Universidade Fernando Pessoa

Análise e Aplicação da Tecnologia Blockchain na Gestão de Diplomas do Ensino Superior

Fernando Richter Vidal

Faculdade de Ciência e Tecnologia

Universidade Fernando Pessoa

Uma dissertação submetida ao grau de

Mestre

Orientadores:

Professor Doutor Feliz Gouveia

Professor Doutor Christophe Soares

Julho de 2020

Resumo

As certificações académicas permitem reconhecer habilidades e conhecimentos adquiridos, e impactam diretamente de forma positiva na vida social das pessoas. Os diplomas emitidos nos moldes tradicionais, em papel, podem ser sujeitos a falsificação ou a impossibilidade de verificação devido à indisponibilidade da entidade emissora. Recentemente os modelos digitais de certificações académicas avançaram em questões importantes como a sua emissão, mas ainda podem falhar na verificação e na partilha entre partes, devido a oferecerem arquiteturas centralizadas. Motivados pelos bons resultados obtidos em outras áreas, várias iniciativas de certificados digitais propuseram a utilização da tecnologia *blockchain*. Este trabalho consiste em identificar, analisar e testar algumas das ferramentas baseadas em *blockchain* que estão emergindo, no âmbito da criação de certificados universitários mais eficientes, confiáveis e independentes. Será implementado um protótipo capaz de emitir, verificar e partilhar certificados; é também avaliado o uso da tecnologia e são apresentados os resultados dessa experiência. Além disso, o trabalho apresenta uma visão sobre o estado atual de desenvolvimento e maturidade em que tais ferramentas se encontram, relatando os avanços e as limitações encontradas, e expõe questões que ainda precisam ser resolvidas.

Abstract

Academic certifications make it possible to recognize acquired skills and knowledge and have a direct positive impact on people's social lives. Diplomas issued in the traditional way, on paper, may be subject to forgery or impossibility of verification due to the unavailability of the issuing entity. Recently, digital models of academic certifications have advanced on important issues such as their issuance, but may still fail to verify and share between parties, due to their centralized architectures. Motivated by the good results obtained in other areas, several digital certificate initiatives have proposed the use of blockchain technology. This work consists of identifying, analyzing, and testing some of the blockchain-based tools that are emerging, within the scope of creating more efficient, reliable, and independent university certificates. A prototype capable of issuing, verifying, and sharing certificates will be implemented; the use of technology is also evaluated and the results of this experiment are presented. In addition, the work presents an overview of the current state of the art and maturity of these tools, reporting the advances and limitations encountered and exposes issues that still need to be resolved.

Agradecimentos

Aproveito este espaço para expressar minha profunda gratidão.

Primeiramente gostaria de agradecer a Deus, por ter iluminado os meus caminhos e me conduzido até aqui.

Aos meus orientadores Professor Doutor Feliz Gouveia e Professor Doutor Christophe Soares, que dedicaram parte do seu precioso tempo e compartilharam seus melhores conhecimentos para poder me ajudar. Sou muito grato por terem me corrigido quando necessário sem nunca me desmotivar.

E por último a minha família que manteve o apoio incondicional, mesmo em momentos em que se tornou necessário ausentar-me, para dedicar parte do meu tempo para este trabalho.

Conteúdo

Lista de Figuras

Lista de Tabelas

Acrónimos

UFP *Universidade Fernando Pessoa*

ECTS *European Credit Transfer and Accumulation System*

IBGE *Instituto Brasileiro de Geografia e Estatística*

PNADC *Pesquisa Nacional por Amostra de Domicílios Contínuas*

MIT *Massachussets Institute of Technology*

W3C *World Wide Web Consortium*

A3ES *Agência de Avaliação e Acreditação*

MEC *Ministério da Educação*

GDPR *General Data Protection Regulation*

LGPD *Lei Geral de Proteção de Dados Pessoais*

LF *Linux Foundation*

IoT *Internet of Things*

CCE *Criptografia de Curva Elíptica*

RSA *Rivest-Shamir-Adleman*

UE *União Europeia*

DH *Diffie-Hellman*

UK *United Kingdom*

PDDE *Protocolizadora Digital de Documentos Eletrónicos*

ECDSA *(Elliptic Curve Digital Signature Algorithm*

SHA *Secure Hash Algorithm*

MD5 *Message Digest 5*

MD6 *Message Digest 5*

LM *Learning Machine*

DID *Decentralized Identifier*

VC *Verifiable Credentials*

URL *Uniform Resource Locator*

PoW *Proof of Work*

PoS *Proof of Stake*

DPoS *Delegated Proof of Stake*

PoA *Proof of Authority*

PBFT *Practical Byzantine Fault Tolerance*

dBFT *Delegated Byzantine Fault Tolerance*

P2P *Peer-to-Peer*

DNS *Domain Name System*

DDoS *Distributed Denial of Service*

EVM *Ethereum Virtual Machine*

JSON *JavaScript Object Notation*

HTML *Hypertext Markup Language*

HTTP *Hypertext Transfer Protocol*

API *Application Programming Interface*

CSV *Comma-Separated Values*

MOOCs *Massive Open Online Courses*

UTXO *Unspent Transaction Output*

IPFS *InterPlanetary File System*

IDE *Integrated Development Environment*

FTP *File Transfer Protocol*

TLS *Transport Layer Security*

DAOs *Decentralized Autonomous Organization*

CRL *Certificate Revocation List*

LM *Learning Machine*

Capítulo 1

Introdução

A autenticidade dos diplomas académicos é motivo de grande preocupação para os empregadores e outras autoridades que precisam verificar esses graus. A questão da autenticidade coloca-se porque as instituições emissoras não existem mais ou porque falham ao tentar manter os registos válidos. Em ambas as situações, a verificação da autenticidade dos diplomas pode ser difícil de ser alcançada. Em um mercado educacional mundial, em que cada vez mais instituições estão envolvidas, fica cada vez mais difícil acompanhar os diferentes procedimentos para verificar a autenticidade dos diplomas (Vidal et al., 2019).

As organizações no anseio de encontrar profissionais qualificados, muitas vezes deparam-se com uma imensa variedade de informações, que estão distribuídas das mais diversas formas, como por exemplo, plataformas digitais, correios eletrónicos, e sistemas de arquivos entre outros. É muito provável que haja informações redundantes e desatualizadas sobre um mesmo candidato, bem como sejam encontradas divergências de informações sobre as qualificações informadas. Diante desse grande número de informações dispersas, questões como fiabilidade, imutabilidade e descentralização podem contribuir para uma boa escolha.

As qualificações são tão importantes para os indivíduos, que impactam diretamente em suas vidas sociais. Segundo pesquisa realizada pelo *Pesquisa Nacional por Amostra de Domicílios Contínuas* (PNADC), divulgada através do *Instituto Brasileiro de Geografia e Estatística* (IBGE), demonstrou que o nível de instrução é determinante para a renda dos brasileiros. De acordo com os resultados apresentados, quem conclui o ensino superior consegue praticamente o triplo de remuneração em comparação com aqueles que têm apenas o ensino médio. A diferença pode ser ainda mais acentuada, se comparada entre trabalhadores sem nenhum grau de instrução com aqueles que possuem ensino superior.

Nota-se claramente a medida que os profissionais vão se qualificando, o conhecimento é transformado diretamente em renda, e consequentemente em uma melhor qualidade de vida figura 1.1.

Segundo o Eurostat[3], todos os anos os diplomados do ensino superior na Europa ultrapas-

[2]Gráfico 1.1 elaborado através das informações extraídas da PNADC do primeiro semestre de 2018 (IBGE, 2018)

[3]`http://ec.europa.eu/eurostat`

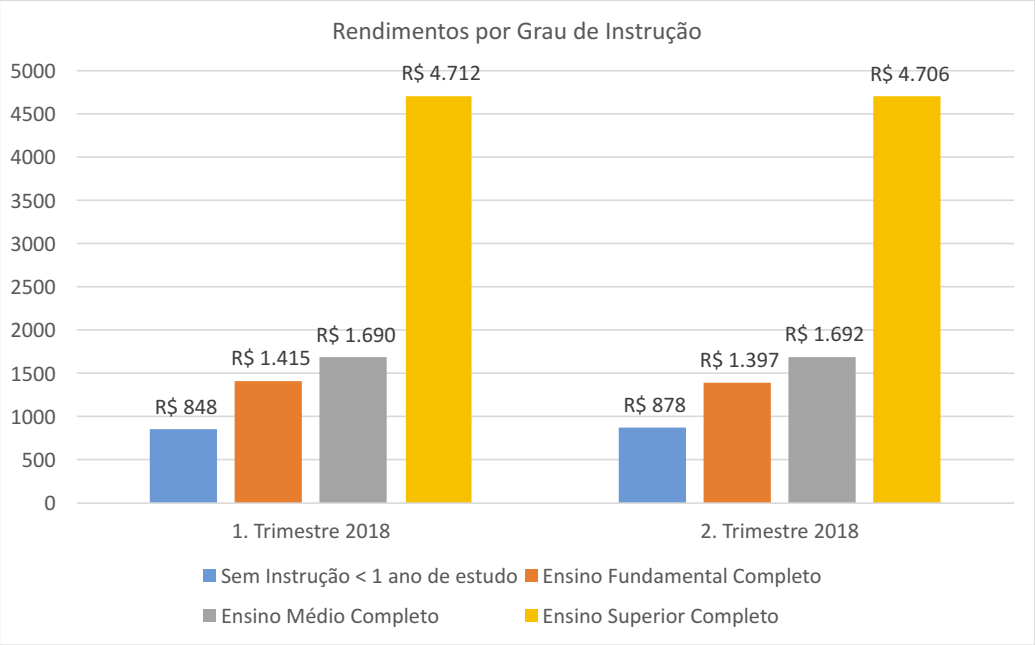

Figura 1.1: *Rendimentos por grau de instrução*[2]

sam 4,5 milhões, sendo a França e o Reino Unido os países líderes com mais de 740.000 diplomados por ano (European Commission/EACEA/Eurydice, 2018). A fração de pessoas com um diploma universitário entre 30 e 34 anos quase duplicou em quatorze anos, passando de 23% em 2002 para 39% em 2016. Estes números confirmam um aumento contínuo e constituem base sólida que justificam a criação de soluções para verificar a autenticidade dos diplomas universitários (Oliver et al., 2018).

Dada a importância das certificações académicas e do impacto gerado pelo êxito de tal obtenção de grau, como garantir que perante as diversas informações dispersas e muitas vezes duvidosas, que as qualificações representadas por diplomas e certidões, foram realmente emitidas por tais instituições ? Qual é o grau de certeza que as avaliações ali representadas, de fato expressam as frequências e pontuações alcançadas ?

É certo de que a inviabilidade de uma pesquisa aprofundada sobre a veracidade dos dados, expõe o avaliador a riscos de tomar decisões equivocadas, como por exemplo uma contratação indevida, ou deixar de oferecer oportunidades a profissionais, mediante a falta de mecanismos eletrónicos e seguros para validar supostas capacitações apresentadas.

O modelo em papel, ainda amplamente utilizado, somente pode ser destruído por quem o detém fisicamente. Isso significa, que o modelo impresso, não oferece autonomia para as entidades emissoras revogarem documentos que foram emitidos erradamente. Diante da sua importância social, e da fragilidade conhecida por tal modelo, alguns governos e também a iniciativa privada, investiram na criação de soluções digitais. Algumas dessas ferramentas podem ser conferidas na secção 4.1.5.

Analisando essas soluções e comparando com os processos envolvidos em uma certificação (constituídos por emissão, verificação e partilha (Grech and Camilleri, 2017)), percebe-se que tais modelos avançaram nas questões de emissão, mas ainda podem falhar na verificação e na partilha, por conta de oferecerem arquiteturas centralizadas, com um único ponto de falha.

A solução em *blockchain* devido à sua natureza distribuída, pode atender os demais processos, por conta de distribuir os registos académicos e operar de forma descentralizada. Isso significa que, caso esta tecnologia seja adotada, além de uma maior fiabilidade, alunos e instituições ganhariam soberania nas revogações, verificações e partilhas.

1.1 Motivação

Nos modelos atuais, os certificados que são emitidos por instituições de ensino superior, passam por um processo de autorização escalonado da seguinte maneira.

i. O órgão de acreditação avalia se tal organização atende aos requisitos pretendidos para lecionar cursos superiores naquele país. Por exemplo, na Europa existem agências de acreditação; em Portugal a *Agência de Avaliação e Acreditação* (A3ES) [4] tem o papel de avaliar, acreditar e garantir a qualidade das instituições nacionais. No Brasil, o *Ministério da Educação* (MEC) [5] é responsável por autorizar a abertura ou encerramento de cursos superiores, bem como avaliar a qualidade de ensino de tais instituições.

ii. Com o parecer positivo, as instituições de ensino estão autorizadas a emitirem certificados para os alunos que atingirem os critérios definidos.

iii. Os interessados nas informações contidas em cada certificado, precisam consultar a entidade emissora para receberem um parecer quanto a veracidade dos dados. Na dúvida sobre a idoneidade ou existência de uma instituição de ensino, é necessário consultar o órgão de acreditação.

Nota-se, que de forma centralizada, a única maneira de garantir a autenticidade dos documentos é verificá-los perante a universidade ou instituição que os emitiu. Além disso, é preciso consultar o órgão de acreditação ou regulador, para certificar que a questão temporal esteja atendida, garantindo assim, que a entidade no dado momento em que o documento foi emitido, estava devidamente autorizada a emitir aquele tipo de diploma.

Também há um problema caso a universidade encerre suas atividades. Nesta situação, fica muito mais difícil confiar puramente no documento emitido em papel, sem que haja uma

[4] https://www.a3es.pt/pt/o-que-e-a3es/missao/
[5] http://portal.mec.gov.br/sesu-secretaria-de-educacao-superior/apresentacao/

conferência junto do órgão emissor. Considerando que possa existir um fluxo permanente de abertura e encerramento de instituições de ensino, uma boa prática seria constantemente verificar junto dos órgãos de acreditação, se um diploma emitido pode ainda ser considerado válido, mediante sua regularização nos órgãos governamentais.

Nesse cenário fica claro que há uma dependência da entidade emissora e que existe uma questão de autonomia e fiabilidade não resolvida. Há riscos, mesmo que remotos, de acontecer um encerramento de um órgão de acreditação, ou um desastre relacionado com a perda generalizada de dados na reguladora. Considerando essas hipóteses, as instituições credenciadas por tais órgãos, mesmo que temporariamente, ficam vulneráveis, sem a possibilidade de oferecerem quaisquer garantia sobre a sua autenticidade.

Diante desses problemas, o uso da tecnologia *blockchain* torna-se perfeitamente aplicável, devido às suas propriedades de descentralização e imutabilidade. Na área da educação, existem plataformas sendo desenvolvidas para armazenar certificados na *blockchain* (M. L. MIT Media Lab, 2016) (Turkanović et al., 2018), que ainda estão sendo testados pelo meio académico, mas que projetam ganhos expressivos de fiabilidade.

Comparando com o modelo em papel, em que há pouca garantia contra fraudes e adulterações, a solução é promissora e resolve bem os problemas citados. Como qualquer outra nova solução, há sempre limitações que só poderão ser descobertas após iniciar-se experiências. Nesse sentido, este trabalho contribui fornecendo uma aplicação na área da educação, para a emissão, a partilha e a verificação de certificados académicos usando *blockchain*.

1.2 Objetivos

Embora desde os anos 90 já tenham sido feitas experiências com tecnologias distribuídas de moeda digital, esta área apenas ganhou notoriedade em 2008, diante da publicação anónima do pseudónimo Satoshi Nakamoto (Nakamoto, 2008), que ficou conhecida como *Bitcoin blockchain*, ie., ele propôs uma criptomoeda descentralizada. Desde então, várias outras criptomoedas têm surgido, explorando o uso desta tecnologia, como por exemplo, *Ethereum* [6] ,*Litecoin* [7], *Monero* [8] e *Neo* [9].

Por conta de não haver relatos de nenhum grande problema até então, a aplicação logo chamou a atenção e ganhou notoriedade. Negócios que requerem alta confiabilidade e honestidade podem atrair clientes (Zheng et al., 2017). A contar do surgimento da *Bitcoin*, em mais de uma década, não se sabe de nenhuma grande falha de segurança da cripto-

[6] https://www.ethereum.org/
[7] https://litecoin.org
[8] https://monero.org
[9] https://neo.org/?culture=en-us

moeda, o que despertou interesse noutros setores de atividade, como por exemplo na área académica.

Investigar seu potencial em aplicações e demonstrar que a *blockchain* pode ser utilizada para resolver problemas na área da educação, é um dos objetivos deste trabalho. Também pretende-se explorar a fiabilidade, permanência e distribuição oferecida pela tecnologia, de maneira a fornecer uma maior autonomia para profissionais, estudantes, empresas e universidades sobre o âmbito dos diplomas. Propõe-se avaliar todas estas questões através da implementação de um sistema de emissão e verificação de certificados, desenvolvido usando uma plataforma de *blockchain*.

Faz parte também deste estudo avaliar as ferramentas emergentes que oferecem uma plataforma de trabalho, como por exemplo *HyperLedger* [10] da *Linux Foundation* (LF) e *Blockcerts* do *Massachussets Institute of Technology* (MIT), e apresentar os critérios utilizados na escolha realizada. Não será proposto nenhum modelo novo de implementação da *blockchain*, ou de plataforma de geração de certificados, a tecnologia já existe, inclusive na modalidade de código aberto. Este trabalho também não abrange um estudo sobre o estado da arte de criptografias ou modelos matemáticos incorporados nas ferramentas utilizadas.

A figura 1.2 propõe que o processo de verificação dos certificados seja realizado de forma descentralizada e automática. Através das técnicas demonstradas ao longo dos próximos capítulos, com o uso da *blockchain*, é possível validar um certificado, mesmo que a entidade emissora tenha encerrado suas atividades. Neste cenário, a autonomia é mantida para os interessados que poderão verificar de forma instantânea, sem a necessidade de contactar uma terceira parte, a veracidade das informações disponibilizadas.

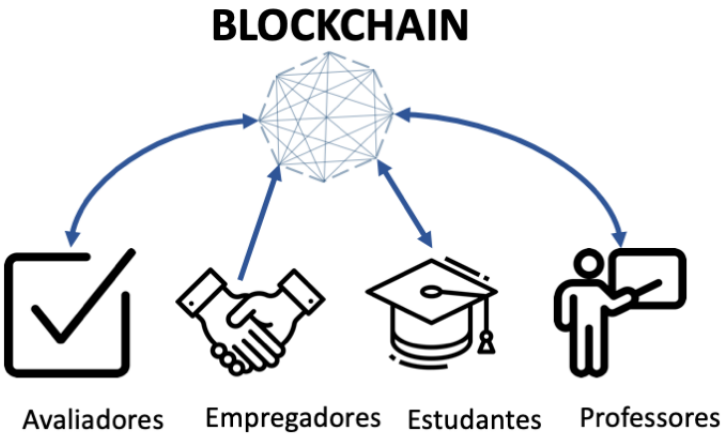

Figura 1.2: *Interessados na área da educação e suas interações, adaptado de (Grech and Camilleri, 2017)*

[10] `https://Hyperledger.org`

1.3 Trabalhos existentes

Existem projetos em investigação que pretendem disponibilizar diplomas em formato digital usando *blockchain*, como por exemplo a *Learning Machine* (LM) [11], uma *startup* que nasceu por iniciativa do MIT junto com a *World Wide Web Consortium* (W3C).

Além do MIT, a Universidade de Nicósia e pesquisadores da Universidade de Birminghan estão desenvolvendo seus próprios sistemas usando a especificação código aberto *Blockcerts* (Grech and Camilleri, 2017). O Governo de Malta tem feito esforços para aplicar a tecnologia, de forma pioneira, também na área da educação (Diacono, 2017). A Universidade de Rosário na Colômbia está testando a *Blockcerts*, publicando certificados na rede *Hyperledger* (Iragorri, 2018). A Universidade Nacional de la Plata tem desenvolvido uma plataforma baseada em *blockchain* para verificação de registos académicos (Turkanović et al., 2018). A mesma abordagem também foi adotada por outra instituição de ensino na Argentina, a CESYT (Amati, 2015). Em 2016, a instituição de ensino Parisian Leonardo da Vinci Engineering na França, anunciou que iria começar a verificar diplomas recorrendo a uma rede *Bitcoin* (Das, 2016).

Também há soluções que estão nascendo direto da iniciativa privada, como por exemplo, *Sony Global Education* (Sony, 2016; Russel, 2017) e *Civic* [12].

Segundo o relatório de (Grech and Camilleri, 2017), é provável que o número de empresas oferecendo certificados na *blockchain* aumente num futuro próximo.

1.4 Metodologia aplicada

Este trabalho baseia-se na análise bibliográfica sobre *blockchain* e o seu potencial em aplicações na área da educação, mais especificamente na emissão e verificação de diplomas de ensino superior. Diante de uma pesquisa realizada sobre algumas das ferramentas existentes, é feita uma escolha para implementação de um protótipo capaz de testar os processos de emissão, verificação e partilha. O protótipo construído deve servir como prova de conceito. Finalmente, é feita uma análise dos resultados e são retiradas as respetivas conclusões.

[11] `https://www.learningmachine.com/about`
[12] `https://www.civic.com`

1.5 Estrutura do trabalho

É apresentado no Capítulo 1, as motivações que levaram a elaboração deste trabalho, as contribuições que se pretendem dar na área da gestão de certificados, e os benefícios que a tecnologia empregada por este trabalho pode oferecer. Também são citados outros trabalhos que abordam o mesmo tema.

O Capítulo 2 aborda alguns dos conceitos da tecnologia usada na *blockchain*, servindo como base para entendimento de seu funcionamento. Além disso, o conceito de certificado é também apresentado.

A solução proposta para o problema é construída utilizando tecnologia *blockchain*, por essa razão, o Capítulo 3 detalha a estrutura da tecnologia, e revisa duas das mais importantes aplicações, o *Bitcoin* e o *Ethereum*.

O Capítulo 4 apresenta algumas soluções específicas em *blockchain*, que despontam para armazenar o certificado académico. Um breve estudo entre as soluções é apresentado, destacando-se os pontos positivos e negativos de cada arquitetura. Finaliza-se justificando o porquê do *Blockcerts* ter sido escolhido para a construção do protótipo.

O Capítulo 5 detalha as especificações funcionais do protótipo construído e também como a arquitetura foi pensada para ser integrada com o *Blockcerts*. Neste capítulo também é apresentada as parametrizações que foram implementadas para tornar o sistema aderente a diferentes tipos de emissoras. Finaliza-se o capítulo demonstrando que a solução idealizada pelo protótipo deve ser transparente ao tipo de *blockchain*, por essa razão, foram implementadas duas redes: *Bitcoin* e *Ethereum*.

O Capítulo 6 apresenta cenários de testes e simulações utilizando o protótipo desenvolvido, tais como: adulteração de certificados, verificação de revogação, custos de emissão e validação desconectada. O capítulo finaliza avaliando cada um dos resultados.

Encerra-se com o Capítulo 7, em que se discutiu os resultados obtidos. Nesta parte final do trabalho, são revistos os objetivos e o seu grau de cumprimento, assim como as limitações encontradas. E por último, são listadas aqui as oportunidades para trabalhos futuros, sobre questões que ainda podem ser exploradas.

Capítulo 2

Revisão Bibliográfica

2.1 Revisão de conceitos

Nas próximas secções, pelo fato de formarem os pilares da tecnologia *blockchain*, são primeiramente apresentadas as definições de criptografia assimétrica, assinatura digital, condensados criptográficos e árvore de *Merkle*. O que percebe-se observando esses componentes, é que a *blockchain* utilizou técnicas que já existiam e conseguiu agregá-las numa solução tecnicamente funcional.

2.1.1 Criptografia assimétrica

O maior problema a ser tratado pela criptografia é a confidencialidade, ou seja, prevenir que uma informação de maneira não autorizada, seja extraída em um canal de comunicação inseguro. Foi na década de 70 (Ellis, 1970)(Hellman and Whitfield, 1976) que a utilização da criptografia em canais inseguros, com problemas de troca de chave, ganhou força com o surgimento do conceito de chave assimétrica. A ideia proposta por *Diffie-Hellman* (DH) (Hellman and Whitfield, 1976) era de criar um sistema que fosse capaz de cifrar uma mensagem com o uso de duas chaves distintas: uma pública e outra privada. A primeira como o nome próprio sugere, poderia ser divulgada abertamente ao público, quanto a segunda, deveria ser mantida em segredo.

Estas propriedades mudaram os rumos da criptografia, uma vez que tornou-se possível a um emissor distribuir livremente uma chave pública, que permitia cifrar uma mensagem, mas que era incapaz de decifrá-la. Isso significa que mesmo que um atacante descobrisse a chave pública, nada poderia fazer para infringir a confidencialidade, por conta de unicamente o proprietário da chave privada ser capaz de decifrar tal mensagem.

Este novo mecanismo proposto por DH, não veio para substituir o sistema de chaves simétricas, até mesmo porque não atingem a mesma performance. A sua contribuição deu-se no que refere-se a distribuição, eliminando a necessidade de trocar fisicamente chaves privadas, para que uma comunicação segura ocorresse. Desta maneira, tornou-se possível distribuir com segurança as chaves públicas simétricas de forma eletrónica,

revolucionando a maneira de como funcionava a comunicação segura. A criptografia assimétrica é baseada numa função com as seguintes características:

– É fácil utilizar a função para cifrar.

– É impossível inverter a função, isto é, decifrar, a menos que se detenha um segredo.

A figura 2.1 demonstra um cenário em que um usuário gera uma chave simétrica, e de forma segura, realiza a sua transmissão a um destinatário. Para isso, uma mensagem contendo o conteúdo da chave simétrica é criada e cifrada com a chave pública disponibilizada pelo destinatário. A mensagem criptografada é transmitida pela rede, e mesmo que um atacante descubra a chave pública em que a mensagem foi cifrada, nada poderá fazer, porque ela apenas serve para cifrar e não para decifrar uma mensagem criptografada. Assim, somente poderá decifrar a mensagem, em cujo conteúdo está a chave simétrica, o proprietário da chave privada correspondente à chave pública que a cifrou. Desta maneira, de forma segura, apenas o destinatário pode recuperar a chave simétrica enviada pelo usuário, e a partir de então, usá-la para a comunicação. Esta é a base geral de um protocolo de comunicação segura como o *Transport Layer Security* (TLS).

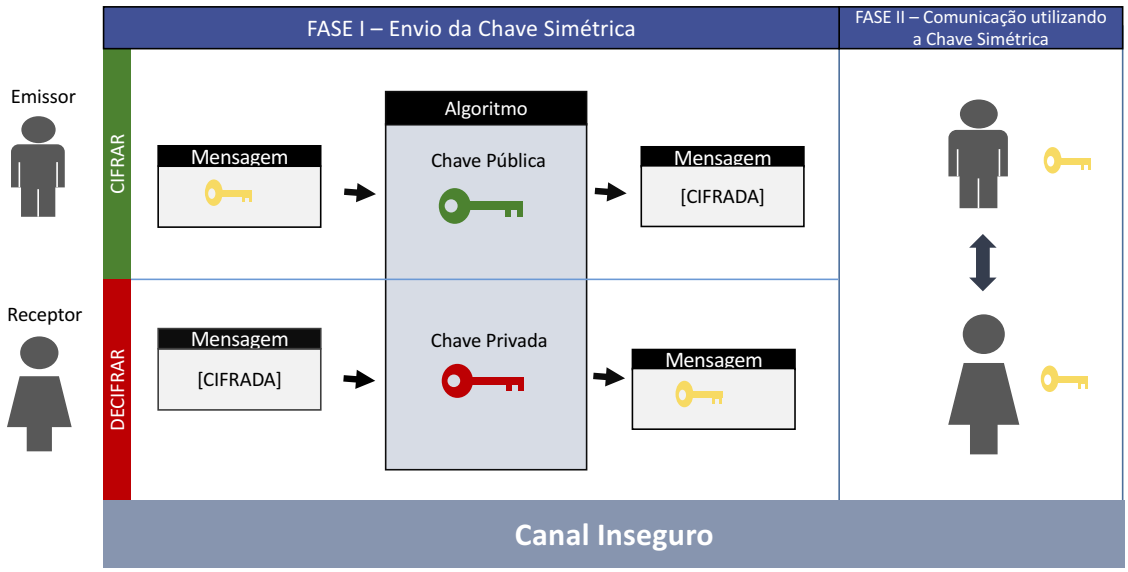

Figura 2.1: *Distribuição de chaves simétricas usando criptografia assimétrica*

O RSA foi um dos primeiros sistemas criptográficos assimétricos, o primeiro a ter uma implementação, e ainda é amplamente utilizado. Neste algoritmo, o texto simples e o texto cifrado são considerados como inteiros entre 0 e $n-1$, em que n é o módulo (Al Hasib and Haque, 2008). A escolha inicial da função a utilizar recaiu sobre a exponenciação modular, sendo o módulo, atualmente da ordem de 2^{2048}, o produto de dois fatores primos.

A chave pública é constituída pelo módulo e pelo expoente público. A chave privada é o expoente privado.

De forma simplificada, o funcionamento do algoritmo é descrito em três fases (geração da chave, cifrar e decifrar) (Al Hasib and Haque, 2008):

Geração da chave

1. Seleciona-se dois números primos grandes p e q, tal que $p \neq q$. Considerando o tamanho de 2048 *bits* cada, serão gerados números com 617 dígitos.
2. Calcula-se o valor de n, através do resultado do produto $n = p.q$.
3. Calcula-se o *totiente*, $\varphi(n) = (p-1).(q-1)$.
4. Escolhe-se um número inteiro e (expoente público) em que $1 < e < \varphi(n)$. Em outras palavras, procura-se um número $e > 1$, onde o $MDC(e, \varphi(n)) = 1$.
5. Calcula-se um segredo d na qual $d.e \equiv 1(mod n)$.
6. Chave pública será (n, e) e a chave privada será (n,d).

Cifrando

1. Obtêm-se a chave pública do destinatário (n, e).
2. Converte-se a mensagem em texto para números inteiros positivos m.
3. Calcula-se o texto cifrado como $c = m^e \ mod \ n$.
4. Envia-se o texto cifrado c para o destinatário.

Decifrando

1. O destinatário usa a sua própria chave privada (n, d) para calcular $m = c^d \ mod \ n$.
2. Converte-se para texto, o conjunto de inteiros representados.

A maneira como a segurança é garantida em um sistema criptográfico assimétrico, é dada pelo esforço necessário para efetuar um cálculo matemático (Impagliazzo and Luby, 1989). Este esforço contabilizado em tempo computacional significa que quanto maior for, mais eficiente é o algoritmo. Por exemplo, DH exploraram a dificuldade de reverter-se operações que eram baseadas em cálculos logarítmicos (Hellman and Whitfield, 1976), o RSA realiza um cálculo matemático que explora a dificuldade de calcular fatores primos de números grandes (Rivest et al., 1983), a CCE, baseia-se nas propriedades matemáticas das curvas elípticas (Hankerson et al., 2004).

A segurança de um bom algoritmo de criptografia concentra-se na chave e não em sua implementação. Desta forma, mesmo que se descubra o funcionamento do algoritmo, preservando-se a privacidade da chave, a segurança ainda está garantida. Isso significa que quanto maior a dificuldade em encontrar a chave, menor são as chances de que uma

mensagem seja quebrada.

A tecnologia *blockchain* utiliza o algoritmo de criptografia assimétrica CCE, que até então é dado como muito seguro. Diversos trabalhos foram realizados comparando o CCE com outros algoritmos já consolidados, como por exemplo o RSA e demonstraram que o CCE apresentou resultados superiores (Mahto and Yadav, 2017)(Gura et al., 2004)(Singh et al., 2016)(Mahto and Yadav, 2018).

Tal sistema foi proposto em 1985 por (Miller, 1986) e (Koblitz, 1987), tendo como base a matemática envolvendo curvas elípticas. As chaves públicas e privadas deste algoritmo são calculadas a partir de repetidas operações de adição de coordenadas polares, obtidas pela interseção da curva elíptica com diversas retas tangentes.

Como pode-se notar, a tabela 2.1 apresenta um comparativo entre o tamanho de chaves geradas para um mesmo nível de segurança, entre o algoritmo CCE e RSA. Com base na tabela apresentada, conclui-se, que o CCE torna-se interessante, pelo fato de que sua criptografia é capaz de proporcionar um mesmo nível de segurança que outros algoritmos, mas gerando chaves menores. Desta maneira, o CCE conseguiu incorporar complexidade na chave, sem aumentar o seu tamanho, simplificando o desenvolvimento de aplicativos de segurança, principalmente para dispositivos móveis que têm um poder de processamento e de armazenamento mais limitado.

Tabela 2.1: *Comparativo de tamanho de chave entre RSA e CCE (Barker et al., 2012)*

Nível de segurança de *bits*	**RSA**	**CCE**
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

Outro fato que fortalece o uso do CCE na *blockchain* deve-se a proporção em que o tamanho da chave aumenta, em relação ao aumento do nível de segurança. Por outras palavras, caso haja a necessidade de aumentar a segurança da criptografia, aumentando o número de *bits*, o tamanho da chave cresce, mas de forma moderada. Comparando com o RSA, pode-se notar através da figura 2.2, que o comportamento da função elíptica é muito mais linear.

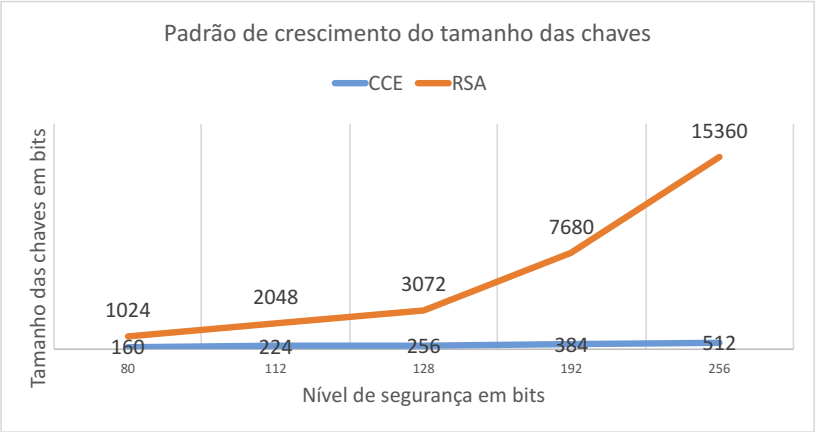

Figura 2.2: *Padrão de crescimento do tamanho das chaves adaptado de (Mahto and Yadav, 2018)*

2.1.2 Assinatura digital

De forma semelhante ao funcionamento das assinaturas em papéis, esta tecnologia foi criada para que fosse possível, mas de forma eletrónica, que um registo digital tenha a sua autenticidade comprovada. Uma assinatura digital deve ter as seguintes propriedades (Vigil et al., 2015) :

- Autenticidade: quem recebe a mensagem, deve ser capaz de confirmar que a assinatura contida no documento foi realmente feita pelo emissor.
- Integridade: a assinatura deve corresponder fielmente ao conteúdo do documento, de modo que qualquer alteração na mensagem, torna inválida a correspondência entre elas.
- Não repúdio: o recetor tem a garantia de que o emissor uma vez identificado, não pode negar a autenticidade da mensagem.
- Prova de existência: em algumas situações, é preciso garantir que a informação disponibilizada existiu, mesmo que apenas em um dado momento temporal.

Uma assinatura em papel geralmente usa algum tipo de marca física para identificar a autenticidade do documento. A assinatura manuscrita indica que o proprietário da assinatura reconhece o documento e prova sua identidade através dela. Já no documento digital é preciso representar essas propriedades através de algoritmos e mecanismos digitais. Para garantir que uma assinatura é de fato verdadeira, existem entidades, denominadas autoridades certificadoras, que garantem que uma assinatura feita com tal certificado, é de propriedade do autor da mensagem. Essas autoridades fazem a ligação entre o certificado e o assinante, garantindo dessa maneira o princípio da autenticidade.

A integridade do documento feito em papel é feita através de uma análise visual, como por exemplo, a inexistência de rasuras. No entanto, para atingir essa propriedade no documento digital, usa-se de recursos de criptografia. Como pode-se notar pela figura 2.3, o processo de gerar uma assinatura digital de um documento, garante que sua adulteração seja facilmente percebida.

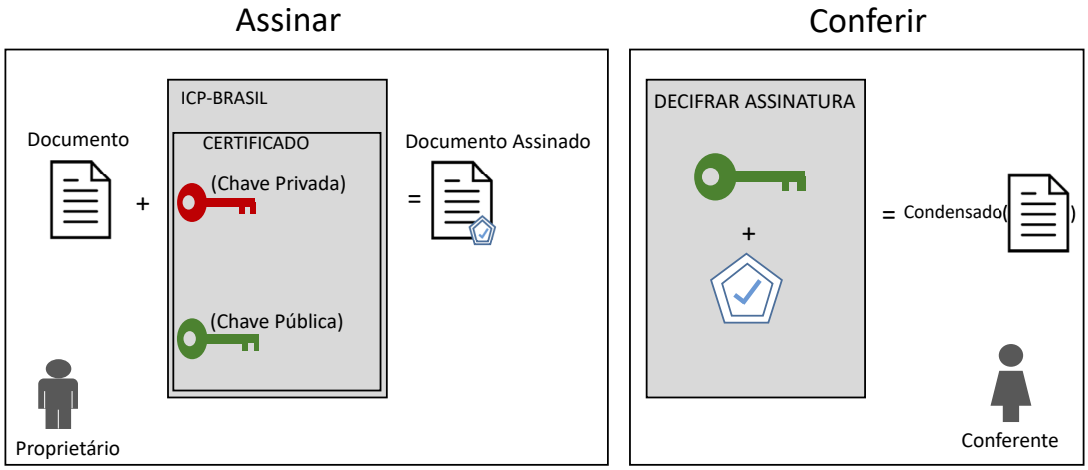

Figura 2.3: *Assinatura digital e conferência de um documento*

Tomando como exemplo o Brasil, no quesito não repúdio, o modelo em papel necessita do reconhecimento em cartórios autorizados, para testificar que aquela marca de assinatura pertence exclusivamente a um determinado autor.

E por último, a prova de existência. Exemplificando com base no cenário brasileiro, uma assinatura em papel caracteriza-se por ser irretroativa, ou seja, não se pode assinar um documento de forma retroativa. Embora seja passível de falhas, essa propriedade de certa maneira garante que tal documento existiu em determinada época. Reproduzir esta característica no meio digital pode ser algo muito mais complexo, levando em conta que uma data pode ser facilmente adulterada sem deixar pistas. Uma alternativa criada para este problema se dá através dos protocolos criptográficos de datação, que funcionam como uma espécie de âncora temporal. Por exemplo, pode-se implementar um servidor *Protocolizadora Digital de Documentos Eletrónicos* (PDDE) que garante a irretroatividade a documentos eletrónicos.

Existem muitas implementações consolidadas do uso de registos eletrónicos, principalmente em questões de interesse público. Isto apenas tornou-se possível por conta da assinatura digital, que ofereceu segurança através das propriedades citadas, para que um registo em papel fosse transformado em digital.

Por exemplo, a França na região de Alsace-Moselle, com o objetivo de reduzir espaço físico e facilitar o acesso público, transformou o sistema de registo de imóveis de físico

para digital, assim como a Estônia adotou a mesma iniciativa (Vigil et al., 2015). As faturas eletrónicas podem ser usadas para fins fiscais na *União Europeia* (UE), conforme a diretiva 2001/115/EC (Commission, 2002). No Brasil deu-se início em 2006 a emissão eletrónica de notas fiscais, para indústria e comércio [1].

Outro setor que tem se beneficiado com a digitalização é a saúde. Em muitos países é necessário manter os registos médicos por anos, e a melhor maneira de fazer isto é utilizando o formato digital. Por exemplo, na Alemanha, pode chegar a 30 anos a necessidade de manter os registos usados. No *United Kingdom* (UK) a obrigatoriedade pode chegar em até 10 anos após o paciente falecer (Vigil et al., 2015).

Desde o início dos *Bitcoin*s, notou-se que a *blockchain* não tinha interesse em fornecer confidencialidade sobre as informações, pelo contrário, em sua essência, a tecnologia propunha sempre uma cadeia de registos públicos. No entanto, a autenticidade sempre foi um fator crítico de sucesso, tornando fundamental que cada registo no bloco seja feito apenas pelas pessoas autorizadas a fazê-lo. Por esse motivo, a criptografia na *blockchain* é normalmente aplicada como assinatura e não como encriptação de mensagem (Krzysztof Okupski, 2014).

O algoritmo utilizado para assinar as transações na *blockchain*, é o *(Elliptic Curve Digital Signature Algorithm* (ECDSA), que baseia-se em curvas elípticas, conforme mencionado na secção 2.1.1. Informação detalhada do funcionamento deste algoritmo pode ser encontrada em (Johnson et al., 2001).

2.1.3 Função de condensado

De forma intuitiva pode-se afirmar que uma função unidirecional $f(x)$ facilmente é calculada e dificilmente é revertida. Considerando a expressão $y = f(x)$, podemos afirmar que quando é fornecido o valor de x, bem como a função de f, é muito fácil calcular o y. No entanto, levando em conta que se conhece o y e a função de f, é praticamente impossível calcular o valor de x (Merkle, 1990).

Mais precisamente, uma função unidirecional f é uma função que dado um conjunto de objetos x, resulta em um conjunto de valores y que respeita duas propriedades (Lamport, 1979):

- Dado qualquer valor de y é computacionalmente inviável encontrar o objeto x correspondente calculado pela função f.
- Dado qualquer objeto de x é computacionalmente inviável encontrar um outro objeto x', calculando o valor de x.

[1] `http://www.nfe.fazenda.gov.br/portal/principal.aspx`

A função de condensado (em inglês na literatura chama-se *hash*) é uma função unidirecional muito utilizada em criptografia, que baseia-se em um algoritmo que não utiliza nenhum tipo de chave. Elas são assim chamadas porque uma vez executadas, não há meios de reverter a criptografia para a fonte original. A entrada pode ser uma sequência variável de caracteres e a saída consiste em uma sequência de caracteres de comprimento fixo, denominado valor condensado (Conrad et al., 2017).

Embora o resultado da função seja um dado comprimido, esta função não pode ser utilizada para tal propósito (St Denis and Johnson, 2007), por conta da impossibilidade de reverter a operação. Desta forma, as funções de condensado são geralmente usadas para prover integridade. Por exemplo, pode-se afirmar que caso o condensado de um texto mude, significa que o texto a ele relacionado foi alterado (Conrad et al., 2017).

Um bom algoritmo de condensado, poderia ser assim classificado, como aquele que não gera colisões, e permanece irreversível. Uma colisão ocorre, quando duas origens distintas, geram o mesmo resultado condensado. Na figura 2.4, as fontes y e z representam, por exemplo, dois textos distintos. Ao calcular os seus respetivos condensados, obtêm-se como resultado uma mesma saída. Neste caso, não seria possível identificar se o condensado obtido refere-se a y ou z, por essa razão nesses casos conclui-se que ocorreu uma colisão.

Teoricamente é impossível evitar que não ocorra colisões. Isso se deve pelo fato de que existem maiores possibilidades de textos aleatórios, do que a quantidade de condensados possíveis a serem gerados (Conrad et al., 2017).

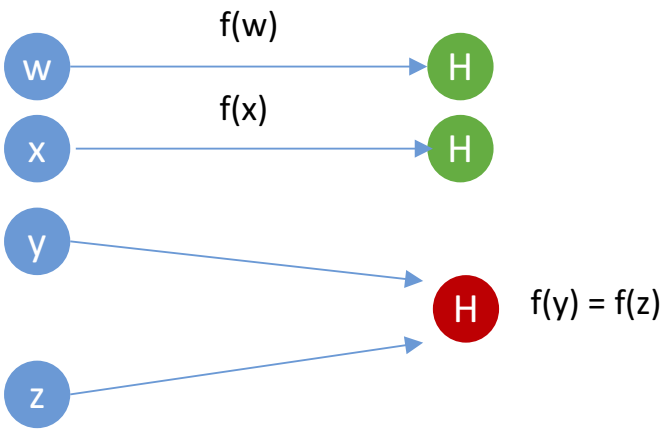

Figura 2.4: *Colisão de função condensado, adaptado de (Narayanan et al., 2016)*

Embora populares por muito tempo, os algoritmos de condensado *Message Digest 5* (MD5) e *Secure Hash Algorithm* (SHA)-1 tornaram-se obsoletos. SHA-1, gerando con-

densados de 160-*bits* e o MD5 de 128-*bits*, foram evidenciados de fraquezas e facilidade em gerar colisões. Por essa razão, sempre que se constatar colisões em um algoritmo, surge a necessidade de buscar novos mecanismos de geração do condensado. Foi o que ocorreu para esses dois algoritmos, onde emergiram novas versões como SHA-2 e *Message Digest 5* (MD6). Curiosamente, o nome SHA se dá pela série de condensado fornecida pelo algoritmo. Por exemplo, SHA-1, cria valores de condensado de 160-*bits*, SHA-2 inclui versões: SHA-224, SHA-256, SHA-384 e SHA-512, todos nomeados em referência ao número de dígitos que cada um cria.

A função de condensado é utilizada na tecnologia *blockchain*, por conta de suas características de padronizar informações em um tamanho fixo de caracteres e gerar códigos exclusivos. Por exemplo, uma transação pode conter inúmeras informações, como: endereço de origem, endereço de destino, data de criação e o ativo representado na operação. Todas essas informações contidas em uma transação, são processadas através de uma função de condensado e representadas através de um único código. Levando em conta que nunca haveria duas transações com exatamente as mesmas informações, esse código gerado é exclusivo e serve como um identificador único da transação. Maiores informações podem ser observadas na secção 3.1.3, de como a função é aplicada na formação e no relacionamento dos blocos.

2.1.4 Árvore de *Merkle*

Ralph Merkle apresentou em 1987 (Merkle, 1988) uma técnica de criptografia, utilizando funções unidirecionais, que permitia criar estruturas de armazenamento de dados seguras. Embora não nomeado pelo artigo referido, este mecanismo ganhou o nome do autor e ficou conhecido como árvore de *Merkle*.

Conforme apresenta figura 2.5, a árvore é construída das folhas para a raiz, realizando uma operação de condensado por pares. Na base da árvore está a entrada de dados, nesse caso representada pelas variáveis a, b, c, d, e. Na imagem percebe-se que é realizada uma operação com a função condensado sobre os dados, e logo em seguida um processo de concatenação e recálculo dos condensados.

O resultado da operação $T1$, é concatenado com resultado da operação $T2$, desta forma obtém-se um novo valor de condensado denominado $T6$. Esse cálculo utilizando o resultado do condensado do nó anterior, realizado em pares, é repetido até chegar no nó raiz. Desta forma, o nó raiz é formado com base na informação de todos os níveis anteriores, pela seguinte expressão:

$$\text{Raiz:} = H\left\{H\left[H\left(H(a)+\!+H(b)\right)+\!+H\left(H(c)+\!+H(d)\right)\right]+\!+H\left[H\left(H(e)+\!+H(e)\right)+\!+H\left(H(e)+\!+H(e)\right)\right]\right\}$$

Nota-se, que quando o número de folhas é ímpar, a ultima folha é duplicada e o cálculo

do condensado é feito com o próprio nó. Observa-se na figura 2.5 que as operações $T5$ e $T8$ não dispõe de outro nó para formar um par.

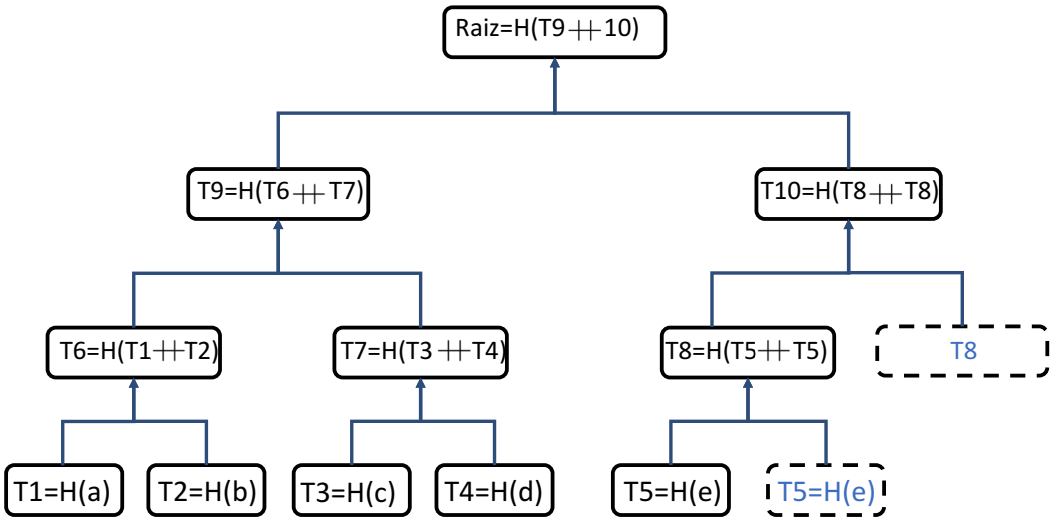

Figura 2.5: *Árvore de Merkle*

A árvore de *Merkle* é parte fundamental da tecnologia *blockchain*, porque oferece uma maneira eficiente de garantir em grandes volumes, a integridade dos conteúdos armazenados. Isto se deve ao fato de seus mecanismos de verificação serem rápidos e conseguirem validar uma informação, mesmo diante de uma grande massa de dados. Por outras palavras, as árvores de *Merkle* são uma forma simples de verificar a integridade de um nó sem ter que verificar toda a árvore.

Todas as transações de um bloco são sintetizadas e representadas através de um identificador único. Assim é possível verificar rapidamente se uma transação está ou não inclusa em um bloco, consultando apenas o registo sumariado. Este registo é chamado de raiz de *Merkle*, e faz parte do cabeçalho de cada bloco, que será melhor detalhado na secção 3.1.3.

A figura 2.6, permite refletir a quantidade de comparações de condensados necessárias, considerando uma arquitetura sequencial, para verificar se a mensagem g, foi adulterada. Neste cenário, seria necessário calcular o condensado da mensagem g e compará-la com os outros 15 condensados, para assim conseguir comprovar se o resultado obtido é igual ao condensado confiável que está armazenado na raiz. Caso o cálculo total dos condensados seja igual ao valor armazenado na raiz, pode-se afirmar que a mensagem g não foi adulterada.

A eficiência da árvore de *Merkle* pode ser comprovada pela figura 2.7, que ao invés de

15 comparações, apenas 4 são necessárias. Em termos computacionais, essa economia de passos, resulta em uma necessidade mais enxuta de recursos como memória e espaço em disco.

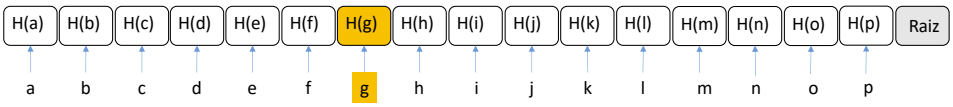

Figura 2.6: *Registo de um bloco sendo verificado de forma sequencial*

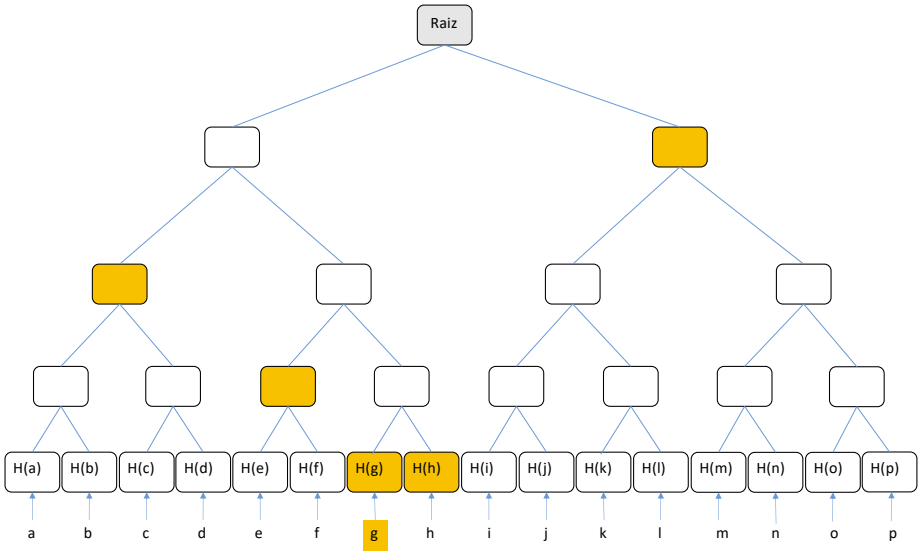

Figura 2.7: *Registo de um bloco sendo verificado pela árvore de Merkle*

No protótipo apresentado por este trabalho, utiliza-se *Bitcoin* e *Ethereum*, e ambas utilizam a árvore de *Merkle*. Maiores informações de como calcular a raiz de *Merkle*, podem ser verificadas no Apêndice A.

2.1.5 Certificados académicos

Credenciar um indivíduo em uma determinada habilidade, tem como objetivo fazer com que outras pessoas passem a reconhecer a sua capacidade e mérito. O certificado além de ultrapassar barreiras de nacionalidade, serve como forma de demonstrar a sua capacidade e aptidões onde outras características como a linguagem, nacionalidade, identidade religiosa, não podem ser pressupostas (Smolenski and Hughes, 2016).

Como observa (Schmidt, 2017), sistemas de credenciais ineficientes limitam a nossa capacidade de criar novos caminhos para a educação, especialmente para aqueles que não têm acesso e mais precisam. Um desafio para as pessoas sem educação formal é traduzir seu aprendizado em empregos, porque muitas vezes não possuem credenciais que afirmam suas habilidades e experiência.

Para que um certificado seja emitido, é necessário que ocorra um processo de certificação gerido por um emissor reconhecido e qualificado, que dado o fim do processo, seja possível atestar a qualificação obtida para quem o reivindica. Mais precisamente na educação, a certificação é usada em muitos cenários como por exemplo:

- Obtenção de resultados, independentemente da forma de aprendizagem.
- Validação da competência de um professor.
- Um processo de aprendizagem realizado por um aluno.
- Uma organização ou curso educacional que atenda a certos critérios de qualidade.
- Um órgão de acreditação autorizado a emitir certificações.

A certificação envolve três processos: emissão, verificação e partilha (Grech and Camilleri, 2017) que serão desenvolvidos a seguir.

2.1.5.1 Processo de emissão

Este é o processo em que concretiza-se o reconhecimento de uma reivindicação, na qual um emissor evidência um destinatário através de uma assinatura em um certificado. Geralmente esses dados são registados em um banco de dados centralizado, sem uma padronização universal.

É necessário que o emitente identifique-se no documento de forma que posteriormente, seja possível verificar a sua identidade. Nas emissões em papéis, geralmente adota-se logótipos e assinaturas reconhecidas por algum órgão responsável. Por exemplo, há uma convenção entre vários países, entre os quais o Brasil [1], que reconhecem de forma bidirecional, um certificado padronizado pela apostila de Haia.

Na prática, a apostila de Haia consiste em um selo ou carimbo emitido pelas autoridades competentes, que é colocado no documento como forma de certificar sua autenticidade junto ao órgão do qual foi expedido, para que assim seja válido no país requerido. No entanto, para que os cartórios brasileiros reconheçam os documentos no formato da apostila de Haia, é necessário que a assinatura contida no certificado pertença a um responsável legal da instituição que os emitiu, e esteja devidamente registada em cartório, de maneira

[1] http://www.cnj.jus.br/poder-judiciario/relacoes-internacionais/convencao-da-apostila-da-haia/paises-signatarios

que exista a possibilidade de emitir uma cópia autenticada.

Nota-se pela tabela 2.2 [1], no exemplo da apostila de Haia, levando em conta que os custos apresentados são por documento e acrescidos de taxas administrativas aplicadas pelos cartórios, a identificação do emissor pode ser um processo complexo e custoso. Felizmente, existem soluções baseadas em tecnologia *blockchain* que podem modificar este cenário. Por exemplo, a plataforma *Blockcerts* permite que um emissor seja identificado através de um endereço público que fica anexado em cada certificado emitido. Desta forma, não seria necessário despender recursos financeiros, para identificar de forma segura, que aquele certificado foi realmente emitido por aquele emissor.

Tabela 2.2: *Custo de emissão da apostila de Haia no Brasil por unidade federativa*

Grupo a.	**R$**	**Grupo b.**	**R$**	**Grupo c.**	**R$**
SP	107,10	**MG**	103,00	**PR**	93,30
MA	85,40	**MS**	83,70	**MT**	78,40
PE	77,46	**SE**	65,08	**PA**	62,70
RN	62,27	**GO**	57,80	**TO**	56,49
BA	54,24	**AP**	51,30	**ES**	50,86
RS	50,32	**AL**	50,00	**RJ**	49,98
PI	47,19	**AM**	43,20	**CE**	42,10
DF	37,30	**RO**	36,49	**AC**	35,80
SC	34,00				

Existem outras abordagens que estão emergindo, que integradas com a tecnologia *blockchain*, e que também podem resolver o problema da identificação. Exemplos disso são o uso de *Decentralized Identifier* (DID) e *Verifiable Credentials* (VC) (Lagutin et al., 2019), descritos a seguir.

DID é um novo tipo de identificador verificável, que utiliza um sistema de identidade digital descentralizada. Esses novos identificadores foram projetados para permitir que o controlador de um DID prove controle sobre ele e seja implementado independentemente de qualquer registo centralizado, provedor de identidade ou autoridade de certificação (Reed et al., 2019).

VC pode representar todo e qualquer tipo de informação que uma credencial física representa. Por fazer uso de tecnologias, como por exemplo, assinaturas digitais, faz com que seu processo de verificação contra fraude seja mais evidente e mais confiável, do que os certificados físicos (Dalgleish et al., 2017).

[1] Valores baseados na tabela de emolumentos de 2018, disponibilizado pelos tribunais de justiça estaduais: `http://www.anoreg.org.br/site/tabela-de-emolumentos/`

2.1.5.2 Processo de verificação

Este é o processo na qual uma terceira parte verifica a autenticidade do certificado, que pode ocorrer de três formas:

i. Baseada em características de segurança, ou seja, em marcas construídas dentro do próprio certificado. Isto poderia incluir medidas para verificar o selo de autenticidade, um papel especial seguro, uma assinatura.

ii. Verificação junto ao emissor, ou seja, uma terceira parte é contactada para perguntar ao emissor original, se realmente foi emitido o certificado. Neste caso, o emissor depende de uma consulta à sua base de dados para comparar o documento com os registos lançados.

iii. Verificação comparativa baseada em um banco de dados centralizado. Neste cenário, o emissor teria registado todos os certificados emitidos em um banco de dados, na qual é permitido a qualquer pessoa consultar este banco de dados e assim realizar a comparação.

2.1.5.3 Processo de partilha

Os certificados não apenas determinam quem pode transmitir o conhecimento, mas também ajudam a identificar os membros de uma comunidade que tenham habilidades semelhantes (Schmidt, 2017). Essa informação pode ser muito útil, como por exemplo, identificar em uma determinada região, pessoas com habilidades para tratar um problema específico que emergiu naquele local. Com os modelos atuais em papel, é muito difícil conseguir indexar toda essa informação dos credenciados, e mesmo que fosse fácil, ainda haveria o problema da verificação da veracidade dos dados. Com a tecnologia *blockchain*, isso pode ser possível, por conta da distribuição que ocorreria nas informações dos credenciados e pela fiabilidade oferecida.

Desta forma, este processo refere-se quando um destinatário de um certificado o compartilha com outras pessoas. Existem três formas de fazê-lo (Grech and Camilleri, 2017):

i. Transferindo diretamente o certificado, ou a cópia do certificado para o interessado. Essa transferência pode ser feita de forma física ou por meio eletrónico.

ii. Armazenar o certificado sob guarda de um responsável, que fica autorizado para compartilhar somente com as pessoas que forem autorizadas pelo portador original.

iii. Colocar em um registo público o certificado, onde qualquer pessoa possa consultá-lo.

2.2 Conclusão

Este capítulo apresentou os principais conceitos dos diferentes componentes utilizados no *blockchain*. A tecnologia utiliza criptografia assimétrica e assinaturas digitais para registar as transações dentro do bloco. As verificações de registos dentro do bloco são possíveis independentemente do tamanho da rede, graças a técnica empregada pela raiz de *Merkle*. As funções unidirecionais, são aplicadas em diversos pontos da *blockchain*, como por exemplo, calcular dados do cabeçalho do bloco. Também foi introduzido por este capítulo os processos envolvidos na gestão de certificados. Os conceitos aqui revisados, servirão de apoio para o entendimento dos demais capítulos.

Capítulo 3

Tecnologia *Blockchain*

3.1 Introdução a *blockchain*

Blockchain é uma tecnologia que propõe organizar e distribuir um conjunto de dados de forma segura, gravados em um livro de registos compartilhados e distribuídos, que preserva as informações registadas de forma perpétua e as relacionam cronologicamente através de uma cadeia de blocos (Turkanović et al., 2018). Desde sua exposição em 2008 (Nakamoto, 2008) através da aplicação *Bitcoin*, o nível de fiabilidade logo chamou a atenção da comunidade académica, e percebeu-se que tais propriedades de segurança poderiam oferecer diferenciais comparados aos convencionais mecanismos de banco de dados.

O conceito sobre computação distribuída vem dos anos 80, cujo objetivo na época era escalar processamento, potencializar performance em banco de dados. Embora tenha atingido esses objetivos, surgiram algumas questões importantes nesse modelo como: Quem pode ser confiável na rede para replicar uma informação confidencial? Qual a identidade real de quem está *on-line*? Quem pode manipular o quê no banco de dados? Como gerir conflitos em caso de processamento simultâneo? Com o passar dos anos, diversas pessoas têm tentado fornecer soluções para estas questões, mas diante de suas características revolucionárias, a *blockchain* pode ser uma alternativa capaz de oferecer um caminho diferente para os problemas citados (Fundation, 2018).

A *blockchain* incorporou o princípio da distribuição e acrescentou a criptografia. Como resultado ela conseguiu obter uma maneira de compartilhar informações de uma forma realmente segura, a prova de adulteração, ganhando reconhecimento ao garantir propriedades como transparência e fiabilidade.

Ao contrário dos banco de dados tradicionais, a proposta dessa tecnologia foi de organizar os dados em uma cadeia de blocos, que relaciona os registos através de uma chave calculada computacionalmente pelas informações que compõem o próprio bloco e tem como regra, a obrigatoriedade de novos registos serem apensos no final da cadeia, de forma a respeitar uma ordem temporal de criação.

Esse conjunto de blocos formam um livro de registos, que contém todo o histórico de

todas as transações que foram operadas na rede. De modo permanente, cada registo feito no livro de registos não pode ser mais apagado sem comprometer a integridade do livro, o que remete a mais uma propriedade importante, a imutabilidade.

Na prática, quando existe uma real necessidade de corrigir um lançamento feito de forma errónea, o procedimento é sempre gerar uma nova transação, que de alguma forma indique que o lançamento anterior é inválido e que deve prevalecer o último registo gravado.

Toda operação dentro de uma rede *blockchain*, é escrita utilizando uma chave privada, direcionada de um endereço público a outro. Em uma analogia com o sistema de caixa de correios, somente quem tem a senha do endereço do remetente, consegue escrever uma mensagem e enviá-la para um destinatário. A diferença entre uma operação e outra, é que o endereço público na *blockchain*, não diz nada em relação ao proprietário. Isso remete a uma outra propriedade importante, a do anonimato. Ao longo desse capítulo, na secção 3.1.3 será abordado novamente esse assunto.

A medida que os estudos avançam sobre o tema, especialistas convergem sobre a importância da tecnologia e qual o potencial de sua expansão (Taylor, 2015).

Iansiti e Lakhani (Iansiti and Lakhani, 2017a)(Iansiti and Lakhani, 2017b) acreditam que a tecnologia disruptiva da *blockchain* possa confrontar os tradicionais modelos de negócios, e transformá-los. Diedrich complementa essa afirmação: "mais do que usar o *blockchain* para gerenciar uma empresa: o código é toda a empresa", (Diedrich, 2016). O modelo de execução distribuída e autónoma possibilita o surgimento das *Decentralized Autonomous Organization* (DAOs) (João, 2018), um novo modelo de negócios que surgiu com a chegada da *blockchain*.

De acordo com os trabalhos (Schwab, 2017) (Chung and Kim, 2016) a *blockchain* pode ser considerada como parte da quarta revolução industrial. Segundo (Swan, 2015), o desenvolvimento de aplicações *blockchain* pode ser dividido em três estágios:

- *blockchain* 1.0, desenvolvimento de aplicativos para criptomoedas em redes P2P, com a finalidade de uso exclusivo em sistemas de pagamento financeiro.
- *blockchain* 2.0, a extensão de aplicações com escopos que vão além de transações de pagamento, como por exemplo, estoque, empréstimos, propriedade intelectual.
- *blockchain* 3.0, uso mais abrangente, permitindo aplicações mais complexas que geralmente são demandadas por áreas de: saúde, governamentais, educacionais.

Considerando essa classificação, as aplicações que estão mais maduras e conseguem explorar a tecnologia de uma maneira satisfatória, ainda se enquadram nas fases 1.0 e 2.0 (Chen et al., 2018). Isso significa que ainda há muito potencial a ser explorado e que de fato ainda há muitas questões a serem resolvidas, quando considerado o uso da tecnologia para aplicações comerciais. Na área da educação, na qual foca-se esse trabalho, existem

vários estudos de como a tecnologia pode ser aplicada (Sharples and Domingue, 2016).

3.1.1 Tipos de rede *blockchain*

Existem três principais tipos de classificações de rede *blockchain*: pública, privada e consórcio (Turkanović et al., 2018), que são definidas a seguir:

3.1.1.1 Pública não permissionada

O ambiente é livre para qualquer participante que queira entrar na rede, no entanto, as regras de validação das transações são pré-definidas e não podem ser alteradas por nenhum membro. Curiosamente, mesmo sem autonomia para alterar as regras, nesse tipo de rede, qualquer membro tem chance de participar do processo de consenso. Por operar nesse contexto, esse tipo de ambiente oferece uma transparência a nível público, já que é permitido a qualquer um visualizar as operações que estão registadas no livro de registos, mesmo que o integrante da rede esteja sob o anonimato. O exemplo mais comum desse tipo de rede não permissionada é o *Bitcoin* (Grech and Camilleri, 2017).

3.1.1.2 Privada permissionada

Nesse tipo de rede as regras são delegadas conforme o interesse do negócio. A organização que detém o controle da rede, pode definir por exemplo quais serão os usuários que podem decidir sobre o consenso ou como será feita a gestão de adesão de novos membros. No contexto de rede privada permissionada, os participantes podem consultar apenas as operações que lhe são pertinentes e o controle sobre os mecanismos de consenso passam a ser operados por um grupo que foi pré-definido para esse papel. Com essas características citadas, esse tipo de categoria de *blockchain* acaba não sendo tão democrática como a rede pública.

3.1.1.3 Híbrida consórcio

O consórcio é uma categoria que propõe oferecer algumas propriedades da rede pública com outras da rede privada, dentro de um mesmo ambiente. Enquanto na pública o foco é tornar o dado contido na *blockchain* acessível e visível a todos, na privada o objetivo é justamente o oposto. No entanto, em alguns cenários seria interessante que o acesso a rede continuasse de forma pública, mas que fosse possível tornar alguns dados encriptados para preservar a privacidade, e o anonimato de um participante. Por questões de performance, determinados tipos de aplicações não necessitam de uma prova de trabalho (PoW) computacionalmente tão custosa, como por exemplo da *Bitcoin*, e que eleger um

grupo de nós responsáveis pelo processo de consenso, otimizaria o sistema e ofereceria ganhos de desempenho para a aplicação (Zheng et al., 2017).

3.1.1.4 Comparativo entre as redes

Analisando os conceitos das classificações, nota-se que dependendo da aplicação a ser construída, torna-se mais interessante a utilização de um tipo de formato de rede que a outro.

No caso das criptomoedas cujo objetivo é oferecer mecanismos para troca de ativos diretamente entre negociadores, sem que haja um banco central ou órgão governamental regulador, e que permita a livre participação de novos membros, é necessário que seja usada uma *blockchain* pública. Por tratar diretamente com questões financeiras, é fator crítico de sucesso que a rede opere com segurança máxima e que seja completamente descentralizada. Essas propriedades são totalmente atendidas na rede pública como mostra a tabela 3.1. Já aplicações que requeiram alta taxa de transmissão não são muito indicadas para operarem em um ambiente público.

Conforme a tabela 3.1 a rede consórcio varia o grau de sua segurança, por conta de combinar algumas características da rede pública com a privada. Quanto mais próximas suas propriedades de uma rede pública, mais segura ela é classificada, e quanto mais próximas de uma rede privada, menor é a sua segurança. Isso se deve ao fato principalmente da rede pública ter um tamanho muito maior que a privada. Essa é uma das características que impactam diretamente na proteção contra fraude. Por exemplo, para um atacante alterar uma transação antiga, seria necessário alterar todos os demais elos. Mesmo em redes com grandes volumes de dados, a deteção de um registo malicioso dentro da rede é facilmente percebido, graças ao mecanismo da árvore de *Merkle* que a tecnologia emprega, detalhada na secção 2.1.4.

Por ter uma maior flexibilização do que as redes públicas e privadas, e continuar oferecendo descentralização, as redes do tipo consórcio são indicadas para o desenvolvimento de aplicações que necessitam de algum tipo específico de controle, e ainda sim, pretendem prover um maior grau de segurança.

Tabela 3.1: *Comparativo de características entre tipos de blockchain, adaptado de (Zheng et al., 2017)*

Característica	Pública	Privada	Consórcio
Consenso	Todos os membros	Restrito a organização	Membros eleitos
Leitura	Todos os membros	Arbitrária	Arbitrária
Segurança	Alta	Questionável	Alta ou média
Eficiência	Baixa	Alta	Alta
Centralizada	Não	Sim	em Parte

3.1.2 Desafios com legislação - Europa e Brasil

A natureza disruptiva da *blockchain*, além de ter produzido novas questões técnicas, também gerou impacto em alguns temas da sociedade. Nessa secção, é levantado um ponto em que a tecnologia pode entrar em conflito com as novas legislações de privacidade de dados, que estão surgindo no mundo.

Em 2018 através de uma reportagem investigativa do New York Times [1], veio a público que os dados de milhares de usuários do Facebook que participaram de um *quizz*, organizado pela empresa de consultoria Cambridge Analytica, tiveram seus dados divulgados.

Impulsionada por eventos como este e atendendo ao clamor social inerente a privacidade, foi criada uma legislação na Europa, a *General Data Protection Regulation* (GDPR), para proteção de dados. Aprovada no ano de 2016 e vigorando desde o dia 25 de maio de 2018, ela intitula-se como a mais importante mudança em regulação de privacidade de dados dos últimos 20 anos (Union, 2018).

O Brasil, seguindo a linha de outros países, aprovou no dia 10 de julho de 2018 sua lei de proteção e privacidade de dados, a *Lei Geral de Proteção de Dados Pessoais* (LGPD). Segundo o artigo 18 inciso VI da referida lei [2], o titular dos dados pessoais tem direito a obter a qualquer momento, mediante requisição, a eliminação dos dados que foram tratados com seu consentimento.

Há muito tempo usuários já vinham sendo prejudicados por falhas relacionadas a sistemas centralizados, que colocavam em cheque seus direitos a privacidade e liberdade. Diante da propriedade da imutabilidade e distribuição da *blockchain*, como a GDPR e a LGPD poderão cumprir os seus papeis de legislar nessa questão, se é praticamente impossível apagar um dado gravado na rede? Como fica o direito ao esquecimento dos usuários?

[1] `https://nytimes.com/2018/03/17/us/politics/cambridge-analytica-trump-campaign.html`

[2] `http://www.planalto.gov.br/ccivil_03/_Ato2015-2018/2018/Lei/L13709.htm`

Este é um exemplo de como a regulamentação, por vezes, busca resolver um problema olhando no espelho retrovisor, ao em vez de olhar a estrada à frente. Quando os formuladores de políticas europeias estavam debatendo e finalizando aspetos do GDPR, a tecnologia *blockchain* não era considerada como mais do que tendo um nicho de aplicação, muito específico (Revoredo and Borges, 2018).

Este trabalho cita esses pontos da lei Europeia e Brasileira porque há um risco para a tecnologia que questões legais como essas, possam ganhar força e serem impeditivas. Uma empresa que visa investir, por exemplo no Brasil, com um produto baseado em *blockchain* pode pesar negativamente na decisão a existência de uma ameaça externa como essa, e acabar fazendo-a desistir do investimento. Neste ponto, a lei parece ser incompatível com a tecnologia, e torna-se necessário rever o que pode ser feito, em relação a isso, tanto na tecnologia quanto na lei. Discutir a questão de regular tecnologias disruptivas como a *blockchain*, ainda mais quando estão em sua fase de juventude, pode travar a sua evolução e comprometer o seu verdadeiro potencial. Há muitas incertezas, e o futuro da tecnologia ainda não pode ser previsto, por isso já existem críticas sobre a real necessidade de começar um processo de regulação, sobre uma incerteza.

3.1.3 Blocos e suas transações

O ponto mais marcante da tecnologia *blockchain*, é a forma como os blocos são estruturados. De modo encadeado, um bloco é ligado a outro, formando uma grande cadeia, que é chamada de livro de razão. Como pode-se notar pela figura 3.1, cada bloco é composto por duas principais áreas: cabeçalho e conteúdo. É no cabeçalho que ficam armazenadas informações de controle do bloco, como por exemplo, o condensado do bloco anterior, utilizado para ligar o bloco atual com o bloco anterior. Já na área em que refere-se ao conteúdo, ficam armazenadas as transações responsáveis pela transferências de ativos.

É interessante notar, que pelo fato de cada bloco armazenar em seu cabeçalho o condensado do cabeçalho do bloco anterior, a adulteração de um bloco da cadeia somente seria possível, se todos os demais blocos anteriores também fossem alterados. Na prática isso demandaria um processamento elevado, uma vez que há sempre novos blocos sendo adicionados ao livro. Outro ponto que dificultaria ainda mais essa alteração, seria a distribuição. Estas funcionalidades fazem com que a *blockchain* ganhe uma propriedade, muito importante, de imutabilidade.

Há também outra informação muito importante no cabeçalho, chamada de raiz de *Merkle*. De nada adiantaria ter um mecanismo tão eficiente capaz de tornar os registos imutáveis, se não fosse computacionalmente viável verificá-los. Na secção 2.1.4 é dada a definição de como calcular a raiz de *Merkle*.

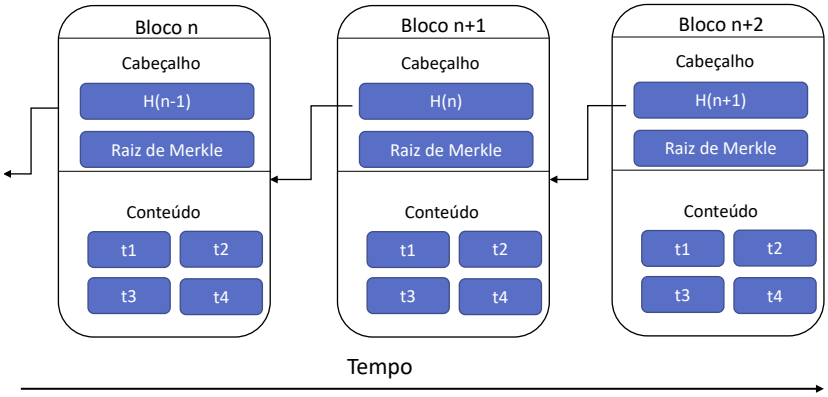

Figura 3.1: *Estrutura simplificada da blockchain, adaptado de (Grech and Camilleri, 2017)*

O ativo é um objeto simbolizado, que tem a sua representação binária armazenada dentro do livro de razão, como por exemplo: um veículo, um contrato, ou uma moeda. Basicamente as transações servem para realizar a transferência de um ativo de uma conta para outra. O proprietário de um ativo, consegue movimentá-lo dentro da *blockchain*, através do uso de uma conta capaz de registar uma transação dentro do bloco (Palma et al., 2019).

A conta é identificada por uma chave pública e controlada por sua correspondente chave privada. Como a chave privada fica armazenada de forma segura com o proprietário do ativo, somente ele pode manipulá-lo. A chave publica, garante duas propriedades interessantes: o anonimato (não é possível descobrir dados ou informações do proprietário do ativo) e identificação (se necessário, o proprietário consegue provar que ele é o dono de um ativo que está vinculado a um endereço público, através de sua chave privada).

Todas as operações envolvendo os ativos dentro do livro de razão são realizadas em uma ordem cronológica de tempo, ou seja, não há maneiras de editar uma ação sobre um ativo, que já tenha sido confirmada. A seguir é demonstrado, de forma simplificada, como é feito um registo de uma transação no livro de razão.

A tabela 3.2 apresenta uma operação simplificada de registo de transferência de moeda. No exemplo, alunos desejam realizar um pagamento para a universidade *Universidade Fernando Pessoa* (UFP). O primeiro consiste em tornar a transação anónima, ou seja, os atores da operação deixariam de ser identificados pelos seus nomes e passariam a receber um endereço público:

1. ms7WhwMSYA (Fernando Vidal)
2. msfq9pLNxY (Maria)
3. mxLJbrBYE5 (Alice)
4. mi85QzFKBC (UFP)

Tabela 3.2: *Registo de transferência de ativos simplificada*

De	**Para**	**Ativo**
Fernando Vidal	UFP	USD 100,00
Maria	UFP	USD 89,00
Alice	UFP	USD 103,00

O segundo passo é oficializar o pedido de registo da transação para dentro de um bloco. Nota-se pela tabela 3.3, que os alunos estão tentando enviar valores do endereço de suas contas, para o endereço de conta da UFP. Neste momento é gerado o identificador da transação, calculando-se o condensado do conteúdo da mensagem. Para este exemplo, concatena-se os endereços públicos dos alunos, universidade e valores dos ativos, resultando na seguinte expressão:

$$Txid1 = H(ms7WhwMSYA +\!+ mi85QzFKBC +\!+ '100,00')$$

Cada aluno utilizando a sua correspondente chave privada, assina sua própria transação, deixando-a em uma espécie de área temporária. Este é um processo importante, porque garante que não haja transações com endereços públicos inexistentes, ou mesmo o uso de um endereço por pessoas não autorizadas. As transações pendentes ficam aguardando para serem validadas pelo mecanismo de consenso, que é melhor detalhado na secção 3.1.4.

Tabela 3.3: *Registo de transferência de ativos simplificada, aguardando aprovação*

De	**Para**	**Ativo**	**Assinada**	**Status**
ms7WhwMSYA	mi85QzFKBC	USD 100,00	Sim	Pendente
msfq9pLNxY	mi85QzFKBC	USD 89,00	Sim	Pendente
mxLJbrBYE5	mi85QzFKBC	USD 103,00	Sim	Pendente

O terceiro e último passo é dado após a confirmação dos mecanismos de consenso. As transações pendentes já tiveram as suas assinaturas digitais verificadas, e também já foi criado um novo bloco para registá-los. Agora é necessário torná-las imutáveis. Seria muito perigoso deixá-las sem nenhum tipo de mecanismo de proteção contra adulteração, considerando que elas ficarão distribuídas localmente para todos os participantes da rede. Esta proteção é feita incluindo no cálculo do condensado do bloco, a informação do condensado do bloco anterior. A seguir, baseado nos dados da tabela 3.4, esta operação de cálculo de condensado do bloco é demonstrada:

Tabela 3.4: *Registo de transferência de ativos simplificada, gerado identificador da transação*

Txid	**Status**
2c9a2d7560f9c1e27d69294546ffce4fac79a45cb643d28d30fd27dd45e54afa	Pendente
8579a85857fd9e73b384616dad153a7245dd9d26af5c640b8ad5b5bda1b37438	Pendente
93f7d6b3fdbfd4296029875e0437d5583a39b967bb5ab7b77fd1902f56a34ed3	Pendente

Primeiramente calcula-se o condensado, concatenando todas as transações que deverão compor o bloco.

H(b')
(
2c9a2d7560f9c1e27d69294546ffce4fac79a45cb643d28d30fd27dd45e54afa ++
8579a85857fd9e73b384616dad153a7245dd9d26af5c640b8ad5b5bda1b37438 ++
93f7d6b3fdbfd4296029875e0437d5583a39b967bb5ab7b77fd1902f56a34ed3
)

H(b')=**3778a70a39e258861a29ffc66d088490d387545dca11810a0633a5ec0b9cf6b3**

Em sequência, concatena-se o valor obtido por H(b'), com o condensado do bloco anterior, e recalcula-se o condensado final do bloco. Para este exemplo, considera-se um hipotético condensado anterior com o valor de: 0a110db88685a1be06507221727549f4b7249a460d80d58d9145db163b9dc476.

H(b)
(
3778a70a39e258861a29ffc66d088490d387545dca11810a0633a5ec0b9cf6b3 ++
0a110db88685a1be06507221727549f4b7249a460d80d58d9145db163b9dc476)
)

Finalmente tem-se o valor final do condensado do bloco gerado:

H(b)=**66e859f1edf7cededc5614f05f6a1d14d1a8a061b0e54522605d3a5b774c8480**

Representando de forma gráfica, o bloco deste exemplo pode ser visualizado através da figura 3.2.

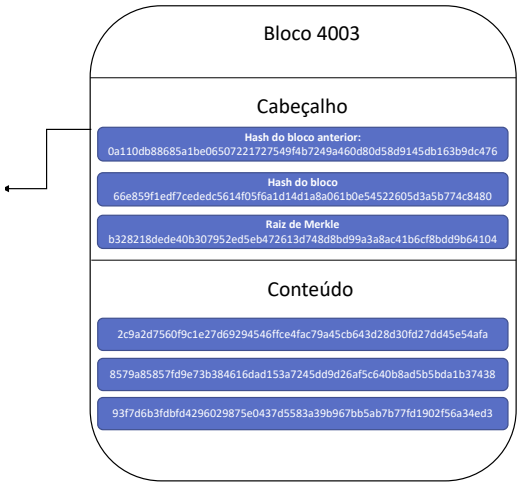

Figura 3.2: *Bloco gerado da simulação*

Existem diferentes implementações de *blockchain*, isso significa que podem haver variações de formatos dos blocos, bem como ocorrerem adaptações de algumas funções da tecnologia. Por exemplo, no *Ethereum* é utilizada uma implementação diferente da árvore de *Merkle*, denominada a árvore *Merkle* Patricia (Wood, 2017), no *Hyperledger* [1] a utiliza-se a função de condensado SHA3 ao invés da função SHA2 implementada pelo *Bitcoin*. Embora existam variações, de forma geral, a estrutura de todas implementações seguem um mesmo formato.

Para detalhar um pouco melhor o funcionamento dos blocos e suas transações, será utilizado como exemplo, a arquitetura da *Bitcoin*. Não são apenas os blocos que estão encadeados na *blockchain*, há também relacionamentos entre as transações que compõem um bloco. Isto garante por exemplo, uma ordem cronológica de ocorrência dos fatos.

Como pode-se notar pela figura 3.3, dentro de uma transação existem três principais áreas:

– Entradas apresenta uma lista com todos os endereços públicos que registaram uma entrada na transação. Cada entrada é registada com uma assinatura digital, para que possa ser possível verificar a sua autenticidade.

– Saídas apresenta uma lista com todos os endereços públicos que registaram uma saída na transação.

– Informações adicionais armazenam dados como por exemplo, o tamanho da transação em *bytes*, a quantidade de entradas/saídas juntas totalizadas, e a versão do protocolo.

[1] `https://openblockchain.readthedocs.io/en/latest/protocol-spec/`

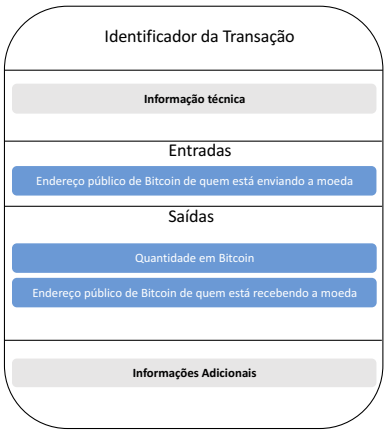

Figura 3.3: *Transação Bitcoin*

As operações de entrada e saída estão sempre relacionadas. Isso significa que sempre que existir um registo de entrada de um endereço público em uma transação, deve também existir um registo de saída para o mesmo endereço. Como pode-se notar pela figura 3.4, a entrada de uma transação é sempre sucedida a saída de uma outra transação, que tenha o endereço público correspondente.

O processo de validação usa essa correspondência, verificando a chave pública previamente inserida na saída, com a chave pública informada no campo da entrada. Logo, pode-se concluir que, caso elas sejam diferentes, a transação é inválida e esta não pode ser adicionada nesta cadeia. No entanto, caso esse primeiro critério seja atendido, é realizada uma verificação da assinatura digital do campo de entrada, utilizando a chave pública do campo de saída da transação anterior. O processo de validação da assinatura digital, pode ser conferido na secção 2.1.2.

As transações da *blockchain* são semelhantes com os registos de um livro de contabilidade. Cada lançamento é feito aos pares, debitando uma conta e creditando outra. No caso da transação, as contas de débito são referentes aos lançamentos do campo entrada e os créditos referentes aos lançamentos do campo saída. Como pode-se notar na figura 3.4, o crédito e débito são representados pelos sinais $+$ e $-$ respetivamente. No entanto, há um ponto que difere as operações das transações de um livro de contabilidade: a soma dos débitos e créditos não necessariamente totalizam a mesma quantia. Na prática, as saídas somam sempre uma quantia menor que as entradas e a diferença é chamada de taxa da transação, uma espécie de um pequeno pagamento coletado para o *mineiro*, i.e., o utilizador que conseguiu acrescentar a transação nesta cadeia (Antonopoulos, 2014).

Também pode-se notar na figura 3.4, que as transações podem ser classificadas em três tipos de operações (Antonopoulos, 2014):

- **Comum**: é a forma mais frequente de uma transação *Bitcoin*, que representa um simples pagamento de um endereço a outro, cuja operação envolve um troco do recetor para o emissor. Este tipo de transação tem sempre uma entrada e duas saídas (crédito recebido e o crédito do troco).

– **Distribuição**: é um tipo de transação que também é visto com frequência no livro de razão do *Bitcoin*. Sua função é distribuir um valor agregado de um endereço, a vários outros endereços, através do registo de diversas saídas. Geralmente esse tipo de operação é feito por entidades comerciais que querem distribuir fundos, como por exemplo, o processamento de uma folha de pagamento de funcionários.

– **Agregação**: é um tipo de transação que tem como objetivo juntar os valores de várias entradas e gerar um único valor agregado através de uma única saída. Isso representa o equivalente no mundo real, a troca de várias moedas e notas de pequeno valor, por uma única nota de um maior valor. Esses tipos de transações são geralmente feitas por aplicativos de carteiras que pretendem juntar pequenos valores, como por exemplo, recebimentos provenientes dos trocos.

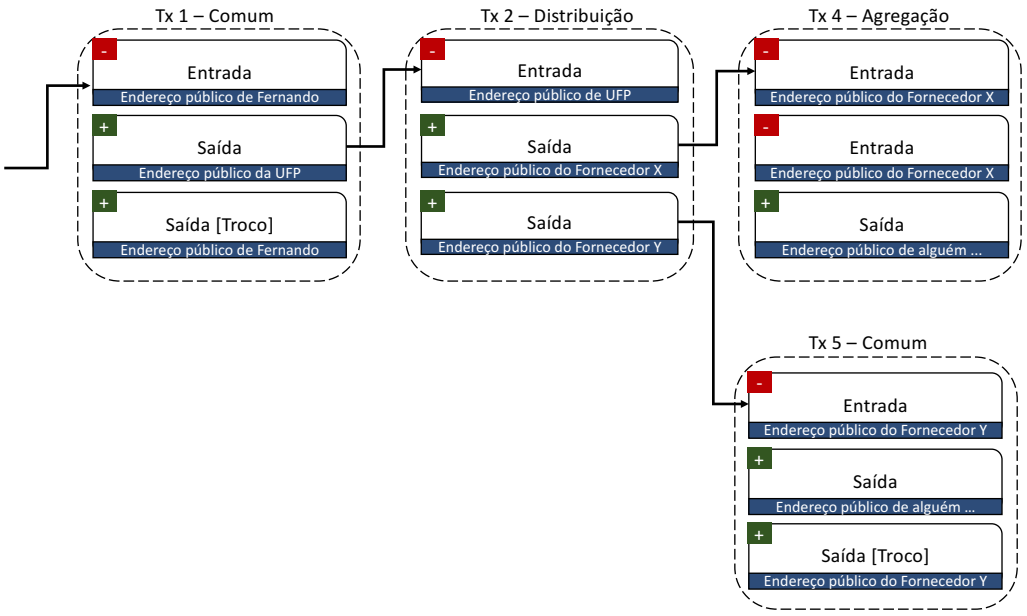

Figura 3.4: *Encadeamento e tipo das transações, adaptado de (Antonopoulos, 2014)*

No apêndice B, é ampliada a discussão de blocos e transações desta secção. No documento adicional, é feita uma análise de um bloco real da rede *Bitcoin*, na qual detalha-se as operações contidas no bloco, e realiza-se um cálculo para encontrar a raiz de *Merkle*.

3.1.4 Mecanismos de consenso

Uma boa maneira de explicar um conceito é realizando uma analogia com uma situação do mundo real. Os mecanismos de consenso podem ser comparados ao problema apresentado por (Lamport

et al., 1982), sobre um grupo de generais do exército de Bizâncio, a situação é a seguinte:

Existem vários grupos do exercito de Bizâncio cercando a cidade inimiga, na qual cada grupo têm o seu próprio general, que é responsável pelas tomadas de decisões daquela divisão. A comunicação entre eles é feita através de mensageiros, o que pode tornar a tarefa mais difícil. Por exemplo, estes podem ser capturados e a mensagem nunca chegar ao seu destino. Após observarem o inimigo, eles precisam decidir em conjunto um plano de ação. Entretanto, alguns dos generais podem ser traidores e acabarem por tentar sabotar o plano dos leais comandantes. Para garantir o sucesso da operação, é preciso que haja um algoritmo, que garanta ao menos as seguintes propriedades:

- A convergência de todos os generais leais a um mesmo plano de ação.
- Um pequeno número de traidores não pode influenciar o grupo de leais a irem pelo mau caminho.

Os mecanismos de consenso são algoritmos que fazem este papel descrito pelo problema de Bizâncio, nas redes *blockchain*. Neste tipo de ambiente distribuído, em que nenhuma parte é centralizada, torna-se um grande desafio garantir que o livro de razão espalhado entre vários nós permaneça consistente e confiável. Em outras palavras, é um método capaz de tomar decisões pelo grupo, pelo bem da maioria.

Seu papel é de tamanha importância dentro da tecnologia *blockchain*. Segundo (Antonopoulos, 2014), o mecanismo de consenso foi a principal inovação de Satoshi Nakamoto. Todas as demais propriedades do *Bitcoin*, incluindo moeda, transações, pagamentos e o modelo de segurança que não depende da autoridade central ou da confiança, derivam desta técnica.

Embora os modelos de consenso da *blockchain* objetivam manter um ambiente de igualdade e justiça, os seus efeitos colaterais já são percebidos. Por exemplo, na internet, a escalabilidade é um requisito crucial para o sucesso de uma aplicação. Os mecanismos mais difundidos como PoW e PoS, apresentam sérios problemas, como o excessivo consumo de energia e o fenómeno do rico fique mais rico ao favorecer os mineiros que têm mais recursos computacionais (Zheng et al., 2017).

Felizmente já existem diversas outras alternativas de mecanismos de consenso, com foco em aplicações diferentes, que podem resolver inúmeros tipos de situações. Nesta secção, além dos tradicionais PoW e PoS, são listadas algumas outras opções como DPoS, PoA , PBFT, Tendermint e dBFT:

3.1.4.1 PoW

Este foi o primeiro algoritmo de consenso, apresentado por (Nakamoto, 2008). Na realidade ele foi originalmente inventado para combater o envio em massa de correio eletrónico (Dwork and Naor, 1993) e adaptado por Nakamoto. Além do *Bitcoin*, ele também é implementado por outras criptomoedas como *Ethereum*, *Litecoin* e *Dogecoin* [1].

[1] `http://dogecoin.com/`

O princípio de seu funcionamento baseia-se em resolver um problema matemático complexo para descobrir um número que serve como uma espécie de autorização para registar um novo bloco na *blockchain*. O grau de dificuldade pode ser aumentado em tempo de execução do algoritmo, para garantir um tempo de bloqueio maior e controlar o fluxo e tamanho da rede. Em contrapartida, o processo de verificação da genuinidade do número encontrado é extremamente simples e não requer grande esforço computacional.

A seguir é detalhado o funcionamento deste algoritmo com base na implementação do *Bitcoin*.

É dada a definição de mineiros aos nós que estão calculando os valores de Bizâncio, e mineração o procedimento que eles realizam. Os mineiros tem basicamente duas funções principais (Courtois and Bahack, 2014):

- Verificar a correlação das transações *Bitcoin* e aprová-las.
- Produzir moedas através de um grande esforço computacional.

No PoW, os mineiros ficam constantemente calculando um valor de Bizâncio do bloco, na tentativa de encontrar o resultado esperado. De forma geral, esse cálculo é dado pela seguinte fórmula, na qual D corresponde aos dados do cabeçalho do bloco e N a um número aleatório:

$H = \text{SHA}(\text{SHA}256(D +\!\!+ N))$

Este valor H, precisa ser convertido em um valor inteiro de 256 *bits*, que terá cerca de 64 (ou mais) zeros a esquerda (Courtois et al., 2013). O número resultante deve ser menor do que o número alvo X (nível de dificuldade, informação propagada por toda a rede), que serve como uma espécie de parâmetro do sistema.

Caso o Bizâncio H resultante não atenda a expressão $Int(H) < X$, é modificado o valor de N e feita uma nova tentativa de cálculo.

Não há uma forma direta e sistémica de atingir o número desejado N, fazendo com que a mineração seja uma operação de acaso, aleatória. As chances de alguém vencer essa loteria são muito pequenas e proporcionais ao poder de computação multiplicado por 2^{-64}. O ajuste do esforço pode ser feito aumentando ou diminuindo o número alvo X. Quanto menor, mais difícil fica de atingi-lo e quanto maior, melhores são as probabilidades de alcançá-lo. Na prática, essa probabilidade diminui com o tempo, à medida que mais mineradores ingressam na rede e a dificuldade de encontrar um bloco aumenta (Courtois and Bahack, 2014).

Na hipótese de ao mesmo tempo, mais de um minerador resolver o problema computacional, somente um deles deverá ser escolhido vencedor. Essa situação é chamada de bifurcação e ocorre raramente na rede, em cerca de 1.69% dos blocos gerados (Decker and Wattenhofer, 2013). No caso de diferentes mineiros continuarem registando blocos sequencialmente na bifurcação, uma escolha deverá ser feita e um dos caminhos será descartado. O protocolo que resolve essa questão é chamado de regra da cadeia mais longa, e como o próprio nome diz, o segmento escolhido é aquele que for o mais longo.

Para evitar bifurcações prolongadas é necessário que o tempo de criação de novos blocos não

seja muito longo. Na *Bitcoin*, este tempo gira em torno de 10 minutos, então o prolongamento é limitado a esse tempo. A representação de uma bifurcação é dada pela figura 3.5.

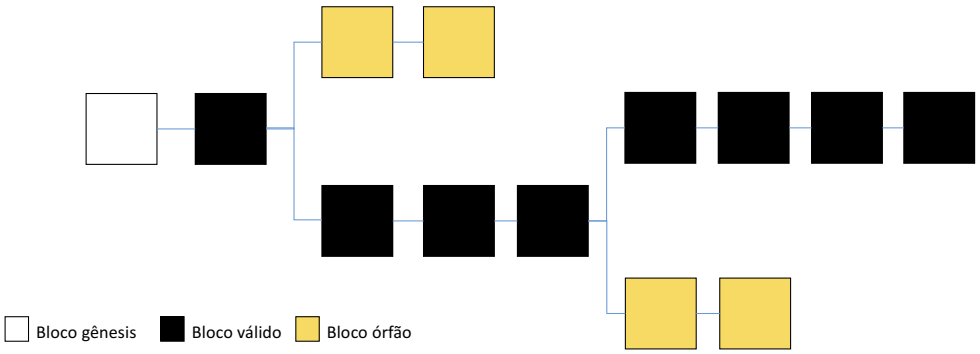

Figura 3.5: *Bifurcação e aplicação da regra da maior cadeia, adaptado de (Krzysztof Okupski, 2014)*

3.1.4.2 PoS e variantes DPoS, PoA

Este mecanismo surgiu como uma alternativa para reduzir o processamento apresentado pelo PoW e consequentemente economizar energia. A primeira criptomoeda a implementar o PoS foi o *Peercoin* [1], no entanto a aplicação é híbrida, ou seja, ainda se faz uso do processo de mineração de PoW (Cachin and Vukolić, 2017). A *Blackcoin* [2] também foi pioneira no uso deste mecanismo, desta vez por ser a primeira criptomoeda a usar puramente em seus protocolos o algoritmo PoS (Cachin and Vukolić, 2017). Existem diversas outras redes que utilizam este mecanismo de consenso como por exemplo *Decred*, [3] e *Nxt* [4].

O algoritmo parte do princípio de oferecer um voto de confiança a um determinado tipo de minerador, classificado de acordo com suas posses de moedas. Acredita-se que os usuários com mais moedas seriam menos propícios a atacarem a rede. De fato, essa característica desencoraja um nó de burlar as regras, uma vez que uma irregularidade poderia ocasionar na perda de suas moedas. Desta forma, uma suposta ação imprópria deve no mínimo compensar a possível perda de todos os ativos.

No entanto, o processo de seleção do nó, não é justo nem democrático. Percebe-se que há uma tendência dos nós mais ricos aumentarem ainda mais as suas riquezas, porque as recompensas recebidas pela geração de novos blocos, serão sempre distribuídas entre os que possuem os maiores recursos financeiros. Diante disso, novas soluções emergiram misturando este critério de participação com outros critérios, de maneira a deixar o processo mais justo.

Por exemplo, o *Blackcoin* (Vasin, 2014) usa a aleatoriedade para prever o próximo minerador. Ele usa uma fórmula que procura pelo menor valor de condensado, e combina com o peso da

[1] `https://peercoin.net/`
[2] `https://blackcoin.org/`
[3] `https://www.decred.org/`
[4] `https://nxtplatform.org/`

participação. *Peercoin* (King and Nadal, 2012) a seleção com a idade. Nesta rede, quem tem a moeda mais antiga, juntamente com um bom aporte de moedas, tem maiores probabilidades de gerar o próximo bloco.

O PoS ainda encontra-se em desenvolvimento e existem ainda muitas questões em torno de seu funcionamento (ul Hassan et al., 2019). Desde então, tem surgido variações do protocolo, como por exemplo DPoS e PoA com o propósito de reduzir o processamento face ao PoW, e melhorar questões de escalabilidade.

Por exemplo, no DPoS é possível eleger outros nós, denominados testemunhas, que votarão em seu nome. As testemunhas são eleitas e pagas logo após produzirem um bloco. O valor de pagamento é previamente acordado. Há acordos neste protocolo que permitem colocar em segunda ordem, um grupo substituto caso alguma testemunha saia, ou seja banido por ter tentado executar alguma tentativa maliciosa. Isso torna o sistema mais eficiente e resiliente a falhas.

O PoA é uma variação do PoS, frequentemente usado em redes de testes cujo principal propósito é experimental (Exemplo: *Rinkeby* e *Rospten Ethereum*). A ideia deste mecanismo é muito similar ao PoS, mas ao invés de utilizar como parâmetro o valor monetário dos nós, ele usa a identidade que é colocada como participação. Isso faz com que o PoA seja mais indicado em redes permissionadas onde as identidades dos nós são conhecidas, e já existem tratativas de permissões que poderão ser utilizadas para indicar quem poderá minerar novos blocos (ul Hassan et al., 2019).

3.1.4.3 PBFT e variantes *Tendermint*, dBFT

O PBFT é um protocolo de consenso que implementa o problema do exército de Bizâncio (Castro and Liskov, 2002). O algoritmo é projetado para isolar nós ou mensagens maliciosas de maneira a manter a rede confiável e funcional. Na prática, o PBFT consegue suportar até $1/3$ de participantes que estejam corrompidos com o objetivo comum da rede. Um exemplo de solução que utiliza este algoritmo é o *Hyperledger Fabric* (Sousa et al., 2018).

A criação de novos blocos é realizada através de processos organizados em rodadas. O primeiro nó a ser selecionado em uma rodada, deve atender a algumas regras específicas, já que terá papel importante no processo, como por exemplo ser responsável por realizar o pedido da transação. O processo completo pode ser dividido em três fases: preparatória, preparada e de entrega. Para que um nó passe para um próxima fase, é preciso que ele receba votos de pelo menos $2/3$ de todos os outros participantes (Zheng et al., 2017).

O *Tendermint* é uma variante do protocolo PBFT (Castro and Liskov, 2002). Enquanto no algoritmo PBFT qualquer cliente envia uma nova transação diretamente para todos nós, no *Tendermint* a tarefa de criação é somente limitada a nós validadores, usando um protocolo chamado de "gossip"protocol (Cachin and Vukolić, 2017). De forma semelhante, este opera em três passos: pré-voto, pré-entrega e entrega. Em contraste com o PBFT, no *Tendermint*, os nós precisam bloquear suas moedas para serem validadores. Se um validador é identificado como desonesto, suas moedas já estão reservadas e ele pode ser punido (Zheng et al., 2017).

E por último a variante dBFT, um mecanismo de consenso tolerante a falhas Bizantino proposto pela criptomoeda *NEO* [1] de origem chinesa. Este tipo de consenso vem ganhando força e sendo cada vez mais aplicado. É o caso da *Binance* [2], atualmente responsável por realizar as maiores trocas de moedas digitais do mundo, que decidiu utilizar este mecanismo de consenso.

De maneira semelhante ao algoritmo PoS, o dBFT comporta-se de forma escalável, permitindo a participação em grande escala de consenso, através do voto por procuração. O seu processo de delegação é feito pelos participantes portadores de uma chave eletrônica *NEO*. Com a chave eletrônica, o nó pode apoiar um outro nó, que recebe o nome de contador, em um processo de votação. A geração de novos blocos é feita por um grupo de contadores, que aplicam um mecanismo de consenso PBFT (NEO, 2014).

Uma das características mais marcantes deste algoritmo é que após a confirmação final, um bloco não pode ser bifurcado, ou seja, não há reversão ou revogação de transações. Esse fato faz com que a rede seja muito eficiente, mas no entanto, tenda a uma forma mais centralizada.

3.1.4.4 Comparativo entre os mecanismos

Os algoritmos de consensos são responsáveis pelo crescimento sustentável das redes *blockchain*. São eles que trazem fiabilidade a rede, e garantem que a descentralização disruptiva ocorra. No entanto, em cada implementação há vantagens e desvantagens que precisam ser analisadas para o tipo de aplicação e objetivos pretendidos.

Por exemplo, no *Bitcoin* percebe-se que houve uma preocupação muito grande com a segurança, por essa razão o sistema precisava de um mecanismo de consenso robusto que trouxesse a fiabilidade necessária para que os usuários colocassem dinheiro na rede. Diante desse objetivo o PoW foi fundamental para o sucesso da criptomoeda. No entanto, a mineração requer uma grande capacidade computacional, o que se traduz em uma alta quantidade de energia dispensada. Em aplicações que precisam de uma maior taxa de transmissão não é indicado este tipo de mecanismo de consenso.

O *Ethereum* deslumbrando uma melhor escalabilidade, vem sendo migrado [3] do PoW para PoS. Para uma criptomoeda a segurança é um fator importante, sendo que um ataque a sistema PoS é considerado uma operação computacionalmente muito custosa. Por essa razão, este sistema acaba sendo considerado como muito seguro.

Variantes de algoritmos existentes começaram a surgir, propondo resolver os problemas identificados na versão original. Na prática, as melhorias acabam criando novas possibilidades de aplicações do protocolo. Por exemplo, DPoS além de resolver o problema de centralização da função de mineiro do PoS, trouxe uma característica nova: os mineiros colaboram para a criação de blocos ao invés de competirem entre si. Essa característica potencializa o poder da rede no quesito de escalabilidade.

[1] `https://neo.org/`
[2] `https://www.binance.com/pt`
[3] https://github.com/*Ethereum*/wiki/wiki/Proof-of-Stake-FAQs

Outros mecanismos utilizam-se da identidade para atingir um bom desempenho na distribuição. O PBFT e suas variantes (*Tendermint*, dBFT) pré-selecionam os nós através da informação de suas identidades. No entanto, essa necessidade de saber previamente a identidade de cada minerador, faz com que a rede acabe perdendo a propriedade do anonimato.

Uma nova característica obtida por um novo protocolo, pode acarretar em um novo tipo de limitação ou restrição. Por exemplo, a alta taxa de transmissão apresentada pelos protocolos e variantes PBFT, como efeito colateral, restringe a entrada da rede. Comparados aos protocolos PoW, PoS e DPoS, os nós podem entrar na rede livremente.

Um bom algoritmo de consenso significa ser eficiente, seguro e conveniente (Zheng et al., 2017). No entanto estas propriedades carecem de ser avaliadas na aplicação em que este mecanismo será implementado. A tabela 3.5 demonstra algumas das diferentes características apresentadas por estes mecanismos de consensos.

Tabela 3.5: *Comparativo entre os algoritmos de consenso*

Característica	**PoW**	**PoS**	**PBFT**
Entrada de Participantes	Livre	Livre	Restrita
Taxa de Transferência	Baixo	Média	Alta
Centralizado	Não	Não	Sim
Democrático	Sim	Não	Sim

3.1.5 Taxonomia dos ataques a *blockchain*

Segundo (Walport, 2016) a segurança pode ser simplesmente definida como: "coisas que devem acontecer, acontecem; e coisas que não deveriam acontecer, não acontecem". A *blockchain* busca concretizar este pensamento. Como visto na secção 3.1.4, os algoritmos conseguem isolar comportamentos indevidos e garantir que coisas esperadas aconteçam. Nesta secção é revisado o conceito de rede distribuída P2P, cuja *blockchain* utiliza. Considerando este tipo de ambiente, é discutida a viabilidade de um ataque capaz de comprometer este tipo de rede.

Apesar dos recursos funcionais que a *blockchain* oferece para as aplicações, como em qualquer projeto, existe a necessidade de avaliação dos riscos envolvidos. Esta secção, avalia os possíveis tipos de ataques que uma rede *blockchain* pode sofrer, bem como as chances de serem executados com sucesso, e identifica soluções para evitá-los.

Segundo (Saad et al., 2019) os ataques podem ser classificados em três grandes categorias: **ataques relacionados a estrutura da *blockchain***, usadas na criação do livro de razão (e.g., bifurcação, blocos obsoletos, blocos órfãos); **ataques relacionados a arquitetura** P2P (e.g., mineração egoísta, ataque 51% , atraso do consenso, ataque *Distributed Denial of Service* (DDoS), ataque *Domain Name System* (DNS)); **ataques relacionados as aplicações da *blockchain*** (e.g., gastos

em duplicidade, roubo da carteira).

3.1.5.1 Ataques relacionados a estrutura da *blockchain*

Como visto na 3.5, uma bifurcação representa uma condição na qual nós na rede tem diferentes visões sobre o estado de persistência da *blockchain*. Essas bifurcações podem ser criadas involuntariamente por conta de algum mau funcionamento ou incompatibilidade entre atualizações de *softwares* clientes (Saad et al., 2019). Ao serem criadas, geram um estado de inconsistência a ser explorado por atacantes, causando confusões, transações fraudulentas ou comprometendo a confiança da rede (Kwon et al., 2017).

Um exemplo de grande bifurcação foi a que ocorreu em agosto de 2017 no *Bitcoin*, durante a condução de criação do *Bitcoin Cash* (Javarone and Wright, 2018). No caso do *Bitcoin Cash*, um grupo de programadores da *Bitcoin* resolveram aumentar de 1MB para 8MB o tamanho do bloco, e como a alteração não foi aceita pela maioria dos usuários, essa bifurcação acabou gerando essa nova criptomoeda. Outro exemplo de grande bifurcação, ocorreu em outubro de 2017, quando o *Bitcoin Gold* foi criado (Hanke, 2016).

Um outro caso de grande bifurcação ocorreu logo após *hackers* roubarem mais de um terço das moedas do aplicativo DAO (Siegel, 2016). Para reverter as transações e recuperar milhões de dólares, o *Ethereum* realizou uma grande bifurcação em sua rede, e exigiu consenso da maioria dos nós da rede.

Esses são bons exemplos de alvos, cujo um ataque poderia ter sido tentado. Imaginando que perante esses cenários, ocorresse um ataque DDoS, atrasando ou até mesmo impedindo que o consenso fosse atingido, os impactos seriam muito negativos em ambos os casos (Saad et al., 2019).

Um dos produtos gerados pelas bifurcações, são os blocos obsoletos. Esses blocos foram gerados de forma genuína, mas não serão aceites pela rede. Nesses casos, é possível que dois mineiros tenham encontrado uma solução válida, mas a rede eventualmente aceita apenas um vencedor e descarta o resto. Como resultado, todos os outros blocos válidos são descartados e não apensos na *blockchain*. A ocorrência de blocos obsoletos é dada com maior frequência em *blockchain* públicas por conta de suas condições competitivas (Saad et al., 2019). Na secção 3.1.5.2, é citado um possível ataque, chamado de mineração egoísta, que utiliza blocos obsoletos.

Outra forma de explorar inconsistências é através dos blocos órfãos. Desta maneira, mesmo sendo um bloco válido, por estar relacionado a uma origem inválida, ele também será descartado pela rede. A figura 3.6 apresenta uma cadeia, na qual podem ser encontrados blocos órfãos e obsoletos.

O primeiro bloco órfão criado no *Bitcoin* foi encontrado em 18 de março de 2015, em um período que começaram a surgir tais blocos. A tendência foi reduzindo em 2016 e de junho de 2017, até então nenhum bloco órfão tem sido adicionado. Uma característica de blocos órfãos é que são geralmente encontrados em criptomoedas onde a criação do tempo de blocos é mais curta (Saad et al., 2019).

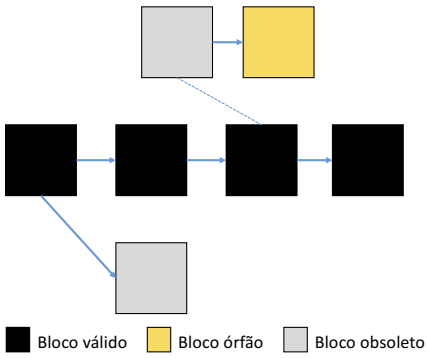

Figura 3.6: *Bloco obsoleto vs. bloco órfão adaptado de (Saad et al., 2019))*

3.1.5.2 Ataques relacionados a arquitetura P2P

As redes P2P surgiram com uma nova abordagem diante do tradicional conceito cliente-servidor. No modelo cliente-servidor, as estações e o servidor, tinham papéis bem definidos: solicitar e servir. Já nas redes P2P essas funções são agregadas ao nó da rede, podendo operar ora como cliente ora como servidor.

Essa característica torna as redes P2P tolerante a falhas e muito eficientes. Ao contrário do modelo cliente-servidor, a medida que nós entram na rede a sua comunicação e o seu potencial computacional aumenta. A *blockchain* opera em redes P2P, ficando distribuída, sendo capaz de isolar situações pontuais de falhas, sem comprometer o seu funcionamento.

Os ataques associados a rede *blockchain*, incluem por exemplo: ataques de DNS, mineração egoísta, ataques 51% , ataques de DDoS, atrasos no processo de consenso e ataques eclipse, nos quais este trabalho apresenta a mineração egoísta e o ataque 51%, podendo os demais serem conferidos em (Saad et al., 2019).

O ataque de mineração egoísta (Eyal and Sirer, 2018) é uma estratégia adotada por certos mineradores que tentam aumentar suas recompensas deliberadamente, mantendo seus blocos privados (Leelavimolsilp et al., 2018),(Grunspan and Pérez-Marco, 2018). O nome egoísta, se dá pelo fato de ao invés de liberar os blocos para o público, esses mineradores continuam a trabalhar em seus próprios blocos privados, a fim de obterem uma cadeia pública mais longa na *blockchain*.

Essas atividades conduzem a uma corrida de blocos entre uma cadeia pública de mineradores honestos, com uma cadeia privada de mineradores egoístas. Uma vez que o *blockchain* público começa a se aproximar do comprimento de sua cadeia privada, mineiros egoístas liberam seus blocos para reivindicar recompensas.

Os ataques de mineração egoísta podem produzir resultados indesejados a redes, por invalidar os blocos de mineiros honestos que contribuem para a *blockchain*. Além do mais, todas as transações do bloco minerado honestamente podem ser perdidas.

O ataque 51%, também conhecido como maioritário, se concretiza quando um atacante ou um grupo de atacantes conseguem maior poder computacional do que a porção honesta da rede, e

assim podem alterar e comprometer os registos armazenados na *blockchain* (ul Hassan et al., 2019).

Durante a competição da corrida aos blocos, na hipótese do atacante obter uma taxa de condensado superior a 50%, é obtido o controle sobre os demais mineradores, e a probabilidade de juntarem com sucesso seus blocos a rede será muito grande (B. Community, 2017). Com o controle em mãos, os atacantes são capazes de realizar as seguintes ações (Saad et al., 2019):

- Evitar que transações ou blocos sejam verificados, forçando-os a serem invalidados.
- Reverter transações durante o período de controle, permitindo duplos consumos.
- Criar uma grande bifurcação e dividir a rede.
- Impedir que outros mineradores procurem por blocos durante um curto período de tempo.

3.1.5.3 Ataques relacionados as aplicações da *blockchain*

O terceiro e último item refere-se aos riscos que aplicações de terceiros, podem trazer a tecnologia. Credenciais, como por exemplo chaves, que são armazenadas eletronicamente em uma carteira digital, podem sofrer ataques do tipo "roubo de carteira", e assim darem controle total de movimentação dos ativos ao atacante. Com a oferta de muitos aplicativos de terceiros fornecendo este tipo de serviço, usuários podem acabar optando por um sistema com vulnerabilidades, como por exemplo, armazenamento de dados em *plain*, e correr riscos de terem suas autenticações furtadas.

Um exemplo foi o caso do roubo de $50 milhões USD que ocorreu em 2016, do aplicativo *DAO*. Este *software* operava baseado em contratos inteligentes do *Ethereum* e por um erro em seu código, um atacante desconhecido conseguiu roubar as chaves eletrónicas que movimentavam moedas (Siegel, 2016). No mesmo ano, em agosto de 2016 , *Bitcoin*s que valiam na época $76 milhões USD foram roubados de uma plataforma de troca de moedas chamada *Bitfinex* (Baldwin, 2016).

3.1.5.4 Análise dos ataques

O *Bitcoin* é um bom exemplo de *blockchain* seguro. Mesmo diante de vários ataques por conta de sua popularidade e capital envolvido, tem se mostrado robusto e confiável. Por exemplo, em maio, agosto e novembro de 2017, blocos de memória do *Bitcoin* foram saturados com transações temporárias criando uma espécie de barreira, atrasando o processo de verificação de transação e aumentando a taxa de mineração (Saad et al., 2018). Os ataques não prejudicaram os ativos dos usuários, ou seja, as informações permaneceram íntegras. Como qualquer outra rede, a *blockchain* está sujeita a ataques do tipo DDoS e DNS, no entanto, pelas suas propriedades criptográficas e encadeamento, há uma segurança adicional perante os modelos de redes tradicionais.

Em relação aos ataques de 51%, embora teoricamente possíveis, são muito improváveis. As maiores redes de *blockchain* dispõem de uma alta taxa de condensados agregada, e atingir a maioria

requer um investimento extremamente alto. A tabela 3.6, apresenta 4 importantes criptomoedas, e simula o valor necessário a ser "investido"para executar este ataque. O que percebe-se, é que quanto maior a rede, mais segura ela fica, o que explica o motivo do *Bitcoin* ser mais seguro, quando comparado com redes de menor dimensão (Kroll et al., 2013).

Essa previsibilidade e estimativa do quão custoso seria atacar a *blockchain*, permite que novos algoritmos de consenso sejam criados ou melhorados, de maneira que a tecnologia possa oferecer mecanismos cada vez mais seguros. Isso foi demonstrado na secção dos mecanismos de consenso 3.1.4, na qual diversas propostas são apresentadas, e tornam o ataque cada vez menos exequível.

Tabela 3.6: *Custo de uma hora de ataque 51% em dólares, adaptado de (Saad et al., 2019)*

Criptomoeda	**Capacidade**	**Algoritmo de Chave**	**Taxa de *Hash***	**Custo**
Bitcoin	112.7B	SHA-256	35,604 PH/s	486K
Ethereum	49.5B	Ethash	222 TH/s	347K
B.Cash	14.9B	SHA-256	5,023 PH/s	68K
Litecoin	5.7B	Scrypt	327 TH/s	60K

Os fatos sobre os ataques ocorridos nas carteiras, ou nos roubos de chaves, não traduzem na insegurança da tecnologia. Em uma analogia com as chaves físicas, se alguém esquecer as chaves dentro de um veículo, caso ocorrer um roubo não significa que todos os veículos com o mesmo modelo de chave estão vulneráveis ao mesmo crime.

As bifurcações que abrem espaço para que atacantes gerem inconsistências podem ser controladas através do intervalo de criação de novos blocos. Quanto menor for esse parâmetro, maior é o número de blocos gerados, e consequentemente maior a probabilidade de ocorrerem: bifurcações, blocos obsoletos e blocos órfãos. Gervais et al. (Gervais et al., 2016), examinaram a redução do intervalo, observando 10.000 blocos simulando em NS-3 [1]. Como pode-se notar através da figura 3.7, o número de bifurcações é reduzido a percentuais extremamente baixos, quando expostas a intervalos maiores.

[1] `https://www.nsnam.org/`

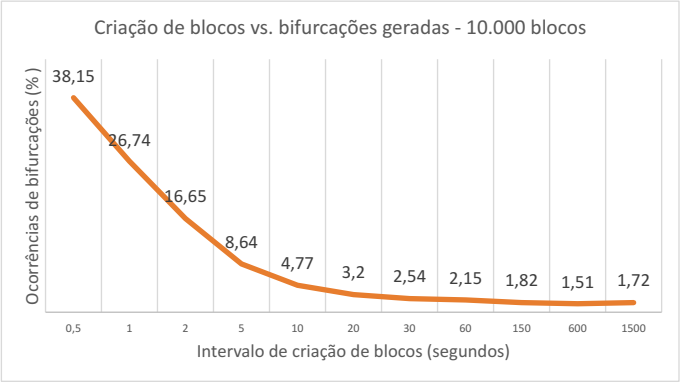

Figura 3.7: *Criação de blocos vs. bifurcações geradas*

3.1.6 Criptomoedas

As criptomoedas são um novo tipo de ativo digital, que ao invés de utilizarem material físico e autoridade centralizadora, dependem de protocolos criptográficos distribuídos para operarem como moeda. *Bitcoin* foi a primeira criptomoeda a ganhar popularidade, desde então milhares [1] de moedas alternativas têm surgido (Krafft et al., 2018).

Este trabalho apresenta o *Bitcoin* e *Ethereum*, mas não como moedas digitais, e sim como redes de *blockchain* que podem ser utilizadas em aplicações na área da educação.

3.1.6.1 Introdução *Bitcoin*

Com a popularidade conquistada pelo *Bitcoin*, informações sobre a tecnologia *blockchain* foram sendo divulgadas, e geraram-se algumas confusões nos conceitos. É comum o *Bitcoin* ser confundido como a própria *blockchain*, como a provedora de todas as demais aplicações que foram construídas, quando na realidade é o contrário. Embora ela tenha tido um papel crucial para a divulgação e propagação da tecnologia, o *Bitcoin* é um produto gerado dela e não faz parte do alicerce da *blockchain*. Isso significa que é perfeitamente possível construir outras soluções baseadas em *blockchain*, sem que haja nenhuma dependência de componentes da *Bitcoin*. Fazendo uma analogia é como se o sistema operacional fosse a *blockchain* e o *Bitcoin* uma de muitas aplicações que rodam nesse sistema.

Criada para operar em redes públicas, não permissionadas, o *Bitcoin* evidenciou um problema: seria incapaz de trabalhar com negociações que envolvam uma alta frequência de troca. Cada bloco do *Bitcoin* leva em torno de 10 minutos para ser minerado e tem uma limitação de tamanho de 1MB. Consequentemente a rede *Bitcoin* é restrita a uma taxa de 7 transações por segundo. Projetando um cenário em que nós realizam milhares de transações por segundo, o tempo de espera seria inaceitável. A figura 3.8 apresenta uma comparação do gasto de energia do *Bitcoin*,

[1] `https://www.investing.com/crypto/currencies`

em relação ao consumo de energia de outros países. Embora esse tipo de problema pode ser resolvido através da implementação de novos mecanismos de consenso (descritos na secção 3.1.4), para a aplicação de certificados académicos essa limitação não impacta negativamente o projeto.

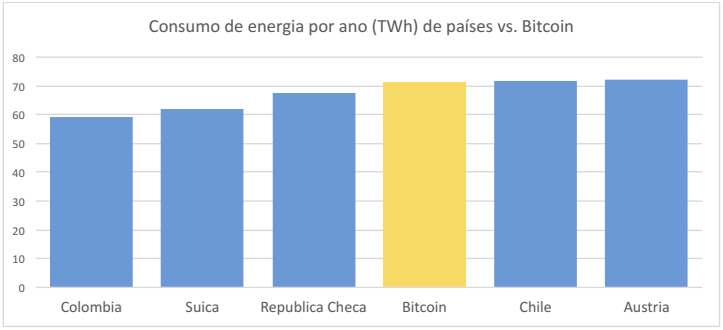

Figura 3.8: *Consumo de energia de países vs. Bitcoin, adaptado de (Saad et al., 2019)*

Para o desenvolvimento de aplicações, o *Bitcoin* oferece os seguintes ambientes:

- ***Mainnet*** [1]: é o ambiente oficial da *Bitcoin* em que estão as moedas reais e as suas respetivas transações realizadas. É altamente recomendado que apenas aplicações maduras e homologadas sejam configuradas para operar nesse tipo de rede. Uma vez que os dados não podem ser apagados, e há um custo envolvido em cada transação, este trabalho opta por não operar nesse modo, em virtude de ser um estudo de caso.

- ***Testnet*** [2]: é um ambiente preparado para homologação de aplicações, pelo fato de ser muito próximo ao ambiente *Mainnet*. Quando se instala esse tipo de rede é preciso obter créditos, em uma operação muito parecida com o processo de compra da *Mainnet*. Nesse caso, há páginas na Internet que fornecem créditos fictícios, os chamados *faucets*, que trazem ao desenvolvedor a experiência de operar com um número limitado de créditos. Este trabalho opta por testar a emissão, validação dos certificados nesse tipo de ambiente, proporcionando uma experiência mais próxima a de um cenário real.

- ***Regtest*** [3]: é um ambiente também para desenvolvimento de aplicações, no entanto dá um controle maior ao desenvolvedor do que o ambiente *Testnet* . Aqui é possível gerar um número ilimitado de créditos fictícios através de comandos do próprio ambiente, o que não é possível na *Testnet*. A sua importância para aplicações é de servir como uma primeira fase do desenvolvimento, já que limitar a quantidade de créditos, pode impactar negativamente no tempo da construção dos aplicativos.

Atualmente o tamanho da rede *Bitcoin* está em 290GB, considerando os atuais padrões de armazenamento, ela ocupa um espaço considerável. Dessa forma, um dos desafios a serem tratados pela

[1] `https://Bitcoin.org/en/glossary/mainnet`
[2] `https://Bitcoin.org/en/glossary/testnet`
[3] `https://Bitcoin.org/en/glossary/regression-test-mode`

tecnologia refere-se ao aumento do tamanho da rede. Enquanto não existam soluções concretas, seja para enviar dados para histórico, ou por exemplo reduzir as informações através de compactação, por conta desse grande volume, cada vez menos usuários são encorajados a mantê-la em disco, o que pode ocasionar em uma tendência de centralização, por conta da redução do número de usuários. Além disso, o uso de *data centers* pode contribuir para esse movimento de centralização, contrariando o princípio de P2P, cujo quanto maior o número de nós independentes, melhor se torna a rede.

3.1.6.2 Introdução *Ethereum*

Diferentemente do *Bitcoin*, na qual não se conhece a identidade de Satoshi, o *Ethereum* surgiu através de uma publicação (Buterin, 2014) de um jovem de 19 anos chamado Vitalik Buterin. O objetivo era de criar uma rede *blockchain* mais voltada ao desenvolvimento de aplicações. Nota-se, que o *Ethereum* é a plataforma onde a moeda é gerida e não a própria moeda.

Logo que a comunidade tomou conhecimento do documento público, algumas dezenas de pessoas começaram a oferecer ajuda e a contribuir para a evolução do projeto. No inicio de 2013, Vitalik e Gavin refinaram as ideias até então propostas, e juntos construíram o protocolo que ganhou o nome de *Ethereum* (Antonopoulos and Wood, 2013).

Um dos pontos mais marcantes do protocolo *Ethereum* é a presença de uma máquina virtual, *Ethereum Virtual Machine* (EVM), por conta de utilizar uma linguagem de programação *Turing-complete*, que a torna capaz de executar diversas linguagens de programação como: *C++*,*Go*,*Haskell*, *Java*, *JavaScript*, *Python*, *Ruby*, e *Rust* por exemplo. Um dos principais objetivos da EVM, é habilitar a criação de contratos inteligentes.

Apresentado por Nick Szabo, um contrato inteligente é um código determinístico, que executado por diferentes computadores, produzem uma mesma saída esperada (Szabo, 1997). Seu principal objetivo é permitir que partes que não se conheçam firmem acordos sem que haja a necessidade de uma terceira parte intermediando o processo. Isso é perfeitamente possível por conta da previsibilidade, transparência e fiabilidade que tais algoritmos oferecem. Uma vez definidas as clausulas do contrato, elas ficam expostas de forma transparente e são imutáveis (nenhuma parte pode alterá-las, ou interferir em sua execução).

O programa que representa o contrato inteligente é armazenado dentro da *blockchain*, e tem um identificador único calculado através do condensado do próprio conteúdo do código. Ao invés de optar por uma chave pública, ou por qualquer outro tipo de identificador para o contrato, utilizar um identificador variável através do cálculo do condensado do próprio código, garante que caso um único símbolo seja alterado do *script*, o contrato possa ser facilmente invalidado.

Nem todas as *blockchain*s foram preparadas para suportar contratos inteligentes, como por exemplo o *Bitcoin*, embora fosse perfeitamente possível armazenar tais contratos na rede, em forma de *scripts* executáveis que não poderiam ser alterados. No *Ethereum*, a implementação destes contratos inteligentes, se dá por uma configuração de um conjunto de condições pré-determinadas, construída através de um programa de computador, em uma linguagem de programação suportada

pela EVM.

No entanto, todo o poder computacional que a EVM proporciona a rede, depende de um "combustível digital"chamado *Ether*. O valor cobrado pela operação é calculado com base na quantidade informada de "gás", uma espécie de parâmetro para indicar a prioridade da ação, que direciona o poder computacional requerido, bem como o tempo de execução.

Ao invés de ser utilizado apenas como moeda digital, ou como uma alternativa de dinheiro, o *Ether* é um criptoativo que serve para "pagar"pelos recursos computacionais gastos para executar os contratos inteligentes demandados pelas aplicações descentralizadas (Inci and Lagasse, 2019). Por exemplo, uma aplicação de emissão de certificados, precisa de processamento computacional para registar um diploma na rede, e esta operação deve ser paga em *Ether*.

De forma semelhante ao *Bitcoin*, existem diversas redes de testes de desenvolvimento para *Ethereum*, que são largamente usadas antes da publicação do código oficial de uma aplicação na rede principal. Para este trabalho, foi utilizada a rede *Rospten* [1].

3.2 Conclusão

Este capítulo apresentou a tecnologia *blockchain*. Percebe-se que os tipos de redes oferecidos por essa tecnologia, suportam as mais variadas aplicações. No caso dos certificados académicos, as redes públicas e consórcio, são mais indicadas por conta de oferecerem uma maior distribuição. No capítulo também é apresentado o conceito de blocos e de suas transações, que contribui para o entendimento dos demais capítulos, na qual implementa-se a gravação dos registos dos certificados na *blockchain*. Os mecanismos de consensos demonstrados, salientam a segurança da *blockchain*, e revelam como o sistema sincroniza as informações. Essas informações contribuem para o entendimento dos demais capítulos, de como os certificados são replicados. E por último o capítulo relata as redes *Bitcoin* e *Ethereum*, em que ambas são implementadas na aplicação CertEdu.

[1] `https://ropsten.etherscan.io/`

Capítulo 4

Plataformas de *Blockchain*

4.1 Introdução

Agilidade é um requisito sempre muito exigido pelo mercado, e qualquer tecnologia por mais inovadora que se apresente, não tem vida longa caso não prove ser produtiva. De que adiantaria a possibilidade de execução de aplicativos em dispositivos móveis, se não houvessem plataformas e ferramentas que acelerassem as entregas e permitissem a escalabilidade na construção de programas. É provável que os *smartphones* não fossem atingir o sucesso que tem hoje, senão fosse graças aos esforços despendidos pelos seus fabricantes.

A mesma preocupação pode ser colocada para a *blockchain*, ainda mais diante de diversas oportunidades de aplicações que estão sendo prospetadas. Don e Alan Tapscott (Tapscott and Tapscott, 2016) escreveram: "Não é o *Bitcoin*, e seus ativos especulativos, que deveriam interessar a você", claramente referindo-se a esse imenso potencial ainda não explorado em aplicações.

Felizmente alguns grupos de empresas e também a comunidade académica, têm trabalhado sobre o tema e começaram a surgir as primeiras plataformas de construção ágil para *blockchain*s. Este trabalho apresenta nessa secção algumas dessas plataformas sob a ótica de utilizá-las no desenvolvimento de aplicativos na área da educação. Mediante as arquiteturas selecionadas, são abordadas e analisadas as características oferecidas para a criação de um aplicativo para emissão e verificação de certificados, bem como as limitações encontradas em cada solução.

4.1.1 Visão geral do *Blockcerts*

Com esforço colaborativo do MIT, e a iniciativa da empresa LM, nasceu uma das ferramentas mais promissoras relacionadas a emissão de certificados em *blockchain*, o *Blockcerts*. Em 2016, alinhados sobre o potencial do projeto, o MIT e a LM construíram o protótipo da solução no formato de código aberto, para que qualquer emissora que queira emitir seus certificados, possam assim fazê-los de forma gratuita (M. L. MIT Media Lab, 2016). No ano seguinte, cento e onze estudantes do MIT foram os primeiros a receberem seus diplomas nos seus *smartphones*, no formato digital.

Desde sua conceção, a facilidade tem sido um dos pontos centrais do projeto, que pode ser evidenciado nessa frase de Chris Jagers um dos co-fundadores da LM (platform, 2017): "É um grande obstáculo dizer aos alunos que gerem pares de chaves públicas-privadas para o *blockchain* do *Bitcoin*"e completa,"Ninguém tem a menor ideia do que você está falando". O aplicativo "wal-

let"resolveu esse problema de uma forma muito elegante. De forma transparente, os estudantes não precisam saber o que é uma chave pública ou privada, ao invés disso uma frase secreta de fácil entendimento é gerada para o usuário. Todo trabalho de gerar o par de chaves e sincronizar com a emissora é feita de forma transparente, oferecendo uma boa experiência para os estudantes.

Embora seja uma empresa privada, a LM defendeu o projeto de código aberto, defendendo que para seu mercado seria mais interessante se concorrentes utilizassem a mesma plataforma. Desta forma em sinergia com o MIT, o padrão vem ganhando força, principalmente pela sua flexibilidade de funcionar em qualquer tipo de *blockchain*.

A figura 4.1 apresenta um cenário de funcionamento da *Blockcerts*, envolvendo uma aluna, uma universidade e empregadores. O que pode ser evidenciado nessa arquitetura, é a autonomia conquistada pela aluna Joana, que tem a liberdade de enviar seus registos para quem ela desejar. Também ganham agilidade no processo, os empregadores que não precisam mais de consultar uma terceira parte, na validação das qualificações da Joana.

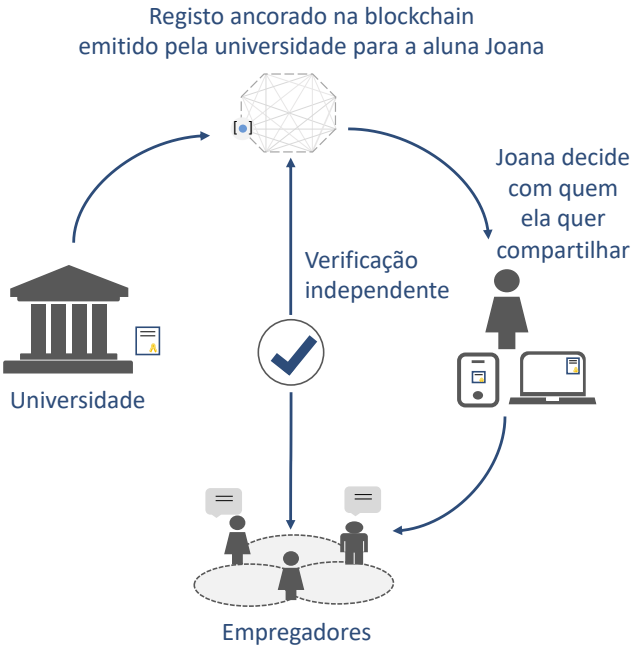

Figura 4.1: *Autonomia e agilidade em certificados com blockchain*

O padrão técnico do *Blockcerts* foi pensado para trabalhar com qualquer *blockchain*, evitando assim que o sucesso do projeto seja condicionado a evolução de um outro produto. Com esse tipo de abordagem, é possível acreditar que o *Blockcerts* tenha um ciclo de vida mais longo, já que a sua evolução tecnológica deve agregar inúmeras particularidades encontradas em cada *blockchain* implementada. Em 2016 quando o projeto teve seu início, a integração era somente possível com *Bitcoin*, mas logo se estendeu ao *Ethereum*. Em 2018, a Universidad del Rosario, na Colômbia (Iragorri, 2018) construiu a integração com *Hyperledger*.

[0]A figura 4.1 ilustra o funcionamento da solução *Blockcerts*, demonstrando a independência conquistada no processo, adaptada de `https://www.learningmachine.com/badges-and-blockcerts`

Outra característica importante é que o *Blockcerts* construiu todos os seus esquemas de forma alinhada com a especificação *Open Badges* (IMS Global Learning Consortium, 2018). O padrão *Open Badges* permite compartilhar através da web identificadores digitais portáteis, que possuem metadados sobre habilidades e conquistas, e podem ser verificados de forma eletrónica.

No entanto, ao invés de usar um simples formato de imagem o *Blockcerts* foi projetado para incorporar qualquer tipo de dados e gerar qualquer tipo de exibição. Além disso, esses registos são assinados criptograficamente pelo emissor, incluem chaves de destinatário e são registados em um *blockchain* para verificação posterior. Em resumo, o *Blockcerts* é fundamentalmente diferente do padrão *Open Badge*, oferecendo as seguintes inovações (Jagers, 2019):

- Evidencia de adulteração.
- Posse para emissor e receptor.
- Forma mais flexível.
- Compartilhamento em linha e desligado da rede.
- Verificação independente.

No estágio atual em que o projeto se encontra, depois de ter sido testado em vários ambientes e atingido um bom nível de maturidade, o *Blockcerts* é reconhecido como uma plataforma promissora na emissão e verificação de certificados (Oliver et al., 2018). A comunidade do projeto é muito ativa e são frequentes as publicações de atualizações, contendo correções e melhorias.

O projeto está escrito na linguagem de programação *Python*, que é uma linguagem de programação de alto nível, interpretada, imperativa, e orientada a objetos. Foi criada pelo holandês Guido Van Rossum em 1991, onde foi concebido a partir da linguagem existente na época, chamada ABC, e apesar de não ser tão recente, esta linguagem está em ascensão no mercado de tecnologia da informação, por ser uma ferramenta robusta e capaz de realizar tarefas complexas, e ao mesmo tempo de fácil manipulação e aprendizado (Borges, 2014). A figura 4.2 e 4.3 demonstram trechos do código que implementam dois conectores distintos para as redes *Ethereum* e *Bitcoin* respetivamente, que registam o condensado do certificado na *blockchain*.

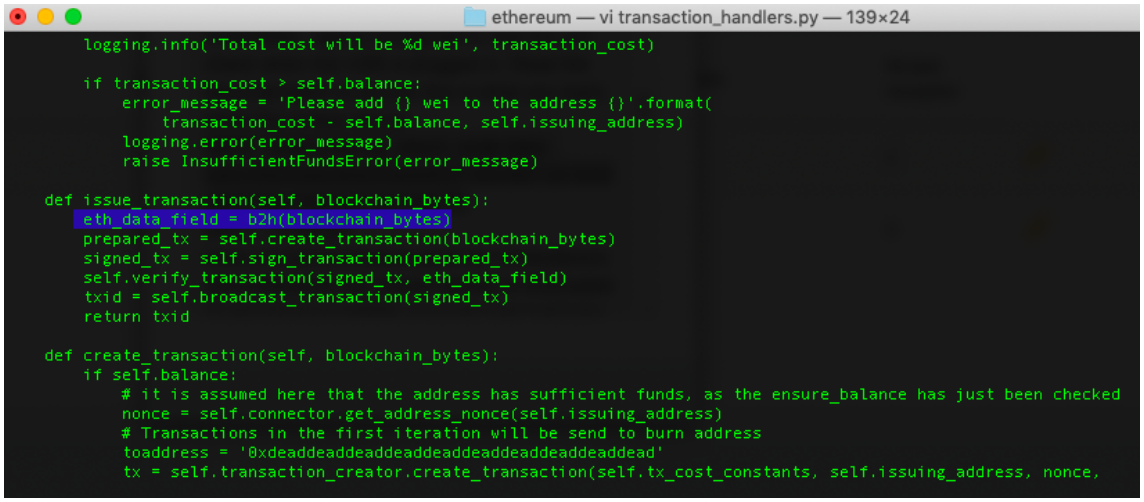

Figura 4.2: *Trecho de código do Blockcerts, implementando funções no Ethereum*

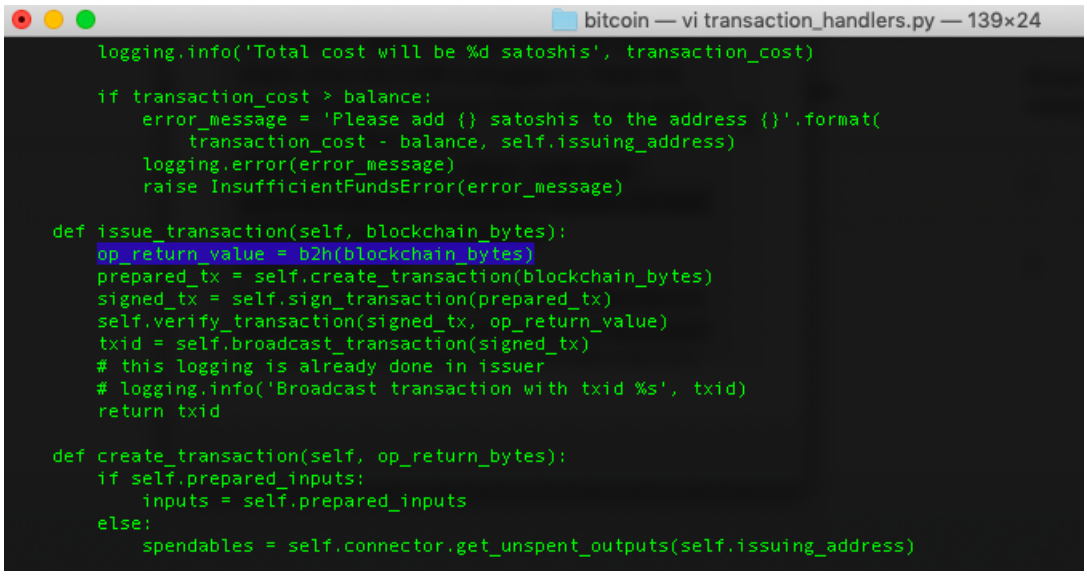

Figura 4.3: *Trecho de código do Blockcerts, implementando funções no Bitcoin*

Blockcerts usa diferentes camadas que trabalham juntas para criar os condensados para cada lote de certificados, emitindo-os na *blockchain* e, posteriormente, permitindo que plataformas web imprimam os certificados usando objetos JSON e verifique-os na *blockchain* (Oliver et al., 2018). Conforme ilustra a figura 4.4, a seguir são detalhados esses componentes:

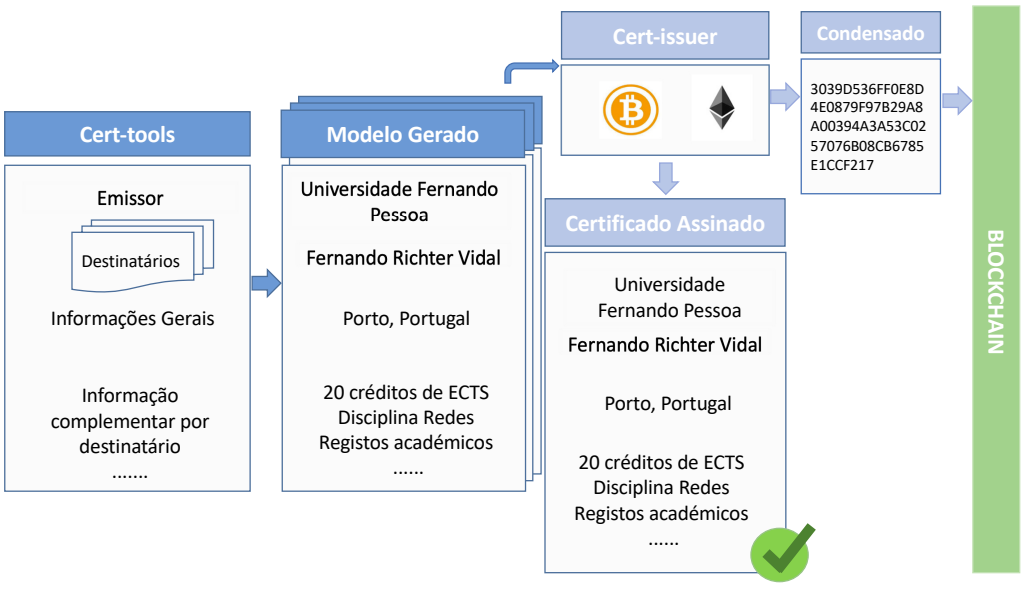

Figura 4.4: *Interação entre componentes do Blockcerts*

4.1.1.1 Componente *cert-tools* para criar modelos de certificados

Os certificados podem ter algumas particularidades, que variam de acordo com o tipo de certificação emitida. Por exemplo, existem diferenças nas informações entre diplomas de conclusão de cursos completos e parciais. O formato de um diploma de conclusão de um curso de graduação, é diferente por exemplo, de um formato de certificado de uma conclusão de créditos de *European Credit Transfer and Accumulation System* (ECTS). Enquanto os cursos de graduação colocam informações resumidas no diploma, a conquista de um crédito de ECTS, necessita informar dados que comprovem a equivalência do tempo dedicado à disciplina frequentada de acordo com o padrão Europeu.

Por essas situações o *Blockcerts* fornece um módulo denominado *cert-tools*, para personalizar o modelo de emissão do certificado, e assim tornar a versão digital do diploma mais fiel possível com a versão em papel. Em cada modelo é possível configurar o título, logótipo, descrição, histórico, criar informações personalizadas de uso comum ou mesmo criar informações personalizadas de uso exclusivo para um destinatário.

A figura 4.4 apresenta um processo de criação de um certificado digital na *blockchain*, usando os componentes do *Blockcerts*. A primeira etapa é definir nos campos oferecidos, quais serão utilizados para criar o modelo. A ferramenta *cert-tools* oferece diversos campos, muitos são opcionais, por isso cabe a cada aplicação decidir quais serão utilizados em cada sistema. Nesse exemplo da figura 4.4, o nome da emissora é fixado no modelo, e os campos dos destinatários, no caso os alunos, estão contidos em uma lista externa. Há um campo de informação global, que serve para replicar uma mesma informação para todos os modelos criados, e por último um campo de informações adicionais, que serve para indicar a cada aluno os créditos que foram obtidos, bem como uma descrição do conteúdo da disciplina, ou seja, informações importantes para auxiliar na

validação do crédito ECTS obtido.

4.1.1.2 Componente *cert-issuer* para publicação de certificados na *blockchain*

O *Blockcerts* foi construído de forma que fosse compatível a operar com qualquer tipo de *blockchain*, como por exemplo, *Bitcoin*, *Ethereum*, *Hyperledger*. Isso permite uma flexibilidade muito grande na hora de construir aplicações, oferecendo algumas opções ao desenvolvedor. Como por exemplo, este pode construir uma solução pública utilizando *Bitcoin*, ou optar por algo mais restrito, criando a solução sob uma rede *Hyperledger* que é do tipo privada.

O componente responsável por dar essa flexibilidade é o *cert-issuer*. Conforme a figura 4.4, ele é uma espécie de conversor que recebe um modelo de certificado ainda não assinado, e o transforma em um certificado digital. Além disso ele calcula o condensado do arquivo digital gerado e faz o seu registo permanente na *blockchain*. Nota-se que na figura 4.4, há logótipos do *Bitcoin* e do *Ethererum*, que representam a possibilidade do *cert-issuer* fazer a conversão para qualquer tipo de *blockchain*.

No padrão mantido pela comunidade de código aberto, estão as redes *Bitcoin* e *Ethererum*. A integração com as demais redes estão surgindo conforme iniciativas, como por exemplo da Universidade do Rosário na Colômbia (Iragorri, 2018), que construiu o conector para *Hyperledger*, e o está testando de forma experimental.

Outro ponto notável da figura 4.4 é a possibilidade de assinar um grupo de modelos de uma única vez. Isso é possível porque dentro de cada modelo há uma lista de destinatários que pode conter *n* alunos. Ao converter, o *cert-tools* entrega vários certificados gerados, mesclando as informações fixas do emissor com as de cada aluno da lista. O *cert-issuer* agrupa esse lote de certificados e os emite em uma única transação na *blockchain*. Isso é interessante por uma questão de custos, já que cada transação para registar na *blockchain* tem uma taxa, o que poderia tornar o projeto economicamente inviável, caso fosse necessário registar individualmente cada diploma.

Nem sempre foi assim, na primeira versão cada certificado era identificado através de uma transação única, o que ocasionava altos custos para a operação. Na segunda versão proposta pelos pesquisadores do MIT, era possível fundir um conjunto de condensados através do uso de uma árvore de *Merkle*. Desta maneira, publica-se a raiz de *Merkle* na rede, ao invés do condensado individual de cada certificado, economizando e muito no custo total das transações (Oliver et al., 2018).

4.1.1.3 Componente *cert-viewer* para visualização e verificação de autenticidade

Este é o componente da *Blockcerts* que tem mais variações de implementações de sua funcionalidade. O seu papel é de gerar graficamente o certificado digital e de oferecer mecanismos para que os interessados consigam verificar de forma independente a autenticidade do documento.

Não existe ainda um padrão definido sobre o formato do certificado digital. Embora isso não seja um problema técnico de segurança, pode causar alguma desconfiança para um avaliador que re-

ceba um mesmo diploma emitido pela mesma emissora, mas em formatos diferentes. Felizmente a função de verificação pode garantir ao interessado a veracidade dos dados, podendo inclusive validar o arquivo em verificadores universais diferentes, como por exemplo, o oferecido pela própria *Blockcerts* [1].

A seguir, duas imagens são apresentadas 4.5 e 4.6, demonstrando essa questão de diferentes visualizações de um certificado.

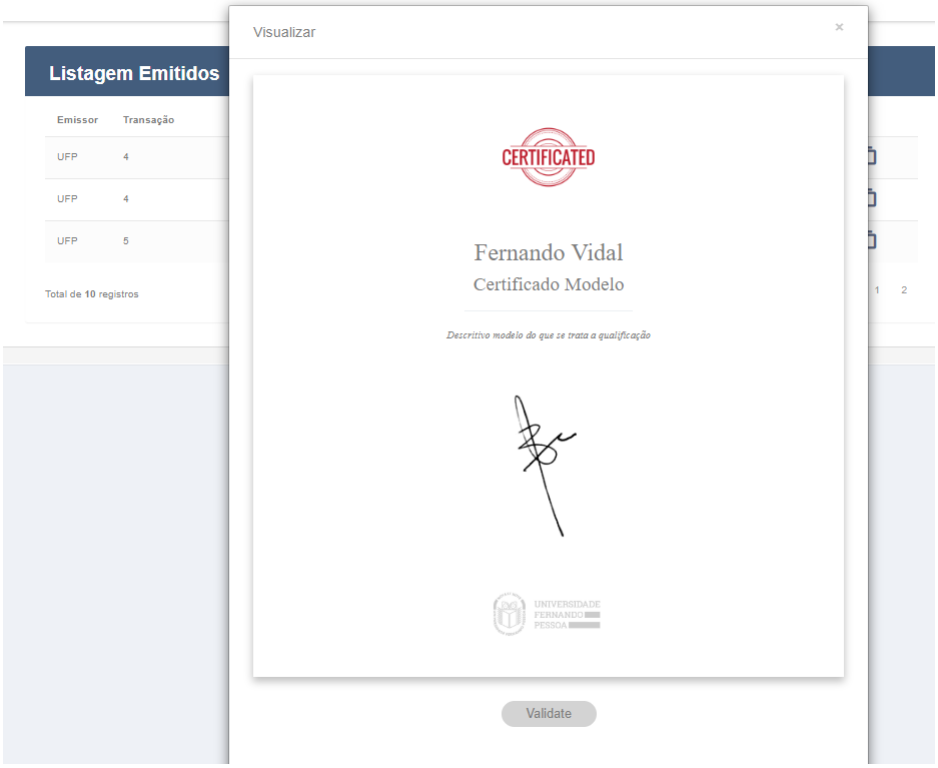

Figura 4.5: *Modelo 1 de visualização de um certificado digital emitido na Blockcerts*

[1] https://www.*Blockcerts*.org/

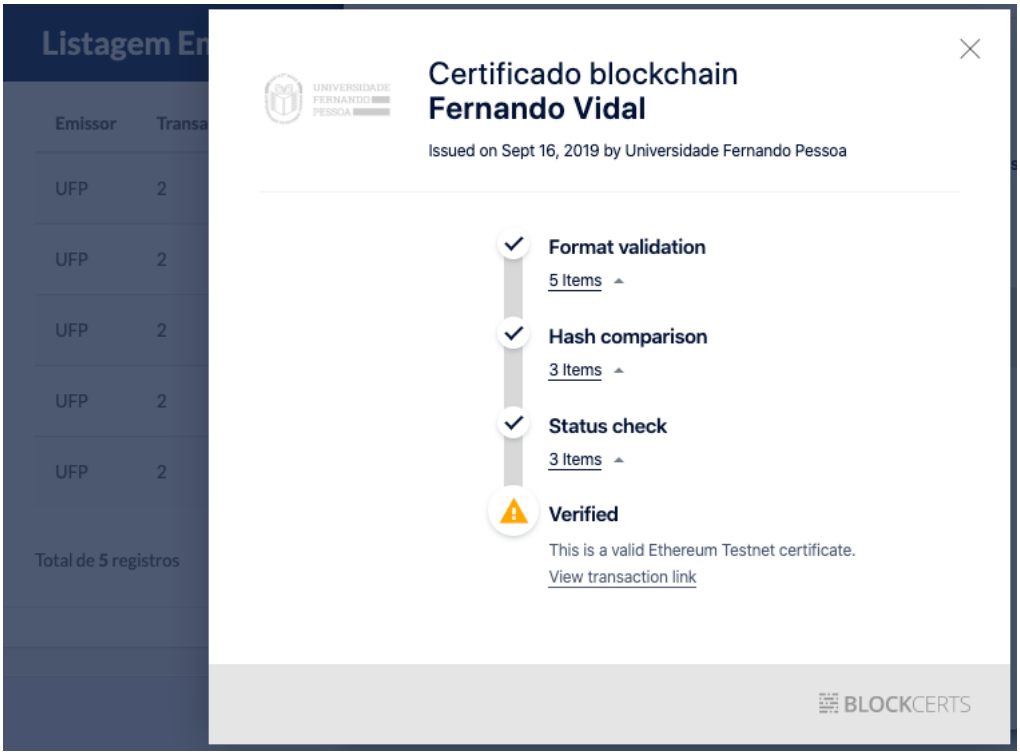

Figura 4.6: *Modelo 2 de visualização de um certificado digital emitido na Blockcerts*

4.1.1.4 Aplicação móvel *cert-wallet*

O *cert-wallet* é um aplicativo da *Blockcerts* disponibilizado também no formato de código aberto, que tem como objetivo oferecer aos usuários autonomia sobre seus registos. Uma vez que o usuário baixa o aplicativo em seu dispositivo pessoal, sua identidade é vinculada a uma frase criada automaticamente pelo *Blockcerts*, que passa a ser uma espécie de chave privada.

O aluno é soberano para cadastrar na sua conta um emissor e resgatar os certificados que foram por ele emitidos. A autonomia também é evidenciada pelo fato do estudante ter a liberdade de enviar seu diploma para quem ele desejar, diretamente sem intervenção de nenhum terceiro. A diferença do processo digital com o papel é que o registo recebido pelo terceiro pode ser verificado diretamente em uma rede pública da *blockchain* (no caso deste trabalho, *Bitcoin* ou *Ethereum*), sem nenhuma consulta a universidade emissora.

O aplicativo tem três funções principais:

1. Gerar e enviar o endereço público do aluno para os sistemas emissores de forma automática
2. Armazenar os certificados digitais
3. Compartilhar os certificados digitais

A figura 4.7, apresenta o funcionamento da primeira função do *cert-wallet*. Nota-se, que o usuário informa um endereço de um arquivo de perfil de emissor, e também um código único e descartável

(em inglês na literatura chama-se *nounce*). Este código serve como uma espécie de credencial do aluno para acessar o serviço que está disponibilizado. Logo após esse procedimento, *cert-wallet* faz acesso a um outro endereço que está embarcado no arquivo de perfil, e envia para tal endereço a chave pública gerada pelo aplicativo e também o código informado.

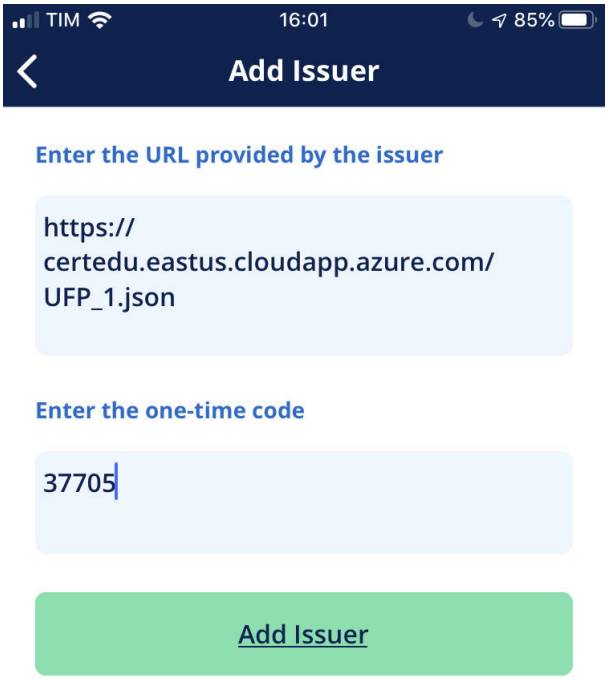

Figura 4.7: *Interface do aplicativo cert-wallet responsável por gerar o endereço público do aluno*

Do outro lado da operação, no segundo endereço referido, está operando um software que está preparado para receber a chave pública e comparar o código recebido, validando ou não a mensagem. Caso positivo, o *cert-wallet* recebe uma sinalização para cadastrar o emissor em seu banco de dados, e a chave pública do aluno é armazenada no aplicativo emissor para que possa ser utilizada em futuras emissões.

A segunda função do aplicativo refere-se ao armazenamento dos certificados, que como nota-se na figura 4.8, pode ser feito de duas maneiras: informando um endereço *Uniform Resource Locator* (URL) na qual o certificado está publicado, por exemplo `https://certedu.eastus.cloudapp.azure.com/certdata/2137705.json`, ou através da importação direta de um arquivo JSON.

Imediatamente após a importação de uma credencial, o aplicativo inicia um processo de validação e verificação da autenticidade do arquivo, informando no final do processo se o certificado é válido ou não.

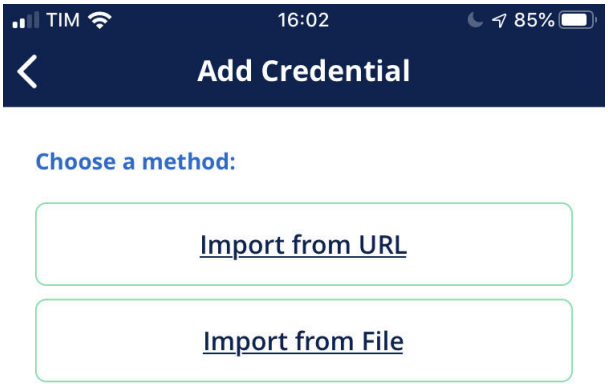

Figura 4.8: *Formas de armazenar um certificado no cert-wallet*

A terceira e última função do *cert-wallet*, refere-se a possibilidade dada aos usuários de compartilharem seus certificados. Ao selecionar um certificado, é possível enviar fisicamente o arquivo para um outro usuário ou simplesmente fornecer o endereço que o referencia.

4.1.2 Visão geral do *BTCert*

BTCert (Rujia and Galind, 2017) é uma iniciativa da Universidade de Birmingham, no UK, que nasceu inspirada no projeto *Blockcerts*, e foi escolhido para ser analisado nesta secção deste trabalho, por também utilizar como base a iniciativa do MIT. Na realidade o projeto foi uma tentativa de gerar um subproduto do *Blockcerts*, melhorando pontos que ainda são considerados fracos da tecnologia. A proposta basicamente ataca os seguintes pontos:

- Problema de segurança da chave.
- Problema de revogação do certificado.
- Problema de identidade da instituição concedente do certificado.

No primeiro quesito, o *BTCert* relata uma preocupação de como a segurança do sistema está centralizada na chave privada. Segundo os autores do *BTCert*, o problema do *Blockcerts* é que caso a chave for roubada ou apagada, seria impossível revogar registos gerados indevidamente. Outra questão por eles colocada, é que a monopolização do direito de emitir certificados ao possuidor da chave privada, pode levar a questões legais e facilitar a corrupção.

Para resolver o problema colocado, o sistema propõe o uso de múltiplas assinaturas, garantindo que ambos os interessados, emissor e recetor, autorizem uma certificação. O protocolo de múltipla assinatura é um conceito conhecido no mundo de criptografia de chave pública, que habilita partes a assinarem digitalmente em conjunto uma mensagem acordada, utilizando chaves privadas próprias (Gilboa, 1999). Tal conceito já é uma prática comum no mundo das criptomoedas, e

M-para-N carteiras de *blockchain* podem ser criadas (Zhou et al., 2016), na qual M representa o número mínimo de assinantes e N o número máximo de endereços públicos.

O algoritmo do *BTCert* que faz uso de múltiplas assinaturas é dividido em três fases: estágio de inicialização, estágio de assinatura, e estágio de confirmação. O estágio de inicialização é responsável por gerar o endereço *Bitcoin* da combinação e também um código chamado de resgate; o estágio de assinatura é responsável por adicionar a assinatura necessária para a execução; o estágio de confirmação da assinatura é executado no minerador e serve para liberar a transação. A figura 4.9, apresenta um exemplo de um processo de múltipla assinatura, em que estão envolvidas três chaves públicas, sendo necessárias duas chaves privadas para assinar a mensagem. O código gerado que resulta na chave pública combinada, nada mais é do que um endereço de *Bitcoin*. É este endereço que será divulgado na rede. Exemplo de um endereço combinado: $3QJmV3qfvL9SuYo34YihAf3sRCW3qSinyC$.

O código de resgate é utilizado desde a assinatura digital, até a liberação da transação pelos mineradores. Exemplo de um código de resgate:
$52410491bba2510912a5bd37da1fb5b1673010e43d2c6d812c514e91bfa9f2eb129e1c183329$
$db55bd868e209aac2fbc02cb33d98fe74bf23f0c235d6126b1d8334f864104865c40293a680$
$cb9c020e7b1e106d8c1916d3cef99aa431a56d253e69256dac09ef122b1a986818a7cb624532$
$f062c1d1f8722084861c5c3291ccffef4ec687441048d2455d2403e08708fc1f556002f1b6cd$
$83f992d085097f9974ab08a28838f07896fbab08f39495e15fa6fad6edbfb1e754e35fa1c7$
$844c41f322a1863d4621353ae$.

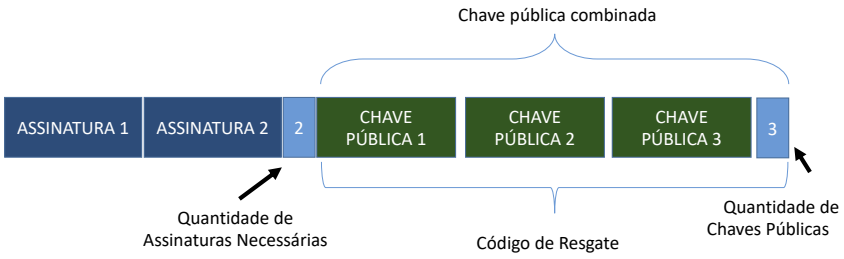

Figura 4.9: *Múltipla assinatura*

A segunda proposta apresentada pela ferramenta, propõe modificar o mecanismo de revogação existente do *Blockcerts*. Na forma atual, a ferramenta do MIT utiliza uma lista hospedada em um endereço URL, que contém os certificados revogados. Essa lista somente pode ser atualizada pela instituição que os emitiu. No entanto, os interessados podem consultar a lista, para verificar se o certificado recebido ainda encontra-se válido. Considerando que a URL é fixa e incorporada nos certificados emitidos, não é razoável dizer que ela é um mecanismo inteiramente fiável.

Para resolver este tipo de problema, o sistema propõe voltar ao mecanismo de emissão da primeira versão do *Blockcerts*, mas apenas para revogação. Revogar um certificado, significa enviar um *Bitcoin* para um endereço de revogação do emissor, que está embarcado no certificado, gerando uma transação para cada registo a ser revogado.

Na prática, como pode-se notar pela figura 4.10, o registo gerado por uma emissão de certificado, sempre deve gerar uma transação do tipo distribuída (maiores informações sobre esse tipo de transação pode ser revisto na secção 3.1.3), que tem sempre uma entrada e duas saídas.

Em uma saída estão os créditos do endereço público da universidade, prontos para serem usados em uma emissão, enquanto na outra saída, um pequeno crédito está disponível para ser transferido para o endereço público da revogação.

O crédito reservado de revogação fica ali parado, pronto para ser usado em caso de cancelamento. Quando um certificado precisa ser revogado, é feita uma operação de transferência para tal endereço. O conjunto formado de registos originários dessas operações, formam uma área de lixeira.

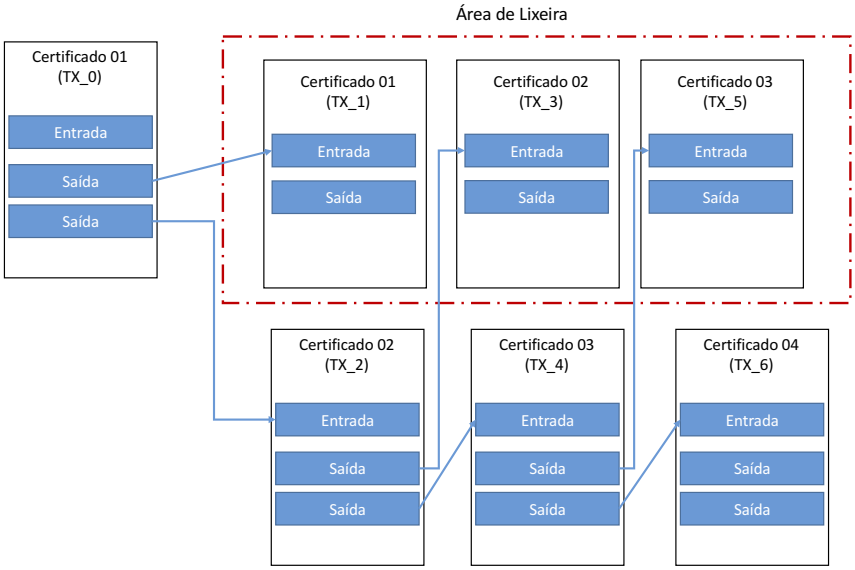

Figura 4.10: *Sistema de revogação do BTCert, adaptado de (Rujia and Galind, 2017)*

O processo de verificação se dá consultando de trás para frente, ou seja, da entrada para a saída. Como visto na secção 3.1.3 as transações têm as suas relações de entrada com a saída anterior. Desta maneira, o *BTCert* propõe identificar um certificado revogado, consultando o endereço embarcado no certificado, verificando se existe movimentação na saída anterior, utilizando o endereço de revogação. Caso positivo, por questões de segurança, uma segunda validação é feita, verificando se este endereço é de propriedade da universidade.

O último ponto a ser analisado da solução, refere-se de como é tratada a questão da identidade do emissor. O *Blockcerts*, determina se um certificado é real ou não, apenas verificando o carimbo de data e hora, analisando a chave do proprietário, para saber se a instituição concedente do certificado, possuía a chave quando o certificado foi emitido. A forma tradicional para provar a autenticidade de um certificado digital, é através da assinatura digital feita com uma chave privada da instituição que o criou. O certificado assinado é divulgado com a chave pública da universidade, permitindo aos interessados utilizá-la para conferir a autenticidade. No entanto, caso a instituição

mude sua chave pública, esse processo pode ficar comprometido.

Para esta situação, o *BTCert* propõe o uso de um algoritmo de identificação federada confiável, que é dividido em duas fases: caminho confiável e identificação federada. O caminho confiável consiste em guardar a rota que conecta todos os recursos necessários, como por exemplo a árvore *Merkle*, a transação, a *blockchain*, e relacioná-la ao endereço de emissão. A identidade federada é uma prova de que o endereço de emissão pertence à uma instituição específica. A identidade federada foi projetada baseada no padrão JSON, e contém todos os metadados que comprovam a propriedade.

Outra semelhança com o projeto do MIT se dá pelo fato da plataforma disponibilizar um aplicativo para que os alunos possam armazenar e verificar seus certificados.

Uma característica marcante desta solução é que ela foi pensada apenas para funcionar em *Bitcoin*, inclusive leva em seu nome as iniciais da criptomoeda. Embora o *BTCert* tenha feito algumas propostas interessantes para resolver os problemas apresentados pelo *Blockcerts*, percebe-se através da página do projeto no GitHub (Rujia and Galind, 2017), que a proposta não ganhou força, e não aderiram ao projeto novos desenvolvedores.

4.1.3 Visão geral do *EduCTX*

O ECTS é uma ferramenta de ensino superior que nasceu em 1989 dentro do programa Erasmus, desenvolvida pela Comissão Europeia e adotada pelos membros da UE. O objetivo deste sistema é contribuir para o planeamento, entrega e avaliação dos cursos oferecidos, bem como facilitar a mobilidade dos estudantes ao reconhecer suas qualificações, experiências e conquistas em diferentes instituições. O *EduCTX* [1], é uma plataforma construída para tratar os ECTS sob o formato digital. Ela foi escolhida para ser analisada nesta secção, por conta de ter uma abordagem diferente, oferecendo recursos para gerir os créditos obtidos nas certificações. Outra questão interessante, é o seu propósito de fomentar o uso compartilhado do sistema pelas universidades, ao invés de oferecer um conjunto de ferramentas e sugerir personalizações, como exemplo é o caso das plataformas apresentadas até então. Qualquer tipo de órgão acreditador e seus membros podem entrar na rede. A figura 4.11 apresenta uma visão geral do sistema.

[1] `https://EduCTX.org/`

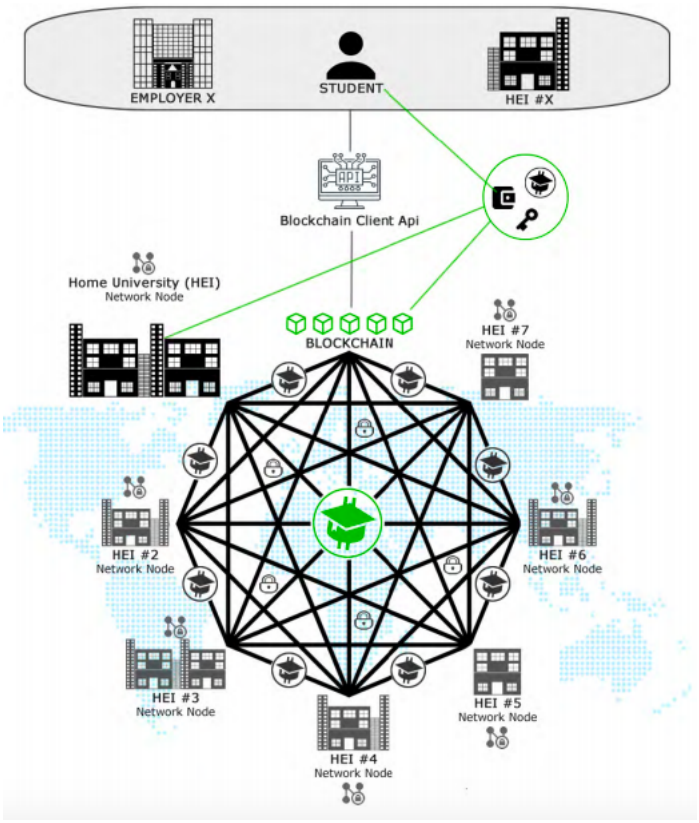

Figura 4.11: *Visão geral da plataforma EduCTX (Turkanović et al., 2018)*

O projeto foi construído utilizando a *blockchain* de código aberto, chamada *Ark* (Ark, 2016). O mecanismo de consenso adotado por esta rede é o DPoS, e condiz muito com a necessidade do projeto *EduCTX*, já que as instituições de ensino superior são nós confiáveis e não há necessidade de minerar transações. A *Ark* também dá suporte para múltiplas assinaturas, mais uma característica que atende a necessidade do projeto *EduCTX*.

Os ECTS representam o volume de aprendizado associado a um estudante, na qual um crédito do sistema corresponde entre 25 a 30 horas de trabalho realizado, incluindo avaliações. O *EduCTX* criou um modelo capaz de gerir eletronicamente tais créditos através do uso da tecnologia *blockchain*. As fichas ECTX são o equivalente aos créditos dos estudantes, que são conquistados ao completar um curso. Neste formato, cada estudante irá guardar em uma carteira *EduCTX* de *blockchain* as fichas coletadas. Na prática, toda vez que um estudante completa um curso, a instituição de ensino superior irá transferir fichas para seu endereço público da *blockchain*.

Basicamente a plataforma armazena as seguintes informações na *blockchain*:

- A entidade de ensino superior é registada com o seu nome oficial na rede.
- O estudante é registado anonimamente.
- O crédito correspondente ao curso.
- O nome do curso realizado.

O aluno utiliza seu próprio endereço público no sistema para receber as fichas ECTX, o que dá a ele a possibilidade de comprovar a conclusão de cursos, sem a dependência de ocorrer uma ação administrativa. Além disso, essa característica impede o surgimento de qualquer tipo de barreira de linguagem.

Assim como o *BTCert*, o *EduCTX*, também faz o uso de um sistema de múltipla assinatura. Por questões de segurança, estudantes têm como requisito o formato 2-2, mantendo dependência junto a suas instituições de ensino, garantindo desta maneira, que alunos não sejam capazes de transferirem créditos para outros endereços.

O uso de múltiplas assinaturas não se limita apenas ao aluno. Por exemplo, ela também é utilizada para autorizar uma entrada de uma instituição de ensino na rede. Tal abordagem beneficia toda a rede no aspeto de segurança ao impedir a entrada arbitrária e aleatória de participantes, garantindo por exemplo, que nós que não representam instituições de ensino, não possam entrar na rede e assim evitar que fossem geradas fichas e notas indesejadas.

A rede *blockchain* utilizada pelo *EduCTX* pode ser considerada do tipo consórcio, na qual cada nova instituição de ensino que entra na rede, precisa passar pela revisão das demais instituições participantes. Este é o motivo do protocolo utilizar o DPoS como mecanismo de consenso, por conta da necessidade das instituições delegarem o voto para confirmarem as transações e selarem os blocos.

No entanto para garantir a democracia da plataforma, garantindo que não haja fins lucrativos, a recompensa da mineração deste protocolo é praticamente levada a zero.

E por ultimo, a *blockchain Ark* escolhida fornece mais de 12 linguagens de programação, o que na visão dos responsáveis pelo projeto, pode encorajar as instituições de ensino, empregadores e estudantes, a entrarem na plataforma, por conta de optarem por uma linguagem de programação familiar a suas escolhas.

4.1.4 Visão geral do *Hyperledger*

Em 2015, muitas companhias despertaram o interesse na *blockchain*, e logo perceberam que alcançariam mais rapidamente seus objetivos trabalhando juntas do que separadamente. Movidas pelo desejo de criar uma ferramenta que fosse capaz de atender as mais variadas necessidades, começaram a moldar um projeto de código aberto denominado *Hyperledger*.

Segundo (Fundation, 2015) "O *Hyperledger* é um esforço colaborativo de código aberto criado para promover as tecnologias de *blockchain* de vários setores. É uma colaboração global, hospedada pela LF, incluindo organizações em finanças, serviços bancários, *Internet of Things* (IoT), cadeias de suprimentos, manufatura e tecnologia".

Uma vez que as organizações têm as suas particularidades, o projeto defende que dificilmente um único padrão de *blockchain* seria capaz de atendê-las, e que seria preciso oferecer um leque de soluções aplicáveis a cada contexto. Na figura 4.12 é apresentado o conceito de estufa, uma representação gráfica de como a solução pode atender a diferentes necessidades, desde a sua ideia

(semente) até a versão final de produção (colheita).

Figura 4.12: *Estufa Hyperledger*

(Fundation, 2018)

Dentre as soluções pesquisadas, este trabalho optou por escolher esta plataforma para ser avaliada, por conta de não ser algo específico criado para educação ou emissão de certificados. A intenção aqui é avaliar até que ponto utilizar uma ferramenta específica pode contribuir ou limitar a construção de uma solução. Assim, das soluções disponibilizadas pelo *Hyperledger*, este trabalho avalia o *Hyperledger Fabric* e a *Hyperledger Indy*, que serão descritas a seguir.

4.1.4.1 *Hyperledger Fabric*

O *Hyperledger Fabric* é uma solução para construção de *blockchain* voltada para qualquer tipo de indústria. A plataforma foi escrita em Golang e dispõe de uma arquitetura modular que permite a implementação de seus componentes (Cachin and Vukolić, 2017).

Diferentemente do *Bitcoin*, este tipo de rede é privada, e por essa razão tem características diferentes, que podem ser interessantes para aplicações na área da educação. Enquanto as *blockchain*s públicas impedem a escolha do modo de operação do mecanismo de consenso, o *Hyperledger* dispõe de uma variedade de componentes de consenso, que podem ser escolhidos diante da necessidade apresentada pela aplicação. Outra configuração interessante, a rede permite oferecer formas de controlar a entrada de membros.

Uma primeira questão, é que ela permite executar aplicações sem a dependência de criptomoedas, podendo o ativo ser representado arbitrariamente. Isso significa que é possível definir o tipo do ativo que será transferido, oferecendo uma melhor representação para desenvolvedores construírem sistemas. Por exemplo, no problema de emissão e verificação de certificados, levando em conta uma implementação em *Hyperledger*, ao invés de representar um certificado através de *Bitcoin*s ou Ether, poderia ser considerado ativo o próprio certificado.

O *Hyperledger*, por ser uma *blockchain* privada, é muito mais flexível. Além de permitir a alternância do mecanismo de consenso, ela habilita a manipulação da função de validação dos nós, fazendo com que um nó ora assuma um papel de validador, ora de replicador. Essas facilidades de ajustar a rede, contribuem muito para o aumento de sua escalabilidade (Fundation, 2018).

Outra característica dessa ferramenta de *blockchain*, é que ela permite criar canais privados entre os participantes. Isso pode ser interessante quando há necessidade de oferecer uma segurança para membros que não queiram ter seus dados expostos para toda a rede, desejando que a informação enviada, apenas seja transmitida para o destinatário interessado. Por exemplo, a troca de informação entre um fornecedor e um cliente, na qual há uma preocupação do fornecedor em expor suas informações para os demais concorrentes.

De forma semelhante ao *Ethereum*, o *Hyperledger* também permite a execução de contratos inteligentes. No entanto, aqui eles são chamados de "chaincode", que são códigos que executam uma regra de negócio. Na versão chamada de "Fabric V1", o protocolo separou a execução dos contratos inteligentes dos demais pedidos de transações, com a intenção de evitar conflitos (Cachin and Vukolić, 2017). Esta abordagem tem muitas vantagens, como melhor escalabilidade, premissas para separar pedidos de validações de transações confiáveis, suporte de contratos inteligentes não determinísticos, particionamento de códigos de contratos inteligentes, dados entre nós (Vukolić, 2017).

Pensando na utilização desta plataforma para desenvolver uma aplicação de emissão e verificação de certificados académicos, algumas questões são colocadas. Por exemplo, não há nenhum modelo pronto para agilizar o desenvolvimento. Seria necessário construir uma arquitetura completa para gerar modelos de certificados, assim como as funções de verificação e validação, e também seria necessário criar todo o projeto de geração e visualização dos certificados.

Na prática, o seu uso seria apenas vantajoso para registar o ativo na rede. Este trabalho cita um exemplo de implementação em *Hyperledger* para emissão e verificação de certificado (Iragorri, 2018), mas mesmo no exemplo citado, não é o *Hyperledger* o verdadeiro responsável por criar todo o sistema. O trabalho trata-se de uma utilização do *Blockcerts* para criar todas as tratativas necessárias para implementar a solução, e apenas é utilizado o *Hyperledger* como banco de dados.

4.1.4.2 *Hyperledger Indy*

O *Hyperledger Indy* é uma ferramenta que faz parte do pacote *Hyperledger*, criado especificamente para tratar da questão da identidade descentralizada. A plataforma fornece ferramentas, bibliotecas e componentes reutilizáveis para criar aplicações de identidades digitais em *blockchain*.

Este trabalho avalia este tipo de rede, por conta da necessidade que a aplicação proposta por este trabalho demanda. É preciso identificar os alunos e a universidade dentro da *blockchain*.

As identidades representadas pelo *Hyperledger Indy*, são interoperáveis, ou seja, funcionam entre domínios e aplicativos distintos. Fazendo uma analogia, seria o mesmo que dizer que concorrentes confiam em uma mesma fonte compartilhada, porque ela é verdadeira. Um dos objetivos do

Hyperledger Indy é responder a questões como: "Com quem estou lidando?", "Como posso verificar se existem dados sobre a outra parte nessa interação?". Caso as respostas para essas questões venham de forma sólida, então pode ocorrer uma interação confiável (Fundation, 2018).

As principais características desta *blockchain* são:

- **Soberania**: Utiliza-se de artefactos para armazenar a identidade em um livro de razão distribuído. Esses artefactos podem ser chaves públicas, provas de existência, criptografia, acumuladores que permitem a revogação e assim por diante. Ninguém além do verdadeiro dono pode alterar ou remover uma identidade.
- **Privacidade nativa**: *Hyperledger Indy* preserva a privacidade, pois todo proprietário de identidade pode operar sem criar nenhuma correlação de risco.
- **Reivindicar verificação**: As reivindicações de identidade podem ser solicitadas com mecanismos menos privativos, como por exemplo, utilizar dados de credenciais familiares, certidões de nascimento, carteiras de motorista, passaportes; ou podem ser mais sofisticados, e através de combinações com provas de zero conhecimento, permitir seletividade e divulgação apenas de dados exigidos por algum contexto específico.

O que percebe-se analisando esta ferramenta, é que ela pode ser utilizada apenas como parte da solução do problema de emissão de certificado. Desta forma, para oferecer uma proposta completa para o problema deste trabalho, seria necessário combinar uma solução de *Hyperledger Fabric* (tratando toda a gestão e emissão do certificado) com *Hyperledger Indy* (tratando toda a questão de identificação requerida).

4.1.5 Soluções não-*blockchain*

Nesta secção são listadas algumas soluções para emissão e verificação de certificados, que não utilizam *blockchain*, e são alternativas puramente digitais. No final da secção são listados alguns diferenciais da solução *blockchain* em comparação a essas soluções.

Diploma digital

No Brasil, o governo iniciou em 2019 o projeto Diploma Digital [1], uma iniciativa voltada para instituições de ensino pertencentes ao sistema federal.

Verification Secure Code

O *Verification Secure Code* [2] é uma ferramenta desenvolvida pelo governo da Espanha, mais precisamente pelo Ministério da Justiça. A segurança do sistema está em gerar um código seguro e anexá-lo no documento oficial. O processo de verificação de autenticidade se dá através de um recurso disponibilizado em sua página oficial. Não há nada de inovador presente no projeto, basicamente se utilizam funções de condensado.

[1] http://portal.mec.gov.br/diplomadigital/

[2] `https://sede.ordenacionjuego.gob.es/en/csv`

Digital ID Certificate

Outra solução digital, também criada pelo governo da Espanha para tratar da segurança de documentos é a *Digital ID Certificate*. Trata-se de um complemento da solução anterior. Aqui, as certificações permitem obter a fonte confiável que originou o documento, validando a assinatura da entidade responsável de seu proprietário, através de um carimbo de data e hora presente em sua criação. O *Digital ID Certificate*, também evita que outras pessoas modifiquem o documento, mas o serviço é todo centralizado no banco de dados do governo.

Qualification Check

O *Qualification Check* [1], é uma organização que atua no mercado de oferta de empregos, e dispõe de um serviço de verificação, vocacionado para universidades. O benefício proposto é evitar a burocracia e ganhar tempo, executando a função de validação de veracidade dos dados, junto das universidades.

Prospects

Prospects [2] é uma empresa do UK, que desenvolve serviços para estudantes que concluem seus estudos no país. Através de uma base de dados centralizada, ajuda companhias a validarem as graduações de estudantes de escolas e universidades. Por conta de ser uma fonte de informação sobre graduações de aluno segura, o *Prospect* tem uma parceria com governo e auxilia os alunos a encontrar trabalho.

***Massive Open Online Courses* (MOOCs)** Plataformas de ensino em massa MOOCs são ferramentas que oferecem um programa de aprendizado, geralmente compostas por vídeos e interações com professores (Ch and Popuri, 2013), e implementam certificados digitais como comprovantes de suas conquistas (edX [3], Cousera [4]). Milhares de alunos obtêm suas certificações e esperam receber um comprovativo das competências adquiridas.

Embora os certificados digitais inovaram em relação ao modelo em papel, analisando os diversos pontos apresentados pela tabela 4.1, nota-se que tais soluções (não-*blockchain*), não conseguem atingir os mesmos benefícios oferecidos por soluções *blockchain*.

[1] `https://www.qualificationcheck.com/`
[2] `https://www.prospects.ac.uk/`
[3] https://www.edx.org
[4] https://www.coursera.org/

Tabela 4.1: *Comparativo entre certificados digitais x soluções blockchain*

Propriedades	**Certificado digital**	**Certificado *blockchain***
Confiabilidade	Segurança depende unicamente da assinatura digital	Combinação de diversos mecanismos criptográficos
Privacidade	Sensível ao vazamento de dados, toda informação está contida no certificado	Somente o condensado é publicado
Autonomia	Assinaturas digitais necessitam de órgãos centrais reguladores	A tecnologia dispensa o uso de terceiros
Padronização	Em muitos países não há autoridades certificadoras que padronizam o uso das assinaturas digitais	Não depende de nenhum órgão regulador
Perda de dados	Fácil de serem eletronicamente destruídos e dependem de mecanismos de *backup*	Os dados são naturalmente distribuídos, o que torna a perda de informação praticamente impossível
Prova de existência	Datas geradas dependem da idoneidade do assinante	Datas dos eventos refletem precisão

4.1.6 Comparativo entre plataformas

A avaliação dessas soluções de emissão de certificado foi orientada pelos seguintes requisitos:

- Funcionar em qualquer *blockchain*.
- Utilizar uma tecnologia confiável.
- Oferecer privacidade.
- Utilizar rede pública e descentralizada.
- Ter licenciamento sob o formato de código aberto.
- Ser eficiente na questão de revogação de certificados.
- Ser eficiente em questões de custo por emissão.

A tabela 4.2 apresenta uma comparação entre algumas das propriedades que eram desejadas de serem encontradas nas soluções.

Tabela 4.2: *Comparativo entre as ferramentas analisadas*

Plataformas	Compatibilidade de *Blockchain*	Identificação Auto-suficiênte	Comunidade Participativa	Rede Pública	Tratativas de Privacidade
BTCert	não	sim	não	sim	sim
Hyperledger	não	sim [1]	sim	não	sim
EduCTX	não	sim	não	não	sim
Blockcerts	sim	sim	sim	sim	sim

Nas soluções avaliadas, foi escolhida para a implementação do protótipo deste trabalho, a ferramenta *Blockcerts*. O motivo se deu pelo fato de melhor atender aos requisitos colocados, principalmente no que diz respeito a funcionar em qualquer *blockchain*. Na realidade, o *Blockcerts* foi a única solução de emissão de certificados que nasceu com o requisito de funcionar com qualquer *blockchain*. Esse requisito aumenta e muito a complexidade da solução. Por exemplo, como assegurar que o processo de revogação irá funcionar para qualquer rede? ou como garantir baixo custo, considerando o uso em qualquer *blockchain*?

Não fazer aposta em uma única tecnologia de *blockchain* é importante, porque mantém o ciclo de vida da aplicação longo. Na hipótese do surgimento de uma nova *blockchain*, e na hipotética situação em que o *Bitcoin* e *Ethereum* ficarem obsoletos e não serem mais usados, o *Blockcerts* não seria afetado por este movimento, e continuaria por ser uma solução válida para a emissão dos certificados.

Outro fato identificado é que o *Blockcerts* permite a emissão de certificados em lote (o que se traduz em baixo custo). Além disso, há muitos desenvolvedores melhorando o código, o que significa que o projeto recebe constantes atualizações. Já as soluções *BTCert*s e *EduCTX*, percebe-se observando os dados de atualizações disponibilizados no repositório público do *Github*, que os projetos não recebem um bom volume de atualizações, e a comunidade não tem levantado questões técnicas sobre a tecnologia. Nesse aspeto de evolução, o *Hyperledger* se iguala ao *Blockcerts*, pois apresenta um ótimo volume de atualizações.

Algumas abordagens do *BTCert*s são interessantes, como por exemplo, no que se refere a revogação. O modelo proposto pode ser facilmente adaptado para qualquer tipo de *blockchain*. No entanto, é preciso discutir melhor a solução proposta, por conta do alto custo para revogar um lote de certificados, já que seria necessário dispensar de forma individual créditos para cada certificado contido no lote. Também o modelo não define de forma clara onde ficariam registadas as informações do histórico da revogação, como por exemplo, motivo do cancelamento, e datas, já que o campo *OP_RETURN_DATA* tem um tamanho limitado de 83 bytes (Bartoletti and Pompianu, 2017) .

[1] Precisa combinar *Hyperledger Fabric* + *Hyperledger Indy*

O *EduCTX* propõe em suas operações, usar um sistema de múltiplas assinaturas. Pensando na *blockchain Ark* em que o projeto foi desenhado, pode ser uma boa prática o uso desse recurso, mas diante do desejo da solução ser compatível com outras *blockchain*s, o custo do recurso pode ser um impeditivo. Imaginando a operação de múltiplas assinaturas operando por exemplo em *Bitcoin*, haveria um custo do lado do aluno, para registar parte da sua operação.

As fichas representadas por créditos digitais sinalizam que possam surgir necessidades, na área da educação, de controlar campos mais estruturados. Imaginando uma implementação do *Blockcerts* para tratar os créditos de ECTS, percebe-se que seria mais complicado de resgatar as informações, utilizando meramente os campos informativos contidos em cada certificado emitido. Por exemplo, como calcular o saldo restante de créditos de um aluno para atender um determinado curso? Como evitar que sejam transferidos em duplicidade créditos para um mesmo aluno?

O mecanismo de consenso proposto pelo *EduCTX*, o torna ainda interessante para a área da educação. Não faz sentido minerar blocos para registar transações que contém certificados académicos ou créditos de ECTS. As universidades são nós confiáveis e responsáveis pelos dados que fornecem.

O *Blockcerts* usa uma rede de *blockchain* pública, o que torna o sistema interessante por conta da distribuição. Essa característica permitiria por exemplo, mesmo que um diploma fosse emitido em uma *blockchain* obsoleta, através de uma cópia local, continuar a verificar e validar certificados antigos. No entanto, nas soluções do tipo *Hyperledger*, por conta de utilizarem rede privada, isso não seria possível.

Para a aplicação de emissão de certificados que este trabalho avalia, o *Hyperledger* tem como ponto forte a flexibilidade, e como ponto fraco a falta de modelos capazes de agilizar o desenvolvimento. Seria interessante se já existissem bibliotecas pré-montadas de código aberto, para esse fim.

Outra questão envolvendo o *Hyperledger* é que para conseguir identificar a universidade ou o aluno, seria necessário combinar o uso de duas tecnologias da estufa (*Fabric* + *Indy*), o que pode tornar o desenvolvimento da aplicação ainda mais complexo.

4.2 Conclusão

Este capítulo apresentou algumas das plataformas que podem ser utilizadas para construção de soluções *blockchain*, para gestão de certificados académicos. De maneira a demonstrar a diversidade de opções disponíveis, foram apresentadas quatro soluções. O *Blockcerts*, posicionado como pioneiro na gestão de certificados *blockchain*, seguido pelo *BTCert*s, um subproduto que surge na promessa de resolver algumas fraquezas identificadas no *Blockcerts*. No capítulo também é apresentado o *EduCTX*, uma vertente diferente para digitalizar os créditos ECTS, e que implementa uma rede do tipo consórcio, tendo seu foco no uso de múltiplas assinaturas. Por último é retratado o *Hyperledger*, uma plataforma de desenvolvimento de aplicações *blockchain* mais genérica, que de forma mais minuciosa, permitiria a construção de um sistema de emissão e verificação de

certificados.

Capítulo 5

Implementação

5.1 Arquitetura proposta

Com a intenção de também deixar a sua contribuição nesta tecnologia emergente, este trabalho pretende construir um protótipo e testar a emissão e verificação de certificados académicos recorrendo a um *blockchain*. A aplicação chamada de CertEdu foi construída com base na plataforma *Blockcerts*, analisada na secção 4.1.1, e emite documentos eletrónicos nas redes *Bitcoin* e *Ethereum*.

Não foi considerada a possibilidade de desenvolver uma nova plataforma que fosse capaz de criar os certificados digitais na *blockchain*. Assim, optou-se por explorar uma tecnologia existente. O *Blockcerts* tem avançado em algumas questões, e dispõe de um bom número de desenvolvedores focados neste problema. Observando as outras soluções, percebe-se que embora ofereçam alguma inovação em relação a solução do MIT, suas propostas não atraíram o interesse da comunidade.

O objetivo é entender melhor os desafios e benefícios envolvidos no processo de digitalização dos diplomas, através de uma implementação de um protótipo. Para isso foi necessário configurar um ambiente na nuvem e desenvolver uma aplicação capaz de interagir com o *Blockcerts*, e explorar todos os seus recursos. A arquitetura proposta para este trabalho é retratada pela figura 5.1.

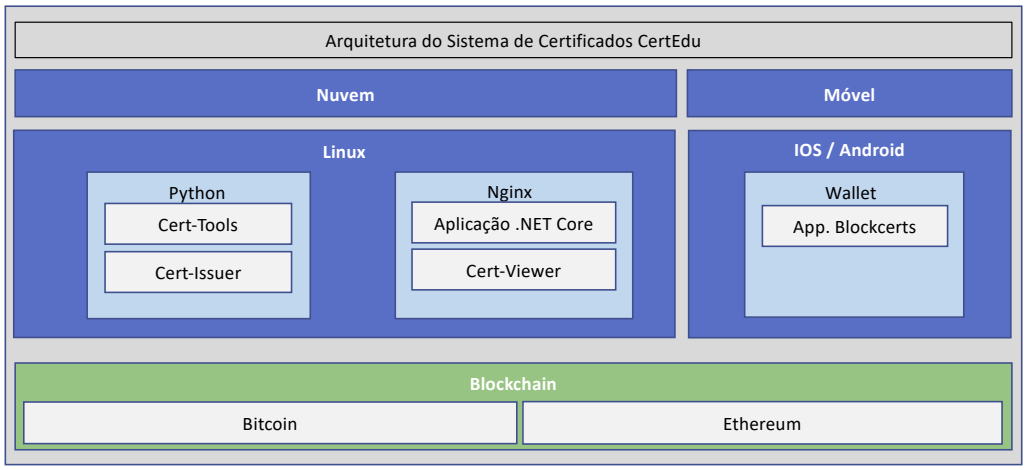

Figura 5.1: *Arquitetura do protótipo*

Ainda em sua juventude, o *Blockcerts* oferece integração com aplicativos através de linha de co-

mando. Parte dos componentes até disponibilizam uma interface mais amigável de integração, como por exemplo a *Application Programming Interface* (API) do *cert-issuer*, mas isso ainda é insuficiente, porque outros componentes, como por exemplo, o gerador de modelos *cert-tools*, ainda requerem execução na linha de comando. Por conta deste fato, é necessário que no mesmo ambiente em que estão disponibilizados os aplicativos *cert-tools* e *cert-issuer*, também esteja o serviço de *Hypertext Transfer Protocol* (HTTP) responsável por hospedar a aplicação. Dessa forma a aplicação consegue localmente executar os comandos em *bash* e devolver ao navegador uma página *Hypertext Markup Language* (HTML) com os resultados do processamento dos componentes *Blockcerts*.

Uma alternativa a esta limitação, seria utilizar os códigos em *Python* e modificá-los para que as chamadas fossem feitas diretamente pela aplicação, mas como trata-se de uma tecnologia nova e está em constante processo de atualização, optou-se por usá-la em sua forma nativa. Dessa maneira, quando houver uma atualização, basta atualizar os componentes e as chamadas por linha de comando continuarão a funcionar. Também foi decidido em montar o ambiente de forma pura, não utilizando contentores (por exemplo *docker*) ou ambientes virtuais prontos. A intenção dessa ação é avaliar como a ferramenta se comporta, diante de atualizações de versões de *Python*, ou pacotes do sistema operacional. A aplicação oferece três tipos de perfis:

- **Alunos**: Além de terem a opção de consultar seus certificados em seus dispositivos móveis, haverá um portal em que eles terão acesso para visualizarem seus documentos. No portal o aluno pode descarregar o seu certificado digital e guardá-lo. Mesmo que a universidade ou o portal encerre suas atividades, o certificado ainda continua existindo e poderá ser conferido por qualquer outra pessoa. No portal o aluno poderá fazer solicitações ao setor de registo da universidade, solicitando por exemplo uma revogação de um diploma que foi emitido com informações incorretas.

- **Terceiros**: (Empregadores, outras universidades) – Os demais interessados, como por exemplo, os empregadores, terão a liberdade de conferir a veracidade de um certificado emitido pela certificadora, através de um acesso ao portal. Nota-se que o portal é apenas um facilitador que centraliza os diplomas emitidos. Mesmo que o portal ou a universidade encerre suas atividades, o interessado pode ainda conferir um certificado emitido, através de outros portais que tenham um verificador universal.

- **Administração da certificadora**: De forma unidirecional, apenas o setor de registos deve ser capaz de criar os modelos dos certificados e publicá-los na *blockchain*. A construção desta aplicação tem seguido o princípio da *Blockcerts* de ter uma usabilidade muito simples. O objetivo é retirar passos intermediários desnecessários para que o usuário atinja o seu fim de forma rápida. Por exemplo, o *Blockcerts* precisa que seja informado uma lista em formato *Comma-Separated Values* (CSV) com os destinatários que receberão o certificado. Nessa lista há informações como: chave pública, nome do aluno e endereço de correio. Para evitar que o administrador precise gerar o arquivo, carregá-lo, consistir os dados, a aplicação gera o arquivo em tempo de execução, tornando transparente o processo ao usuário. Há também um cuidado da maneira como a aplicação integra os processamentos dos

componentes da *Blockcerts*, *cert-tools* e *cert-issuer*, dando a impressão ao usuário de estar executando uma única ação.

5.1.1 Desafios da solução

Esta secção enumera os principais pontos a serem tratados por uma aplicação *blockchain* de gestão de certificados. Desta forma, o trabalho avalia tais questões mediante a CertEdu, e evidencia quais questões são atendidas e quais ainda precisam ser trabalhadas. Em relação aos pontos ainda não solucionados, são apresentadas algumas das possíveis soluções para esses problemas.

Tabela 5.1: *Processos e desafios relacionados com a solução blockchain*

Processo(s)	**Desafio**	**Atendido**
Emissão	Financeiramente viável a emissão em grandes volumes	sim
Emissão	Identificação de universidades	sim
Emissão	Permitir revogação em caso de emissão equivocada	parcial
Verificação	Verificação em base local (descentralizada)	parcial
Verificação	Verificação em verificadores universais genuínos	sim
Verificação	Estar disponível permanentemente	sim
Partilha	Identificação de alunos	sim
Partilha	Autonomia para distribuir os diplomas	sim
Partilha	Visualização gráfica das informações registadas	sim
Todos	Compatibilidade de operação em qualquer *blockchain*	sim

A implementação do aplicativo CertEdu, evidencia as informações da tabela 5.1, na qual percebe-se uma boa aderência da tecnologia *blockchain*, no uso de aplicações para emissão e verificação de certificados. No entanto, nota-se que os pontos ainda remanescentes, são importantes e encontrar uma solução para eles, sem alterar os demais pontos já solucionados, é uma tarefa complexa. A seguir, os tópicos parcialmente atendidos são discutidos:

5.1.1.1 Validação em base local (verificação descentralizada e distribuída)

Oferecer os registos académicos distribuídos e descentralizados, é um dos principais benefícios que podem ser oferecidos por uma solução que implementa *blockchain* para emissão e verificação de certificados. Comparando com as demais soluções digitais, sem o suporte de uma estrutura como a *blockchain*, é praticamente impossível garantir tais propriedades.

Por exemplo, o *Open Badge* embora consiga oferecer uma autonomia para que os interessados validem o certificado hospedado, tal contexto do diploma é apenas relevante ao endereço em que o arquivo está armazenado. Plataformas *on-line*, como por exemplo as MOOCs, falham no sentido de toda estrutura estar centralizada.

Através do protótipo desenvolvido, percebe-se que ao utilizar uma rede *blockchain* pública, os registos permanentemente gravados poderiam ser consultados de forma perpétua. Um usuário poderia por exemplo, baixar uma cópia da rede localmente, e mesmo desligado da rede continuar a verificar a veracidade de certificados, que ali estejam contidos.

No entanto, também percebe-se através da implementação, que há uma dependência de um arquivo hospedado no servidor do emissor, para verificar se o certificado está revogado, o que torna o processo parcialmente atendido.

A solução para este problema, é alterar o processo de validação de revogação, removendo a dependência do ponto centralizador, fazendo com que a informação do certificado revogado esteja armazenada dentro da própria *blockchain*, permitindo assim, que uma cópia completa local, permita em sua completude, a verificação de um certificado.

LM e *NextID* no final de 2019, publicaram um artigo (Ronning and Chung, 2019) com uma proposta para implementação da versão 3.0 do *Blockcerts*, na qual sugerem o uso dos padrões VC [1] e DID [2]. Uma das vantagens apresentadas nesta proposta de versão, é justamente substituir o perfil do emissor por tais tecnologias citadas. Combinando VC e DID, não seria mais necessário um arquivo hospedado para identificar o emissor de um certificado emitido. Até o presente momento deste trabalho, não há ainda nenhuma versão beta que permita testar a funcionalidade.

5.1.1.2 Limitações encontradas na revogação

Como citado em 5.1.1.1, o protótipo constatou um ponto de dependência da solução, no que refere-se a revogação. A figura 5.2 apresenta uma visão de como é o processo de revogação oferecido pelas bibliotecas do *Blockcerts*.

[1] https://w3c.github.io/vc-data-model/
[2] https://w3c.github.io/did-core/

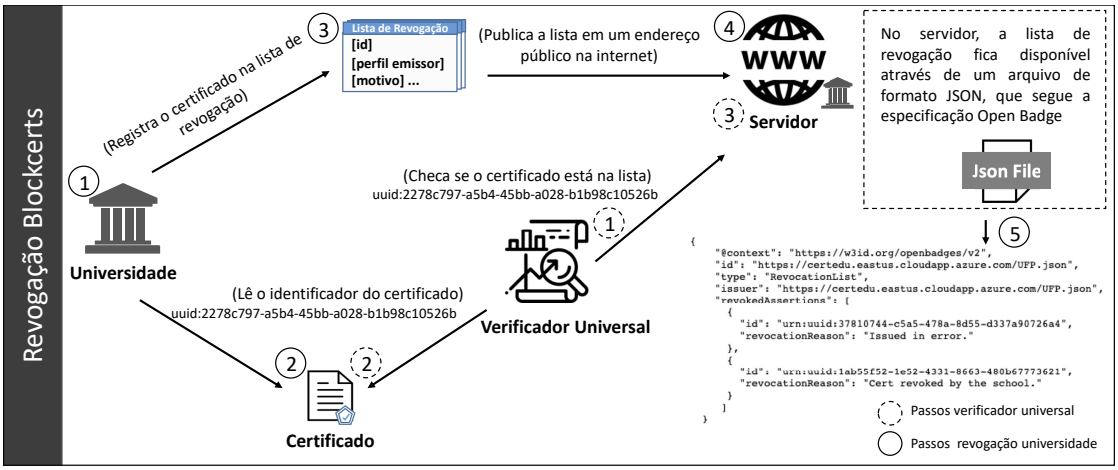

Figura 5.2: *Processo de revogação da Blockcerts (Vidal et al., 2020)*

Esse modelo baseado em *Certificate Revocation List* (CRL), apresenta alguns problemas como por exemplo:

- Caso o servidor do emissor fique inacessível um certificado não poderia ser identificado como válido ou revogado.
- A segurança da informação de revogação depende da segurança implementada nos mecanismos do servidor em que está hospedado.
- O arquivo é facilmente fraudável e inseguro, como nota-se na figura 5.3.
- Há riscos do arquivo ser corrompido, afetando a verificação de vários certificados.

certedu.eastus.cloudapp.azure.com/rev_UFP.json

```
{
    "@context": "https://w3id.org/openbadges/v2",
    "id": "https://certedu.eastus.cloudapp.azure.com/UFP.json",
    "type": "RevocationList",
    "issuer": "https://certedu.eastus.cloudapp.azure.com/UFP.json",
    "revokedAssertions": [
      {
        "id": "urn:uuid:37810744-c5a5-478a-8d55-d337a90726a4",
        "revocationReason": "Issued in error."
      },
      {
        "id": "urn:uuid:1ab55f52-1e52-4331-8663-480b67773621",
        "revocationReason": "Cert revoked by the school."
      }
    ]
}
```

Figura 5.3: *Exemplo de arquivo de revogação de certificados*

Como observa (Vidal et al., 2019), uma solução consistente para emissão e verificação de diplomas, precisa ser agnóstica a *blockchain*. A tecnologia *blockchain* é nova, sendo provável que muitas redes venham a desaparecer e muitas outras a surgir no futuro.

Uma possível solução para este problema seria gravar a informação de revogação dentro da própria *blockchain*, através de uma informação reduzida que seria passível de ser verificada. Um exemplo disso é o mecanismo implementado pela solução *BTCerts* 4.1.2, na qual utiliza-se operações de *Unspent Transaction Output* (UTXO) para gerar tais informações.

Outra solução possível para esse problema seria o uso de contratos inteligentes. Por exemplo, o trabalho de (Santos and Duffy, 2019), implementa um modelo de revogação para *Ethereum* e para *Blockcerts*. A solução é interessante, mas diante do objetivo de funcionar em qualquer tipo de *blockchain*, ela é limitada. Por exemplo, embora existam algumas tentativas para funcionar contratos inteligentes no *Bitcoin*, tal criptomoeda foi desenvolvida para manter transações e não para executar códigos.

A primeira opção apresentada oferece uma maior compatibilidade, mas precisa ser trabalhada por ter uma limitação de armazenamento de dados. O processo de revogação, além de informar que um certificado está revogado, precisa oferecer a possibilidade de visualizar informações complementares, como por exemplo: motivo da revogação e data da revogação. Analisando o *Bitcoin*, de forma singular os 83 *bytes* disponíveis são insuficientes (Bartoletti and Pompianu, 2017), e combinar um arranjo de registos para obter um espaço maior, pode tornar o processo extremamente dispendioso.

Por essa razão, o modelo proposto (Vidal et al., 2020), é de implementar um arquivo JSON, contendo as informações complementares, para que assim possam ser verificadas pelos interessados. O funcionamento é semelhante ao padrão de criação de certificado, no entanto serviria como uma espécie de informação mais detalhada acerca de uma revogação.

Esse mecanismo de revogação, permite aos interessados verificar aquilo que é mais importante, se um certificado está ou não cancelado. Na pior das hipóteses, como a perda do arquivo, não seria possível identificar as razões e a data em que foi feito o cancelamento, no entanto o sistema continuaria a funcionar, fornecendo aos interessados o estado correto do certificado. Na prática, há pouca a necessidade de armazenar o arquivo complementar, mas em situações excecionais, é dada essa possibilidade. Para complementar a solução, pode-se utilizar um mecanismo *InterPlanetary File System* (IPFS) para distribuir o arquivo complementar e assim evitar que haja centralização.

Basicamente o IPFS é um sistema de arquivos distribuídos, na qual pode-se armazenar um arquivo em um local universal. Essa característica provê naturalmente um *backup*, já que os arquivos são replicados aos pares. Além disso, é um sistema muito eficiente, uma vez que pode-se acessar arquivos em endereços vizinhos, normalmente mais próximos que os modelos centrais. O IPFS não possui um único ponto de falha, e os nós não precisam confiar mutuamente uns nos outros. Os dados armazenados são distribuídos e mantidos em diferentes lugares do mundo (Wang et al., 2018).

Há algumas semelhanças entre o funcionamento do IPFS com a *blockchain*. Além da topologia de

rede P2P, o IPFS também referencia seus registos utilizando funções unidirecionais. Isso significa, que ao publicar um arquivo na rede, é calculado o condensado do seu conteúdo, evitando assim que haja arquivos duplicados no ambiente. Por exemplo, os trabalhos (Kumar and Tripathi, 2019), (Wang et al., 2018), (Rajalakshmi et al., 2018), implementam soluções de distribuição de arquivos, combinando IPFS e *blockchain*.

De forma simplificada, o desenho deste modelo de revogação é representado pela figura 5.4.

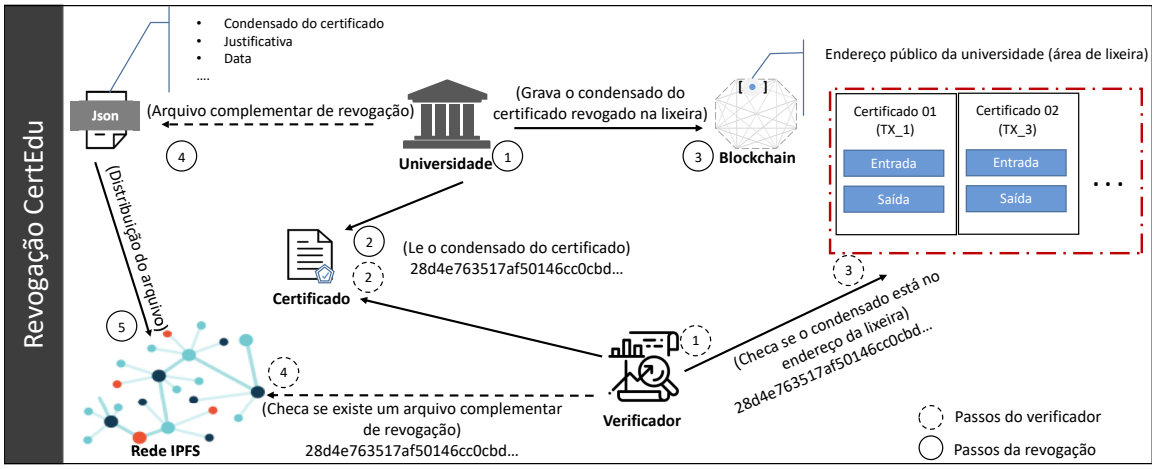

Figura 5.4: *Mecanismo proposto para tratar a revogação (Vidal et al., 2020)*

O modelo usa a ideia da primeira versão do *Blockcerts*, na qual a informação de revogação é armazenada dentro da *blockchain*, ao invés de usar um mecanismo externo. Contudo, no modelo proposto são adicionadas algumas melhorias que foram identificadas na abordagem do *BTCerts*. O mecanismo do *BTCerts* preocupa-se com o endereço que transfere dinheiro para o endereço revogado, enquanto a primeira versão do *Blockcerts*, se concentra em saber se o dinheiro do endereço revogado é gasto (Vidal et al., 2020).

Propõe-se validar o estado de entrada, ao invés do estado de saída, criando uma área de lixeira. Como observa (Rujia and Galind, 2017), em termos de consumo, a primeira abordagem do *Blockcerts*, é mais custosa, porque cada revogação, precisa de pelo menos dois pagamentos, enquanto esta abordagem, somente necessita despender uma única transação para cada revogação.

Além disso, o modelo propõe ligar o condensado do certificado, com o arquivo complementar em formato JSON, e transmiti-lo a um recurso distribuído, como IPFS. Como é gerado com SHA256 o tamanho do condensado do certificado é de trinta e dois *bytes*.

A tabela 5.2, compara as diferentes abordagens dos mecanismos de revogação. Este modelo propõe armazenar a informação da revogação em um arquivo distribuído, tornando assim a solução compatível com qualquer tipo de *blockchain*, e resguarda o mecanismo de enfrentar problemas com a limitação de espaço.

Tabela 5.2: *Mecanismos de revogação*

Ferramentas	**Característica**	**Arquivo distribuído**	**Qualquer *blockchain***
Blockcerts	CRL	Não	Sim
BTCerts	UTXO	Não	Não
EduCTX	Multi-assinatura	Não	Não
Outros [1]	Contratos inteligentes	Não	Não
CertEdu	Ligação por condensado	Sim	Sim

A revogação de certificados na *blockchain* tem sido abordada por diversas formas, como demonstrado por (Vidal et al., 2020). Embora o MIT tenha atacado o problema em duas versões do *Blockcerts*, como pode-se notar em seu plano [2], ainda busca uma melhor alternativa para a solução. A segunda versão do *Blockcerts*, que utiliza CRL, trouxe uma grande compatibilidade e redução de custos, no entanto gerou uma dependência do emissor que compromete os benefícios da distribuição e autonomia da *blockchain*.

No trabalho publicado pela LM, na qual está descrita a proposta da terceira versão do *Blockcerts*, percebe-se alguns benefícios, como a descentralização do perfil do emissor. No entanto, em sua proposta, a lista de revogação ainda é mantida como forma de verificação de revogação (Ronning and Chung, 2019).

Uma boa solução deve permitir a compatibilidade entre *blockchain*s e preservar os benefícios que a tecnologia oferece, como autonomia e distribuição. À medida que a tecnologia se expande e novas redes surgem, a melhor solução para o problema de revogação é usar um padrão que funcione entre as diversas redes (Vidal et al., 2020).

O modelo proposto fornece disponibilidade da informação de revogação, além de ser independente de qualquer ação de terceiros. Como propõe armazenar o condensado do arquivo na área de lixeira, a privacidade permanece segura.

A adoção de um arquivo adicional resolve o problema de limitação de espaço encontrado na arquitetura *blockchain* e, ao contrário da solução atual do MIT, na qual a perda do arquivo de lista compromete todo o processo de revogação, aqui apenas parte das informações é afetada, sendo que, a principal informação, que garante aos interessados saber se o certificado está ativo ou revogado, está preservada.

5.2 Funcionalidades do sistema

Esta secção do capítulo apresenta a implementação do protótipo utilizado para avaliar uma aplicação da tecnologia *blockchain* na área da educação. Alguns requisitos foram identificados e

[1] (Palma et al., 2019)(Santos and Duffy, 2019)
[2] https://www.*Blockcerts*.org/guide/roadmap.html

implementados com intuito de fornecer uma melhor visão sobre o uso da tecnologia.

Tabela 5.3: *Requisitos e objetivos a serem avaliados*

Tipo	**Requisito**	**Objetivo**
Funcional	Aplicação com três perfis: Administradores, Alunos e Empregadores	Autonomia
Funcional	Emissão de certificados deve permitir várias *blockchain*s	Compatibilidade
Funcional	Permitir a criação de grupos e gerenciar emissões em lotes	Viabilidade financeira
Funcional	Gerenciar várias universidades	Padronização
Funcional	Editor de modelos de certificados, seguindo a especificação do *Blockcerts*	Usabilidade
Funcional	Implementar graficamente todos os parâmetros do *Blockcerts*	Avaliar flexibilidade
Funcional	Embarcar verificador de certificados	Segurança
Funcional	Utilitário para gerar endereços e créditos fictícios (*Bitcoin* e *Ethereum*)	Agilizar simulações
Não-funcional	Sistema deve operar na nuvem	Escalabilidade
Não-funcional	Integrar com aplicativo móvel *Blockcerts*	Identificação
Não-funcional	Transparente ao usuário as conexões entre os componentes do *Blockcerts*	Integração

5.3 Configuração do ambiente

Conforme apresenta a figura 5.5, para este trabalho foi utilizada a seguinte configuração de infraestrutura:

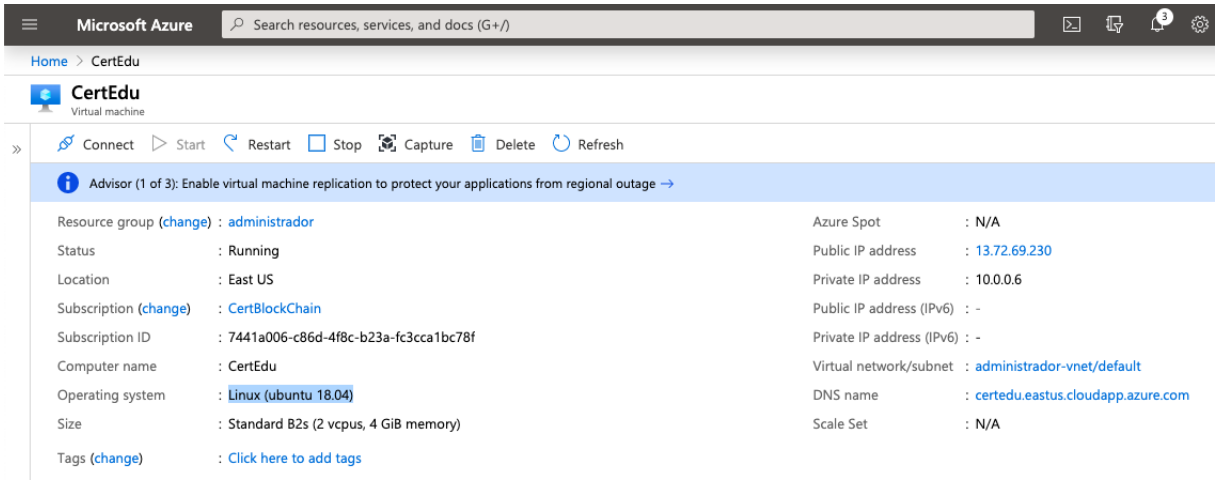

Figura 5.5: *Infraestrutura da aplicação CertEdu*

A preparação do ambiente inicia-se com a configuração de duas redes de *blockchain*: *Bitcoin* e

Ethereum. O intuito da aplicação é simular um ambiente próximo a uma situação real, por essa razão, optou-se em configurar as redes *TestNet* (*Bitcoin*) e *Rospten* (*Ethereum*). Portais de *faucets* [1], [2], [3], [4] foram utilizados para transferir créditos para as contas da aplicação, e assim emular um cenário próximo a realidade, em que há recursos financeiros limitados para gerar certificados. Maiores informações sobre essas redes podem ser revistas na secção 3.1.6.

Conforme nota-se nas figuras 5.6 e 5.7, através de *scripts*, o ambiente configurado na nuvem realiza constantemente a sincronização entre o banco de dados local e o distribuído. Isso significa que quando um certificado é emitido, é necessário aguardar o processo de sincronização, para que o registo publicado na *blockchain* pública seja replicado para o ambiente local.

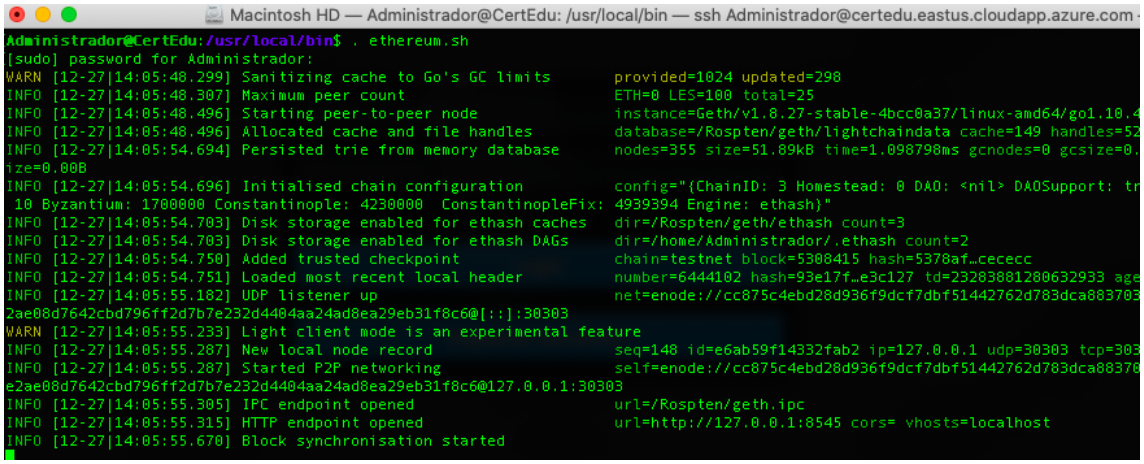

Figura 5.6: *Processo de sincronização rede Ethereum*

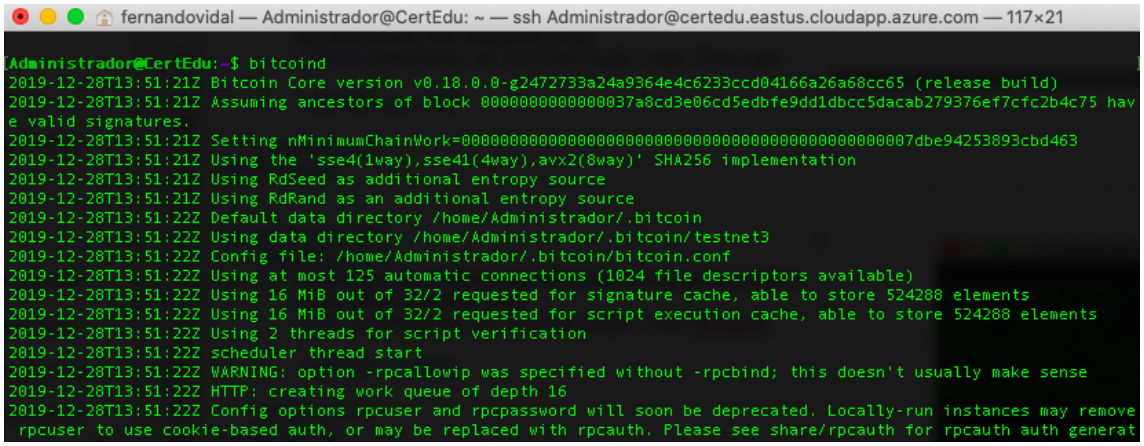

Figura 5.7: *Processo de sincronização rede Bitcoin*

A figura 5.8 apresenta o espaço de armazenamento necessário para que as transações da rede *Ethereum Rospnet* seja uniformizada. Esse ambiente de replicação permite notar o comportamento

[1] https://faucet.ropsten.be/
[2] https://faucets.blockxlabs.com/Ethereum
[3] https://coinfaucet.eu/en/btc-testnet/
[4] https://faucet.uo1.net/send.php

de algumas propriedades de uma *blockchain* pública, como por exemplo, dados que nunca se apagam. Isso significa que, para uma aplicação deste género, é necessário ter uma infraestrutura capaz de ser incrementada sistematicamente.

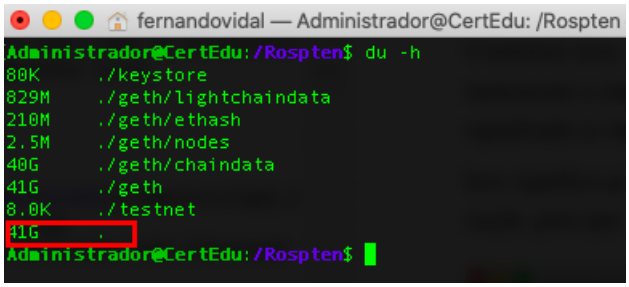

Figura 5.8: *Tamanho da rede de testes Rospnet*

A linguagem de programação escolhida para este projeto foi a *.NET Core* [1], uma ferramenta de propriedade da Microsoft, adaptada para transferir as ferramentas da arquitetura *.NET*, originalmente exclusiva de plataformas *Windows*, para operar em multi-plataformas (*Windows*, *Linux* e *Mac*). A tecnologia oferece para cada sistema operacional uma *Integrated Development Environment* (IDE) de código aberto correspondente, na qual este trabalho utilizou-se do *Visual Studio Community For Mac*.

Conforme nota-se na figura 5.5, o ambiente escolhido roda um sistema operacional *Linux Ubuntu 18.04*. Assim, foi instalado no ambiente o pacote *.NET core* 2.2, e configurado um servidor *nginx* [2] de HTTP. A seguir são mostrados trechos do arquivo de configuração *nginx.conf*:

[1] `https://docs.microsoft.com/pt-br/dotnet/core/about`
[2] `https://nginx.org/en/`

```
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;
events {
	worker_connections 768;
	# multi_accept on;
}
http {
	## Basic Settings ##
	sendfile on;
	tcp_nopush on;
	tcp_nodelay on;
	keepalive_timeout 65;
	types_hash_max_size 2048;
	include /etc/nginx/mime.types;
	default_type application/octet-stream;

	## SSL Settings ##
	ssl_protocols TLSv1 TLSv1.1 TLSv1.2; # Dropping SSLv3, ref: POODLE
	ssl_prefer_server_ciphers on;

	## Logging Settings ##
	access_log /var/log/nginx/access.log;
	error_log /var/log/nginx/error.log;

	## Gzip Settings ##
	gzip on;

	## Virtual Host Configs ##
	include /etc/nginx/conf.d/*.conf;
	include /etc/nginx/sites-enabled/*;
}
```

Por último, devido a necessidade de haver um servidor para armazenar e hospedar arquivos publicados no processo de certificação, como perfil, lista de revogação e certificado, a aplicação sugere a configuração de um servidor de *File Transfer Protocol* (FTP) para agilizar os testes e centralizar as configurações. Este recurso é criado no ambiente de simulações, na hipótese de ser o recurso contratado pelas universidades para gerar e hospedar os arquivos que serão expostos aos interessados. Na secção 5.4.3, é detalhada na área de parametrização do CertEdu, como realizar este apontamento para um servidor de FTP, informando as credenciais e endereços.

No ambiente de simulação, foi utilizado o *vsftpd* [1] para prover o serviço de FTP. A seguir, é demonstrado um trecho de sua configuração:

```
listen=No
listen_ipv6=YES
anonymous_enable=NO
local_enable=YES
write_enable=YES
local_umask=022
dirmessage_enable=YES
use_localtime=YES
xferlog_enable=YES
connect_from_port_20=YES
local_root=/var/www/html/wwwroot
chroot_local_user=YES
secure_chroot_dir=/var/run/vsftpd/empty
pam_service_name=vsftpd
rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
ssl_enable=NO
pasv_enable=Yes
pasv_min_port=10000
pasv_max_port=10100
allow_writeable_chroot=YES
```

5.4 Protótipo CertEdu

Este trabalho apresenta o projeto CertEdu, uma ferramenta criada para simular e testar o uso da tecnologia *blockchain* para emissão e verificação de certificados académicos. Ao observar sua arquitetura construída, este trabalho contribui aos interessados em desenvolver soluções similares, fornecendo informações técnicas de integrações, avanços e limitações, através do compartilhamento da experiência realizada.

O CertEdu foi construído sobre plataforma *.NET Core 2.2*, com a linguagem de programação *C#*. Utiliza banco de dados *MySQL* [2] operando com *Entity Framework Core*. Sua interface de *front-end* combina o uso de HTML5, *JavaScript* e *BootStrap* [3].

A figura 5.9, apresenta uma visão geral da aplicação, com ênfase nos processos de emissão e verificação, e são detalhados a seguir.

[1] http://vsftpd.beasts.org/
[2] https://www.mysql.com/
[3] https://getbootstrap.com/

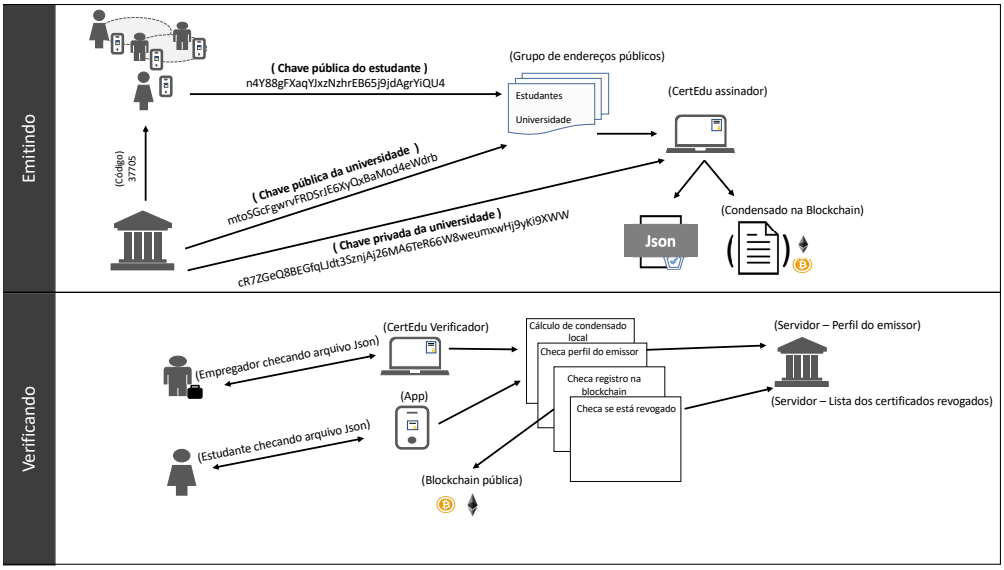

Figura 5.9: *Visão geral da aplicação CertEdu*

5.4.1 Processo de emissão

O primeiro passo do processo de emissão é gerar um identificador, uma espécie de código, que deve ser enviado ao estudante que deseja ter o seu certificado emitido. O estudante utilizando o aplicativo móvel *Blockcerts*, informa o código recebido e aponta no aplicativo para o local onde está hospedado o arquivo de perfil da universidade. Ao concluir esse registo, o aplicativo móvel, automaticamente gera um endereço público para o aluno, e o envia para a universidade através de um ponto de entrada (API) que está identificado no arquivo de perfil da universidade. Findado essa primeira etapa, a universidade já tem um endereço público que representa o aluno, que poderá ser adicionado ao certificado.

A segunda etapa notada pela figura 5.9, é a formação de grupos de endereços públicos de alunos, para serem emitidos simultaneamente. A CertEdu, permite que o administrador crie diversos grupos com 2 ou mais alunos, para que possam ser relacionados futuramente em um processo de emissão. Por exemplo, um grupo de alunos que está cursando uma mesma disciplina pode ser cadastrado, e dado o fim do curso, uma emissão é processada para esse grupo. O endereço público de cada aluno, está sempre relacionado a um tipo de *blockchain*. Isso significa que se um aluno enviou um endereço público de *Bitcoin*, o grupo em que ele será incluído deve ser da mesma categoria.

Por último, a última fase do processo de emissão é associar um grupo de estudantes com um modelo pré-moldado, gerar certificados individuais para cada aluno do grupo, e assinar os documentos com a chave privada da universidade. Como resultado deste processamento, é gerado o arquivo digital do certificado, representado na figura 5.9 através de um arquivo JSON. Finalmente, o condensado do certificado digital é calculado e registado na *blockchain*, para que seja possível a realização de futuras verificações sobre a veracidade do arquivo digital.

Algumas imagens da aplicação relacionadas a esse processo são aqui apresentadas:

1. A figura 5.10 mostra a área da aplicação onde os modelos de certificados são criados. As informações necessárias para cada modelo são: logótipo, título, descrição, histórico, e um campo de parâmetro que indica se o correio eletrónico do aluno deve ficar exposto no arquivo digital ou não (por questões de privacidade). Nota-se também na imagem que há um campo de estado, orientado por cores, que indica o sucesso ou fracasso da criação do modelo. Na prática, ao cadastrar um certificado, o CertEdu tenta gerar o modelo interagindo com o componente *cert-tools*. Na hipótese de alguma razão a ação falhar, como por exemplo: falta de permissão no diretório de saída, chave privada da universidade inválida, ou algum outro erro retornado pelo *cert-tools*, o registo fica pendente com o estado a vermelho, e não pode ser utilizado.

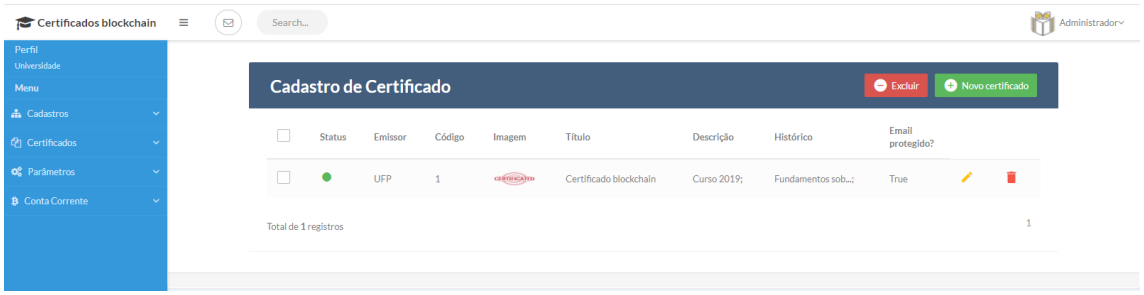

Figura 5.10: *CertEdu, modelo de certificado*

2. A figura 5.11 mostra na aplicação o cadastro de entidades emissoras. Nota-se que o sistema foi modelado para permitir múltiplas entidades, ou locais da mesma entidade. As informações que são solicitadas são códigos para identificar o emissor, nome, correio eletrónico, logótipo da entidade, imagem com a assinatura do responsável legal, endereço da página da Internet, endereço onde ficará o perfil da universidade e endereço onde ficará a lista de revogação.

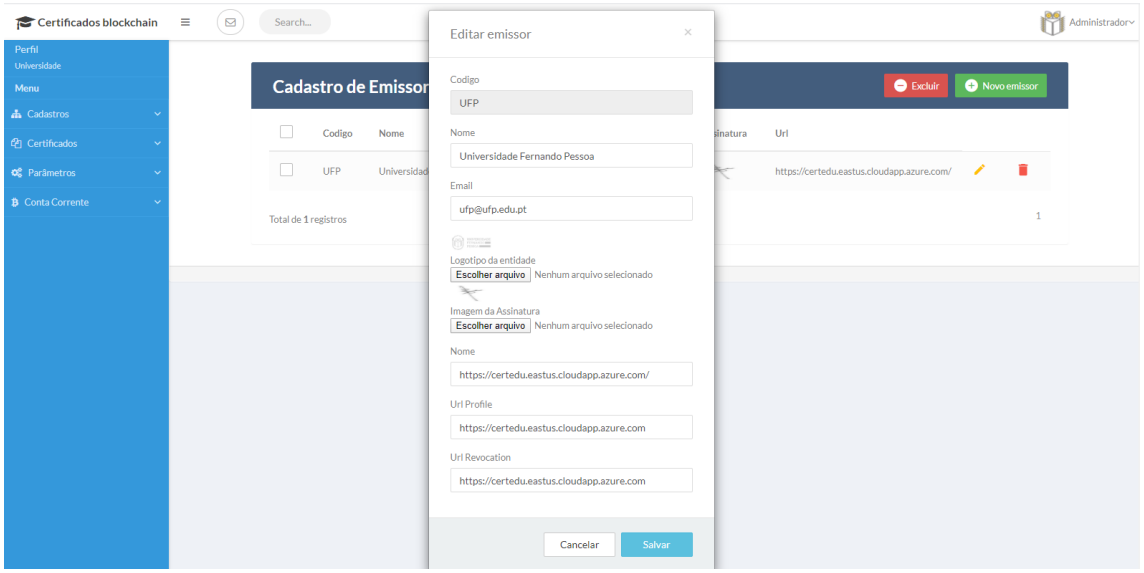

Figura 5.11: *CertEdu, múltiplos emissores*

3. A figura 5.12 mostra uma área da CertEdu em que devem ser configuradas as chaves públicas de cada entidade. A chave é embarcada em cada certificado emitido, servindo como uma espécie de identificador da universidade. O interessado consegue validar se a chave realmente pertence a universidade, por conta do processo de verificação que consulta a chave no endereço de perfil da universidade, confirmando assim a sua identidade. Cada entidade cadastrada no sistema somente pode ter uma única chave pública por tipo de *blockchain* correspondente. Através da imagem, percebe-se que a entidade UFP, possui duas chaves públicas cadastradas, uma para *Bitcoin* e outra para *Ethereum*.

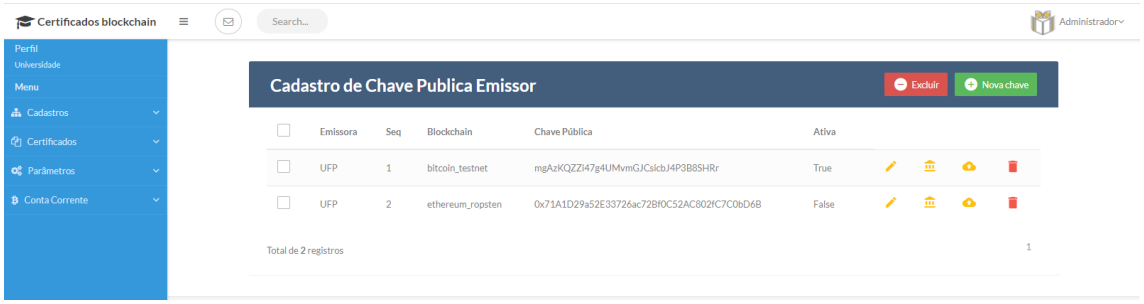

Figura 5.12: *CertEdu, chaves públicas das entidades emissoras*

5.4.2 Processo de verificação

Conforme apresenta figura 5.9, para que seja possível realizar uma verificação é necessário possuir o arquivo digital do certificado. O CertEdu permite resgatá-lo de duas maneiras: através de um acesso ao portal ou solicitá-lo diretamente ao aluno (que tem autonomia para enviá-lo diretamente do seu dispositivo móvel). A próxima etapa é o CertEdu acionar o componente *cert-viewer*,

e iniciar um processo de verificação em vários níveis como: comparar o condensado local do certificado, validar chaves com endereço de perfil da universidade, validar se o condensado está devidamente registado na *blockchain* correspondente, e validar se o certificado está revogado. Este processo é o mesmo que seria executado por verificadores universais, no entanto, o CertEdu tem embarcado essa ferramenta para tornar o sistema mais amigável.

Algumas imagens da aplicação relacionadas a esse processo são aqui apresentadas:

A figura 5.13 mostra uma área da aplicação em que são listados os certificados emitidos. Nota-se na imagem que há três botões de ações para cada certificado e um indicador de estado. O primeiro botão tem a função de descarregar o arquivo JSON (representação do certificado digital). O segundo botão aciona o verificador embarcado (figura 5.14), e o terceiro botão serve para cancelar um certificado que foi emitido (figura 5.15). Em relação ao botão de estado, caso ocorra algum problema ao registar o condensado do certificado na *blockchain*, o indicador ficará vermelho e clicando sobre ele, o sistema exibe o detalhe do erro.

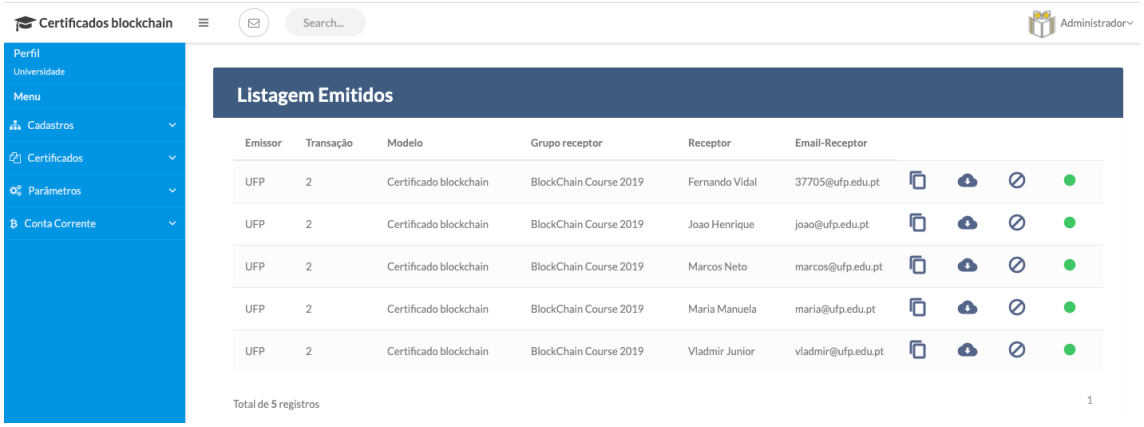

Figura 5.13: *CertEdu, lista de certificados emitidos*

Figura 5.14: *CertEdu, verificador embarcado*

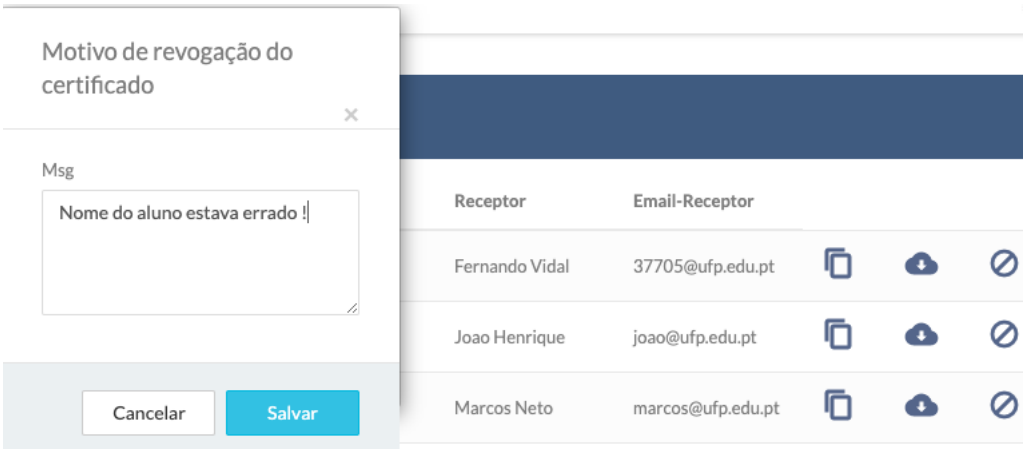

Figura 5.15: *CertEdu, função de cancelamento de um certificado*

5.4.3 Parâmetros do sistema

Esta secção da aplicação permite aos administradores configurar os parâmetros da ferramenta. As configurações aqui disponibilizadas estão todas relacionadas com as opções fornecidas pela plataforma *Blockcerts*.

Conforme apresenta a figura 5.16, o sistema está modelado de modo a permitir que cada unidade emissora tenha suas configurações personalizadas. Por exemplo, há um campo para informar o local onde está armazenada a chave privada. Geralmente, por questões de segurança, a chave privada é um dispositivo externo, USB, que fica em poder do proprietário. Nesse exemplo de configuração, a universidade deve colocar e conectar o dispositivo USB no servidor, para que o sistema consiga emitir os certificados.

Uma outro exemplo personalizado de configuração, são as credenciais do servidor FTP do emissor. Como demonstrado até então, o sistema armazena alguns arquivos de controle de formato JSON, seguindo a especificação *Open Badge*, que são utilizados pelo sistema para garantir a identidade do emissor e também de servirem como parte do mecanismo criado para tratar da revogação de certificado. Com a configuração cadastrada, o sistema gera o arquivo no respetivo formato, e o publica no endereço cadastrado.

Os parâmetros aqui listados nesta secção do sistema, são combinados com outras configurações informadas pelos usuários no sistema. Na figura 5.12, nota-se que a entidade UFP, tem duas redes configuradas: *Bitcoin* e *Ethereum*. Também observa-se na figura 5.10, que existem modelos criados, com imagens e textos pré-definidos. Os diretórios em que serão geradas as imagens desses modelos criados, bem como os certificados assinados e não assinados, deverão obedecer a estrutura informada por esta secção de parâmetros.

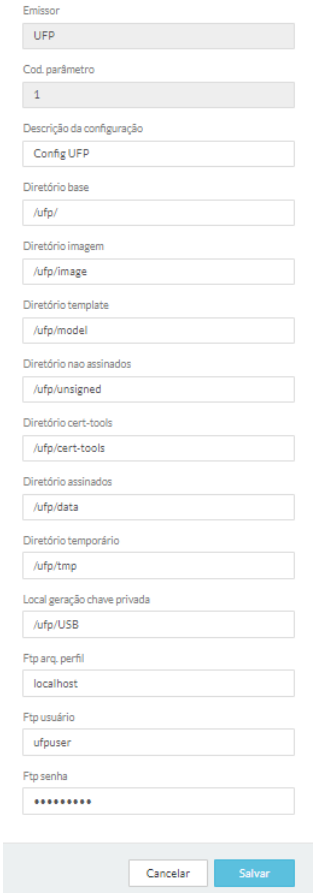

Figura 5.16: *CertEdu, parâmetros*

5.4.4 Processo de atualização

Como pode-se notar através da figura 5.1, a arquitetura do CertEdu foi planeada de modo a tornar possível atualizações isoladas de componentes do sistema. Nesta secção é demonstrado um exemplo de atualização do componente *cert-issuer*, que é realizada sem afetar as demais partes do protótipo.

A atualização deve ser feita dentro de uma sessão de terminal da máquina Linux que compõe a arquitetura do sistema, e deve obedecer uma sequência de dois passos, conforme apresentam as figuras 5.17 (passo 1), 5.18 (passo 2) e 5.19 (passo 2).

O primeiro passo é acionar uma ferramenta de controle de versão, o *github* [1], que tem a função de descarregar a última versão atualizada do *cert-issuer*, para um diretório local. Esta operação é apresentada na figura 5.17.

[1] https://github.com/

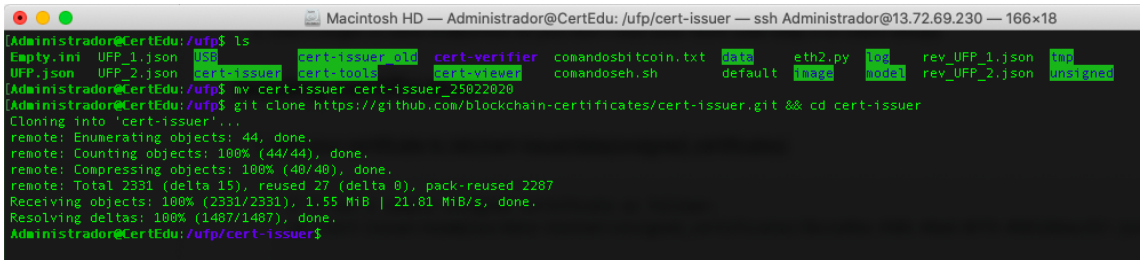

Figura 5.17: *Atualização cert-issuer passo 1*

O segundo passo é executar um programa do *Blockcerts* para instalação/atualização do componente, o *setup.py*, que vem junto com a pasta baixada através do passo 1. O programa *setup.py* quando acionado sem nenhuma diretiva, indica por padrão que o *cert-issuer* será configurado em modo *Bitcoin*. Caso houver a necessidade de preparar o ambiente para *Ethereum*, é preciso acionar o comando com a seguinte diretiva: *python3 setup.py experimental –blockchain=Ethereum*. O início do comando pode ser observado através da figura 5.18.

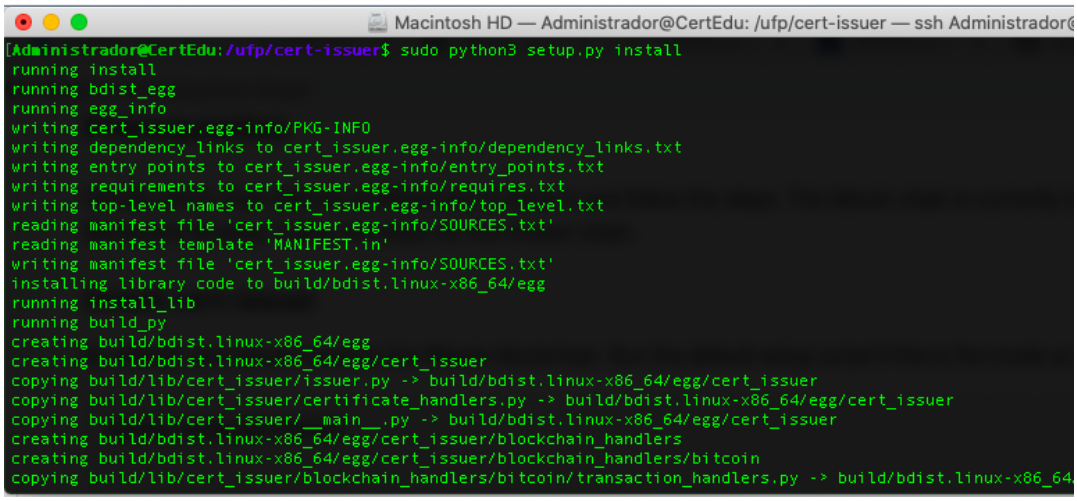

Figura 5.18: *Atualização cert-issuer passo 2*

A figura 5.19, apresenta a conclusão do comando de configuração da rede *blockchain*, em que serão emitidos os certificados.

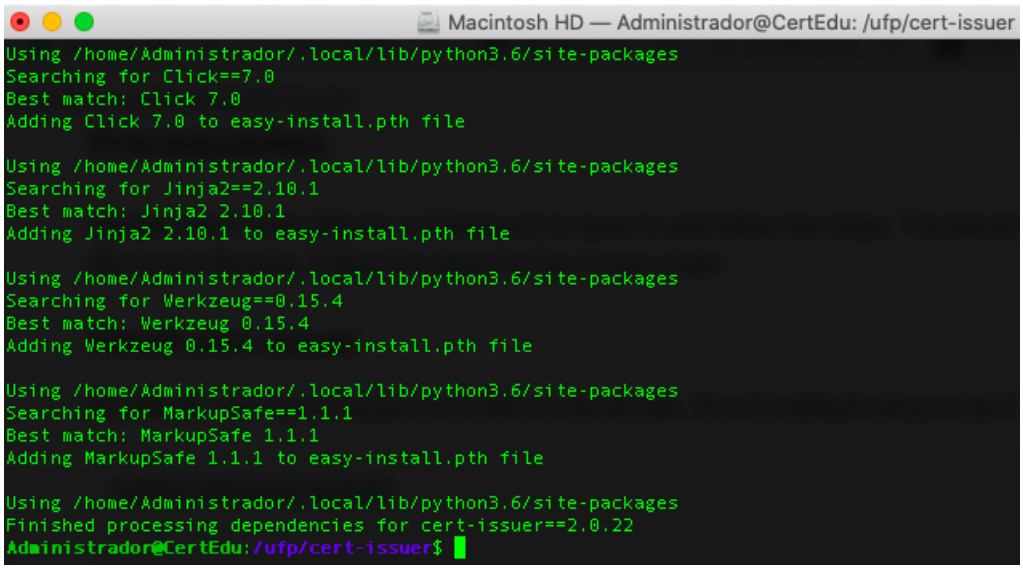

Figura 5.19: *Atualização cert-issuer passo 3*

Durante a execução dos instaladores, é provável que alguns pacotes utilizados pelo *Blockcerts*, acusem erros de dependência. Enquanto esses erros não forem resolvidos, o programa não conclui a instalação do componente.

5.5 Conclusão

Este capítulo apresentou a implementação do protótipo CertEdu, que foi construído utilizando a plataforma *Blockcerts*. Detalhou-se a arquitetura necessária para o funcionamento deste tipo de sistema, bem como enumeradas as principais funcionalidades que uma solução de emissão e gestão de certificados *blockchain*, precisa atender. O capítulo também mostrou algumas das possíveis parametrizações que podem ser realizadas. De forma preliminar é apontado no capítulo, os desafios que uma solução de certificados *blockchain* precisa encarar, sendo que no próximo capítulo, estes serão detalhados através de testes e simulações.

Capítulo 6

Testes e Avaliação

6.1 Introdução

Os testes a seguir simulam algumas das situações que podem ocorrer diante de uma emissão de certificado *blockchain*. Primeiramente, são realizadas tentativas de fraudar o certificado, seja através de edição de dados, seja através de usurpação de propriedade. Logo após esse teste, é avaliado como o sistema de validação se comporta de forma desconectada. E por último, os custos de emissão são estimados.

Utilizando o CertEdu construído por este trabalho, foram emitidos alguns certificados em rede *Bitcoin TestNet*, como por exemplo, o representado em código através das figuras 6.1 e 6.2, e graficamente através da figura 6.3.

Deve-se observar algumas informações importantes para as simulações que venham a seguir. Na figura 6.2, na parte inferior da imagem é informada a raiz de *Merkle* (*MerkleRoot*) e o condensado do certificado (*targetHash*). Nesse caso do exemplo dado, como trata-se de apenas um certificado emitido (não houve emissão em lotes), a informação é a mesma para ambos os campos.

A figura 6.4, demonstra o condensado do certificado registado na *blockchain*. Desta maneira, os verificadores universais conseguem comparar o condensado do certificado com o campo de dados da *blockchain* e assim informar ao usuário que tal certificado está devidamente armazenado.

Figura 6.1: *Certificado emitido para testes, formato JSON parte I*

```
criteria:
    narrative:          "Cursou as disciplinas ..."
signatureLines:
  0:
    type:
      0:                "SignatureLine"
      1:                "Extension"
    jobTitle:           "Universidade Fernando Pessoa"
    image:              "…zgI7/xAUAAAAAElFTkSuQmCC"
    name:               ""
verification:
  type:
    0:                  "MerkleProofVerification2017"
    1:                  "Extension"
  publicKey:            "ecdsa-koblitz-pubkey:mgAzKQZZi47g4UMvmGJCsicbJ4P3B8SHRr"
signature:
  type:
    0:                  "MerkleProof2017"
    1:                  "Extension"
  merkleRoot:           "dd2bbeb060e9b6d7707320cfa745ed0976f5662493500ebb4321692c47da1d43"
  targetHash:           "dd2bbeb060e9b6d7707320cfa745ed0976f5662493500ebb4321692c47da1d43"
  proof:                []
  anchors:
    0:
      sourceId:         "906d0e9895198aabc54aabd79aff56a10ced2aab26fc1bc80cf25bb9f3e20868"
      type:             "BTCOpReturn"
      chain:            "bitcoinTestnet"
```


Figura 6.2: *Certificado emitido para testes, formato JSON parte II*

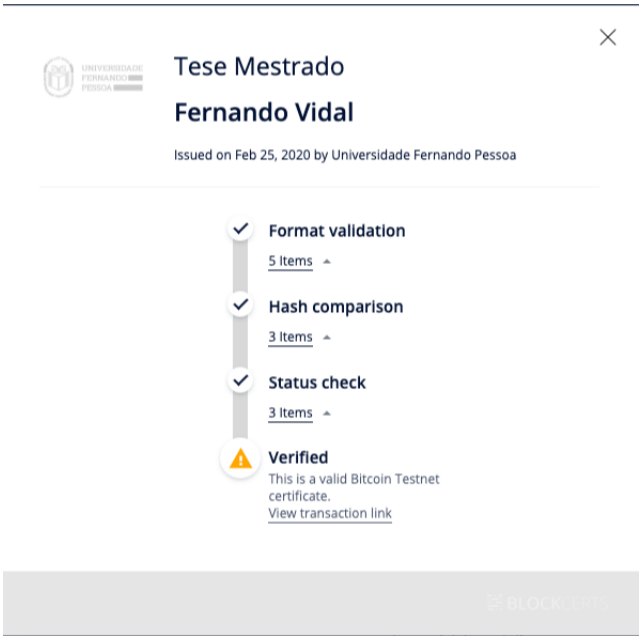

Figura 6.3: *Certificado emitido para testes, visualização gráfica*

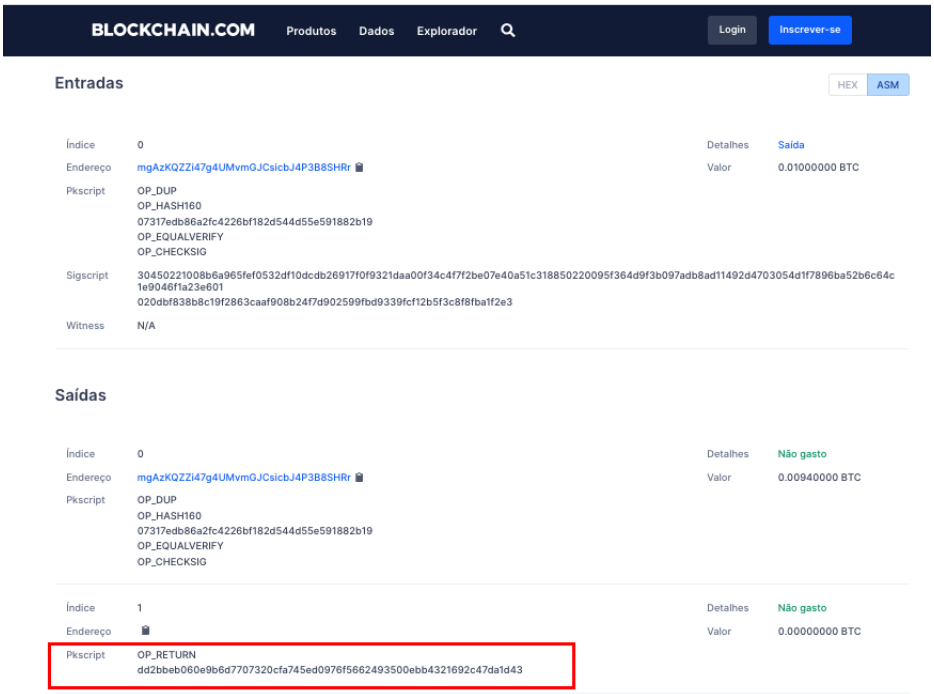

Figura 6.4: *Visualização do condensado do certificado, armazenado em uma transação da blockchain na rede Bitcoin*

No exemplo a seguir é apresentado um cenário em que vários certificados foram agrupados e emitidos em um único registo (emissão em lote). Nota-se que o valor do campo raiz de *Merkle*, continua sendo armazenado na *blockchain* (vide figura 6.6). No entanto, há uma diferença neste trecho do arquivo JSON do certificado. Percebe-se a presença de informações adicionais, que são representadas pelo campo *proof*, conforme demonstra a imagem 6.5. Com esses dados é possível validar um certificado, mesmo que ele faça parte de um lote, através de uma combinação e cálculos da raiz de *Merkle*, já demonstrada por este trabalho na secção 2.1.4. Maiores informações de como é implementava a prova de *Merkle* nos arquivo JSON do *Blockcerts*, podem ser obtidas em (W3C, 2017).


```
▼ merkleRoot:    "2984f351c8fb2a51e0163ecc4f859662fc4f7e9ec083c396487dd0c45de8319d"
▼ targetHash:    "a6a4cf5bedb4213a775d00173100bd62cfe37dadaf770207ce38882bdef47721"
▼ proof:
  ▼ 0:
    ▼ left:      "18bc681974f95703f9b25601c86274bc5996b524e6989869bd04577ca8731dee"
  ▼ 1:
    ▼ left:      "acc959384945bfa8faabb4dd5d07f350e985010994e0988bf347e18dc6dccb94"
  ▼ 2:
    ▼ right:     "58da35bea7557ca4ee2ab2205c93277f0e9e30e8a8ae3f78c0b0bafd2060fc93"
```


Figura 6.5: *Certificado emitido em lote no formato JSON*

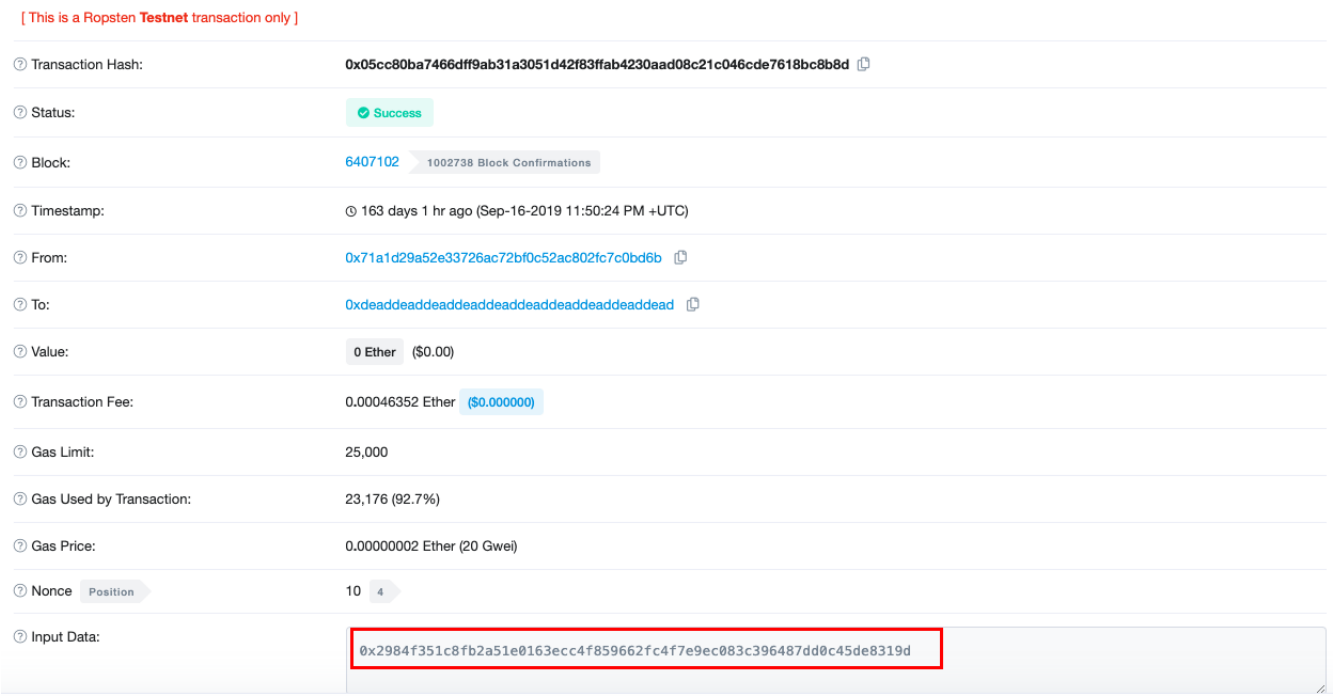

Figura 6.6: *Visualização do condensado do certificado, armazenado em uma transação da blockchain na rede Ethereum*

6.2 Tentativas de adulteração

O *Blockcerts* promete ser capaz de identificar adulterações em um certificado emitido. Esta secção simula tentativas de fraudar um certificado, que são apresentadas a seguir.

6.2.1 Edição de dados

Nesta simulação o diploma representado pela figura 6.3, é alterado. Utilizando um simples editor de texto, modifica-se uma informação original do diploma, incluindo uma letra R no nome do aluno. Nota-se, que o arquivo do diploma digital não é criptografado, e pode ser facilmente editado por usuários que tenham um mínimo de conhecimento em ferramentas de edição de texto.

O resultado da operação de validação do documento alterado, pode ser vista através da figura 6.7, e as regras processadas pelo verificador podem ser observadas através da tabela 6.1.

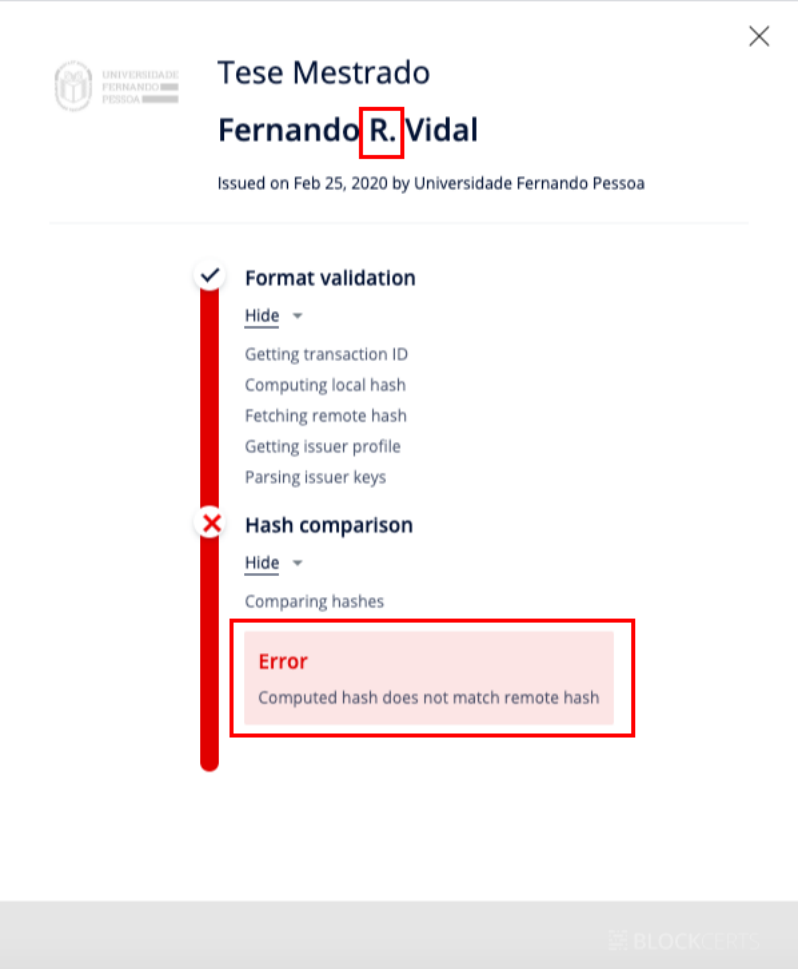

Figura 6.7: *Validação de certificado adulterado*

Tabela 6.1: *Resultado das validações executadas pelo verificador universal*

Nº	**Regra**	**Atendida ?**
1	O número da transação contida no certificado (*sourceId*) existe na *blockchain* referenciada ?	Sim
2	O cálculo do condensado das informações contidas no arquivo, confere com valor informado do campo *targetHash* ?	Não
3	O *targetHash* confere com o conteúdo armazenado *blockchain* ?	Não
4	O endereço público do emissor (servidor) está accessível ?	Sim
5	A chave pública do emissor embarcada no certificado, confere com a chave hospedada no servidor do emissor ?	Sim

Percebe-se nessa experiência que o verificador é capaz de identificar qualquer adulteração de conteúdo do arquivo. Analisando as regras do verificador, mesmo que um atacante adulterasse um certificado e recalculasse o condensado dos dados, a fraude seria detetada, por conta do conteúdo

do campo *targetHash* não conferir com o valor armazenado na *blockchain*.

6.2.2 Apropriação não autorizada

Esta outra simulação de fraude discute a possibilidade de um atacante alterar os emissores e/ou destinatários de um certificado genuíno. Por exemplo, seria interessante a criminosos replicarem certificados verdadeiros de universidades com menor visibilidade, alterando-os como se fossem de prestigiadas universidades. Da mesma maneira, uma fraude rentável seria oferecer diplomas genuínos para destinatários mal intencionados, que pretendem acumular "conquistas".

O *Blockcerts* embarca a chave pública do aluno nos dados do certificado, na secção *recipientProfile*, no campo *publicKey*. Da mesma maneira, na secção *verification*, está embarcada a informação do emissor. Isso significa que tais campos fazem parte do cálculo do condensado (visto na secção 6.2.1), o que garante proteção contra adulteração.

No entanto, um atacante poderia alterar os dados do certificado, informando outras chaves públicas de emissores e destinatários, e recalcular o condensado do certificado. Adicionalmente, para evitar que a regra 3 (tabela 6.1) identifique a fraude, o atacante poderia alterar o campo *sourceId* para uma transação que ele mesmo criou, na qual contém exatamente o valor do condensado do certificado fraudado.

Nesse caso a regra 3 vai passar como verdadeira, porque tanto o cálculo local do condensado quanto a comparação com a informação armazenada na *blockchain* acusarão como válidas. No entanto, o *Blockcerts* consegue identificar essa fraude, porque além de validar o conteúdo da transação, confere se o endereço da transação é realmente de propriedade da universidade emissora. Isso é possível porque o sistema compara o endereço público da transação, com o endereço público hospedado no servidor do emissor.

A figura 6.8 apresenta trechos do arquivo de perfil da universidade, destacando a informação da chave pública. O certificado adulterado, e a transação forjada, apontam para o condensado *mgAzKQZZ*
*i*47*g*4*UMvmGJCsicbJ*4*P*3*B*8*SHRr*. No entanto, a chave pública verdadeira da universidade, aponta para o valor *JKmg*4*UDWqBcnhklMEsDgdDHsRrF*.

Conforme apresenta a figura 6.9, o validador deteta a inconsistência dessas informações, impedindo que tal fraude possa ser concretizada.

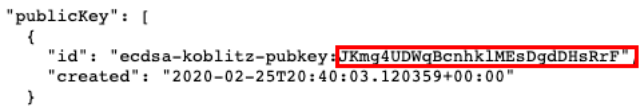

Figura 6.8: *Arquivo de identificação do emissor*

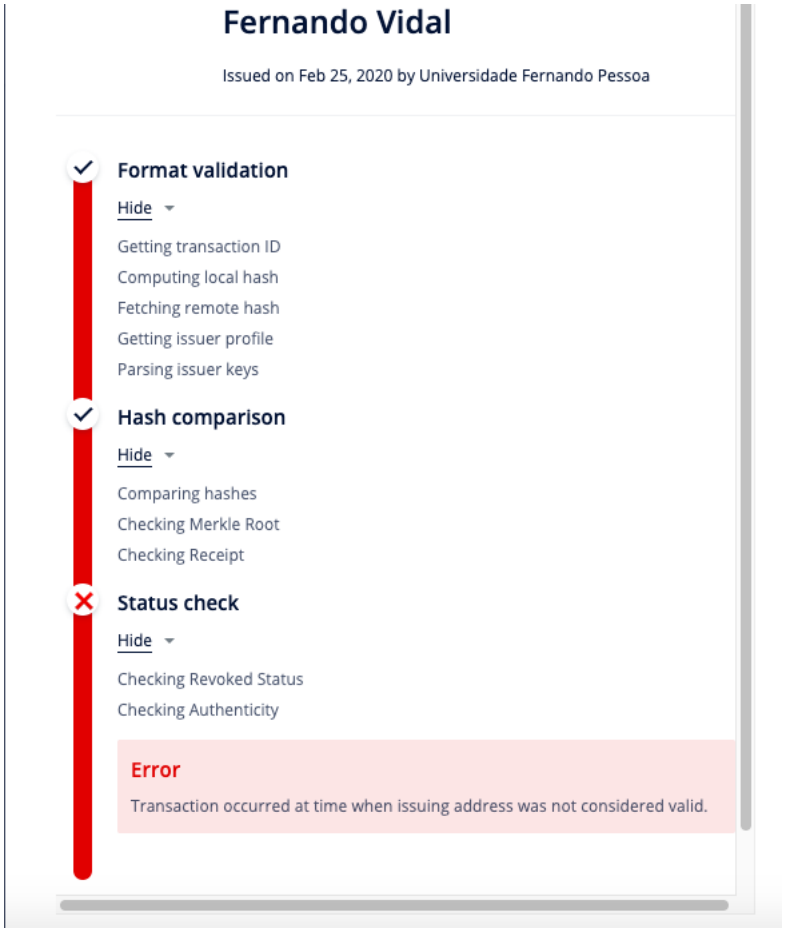

Figura 6.9: *Verificação de certificado com o emissor adulterado*

6.3 Chave roubada e revogação

Esta simulação, avalia a capacidade do *Blockcerts* de identificar fraudes que possam ocorrer em situações na qual a universidade tenha tido a sua chave privada roubada.

Os diplomas digitais emitidos por soluções não-*blockchain*, como as que assinam utilizando uma chave RSA, não garantem em suas provas de assinatura, que a data de criação associada a sua assinatura, condiz com a realidade. Na maioria dos casos, o emissor que assina com as chaves de sua propriedade, deve assinar com a data e hora correta (Ronning and Chung, 2019).

Em situações que a chave privada é roubada o atacante pode querer emitir diplomas retroativos, conseguindo assim garantir conquistas ilegais para qualquer destinatário.

A universidade quando se dá conta que sua chave foi roubada, revoga a validade a partir da data em que se deu o roubo. No entanto, nas soluções não-*blockchain*, como as datas dos certificados não podem ser confiáveis, fica muito difícil determinar quais diplomas ainda são válidos.

O que deve acontecer em uma situação como esta, é que toda credencial emitida com uma chave roubada precisa falhar (Ronning and Chung, 2019). Portanto, utilizando os carimbos confiáveis

de data de uma *blockchain*, pode-se calcular a verdadeira data de emissão de um diploma, e assim determinar a revogação / expiração. Assim, fica muito confiável separar as credenciais válidas das inválidas. Por exemplo, todas as credenciais que estão referenciadas em uma *blockchain*, antes da data reportada do roubo, permanecem de forma segura válidas e não são afetadas por nenhum processo de revogação.

As figuras 6.10 e 6.11, demonstram como o campo de data de emissão de um certificado, pode ser facilmente comparado com seu registo na *blockchain*. Consultando a data da transação correspondente (indicada no campo *sourceId*), consegue-se facilmente realizar a comparação.

Figura 6.10: *Data de emissão do certificado*

No entanto, a análise deve observar que ferramentas de terceiros utilizadas para consultar blocos, podem apresentar informações inconsistentes. Isso não significa que o dado esteja armazenado de forma errada. Na figura 6.12, a informação de data de registo do certificado na *blockchain*, aponta para o horário de 17:58, quando na realidade o certificado foi emitido as 20:58. Essa diferença, nada mais é do que uma apresentação dos dados em relação ao fuso horário. Isso se dá, porque o CertEdu está rodando em um servidor na nuvem, com fuso horário de +3 horas que a consulta realizada, para apresentar dados no fuso horário do Brasil.

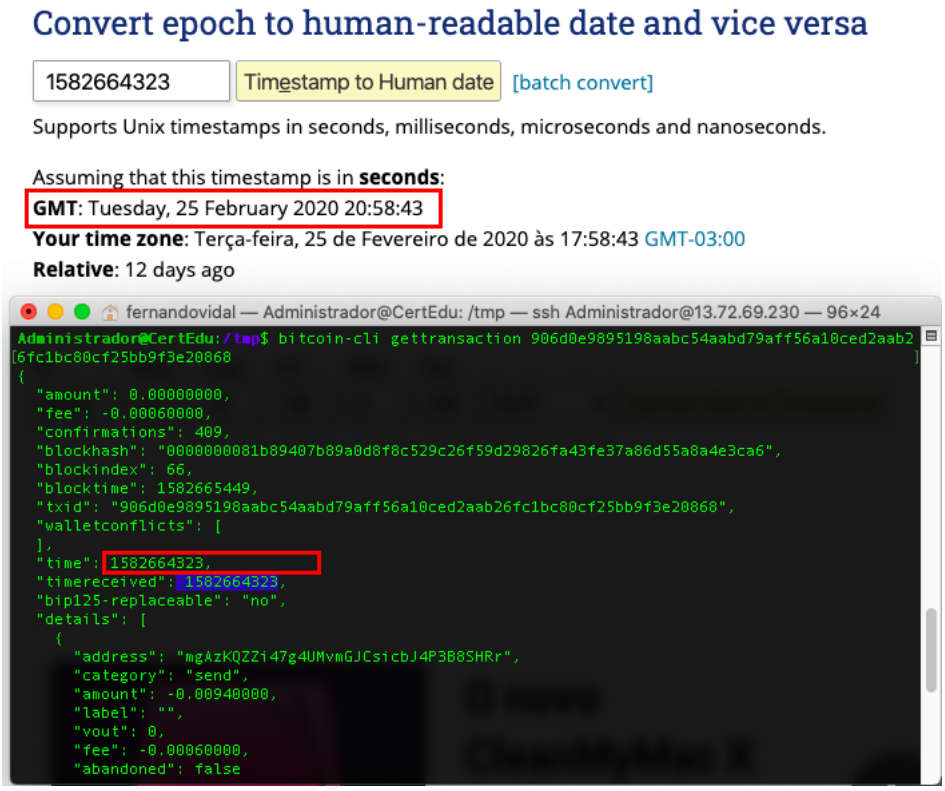

Figura 6.11: *Data de registo da emissão do certificado na blockchain*

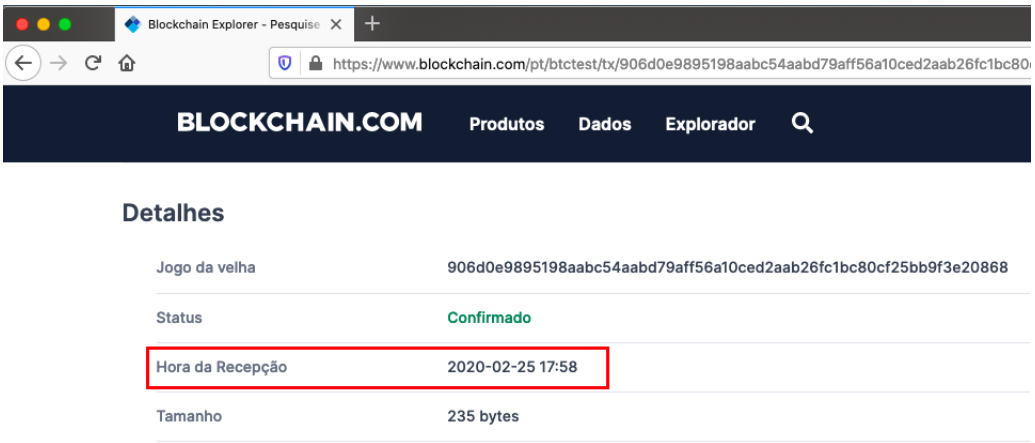

Figura 6.12: *Data de registo da emissão do certificado na blockchain, visualizada através de ferramentas de terceiros*

Nesta questão, conclui-se, que a solução *blockchain* oferece um diferencial em relação a soluções sem *blockchain* , uma vez que garante a estudantes, que os seus diplomas antigos não serão afetados, mesmo que situações inesperadas como um roubo de chave privada aconteça.

6.4 Validação sem acesso ao servidor

Esta simulação é dividida em duas partes. Primeiro, avalia-se a possibilidade do servidor do emissor falhar e depois testa-se a funcionalidade de continuar validando de forma não sincronizada, registos em uma *blockchain* local.

6.4.1 Servidor do emissor inacessível

Pretende-se com essa simulação avaliar o comportamento do sistema, ao deparar-se com uma falha do servidor da universidade. Para isso, foi desligado o servidor em que estão hospedados os arquivos de perfil e revogação, e realizada uma verificação de um certificado genuíno.

Nota-se através da figura 6.13, que o verificador não foi capaz de dizer se o certificado era válido ou não.

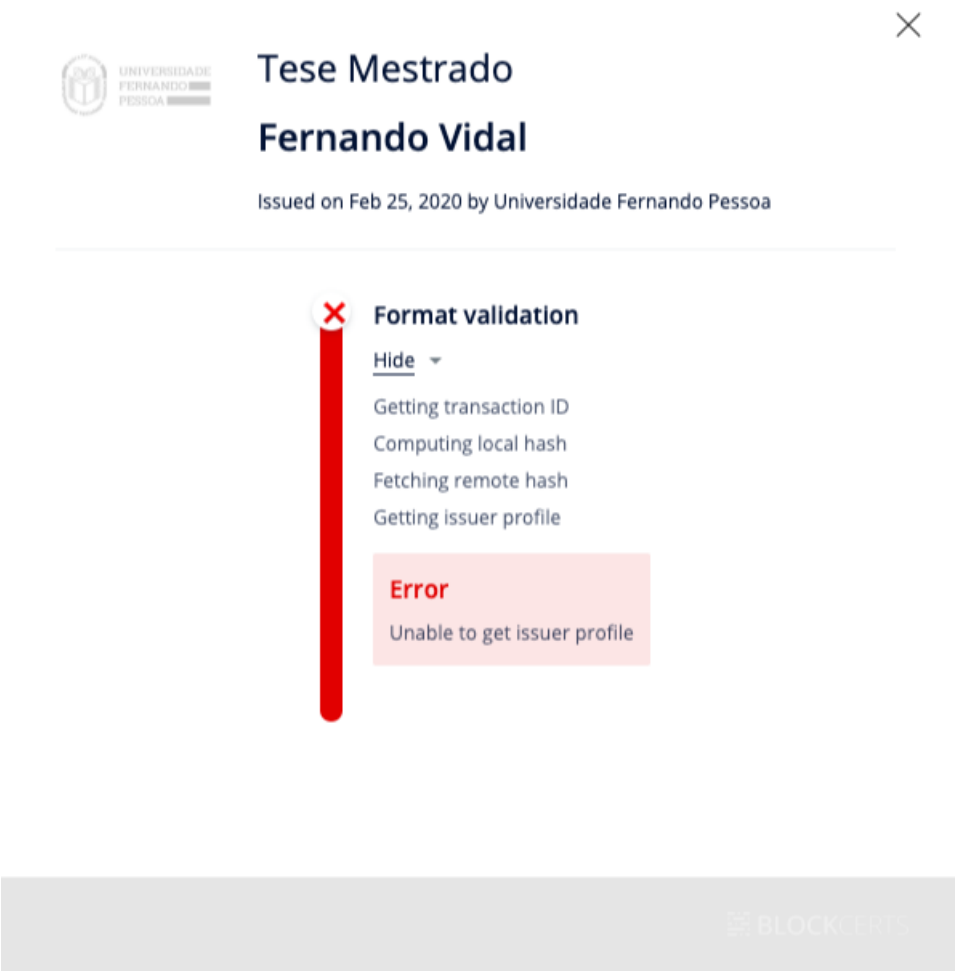

Figura 6.13: *Verificação local, erro ao conectar emissor*

Conclui-se com o teste que a solução ainda não é capaz de oferecer independência em relação ao emissor. Além disso, caso um atacante consiga invadir o servidor, mesmo que temporariamente,

seria possível concretizar os ataques descritos na secção 6.2, por conta da possibilidade de alterar as informações armazenadas no perfil do emissor.

6.4.2 *Blockchain* não sincronizada

Um dos benefícios em utilizar a *blockchain*, seria a possibilidade de continuar a validar certificados, em uma *blockchain* local, mesmo que ela deixe de existir.

Analisando as bibliotecas do protótipo implementado, percebe-se que o verificador utiliza-se de interfaces API, disponibilizadas por ferramentas de consulta de blocos, para verificá-los. Isso significa que para uma validação local seria necessário modificar o endereço de verificação para a base local. Como trata-se de uma operação relativamente fácil, pode-se dizer que a ferramenta atende esse quesito.

Uma outra situação que pode ser simulada, é o verificador validar uma emissão de um certificado, que ainda não foi autorizada pelos mecanismos de consenso. Como citado na secção 3.1.3, a validação dos blocos é um dos pilares que traz toda segurança da tecnologia. Validar um diploma de um registo ainda não confirmado, poderia ser considerado uma falha muito grave.

A figura 6.14, demonstra o resultado dessa operação, e como nota-se o resultado, pode-se afirmar que o sistema também passou neste teste de segurança.

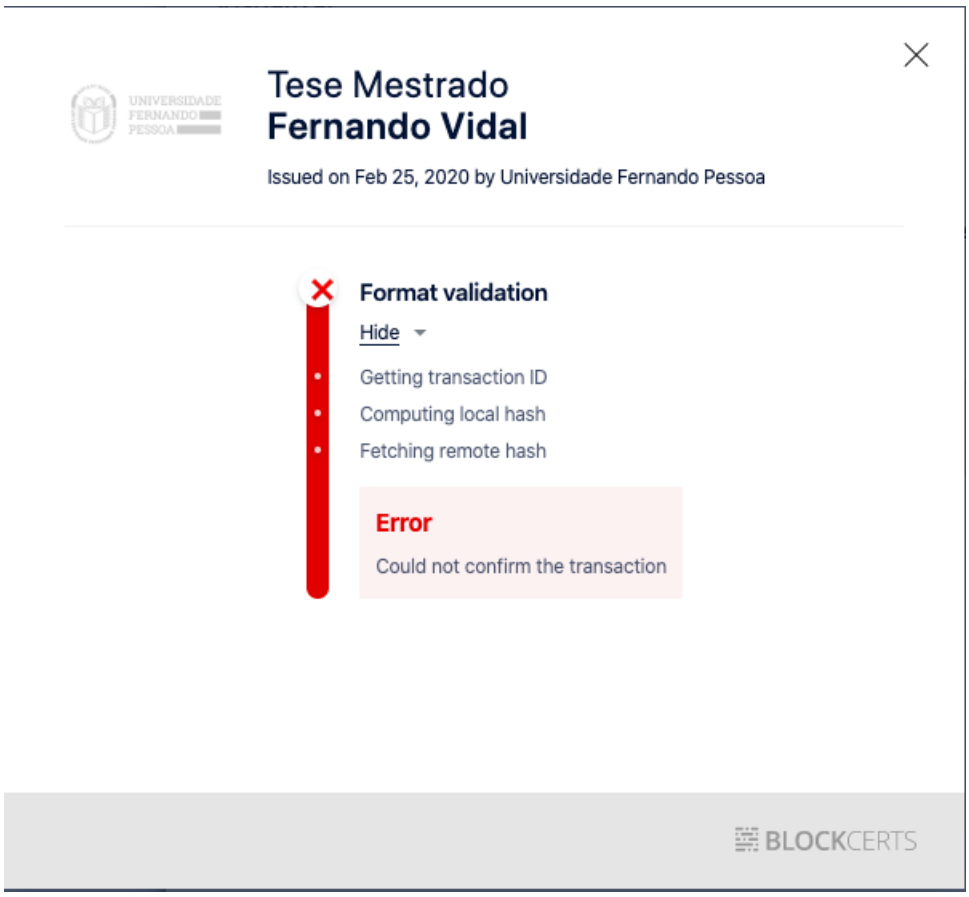

Figura 6.14: *Bloco ainda não confirmado pelos mecanismos de consenso*

6.5 Emissão e custos

A seguir é demonstrado como o processo de publicação pode ser testado, bem como estimar os custos envolvidos. Na primeira simulação é gerado um certificado de forma individual e feito o seu registo público na *Bitcoin*. Na segunda simulação, o objetivo é testar a emissão de vários diplomas sendo emitidos ao mesmo tempo, através do processamento de um único lote. Os passos para que isso aconteça na aplicação são: cadastro do aluno, criação de grupo com alunos relacionados e publicação do grupo.

A primeira simulação é apresentada através da figura 6.15, como se pode notar, um aluno individualmente é relacionado como recetor de um certificado. Sua chave pública é informada e o cadastro deste grupo, que contém apenas um aluno, é realizado no sistema. O registo criado é vinculado a descrição *Single Student*.

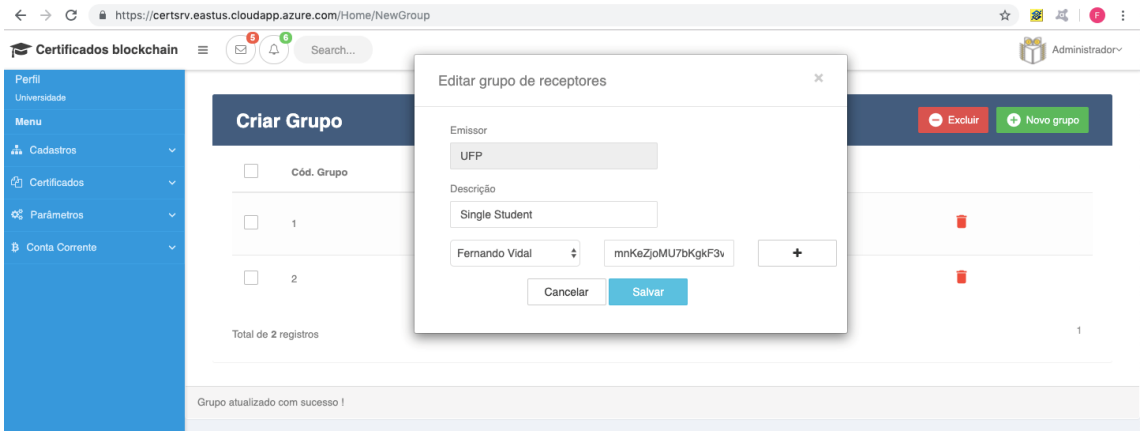

Figura 6.15: *Simulação 1 - destinatário que receberá o certificado*

O próximo passo é selecionar o grupo, o emissor, e o modelo do certificado. Em seguida, realizar a efetivação da publicação na *blockchain*, conforme ilustra a figura 6.16.

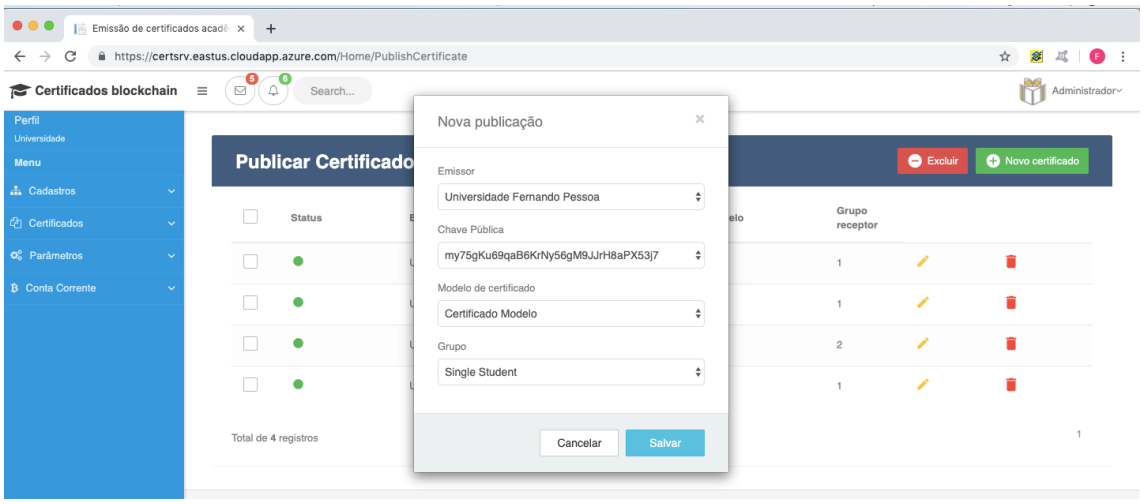

Figura 6.16: *Simulação 1 - publicação de um certificado*

Na segunda simulação, conforme apresenta a figura 6.17, é criado um grupo com três alunos. O objetivo desta experiência é de efetivar uma publicação em lote, e constatar que embora três certificados foram emitidos, um único registo de transação é feito na *Bitcoin*, portanto o custo para essa operação é o mesmo que o gerado pela primeira simulação.

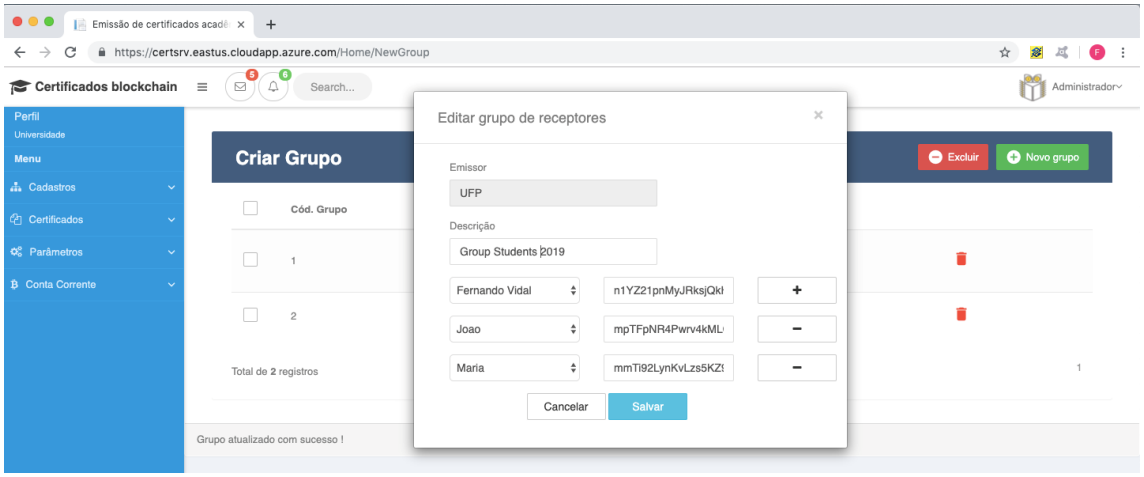

Figura 6.17: *Simulação 2 - grupo de destinatários que receberão o certificado*

Como se pode notar através da figura 6.18, o custo de Satoshi foi o mesmo para emitir 1 ou 3 certificados (133500). Isso é possível por que o *Blockcerts* combina o condensado de todos os certificados envolvidos no lote e regista apenas um único condensado no registo da *blockchain*. Segundo a documentação do *Blockcerts*, é possível emitir cerca de 2000 certificados em cada lote [1].

Considerando a taxa padrão de 60,000 Satoshis para emitir cada certificado, que hoje seria em torno de $5 USD, seria inviável caso a única opção fosse o registo individual do diploma na rede. Felizmente a opção em lote viabiliza o projeto e na hipótese de utilizar a carga máxima de 2000 certificados, o custo por documento emitido seria em torno de $0,0025 USD, valores totalmente irrisórios diante do benefício oferecido. Além disso essa taxa pode ser modificada através de um parâmetro do *cert-issuer*. Quanto maior o valor, mais rápido o certificado é registado na rede e quanto menor o valor maior pode ser o tempo de espera [2] .

[1] `https://github.com/blockchain-certificates/cert-issuer`
[2] `https://bitcoinfees.earn.com/`

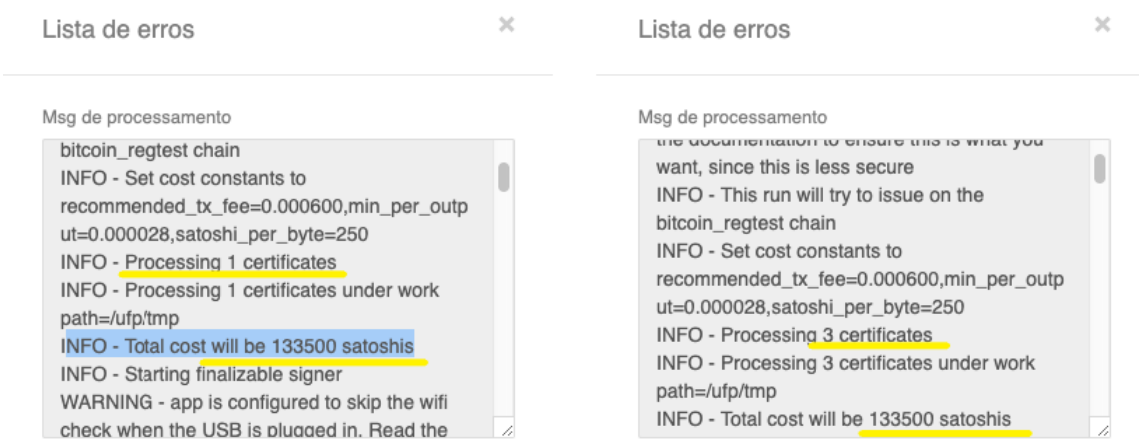

Figura 6.18: *Comparação de custos entre emissão individual ou em lote de certificados*

6.6 Conclusão

O capítulo teve como finalidade executar testes sistemáticos, capazes de gerarem resultados para a avaliação da solução *blockchain*, para gestão de diplomas de ensino superior. Foram emitidos certificados nas redes *Bitcoin* e *Ethereum*, em modo *TestNet* e *Rospnet*, que são muito próximos de um cenário real. Em relação ao primeiro conjunto de testes relacionados a proteção contra adulteração, a tecnologia portou-se de forma consistente e segura. Os testes também demonstraram, que os recursos oferecidos pelas soluções *blockchain*, permitem proteger de uma forma mais eficiente, de atacantes que tenham roubado a chave privada da universidade, e venham a emitir certificados retroativos.

Já em relação ao funcionamento desconectado, os testes apontaram que uma validação local de um certificado ainda não publicado (pendente de aceite nos mecanismos de consenso), não seja considerado válido, oferecendo assim um benefício contra a criação de registos inexistentes. No entanto, ao tentar validar registos válidos, percebeu-se que a operação funciona condicionada a disponibilidade do servidor da universidade, o que permite concluir que nesse aspeto a tecnologia atende apenas parcialmente a questão.

E por último, os testes de emissão permitem concluir que a solução-*blockchain* é perfeitamente viável do ponto de vista financeira, por conta da tratativa de emissão em lote, que utiliza o conceito da raiz de *Merkle* para sua verificação.

Capítulo 7

Conclusão

7.1 Resultados

O presente trabalho, analisou e aplicou a tecnologia *blockchain* na gestão de diplomas do ensino superior. As revisões bibliográficas apresentadas nos capítulos 2 e 3, permitiram uma compreensão mais acentuada da tecnologia, de maneira a identificar pontos de aplicação para a área da educação, na emissão e verificação de diplomas.

Conforme avaliado no capítulo 4, existem diversas soluções sendo propostas para digitalizar um certificado dentro da *blockchain*. No entanto, percebeu-se que a solução em destaque é a *Blockcerts*, e está mais próxima de se tornar uma solução definitiva para este problema. O grande diferencial da *Blockcerts* em relação as demais, é o fato do projeto estar sendo preparado para funcionar em qualquer *blockchain*.

A materialização do objetivo de utilizar a *blockchain* para gestão de diplomas, concretizou-se diante da construção do protótipo CertEdu, em que foi possível operar emissões, revogações, partilhas e verificações de certificados académicos.

Diante das simulações apresentadas pelo capítulo 6, conclui-se que a tecnologia é inovadora e segura, garantindo que adulterações não sejam permitidas. Também foi evidenciado, que a *blockchain* oferece melhores recursos para atuar no caso de perda ou roubo da chave privada da universidade, protegendo a entidade contra emissões indevidas retroativas. Além disso, o nível de privacidade oferecido pela tecnologia, por gravar apenas o condensado do certificado na *blockchain*, torna a solução menos propensa ao vazamento de dados, do que as soluções de certificados digitais sem *blockchain*.

No entanto, os testes também apontaram que o funcionamento desconectado, ainda é uma questão que precisa ser trabalhada. Embora algumas *blockchain*s já ofereçam recursos, como o contrato inteligente, que permitiria facilmente resolver os pontos de centralização colocados, a premissa da solução operar em qualquer tipo de *blockchain* não foi atendida, por isso essas questões que ainda impedem a descentralização, ainda são colocadas por este trabalho como pendentes.

Por fim, conclui-se que a aplicação da *blockchain* na gestão de certificados académicos é perfeitamente possível e que a tecnologia já oferece benefícios em relação a privacidade e revogação, em comparação as soluções digitais. No entanto, para que seja possível usufruir de todo o potencial da tecnologia, os pontos de centralização endereçados por esse trabalho, como a validação do perfil emissor e revogação, precisam ser migradas para funcionalidades que operem dentro da própria *blockchain*.

7.2 Trabalhos futuros

O sistema desenvolvido é um protótipo de uma implementação de uma solução para emissão e verificação de certificados acadêmicos. Pode-se futuramente evoluir a ferramenta, incorporando outras *blockchain* e ampliar o objeto de análise.

Também há espaço para investigação de novos métodos e técnicas capazes de descentralizar os pontos de centralização, identificados na solução *Blockcerts*.

Em relação ao verificadores, existe uma oportunidade de padronizar seu funcionamento, bem como padronizar a visualização digital dos certificados.

Os VCs, têm estado cada vez mais presente nas últimas publicações do *Blockcerts*, o que aponta uma nova tendência de padronização. Por essa razão, como trabalho futuro caberia uma análise mais profunda do modelo VC, para transformar o genérico arquivo JSON apresentado pelo modelo de revogação, em um padrão universal. Ainda em relação a revogação, pode-se pensar em aprofundar as análises em relação aos custos gerados pela solução, em diversos tipos de cenários.

E por último, assim como as operações corretivas enfrentam grandes desafios nas redes *blockchain*, também não é uma tarefa fácil mover dados antigos para arquivo morto. Desta forma, pode ser avaliado como trabalho futuro, analisar como os dados acadêmicos obsoletos, poderiam ser movidos para histórico.

Referências

Abdullah Al Hasib and Abul Ahsan Md Mahmudul Haque. A comparative study of the performance and security issues of AES and RSA cryptography. In *Proceedings - 3rd International Conference on Convergence and Hybrid Information Technology, ICCIT 2008*, 2008. ISBN 9780769534077. doi: 10.1109/ICCIT.2008.179. 9, 10

Franco Amati. First official career diplomas on Bitcoin's blockchain, 2015. URL `https://blog.signatura.co/first-official-career-diplomas-on-bitcoin-s-blockchain-69311acb544d`. 6

Andreas M Antonopoulos. *Mastering Bitcoin: Unlocking Digital Crypto-Currencies*. O'Reilly Media, Inc., 1st edition, 2014. ISBN 1449374042, 9781449374044. vii, 33, 34, 35

Andreas M. Antonopoulos and Gavin Wood. *Mastering Ethereum*. Cambridge University Press, 2013. ISBN 9788578110796. doi: 10.1017/CBO9781107415324.004. 47

Ark. All-in-One Blockchain Solutions., 2016. URL `https://ark.io/`. 62

B. Community. The 51% attack, 2017. URL `https://learncryptography.com/cryptocurrency/51-attack`. 43

Clare Baldwin. Bitcoin worth $72 million stolen from Bitfinex exchange in Hong Kong, 2016. URL `https://www.reuters.com/article/us-bitfinex-hacked-hongkong-idUSKCN10E0KP`. 43

Elaine B Barker, William C Barker, William E Burr, W Timothy Polk, and Miles E Smid. Recommendation for Key Management - Part 1: General. *NIST Special Publication 800-57*, 2012. doi: 10.6028/NIST.SP.800-57pt1r4. x, 11

Massimo Bartoletti and Livio Pompianu. An analysis of Bitcoin OP_RETURN metadata. *CoRR*, abs/1702.0, 2017. URL `http://arxiv.org/abs/1702.01024`. 69, 77

L E Borges. *Python para Desenvolvedores: Aborda Python 3.3*. Novatec Editora, 2014. ISBN 9788575224052. URL `https://books.google.com.br/books?id=eZmtBAAAQBAJ`. 51

Vitalik Buterin. A next-generation smart contract and decentralized application platform. *Ethereum*, 2014. 47

Christian Cachin and Marko Vukolić. Blockchain consensus protocols in the wild. In *Leibniz International Proceedings in Informatics, LIPIcs*, 2017. ISBN 9783959770538. doi: 10.4230/LIPIcs.DISC.2017.1. 37, 38, 64, 65

Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance and proactive recovery. *ACM Transactions on Computer Systems (TOCS)*, 20(4):398–461, nov 2002. ISSN 0734-2071. doi: 10.1145/571637.571640. URL http://dl.acm.org/doi/10.1145/571637.571640. 38

Sai Kiran Ch and Sushmitha Popuri. Impact of online education: A study on online learning platforms and edX. In *2013 IEEE International Conference in MOOC, Innovation and Technology in Education (MITE)*, pages 366–370. IEEE, dec 2013. ISBN 978-1-4799-1626-9. doi: 10.1109/MITE.2013.6756369. URL http://ieeexplore.ieee.org/document/6756369/. 67

Guang Chen, Bing Xu, Manli Lu, and Nian-Shing Chen. Exploring blockchain technology and its potential applications for education. *Smart Learning Environments*, 5(1):1, 2018. ISSN 2196-7091. doi: 10.1186/s40561-017-0050-x. URL https://slejournal.springeropen.com/articles/10.1186/s40561-017-0050-x. 24

Mihyun Chung and Jaehyoun Kim. The internet information and technology research directions based on the fourth industrial revolution. *KSII Transactions on Internet and Information Systems*, 2016. ISSN 22881468. doi: 10.3837/tiis.2016.03.020. 24

European Commission. Council Directive 2001/115/EC of 20 December 2001 amending Directive 77/388/EEC with a view to simplifying, modernising and harmonising the con- ditions laid down for invoicing in respect of value added tax. *Official Journal L 015*, pages 0024 – 0028, 2002. 14

Eric Conrad, Seth Misenar, and Joshua Feldman. Domain 3. In *Eleventh Hour CISSP®*, pages 47–93. Elsevier, 2017. doi: 10.1016/B978-0-12-811248-9.00003-6. URL https://linkinghub.elsevier.com/retrieve/pii/B9780128112489000036. 15

Nicolas T Courtois and Lear Bahack. On Subversive Miner Strategies and Block Withholding Attack in Bitcoin Digital Currency. *CoRR*, abs/1402.1, 2014. URL http://arxiv.org/abs/1402.1718. 36

Nicolas T Courtois, Marek Grajek, and Rahul Naik. The Unreasonable Fundamental Incertitudes Behind Bitcoin Mining. *CoRR*, abs/1310.7, 2013. URL http://arxiv.org/abs/1310.7935. 36

Tim Dalgleish, J. Mark G.. Williams, Ann-Marie J. Golden, Nicola Perkins, Lisa Feldman Barrett, Phillip J. Barnard, Cecilia Au Yeung, Victoria Murphy, Rachael Elward, Kate Tchanturia, and Edward Watkins. Verifiable Claims Data Model and Representations, 2017. URL https://w3c.github.io/vc-data-model/. 20

Samburaj Das. Parisian Engineering School Will Certify Diplomas on the Blockchain, 2016. URL https://www.ccn.com/parisian-engineering-school-will-certify-diplomas-blockchain/. 6

Christian Decker and Roger Wattenhofer. Information propagation in the Bitcoin network. In *13th IEEE International Conference on Peer-to-Peer Computing, IEEE P2P 2013 - Proceedings*, 2013. ISBN 9781479905218. doi: 10.1109/P2P.2013.6688704. 36

Tim Diacono. Maltese government wants to be pioneer in embracing blockchain and woo companies to the island, 2017. URL https://www.maltatoday.com.mt/business/business{_}news/79185/malta{_}lays{_}the{_}ground{_}for{_}a{_}blockchain{_}revolution{#}.W55Wwi3Oqu5. 6

Henning Diedrich. *Ethereum: Blockchains, Digital Assets, Smart Contracts, Decentralized Autonomous Organizations*. CreateSpace Independent Publishing, Scotts Valley:CA, 2016. 24

Cynthia Dwork and Moni Naor. Pricing via processing or combatting junk mail. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1993. ISBN 9783540573401. 35

J.H. Ellis. The possibility of secure non-secret digital encryption. *UK Communications Electronics Security Group*, 1970. 8

European Commission/EACEA/Eurydice. The European Higher Education Area in 2018: Bologna Process Implementation Report. *European Education*, 35(2):9–15, jul 2018. ISSN 1056-4934. doi: 10.2753/EUE1056-493435029. URL https://www.tandfonline.com/doi/full/10.2753/EUE1056-493435029. 2

Ittay Eyal and Emin Gün Sirer. Majority is not enough. *Communications of the ACM*, 61(7): 95–102, jun 2018. ISSN 00010782. doi: 10.1145/3212998. URL http://dl.acm.org/citation.cfm?doid=3234519.3212998. 42

Linux Fundation. HyperLedger, 2015. URL https://www.hyperledger.org. 63

Linux Fundation. Relatório Técnico da Hyperledger. Technical report, Linux Fundation, 2018. URL https://www.hyperledger.org/wp-content/uploads/2018/07/HL{_}Whitepaper{_}IntroductiontoHyperledger.pdf. 23, 64, 65, 66

Arthur Gervais, Ghassan O. Karame, Karl Wüst, Vasileios Glykantzis, Hubert Ritzdorf, and Srdjan Čapkun. On the security and performance of Proof of Work blockchains. In *Proceedings of the ACM Conference on Computer and Communications Security*, 2016. ISBN 9781450341394. doi: 10.1145/2976749.2978341. 44

Niv Gilboa. Two party RSA key generation. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1999. ISBN 3540663479. doi: 10.1007/3-540-48405-1_8. 58

Alexander Grech and Anthony F Camilleri. *Blockchain in Education*. Publications Office of the European Union, 2017. ISBN 978-92-79-73497-7. doi: 10.2760/60649. URL https://ec.europa.eu/jrc/en/open-education/legal-notice. vii, 3, 5, 6, 19, 21, 25, 29

Cyril Grunspan and Ricardo Pérez-Marco. On profitability of selfish mining. *CoRR*, abs/1805.0, 2018. URL http://arxiv.org/abs/1805.08281. 42

Nils Gura, Arun Patel, Arvinderpal Wander, Hans Eberle, and Sheueling Chang Shantz. Comparing elliptic curve cryptography and RSA on 8-Bit CPUs. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2004. ISSN 03029743. 11

Timo Hanke. AsicBoost - {A} Speedup for Bitcoin Mining. *CoRR*, abs/1604.0, 2016. URL http://arxiv.org/abs/1604.00575. 41

Darrel Hankerson, Scott Vanstone, and Alfred Menezes. *Guide to Elliptic Curve Cryptography*. Springer Publishing Company, Incorporated, 2004. doi: 10.1007/b97644. 10

Martin E Hellman and Diffie Whitfield. New Directions in Cryptography. *IEEE Transactions of Information Theory*, 1976. 8, 10

M Iansiti and K. R Lakhani. The Blockchain Revolution. *Harvard Business Review,*, pages 95(2), 20–20, 2017a. 24

Marco Iansiti and Karim R. Lakhani. The truth about blockchain, 2017b. ISSN 00178012. 24

IBGE. Rendimento médio real mensal do trabalho principal, por níveis de instrução, 2018. URL https://sidra.ibge.gov.br/tabela/5438. 1

Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography. In *Annual Symposium on Foundations of Computer Science (Proceedings)*, 1989. ISBN 0818619821. 10

IMS Global Learning Consortium. Open Badges v2.0 IMS Final Release, 2018. URL http://www.imsglobal.org/sites/default/files/Badges/OBv2p0Final/index.html. 51

A. Can Inci and Rachel Lagasse. Cryptocurrencies: applications and investment opportunities. *Journal of Capital Markets Studies*, ahead-of-p(ahead-of-print), sep 2019. ISSN 2514-4774. doi: 10.1108/JCMS-05-2019-0032. URL https://www.emerald.com/insight/content/doi/10.1108/JCMS-05-2019-0032/full/html. 48

Carlos Iragorri. Academic certificates on Hyperledger, 2018. URL https://hgf18.sched.com/event/G8rr/hyperledger. 6, 50, 54, 65

Chris Jagers. Badges and Blockcerts, 2019. URL https://www.learningmachine.com/badges-and-blockcerts/. 51

Marco Alberto Javarone and Craig Steven Wright. From Bitcoin to Bitcoin Cash. In *Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems - CryBlock'18*, pages 77–81, New York, New York, USA, 2018. ACM Press. ISBN 9781450358385. doi: 10.1145/3211933.3211947. URL http://dl.acm.org/citation.cfm?doid=3211933.3211947. 41

Belmiro N. João. Blockchain e o potencial de novos modelos de negócios: um mapeamento sistemático. *Revista de Gestão e Projetos*, 9(3), dec 2018. ISSN 2236-0972. doi: 10.5585/gep.v9i3.11121. URL http://periodicos.uninove.br/index.php?journal=gep{&}page=article{&}op=view{&}path[]=11121. 24

Don Johnson, Alfred Menezes, and Scott Vanstone. The Elliptic Curve Digital Signature Algorithm (ECDSA). *International Journal of Information Security*, 1(1):36–63, aug 2001. ISSN 1615-5262. doi: 10.1007/s102070100002. URL http://link.springer.com/10.1007/s102070100002. 14

Sunny King and Scott Nadal. PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake, 2012. URL https://decred.org/research/king2012.pdf. 38

Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48(177):203–203, jan 1987. ISSN 0025-5718. doi: 10.1090/S0025-5718-1987-0866109-5. URL http://www.ams.org/jourcgi/jour-getitem?pii=S0025-5718-1987-0866109-5. 11

Peter M. Krafft, Nicolás Della Penna, and Alex Sandy Pentland. An experimental study of cryptocurrency market dynamics. In *Conference on Human Factors in Computing Systems - Proceedings*, 2018. ISBN 9781450356206. doi: 10.1145/3173574.3174179. 45

Joshua a Kroll, Ian C Davey, and Edward W Felten. The Economics of Bitcoin Mining, or Bitcoin in the Presence of Adversaries. *The Twelfth Workshop on the Economics of Information Security (WEIS 2013)*, 2013. doi: June11-12,2013. 44

Krzysztof Okupski. Bitcoin Developer Reference. *Bitcoin.Org*, 2014. vii, 14, 37

Randhir Kumar and Rakesh Tripathi. Implementation of Distributed File Storage and Access Framework using IPFS and Blockchain. In *2019 Fifth International Conference on Image Information Processing (ICIIP)*, pages 246–251. IEEE, nov 2019. ISBN 978-1-7281-0899-5. doi: 10.1109/ICIIP47207.2019.8985677. URL https://ieeexplore.ieee.org/document/8985677/. 78

Yujin Kwon, Dohyun Kim, Yunmok Son, Eugene Vasserman, and Yongdae Kim. Be Selfish and Avoid Dilemmas: Fork after Withholding (FAW) attacks on bitcoin. In *Proceedings of the ACM Conference on Computer and Communications Security*, 2017. ISBN 9781450349468. doi: 10.1145/3133956.3134019. 41

Dmitrij Lagutin, Yki Kortesniemi, Nikos Fotiou, and Vasilios A. Siris. Enabling Decentralised Identifiers and Verifiable Credentials for Constrained IoT Devices using OAuth-based Delegation. In *Proceedings 2019 Workshop on Decentralized IoT Systems and Security*, Reston, VA, 2019. Internet Society. ISBN 1-891562-56-8. doi: 10.14722/diss.2019.23005. URL https://www.ndss-symposium.org/wp-content/uploads/diss2019{_}05{_}Lagutin{_}paper.pdf. 20

Leslie Lamport. Constructing Digital Signatures from a One-Way Function. *SRI International Computer Science Laboratory*, 1979. 14

Leslie Lamport, Robert Shostak, and Marshall Pease. The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 1982. ISSN 15584593. doi: 10.1145/357172.357176. 34

Tin Leelavimolsilp, Long Tran-Thanh, and Sebastian Stein. On the Preliminary Investigation of Selfish Mining Strategy with Multiple Selfish Miners. *CoRR*, abs/1802.0, 2018. URL `http://arxiv.org/abs/1802.02218`. 42

M. L. MIT Media Lab. Blockcerts-An Open Infrastructure for Academic Credentials on the Blockchain, 2016. URL `https://hgf18.sched.com/event/G8rr/hyperledger`. 4, 49

Dindayal Mahto and Dilip Kumar Yadav. RSA and ECC: A comparative analysis. *International Journal of Applied Engineering Research*, 2017. ISSN 09739769. 11

Dindayal Mahto and Dilip Kumar Yadav. Performance Analysis of RSA and Elliptic Curve Cryptography. *International Journal of Network Security*, 20(4):625–635, 2018. doi: 10.6633/IJNS.201807_20(4).04. URL `doi.org/10.6633/IJNS.201807{_}20(4).04`. vii, 11, 12

Ralph C. Merkle. A digital signature based on a conventional encryption function. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1988. ISBN 9783540187967. doi: 10.1007/3-540-48184-2_32. 16

Ralph C. Merkle. A Certified Digital Signature. In *Advances in Cryptology — CRYPTO' 89 Proceedings*, pages 218–238. Springer New York, New York, NY, 1990. doi: 10.1007/0-387-34805-0_21. URL `http://link.springer.com/10.1007/0-387-34805-0{_}21`. 14

Victor S. Miller. Use of Elliptic Curves in Cryptography. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1986. ISBN 9783540164630. doi: 10.1007/3-540-39799-X_31. 11

Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. *Www.Bitcoin.Org*, 2008. ISSN 09254560. doi: 10.1007/s10838-008-9062-0. 4, 23, 35

Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. *Bitcoin and Cryptocurrency Technologies Introduction to the book*. Princeton University Press, 2016. ISBN 9781400884155. doi: 10.1057/palgrave.jit.2000080. vii, 15

NEO. The dBFT Algorithm, 2014. URL `https://docs.neo.org/developerguide/en/articles/consensus/consensus{_}algorithm.html`. 39

Miquel Oliver, Joan Moreno, Gerson Prieto, and David Ben\'\itez. Using blockchain as a tool for tracking and verification of official degrees: business model. In *Using blockchain as a tool for tracking and verification of official degrees: business model*, 29th European Regional

Conference of the International Telecommunications Society (ITS): "Towards a digital future: Turning technology into markets?", Trento, Italy, 1st - 4th August 2018, Trento, 2018. International Telecommunications Society (ITS). URL http://hdl.handle.net/10419/184958. 2, 51, 52, 54

Lucas M. Palma, Martín A. G. Vigil, Fernando L. Pereira, and Jean E. Martina. Blockchain and smart contracts for higher education registry in Brazil. *International Journal of Network Management*, 29(3):e2061, may 2019. ISSN 1055-7148. doi: 10.1002/nem.2061. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nem.2061. 29, 79

CSN platform. MIT to Issue Diplomas using Bitcoin Blockchain, 2017. URL https://medium.com/@csn{_}platform/mit-to-issue-diplomas-using-bitcoin-blockchain-428a04a63f5. 49

A Rajalakshmi, K.V Lakshmy, M Sindhu, and P Amritha. A Blockchain and IPFS based framework for secure Research record keeping. *International Journal of Pure and Applied Mathematics*, 119(15), 2018. URL https://acadpubl.eu/hub/2018-119-15/4/751.pdf. 78

Drummond Reed, Manu Sporny, Dave Longley, Christopher Allen, Ryan Grant, and Markus Sabadello. Decentralized Identifiers (DIDs) v0.11, 2019. URL https://w3c-ccg.github.io/did-spec/. 20

Tatiana Revoredo and Rodrigo Borges. Blockchains e as leis de proteção de dados: incompatíveis ?, 2018. URL https://politica.estadao.com.br/blogs/fausto-macedo/blockchains-e-as-leis-de-protecao-de-dados-incompativeis/. 28

R. L. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, 1983. ISSN 15577317. doi: 10.1145/357980.358017. 10

Anthony Ronning and Wong Wai Chung. Blockcerts V3 Proposal, 2019. URL https://nbviewer.jupyter.org/github/WebOfTrustInfo/rwot9-prague/blob/master/final-documents/BlockcertsV3.pdf. 75, 79, 100

Li Rujia and David Galind. BTCert, 2017. URL https://github.com/BlockTechCert/BTCert. vii, 58, 60, 61, 78

Russel. Sony wants to digitize education records using the blockchain., 2017. URL https://techcrunch.com/2017/08/09/sony-education-blockchain. 6

Muhammad Saad, My T Thai, and Aziz Mohaisen. POSTER: Deterring DDoS Attacks on Blockchain-based Cryptocurrencies Through Mempool Optimization. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, ASIACCS '18, pages 809–811, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5576-6. doi: 10.1145/3196494.3201584. URL http://doi.acm.org/10.1145/3196494.3201584. 43

Muhammad Saad, Jeffrey Spaulding, Laurent Njilla, Charles A Kamhoua, Sachin Shetty, DaeHun Nyang, and Aziz Mohaisen. Exploring the Attack Surface of Blockchain: {A} Systematic Overview. *CoRR*, abs/1904.0, 2019. URL http://arxiv.org/abs/1904.03487. vii, x, 40, 41, 42, 43, 44, 46

Joao Santos and Kim Hamilton Duffy. A Decentralized Approach to Blockcerts Credential Revocation, 2019. URL https://github.com/WebOfTrustInfo/rwot5-boston/blob/master/final-documents/blockcerts-revocation.md. 77, 79

J. Philipp Schmidt. Credentials, Reputation, and the Blockchain, 2017. URL http://er.educause.edu/articles/2017/4/credentials-reputation-and-the-Blockchain. 19, 21

Klaus Schwab. *The Fourth Industrial Revolution*. World Economic Forum, New York, 2017. 24

Mike Sharples and John Domingue. The blockchain and kudos: A distributed system for educational record, reputation and reward. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 9891 LNCS, pages 490–496, 2016. ISBN 9783319451527. doi: 10.1007/978-3-319-45153-4_48. 25

David Siegel. Understanding The DAO Attack, 2016. URL https://www.coindesk.com/understanding-dao-hack-journalists. 41, 43

Soram Ranbir Singh, Ajoy Kumar Khan, and Soram Rakesh Singh. Performance evaluation of RSA and Elliptic Curve Cryptography. In *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, pages 302–306. IEEE, dec 2016. ISBN 978-1-5090-5256-1. doi: 10.1109/IC3I.2016.7917979. URL http://ieeexplore.ieee.org/document/7917979/. 11

Natalie Smolenski and Dan Hughes. *Academic Credentials in an Era of Digital Decentralization*. Learning Machine Research, Anaheim, CA, 2016. URL https://www.academia.edu/29403234/Academic{_}Credentials{_}in{_}an{_}Era{_}of{_}Digital{_}Decentralization. 18

Sony. Global Education Develops Technology Using Blockchain for Open Sharing of Academic Proficiency and Progress Records, 2016. URL https://www.sony.net/SonyInfo/News/Press/201602/16-0222E/index.html. 6

Joao Sousa, Alysson Bessani, and Marko Vukolic. A Byzantine Fault-Tolerant Ordering Service for the Hyperledger Fabric Blockchain Platform. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 51–58. IEEE, jun 2018. ISBN 978-1-5386-5596-2. doi: 10.1109/DSN.2018.00018. URL https://ieeexplore.ieee.org/document/8416470/. 38

Tom St Denis and Simon Johnson. Hash Functions. In *Cryptography for Developers*, pages 203–250. Elsevier, 2007. doi: 10.1016/B978-159749104-4/50008-X. URL https://linkinghub.elsevier.com/retrieve/pii/B978159749104450008X. 15

Melanie Swan. *Blockchain - Blueprint for a new economy*. O'Reilly Media, Inc., 2015. ISBN 9781119371236. doi: 10.1109/CANDAR.2017.50. 24

Nick Szabo. Formalizing and securing relationships on public networks. *First Monday*, 1997. ISSN 13960466. doi: 10.5210/fm.v2i9.548. 47

Don Tapscott and Alex Tapscott. Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money ... *Sage Publications, Inc.*, 2016. ISSN 0717-6163. 49

Simon Taylor. Blockchain : understanding the potential Blockchain : understanding the potential. *Barclays*, 2015. 24

Muhamed Turkanović, Marko Hölbl, Kristjan Košič, Marjan Heričko, and Aida Kamišalić. EduCTX: A blockchain-based higher education credit platform. *IEEE Access*, 2018. ISSN 21693536. doi: 10.1109/ACCESS.2018.2789929. viii, 4, 6, 23, 25, 62

Fakhar ul Hassan, Anwaar Ali, Siddique Latif, Junaid Qadir, Salil Kanhere, Jatinder Singh, and Jon Crowcroft. Blockchain And The Future of the Internet: {A} Comprehensive Review. *CoRR*, abs/1904.0, 2019. URL http://arxiv.org/abs/1904.00733. 38, 43

European Union. General Data Protection Regulation, 2018. URL https://eugdpr.org. 27

Pavel Vasin. BlackCoin's Proof-of-Stake Protocol v2, 2014. URL https://blackcoin.org/blackcoin-pos-protocol-v2-whitepaper.pdf. 37

Fernando Vidal, Feliz Gouveia, and Christophe Soares. Analysis of Blockchain Technology for Higher Education. In *2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pages 28–33. IEEE, oct 2019. ISBN 978-1-7281-2542-8. doi: 10.1109/CyberC.2019.00015. URL https://ieeexplore.ieee.org/document/8945824/. 1, 77

Fernando Richter Vidal, Feliz Gouveia, and Christophe Soares. Revocation Mechanisms for Academic Certificates Stored on a Blockchain. In *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–6, Sevilha, jun 2020. IEEE. ISBN 978-989-54659-0-3. doi: 10.23919/CISTI49556.2020.9141088. URL https://ieeexplore.ieee.org/document/9141088/. viii, 76, 77, 78, 79

Martín Vigil, Johannes Buchmann, Daniel Cabarcas, Christian Weinert, and Alexander Wiesmaier. Integrity, authenticity, non-repudiation, and proof of existence for long-term archiving: A survey. *Computers & Security*, 50:16–32, may 2015. ISSN 01674048. doi: 10.1016/j.cose.2014.12.004. URL https://linkinghub.elsevier.com/retrieve/pii/S0167404814001849. 12, 14

Marko Vukolić. Rethinking permissioned blockchains. In *BCC 2017 - Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts, co-located with ASIA CCS 2017*, 2017. ISBN 9781450349741. doi: 10.1145/3055518.3055526. 65

W3C. Merkle Proof Signature Suite 2017, 2017. URL `https://w3c-dvcg.github.io/lds-merkleproof2017/`. 96

Mark Walport. Distributed ledger technology: Beyond blockchain. *Government Office for Science*, 2016. 40

Shangping Wang, Yinglong Zhang, and Yaling Zhang. A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems. *IEEE Access*, 2018. ISSN 21693536. doi: 10.1109/ACCESS.2018.2851611. 77, 78

Gavin Wood. Ethereum: A Secure Decentralised Generalised Transaction Ledger. EIP-150 REVISION. *2017*, 2017. 32

Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. In *Proceedings - 2017 IEEE 6th International Congress on Big Data, BigData Congress 2017*, 2017. ISBN 9781538619964. doi: 10.1109/BigDataCongress.2017.85. x, 4, 26, 27, 35, 38, 40

X Zhou, Q Wu, B Qin, X Huang, and J Liu. Distributed Bitcoin Account Management. In *2016 IEEE Trustcom/BigDataSE/ISPA*, pages 105–112, 2016. doi: 10.1109/TrustCom.2016.0052. 59

Apêndice A

Algoritmo raiz de *Merkle*

Algorithm 1: Cálculo da raiz de *Merkle* do *Bitcoin*

Entrada: L `// Lista dos identificadores (condensados) das transações contidas no bloco`
Saída : S `// Texto que representa a raiz de Merkle do bloco`

```
1  Function ArvoreMerkle (L):
2      i ← 0
3      foreach txid ∈ L do
4          Li[i] ← FormataBigEndian(txid)
5          i++;
6      end
7      raiz ← ConstruirArvore(Li)
8      raiz ← FormataBigEndian(raiz)
9      return raiz
10 Function FormataBigEndian (txid) :
11     tamanho ← length of txid
12     for i ← 0 to tamanho by 2 do
13         txidRet ← txid[i+1] ++ txid[i]
14     end
15     tamanho ← length of txidRet − 1
16     while tamanho ≥ 0 do
17         reverso ← reverso ++ txidRet[tamanho]
18         tamanho − −
19     end
20     return reverso
21 Function ConstruirArvore (L) :
22     if L ≡ ∅ then
23         return ∅
24     end
25     ultimoNo ← length of L
26     if ultimoNo ≡ 1 then
27         return L[0]
28     end
29     if (ultimoNo (mod 2)) ≫ 0 then
30         L[ultimoNo ⊕ 1] ← L[ultimoNo]   // Árvore não está balanceada, é necessário replicar o último
           nó para formar o par faltante
31     end
32     ultimoNo ← length of L
33     for i ← 0 to ultimoNo by 2 do
34         folhaPares = L[i] ++ L[i + 1]
35         L′[i] ← SHA256(folhaPares)
36     end
37     return ConstruirArvore(L′)                    // Chamada recursiva até atingir o nó raiz
38
```

Apêndice B

Análise do bloco 110041

Neste anexo estende-se a discussão sobre blocos e transações, através da análise de alguns blocos e transações de operações reais que ocorreram na rede oficial da *Bitcoin*. Existem algumas ferramentas que permitem explorar os blocos da *Bitcoin*, representando de forma gráfica as informações contidas no bloco, como por exemplo: *Block Explorer* [1], *Blockchain Info* [2], *BlockCypher* [3],*BTC* [4], *BlockTrail* [5]. Neste trabalho, foi utilizado a ferramenta *Block Explorer*.

O bloco número 110041 é consultado e apresentado através da figura B.1. Como pode-se notar, a imagem apresenta as informações de cabeçalho como o condensado do bloco anterior, o número de transações contidas no bloco, e quantas confirmações foram feitas. E também percebe-se as transações que foram realizadas pelo bloco. No exemplo nota-se que a primeira transação deste bloco, é a recompensa dada ao endereço público do minerador em 50 *Bitcoin*s, por ter conseguido resolver um problema matemático, que resultou na criação de um novo bloco.

[1] https://blockexplorer.com/
[2] https://www.blockchain.com/explorer
[3] https://live.blockcypher.com/btc
[4] https://btc.com
[5] https://www.blocktrail.com

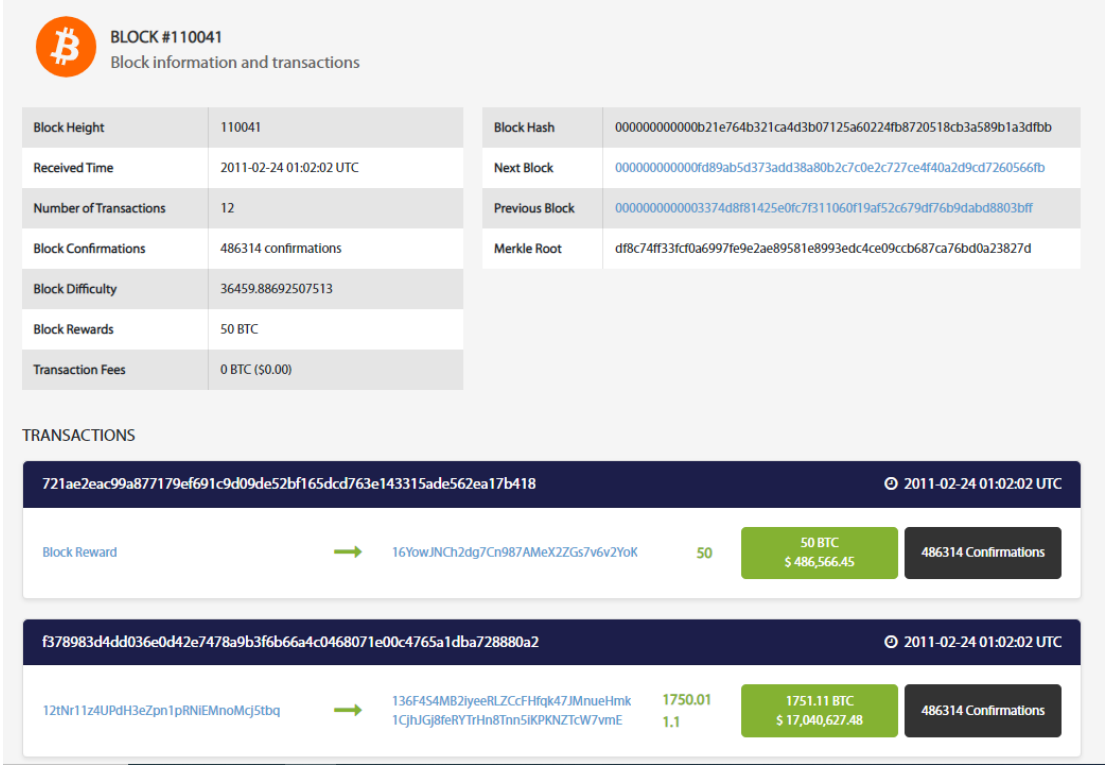

Figura B.1: *Bloco 110041, na rede oficial da Bitcoin (mainnet)*

A figura B.2 detalha uma transação contida no bloco 110041. No exemplo nota-se: o tamanho da transação (259 *bytes*), a taxa de pagamento que não existiu (sendo demonstrada como 0), qual foi a data e hora da transação (24/02/2011 01:02:02), e os totais de entradas e saídas em *Bitcoins* (1.751,11 valor estimado em 15 milhões de dólares, nas conversões atuais).

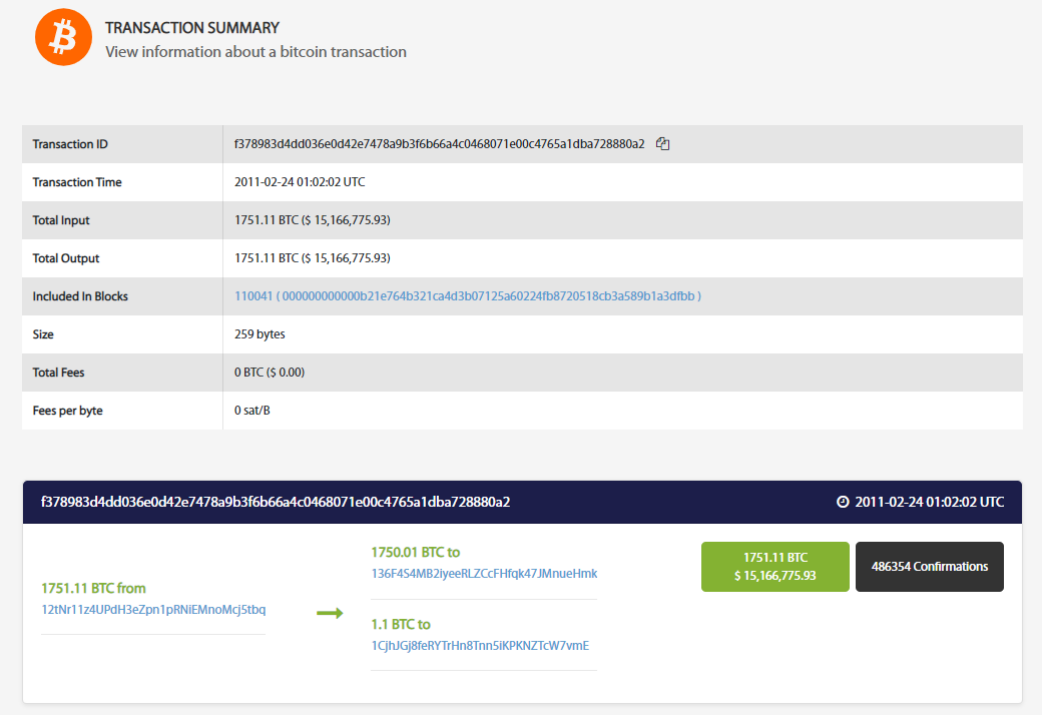

Figura B.2: *Transação do bloco 110041, na rede oficial da Bitcoin (mainnet)*

A última análise sobre o bloco 110041 se dá pela maneira como foi calculada a raiz de *Merkle*. Antes de apresentar o algoritmo, é necessário relatar que os identificadores das transações contidas nos blocos de *Bitcoin* estão representados no formato *little-endian* e precisam ser convertidos para *big-endian*. Esses formatos referem-se como os computadores armazenam os *bytes*.

Nos condensados das transações, os octetos são revertidos quando convertidos para a forma hexadecimal. Isso faz com que os blocos apareçam na forma *big-endian*, que é a maneira como os humanos escrevem os números. Por essa razão é necessário considerar esse passo de conversão adicional para o cálculo da raiz de *Merkle*, que é ilustrado pelo algoritmo no apêndice A:

Seguindo a sequência do algoritmo, para calcular a raiz de *Merkle* do bloco 110041 é necessário seguir os seguintes passos:

O primeiro passo para o cálculo utiliza dados da tabela B.1, na qual estão representadas as informações das 12 transações do bloco. A tabela também apresenta as informações convertidas para o padrão *big-endian*.

Tabela B.1: *Lista das transações do bloco 110041 e conversão para formato big-endian*

Txid	**Txid *big-endian***
721ae2eac99a877179ef691c9d09de52bf165dcd763e143315ade562ea17b418	18b417ea62e5ad1533143e76cd5d16bf52de099d1c69ef7971879ac9eae21a72
f378983d4dd036e0d42e7478a9b3f6b66a4c0468071e00c4765a1dba728880a2	a2808872ba1d5a76c4001e0768044c6ab6f6b3a978742ed4e036d04d3d9878f3
43e6da95ee91e170097712d4a2e57f607004f3f744ed1303404a6a86b92c8cd5	d58c2cb9866a4a400313ed44f7f30470607fe5a2d412770970e191ee95dae643
c78900056b8c172639d3d30505111a2d8a5a7f9008cef2992a4da1daea2dc7d9	d9c72deadaa14d2a99f2ce08907f5a8a2d1a110505d3d33926178c6b050089c7
d656934f05220753ede3c0f95366d2ec0ae536b07498949087a0a95e71860dd9	d90d86715ea9a08790949874b036e50aecd26653f9c0e3ed530722054f9356d6
fd1f4cc46346709f0bf3131705d00b7f1b9f26c82b6a09f9496de7695e6d6384	84636d5e69e76d49f9096a2bc8269f1b7f0bd0051713f30b9f704663c44c1ffd
c9b8471609b4fa24b5a551b73ff3f1c802246ffff99d3a28b55a6c2d92501719	191750922d6c5ab5283a9df9ff6f2402c8f1f33fb751a5b524fab4091647b8c9
688986f3c33a0ec96c906342943fba8a7fc68dab740cda14f30e135b206feb90	90eb6f205b130ef314da0c74ab8dc67f8aba3f944263906cc90e3ac3f3868968
f6049f23939fca57e5704fcad369554bb5419d61c535a0ad507db7dda3124239	394212a3ddb77d50ada035c5619d41b54b5569d3ca4f70e557ca9f93239f04f6
f19a73569c6ad351d6135f93cee9b013a89995cc38d5390ccb866c55be2b8e71	718e2bbe556c86cb0c39d538cc9599a813b0e9ce935f13d651d36a9c56739af1
e623e778fddf54ac0e49f1369913648501d3264debb6f630daffa761d592013d	3d0192d561a7ffda30f6b6eb4d26d3018564139936f1490eac54dffd78e723e6
4a0a152c6770cfa0eac216cef63de7d75030bcb6e0e53ec187642346cad1288a	8a28d1ca46236487c13ee5e0b6bc3050d7e73df6ce16c2eaa0cf70672c150a4a

A tabela B.2 demonstra a primeira iteração do algoritmo, em que um primeiro agrupamento em pares de condensados é realizado. Após o processo de concatenar, o resultado é convertido em um vetor de 64 *bytes* que será novamente submetido a função de condensado.

Tabela B.2: *Lista de pares das transações do bloco 110041 na primeira iteração do algoritmo*

Condensado de pares
18b417ea62e5ad1533143e76cd5d16bf52de099d1c69ef7971879ac9eae21a72 ∥ a2808872ba1d5a76c4001e0768044c6ab6f6b3a978742ed4e036d04d3d9878f3
d58c2cb9866a4a400313ed44f7f30470607fe5a2d412770970e191ee95dae643 ∥ d9c72deadaa14d2a99f2ce08907f5a8a2d1a110505d3d33926178c6b050089c7
d90d86715ea9a08790949874b036e50aecd26653f9c0e3ed530722054f9356d6 ∥ 84636d5e69e76d49f9096a2bc8269f1b7f0bd0051713f30b9f704663c44c1ffd
191750922d6c5ab5283a9df9ff6f2402c8f1f33fb751a5b524fab4091647b8c9 ∥ 90eb6f205b130ef314da0c74ab8dc67f8aba3f944263906cc90e3ac3f3868968
394212a3ddb77d50ada035c5619d41b54b5569d3ca4f70e557ca9f93239f04f6 ∥ 718e2bbe556c86cb0c39d538cc9599a813b0e9ce935f13d651d36a9c56739af1
3d0192d561a7ffda30f6b6eb4d26d3018564139936f1490eac54dffd78e723e6 ∥ 8a28d1ca46236487c13ee5e0b6bc3050d7e73df6ce16c2eaa0cf70672c150a4a

O resultado é novamente colocado em pares, como mostra a tabela B.3

Tabela B.3: *Lista de pares das transações do bloco 110041 na segunda iteração do algoritmo*

Condensado de pares
D4EF3884D3CE70E97F6E37EB21B9416FC4A0381CAF7264683427C30B55FA68D0 ∥ 67FB6A932B9FAB96C976C4A7D925B1A39EA432C54C5B62C524D00F2C869B7573
E634804FD89595A26C4FF88346EF21427A8A51D4CC59A6D213591FF9A81F568C ∥ B6798EE991086FDF8C539C91DCE561AB4F3794567E3F539B24BEBC5AAAFA0E2A
20861FE436D89320894BC72BA1EC3E68B94B4126F8262CD4C4607089640444F1 ∥ CA163B63FEF53D7A9852198B37003618DF2C4AF7E0A576A3401D240E2254D514

O resultado é novamente colocado em pares como mostra a tabela B.4

Tabela B.4: *Lista de pares das transações do bloco 110041 na terceira iteração do algoritmo*

Condensado de pares
13BBD1A4A80C74455C732DF2A4567A562F6BC26D3D3101AFF1E35E50B54DD44C + 4B7125DC552382DF6A27676CA3C219DF4222DD278065E14141FC1087D48A2000
07128A33CBEF7507AD7BF281065289C942CA7C082E6FB9B3BC36666CE714B1EB + 07128A33CBEF7507AD7BF281065289C942CA7C082E6FB9B3BC36666CE714B1EB

A última iteração de pares, demonstrado pela tabela B.5

Tabela B.5: *Lista de pares das transações do bloco 110041 na quarta iteração do algoritmo*

Condensado de pares
3AED1D4757C62B9A029519C625DA7F2A81423DB84511D4A852531C1A13663830 + 819249E3B2FAFEB8E1503AB3A7F62EF129750574FA09DEDB288D4E8A242C0C28

No final do processamento encontra-se o valor no formato *little-endian*: $7d82230abd76ca87b6cc09cec4ed93891e5889aee2e97f99a6f0fc33fff748cdf$. Novamente executando a função para conversão do formato para *big-endian*, encontra-se o valor: $df8c74ff33fcf0a6997fe9e2ae89581e8993edc4ce09ccb687ca76bd0a23827d$, exatamente o mesmo do campo raiz de *Merkle* representado na figura B.1