Pataca

um sistema de promoção da descentralização da moeda através de moedas criptográficas

Ricardo Tavares Macedo

Trabalho apresentado
ao
Instituto de Matemática e Estatística
da
Universidade de São Paulo
para
obtenção do título
de
Bacharel em Ciência da Computação

Orientador: Prof. Dr. Flávio Soares
Coorientador: Prof. Dr. Gilson Schwartz

São Paulo, dezembro de 2014

Agradecimentos

Aos professores Flávio Soares (IME) e Gilson Schwartz (ECA) por me orientarem na confecção deste trabalho de conclusão de curso.

À equipe do Mercado Bitcoin por todo o apoio e experiências providos.

À minha namorada Julia, minha família e meus amigos, pelo inestimável apoio e incentivo.

Resumo

MACEDO, R. T. **Pataca: um sistema de promoção da descentralização da moeda através de moedas criptográficas**. 2014. Trabalho de formatura (Bacharelado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2014.

Até 1971, as moedas nacionais eram conversíveis em uma certa quantidade em ouro ou em prata. Com o fim dos lastros em metais, ocorreu uma flexibilização das políticas de criação de moeda no final do século XX. Em meio a isso, dispositivos computacionais se tornaram mais baratos e poderosos, ampliando o poder de processamento disponível a um indivíduo, fomentando sistemas menos centrais de transporte de informação e divulgação de conhecimento, como a internet, a Wikipédia e os softwares livres. Em 2009, um pseudônimo chamado Satoshi Nakamoto inicia um sistema de transferência de valores sem intermediários chamado bitcoin. Utilizando esses conceitos, o Pataca implementa um sistema de gerenciamento de economias locais que usa o bitcoin como meio de troca de valores entre instâncias.

Palavras-chave: bitcoin, moeda, descentralização, economia local.

Abstract

MACEDO, R. T. **Pataca: a system for promotion of the decentralization of currency through cryptographic currencies**. 2014. Trabalho de formatura (Bacharelado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2014.

Until 1971, the national currencies were convertible in a certain amount of gold or silver. As the gold standard ended, a flexibilization of the money creation policies occurred by the end of the 20th century. Among all this, computational devices became cheap and powerful, increasing the processing power available to an individual, fostering less central systems for information transport and knowledge dissemination, as it happens with the internet, Wikipedia and Free/Libre/Open Source Software. In 2009, a pseudonym called Satoshi Nakamoto starts a value transfer system without intermediates called bitcoin. Using these concepts, Pataca implements a local economy management system which uses bitcoin as a medium of value exchange among instances.

Keywords: bitcoin, currency, decentralization, local economy.

Sumário

Capítulo 1

Introdução

A discussão a respeito da descentralização da moeda tomou força após a crise financeira de 2008, principalmente com o surgimento do bitcoin, uma sistema de transferência de valores baseado em criptografia. Porém, para compreendermos o que isto significa, precisamos estudar a origem e o funcionamento da moeda nos últimos séculos, as mudanças que ocorreram na segunda metade do século XX e como o sistema financeiro mundial atualmente se estrutura.

Ao longo da história, a moeda adquiriu um caráter de controle e registro efetuado por um grupo ou governo: somente moedas que portem uma certa inscrição, geralmente dada localmente, são válidas na obtenção de bens e serviços dentro de um certo limite geográfico. Esta característica do monopólio do poder de emissão é que caracteriza a centralidade da moeda, que encontra sua contrapartida com o surgimento do bitcoin em 2009. Diferente das moedas nacionais, o bitcoin possui uma política clara na taxa de emissão, além de uma estrutura criptográfica que o transforma em um sistema eletrônico de transferência de valores sem intermediários.

1.1 Motivação

Atualmente, economias locais, tais como programas de fidelidade ou moedas de jogos, são isoladas. Algumas apresentam algum tipo de serviço de câmbio entre a moeda local e uma moeda nacional, porém, os exemplos são raros.

A motivação por trás do desenvolvimento do Pataca provém do desejo de interligar tais economias, utilizando moedas criptográficas como estoque de valor. Isto possibilita transacionar valores entre comunidades distintas, aumentando seu alcance e interação, sem a utilização de um servidor central de intermediação. Tais comunidades podem gerenciar a emissão de suas moedas para seus membros, controlando a taxa de câmbio a partir da quantidade de moedas criptográficas armazenadas.

1.2 Objetivo

Este trabalho apresenta o objetivo de desenvolver a base para a geração de um sistema aberto e extensível de administração de economias locais, com um ou mais lastros em moedas criptográficas atrelados às denominações monetárias geradas.

O sistema deverá possuir código aberto, além de uma API para interação.

1.3 Organização

Esta monografia reúne diversos conceitos, agrupando-os em capítulos da seguinte maneira:

Capítulo 2 Um breve histórico da moeda, relatando seu surgimento, representação e cenário recente.

Capítulo 3 Conceitos sobre descentralização, além de exemplos recentes e atuais.

Capítulo 4 Descrição do funcionamento da moeda criptográfica bitcoin.

Capítulo 5 Descrição do sistema Pataca, para administração de economias locais e interação com economias externas.

Capítulo 6 Relatos acerca de outras atividades correlatas empreendidas ao longo do ano, além de uma apreciação pessoal e crítica.

Capítulo 2

Uma breve história da moeda

Este capítulo apresenta um breve relato do surgimento da moeda e sua representação ao longo da história humana, desde o início da utilização de metais como reserva de valor, até o total desaparecimento do mesmo, no final do século XX.

2.1 Origem da moeda

Não há registros concretos sobre o surgimento da moeda, já que tal ícone de representação de valor predata a invenção da escrita. Também, não há evidências de que, antes do advento de um meio comum de troca, seres humanos tenham utilizado o escambo para adquirir os bens que necessitassem ou desejassem.[12] A teoria mais aceita é de que as sociedades antigas se organizassem em *"economias da dádiva"*, onde indivíduos doam bens e serviços entre si, sem obrigação da reciprocidade.

Os primeiros registros de transações comerciais, com um conceito mais bem formado de moeda, estão em tábuas cuneiformes produzidas por templos e palácios sumérios desde 3500 AC. Tal civilização mesopotâmica desenvolveu um sistema baseado em uma unidade de prata chamada *shekel*, divisível em 60 *minas*, correspondente a uma porção de cevada, sob o princípio de que há 30 dias em um mês e um trabalhador recebe duas porções por dia. Sob esta perspectiva, é possível notar que, neste caso, a moeda foi uma invenção para controlar recursos e a movimentação dos mesmos entre diferentes grupos.[9]

A prata, em si, quase não circulava, permanecendo armazenada em alguns locais por, literalmente, milhares de anos. As transações eram efetuadas baseadas em créditos, podendo ser pagas através de diversos meios – móveis, cabras, minerais – já que tais templos e palácios eram tão grandes que se encontrava um uso para quase qualquer material. Comumente, os camponeses pagavam utilizando cevada, por isso a importância de se atrelar uma quantidade em prata a uma quantidade do cereal. Mercadores eram um dos poucos grupos que utilizavam prata para transações comerciais com alguma frequência, mas mesmo eles realizavam grande parte de suas negociações a crédito.

Até cerca de 600 AC, o comércio internacional utilizava metais, como o ouro e a prata, sem qualquer tipo de selo ou denominação, como meio de troca. Neste período, os lídios, um povo da Anatólia, atual Turquia, começaram a cunhar pedaços de *electrum*, uma liga de ouro e prata, com inscrições e símbolos, sendo o mais famoso deles a cabeça de um leão. Estas moedas eram padronizadas em relação a seu peso e sua composição, facilitando as transações comerciais.[14] A partir de então, diversos povos passaram a adotar a técnica da cunhagem de moedas.

2.2 Cédulas

A utilização de papel como uma representação de valor se iniciou na China, no século VII, a partir de notas promissórias. As antigas moedas chinesas eram redondas, com um furo retangular no meio, e era prática comum acumulá-las em cordões, a fim de facilitar o transporte.[15] Grandes mercadores e comerciantes, desejando evitar o peso de carregar um valor elevado em metal e lidar com ele em transações, os deixavam com pessoas confiáveis, as quais emitiam um papel certificando a quantia. Quem apresentasse tal recibo poderia retirar as moedas.

As primeiras emissões centralizadas de cédulas se iniciaram em 960, na dinastia Song, devido à escassez de cobre para cunhagem. Porém, foi apenas em 1274 que o governo central emitiu uma família de cédulas padronizadas a serem utilizadas em todo o império, representando uma certa quantia em ouro ou prata.[8] Em seus relatos de viagem, Marco Polo descreve o uso e emissão de cédulas na China, levando o conceito para a Europa no final do século XIII.[17]

No início do século XVII, ourives londrinos passaram a emitir recibos de depósito pagáveis ao portador. Com a popularização da prática, depositários pediam recibos menores, em valores determinados, para uso como dinheiro.[7] Estas são consideradas as primeiras cédulas modernas.

Pouco após esse processo, cédulas começaram a ser emitidas em maior quantidade do que havia estoques em metal, dando origem à reserva bancária fracional.

2.3 Sistema de Bretton Woods

Próximo do final da Segunda Guerra Mundial, as nações aliadas procuravam evitar uma repetição do Tratado de Versailles, que impôs severas punições à Alemanha e foi considerado culpado pelas tensões que geraram o conflito. Os estoques de ouro e prata dos países estavam baixos, o que impossibilitava o pagamento de dívidas internacionais com moeda forte. Desta forma, 730 representantes dos 44 países aliados se reuniram em Bretton Woods, no estado de New Hampshire, nos Estados Unidos, em julho de 1944 para discutir um novo sistema financeiro entre as nações. O resultado da conferência moldou a economia mundial pelas décadas vindouras, criando instituições como o Fundo Monetário Internacional e o Banco Mundial.

Pelo sistema de Bretton Woods, os Estados Unidos se comprometem a garantir a conversibilidade do dólar em ouro a um preço fixo, nominalmente US$ 35,00 por onça troy. Os países podem, desta maneira, utilizar a moeda estadunidense como reserva de valor, ao invés de manter estoques em metais.

Em 1971, o presidente dos Estados Unidos, Richard Nixon, extingue a garantia de conversão do dólar em ouro, no episódio que ficou conhecido como *Nixon Shock*. Isto levou a uma reorganização das moedas nacionais, principalmente trazendo o retorno do regime de câmbio flutuante. O fim do lastro metálico flexibilizou a impressão de moeda, possibilitando novas ferramentas de estímulo e desenvolvimento econômico nos países, além do desenvolvimento de novas classes de produtos financeiros.

Atualmente, não há mais moedas nacionais que utilizem lastros metálicos.

Capítulo 3

Descentralização

Descentralizar consiste em redistribuir os poderes ou funções atribuídos a uma autoridade ou entidade central. Este conceito pode ser aplicado a diversas áreas do conhecimento humano, podendo apresentar vantagens em relação a outras estruturas centralizadas.

O tema da descentralização tem ganho espaço graças ao aumento da penetração da internet e à queda dos preços de dispositivos computacionais. Novos modelos de produção e distribuição de mídias audiovisuais, softwares, além de conhecimento, de uma forma geral, apresentam-se como alternativa ao sistema centralizado de difusão de informação. Este tem sentido tais efeitos, que são especialmente fortes entre as grandes gravadoras, distribuidoras e estúdios de cinema.

Em seu livro *The Wealth of Networks*[2], o professor Yochai Benkler, do Berkman Center da Universidade de Harvard, argumenta que o aumento da disponibilidade do poder computacional nas mãos do indivíduo dá origem a um novo tipo de produção, a qual ele chama de *Commons-based peer production*, ou produção individual orientada ao domínio público, em uma tradução livre. Isto é possível através da motivação interna do ser humano da obtenção de prazer ao concluir tarefas que ele considere interessantes.

O aumento do poder computacional individual também representa um aumento do alcance que uma determinada mensagem pode obter. Neste sentido, Benkler faz uma analogia entre a disponibilidade de espectro para transmissão de informação através de ondas de rádio e o compartilhamento de espaço dentro das vias digitais.

A seguir, serão apresentados alguns exemplos de sistemas descentralizados.

3.1 ARPANET

Criada em 1969, a ARPANET foi a rede que originou muitos dos conceitos utilizados hoje em dia na internet, tendo sido uma das primeiras redes a utilizar a técnica de troca de pacotes, através do protocolo TCP/IP.

Como não há um servidor central para roteamento, os pacotes são repassados em um sistema de melhor esforço, na esperança de que o próximo nó da rede saiba para onde enviar. Isto aumenta a resistência a falhas, tornando a rede extremamente difícil de ser completamente desligada. Por se tratar de uma rede militar de computadores, esta era uma característica desejada.

Além disso, a descentralização favoreceu a escalabilidade, possibilitando o crescimento da internet até sua escala atual.

3.2 Wikipédia

Apesar de ter sido criada como um projeto de apoio a uma nova enciclopédia online chamada Nupedia, onde os artigos seriam revisados por especialistas nas respectivas áreas do conhecimento, a Wikipédia, lançada em 2001, ganhou mais apelo popular.

Fruto da colaboração de dezenas de milhares de pessoas ao redor do mundo, esta enciclopédia online propõe-se a armazenar todo o conhecimento humano e disponibilizá-lo gratuitamente a quem queira acessá-lo. A Wikipédia conta, atualmente, com mais de 34 milhões de artigos escritos em mais de 200 línguas.[18]

3.3 Software livre

O movimento do software livre teve seu surgimento oficial com o lançamento do *GNU Manifesto*, por Richard Stallman, em 1983. Nele, defende-se a criação de um sistema operacional completo e livre de barreiras de acesso a seu código-fonte. A razão disso foi a mudança nos termos da licença de uso dos programas de computador, em uma era que o software estava se tornando mais caro que o hardware.

Este manifesto organizou a comunidade de programadores que defendiam o livre acesso ao código-fonte, iniciando um ecossistema de aplicações utilizadas ainda hoje em dia, entre elas, o conjunto de ferramentas GNU, utilizado na compilação de softwares. Em 1991, um estudante de computação chamado Linus Torvalds anuncia que está escrevendo um sistema operacional e disponibiliza seu código-fonte para quem quiser ajudá-lo a fazer. Era o início do Linux, um software que iria dominar a indústria da informática décadas mais tarde.

3.4 Exemplos mais recentes

Com as recentes revelações de Edward Snowden a respeito do grau de monitoramento das telecomunicações internacionais por parte da agência de inteligência americana NSA (National Security Agency), novos projetos de descentralização e encriptação na transmissão e armazenagem de dados surgem.

Utilizando como base sistemas já estabelecidos, como o BitTorrent, desenvolvedores apresentam soluções para troca de mensagens instantâneas como o Firechat[1] ou o BitTorrent Bleep[2], além da distribuição mais amigável de mídias audiovisuais, como o Popcorn Time, entre inúmeros outros exemplos.

Uma nova plataforma para o desenvolvimento de aplicações descentralizadas, chamada Ethereum[3], está sendo confeccionada, com lançamento previsto para 2015.

[1] `https://opengarden.com/firechat`
[2] `http://labs.bittorrent.com/bleep/`
[3] `https://ethereum.org`

Capítulo 4

Bitcoin

Desde sua apresentação em 2008, o bitcoin tem sido um conceito debatido por diversas autoridades governamentais e privadas. Baseado em uma tecnologia chamada *blockchain*, ou cadeia de blocos, ele tem sido utilizado como um sistema de pagamentos via internet que não utiliza bancos ou quaisquer outras entidades centrais.

Neste capítulo, será descrita um pouco da história e do funcionamento do bitcoin, largamente baseado no *whitepaper* escrito por Satoshi Nakamoto em outubro de 2008.

4.1 Breve história das moedas digitais

A primeira discussão sobre criptografia em meios eletrônicos de pagamento surge em 1982 com um artigo de David Chaum a respeito de assinaturas cegas para pagamentos eletrônicos não rastreáveis.[4] Nele, o autor propõe um sistema onde chaves criptográficas validam notas assinadas por uma entidade certificadora, no caso, um banco, possibilitando transferir valores através de redes de computadores. Este conceito deu origem a uma empresa chamada DigiCash em 1990, que até 1998 competiu no mercado de pagamentos online, quando pediu falência, devido à preferência dos usuários pelo uso do cartão de crédito.[16] Uma das características do sistema de Chaum era a possibilidade de efetuar micropagamentos, ou seja, a transferência de centavos ou frações.

Após o fim da DigiCash, moedas eletrônicas atraíram atenção através de uso do Linden dollar no Second Life, um software que provê acesso a um mundo tridimensional online. Desde 2003, os usuários podem trocar produtos e serviços, dentro do sistema, por lindens, emitidos pela empresa responsável, a Linden Lab. Existem bolsas para que os chamados residentes possam comprar e vender a moeda por dólares, euros ou outras denominações monetárias. Em 2009, o volume de transações chegou a movimentar US$ 567 milhões no ano.[10]

Em outubro de 2008, um autor chamado Satoshi Nakamoto[1] publica uma proposta para um sistema eletrônico de pagamentos descentralizado em uma lista de e-mails de *cypherpunks*[2]. A sua principal diferença é a ausência da necessidade de confiança em uma autoridade, garantindo a propriedade e unicidade dos valores através de chaves criptográficas. O primeiro nó da rede bitcoin surge em janeiro de 2009, dando início ao *blockchain*.

[1] Não existe informação se este é um nome real ou um pseudônimo de um indivíduo, ou até mesmo um grupo.

[2] *Cypherpunks* são ativistas que pregam o uso de criptografia para promoção da privacidade e como fator de mudanças sociais e políticas.

4.2 Conceitos básicos

O bitcoin é produto do uso intenso de criptografia de chave pública através de um banco de dados distribuído. A segurança do sistema é garantida pelo processo de mineração. Porém, antes é necessário definir alguns conceitos que circundam a criptomoeda:

Satoshi Em homenagem à figura criadora do bitcoin, a unidade básica, que vale 10^{-8} bitcoin, é chamada de satoshi.

Carteira Carteira é o nome dado à coleção de pares de chaves criptográficas públicas e privadas, cujas assinaturas identificam a propriedade de um valor em bitcoins. Estas são, geralmente, arquivos armazenados em computadores pessoais, dispositivos móveis, servidores na internet ou em *QR codes* impressos em papel.

Endereço de carteira Para cada chave criptográfica pública, há um endereço de carteira bitcoin associado. São representados por um identificador entre 26 e 34 caracteres ASCII. Um exemplo de endereço é `168nTVYNAKdHmoEiYKXcTZG34e6r7V83m7`. Uma carteira pode conter vários endereços.

Blocos Blocos são a unidade básica de funcionamento do bitcoin. Eles são responsáveis pela emissão de moeda e processamento de transações, sendo o objetivo da mineração. Também contêm uma coleção de transações não presentes em nenhum outro bloco, além de informações que identificam o seu antecessor no *blockchain*.

Blockchain O *blockchain*, ou cadeia de blocos, é o registro público imutável de todas as transações já efetuadas na rede bitcoin. Ele consiste em uma sequência de blocos, desde o chamado bloco *genesis*[3], que deu início à cadeia. Este é o banco de dados distribuído utilizado.

Mineração A mineração de blocos é o processo através do qual as transações bitcoin são verificadas e armazenadas no *blockchain*. Para isso, é necessário o uso de computadores que se dediquem à busca por blocos, cuja recompensa consiste em uma quantia determinada em bitcoins. Este procedimento será discutido na seção 4.6.

Transação Uma transação bitcoin é a transferência de valores entre chaves criptográficas. Mais especificamente, consiste em assinar uma transação de propriedade do remetente com a chave pública do destinatário, dada pelo endereço da carteira bitcoin.

Codificação em base 58 Para simplificar a representação de inteiros muito grandes, estes são convertidos para a base 58. Ela é similar à base 64, porém, com modificações para evitar a utilização de caracteres não-alfanuméricos e de letras que sejam ambíguas quando impressas. O bitcoin utiliza a seguinte coleção de caracteres:

`123456789ABCDEFGHJKLMNPQRSTUVWXYZabcdefghijkmnopqrstuvwxyz`

O número `pataca`, em base 58, é representado, em base 10, por 31.232.276.251.

[3] O bloco *genesis* não foi minerado, mas sim escrito à mão pela figura de Satoshi Nakamoto.

Nó da rede bitcoin Um nó da rede bitcoin é um computador que executa uma instância de um serviço que implementa o protocolo bitcoin. Através dele, é possível gerar e armazenar chaves criptográficas, transmitir transações e buscar blocos.

Os nós se comunicam através da internet, repassando transações entre si, fazendo com que todos tenham um registro completo e sincronizado de operações realizadas. Os nós verificam se as assinaturas estão corretas e atualizam seus respectivos saldos associados às chaves criptográficas armazenadas localmente.

4.2.1 Estrutura de um bloco

Um bloco contém informações a respeito das transações transmitidas através da rede bitcoin desde o último bloco encontrado. Ele consiste dos campos:

Campo	**Descrição**	**Tamanho**
Número mágico[4]	Sempre `0xD9B4BEF9`	4 bytes
Tamanho do bloco	Número de bytes até o fim do bloco	4 bytes
Cabeçalho	6 outros campos, descritos a seguir	80 bytes
Contador de transações	Inteiro positivo	1-9 bytes
Transações	Uma coleção não-vazia de transações	Variável

Tabela 4.1: Campos presentes em um bloco.

Para o cabeçalho, há os seguintes campos:

Campo	**Descrição**	**Tamanho**
Versão	Versão do bloco	4 bytes
Hash do bloco anterior	Hash de 256 bits do cabeçalho do bloco anterior	32 bytes
Hash da raiz de Merkle	Hash de 256 bits baseado em todas as transações	32 bytes
Timestamp	Segundos desde a Era Unix[5]	4 bytes
Alvo	Número alvo para o valor do hash	4 bytes
Nonce[6]	Número de 32 bits	4 bytes

Tabela 4.2: Campos presentes no cabeçalho de um bloco.

[4]Um número mágico identifica o tipo de informação que o segue.
[5]Número de segundos desde a meia-noite de 1 de janeiro de 1970, em UTC.
[6]Um *nonce* é um número utilizado apenas uma vez em um dado contexto.

4.2.2 Estrutura das transações

As transações possuem os seguintes campos:

Campo	Descrição	Tamanho
Versão	Versão do bloco	4 bytes
Contador de entradas	Inteiro positivo	1-9 bytes
Entradas	Coleção de entradas	Variável
Contador de saídas	Inteiro positivo	1-9 bytes
Saídas	Coleção de saídas	Variável
Expiração	Se diferente de zero e menor que 0xFFFFFFFF, número do bloco ou *timestamp* que expira a transação.	4 bytes

Tabela 4.3: Campos presentes em uma transação bitcoin.

As entradas possuem os seguintes campos:

Campo	Descrição	Tamanho
Hash da transação anterior	Hash duplo SHA-256 de uma transação anterior	32 bytes
Índice de saída	Índice do registro de saída a ser utilizado da transação dada	4 bytes
Tamanho do script	Tamanho do script do registro	1-9 bytes
Script	Script para esta entrada	Variável
Número de sequência	normalmente 0xFFFFFFFF, irrelevante se a expiração for igual a zero	4 bytes

Tabela 4.4: Campos presentes em cada registro de entrada de transação.

As saídas possuem os seguintes campos:

Campo	Descrição	Tamanho
Valor	Valor, em satoshis, a ser transferido	8 bytes
Tamanho do script	Tamanho do script do registro	1-9 bytes
Script	Script para esta entrada	Variável

Tabela 4.5: Campos presentes em cada registro de saída de transação.

Troco

Como os valores movimentados pelas transações são, na realidade, divisões e agrupamentos de outras transações, que possuem como raiz primordial uma transação de geração de moeda, existe a figura do troco. Esta é representada através de um envio a um endereço de carteira de posse do remetente da transação.

Para exemplificar o funcionamento do troco em transações bitcoin, vamos supor que Alice possua um endereço de carteira onde recebeu duas transações, uma de 0,5 bitcoin e outra de 0,7 bitcoin, totalizando 1,2 BTC de saldo. Ela deseja enviar 1 bitcoin a Beto. Para tanto, ela cria uma nova transação e usa como entradas as duas transações recebidas. Como saídas, Alice envia 100.000.000 satoshis (ou 1 bitcoin) ao endereço informado por Beto, além de 20.000.000 satoshis (ou 0,2 bitcoin) para seu endereço original, ou ainda um outro endereço de sua propriedade.

Note que o valor total das entradas é igual ao valor total das saídas, mas nem sempre este é o caso.

Taxa

Para estimular o minerador a processar transações, é comum oferecer um determinado valor em bitcoins, conhecido como *taxa*. Ela é dada pela diferença entre o valor total das entradas e o valor total das saídas.

No processo de busca de um novo bloco, todas essas diferenças são somadas e enviadas a um endereço informado pelo minerador.

4.3 Funções de hash

Uma função de hash é qualquer função que mapeie informações de tamanho arbitrário para um intervalo determinado. Para a classe de funções de hash utilizadas no bitcoin, algumas propriedades são desejáveis:

(a) Determinismo
Sejam F e G informações, h uma função de hash, F_h e G_h o resultado das operações $h(F)$ e $h(G)$, respectivamente. $F_h = G_h$ se, e somente se, $F = G$.

(b) Uniformidade
Uma boa função de hash mapeia informações de forma equiprovável dentro de seu intervalo discreto.

(c) Intervalo bem definido
As informações mapeadas pela função de hash devem cair dentro de um intervalo definido pelas próprias características da função. Este intervalo é, geralmente, dado pelo número de bits. Por exemplo, uma função de hash de 256 bits mapeia informações para naturais entre 0 e 2^{256}.

Hash criptográfico Funções de hash criptográfico são especialmente úteis para verificar a consistência de informações, já que possuem a propriedade de serem não-inversíveis, ou seja, dado um valor de hash F_h, não é possível obter F sem gastar uma enorme quantidade de tempo de processamento.

SHA-256 SHA-256 é uma função de hash criptográfico da família SHA-2, publicada originalmente em 2001 pela *National Security Agency* (NSA).[11] Ela produz valores de saída com 256 bits de comprimento e é largamente utilizada pelo protocolo bitcoin.

A palavra em ASCII `pataca` produz como saída o seguinte número hexadecimal:
`5bd8fe90f9998ceaf0b08f43fff10e7393a21e32a1dfc8d9da6f4a9d68be4d98`

RIPEMD-160 Assim como o SHA-256, o RIPEMD-160 é uma função de hash criptográfico, publicada em 1996 pelos pesquisadores Hans Dobbertin, Antoon Bosselaers e Bart Preneel da *Katholieke Universiteit Leuven.*[5] Como saída, produz valores de 160 bits, e também é utilizada pelo protocolo bitcoin.

A palavra em ASCII `pataca` produz como saída o seguinte número hexadecimal:
`816dc3518886581869df3267999a97229ad2f09e`

Árvore de Merkle Uma árvore de Merkle é uma estrutura onde as folhas são os dados a serem processados e os nós são os valores de hash de seus filhos. Desta forma, a raiz da árvore é um valor de hash de todos os dados presentes na árvore. Esta estrutura apresenta a característica de que, para verificar se um determinado dado está presente em uma árvore com n folhas, apenas é necessário realizar $log(n)$ operações.[13]

4.4 Geração de endereços de carteira

Os endereços de carteira bitcoin são valores de hash de 160 bits da parte pública do par de chaves criptográficas geradas. Estas são criadas utilizando uma variante do *Digital Signature Algorithm* (DSA) que usa curvas elípticas, chamada *Elliptic Curve Digital Signature Algorithm*, ou ECDSA.

O procedimento para geração é descrito a seguir:

1. Dada uma chave privada ECDSA K, pegue a parte pública K_p;
2. Aplique SHA-256 em K_p, obtendo um valor $SHA(K_p)$;
3. Aplique RIPEMD-160 em $SHA(K_p)$, obtendo $RIPE(SHA(K_p))$;
4. Concatene o byte da versão do endereço[7] junto com $RIPE(SHA(K_p))$, obtendo $E = v + RIPE(SHA(K_p))$;
5. Aplique SHA-256 duas vezes em cima do resultado E, obtendo $SHA^2(E)$;
6. Pegue os primeiros 4 bytes de $SHA^2(E)$, que serão utilizados como dígitos verificadores, e concatene com E, obtendo $E + SHA^2(E)[0:4]$;
7. Converta o resultado para a base 58.

4.5 *Proof-of-work*

O sistema *proof-of-work*, ou prova de trabalho, é um modelo proposto por Dwork e Naor[6], apresentado na 12ª Conferência Internacional de Criptologia, em 1992. Originalmente, foi idealizado para combater o envio de e-mails indesejados[8] através da apresentação de prova de que uma certa quantidade de trabalho foi realizada, a fim de que o destinatário aceite a mensagem. Este sistema foi posteriormente implementado em 1997, chamado Hashcash.

4.5.1 Hashcash

O Hashcash, criado por Adam Back[1], adiciona uma entrada no cabeçalho da mensagem eletrônica, cujo valor de hash, utilizando o algoritmo SHA-1, deve iniciar com uma certa quantidade de bits de valor zero. Esta condição faz com que o remetente gaste processamento buscando satisfazê-la.

O e-mail que foi processado pelo Hashcash possui uma entrada em seu cabeçalho no seguinte formato:

```
X-Hashcash:  <versão>:<bits>:<data>:<recurso>:[<ext>]:<aleatório>:<contador>
```

[7] O valor padrão é 0x00, porém, pode ser diferente, caso esteja na rede de testes.
[8] Uma observação interessante é que o chamado *spam* é um problema a mais de duas décadas.

onde:

`X-Hashcash` Nome do cabeçalho.

`<versão>` Versão do sistema, o padrão é 1.

`<bits>` Número de bits zero no início do hash.

`<data>` Data no formato YYMMDD.

`<recurso>` String do recurso: número de IP, e-mail etc.

`<ext>` *(opcional)* Campo de extensão do recurso, não utilizado na implementação padrão.

`<aleatório>` Número aleatório para cada instância de uso.

`<contador>` Campo incrementado na busca de um valor de hash que satisfaça a condição.

Exemplo Suponha que *bar* deseja enviar uma mensagem a *foo* e ambos utilizem o sistema Hashcash. Para que *foo* aceite a mensagem, é necessário que *bar* apresente um cabeçalho cujo valor de hash possua um prefixo de zeros de um certo tamanho mínimo. *Foo* aceita mensagens com 24 bits de prefixo e *bar* sabe disso, portanto, introduz o seguinte cabeçalho `X-Hashcash` na mensagem:

```
1:24:040806:foo::511801694b4cd6b0:1e7297a
```

Foo, ao receber a mensagem, verifica a existência do dado cabeçalho e imediatamente aplica o algoritmo SHA-1 no conteúdo, obtendo o valor hexadecimal:

```
0000008e3caaff34c595a55f7cc53c6b1cd385d1
```

Como nesta representação cada dois caracteres ASCII correspondem a um byte hexadecimal, é fácil verificar que o valor de hash da mensagem possui, de fato, 24 bits de valor zero em seu prefixo, ou três números hexadecimais `00`, portanto, é uma mensagem aceita por *foo*.

Após a verificação do prefixo de bits, *foo* lê o conteúdo do cabeçalho. Após verificação da versão da mensagem, `1`, e do número de bits zero informado, `24`, *foo* checa se a mensagem está em seu período de validade, olhando o campo `data`, cujo valor é `040806`, ou seja, 6 de agosto de 2004. O prazo é definido por *foo* e é comparado com a data de recebimento da mensagem. Caso a mensagem não esteja dentro do mesmo, ela é rejeitada. Como *foo* aceita mensagens com até 2 dias, e ele a recebeu no próprio dia 6, portanto é válida. Finalmente, verifica-se a string do recurso e confere que a mensagem é endereçada a *foo*. ◁

4.6 Mineração

O processo da mineração de bitcoins é o processo de construção do *blockchain*, o que significa buscar que o valor de hash das informações de um novo bloco que se queira encontrar seja menor ou igual que o valor dado pelo campo *alvo* do bloco anterior.

Primeiramente, é necessário obter o valor da raiz da árvore de Merkle das transações recebidas pelo nó. Como recompensa pelo trabalho de processar informações, é permitido que a primeira transação do bloco, chamada *coinbase*, não tenha entradas, recebendo, na saída informada, a criação de uma certa quantidade de bitcoins, dada pela soma da recompensa atual da rede e das taxas de transação pagas pelos usuários.

4.6.1 Parâmetros

Para realizar a busca de blocos, é necessário atentar-se a alguns parâmetros:

Altura do bloco Nome dado ao número do bloco atual. O bloco *genesis* possui altura 0 e seu sucessor possui altura 1.

Alvo Limiar do valor de hash, deve ser menor ou igual a este valor.

Intervalo entre blocos Tempo médio entre descobertas de blocos.

A altura do bloco, representada por H, é utilizada para determinar a recompensa atual da rede e quando o alvo, representado por d, deve ser reajustado, levando em consideração o intervalo entre blocos, dado por Δ_b.

Recompensa

A recompensa R consiste em uma quantidade de bitcoins dada ao descobridor de um bloco. Este é um estímulo para que haja mineradores interessados em processar as transações da rede bitcoin.

O valor, reajustado a cada 210.000 blocos, é dado pela seguinte equação:

$$R = \frac{50}{2^{\lfloor \frac{H}{210000} \rfloor}} \tag{4.1}$$

Ela continuará existindo até que $R < 10^{-8}$, ou 1 satoshi. Isto ocorrerá quando $\lfloor \frac{H}{210000} \rfloor = 33$, o que significa $H = 6930000$.

Reajuste do alvo

Para que a rede possa se ajustar ao poder de processamento presente, a cada 2016 blocos o alvo é reajustado. Dado que $\Delta_b = 10$ minutos, o reajuste dá-se, em média, a cada 14 dias:

$$\Delta_d = 10 \times 2016 \times \frac{1}{60} \times \frac{1}{24} = 14 \text{ dias} \tag{4.2}$$

Nesta ocasião, todos os nós da rede contam quantos blocos foram encontrados nos últimos 14 dias e modificam o alvo da seguinte maneira:

$$d_{\text{novo}} = d_{\text{antigo}} \frac{2016}{n_{\text{blocos}}} \tag{4.3}$$

A equação acima mostra que, caso tenham sido encontrados menos que 2016 blocos nos últimos 14 dias, o alvo aumenta, tornando mais fácil o processo de mineração. O caso de mais blocos terem sido minerados é análogo.

4.6.2 Procurando blocos

A procura por blocos dá-se calculando o valor de hash b_h da concatenação de strings hexadecimais *little endian* dos campos do cabeçalho do bloco, dados pela tabela 4.2.1. Um bloco é encontrado quando $b_h < d$. Desta forma, para minerar incrementa-se o campo *nonce* iterativamente, até que a condição seja satisfeita.

Probabilidade de sucesso

Como a função de hash criptográfico utilizada possui as propriedades citadas na seção 4.3, a probabilidade de encontrar um valor de hash v menor que um dado número d funciona tal qual uma distribuição discreta uniforme. Como a função utilizada retorna um valor de 256 bits, há 2^{256} resultados equiprováveis.

Voltando ao caso de encontrarmos $v < d$ neste espaço amostral, temos:

$$\mathbb{P}(v < d) = \frac{d}{2^{256}} \tag{4.4}$$

A busca do valor de hash se comporta como uma distribuição geométrica, com a esperança da variável aleatória X, representando a quantidade de tentativas até o primeiro sucesso, sendo:

$$E[X] = \frac{1}{\mathbb{P}(v < d)} = \frac{1}{\frac{d}{2^{256}}} = \frac{2^{256}}{d} \tag{4.5}$$

Capítulo 5

Pataca

Neste capítulo, será descrito a peça de software desenvolvida durante este Trabalho de Conclusão de Curso.

5.1 Conceito

Em nosso cotidiano, acessamos diversas micro-economias sem nos darmos conta. Desde as mais óbvias, como programas de pontos de fidelidade de estabelecimentos comerciais, participamos de outras, como a utilização de vales-desconto e cupons; a compra de tickets individuais, no caso dos restaurantes universitários, ou temporais, como no bilhete único mensal; e até mesmo para o empréstimo de livros em bibliotecas, onde pode até ocorrer um pequeno mercado[1].

A aquisição de moedas específicas para cada micro-economia dá-se por diversos meios, entre eles, a troca de moeda nacional por *tokens* – moedas, tickets, crédito – ou a concessão de acesso a algum bem ou serviço provido através do sistema apresentando algum tipo de credencial.

O sistema Pataca se foca em micro-economias baseadas em *tokens*, ou seja, onde o indivíduo possa adquirí-los e negociar bens e serviços por meio deles. Estes podem possuir algum tipo de reconhecimento externo de valor (lastro) ou serem isolados.

5.2 Tecnologias

O Pataca foi escrito em Python 2.7, utilizando como base o framework Twisted Matrix para realizar a comunicação com a rede e o framework SQLAlchemy para realizar o mapeamento objeto-relacional entre o código e o banco de dados.

5.2.1 Python 2.7

A utlização do Python foi uma escolha natural, dada a fluência do autor nesta linguagem. É importante citar a versão utilizada, já que existem dois troncos distintos de desenvolvimento: um que segue nas versões 2.x e um novo em versões 3.x. Esta diferenciação ocorreu devido à mudança de alguns conceitos básicos na linguagem, que quebram a compatibilidade retroativa.

[1] É comum observar pessoas pegando livros umas para as outras, ou emprestando entre si, quando o limite de empréstimos de um determinado indivíduo estourou, ou quando o mesmo está sob período de penalidade, popularmente conhecido como *gancho*.

Um objetivo deste trabalho, ainda que secundário, é possibilitar que o sistema Pataca possa rodar, sem grandes penalidades, em hardwares obsoletos ou do chamado *good enough computing*, que consiste em dispositivos cujo poder computacional não está nas fronteiras do desempenho da tecnologia atual, mas é "bom o bastante" para realizar uma determinada tarefa. Neste caso, o Raspberry Pi apresenta uma plataforma para desenvolvimento e testes cuja linguagem-padrão é, justamente, Python.

Custando 25 ou 35 dólares, dependendo da versão, o Raspberry Pi é uma solução de baixo custo com potência suficiente para rodar diversas aplicações, especialmente as escritas em Python. O dispositivo apresenta processador ARM11 de 1GHz e 256MB ou 512MB de memória RAM, com performance comparável a um Pentium II de 300MHz.

5.2.2 Twisted Matrix

O Twisted Matrix é um framework orientado a eventos que possibilita trabalhar com diversos protocolos de comunicação, como HTTP, SSH, Telnet, entre outros, além da escrita do próprio. Dentro do Pataca, ele é responsável por toda a comunicação com a rede, principalmente HTTP e HTTPS.

Durante a fase de prototipagem, foi utilizado o framework Django, que se orienta mais para desenvolvimento rápido web, não possuindo os controles finos de uso de protocolos que o Twisted Matrix possui. A substituição foi feita por limitações do próprio Django, além de performance.

5.2.3 SQLAlchemy

Como uma parte importante do Pataca, os módulos, podem ser escritos e distribuídos pelos usuários, o acesso direto ao banco de dados através de declarações SQL foi descartada. Ao invés disso, é utilizado um mapeador objeto-relacional, no caso, o SQLAlchemy, que realiza todas as operações de criação de tabelas e consultas aos dados.

O framework mapeia classes para tabelas. As classes herdam de uma classe-base, cuja qual possui métodos de acesso que adicionam funcionalidades às instâncias, facilitando o uso no código e deixando a comunicação com o banco de dados transparente.

O Pataca implementa o SQLAlchemy com poucas alterações, disponibilizando as estruturas que o caracterizam ao desenvolvedor de módulos, como sessões, métodos de consulta, entre outros.

5.3 Acesso

O acesso ao sistema Pataca é feito através de chamadas de API REST. Isto separa a camada de apresentação da camada de aplicação, possibilitando, também, integração com outros sistemas de forma transparente.

É importante citar que é **altamente recomendável** a utilização de um certificado SSL no Pataca, a fim de que os dados trafeguem na rede através de um canal seguro.

5.3.1 Autenticação

A autenticação no Pataca é baseada em *token*, emitido quando o usuário apresenta credenciais válidas. Estes *tokens* possuem validade, dada pelo servidor, e devem ser renovados se não utilizados após algum tempo.

Após a aquisição do *token* pelo usuário, este necessita anexá-lo no cabeçalho `Authorization` de todos seus pedidos POST, da seguinte forma:

```
Authorization: Token <token>
```

onde `<token>` é o valor recebido ao se autenticar no sistema.

5.3.2 Operações

Cada instância do sistema Pataca possui um conjunto de operações que suporta. Estas são acessadas através de requisições GET ou POST feitas ao servidor em uma URI[2] específica, podendo ser a raiz.

O funcionamento destas é descrito na seção seguinte.

5.4 Módulos

Todo o sistema Pataca é constituído de módulos. Estes podem ser internos ao sistema, como o sistema de contas e o de autenticação, ou externos, como a API, o lastro e a interface. Os módulos internos não podem, *a priori*, ser atualizados ou intercambiados da mesma forma que os módulos externos, que são subdiretórios e possuem uma estrutura básica bem definida.

5.4.1 Estrutura básica

Os módulos externos adicionam funcionalidades ao sistema na forma de chamadas de API a uma URI informada. Isto é feito através da leitura do arquivo `settings.py`.

Estes arquivos possuem duas variáveis fundamentais:

uri Nome da URI do módulo. Caso seja vazia, representa a raiz.

operations Uma lista de operações, dadas pela função `operation`, que recebe dois argumentos:

1. Nome da operação;
2. Ponteiro de função.

O carregador de módulos do Pataca lê este arquivo e adiciona em sua árvore de processamento as entradas correspondentes. Quando um pedido chega para uma operação em específico, a função indicada é chamada.

5.4.2 Processando pedidos

Cada módulo possui um conjunto de funções, onde cada uma responde a uma certa operação. O roteador de pedidos chama a função específica para uma dada operação informada. Como argumentos, são repassados o objeto `request`, do próprio Twisted Matrix, e um dicionário `data`, com os campos do pedido e seus respectivos dados.

O módulo de autenticação também provê um decorador de função[3] `@auth_required` que realiza a verificação da presença e validade do *token* de acesso. Caso não ocorra erros, adiciona um argumento `user`, com uma instância do objeto que representa o usuário.

[2] *Uniform Resource Identifier.*

[3] Decoradores encapsulam chamadas de função, podendo modificar a pilha ou realizar verificações antes que a própria função seja chamada.

5.5 Funcionalidades

Nesta seção, serão descritas algumas funcionalidades que o sistema Pataca apresenta. Exceto as internas, todas são feitas através de módulos no formato descrito na seção anterior.

5.5.1 Internas

As funcionalidades internas são características intrínsecas do Pataca. Abaixo, algumas:

Mapeamento objeto-relacional

O mapeamento objeto-relacional, dado pelo SQLAlchemy, remove a preocupação do desenvolvedor em conhecer a estrutura do banco de dados em questão para realizar consultas, criar, atualizar ou remover dados.

A partir dos dados presentes no arquivo de configuração geral do Pataca, a conexão com o banco de dados é feita. Diversas implementações são suportadas, algumas são SQLite, PostgreSQL, MySQL, SQL Server e Oracle. É possível também iniciar o Pataca utilizando um banco de dados em memória, para pequenos testes.

Para utilizar a funcionalidade, basta declarar uma classe que herda de `db.models.Model`, que será automaticamente incluída no registro de tabelas do Pataca. Caso o banco de dados ainda não possua tabelas, é possível pedir ao Pataca para criá-las a partir dos modelos.

Perfil

Cada usuário do sistema possui um perfil, onde constam suas informações de login. É possível designar *tags* aos perfis, identificando diferenças de uso geral entre os usuários.

Os perfis armazenam, também, as chaves de acesso, ou *tokens*, de cada usuário e suas respectivas horas de emissão e último uso.

Contas

Todo o sistema Pataca se organiza ao redor das contas. Elas são coleções de transações, as quais geram saldos. Cada conta pertence a uma classe e é de uma determinada moeda[4]. Contas possuem um identificador único e podem pertencer a mais de um usuário.

Contas não podem ter saldos negativos.

Moedas

O Pataca suporta a existência de diversas moedas locais no sistema. *A priori*, não é possível transferir entre contas de moedas diferentes.

5.5.2 Lastro

O lastro é um estoque de valor que uma moeda existente no Pataca pode possuir. Com ele, o valor intrínseco de cada unidade monetária corresponde a uma fração do total de valor armazenado. No Pataca, isto está implementado como uma quantia em bitcoins.

Como há um lastro para cada denominação monetária, é possível definir uma taxa de câmbio para cada uma. Isto é feito dividindo o valor do lastro pela quantidade total de moeda no sistema.

[4] O nome correto seria "denominação", já que ela pode não corresponder a algum tipo de bem localmente negociável. Porém, por motivos de clareza, chamaremos de "moeda" para dar a dimensão correta.

Dada uma moeda M, com n contas de saldo S_i^M, e um valor de lastro L_M, obtemos a taxa de câmbio T_x:

$$T_x(M) = \frac{L_M}{\sum_{i=1}^{n} S_i^M} \tag{5.1}$$

Armazenamento do lastro

O lastro em bitcoins é armazenado, nesta versão inicial, em uma carteira do serviço Blockchain.info[5]. É possível armazenar localmente, instalando o servidor bitcoin e modificando o módulo, acrescentando as chamadas RPC correspondentes.

A razão desta decisão em hospedar externamente vem do objetivo primário de fazer o Pataca rodar em Raspberry Pi, que possui memória, processamento e disco bastante limitados. No dia 12 de outubro de 2014, o *blockchain* do bitcoin já ultrapassava 22GB[6].

Outros lastros

Não há impedimentos para que outros tipos de lastro sejam implementados.

5.5.3 Protocolo

O protocolo Pataca permite, através do uso de lastros, a transferência de valores entre contas que se encontrem em servidores diferentes. Isto é feito através de transações que não requerem confiança, como as presentes na rede bitcoin.

Para realizar transações entre servidores, o Pataca utiliza um sistema de federação no formato `<identificador da conta>@<endereço do servidor>`. Vamos analisar um exemplo:

Exemplo: Alberto possui uma conta, em patacas, num servidor das Organizações Foo e deseja enviar 500 patacas para Bernardo, cuja conta, em estalecas, uma outra moeda, está no servidor do Bar Bar, um popular ponto-de-encontro local.

1. Alberto pergunta qual o endereço da conta de Bernardo, que informa `bernardo@bar.net`;
2. Alberto inicia uma transferência de 500 patacas para `bernardo@bar.net` a partir de sua conta `alberto@foo.org`;
3. O servidor `foo.org` bloqueia o saldo correspondente e notifica o servidor em `bar.net` sobre a transferência, pedindo um endereço de carteira bitcoin que represente a conta `bernardo@bar.net`;
   (a) Caso o servidor `bar.net` não implemente o lastro em bitcoin, a transferência é abortada.
4. O servidor `bar.net` responde a `foo.org` com o endereço e aguarda;
5. Com o endereço bitcoin de destino, `foo.org` verifica a taxa de câmbio, no caso, 0,0002 bitcoin por pataca, envia a quantia de 0,1 bitcoin para a carteira, ou 500 patacas, e notifica `bar.net` da transação;

[5] `http://blockchain.info/`
[6] `https://blockchain.info/charts/blocks-size/`

6. Ao receber a transação, e após um certo número de confirmações da rede bitcoin, `bar.net` credita a conta de Bernardo com 125 estalecas, dado que a taxa de câmbio é de 0,0008 bitcoin por estaleca.

5.6 Expansões possíveis

Devido à característica modular do sistema Pataca, ele pode ser estendido para uma série de novas funcionalidades, tais como:

1. inventários de usuário;
2. mercados;
3. bolsas de troca entre moedas;
4. lojas de módulos, análogas a Play Store e App Store;
5. sistemas de banco central (transferência sem lastro).

5.7 Implementação e desenvolvimento

O desenvolvimento do Pataca é contínuo e pode ser acompanhado através de sua presença online:

Website `http://pataca.io`

Twitter @gopataca

Github `https://github.com/pataca`

Capítulo 6

Outros projetos e impressões

Este capítulo congrega experiências e projetos que me envolvi no ano de 2014 que contribuíram com a confecção do Pataca ou são relacionados a ele. Além disso, consta uma apreciação pessoal e crítica do trabalho do TCC.

6.1 Mercado Bitcoin

O Mercado Bitcoin[1] é a maior bolsa de bitcoins da América Latina, movimentando aproximadamente 3 milhões de reais por mês[3] em negociações de moedas criptográficas. Tive a oportunidade de conhecer a equipe durante a Campus Party 2014, onde fui o primeiro a utilizar o caixa eletrônico que havia sido levado.[2]. A partir de então, trocamos contato e fui contratado como estagiário no final do mês de março.

Durante o ano, lidei diariamente com moedas criptográficas e seus desafios técnicos na manutenção de uma bolsa de valores com dezenas de milhares de usuários. O grande projeto do período foi a reformulação estética e estrutural do website, onde fui responsável pela confecção do novo módulo de autenticação e perfil de usuários, entre outras tarefas. Diversas outras funcionalidades também foram implementadas ao longo do ano.

Além do aprendizado técnico, uma das grandes experiências em trabalhar no Mercado Bitcoin é a possibilidade de vivenciar o cenário de startups brasileiras e internacionais.

6.2 Stellar Foundation

A Stellar Foundation[3] é uma organização sem fins lucrativos cujo objetivo é aumentar a mobilidade do dinheiro entre as pessoas. Ela mantém a rede Stellar, uma infra-estrutura pública que possibilita o envio quase instantâneo de dinheiro entre instituições financeiras, empresas ou indivíduos, de maneira análoga ao sistema SWIFT[4], exceto pelo prazo.

O Stellar, ao contrário de bitcoin, não se baseia em criptografia, mas sim em consenso. Além da própria moeda interna do sistema, há como emitir outras moedas e negociá-las no sistema interno de livro de ofertas, com resolução automática de negociações.

[1] `https://www.mercadobitcoin.net`

[2] `http://tecnologia.ig.com.br/2014-01-28/campus-party-maquina-de-bitcoins-faz-primeira-transacao.html`

[3] `https://www.stellar.org/`

[4] `http://www.swift.com/`

Participamos do projeto, incluindo a escolha do nome do algoritmo de consenso, Firmeza, pouco depois de sua concepção. Um de seus desenvolvedores, Andrew Rogers, passou uma semana em nosso escritório a título de intercâmbio de idéias e soluções. Antes de trabalhar com moedas digitais, Rogers foi gerente do time do Android Auto e trabalhou na tecnologia NFC da Google Wallet.

6.3 Sabir

O projeto Sabir[5] consiste em um projeto semelhante ao Pataca, porém para o que Yochai Benkler chama de *Commons-based peer-production*[2]. Baseado no Ethereum[6], Sabir se propõe a ser uma ponte entre o mercado e o domínio público.

Este trabalho é fruto dos pesquisadores Primavera De Filippi, do Berkman Center da Universidade de Harvard, e de Samer Hassan, da Universidad Complutense de Madrid, e continua em desenvolvimento. O Prof. Gilson Schwartz também teve a oportunidade de conhecer De Filippi durante o Encontro Internacional Culturas e Tecnologias Digitais, promovido pelo SESC São Paulo.

6.4 Uma apreciação pessoal e crítica

Foi bastante gratificante a realização do projeto do Pataca, apesar de todos os percalços enfrentados na realização deste trabalho de conclusão. Os andamentos dos trabalhos do segundo semestre foram prejudicados devido a fatores fora de meu controle, mas o cerne do trabalho, o servidor Pataca, encontra-se escrito.

Um grande empecilho na feitura do projeto foi o sistema de módulos, onde perdi algumas semanas até chegar a uma estrutura que considerei simples e fácil de ser utilizada. Infelizmente, os módulos adicionais pensados, tais como interface web, mercado de módulos (como o Google Play ou App Store), não ficaram prontos. Porém, como este é um projeto com alguns usos já em mente, o desenvolvimento continuará.

Restou, também, uma enorme vontade de continuar pesquisando sobre moedas, um assunto que já me interessava há muitos anos. Outro interesse recém-descoberto são os diversos aspectos da criptografia e técnicas para garantir a segurança de informações.

6.4.1 O trabalho e sua relação com o curso

A realização deste trabalho de conclusão de curso necessitou a utilização de uma grande carga de conhecimento adquirida durante a graduação em Ciência da Computação. Apesar de já programar antes de entrar no curso, sinto que hoje em dia há uma maturidade maior em relação à projeto e escrita de código.

Abaixo, uma relação de matérias que considerei fundamentais para a confecção deste TCC:

1. *MAC0211 – Laboratório de Programação I*
   MAC0242 – Laboratório de Programação II
   MAC0332 – Engenharia de Software
   Auxiliaram a pensar e escrever melhores códigos-fonte, para melhor compreensão por parte de outros desenvolvedores.

[5] `http://www.googlish.com/wp-content/uploads/2014/09/MoneyLabReader-Sabir-1.pdf`
[6] `https://ethereum.org/`

2. *MAC0338 – Análise de Algoritmos*
   Proveu uma maior compreensão a respeito da performance de algoritmos, incluindo ferramentas para julgar quão bom é o código que estou escrevendo.

3. *MAC0426 – Sistemas de Bancos de Dados*
   Fundamental para implementar contas transacionais de forma segura e correta.

4. *MAC0323 – Estrutura de Dados*
   Proveu a base para a compreensão de algoritmos de hashing (tabelas de espalhamento).

5. *MAE0121 – Introdução à Probabilidade e Estatística I*
   MAE0212 – Introdução à Probabilidade e Estatística II
   MAE0228 – Noções de Probabilidade e Processos Estocásticos
   A mineração de bitcoins é um processo estocástico, que pode ser analisado com os conceitos aprendidos nas matérias acima.

Além das matérias citadas acima, vale a menção especial de *MAC0422 – Sistemas Operacionais*, que desmistificou uma série de funcionalidades presentes nos sistemas operacionais, uma curiosidade minha desde pequeno. A compreensão desses conceitos resultou em melhores projetos de softwares que possuem vários subsistemas.

Referências Bibliográficas

[1] Adam Back. A partial hash collision based postage scheme. `http://www.hashcash.org/papers/announce.txt`, Março 1997.

[2] Y. Benkler. *The Wealth of Networks: How Social Production Transforms Markets and Freedom*. Yale University Press, 2006.

[3] Bitcoin Charts. Mercado Bitcoin (BRL) Trade History. `http://bitcoincharts.com/markets/mrcdBRL.html`, Novembro 2014.

[4] David Chaum. Blind signatures for untraceable payments. In *Advances in cryptology*, pages 199–203. Springer, 1983.

[5] Hans Dobbertin, Antoon Bosselaers, and Bart Preneel. Ripemd-160: A strengthened version of ripemd. In *Fast Software Encryption*, pages 71–82. Springer, 1996.

[6] C. Dwork and M. Naor.

[7] A. P. Faure. Money creation: Genesis 2: Goldsmith-bankers and bank notes. *Disponível em SSRN 2244977*, 2013.

[8] J. Gernet. *Daily Life in China, on the Eve of the Mongol Invasion, 1250-1276*. ACLS Humanities E-Book. Stanford University Press, 1962.

[9] D. Graeber. *Debt: The First 5000 Years*. Penguin Books Limited, 2012.

[10] Second Life. 2009 End of Year Second Life Economy Wrap up (including Q4 Economy in Detail).

[11] G.M. Lilly. Device for and method of one-way cryptographic hashing, December 7 2004. US Patent 6,829,355.

[12] M. Mauss and W.D. Halls. *The Gift: The Form and Reason for Exchange in Archaic Societies*. W.W. Norton, 1990.

[13] RalphC. Merkle. A digital signature based on a conventional encryption function. In Carl Pomerance, editor, *Advances in Cryptology — CRYPTO '87*, volume 293 of *Lecture Notes in Computer Science*, pages 369–378. Springer Berlin Heidelberg, 1988.

[14] British Museum. The origins of coinage. `http://www.britishmuseum.org/explore/themes/money/the_origins_of_coinage.aspx`, Novembro 2014.

[15] Museum of Anthropology of the University of Missouri. Chinese coins. `http://anthromuseum.missouri.edu/minigalleries/chinesecoins/intro.shtml`, Novembro 2014.

[16] Julie Pitta. Requiem for a Bright Idea. Forbes http://www.forbes.com/forbes/1999/1101/6411390a.html, Janeiro 1999.

[17] M. Polo and R. Latham. *The Travels*. Penguin classics. Penguin Books Limited, 1958.

[18] Wikipedia. Size of Wikipedia. https://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia, Novembro 2014.