é uma API(*Aplicattion Programming Interface*) *blockchain open source* do mesmo criador da biblioteca NBitcoin. Ela funciona como um *web service* capaz de consultar o *blockchain* (DORIER, 2017).

Por fim, o trabalho foi desenvolvido através da combinação das ferramentas de programação com os resultados das pesquisas realizadas.

3 REVISÃO BIBLIOGRÁFICA

4 REFERENCIAL TEÓRICO

4.1 CRIPTOGRAFIA

A tecnologia de criptomoedas funciona amplamente baseada em métodos de cálculo de *hash* e de criptografia assimétricos (ANTONOPOULOS, 2017). Criptografia é o estudo de técnicas matemáticas relacionadas a aspectos da segurança da informação como confidencialidade, integridade de dados, autenticação de entidades e autenticação de origem de dados (KATZ et al., 1996). As ferramentas de criptografia podem ser avaliadas conforme critérios como nível de segurança, funcionalidade, métodos de operação, performance e facilidade de implementação. O entendimento das atuais ferramentas de criptografia exige a compreensão do conceito de função, no sentido matemático, que no escopo da criptografia podem ser referidas como mapeamento ou transformação (KATZ et al., 1996).

4.1.1 FUNÇÕES DE UMA VIA (*ONE-WAY FUNCTIONS*)

Uma função de uma via é definida como: Uma função f de um conjunto X para um conjunto Y cujo $f(x)$ é "fácil"de computar para todo x pertencente a X mas para "essencialmente todos"elementos y pertencentes a $Im(f)$ é computacionalmente inviável encontrar qualquer x pertencente a X o qual $f(x) = y$. Exemplo: Dado $X = \{1, 2, 3, ..., 16\}$ e definido $f(x) = r_x$ para todo x pertencente a X onde r_x é o resto quando 3^x é dividido por 17:

x	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$f(x)$	3	9	10	13	5	15	11	16	14	8	7	4	12	2	6	1

Dado um número entre 1 e 16, é relativamente fácil encontrar a imagem dele abaixo de f. De qualquer forma, dado um número como 7, sem esta tabela, é difícil encontrar x dado que $f(x) = 7$. Claro que, se o número dado é 3, o valor de x será facilmente encontrado, x será 1. O ponto é que aplicar a função em um $x(f(x))$, para achar seu y, é fácil e relativamente muito mais fácil que dado somente a $f(x)$ e consequentemente seu resultado, encontrar o x que resolve para esse resultado (KATZ et al., 1996).

4.1.2 FUNÇÕES *HASH*

Funções de *hash* são uma subtipo de função de uma via, que são computacionalmente eficientes para mapear dados de tamanho arbitrário para dados de tamanho fixo. A ideia básica é que o valor do *hash* de determinado dado serve como uma representação compacta de um tamanho sempre igual deste dado. Para uma função de *hash* ter uso criptográfico, deve ser

computacionalmente inviável encontrar dois valores de entrada para ela, que computam para um mesmo valor de saída *hash* (tal fato, se ocorre, é chamado de colisão). Um uso comum de funções de para cálculo de *hashs* se dá na verificação de integridade de dados ao longo do tempo, onde um *hash* gerado anteriormente é comparado com um novo cálculo de *hash* sobre os mesmos dados a fim de verificar que não ocorreram mudanças (KATZ et al., 1996).

4.1.3 *TRAPDOOR ONE-WAY FUNCTIONS*

Funções do tipo *Trapdoor*, ou funções de uma via alçapão,são outro subtipo de função de uma via, na qual, apesar da dificuldade de encontrar x dado $f(x)$, é possível dar um segundo dado a ser computado (chamado informação de *trapdoor*), o qual torna o cálculo facilitado (KATZ et al., 1996).

4.1.4 CRIPTOGRAFIA DE CHAVE PÚBLICA/PRIVADA

Criptografia de chave pública utilizada de *one-way trapdoor functions*. O uso deste tipo de função permite a criação de duas chaves criptográficas distintas, com a propriedade que se usada uma destas chaves para encriptação, somente a outra poderá decriptar a cifra criada. Nem mesmo a chave que foi usada para criar a cifra pode descriptografar a mensagem. Um esquema de criptografia é chamado de esquema de encriptação de chave-pública se para cada par de encriptação/decriptação (e, d), uma chave e (a chave pública) fica publicamente disponível, enquanto a outra chave d (a chave privada) é mantida secreta. Para o esquema ser considerado seguro, é necessário que seja computacionalmente inviável computar d a partir de e. Como analogia é possível pensar em um cofre com uma combinação a qual somente Bob conhece. Se Bob deixar o cofre aberto, qualquer um pode colocar dados no cofre e tranca-lo, mas somente Bob poderá abri-lo. Nem mesmo quem trancou o cofre poder-a fazê-lo (KATZ et al., 1996).

4.1.5 ASSINATURAS DIGITAIS

O propósito da assinatura digital (ou de qualquer método de assinatura) é prover um meio pelo qual uma entidade tem sua identidade ligada a uma informação. Para tal, é comum a utilização de criptografia assimétrica.

4.2 BLOCKCHAIN

Como funciona...

4.3 BITCOIN

Bitcoin constitui-se de uma coleção de conceitos e tecnologias que formaram a mais conhecida e valorizada criptomoeda do mercado. Unidades da moeda digital são utilizadas para guardar e transmitir valor pelos participantes da rede. Os seus usuários se comunicam através do protocolo Bitcoin, principalmente via internet, mas no entanto, outras redes podem ser utilizadas. O protocolo Bitcoin é disponibilizado de maneira *open source* e pode rodar em uma larga gama de dispositivos de computação, incluindo computadores portáteis e smartphones, fazendo esta tecnologia altamente acessível (ANTONOPOULOS, 2017).

4.3.1 TRANSAÇÕES BITCOIN: FUNCIONAMENTO

Transações permitem aos usuários da rede de criptomoedas o gasto de valores representados pelas próprias transações (BITCOIN.ORG, 2019a). Em termos simples uma transação diz a rede que o possuidor de um valor em criptomoeda autorizou a transferência da mesma para um novo dono. O novo dono pode criar outra transação para transferi-la para um subsequente e assim por diante (ANTONOPOULOS, 2017). Uma transação contém diversos dados com

funcionalidades diversas usadas em contextos diversos. Abaixo segue a uma descrição simplificada de uma transação em um contexto comum, do tipo *Pay-to-Public-Key-Hash* (que será definido mais adiante).

Alice, a compradora, encontra um anúncio de um smartphone usado, o qual Bob, o vendedor, dispõe a opção de receber o pagamento em Bitcoins. Ele marcam de se encontrar para realizar a transação. A fim de que Alice possa enviar o valor que ela já possui previamente, Bob deverá enviar o endereço no qual ele deseja receber os valores. Para Bob criar um endereço Bitcoin para recebimento ele gera um par de chaves pública e privada com o ECDSA (*Ellipc Curve Digital Signature Algorithm*) com a curva secp256k1, algoritmo padrão para essa etapa da operação na rede Bitcoin. Após, ele executa o cálculo do *hash* da chave pública (pubkey) deste par. Ele constrói o endereço com um número de versão, este *hash* da pubkey, e um *checksum* para verificação de erros de digitação, e codifica tudo em uma string base58.

Alice de posse desse endereço, o decodifica para acessar o *hash* da pubkey do Bob. Ela utiliza este *hash* para criar o chamado pubkey *script* da transação. Quem provar possuir a chave privada gerada juntamente com a chave pública disponível no pubkey *script* poderá gastar esta transação. Ela constrói o output da transação informando no campo *value*, a quantidade de unidades a ser enviada, e referenciando este pubkey *script* recém criado. O input da transação que Alice criá, referência, através dos chamados txid's, *outputs* de uma ou mais transações anteriores, registradas no *blockchain*, a qual ela possui a chave privada correspondente ao *hash* da chave pública contido no pubkey *script*. Os *outputs* destas transações anteriores devem ter a soma do campo value resultante em no mínimo o valor do informado no output da transação que Alice está criando. Este conjunto de referências é chamado *outpoint*. Para o input, Alice também cria o *signature script* ou scriptSigs. Este *script* contém a chave pública completa informada para endereçar os *scripts* aos quais ela está referenciando no campo *outpoint*. Ele também contém dados das transações origem assinados digitalmente com a chave privada correspondente a cada chave pública de cada output referenciado no *outpoint*. Alice então constrói o input da transação com o conjunto de referências a *outputs* de origem (campo *outpoint*), o *signature script* e *sequence numbers* (números de sequência.

Por fim, Alice divulga essa transação, que é validada pelos pares da rede, adicionada ao *blockchain*, e categorizada como UTXO (*Unspend Transaction Output*), pois ainda não existem inputs subsequentes apontando para o output desta transação (BITCOIN.ORG, 2019a).

Uma transação divulgada na rede só é incluída no *blockchain* após ser validada pelo processo de mineração. A validação de uma transação na rede se dá com base nas regras de consenso incutidas na mesma, aplicadas nos dados disponibilizados pela transação e divulgadas para todos os nós da rede (ANTONOPOULOS, 2017). A criptomoeda Bitcoin utiliza de uma linguagem de *script* para transações chamada simplesmente de "Script", que é baseada em um execução em pilha, de notação polonesa reversa, parecida com a linguagem Forth. Ela também é uma linguagem classificada como Touring incompleta e sem estado (ANTONOPOULOS, 2017).

4.3.2 TRANSAÇÃO: ESTRUTURA BÁSICA E PRINCIPAIS TIPOS

Transações em Bitcoin são transmitidas entre pares na rede em um formato serializado, chamado de formato bruto. Diversas ferramentas utilizadas para manipular dados da rede utilizam a notação hexadecimal para exibi-las. Abaixo segue tabela com descrição de uma transação no chamado *transaction-level* (BITCOIN.ORG, 2019a):

- *VERSION*: Campo com a função de possibilitar a implementação de novas regras sem invalidar anteriores.
- TX_IN_COUNT: Quantidade de inputs da transação.

Tabela 1 – Estrutura de uma transação.

Bytes	Nome	Tipo de Dados	Descrição
4	*version*	uint32_t	Versão da transação.
Variável	tx_in count	compactSize uint	Números de inputs da transação.
Variável	tx_in	txIn	Inputs da transação.
Variável	tx_out count	compactSize uint	Número de *outputs* da transação.
Variável	tx_out	txOut	Outputs da transação.
4	lock_time	uint32_t	Um tempo (*Unix epoch time*) ou número de bloco.

- *TX_IN*: *Inputs* da transação, contendo três campos: o *outpoint*, um *signature script* e um *sequence number* (BITCOIN.ORG, 2019b).
- TX_OUT_COUNT: Quantidade de *outputs* da transação.
- TX_OUT: *Outputs* da transação, contendo dois campos: o campo *value* informando o valor da transação em unidades e um pubkey *script* (BITCOIN.ORG, 2019b).
- LOCK_TIME: O campo LOCK_TIME existe no *transaction-level* desde o começo da utilização da tecnologia. Tal campo define, a partir de que tempo aquela transação poderá ser gasta pelo possuidor (válida na rede). Tal campo adiciona a dimensão de tempo a transações, assemelhando-se com um cheque pré-datado, mas com garantia de não compensação prévia ao tempo estipulado. Valores abaixo de 500 milhões no campo, estipulam o bloco futuro do *blockchain* no qual a transação será válida, valores acima deste são interpretados como *Unix Epoch timestamp's* (segundos desde 1-Jan-1970). O valor zero não restringe a validação imediata da transação. Essa configuração não tem uma grande limitação. O recebedor não tem garantia que aquela transação ainda não terá sido gasta até a data que ele pode fazê-lo, ou seja, o pagador poderá gastar e invalidar a transação antes de ser possível para o recebedor. A fim de implementar a garantia de validade de transação com LOCK_TIME, foi implementado posteriormente na rede (BIB-65) o CLTV(*Check Lock Time Verify*) que permite a validação da transação do pagador, mas não possibilita o gasto pelo recebedor até o tempo futuro definido (ANTONOPOULOS, 2017).

Transações processadas na rede Bitcoin podem enviar Bitcoins para novos donos com um *script Pay to Public Key Hash* ou P2PKH *script*. Estes *scripts* estão contidos na parte anteriormente citada, o *output* da transação. Estes *scripts* de travamento efetivamente travam a transação para um *hash* de chave pública, mais conhecido como endereço Bitcoin. Eles podem ser destravados (gastos), essencialmente, pela apresentação de uma chave pública e de uma assinatura digital correspondente a chave privada da chave pública utilizada para a criação do endereço Bitcoin travado (ANTONOPOULOS, 2017). Também existem os chamados *Pay to Script Hash*, ou P2SH. São *scripts* de um tipo de transação mais novo, introduzido em 2012, que simplifica a criação de uma transação complexa, através da substituição dos *scripts* de validação complexos por um assinatura digital dos mesmos (ANTONOPOULOS, 2017).

4.3.3 BIP'S E GERAÇÃO DE CHAVES HIERÁRQUICAS DETERMINÍSTICAS

4.4 FERRAMENTAS

4.4.1 C# E .NET

4.4.2 XAMARIN

4.4.3 *FRAMEWORK* NBITCOIN

NBitcoin é uma biblioteca Bitcoin que utiliza o *framework*.NET. A mesma é de código aberto e mantida pela comunidade e por seu criador, Nicolas Dorier. NBitcoin dispõe de ferramentas básicas para a implementação de softwares capazes de interagir com diferentes *blockchains* e facilita a construção de soluções para criptomoedas usando a linguagem de programação C#. Ela implementa a grande maioria dos BIP's Bitcoin (HE, 2018).

4.4.4 API'S E QBIT NINJA

Uma API (*Application Programing Interface*) ou Interface de Programação de Aplicação, é um conjunto de protocolos, rotina, funções e/ou comandos que programadores usam para desenvolver software ou facilitar a interação entre sistemas distintos (TECHOPEDIA, 2018). Existem inúmeras API's disponíveis para criptomoedas, , cuja finalidade é fornecer informações sobre um ou mais *blockchain's* no qual as criptomoedas se baseiam. As mesmas podem ser classificadas quanto a funcionalidades como: (a) Integração com Carteiras: capacidade de trabalhar com diversas carteiras. (b) Suporte a transações: capacidade de enviar ou receber as mesmas.(c)Preço. (d)Capacidades Especiais (LASTCALL, 2018).

Tabela 2 – Tabela Comparativa de Algumas API's.:

API	Integração	Suporte a Transações	Habilidades Especiais	Preço
Digital Currency Tickers	Não	Não	Sim	Livre
Crypto Market Intraday Reference Rates	Não	Não	Sim	Livre
CoinAPI	Não	Não	Sim	Livre
ICOs	Não	Não	Sim	Planos Variados
Due Diligence	Não	Não	Sim	Livre
Global Bitcoin Price Index	Não	Não	Sim	Livre
Coinbase	Sim	Sim	Sim	Livre
CoinMarketCap	Não	Não	Sim	Livre
Nexchange	Sim	Sim	Sim	Taxas de Câmbio
GetBalance	Sim	Não	Não	Livre
BitcoinAverage	Não	Não	Sim	Planos Variados
Bitcointy	Não	Não	Sim	Livre
Bitcoin ATMs	Não	Não	Sim	Planos Variados

Fonte: (LASTCALL, 2018)

Na subseção 4.4.4 citaram-se API's comerciais. Para fins de desenvolvimento acadêmico, destaca-se a API QBit Ninja é uma API desenvolvida em código aberto, criada pelo mesmo autor do *framework* C# NBitcoin sendo dependente da classe "NBitcoin.Indexer"que por sua vez apoia-se no serviço de armazenamento Azure da Microsoft, devido a mesma ser de código aberto a de facilitada utilização e extensão (DORIER, 2017).

5 DESENVOLVIMENTO

Através desta pesquisa foi produzido o aplicativo "Árbitro Bitcoin". Este aplicativo visou incorporar os objetivos deste trabalho através das tecnologias referenciadas, utilizando

uma metodologia prática. O software assim como a tecnologia de criptomoedas se baseia na tecnologia de criptografia assimétrica, permitindo a geração de endereços de recebimento e garantindo a unicidade de cada transação na rede, tendo segurança na transmissão de dados. Este tipo de tecnologia de criptografia é largamente utilizado por diversas instituições do sistema monetário tradicional (ROUSE, 2016) e a implementação básica de características encontram-se encapsuladas no *framework* NBitcoin.

O software produzido tem as seguintes funcionalidades: (a)Consulta e exibição de saldo de um determinado endereço da rede; (b)Criação de endereço de recebimento válido; (c)Criação de endereço de recebimento arbitrado válido; (d)Criação de transações comuns para endereço válido; (e)Criação de transações utilizando de um árbitro na transação; (f)Importação, assinatura e exportação de transação serializada, para permitir arbitragem de interações na rede.

O recurso de de *scripts* de multi-assinatura definem uma condição onde N chaves públicas são gravada no script e pelo menos M dos controladores destas chaves devem fornecer assinaturas a fim de desbloquear os fundos representados nessa transação (ANTONOPOULOS, 2017). Esse tipo de *script* foi utilizado no projeto a fim de criar transações arbitradas.

No desenvolvimento do software foi utilizada a linguagem de programação C#, que por sua vez utiliza o paradigma orientada a objetos, sendo fortemente tipada e permitindo a criação de softwares robustos e seguros executados no .NET *framework*, mantendo um sintaxe parecida as principais linguagens de programação do mercado (WAGNER et al., 2015). A utilização desta linguagem proveu um amplo arsenal de recursos prontos os quais foram utilizados, como a linguagem de consulta embutida (LINQ), expressões lambda, assessores de propriedades simplificados e as funcionalidades de paralelismo embutidas na própria linguagem, através das palavras-chave async e await. Estes recursos de paralelismo permitiram rodar processos demorados, como consultas a API web e difusão de transações, em um linha de processamento paralela, mantendo a interface do usuário responsiva.

A plataforma de desenvolvimento Xamarin propiciou um desenvolvimento multi-plataforma rápido e robusto, dando uma estrutura clara e concisa ao projeto. Além disso, a utilização do Ambiente de Desenvolvimento Integrado Visual Studio 2019 revelou-se altamente preparado para projetos do gênero, trazendo agilidade na construção inicial da estrutura básica projeto e auxiliando no controle de código fonte de maneira transparente e simples através do sistema de gerenciamento de código-fonte Git, utilizando do serviço de hospedagem Bitbucket. A IDE integra recurso de dispositivos virtuais Android (AVD - Android Virtual Devices) e IO's, sendo possível a utilização dos dispositivos para *debug* diretamente na interface do Visual Studio, porém, exigindo um hardware robusto para uma execução performática. Devido a limitações de hardware no desenvolvimento em algumas máquinas utilizadas, obtou-se por utilizar o *deploy* do projeto diretamente em dispositivo Android disponível, sendo essa modalidade igualmente compatível com recursos de *debug* disponíveis nas máquinas virtuais, sem sofrer com lentidão em função do hardware utilizado para compilação, *deploy* e *debug*.

A interface de usuário declarada em XAML mostrou-se simples de criar, e a implementação do padrão de design *Model-View-View-Model* permitiu que o software fosse construído de maneira desacoplada e preparada para expansão. O recurso de *data bindings* embtido no plataforma permitiu uma implementação transparente da atualização de dados do *back-end* do aplicativo com seu interface gráfica de usuário. Já utilização do padrão de projeto *Command* na manipulação de interações do usuário com botões e controles da interface a pesar de sua inicial complexidade, simplificou o disparo de eventos a serem tratados no *back-end* assim como trouxe responsividade a interface gráfica, devido a capacidade embutida de gerenciar a ativação e desativação do controle de interface, enquanto se aguardada por resposta de *threads* executadas paralelamente.

A fim de analisar as informações da rede, foi utilizado serviço web da API QBit Ninja, que é hospedada por terceiros e disponibiliza informações sobre blocos, transações e endereços

no *blockchain* Bitcoin (DORIER, 2018), tanto em sua rede principal, a *MainNet*, quanto em na rede utilizada para testes, a *TestNet*. A rede de criptomoedas na qual o software se baseou foi a do Bitcoin, sendo até então utilizada a versão de testes da mesma.

As funcionalidades do aplicativo foram implementadas através do uso do *framework* NBitcoin...

6 CONSIDERAÇÕES FINAIS

Do atingimento dos objetivos do projeto espera-se dispor uma ferramenta de software para dispositivos móveis capaz de realizar transações arbitradas, incluindo um terceiro elemento em uma transação de binária de criptomoedas que realize as mesmas a um custo menor, de maneira simples e eficaz. A aplicação deverá explorar plenamente a capacidade de distribuir o trabalho de processamento das transações entre os indivíduos na rede, promover a liberdade dos usuários liberando-os cada vez mais da utilização dos sistemas tradicionais de negociação, manter as transações a um nível de segurança e confiabilidade nos mecanismos de resolução de conflitos e servir como uma experiência útil a disseminação da informação sobre as capacidades da rede.

REFERÊNCIAS

ABREU, Y. V. de; COELHO, S. B. *EVOLUÇÃO HISTÓRICA DA MOEDA*: Estudo de caso: Brasil (1889 – 1989). 2009. ed. Universidade de Málaga, 2009. Disponível em: <http://www.eumed.net/libros-gratis/2009a/477/index.htm>. Acesso em: 26 mai. 2019. Citado na página 4.

ANTONOPOULOS, A. M. *Mastering Bitcoin*: Programming the open blockchain. 2. ed. O'Reilly Media, 2017. Disponível em: <https://github.com/bitcoinbook>. Acesso em: 29 nov. 2018. Citado 5 vezes nas páginas 5, 6, 7, 8 e 10.

BITCOIN.ORG. *Bitcoin Developer Documentation*. [S.l.], 2019. Disponível em: <https://bitcoin.org/en/developer-documentation>. Acesso em: 16 set. 2018. Citado 2 vezes nas páginas 6 e 7.

BITCOIN.ORG. *Bitcoin Developer Glossary*. 2019. Disponível em: <https://bitcoin.org/en/developer-glossary>. Acesso em: 24 fev. 2019. Citado na página 8.

COSTA, R. Por que inventaram o dinheiro? *nova escola*, set. 2009. Disponível em: <https://novaescola.org.br/conteudo/2273/por-que-inventaram-o-dinheiro>. Acesso em: 15 set. 2018. Citado na página 1.

DICKSON, J. Xamarin mobile development. 2013. Citado na página 4.

DORIER, N. *QBit Ninja*. [S.l.], 2017. Disponível em: <https://qbitninja.docs.apiary.io/#>. Acesso em: 17 mar. 2019. Citado 2 vezes nas páginas 5 e 9.

DORIER, N. *Programming The Blockchain in C#*. 2. ed. [s.n.], 2018. Disponível em: <https://programmingblockchain.gitbook.io/programmingblockchain>. Acesso em: 29 nov. 2018. Citado na página 11.

DORIER, N. Nbitcoin. *GitHub*, mai. 2019. Disponível em: <https://github.com/MetacoSA/NBitcoin>. Acesso em: 26 mai. 2019. Citado na página 5.

ENCYCLOPAEDIA BRITANNICA. *Gold Standard*. Encyclopædia Britannica, inc, 2018. Disponível em: <https://www.britannica.com/topic/gold-standard>. Acesso em: 16 set. 2018. Citado na página 2.

HE, H. Nbitcoin: Introduction to nbitcoin. *C#Corner*, aug. 2018. Disponível em: <https://www.c-sharpcorner.com/article/introduction-to-nbitcoin/>. Acesso em: 31 mar. 2019. Citado na página 9.

KATZ, J. et al. *Handbook of applied cryptography*. [S.l.]: CRC press, 1996. Citado 3 vezes nas páginas 4, 5 e 6.

LASTCALL. The top 13 bitcoin, blockchain & cryptocurrency apis. out. 2018. Disponível em: <https://blog.rapidapi.com/bitcoin-blockchain-cryptocurrency-apis/>. Acesso em: 17 mar. 2019. Citado na página 9.

MICROSOFT. *Documentação do Xamarin*. [S.l.], 2019. Disponível em: <https://docs.microsoft.com/en-us/xamarin/xamarin-forms>. Acesso em: 26 mai. 2019. Citado na página 5.

NOGUEIRA, M. História do comércio. *estudo prático*, 2018. Disponível em: <https://www.estudopratico.com.br/historia-do-comercio/>. Acesso em: 15 set. 2018. Citado na página 2.

PRIBEREAM. *Dinheiro*. 2008–2013. Disponível em: <https://www.priberam.pt/dlpo/Dinheiro>. Acesso em: 15 set. 2018. Citado na página 1.

ROTHBARD, M. Fractional reserve banking. *FEE - Foundation for Economic Education*, out. 1995. Disponível em: <https://fee.org/articles/fractional-reserve-banking-part-ii/>. Acesso em: 16 set. 2018. Citado na página 2.

ROUSE, M. asymmetric cryptography (public key cryptography). *SearchSecurity*, jun. 2016. Disponível em: <https://searchsecurity.techtarget.com/definition/asymmetric-cryptography>. Acesso em: 29 nov. 2018. Citado na página 10.

SCHILDT, H. *C# 4.0: The complete reference*. [S.l.]: Tata McGraw-Hill Education, 2010. Citado na página 4.

SWAN, M. *Blockchain: Blueprint for a new economy*. [S.l.]: "O'Reilly Media, Inc.", 2015. Citado na página 4.

TECHOPEDIA. Application programming interface (api). 2018. Disponível em: <https://www.techopedia.com/definition/24407/application-programming-interface-api>. Acesso em: 17 mar. 2019. Citado na página 9.

WAGNER et al. Introdução à linguagem c# e ao .net framework. jul. 2015. Disponível em: <https://docs.microsoft.com/pt-br/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework>. Acesso em: 29 nov. 2018. Citado na página 10.