SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PARÁ
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

BRENO LANDRY SILVA GOMES
CAROLINA LOPES MENDES

AUTENTICAÇÃO E REGISTRO DE DOCUMENTOS OFICIAIS ATRAVÉS DE UMA REDE *BLOCKCHAIN*

BELÉM
2018

BRENO LANDRY SILVA GOMES
CAROLINA LOPES MENDES

AUTENTICAÇÃO E REGISTRO DE DOCUMENTOS OFICIAIS ATRAVÉS DE UMA REDE *BLOCKCHAIN*

Trabalho de Conclusão de Curso apresentado ao Instituto Federal de Educação, Ciência e Tecnologia do Pará - IFPA – Campus Belém e como requisito para a obtenção de Grau em Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Me. Márcio Góes do Nascimento.

BELÉM
2018

BRENO LANDRY SILVA GOMES
CAROLINA LOPES MENDES

AUTENTICAÇÃO E REGISTRO DE DOCUMENTOS OFICIAIS ATRAVÉS DE UMA REDE *BLOCKCHAIN*

Trabalho de Conclusão de Curso apresentado ao Instituto Federal de Educação, Ciência e Tecnologia do Pará - IFPA – Campus Belém e como requisito para a obtenção de Grau em Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Me. Márcio Góes do Nascimento.

Data da Defesa: 09/11/2018
Conceito: 8,0

_____
Orientador: Prof. Me. Márcio Góes do Nascimento.
Instituto Federal do Pará – Campus Belém

_____
Prof. Me. Cláudio Roberto de Lima Martins
Instituto Federal do Pará – Campus Belém

_____
Prof. Me. Ricardo José Souza de Cabeça
Instituto Federal do Pará – Campus Belém

Aos nossos familiares.

AGRADECIMENTOS

Agradeço todos que estiveram presentes durante essa caminhada acadêmica e que de alguma forma contribuíram para a execução deste trabalho.

Breno Landry Silva Gomes

Agradeço primeiramente a Deus, por mais esta conquista, toda a sabedoria vem d'Ele.

À minha família, que durante toda minha vida me incentivou e trabalhou para que eu concluísse mais esta etapa, em especial, minha mãe Sunamita da Paixão Lopes, por seu apoio e amor incondicional e à minha avó Eduarda Portal e meu irmão, Matheus Eduardo pela ajuda e compreensão sempre que eu precisava de mais tempo para me dedicar aos estudos.

Aos meus queridos amigos, por seu apoio, compreensão e ajuda sempre que foi preciso.

Aos meus professores, que enriqueceram minha experiência de aprendizagem nestes anos de graduação.

Carolina Lopes Mendes

“Nós só podemos ver um pouco do futuro, mas o suficiente para perceber que há muito a fazer”.

Alan Turing

RESUMO

Por muito tempo, a sociedade brasileira autenticou, emitiu, e armazenou os documentos oficiais através dos cartórios, ou no caso das instituições de ensino superior (IES), que são responsáveis por emitir e validar seus próprios documentos oficiais. Porém, com as inovações causadas pela tecnologia da informação, fez-se necessário um sistema que pudesse englobar todas as características e necessidades de um sistema de autenticação digital de documentos oficiais, e que proporcione a validação deste documento no ambiente on-line e off-line. O objetivo deste trabalho é apresentar uma opção de sistema que satisfaça esses requisitos, para isso, foi projetado um sistema de autenticação de documentos oficiais através de uma rede *Blockchain*, utilizando todos os recursos do sistema, através da plataforma de desenvolvimento Ethereum. Para estipular todos os termos e dados inseridos na aplicação, utilizamos os Smart Contracts, que foram implementados na linguagem Solidity, e para fazer a conexão entre o Smart Contract e o Ethereum, foi utilizado a API Web3.js. Além desta API, utilizamos um provedor Ethereum, ou seja, uma carteira eletrônica (eWallet), a *Metamask*, uma extensão do navegador Google Chrome ou Firefox. Esperamos oferecer mais uma opção segura, efetiva e ágil para a validação e emissão de documentos no Instituto Federal de Educação, Ciência e Tecnologia do Pará (IFPA).

Palavras-chave: *Blockchain.* Validação de Documentos. Ethereum. Contratos Inteligentes.

ABSTRACT

For a long time, Brazilian society has authenticated, issued, and stored the official documents through the notaries, or in the case of higher education institutions (HEIs), which are responsible for issuing and validating their own official documents. However, with the innovations caused by information technology, it became necessary a system that could encompass all the characteristics and needs of a system of digital authentication of official documents, and that provides the validation of this document in the online environment and off- line. The purpose of this paper is to present a system option that satisfies these requirements. For this purpose, an official document authentication system was designed through a *Blockchain* network, using all system resources, through the Ethereum development platform. To establish all terms and data entered in the application, we used the Smart Contracts, which were implemented in Solidity language, and to make the connection between Smart Contract and Ethereum, we used the API Web3.js. In addition to this API, we use an Ethereum provider, ie an eWallet, Metamask, a browser extension Google Chrome or Firefox. We hope to offer another safe, effective and agile option for the validation and issuance of documents at the Federal Institute of Education, Science and Technology of Pará (IFPA).

Keywords: *Blockchain*. validation of documents. Ethereum. Smart Contracts.

LISTA DE ILUSTRAÇÃO

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

A autenticação de documentos oficiais sempre foi uma questão muito importante na história brasileira. A necessidade de comprovar a veracidade de algum acordo feito entre as diversas partes de um negócio, de modo que seu conteúdo e a data fossem irrefutáveis estão presentes desde os tempos coloniais, quando Portugal, no intuito de melhor administrar suas terras na América, criou o sistema de capitanias hereditárias, onde dividiu o território brasileiro em 12 partes, cada uma essas partes foi atribuída a um capitão donatário, que tinha plenos poderes para demarcar as terras, distribuir as sesmarias, e administrar do jeito que melhor lhe aprouver. Para isto, muitos cargos foram instituídos, dentre eles, o de tabelião (SOUZA PINTO, 2014).

Nos dias atuais, com o surgimento de uma série de novas tecnologias, dentre elas a internet, que revolucionou a maneira como os documentos são emitidos, armazenados e transmitidos, agilizando os processos e poupando espaço e tempo. Porém, mesmo com todas essas inovações, a questão da autenticação e legalidade de documentos gerados em ambientes virtuais, assim como a preservação e armazenamento desses documentos, é um tema que ainda gera certa preocupação, pois são poucos os meios para atestar a veracidade e sigilo destes documentos (DORNELES, CORRÊA, 2013,).

Em 2008, o grupo de discussão *The Cryptography Mailing,* recebeu um artigo técnico com os fundamentos de uma criptomoeda chamada “Bitcoin”, criada em uma rede peer-to-peer descentralizada e com um sistema de segurança também descentralizado e organizado em blocos, onde as transações, depois de validadas não poderiam ser mudadas ou separadas do bloco. Este artigo, escrito por Satoshi Nakamoto (2008), um pseudônimo do idealizador anônimo da tecnologia, tinha por objetivo uma criptomoeda que fosse utilizada mundialmente, através da qual, transações financeiras online poderiam ser feitas sem a mediação de nenhum tipo de instituição financeira, portanto, um sistema mais direto, distribuído, seguro e acessível (FORMIGONI FILHO; BRAGA; LEAL, 2017).

Uma das razões práticas para que este modelo de transação fosse criada, foi a necessidade de um sistema de transações financeiras que não estivesse baseado em confiança, tanto por parte do vendedor, quanto por parte do comprador, e sim

baseado em um algoritmo para proporcionar uma garantia maior para o vendedor, em casos onde a compra é efetuada e o produto enviado, e logo após é pedido o estorno, configurando assim uma fraude. E também como uma segurança para o comprador, que não necessitará fornecer tantas informações acerca de si mesmo para preencher os requisitos de confiança que o vendedor necessitaria em um sistema tradicional (NAKAMOTO, 2008).

Com o passar dos anos e o crescente uso dos *bitcoins*, a base de dados no qual a criptomoeda se baseia, o *Blockchain*, foi atraindo a atenção de muitas empresas e pesquisadores da área de Tecnologia da Informação e Comunicação (TIC).Muitas de suas características, como segurança, resiliência, imutabilidade e inviolabilidade mostraram-se de grande valia para o mercado tecnológico que previu uma gama de muitas outras aplicações e possibilidades em que o *Blockchain* poderia ser utilizado, não só no mercado financeiro, mas também no mercado logístico, indústria alimentar, entretenimento entre outros (FORMIGONI FILHO; BRAGA; LEAL, 2017).

1.2 Objetivos

Neste trabalho, pretende-se desenvolver um protótipo funcional de um sistema para autenticação e registro de documentos oficiais para instituições de ensino, como o IFPA, onde documentos como diplomas, históricos, declarações de vínculo possam ser registrados e autenticados em uma rede *Blockchain*, para que a própria instituição possa ter mais segurança, agilidade em emitir, receber e registar seus documentos oficiais.

Optou-se por utilizar como exemplo para o protótipo funcional o registro de diplomas, onde as informações necessárias para a emissão do diploma serão cadastradas na base de dados *Blockchain* por meio de um Smart Contract (Contrato Inteligente). Os métodos de registro e consulta, ficarão codificados neste Contrato Inteligente, que será previamente implementado na rede.

Com isso, supõe-se que, utilizar o *Blockchain,* seria uma boa solução para conter as fraudes de documentos escolares, proporcionar mais segurança na emissão, armazenamento e verificação de autenticidade dos documentos oficiais da instituição.

Para elaborar este trabalho, foram feitas pesquisas nos mais recentes artigos publicados sobre o *Blockchain* e suas diversas características, funcionalidades e possibilidades de aplicação. Para o desenvolvimento da aplicação que põe em prática a proposta do trabalho, utilizamos a plataforma *Ethereum,* pois na pesquisa que foi realizada, a plataforma *Ethereum* possui as características mais apropriadas para o desenvolvimento desta aplicação.

Este trabalho está dividido da seguinte maneira. O capítulo 1 é a introdução do trabalho; no capítulo 2, será exposto o *Blockchain com* todas as suas características e mecanismos; no capítulo 3 será indicada a importância da certificação digital de documentos; no capítulo 4, será abordada a construção da aplicação com o *Blockchain*; e no capítulo 5, a conclusão do trabalho.

2 *BLOCKCHAIN*

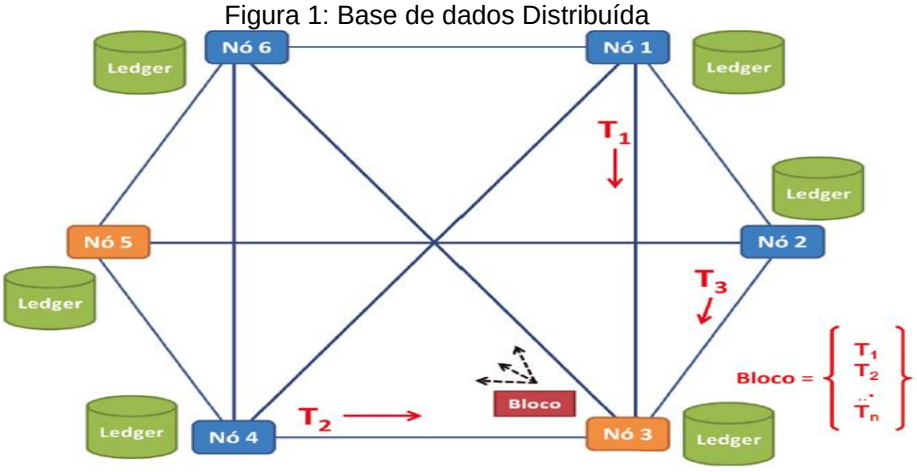
Neste capítulo, será definido e detalhado o funcionamento da tecnologia *Blockchain* na prática, seu mecanismo de consenso, detalhes técnicos e como são realizadas as transações de modo mais seguro possível.

Como já citado, a *Blockchain* nada mais é do que uma base de dados distribuída, que é compartilhada pelos nós de um sistema distribuído que se organiza em uma rede *peer-to-peer*. Cada registro nesta base de dados se chama “bloco”. Nestes blocos estão os registros de todas as transações que ocorreram dentro dele, desde a sua criação, até a mais recente atualização, sendo que, toda a rede, aceita somente a inserção de novos blocos, sendo proibida a exclusão ou a modificação de algum desses blocos (BRAGA, 2017).

A impossibilidade de exclusão e modificação de um bloco já inserido na rede, tornou-se um dos mecanismos para evitar o problema do gasto duplo na rede. De acordo com NAKAMOTO (2008), na rede *Blockchain*, todas as transações são públicas e, depois de validadas e distribuídas pela rede, podem ser verificadas por todos, não podendo ser modificadas ou excluídas de forma alguma, providenciando assim, um histórico completo de todas as transações para todos os nós da rede, dificultando qualquer tentativa de gasto duplo nas transações.

O *Blockchain* é mantido por toda a rede *peer-to-peer* (NAKAMOTO,2008), sem que haja um local principal para o armazenamento e dados na rede, todos os nós têm uma cópia e toda a base de dados, que se mantém íntegra e sincronizada através dos protocolos de consenso (BRAGA, 2017).

Figura 1: Base de dados Distribuída



Fonte: Braga, 2017, p.3.

2.1 Transações

A estrutura das transações dentro da *Blockchain* se organiza de maneira que as transações possam ser totalmente públicas e transparentes e com o intuito de prevenir fraudes, roubos e gastos duplos dentro da rede. Segundo BRAGA (2017) no caso das criptomoedas, a estrutura das transações assemelha-se com balancete de débito e crédito, e são compostas dos seguintes componentes.

Figura 2: Conteúdo de uma transação

Fonte: Braga; Marino; Santos,2017.

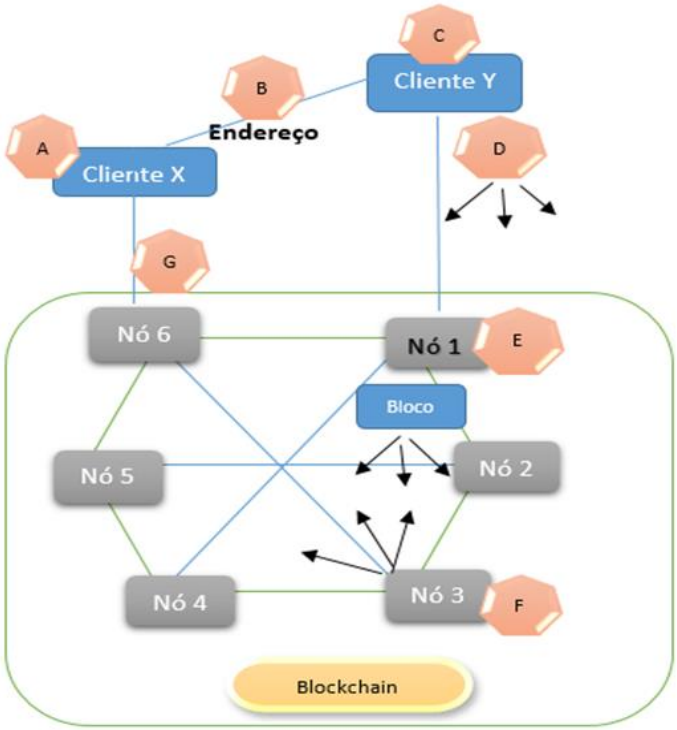
A figura 2 ilustra os componentes contidos em uma transação no *Blockchain*.

- **Valor de entrada**: Valor que o remetente irá enviar na transação.

- **Valor de saída:** Valor recebido na conta de destino.
- ***Hash*:** algoritmo que mapeia e transforma uma entrada de qualquer comprimento em um arquivo de comprimento fixo, através de uma função *hash*. No caso do *Blockchain*, é utilizada a função *hash* SHA256.
- **Timestamp:** Registro da data e hora em que uma transação foi efetivada.
- **Endereço de Destino:** Endereço para o qual o valor da transação será transferido.
- **Assinatura Digital:** Chave criptográfica privada em poder do remetente.

A figura 3 ilustra como uma transação na acontece em uma rede *Blockchain*.

Figura 3: Transação em uma rede *Blockchain*



Fonte: Autores, 2018

Na Figura 3, o cliente X, representa o nó que irá solicitar realizar uma transação em conjunto com o Cliente Y, que representa o nó que irá formar essa transação, em termos práticos, o cliente X, pode ser um estudante ou um departamento da instituição que irá solicitar a emissão de seu diploma, tão logo o Cliente Y, que representa o departamento responsável receba essa solicitação, este irá formar o equivalente a uma transação na rede, onde será verificada todos os dados do aluno, se ele está devidamente matriculado e cumpriu todos os requisitos para a obtenção do diploma. Com todos esses dados devidamente verificados, o cliente Y irá validar essa transação, assina-la digitalmente e divulgar a mesma entre os outros nós da rede P2P. Logo após, os nós da rede trabalham para obter o consenso, seja através das provas de trabalho, participação ou PBFT, após a obtenção do consenso, a transação é incluída em um bloco, e este bloco é incluído na cadeia de blocos.

Após este processo o Cliente X, o solicitante, pode consultar a base de dados do *Blockchain* e vê que sua transação foi validada e concluída, ou seja, que sua solicitação de emissão do diploma foi aceita, validada e o diploma emitido.

Conforme Braga (2017), a natureza assíncrona da comunicação entre os nós da rede P2P, estende o tempo necessário para a realização do consenso entre os nós, o que pode vir a dificultar a confirmação da conclusão da transação em um Blockchain tradicional direcionado a transações financeira. Porém, em um sistema para emissão de diplomas, o tempo de espera de cerca de 10 minutos não representa um tempo longo de espera, já que em um sistema tradicional de emissão

A figura 4, ilustra a estrutura interna do bloco que contém as transações. No cabeçalho do bloco, está o *hash* do bloco anterior, identificação do bloco anterior a este, o *hash* do bloco atual, que tem a função de identificar este bloco, o *timestamp* do bloco, registro da data de criação do bloco, o *nonce*, que é um número pseudoaleatório utilizado na validação do bloco, e a raiz da Árvore de *Merkle*[1], estrutura binária baseada em *hashs* (BRAGA; MARINO; SANTOS, 2017).

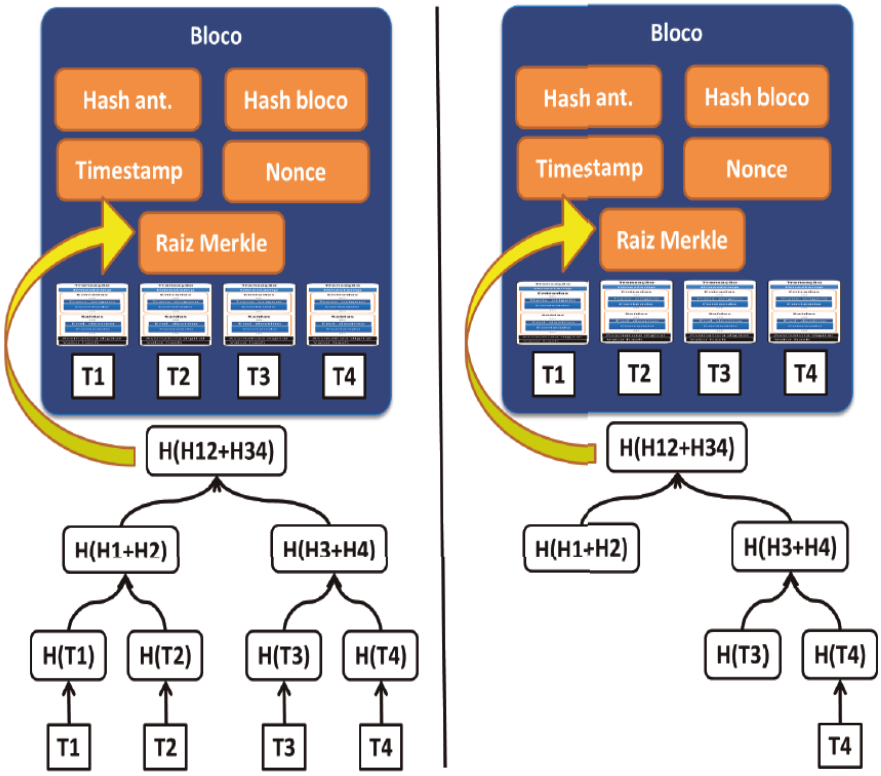
[1] Na árvore de Merkle (Figura 5), as folhas da árvore são os hashs das transações e os hashs dos pais são calculados utilizando o hash dos filhos, até chegar na raiz da árvore (BRAGA,2017).

Figura 4: Bloco com Transações

Fonte: Braga; Marino; Santos ,2017

Nesta Árvore de Merkle (Figura 5), no último nível, estão localizadas as folhas que contém os dados, ou apontadores para as transações. No penúltimo nível da árvore, os pais possuem apontadores *hash* para estes mesmos dados, a partir daí, os pais apontadores são agrupados em pares, até que cheguem à raiz da árvore, o *hash* da raiz da árvore é armazenado no *header* (ou cabeçalho) do bloco em segurança, junto com outras informações. Caso haja alguma alteração nos dados desta árvore, a verificação do *hash* irá apontar a modificação (GREVE et al.,2018).

Figura 5: Árvore de Merkle



Fonte: Braga,2017

2.2 Tipos de Consenso na *Blockchain*

Consenso Distribuído é um conceito de grande relevância para o perfeito funcionamento e segurança de um *Blockchain*. Segundo Braga (2017) consenso significa que a maioria dos envolvidos no processo tem que concordar com tal ação, sendo que consenso é diferente de unanimidade, nem todos os nós devem concordar, basta que a maioria concorde.

No *Blockchain*, existem três tipos de protocolos de consenso que constituem os protocolos mais popularmente utilizados na rede, o protocolo Bizantino, a prova de trabalho, utilizada no Bitcoin e a prova de participação, utilizada no Ethereum.

2.2.1 Consenso Byzantino.

Em uma rede *Blockchain*, no intuito de subverter a computação para transformar um processo correto em um processo malicioso, neste caso, considera-se o modelo de Falhas Byzantinas de Lamport. et al(1982) citado por Greve. et al (2018) no qual, diferentemente de um processo correto, um processo falho pode exibir qualquer comportamento, podendo parar, omitir envios e entregas de mensagens, ou desviar arbitrariamente de sua especificação.

Para que esta falha seja contornada e este problema devidamente solucionado, pode ser utilizado o Algoritmo de Tolerância a Falhas Byzantinas (PBFT). Conforme Chicarino et al (2017) ao usar o PBFT, o *Blockchain* pode tolerar nós defeituosos até X, onde X é uma fração aleatória e conhecida do total de nós da rede, valendo-se de uma máquina de estado replicada em nós diferentes (réplica definida como primária). Seu funcionamento se daria da seguinte maneira, segundo Chicarino et al. (2017):

- Um dos nós da rede envia uma solicitação de serviço para a máquina primária.
- A máquina primária replica os pedidos para os backups.
- As réplicas executam os pedidos e enviam respostas
- O cliente só pode considerar o resultado correto se receber o equivalente a X+1 resultados idênticos de réplicas diferentes.

Lembrando que é necessário que seja conhecido o número total de nós da rede, assim, o PBFT não é recomendado para redes públicas, sendo destinado a redes privadas. O PBFT também garante a consistência e integridade dos dados quando falhas byzantinas ocorrem em até 1/3 do total de nós da rede (CHICARINO et al, 2017).

2.2.2 Prova de Trabalho (PoW)

Na prova de trabalho ou proof of work (PoW), Nakamoto (2008) implementou um sistema onde a *Blockchain* roda em uma máquina de estados replicada, e os nós da rede enviam as transações para a ordenação dentro do bloco durante todo o tempo. Depois que as transações são agrupadas em blocos, o consenso será executado em rodadas para que a ordenação total dos blocos seja concluída.

A prova de trabalho é realizada para que seja eleito um nó para coordenar o consenso (GREVE et al, 2018).

Para se falar de prova de trabalho, faz-se necessário conceituar o que é mineração. Segundo Chicarino et al.(2017), a mineração é o processo de atualização do *Blockchain* utilizado pelo Bitcoin, no qual, os nós especiais, que são chamados de mineradores, são responsáveis por incluir as transações válidas em um bloco, validar esses blocos e acopla-los na cadeia de blocos do *Blockchain*. Esse é o processo de mineração, os mineradores, ao efetuar a prova de trabalho e realizar a mineração, gastam muita energia, e em troca disso, recebem bitcoins como recompensa pelo trabalho realizado.

A prova de trabalho consiste em resolver um desafio criptográfico, onde procura-se por um valor codificado por um algoritmo, SHA-256 (nonce), onde a codificação começa com um número de zero bits, sendo que o trabalho para decifrar este valor é exponencial em número de zero bits necessários para a conclusão da rodada (NAKAMOTO, 2008).

Nonce é um número que se torna uma variável quando em conjunto com o campo chamado "dificuldade alvo" para modificar a saída da função hash no cabeçalho do bloco. Como um exemplo, digamos que seja estabelecido que o número de hash válido para o novo bloco seja composto por 2 zeros, então o minerador, irá por força bruta repetir o nonce com o objetivo de resolver essa equação e gerar o número de hash que foi determinado como válido (GREVE et al.2018).

Dificuldade é o nome que se dá a colisão parcial de hash. Partindo do pressuposto de que cada *hash code* apresenta um resumo único para cada entrada que recebe, caso essa entrada seja modificada em 1 bit, o hash também será modificado completamente. O nonce é a variável utilizada para gerar essa colisão parcial e, assim, dependendo do poder computacional do nó minerador, resolver o problema proposto e gerar o novo bloco (CHICARINO et al., 2017).

Segundo Nakamoto (2008), os passos para rodar a rede são:

1. Surgem novas transações que são transmitidas para todos os nós.
2. Cada nó da rede reuni as novas transações em um bloco.

3. Cada nó busca realizar a prova de trabalho necessário para validar o bloco.
4. Quando o nó decifra a PoW, ele repassa o bloco para todos os nós.
5. O bloco é aceito somente se todas as transações dentro dele forem válidas e não houver gasto-duplo.
6. O indicativo de que o bloco foi aceito na rede, se mostra quando os outros nós começam a trabalhar no próximo bloco da corrente, utilizando o *hash* deste bloco como *hash* do bloco anterior.

Uma das características da PoW é que ela deve ser difícil e trabalhosa para decifrar, mas não impossível, outra característica é que esta prova deve ser verificada de modo fácil e rápido e concluída em cerca de 10 minutos, sendo que a dificuldade da PoW é ajustada a cada 2016 blocos (CHICARINO et al., 2017).

2.2.3 Prova de Participação (PoS)

Na prova de participação, ou Proof of Stake (PoS), em vez de minerar os blocos através da resolução de um problema criptográfico, como na prova de trabalho, o nó selecionado para a criação de um novo bloco é escolhido de maneira probabilística, levando em conta a sua riqueza, ou quantidade de moedas que o nó disponibiliza para a operação.

Em outras palavras, se por um lado um usuário gastasse cerca de R$2.000 em equipamentos para aumentar o poder de processamento computacional a fim de minerar mais moedas através da PoW, no PoS o usuário usaria esse mesmo valor com criptomoedas para aumentar suas chances de ser o nó escolhido para criar o novo bloco e tornar-se um validador. (CHICARINO et al., 2017).

De maneira oposta a prova de trabalho, na prova de participação, a potência computacional e o tempo gasto são trocados por uma forma mais sustentável de realizar esse consenso, com base na quantidade de criptomoedas que cada nó possui. Para que o nó tenha maior probabilidade de ser o sorteado como validador, ele deve ter um maior número de moedas reservadas para isso. Caso ele seja selecionado como o validador e depois de criar o bloco tente alterá-lo ou exclui-lo, ele perderá as suas moedas. Esta é uma das ações previstas como modo de manter a integridade dos nós e evitar ataques (CHICARINO et al., 2017; GREVE et.al., 2018).

Como se pode prever, o nó que investe mais, sempre sairia beneficiado de uma seleção baseada na riqueza dos nós, o que geraria uma certa centralização no processo de seleção do nó gerador do bloco, porém, algumas medidas estão sendo usadas para driblar este problema, como por exemplo, moedas como a NXT ou Blackcoin estão investindo em um processo de aleatorização para selecionar o próximo nó que gerará o próximo bloco.

Este sistema de aleatorização consiste em utilizar uma fórmula que procure o menor valor de *hash*, que será usado em combinação ao tamanho da participação do nó, porém, a fragilidade deste sistema reside no fato de quê, uma vez que as apostas são públicas, os nós da rede podem prever, com certa precisão, o nó que será selecionado para gerar o próximo bloco da rede (CHICARINO et al.,2017).

2.3 Criptografia na *Blockchain*

Segundo Braga e Dahab (2015) a criptografia sempre foi relacionada ao sigilo, porém a criptografia moderna abrange outros temas tais como:

- **Confidencialidade:** uso da criptografia afim de manter a confidencialidade.
- **Autenticação:** validar a identidade de uma entidade.
- **Integridade:** Uso da criptografia para garantir que certos dados não foram modificados desde sua criação.
- **Não-Repúdio:** Uso da criptografia para garantir que o autor de uma determinada mensagem não possa negar a autoria da mesma.

Em uma situação prática, todas essas características têm que estar presente para que um sistema possa ter um nível mais adequado de eficácia e segurança.

Em um *Blockchain* tradicional, os elementos criptográficos mais comuns em sua implementação são: as funções de *hash*, usadas para gerar endereços para as transações dentro dos blocos, que são calculados através de chaves públicas, as assinaturas digitais, que são utilizadas para garantir a irrefutabilidade e autenticidade dessas transações e a criptografia assimétrica, que é utilizada no sentido de garantir integridade, autenticidade e reforçar a irrefutabilidade das transações (BRAGA, 2017).

2.3.1 Criptografia Assimétrica

A criptografia de chave pública (ou assimétrica) utiliza duas chaves que se complementam matematicamente e que trabalharão juntas. Uma das chaves é a chave privada, ou pessoal, que é mantida em segredo pelo dono do par de chaves e é usada para decriptar as mensagens que chegam ao destino. A outra chave é a chave pública, que justamente por ser pública, é utilizada para encriptar as mensagens do remetente. A criptografia de chave pública é de extrema importância para assegurar a integridade da comunicação feita em uma rede pública, como a internet, porém de forma privada e segura (BRAGA; DAHAB, 2015).

2.3.2 Assinatura Digital

A assinatura digital é o resultado de uma operação criptográfica utilizando uma chave privada sobre o conteúdo original. O detentor da chave privada pode gerar mensagens que serão assinadas digitalmente e que podem ser verificadas por qualquer pessoa que possua a chave pública correspondente, o que reforça o conceito de irrefutabilidade, pois o assinante não poderá negar a autoria da mensagem, pois sua assinatura digital é feita com sua chave privada exclusiva (BRAGA, 2017).

Nem sempre essas assinaturas digitais são feitas em cima do conteúdo original, pois não é recomendado na transmissão de dados, além de requerer um processo matemático mais complexo para a criptografia, o que resulta em um desempenho ruim em processadores mais lentos, o que nos leva a conclusão que não é o texto claro inteiro que é assinado digitalmente, mas sim, um resumo desse texto. Um resumo de tamanho fixo que o identifique como único, ou seja, uma função *hash* (BRAGA; DAHAB, 2015).

2.3.3 Funções *hash*

O *hash*, ou função *hash* em ciência da computação é definido como um algoritmo que transforma qualquer documento de comprimento variável em uma sequência de dados fixos (STALLINGS, 2008).

O código *hash* é único para cada documento e de tamanho muito menor que o documento original, além de poder mudar totalmente sua sequência se houver

alguma mudança, por menor que seja, no arquivo original, a função de *hash* também é considerada unidirecional, pois através dela, não é possível decifrar ou recuperar os dados originais (BRAGA; DAHAB, 2015).

A função *hash* é de uma grande importância para o *Blockchain*, pois é através dela que se mantém a integridade da cadeia de blocos. Caso um nó mal-intencionado tente modificar algo nos blocos, a *hash* será alterada automaticamente, o que poderá ser detectado por qualquer outro nó dentro da rede, que reconhecerá a tentativa de fraude, dificultando tentativas de roubo ou fraude dentro da rede (CHICARINO et al.,2017).

2.3.4 Visão Geral

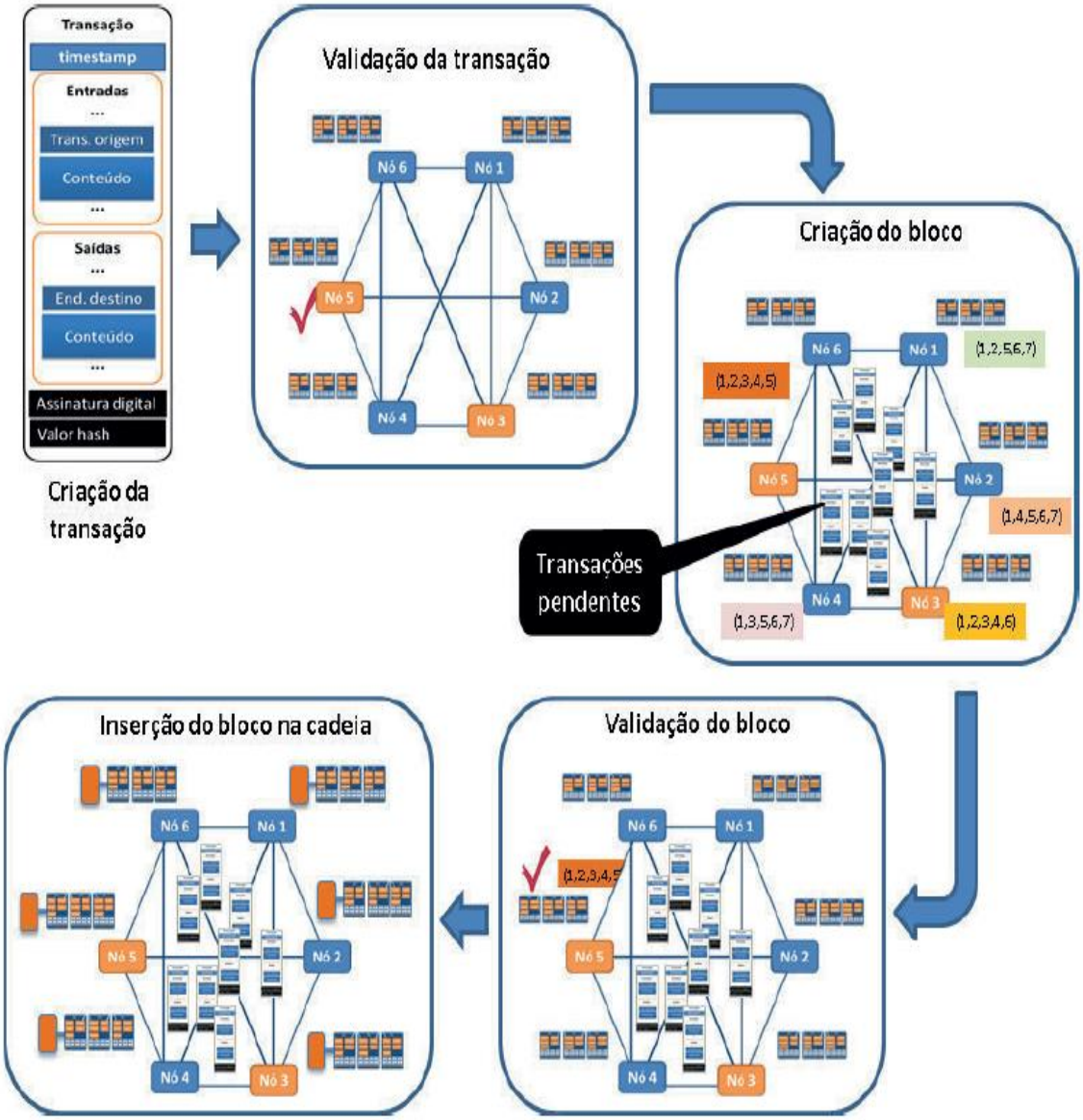
Por ser considerado um processo dinâmico e complexo, muitas vezes se torna um pouco trabalhoso visualizar a cadeia completa de eventos do *Blockchain*.

Conforme Braga; Marino e Santos (2017), a Figura 6, exemplifica como acontece a interação entre os conceitos do *Blockchain* e o processo de validação das transações e inclusão das mesmas no bloco

Primeiramente, um usuário, por meio de um software eWallet (carteira eletrônica) cria uma transação, esta transação será validada por um nó da rede que irá verificar sua integridade e autenticidade, logo após a validação, esta será publicada entre os nós da rede e será encaminhada para ser incluída a um dos blocos da rede

Até que esta transação seja incluída em um bloco, ela será considerada pendente, blocos são criados e validados a todo momento, e os nós incluem transações nestes blocos a todo momento. Estes blocos também são divulgados na rede e se forem validados, através dos protocolos de consenso, são incluídos na cadeia de blocos do Blockchain, de forma imutável e pode ser consultada por todos os nós da rede P2P.

Figura 6: Cadeia de Eventos no Blockchain



Fonte: Braga, Marino e Santos, 2017

2.4 Contratos Inteligentes (*Smart Contracts*).

Em 2014 com a popularização do *Blockchain*, novas versões dele foram surgindo. Enquanto o *Blockchain* 1.0 contempla a descentralização das transações monetárias, o *Blockchain* 2.0, abrange também as DAPPs (Aplicativos Descentralizados) DAOs (Organizações Descentralizadas Autônomas) e DACs (Corporações Autônomas descentralizadas) utilizando a descentralização do mercado de modo mais amplo, contemplando a descentralização de muitos outros

serviços através da rede. Um dos conceitos-chave é que o *livro razão da rede*, antes utilizado para transferir, registrar e confirmar transações financeiras, agora seja utilizado para transferir, validar e registrar todo tipo de contratos e propriedades (BUTERIN, 2014).

Szabo (1997) definiu *Contratos Inteligentes*, ou *Smart Contracts* como um protocolo de transação informatizado que executa os termos de um contrato comum, e satisfazem critérios como confidencialidade, pagamento, cumprimento do contrato entre outros, além de ser um meio para dificultar ações maliciosas e também operar sem a necessidade de intermediários.

Ao implementar um *contrato inteligente*, o programa executável do contrato funciona perfeitamente nos nós do *Blockchain*, não dependendo de uma entidade externa, pois seu desempenho é garantido pelos protocolos de consenso da rede (BRAGA, 2017).

Segundo Chicarino et al. (2017), cada um dos *contratos inteligentes* possuem um endereço e os mesmos são acionados quando uma determinada transação lhes é endereçada, então serão executados automaticamente da maneira que foi determinada previamente em cada nó da rede e de acordo com os dados que foram incluídos na transação, deste modo, cada nó da rede que é habilitado por um contrato inteligente executa uma máquina virtual (apud CHRISTIDIS and DEVETSIKIOTIS, 2016).

Os contratos inteligentes possuem três tipos de características que os distingue: autossuficiência, autonomia e descentralização (CHICARINO et al.,2017).

- **Autonomia:** Após o lançamento e a execução, o contrato e quem o iniciou não necessitam manter o contato sempre.
- **Autossuficiência:** Em sua capacidade de gerar recursos e gerar fundos através dos serviços que oferece e gasta-los em recursos necessários para sua manutenção, como armazenamento e processamento.
- **Descentralização:** Pois os contratos inteligentes não estão alocados em um único servidor central, mas existem e são executados em diversos nós através da rede.

Deste modo, podemos constatar que a adição da tecnologia de contratos inteligentes expandiu os horizontes da tecnologia *Blockchain* possibilitando, assim,

inúmeras outras aplicações onde essas duas tecnologias podem atuar de maneira conjunta.

3 CERTIFICAÇÃO DIGITAL E ASSINATURA DIGITAL

Conforme Dorneles e Corrêa (2013), a internet é um dos meios mais utilizados para o envio e recebimento de dados e informações de todos os tipos, entre indivíduos, governos, instituições privadas, ONGs e etc., no entanto, essas transações virtuais careciam de um mecanismo que, através de uma série de critérios de segurança, pudessem assegurar a veracidade dos dados, tais como:

- **Confidencialidade:** Garantia de que as informações serão acessadas somente por pessoas autorizadas;
- **Integridade:** Proteção da exatidão dos dados e que os mesmos serão entregues completos aos destinatários;
- **Não - Repúdio:** impedimento caso uma das partes envolvidas no processo de envio e recebimento de dados venha contestar falsamente sua participação no processo.

De acordo com o ITI - Instituto Nacional de Tecnologia da Informação (2017), a Certificação Digital é o mecanismo para que todos esses critérios de segurança sejam alcançados, sendo assim, no dia 10 de fevereiro de 2009, o Comitê Gestor da ICP-Brasil que é responsável pela coordenação, implantação e funcionamento desta estrutura, definiu que a Certificação Digital seria tratada como um produto, não como um serviço. A certificação digital se configura como um produto intangível, por ser de natureza eletrônica e como um software personalíssimo, pois não é distribuído e utilizado de maneira padrão por todos os adquirentes, antes deve haver uma coleta de dados pessoais de cada cliente, para que haja uma personalização do software, a fim de torna-lo plenamente utilizável pelo usuário.

A certificação digital funciona como se fosse uma identidade virtual, que torna a identificação de uma mensagem ou transação segura e sem equívocos quanto ao seu autor. Este documento é gerado e assinado por uma Autoridade Certificadora (AC) que relaciona uma entidade qualquer (pessoa, processo ou servidor) a um par de chaves criptográficas, tudo de acordo com as regras estabelecidas pelo Comitê Gestor da ICP- Brasil (ITI,2017).

3.1 ICP - BRASIL - Infraestrutura de Chaves Públicas Brasileiras

O ICP-Brasil foi criado em 2001, por meio do Art. 1º da Medida Provisória, número 2.200-2 de 24 de Agosto de 2001,como uma autarquia federal ligada à casa Civil da Presidência da República, e é responsável pela infraestrutura das chaves públicas brasileiras.

> Art. 1º Fica instituída a Infraestrutura de Chaves Públicas Brasileira - ICP Brasil, para garantir a autenticidade, a integridade e a validade jurídica de documentos em forma eletrônica, das aplicações de suporte e das aplicações habilitadas que utilizem certificados digitais, bem como a realização de transações eletrônicas seguras. (BRASIL, 2001)

A ICP Brasil é composta por uma cadeia hierárquica de confiança que proporciona a emissão de certificados digitais para a identificação virtual do cidadão. O modelo de certificação que é utilizado no Brasil, é o de certificação com raiz única; cabe ao ITI a função de Autoridade Certificadora Raiz - AC-Raiz, além de desempenhar a função de credenciamento e descredenciamento de outros membros da cadeia, além de supervisionar e auditar os processos correntes (ITI,2017)

3.1.2 Autoridade Certificadora

Segundo as regras da ITI (2017) a Autoridade Certificadora - AC, é uma entidade que pode ser de natureza pública ou privada, que está sob a hierarquia do ICP- Brasil e suas responsabilidades são:

- A emissão, distribuição, revogação, renovação e gerenciamento de certificados digitais;
- Verificar se o titular do certificado possui a chave privada correspondente à chave pública do certificado;
- Criar e assinar digitalmente o certificado do assinante. Este certificado, que foi emitido pela AC, equivale a uma declaração da identidade do titular que possui as chaves pública e privada;
- Emitir a Lista de Certificados Revogados – LCR;

- Manter registros de suas operações, de acordo com as regras da Declaração de práticas da Certificação – DPC;
- Estabelecer as políticas de segurança para a garantia da autenticidade da identificação realizada e cobrar o cumprimento das mesmas pelas Autoridades de Registros – AR.

3.1.3 Autoridades de Registros

A Autoridade de Registro funciona como um mediador entre o cliente e a AC e suas atribuições são:

- Recebimento e validação de certificado digital;
- Encaminhar as solicitações de emissão e revogação das certificações digitais;
- Manter o registro de suas operações.

As autoridades de Registro podem ser localizadas fisicamente em uma AC ou realizar suas operações como uma autoridade de registro remota (VIEIRA; ARAÚJO, 2012).

3.1.4 Autoridade Certificadora do Tempo

ACT é uma entidade que viabiliza os serviços de carimbo de tempo emitindo-os para os usuários. A ACT é a principal responsável pelo carimbo de tempo, que em conjunto com a assinatura digital, tem como objetivo delimitar a certo período de tempo a existência do documento certificado. Ao produzir um documento, seu conteúdo é criptografado e recebe os seguintes atributos: ano, mês, dia, hora, minuto e segundo. Se atestado em conjunto com a assinatura feita com o certificado digital, atesta a autenticidade não apenas através da prova temporal, mas também do conteúdo do documento (ITI, 2017).

3.2 Criptografia e Assinatura Digital

Para que a certificação e a assinatura digital venham a dar a devida segurança e veracidade à certificação, é necessário entender sobre a criptografia envolvida no processo de segurança e verificação das assinaturas digitais e certificados, pois, foi através de criação e uma tecnologia de criptografia, patenteada

em 1983 por professores do Instituto de Tecnologia de Massachusetts (MIT), que deu origem a certificação digital (PINHEIRO, 2010).

Para que essas tecnologias sejam devidamente compreendidas, faz-se necessária a compreensão de alguns conceitos técnicos quanto a criptografia utilizada nas assinaturas digitais, tais como criptografia assimétrica e simétrica, chaves criptográficas e resumos criptográficos (DORNELES; CORRÊA, 2013).

3.2.1 Sistemas Criptográficos

Há dois tipos de sistemas criptográficos, o simétrico (Chave Privada) e o assimétrico (Chave Pública). O sistema simétrico de criptografia utiliza uma única chave privada para encriptar e decriptar os dados. No sistema assimétrico, é utilizado duas chaves, uma para encriptar e outra para decriptar os dados, essas duas chaves se relacionam matematicamente e trabalham em pares, cada uma das chaves inverte o trabalho da outra chave. Nos sistemas de chave pública, umas das chaves, a privada, é utilizada para decriptar e a chave pública é utilizada para encriptar (BRAGA; DAHAB, 2015).

3.2.2 Criptografia de Chave Pública

Segundo Oliveira (2012), na criptografia de chave pública, há duas chaves, onde uma delas é de uso pessoal e secreto do dono do par de chaves (chave privada), e a outra chave é conhecida publicamente (chave pública). Estas chaves são construídas para trabalharem juntas. Através da criptografia de chave pública é possível o sigilo dos dados, uma vez que qualquer pessoa que utilize a chave pública pode enviar criptogramas para o dono da chave privada equivalente.

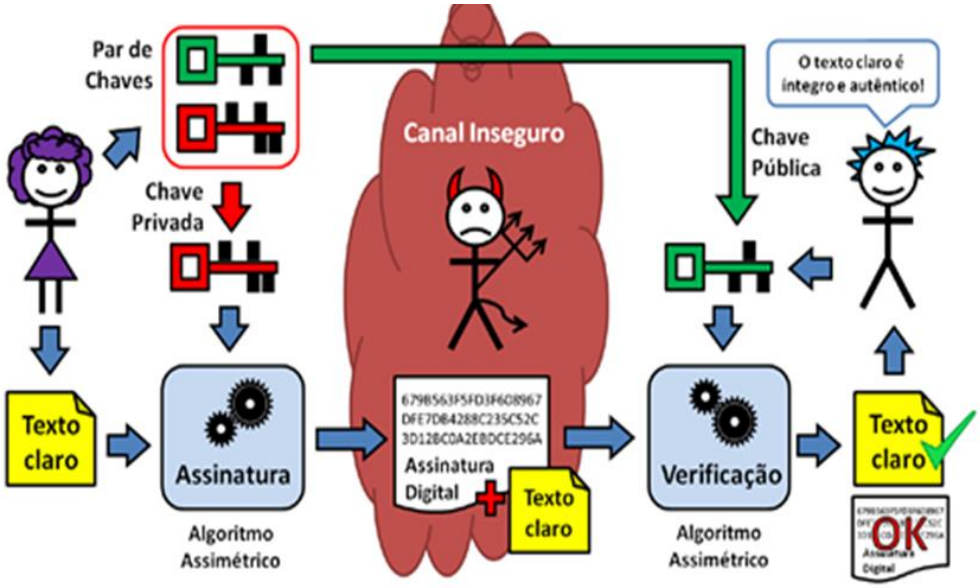
A assinatura digital é o resultado de uma operação criptográfica, realizada com a chave pública de alguém, para encriptar um texto claro, utilizando para isso, o sistema de criptografia assimétrica. Neste caso, a criptografia de chave pública é utilizada para conferir integridade, autenticidade e irrefutabilidade ao documento.

Portanto, o dono da chave privada pode enviar mensagens assinadas, que podem ser lidas por qualquer pessoa que tenha sua chave pública correspondente, e assim, verificar a autenticidade da assinatura digital (DORNELES; CORRÊA, 2013).

O fato de que qualquer pessoa de posse da chave pública possa ter acesso ao conteúdo de uma mensagem assinada digitalmente, embora isso seja um fato que pode vir a comprometer seu sigilo, não compromete a irrefutabilidade da mensagem, visto que a chave pública só pode decriptar mensagens encriptadas com sua chave privada correspondente e de uso exclusivo do dono do par de chaves, garantindo assim a irrefutabilidade do autor da mensagem (BRAGA; DAHAB, 2015).

A Figura 7 ilustra o funcionamento de um sistema criptográfico assimétrico para autenticação.

Figura 7: Criptografia de Chave Pública



Fonte: Braga; Dahab, 2015

Na Figura 7, vê-se que uma pessoa X envia uma mensagem assinada com sua chave privada para a pessoa Y. Esta mensagem passa por um canal inseguro, assim, todos que sabem quais as chaves públicas correspondentes podem ler a mensagem, ou seja, a mensagem não é sigilosa. Porém, não pode ser adulterada, pois somente a pessoa X está de posse de sua chave privada (BRAGA; DAHAB, 2015)

Entretanto, este sistema tem uma falha, pois em um sistema de chaves públicas, para que as chaves sejam gerenciadas de modo correto, primeiro deve-se procurar saber qual a chave pública da pessoa com a qual deseja-se comunicar.

Segundo, deve-se ter uma garantia de que a chave pública que foi envia realmente pertence ao dono legitimo. Sem que haja uma garantia, um intruso pode utilizar chaves falsas para se passar por qualquer uma das partes envolvidas, estabelecendo assim um vínculo de confiança (OLIVEIRA, 2012).

Na prática, quando um emissor envia uma mensagem solicitando a chave pública a alguém, esta mensagem pode ser interceptada por um intruso que mandará como resposta ao emissor, uma chave pública falsificada. Desta maneira, o intruso pode fazer o mesmo processo com o interceptor, então, o emissor e o receptor pensam que estão se comunicando um com o outro, quando na verdade estão sendo interceptados por um intruso. Este intruso pode decifrar todas as mensagens, cifrá-las ou substituí-las por outras mensagens.

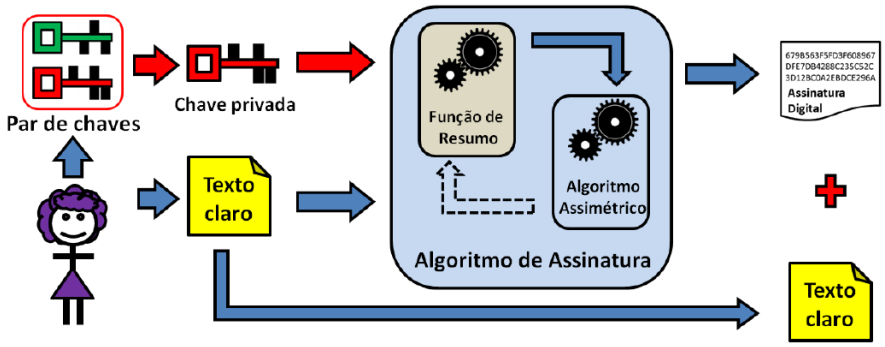
Para que seja evitado este tipo de situação, é que os certificados de chave pública ou certificados digitais são utilizados. Eles são chaves públicas assinadas por uma terceira parte, que seria uma pessoa de confiança. Essa medida tem como objetivos evitar a substituição de chaves públicas por chaves forjadas, o certificado é assinado por uma autoridade certificadora e além da chave pública, contém outros dados do titular das chaves, tais como nome, endereço e outros dados pessoais (OLIVEIRA, 2012).

3.2.3 Assinatura Digital e Resumo Criptográfico

Ao enviar uma mensagem com a assinatura digital, não é o conteúdo original que recebe a assinatura digital e sim, um resumo criptográfico deste texto, o resumo criptográfico atua como um identificador exclusivo da mensagem, isto acontece por vários motivos, tais como o fato de que uma assinatura digital com o mesmo tamanho do texto claro pode vir a dificultar a transmissão de dados. Outro motivo envolve a matemática utilizada na criptografia de chave pública, que é complexa, o que comprometeria o desempenho de modo eficiente em processadores mais lentos. Sendo assim, este identificador único é calculado por rotinas matemáticas, dando origem ao resumo criptográfico, ou função *hash* (BRAGA; DAHAB, 2015).

O *hash*, por ser muito menor, facilita a transmissão de dados . O *hash* gera uma sequência de bits de tamanho fixo e possui um valor único para cada mensagem gerada. Outra característica do *hash* é ser unidirecional, pois é irreversível e as chances de haver dois documentos com o mesmo valor de *hash* é extremamente improvável (OLIVEIRA, 2012).

Figura 8: Como a Função *Hash* Atua no Texto Claro



Fonte: Braga; Dahab, 2015

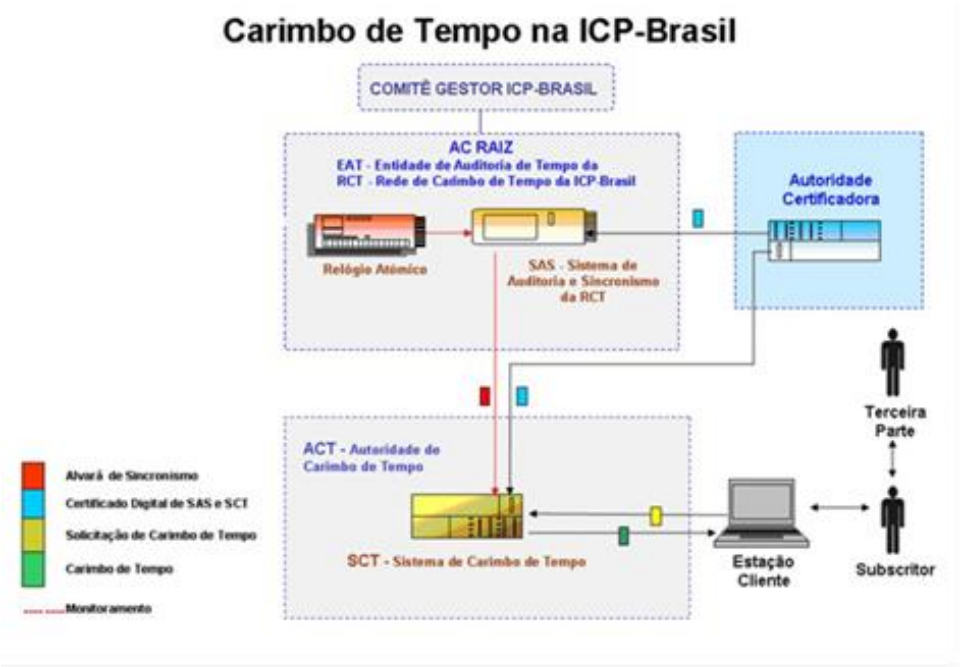
Pode-se ver na Figura 8, vemos que a dona das chaves não assina digitalmente o conteúdo original, pois ele pode ter um tamanho que dificulte a transmissão dos dados. Ao invés disso, ela utiliza um mecanismo de assinatura digital que calcula o *hash* do texto original. Então, o *hash* da mensagem será assinada digitalmente com a chave privada através de um sistema de assinatura digital. O modo como a tecnologia de resumo criptográfico e o sistema assimétrico irão trabalhar em conjunto, depende de cada mecanismo de assinatura digital (BRAGA; DAHAB , 2015).

3.3 Carimbo de Tempo

Segundo Moecke (2008), o carimbo de tempo, constitui uma importante ferramenta de segurança, pois é o responsável por delimitar os marcos de tempo de vida de uma assinatura digital. É de extrema importância saber que a assinatura digital está sendo utilizada em um espaço de tempo, onde se possa ter certeza de que seu certificado, e todos os algoritmos utilizados em sua segurança são válidos. Para isso, é necessário que a assinatura digital esteja atrelada há uma data segura e exata.

A data atrelada a assinatura digital deve ser confiável, por isso não pode ser estabelecida por qualquer pessoa ou órgão não credenciado para isso, pois ocorreria um grande risco de fraude. Sendo assim, foi estabelecido pela ICP- Brasil, uma terceira parte no processo de certificação digital, a Autoridade Certificadora de Tempo – ACT, órgão responsável pela validação do ciclo de via de uma assinatura digital.

Figura 9: Carimbo de Tempo da ICP Brasil



Fonte: ICP- Brasil, 2015

Na Figura 9, podemos ver como é o modelo de funcionamento do carimbo de tempo da ICP- Brasil (2015):

A) O relógio atômico fornece a data e horário preciso para o Sistema de Auditoria e Sincronismo da RCT –Rede de Carimbo de tempo da ICP-Brasil.
B) Estes dados são enviados à uma ACT – Autoridade de Carimbo de Tempo, onde serão registrados no STC- Sistema de Carimbo de Tempo desta ACT.
C) Quando for solicitado um carimbo de tempo por um cliente, este será enviado do STC para a estação cliente.

Para garantir que as ACTs possam ser da mais alta confiança ao assinalar o tempo preciso, elas utilizam o sistema de relógios atômicos de alta precisão, de lugares como o Observatório Nacional ou do Instituto Nacional de Tecnologia da Informação. Os relógios atômicos são regulados a partir da vibração de elementos químicos extremamente estáveis, como exemplo, o Césio 133; a estabilidade do elemento determina a estabilidade do relógio em marcar o tempo de forma precisa (VIVIAN, 2018).

3.3.1 Diferenças entre Carimbo de Tempo e Timestamp.

De acordo com Vivian (2018), “O *Timestamp* é um modelo internacional que é adaptado regionalmente de acordo com as diretrizes definidas pelos órgãos responsáveis pela certificação digital em cada país”.

Algumas características do Timestamp:

- Estrutura de dados criptográfica, baseadas em um padrão internacional, o RFC 3160;
- O RFC 3160 é um documento produzido pelo ETSI – *European Telecommunications Standards Institute,* entidade que é responsável por elaborar diretrizes para diversos setores;
- É baseado em uma estrutura de chaves públicas (PKI).
- Através de um mecanismo de NTP – *Network Time Protocol*, registra em formato de resumo criptográfico a data e horário preciso em que o documento foi criado;
- No caso de uma rede *Blockchain*, o uso do carimbo de tempo é obrigatório.

Segundo o ICP- Brasil (2015), a estrutura básica do *timestamp* foi adaptada para gerar uma versão brasileira, que é a que chamamos de carimbo de tempo, algumas de suas características são:

- Estruturado e modificado pela ICP- Brasil;
- É definido como um documento eletrônico, emitido por uma Autoridade Certificadora de Tempo;
- O uso do carimbo de tempo no âmbito da ICP- Brasil é facultativo, os documentos assinados com chave privada de acordo com as regras da ICP- Brasil, são válidos com ou sem carimbo de tempo;

- Suas diretrizes gerais de implementação estão reunidas no Doc.11 da ICP- Brasil.

3.4 Comparativo entre os Sistemas *Blockchain* para Documentos e Certificação Digital.

Este trabalho visa a apresentação de um projeto para a validação digital de documentos através de uma rede *Blockchain*, portanto, abordamos aqui, todos os mais importantes aspectos e características que este sistema possui, da mesma forma, também abordamos as principais características sobre a certificação digital, pois atualmente, a certificação digital constitui a principal forma de validar e assegurar a veracidade de um documento no ambiente virtual.

Porém, cabe aqui destacar o que difere um sistema do outro, e o porquê de um sistema utilizando o *Blockchain* pode vir a substituir a ideia da certificação digital com êxito.

Dentre as características que difere um sistema *Blockchain*, estão:

- **Autenticação é válida dentro e fora o ambiente virtual:** Uma vez que o documento é validado, este se torna válido tanto dentro do ambiente virtual, como fora dele, através da verificação do *hash* do documento no sistema;
- **Não possui intermediário entre o serviço e o usuário**: Não há necessidade de uma entidade mediadora entre as partes do acordo, já que esta é uma das características principais do *Blockchain*;
- **Autenticação sem prazo de validade:** Ao contrário de uma certificação digital, uma vez que o documento é validado pelos nós da rede e anexado ao bloco, não haveria revogações na autenticação deste;
- **Sem modificações ou exclusão de documentos autenticados:** Uma vez dentro do bloco, não pode haver modificação ou exclusão de um documento, pois os outros nós da rede podem identificar a fraude;
- **Utilização do *timestamp* é obrigatório:** O uso do *timestamp* é obrigatório, pois ao marcar a data e horário da transação, permitirá que seja verificada a data em que o documento foi emitido pela instituição e consequentemente, a partir de que data ele é válido;

- **Maior Agilidade no Processo:** Ao eliminar boa parte dos intermediários que geralmente estão envolvidos em um processo mais tradicional, é uma consequência natural que o processo de emitir o documento seja feito com mais agilidade, em menos tempo.

Em contrapartida, a certificação digital apresenta as seguintes características:

- **É válida somente para documentos eletrônicos:** Não sendo válida, fora do ambiente virtual;
- **Apresenta intermediários:** Como já vimos em outros tópicos, para que o cliente possa obter sua certificação digital, há uma série de autoridades certificadoras e outras entidades como mediadoras neste processo;
- **Prazo de Validade:** A assinatura digital apresenta um prazo de validade para sua utilização, podendo ser revogada pela Autoridades de Registro que são responsáveis pela lista de certificados revogados;
- **Carimbo de Tempo Facultativo:** Apesar de constituir um mecanismo para delimitar o ciclo de vida e validação de certificado, o carimbo de tempo é de uso facultativo, um certificado pode ser utilizado sem que possua um carimbo de tempo;
- **Menor Agilidade no Processo:** Como os certificados digitais são emitidos pelas autoridades certificadoras, através da ICP- Brasil, além de ter outras entidades mediadoras, como a AR e a ACT, o processo naturalmente é mais burocrático, o que compromete a agilidade do processo.

No projeto a ser desenvolvido, as informações referentes ao documento serão preenchidas dentro de um formulário web. As informações serão enviadas para o *Blockchain* através de um contrato inteligente previamente implementado. Após enviadas, as informações serão validadas e salvas no bloco. O bloco conterá uma numeração sequencial, o código *hash* e o timestamp referente a data e hora do registro.

A partir deste momento as informações do documento estarão dentro da *Blockchain* e sujeitas às regras de autenticação da própria plataforma. Deste modo, as informações não mais poderão ser modificadas ou excluídas, somente sendo possível ser consultadas através do código *hash*. Por se tratar de um sistema

distribuído ponto a ponto, a autenticação não necessita de intermediários, tornando o processo mais ágil e desburocratizado.

Ao analisarmos todas as características dos dois sistemas, podemos observar suas diferenças e como são distintos os seus meios de validar, armazenar e emitir os documentos, levando em consideração que o público-alvo também é diferente um do outro, este trabalho visa uma a autenticação de documentos oficiais de uma instituição, enquanto que a certificação digital, já é amplamente utilizada no mercado.

Partindo deste ponto, acredita-se que um sistema de autenticação de documentos oficiais de uma instituição de ensino como o IFPA, utilizando uma rede *Blockchain*, poderá trazer benefícios para instituição como o todo, além de ser uma ferramenta eficaz contra a falsificação de documentos.

4 PROJETO REGISTRO DE DIPLOMAS

O sistema Registro de Diplomas tem como objetivo o registro de diplomas utilizando um contrato inteligente na rede *Blockchain* da *Ethereum*. Após a emissão do documento, as informações estarão salvas no *Blockchain* de forma criptografada. O diploma poderá ser consultado e validado através de uma chave criptográfica.

O contrato inteligente, para o protótipo, será desenvolvido com a linguagem *Solidity* e implementado na rede usando Node.js. Após a implementação o contrato poderá ser consumido através da API Web3.js. A API será responsável pela comunicação entre a aplicação web e o Contrato Inteligente.

Para aplicação web, será utilizada a biblioteca *React*, que permite a construção de telas de forma padronizada através do uso de componentes independentes, tornando o seu desenvolvimento mais dinâmico e de fácil manutenção.

4.1 Requisitos do Projeto

O sistema Registro de Diplomas tem como objetivo o registro de diplomas utilizando um Contrato Inteligente na rede *Blockchain* da *Ethereum*. Após a emissão do documento, as informações estarão salvas no *Blockchain* de forma criptografada. O diploma poderá ser consultado e validado através de uma chave criptográfica.

- **Requisitos funcionais:** abordam o que sistema deve fazer;
- **Requisitos não funcionais:** são características de qualidades que o sistema deve possuir como, confiabilidade, portabilidade, segurança e usabilidade (Sommerville, 2004).

Diante disso, seguem na tabela abaixo os requisitos funcionais (RF) da aplicação:

Tabela 1: Requisitos Funcionais

Referência	Descrição
RF – 01	A aplicação deverá exibir uma lista de diplomas emitidos por data de emissão, nome do aluno, número de matrícula do aluno e/ou curso
RF – 02	A aplicação deverá exibir detalhadamente cada diploma emitido

RF – 03	A aplicação deve ter um formulário para emissão de um novo diploma
RF – 04	A aplicação deve ter uma opção validar a existência desse diploma

Os requisitos não funcionais (RNF) do sistema:

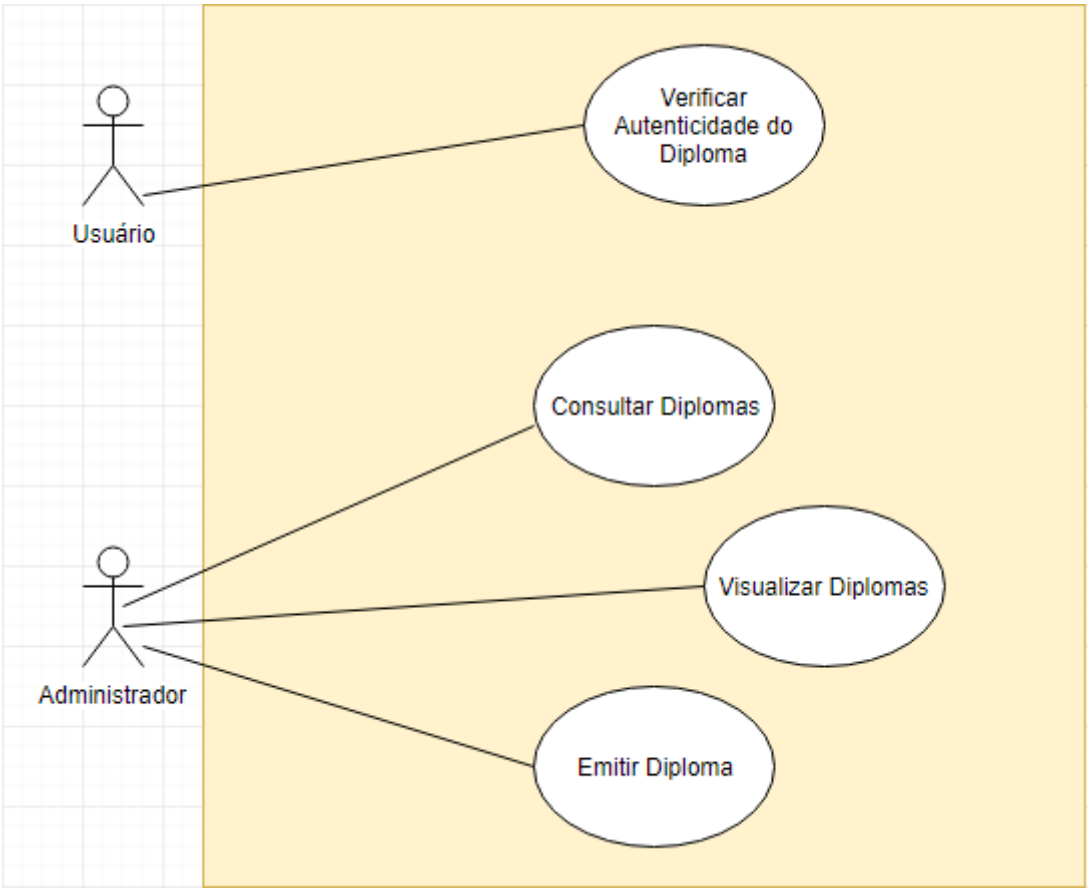
Tabela 2: Requisitos Não Funcionais

Referências	Categoria	Descrição
RNF – 01	Usabilidade	A aplicação deverá ser simples e intuitiva para que o usuário não tenha dificuldades em seu uso.
RNF – 02	Usabilidade	A aplicação deverá ser responsiva, podenda ser operada na maior quantidade possível de dispositivos e formatos de tela
RNF – 03	Implementação	O sistema deverá ser desenvolvido para Web.
RNF – 04	Implementação	A aplicação web deverá ser desenvolvida nas linguagens de programação Javascript, HMTL e CSS.
RNF – 05	Implementação	O contrato inteligente deverá ser implementado na linguagem de programação Solidity
RNF – 06	Interoperabilidade	O sistema deverá se comunicar com o Contrato Inteligente implementado no Blockchain da plataforma Ethereum.

RNF - 07	Segurança	A aplicação deverá garantir a integridade e imutabilidade das informações contidas no diploma após a emissão

Na Figura 10 é mostrado o diagrama de casos de uso.

Figura 10: Caso de Uso



Fonte: Autores, 2018

Para maior compreensão são apresentadas as especificações de cada caso de uso:

a) **Caso de Uso – Verificar autenticidade do diploma**: o usuário digitará o código de verificação do diploma em um campo de texto e clicará no botão verificar autenticidade. O sistema exibirá na tela as informações do diploma e uma mensagem de sucesso;

b) **Caso de Uso – Consultar diplomas:** o administrador digitará nome do aluno, número de matrícula do aluno e/ou data de emissão do diploma e clicará no botão buscar. O sistema exibirá uma lista de diplomas emitidos de acordo com resultado da busca com um botão visualizar para cada item da lista;
c) **Caso de Uso - Visualizar diploma:** O administrador clicará no botão visualizar, presente em cada item da lista de diplomas. O sistema exibirá as informações do diploma e um botão para gerar PDF;
d) **Caso de Uso – Emitir Diploma:** O administrador preencherá um formulário com informações pertinentes ao diploma e clicará no botão Salvar. O sistema irá gravar as informações no Blockchain e exibirá o diploma na lista de diplomas emitidos.

4.2 Arquitetura e Tecnologias

Para o desenvolvimento da aplicação Emissão e Autenticação de Diplomas será usada uma arquitetura de aplicações descentralizadas. As aplicações descentralizadas representam um novo modelo para criar, financiar e operar serviços de software de uma maneira descentralizada. É uma forma de criar um serviço que nenhuma entidade, empresa ou órgão público, possa operá-lo. (NAKAMOTO, 2008)

Para possibilitar a construção da Aplicação de forma descentralizada, optou-se por utilizar a infraestrutura Blockchain da Ethereum, através do uso de contratos inteligentes. Em um contrato convencional, é feito um acordo entre duas ou mais partes, que se obrigam a cumprir o que foi entre elas acordado mediante determinadas condições. Os contratos inteligentes não são tão diferentes dos contratos tradicionais, exceto que são codificados e registrados digitalmente no Blockchain.

Para o desenvolvimento de contratos inteligentes no Ethereum, é utilizada a linguagem de programação Solidity. A Solidity é uma linguagem de alto nível, influenciada por outras linguagens já conhecidas no mercado como, C++, Python e JavaScript. Foi projetada para ser compilada na Ethereum Virtual Machine (EVM). É uma linguagem de tipagem estática, suportando herança, uso de bibliotecas e tipos complexos, entre outros recursos (SOLIDITY, 2016).

4.2.1 Padrão de Projeto

Seguindo um padrão convencional, as aplicações utilizando Blockchain são divididas em três camadas macros, com responsabilidades distintas. A primeira camada é a do sistema distribuído, a Blockchain propriamente dita. Dentro dela encontram-se as funcionalidades necessárias para seu funcionamento, como métodos de consenso e os protocolos de comunicação ponto a ponto.

A segunda refere-se às plataformas (Ethereum, Bitcoin, Hyperledger), que contêm os serviços de apoio à infraestrutura relacionados à gestão de chaves criptográficas, disponibilidade de nós da rede P2P, gestão de identidade, dentre outros.

A terceira é a camada de aplicação, composta pela interface de usuário, regras de negócios e também pelos contratos inteligentes.

4.2.2 Ethereum

O Ethereum é uma plataforma descentralizada que executa contratos inteligentes. Os contratos inteligentes são executados em uma Blockchain personalizada (programável), utilizando-se de uma infra-estrutura global compartilhada extremamente poderosa. Isso permite que os desenvolvedores, armazenem registros de dívidas ou promessas, movimentem fundos de acordo com instruções dadas no passado (como um testamento ou um contrato futuro) e muitas outras coisas que ainda não foram inventadas, tudo sem um intermediário ou risco da contrapartida (ETHEREUM, 2017).

4.2.3 Carteira Eletrônica (eWallet)

Carteira Eletrônica é uma ponte que permite que um usuário execute transações em uma *Blockchain* direto do navegador web, permitindo rodar aplicações descentralizadas sem executar um nó completo da infra-estrutura *Blockchain* . Uma Carteira Eletrônica inclui também um cofre de identidade seguro, fornecendo uma interface de usuário para gerenciar suas identidades em sites diferentes e assinar transações *Blockchain* (METAMASK, 2017).

4.2.4 Contrato Inteligente

Para o projeto, será implementado um contrato inteligente principal chamado de **DegreeFactory** e outro chamado **Degree.** O primeiro servirá como uma fábrica

de contratos. Através do método createDegree, será criado um novo contrato do tipo Degree na rede Ethereum. Este contrato que irá conter as informações pertinentes ao diploma a ser emitido. Segue a implementação do contrato inteligente na Figura 11.

Figura 11: Contrato Inteligente

Degree.sol

```
pragma solidity ^0.4.17;

contract DegreeFactory {
    address[] private deployedDegrees;

    function createDegree(string studentId, string studentName, uint studentRG, string courseName, string endCourseDate) public {
        address newDegree = new Degree(msg.sender, studentId, studentName, studentRG, courseName, endCourseDate);

        deployedDegrees.push(newDegree);
    }

    function getDeployedDegrees() public view returns (address[]) {
        return deployedDegrees;
    }
}

contract Degree {
    address private manager;
    string private studentId;
    string private studentName;
    uint private studentRG;
    string private courseName;
    string private endCourseDate;
    uint private issuenceDate;

    constructor(address creator, string pStudentId, string pStudentName, uint pStudentRG, string pCourseName, string pEndCourseDate) public {
        manager = creator;
        studentId = pStudentId;
        studentName = pStudentName;
        studentRG = pStudentRG;
        courseName = pCourseName;
        endCourseDate = pEndCourseDate;
        issuenceDate = block.timestamp;
    }

    function getSummary() public view returns(
        string, string, uint, string, string, uint, address
    ) {
        return(
            studentId,
            studentName,
            studentRG,
            courseName,
            endCourseDate,
            issuenceDate,
            manager
        );
    }
}
```

Fonte: Autores, 2018

Na tabela abaixo seguem as informações detalhadas do contrato:

Tabela 3: Descrição Contrato Inteligente

DegreeFactory		
Variáveis	**Tipo**	**Descrição**
deployedDegree	address[]	Um array que conterá o

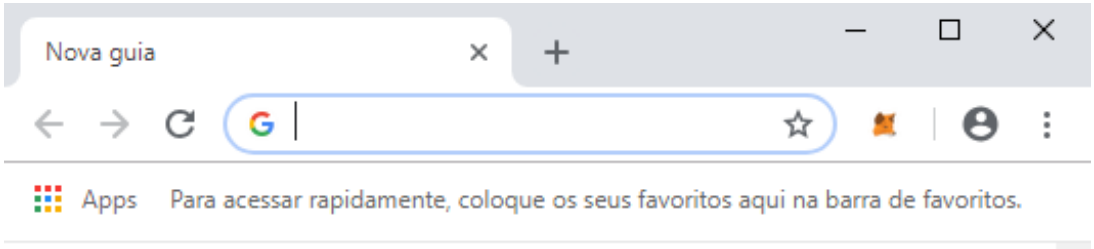
		endereço de todos os diplomas emitidos.
Funções	**Descrição**	
createDegree	Cria um novo contrato do tipo Degree no *Blockchain*	
getDeployedDegrees	Retorna o array de endereços dos contratos Degree implementados.	
Degree		
Variáveis	**Tipo**	**Descrição**
Manager	Address	Endereço hash do usuário que implementou o contrato
studentId	String	Número de matrícula do aluno
studentName	String	Nome completo do aluno
studentRG	Uint	Documento RG do aluno
courseName	String	Nome do curso que o aluno concluiu
courseEndDate	String	Data da conclusão do curso
issuenceDate	Uint	Timestamp da data de criação do bloco
Funções	**Descrição**	
Constructor	Método construtor	
getSummary	Retorna uma lista com as informações dos diplomas emitidos	

Depois de implementado na rede *Ethereum*, um contrato inteligente pode ser consumido por qualquer aplicativo através do uso de uma API. Na Aplicação para Registro de Diplomas será utilizada a web3.js. Web3.js é uma API que permite a interação com um nó da rede *Ethereum*, usando uma conexão HTTP (WEB3.JS, 2016).

Além do web3, será necessário um provedor *Ethereum*, também conhecido como carteira eletrônica (*eWallet*). Para esse projeto será utilizado o *Metamask*, que pode ser instalado como uma extensão de navegador para o *Chrome* ou *Firefox* (Figura 12). Além de ter o plug-in do *Metamask* instalado em seu navegador, o usuário precisará está *logado* ao *Metamask* (Figura 13) para poder efetuar transações de registro no *Blockchain*, como no caso de uso emissão de diplomas.

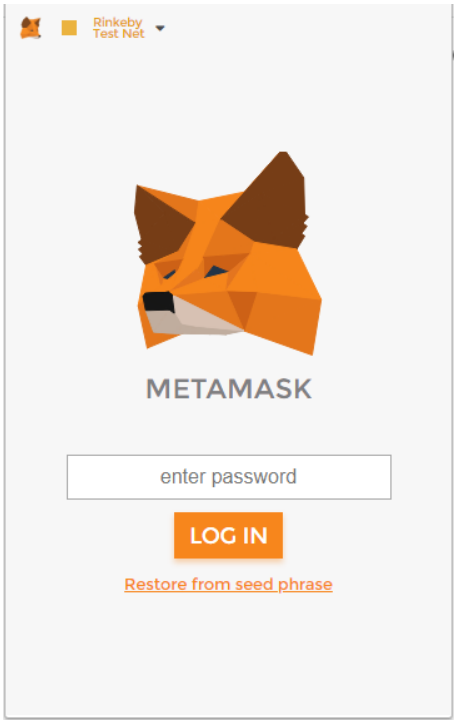
Após *logado* ao *Metamask*, o plug-in irá exibir uma conta com o respectivo código *hash* do usuário. Esse código identificará o usuário na rede. Logo abaixo também será listada as transações mais recentes para aquele usuário (Figura 14). Esse provedor permitirá ao cliente web, interagir com os contratos inteligentes e o web3 fará a comunicação entre o provedor e a aplicação. Na Figura 15 é mostrado um diagrama que exemplifica essa arquitetura.

Figura 12: Plugin Metamask Instalado



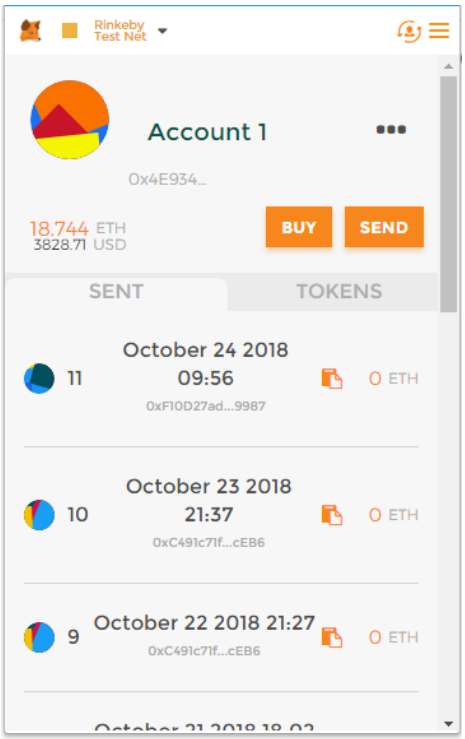
Fonte: Autores, 2018

Figura 13: Login Metamask



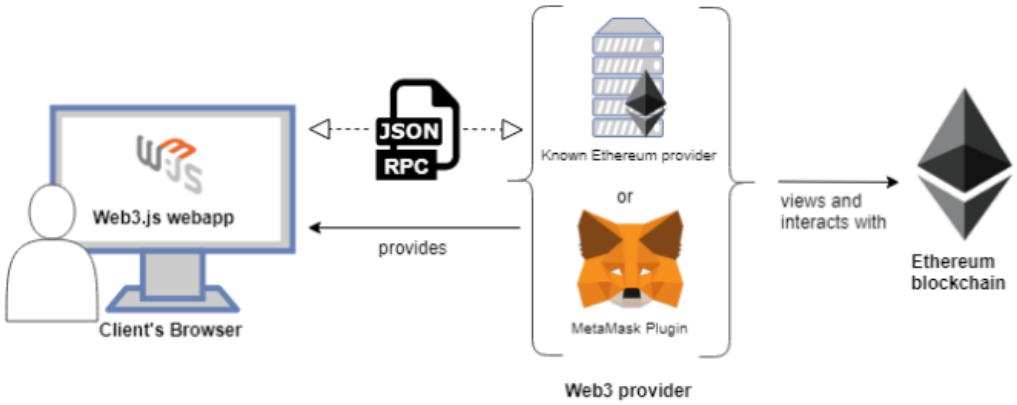
Fonte: Autores, 2018

Figura 14: Listagem de Contas Metamask



Fonte: Autores, 2018

Figura 15: Arquitetura Web3



Fonte: https://github.com/ethereumbook/ethereumbook

Na Figura 16, segue um exemplo de código de uma classe, utilizando web3 e o provedor. E na Figura 17, um exemplo de código consumindo o método getAccounts da classe eth que retorna uma lista de usuários disponíveis no provedor.

Figura 16: Classe Web3

web3.js

```
1  import Web3 from 'web3';
2
3  const web3 = new Web3(window.web3.currentProvider);
4
5  export default web3;
6
```

Fonte: Autores, 2018

Figura 17: Consumindo Web3

exemple.js

```
1  import web3 from '../ethereum/web3';
2
3  class Exemple extends Component {
4      render() {
5          const accounts = await web3.eth.getAccounts();
6          console.log(accounts[0]);
7      }
8  }
9
10 export default Exemple;
11
```

Fonte: Autores, 2018

Abaixo a descrição das funções utilizadas no código das figuras 16 e 17.

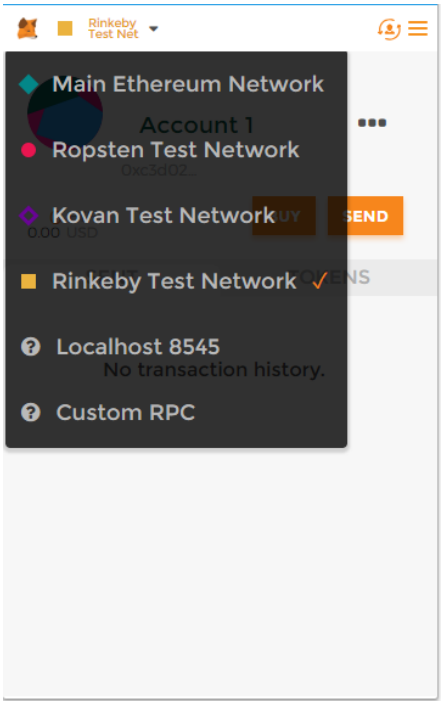
Tabela 4: Descrição Funções Web3

web3.currentProvider	Retornará o provedor atual
web3.eth.getAccounts	Retornará a lista de contas controladas pelo provedor

4.2.5 Redes Públicas do *Ethereum*

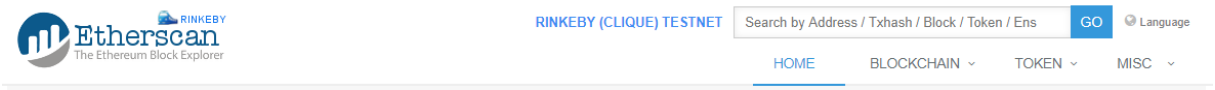
A plataforma *Ethereum* disponibiliza quatro redes públicas (Figura 18). Sendo uma a principal, que opera com uso do *Ether* e outras três redes de testes, que operam com moeda fictícia. Para este projeto, utilizaremos a rede de teste Rinkeby. Nela será implementado o contrato inteligente e será feito o monitoramento das transações. Para pesquisar as transações em um contrato, bastas informar o código hash do contrato implementado, pelo site https://rinkeby.etherscan.io/ (Figura 19) ou informando o código diretamente pela url https://rinkeby.etherscan.io/address/0x123abc.../ (Figura 20).

Figura 18: Redes Ethereum



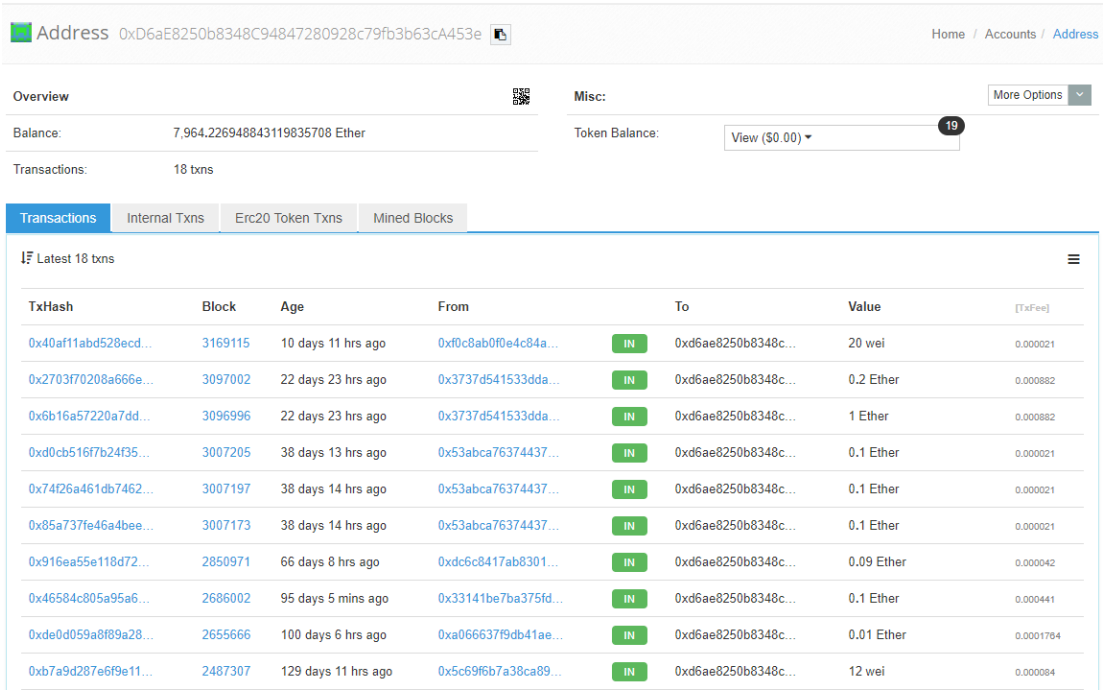
Fonte: Autores, 2018

Figura 19: Pesquisa Etherscan



Fonte: Autores, 2018

Figura 20: Listagem de Contratos Etherscan

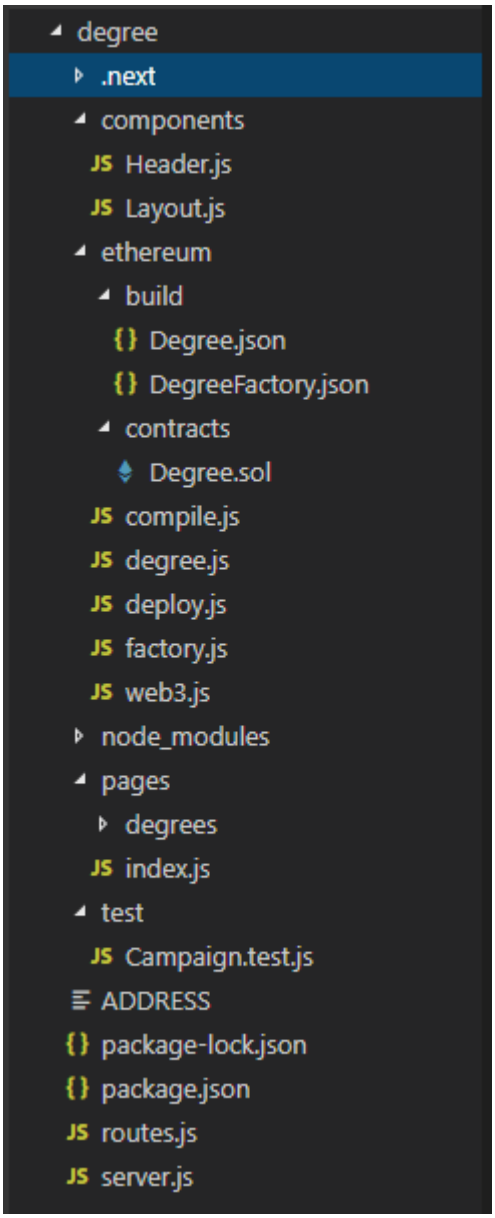


Fonte: Autores, 2018

4.2.6 Estrutura de Diretórios e outras ferramentas

Na figura 21 é apresentada a estrutura de diretórios da aplicação

Figura 21: Estrutura de Diretórios



Fonte: Autores, 2018

Abaixo uma breve descrição de cada diretório da aplicação

- **Next:** Contém a biblioteca do next.js utilizado para facilitar o mapeamento das páginas através de rotas;
- **Components:** Contém os componentes Header.js e Layout.js;
- **Ethereum:** Contém duas subpastas: **contracts**, onde fica o contrato inteligente a ser implementado e **build**, onde ficam os arquivos json dos contratos já implementados. Contém também os arquivos referentes a

compilação e implementação do contrato na Ethereum (compile.js e deploy.js);

- **Node_modules:** Contém todos os arquivos referentes ao Node.js;
- **Pages:** Contém as páginas web do aplicativo;
- **Test:** Contém uma classe de testes.

O front-end, também chamado de programação lado cliente, é tudo aquilo que os usuários enxergam e interagem (páginas web, tela desktop, tela de smartphone, etc). A programação front-end é responsável por capturar, e em alguns casos validar, os dados informados pelo usuário e entrega-los ao lado servidor (BECODE, 2017 https://becode.com.br/back-end-front-end-full-stack/).

Para o front-end da aplicação, será utilizado o React. O React é uma biblioteca Javascript de código aberto para construção de interfaces web, criada e mantida pelo Facebook. Além de ter uma grande aceitação no mercado, sendo aplicado em grandes projetos de interface como o Netflix, Airbnb, Walmart dentre outros (STACK, 2014).

Para facilitar o processo de desenvolvimento optou-se por utilizar um Ambiente de Desenvolvimento Integrado (IDE - Integrated Devlopment Environment). O IDE escolhido foi o Visual Studio Code (VS Code). O VS Code é um editor de código-fonte leve, porém robusto e multiplataforma, desenvolvido pela microsoft. Está disponível para Windows, MacOS e Linux. Ele vem com suporte embutido para JavaScript, TypeScript e Node.js e possui um rico ecossistema de extensões para outras linguagens (CODE, 2017).

4.3 Interface com o Usuário

A camada de interface com usuário será dividida em quatro telas principais, referentes a cada caso de uso. Para criação das telas optou-se por utilizar a biblioteca de interface React, desenvolvida e mantida pelo Facebook. O Ract notabiliza-se por fornecer uma forma desacoplada de criar interfaces HTML através de componentes, tornando o desenvolvimento mais ágil de simples manutenção.

Para estrutura básica do site foram criados os componentes Header (Figura 22) e Layout (Figura 23). No componente Header, estará o menu suspenso de navegação e no componente Layout se encontrará a chamada para folha de estilos

e a base estrutural do layout das páginas. Ambos servirão como uma espécie de página mestra que será chamado em todas as páginas, tornando a navegação padronizada e sem repetição de código.

Figura 22: Componente Header

Header.js

```
1   import React from 'react';
2   import { Menu } from 'semantic-ui-react';
3   import { Link } from '../routes';
4
5   export default () => {
6       return (
7           <Menu>
8               <Link route="/">
9                   <a className="item"> Sistema Emissão Diplomas</a>
10              </Link>
11
12              <Menu.Menu position="right">
13                  <Link route="/">
14                      <a className="item">Diplomas</a>
15                  </Link>
16                  <Link route="/degrees/new">
17                      <a className="item">Emissão</a>
18                  </Link>
19                  <Link route="/degrees/auth">
20                      <a className="item">Validação</a>
21                  </Link>
22              </Menu.Menu>
23          </Menu>
24      )
25  }
26
```

Fonte: Autores, 2018

Figura 23: Componente Layout

Layout.js

```
import React from 'react';
import { Container } from 'semantic-ui-react';
import Head from 'next/head';
import Header from './Header';

export default (props) => {
    return (
        <div>
            <Head>
                <link
                    rel="stylesheet"
                    href="//cdnjs.cloudflare.com/ajax/libs/semantic-ui/2.3.3/semantic.min.css"
                />
            </Head>
            <Header />
            <Container>
                {props.children}
            </Container>
        </div>
    );
}

```

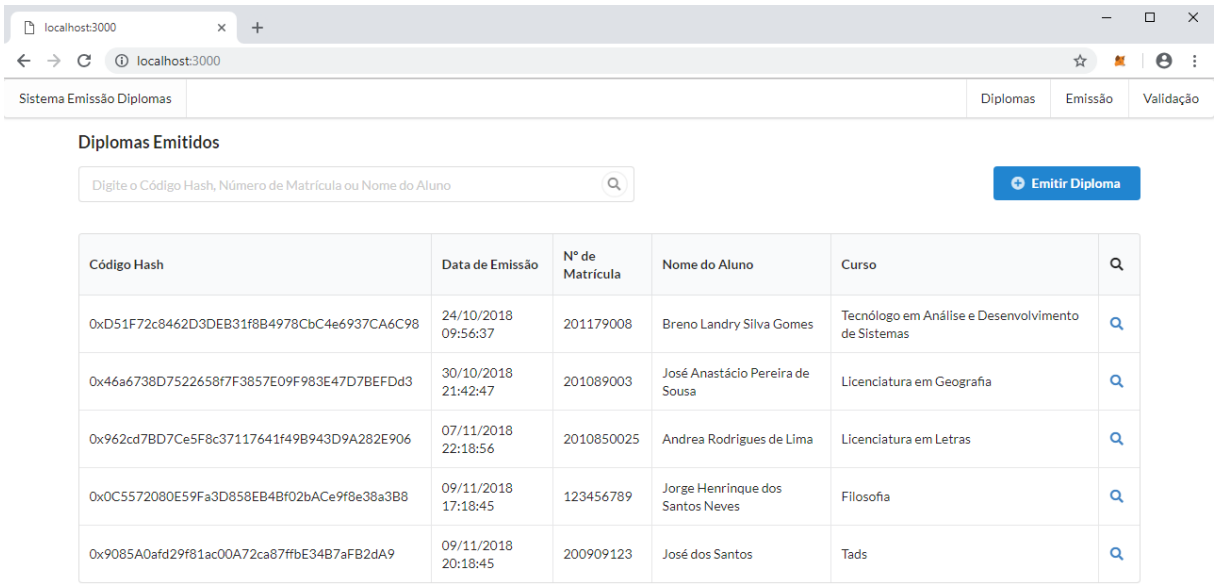
Fonte: Autores, 2018

Juntamente com React, será feito o uso da biblioteca de estilos Semantic UI. A Semantic UI, através de sua folha de estilos, fornece uma série de componentes prontos do próprio react, seguindo os padrões de boas práticas.

4.3.1 Tela Inicial (Diplomas Emitidos)

Na tela inicial, diplomas emitidos, conforme mostrado na Figura 24, é exibido uma lista com os diplomas emitidos. Para cada diploma emitido, haverá uma opção de visualizar o diploma. Acima do grid um botão com a opção Emitir Diploma.

Figura 24: Diplomas Emitidos

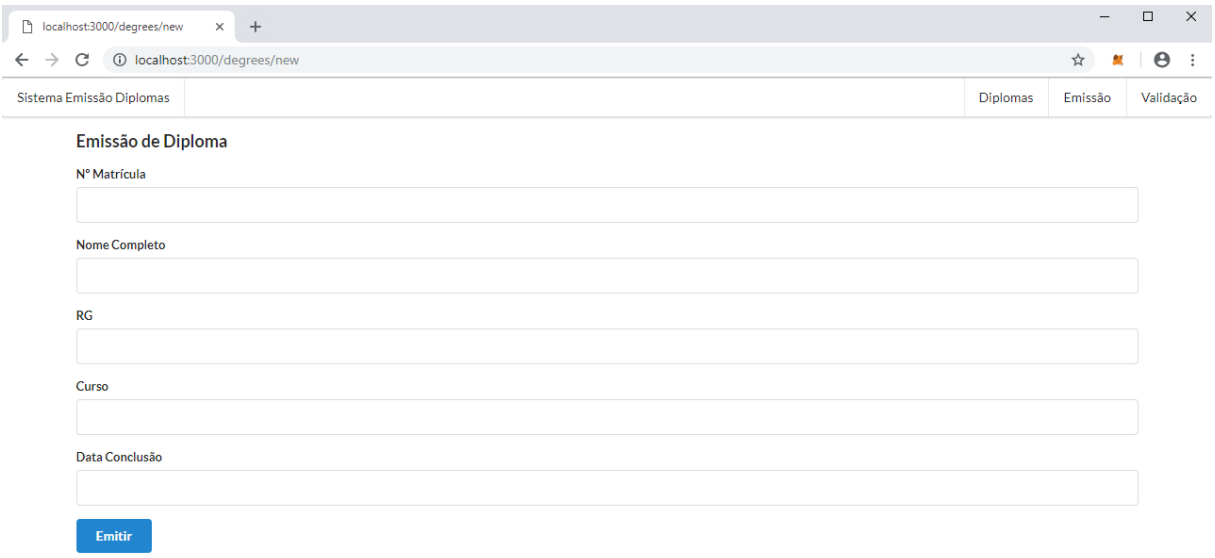


Fonte: Autores, 2018

4.3.2 Tela Emitir Diploma

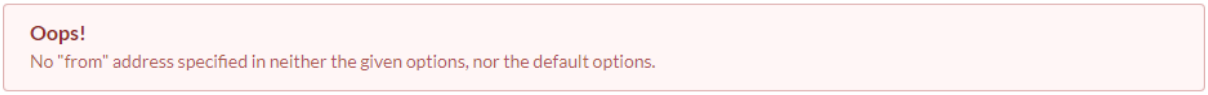
Na tela emitir diploma, será exibido um formulário onde o usuário deverá informar os dados para emissão do diploma (Figura 25). Ao clicar no botão Emitir, a aplicação se conectará ao Metamask. Caso o usuário não esteja logado ao Metamask será exibida uma mensagem de erro (Figura 26). Se o usuário estiver logado ao Metamask, um *popup* será aberto solicitando a confirmação da transação (Figura 27). Caso o usuário rejeite a transação, uma mensagem de erro será exibida (Figura 28). Caso o usuário confirme a transação, a aplicação enviará as informações contidas no formulário para o *Blockchain* . A operação levará alguns segundos até ser validada pelo *Blockchain* . Depois de feita a validação, o bloco será gerado e o usuário será redirecionado para a página inicial de diplomas emitidos, que já exibirá o novo diploma.

Figura 25: Emitir Diploma



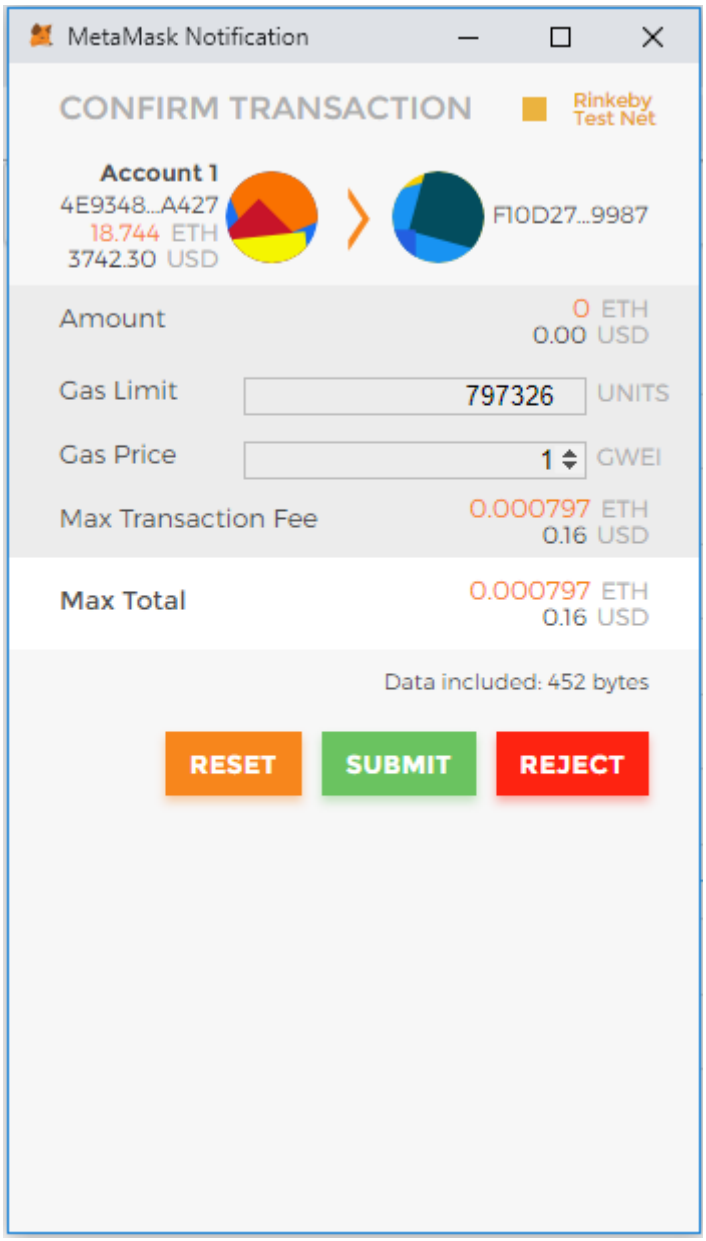
Fonte: Autores, 2018

Figura 26: Mensagem de erro Metamask



Fonte: Autores, 2018

Figura 27: Confirmação Transação Metamask



Fonte: Autores, 2018

Figura 28: Mensagem Erro Transação Rejeitada

Fonte: Autores, 2018

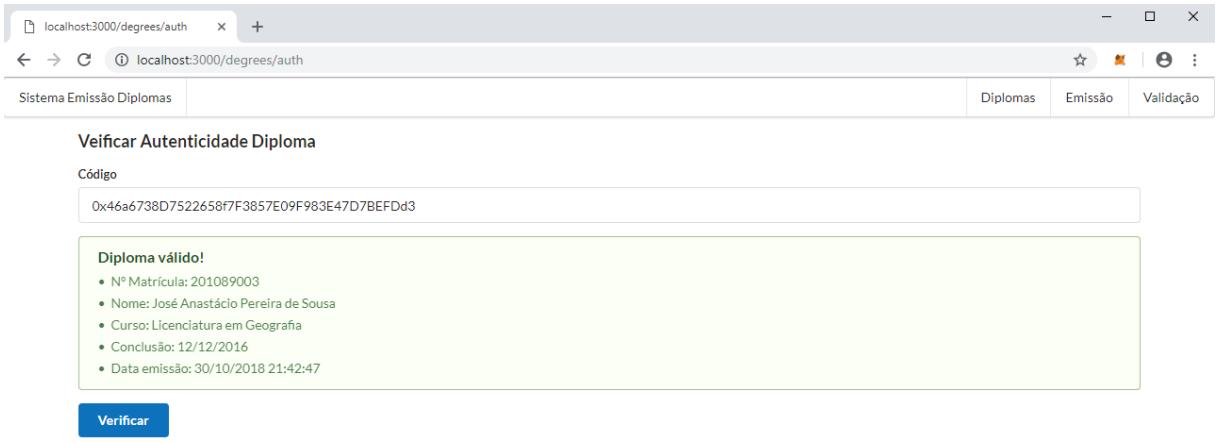
4.3.3 Tela Validar Diploma

Na tela validar diploma (Figura 29), será exibido um campo de texto onde o usuário irá informar o código do diploma, correspondente ao código hash do bloco implementado na *Blockchain* . Após clicar no botão Validar, a aplicação fará a consulta no *Blockchain* pelo código hash informado. Caso o diploma exista, uma mensagem de sucesso será exibida com as informações do diploma (Figura 30). Caso contrário uma mensagem de alerta será exibia para o usuário notificando-o de que o código informado não corresponde a nenhum diploma registrado (Figura 31).

Figura 29: Validar Diploma

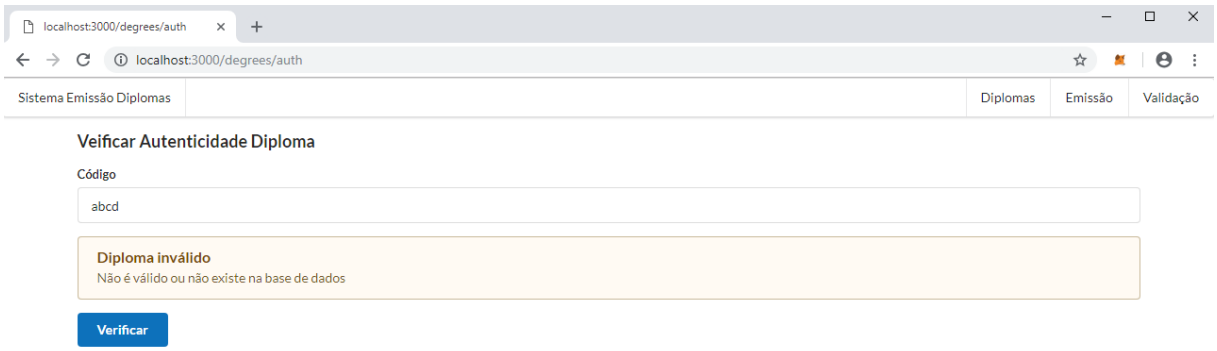
Fonte: Autores, 2018

Figura 30: Diploma Validado



Fonte: Autores, 2018

Figura 31: Diploma Não Validado

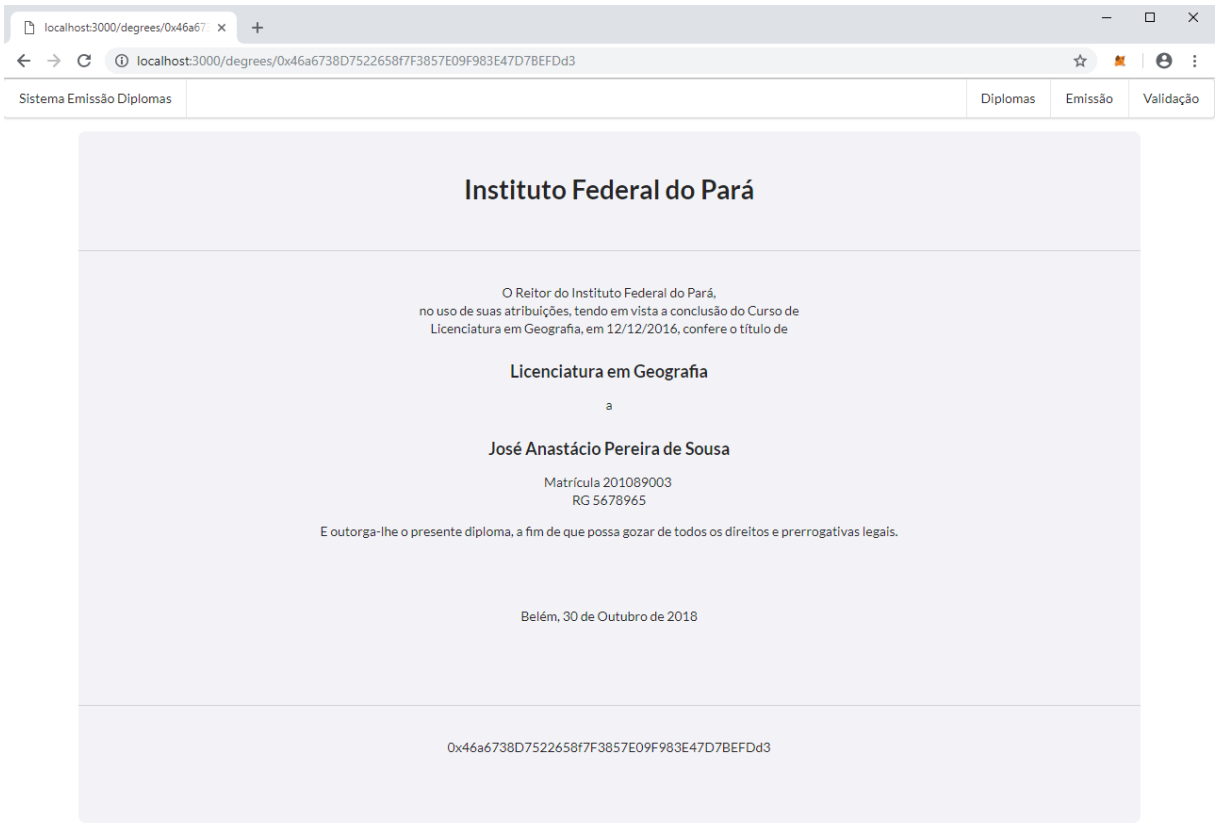


Fonte: Autores, 2018

4.3.4 Tela Visualizar Diploma

Na tela visualizar diploma (Figura 32), o usuário visualizará todas as informações referentes ao diploma selecionado. No topo estará o nome da instituição de ensino (Instituto Federal do Pará). No corpo do texto estarão, o nome do curso, a data de conclusão, o nome do aluno, número de matrícula, RG e a data de emissão por extenso. E no rodapé, o código hash gerado pelo *Blockchain* , que servirá como um código verificador. Todos os campos são recuperados dinamicamente direto do *Blockchain* .

Figura 32: Visualizar Diploma



Fonte: Autores, 2018

5 CONCLUSÃO

Ao iniciar esta pesquisa, constatou-se que o *Blockchain* é uma nova e importante tecnologia, cheia de possibilidades de uso, e de fácil acesso. Esta tecnologia proporciona variadas possibilidades para aplicações em diversas áreas, o que foi de extrema importância para o desenvolvimento deste projeto.

Ao aprofundar nossa pesquisa, descobrimos que esta tecnologia já está em pleno uso em diversos lugares, por diversas instituições, sejam elas, públicas ou privadas, no setor de logística, alimentos, entretenimento, e que já havia várias plataformas proporcionando um ambiente para desenvolvimento de aplicações com o *Blockchain*.

O desenvolvimento do protótipo proporcionou um aprofundamento dos conhecimentos adquiridos durante o curso, em matérias como redes de computadores, IHC, linguagens de programação, sistemas distribuídos e também agregou conhecimentos novos, no campo da criptografia simétrica e assimétrica e também no trabalho com novas tecnologias.

Infelizmente, devido ao tempo limitado para desenvolver este trabalho, não tivemos a oportunidade de implementá-lo de forma mais completa, porém foi projetado e desenvolvido um protótipo funcional, onde estão todas as bases do sistema. Encontramos na Plataforma *Ethereum*, uma ferramenta bem estruturada e eficaz para a implementação deste protótipo e também para pôr em prática outros tipos de projetos utilizando a rede *Blockchain* da plataforma.

Com o desenvolvimento deste protótipo, ficou claro que a tecnologia *Blockchain* é bastante abrangente, não se restringindo somente ao mercado financeiro e de criptomoedas. Com o uso dos Contratos Inteligentes, torna-se possível a utilização do *Blockchain* como um grande banco de dados descentralizado, possibilitando a construção de aplicações descentralizadas de forma mais robusta, utilizando toda a infraestrutura disponibilizada pelo *Ethereum*.

Este trabalho contribui para a comunidade acadêmica, pois, um sistema como este, poderia trazer muitos benefícios para a Instituição, como segurança e maior agilidade na validação e emissão de documentos oficiais, além de ser uma alternativa simples para a verificação da veracidade de documentos expedidos pela

instituição, se tornando assim, uma ferramenta contra a fraude de diplomas, atestados de vínculo e históricos.

Como uma forma de contribuir com instituição de ensino, deixamos como sugestão de trabalhos futuros:

- O desenvolvimento de uma aplicação completa para registro e autenticação de documentos utilizando o *Blockchain Ethereum* e Contratos Inteligentes;
- Estudo sobre vulnerabilidade de segurança em sistemas com *Blockchain*;
- *Blockchain* aplicado à internet das coisas;
- *Blockchain* aplicado às mídias sociais.

REFERÊNCIAS

BRAGA, A. M. Fundamentos,Tecnologia de Segurança e Desenvolvimento de Software. **Tecnologia Blockchain**, Campinas - São Paulo, 2017. Disponivel em: <https://www.cpqd.com.br/wp-content/uploads/2017/09/whitepaper_blockchain_fundamentos_tecnologias_de_seguranca_e_desenvolvimento_de_softwar_FINAL.pdf>. Acesso em: 11 Dezembro 2017.

BRAGA, A. M.; MARINO, F. C. H.; SANTOS, R. R. D. **Segurança de Apliações Blockchain Além das Criptomoedas.** Simpósio Brasileiro em Segurança da Informação e Sistemas Computacionais - SBSeg- 2017. Brasilia: Sociedadde Brasileira de Computação - SBC. 2017. p. 99-148.

BRAGA, A.; DAHAB, R. **Introdução à Criptografia para Programadores:** Evitando Maus Usos da Criptografia em Sistemas de Software. Caderno de minicursos do XV Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais— SBSeg. [S.l.]: [s.n.]. 2015. p. 1-50.

BRASIL. Art. 1º da Medida Provisória ,n. 2.200-2 de 24 de Agosto de 2001. **Criação da Infraestrutura de Chaves Públicas Brasileiras - ICP -Brasil.**, Brasilia-DF, 24 Agosto 2001.

BUTERIN, V. DAOs, DACs, DAs and More: An Incomplete Terminology Guide. **Ethereum**, 2014. Disponivel em: <https://blog.ethereum.org/2014/05/06/daos-dacs-das-and-more-an-incomplete-terminology-guide/>. Acesso em: 20 Setembro 2018.

CHICARINO, V. R. L. et al. **Uso do Blockchain para Privacidade e Segurança em Internet das Coisas**. XVII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais — SBSeg 2017. Brasilia - DF: Sociedade Brasileira de Computação - SBC. 2017. p. 149-200.

CODE, V. **Documentation for Visual Studio Code**. Disponivel em: <https://code.visualstudio.com/docs>. Acesso em: 16 Outubro 2018.

DORNELES, S. L.; CORRÊA, R. S. GESTÃO DE DOCUMENTOS DIGITAIS EM APLICAÇÕES DE CERTIFICAÇÃO DIGITAL. **Informação Arquivística**, Rio de Janeiro, Jul.\Dez. 2013. 3-31.

ETHEREUM. Ethereum - Blockchain App Plataform. **Ethereum.org**, 2017. Disponivel em: <https://www.ethereum.org/>. Acesso em: 05 Setembro 2018.

FOMIGONI FILHO, R. J.; BRAGA, A. M.; LEAL, R. L. V. Uma Visão Geral. **Tecnologia Blockchain**, Campinas- Sâo Paulo, 2017. 30. Disponivel em: <https://www.cpqd.com.br/wp-content/uploads/2017/03/cpqd-whitepaper-blockchain-impresso.pdf>. Acesso em: 19 Março 2018.

GREVE, F. et al. **Blockchain e a Revolução do Consenso sob Demanda**. XXXVII Simpósio Brasileiro de Redes de Computadores - SBRC 2018. [S.l.]: [s.n.]. 2018.

ICP-BRASIL, I. D. C. P. B. **Visão Geral dos Sistemas de Carimbos de Tempo na ICP-BRASIL**. Infraestrutura de Chaves Públicas Brasileiras -ICP/ BRASIL. [S.l.], p. 12. 2015.

ITI, I. N. D. T. D. I.-. Certificado Digital. **ITI - Instituto Nacional de Tecnologia da Informação**, 2017. Disponivel em: <https://www.iti.gov.br/certificado-digital>. Acesso em: 27 set. 2018.

LEAL, R. L. V. Blockchain e Internet das Coisas: Aplicações e Iniciativas. **Tecnologia Blockchain**, Campinas - São Paulo, 2017. Disponivel em: <https://www.cpqd.com.br/wp-content/uploads/2017/11/Whitepaper_Blockchain_e_IoT_-_aplicac%CC%A7o%CC%83es_e_iniciativas_v2.pdf>. Acesso em: 19 Março 2018.

LUU, L. et al. **Making Smart Contracts Smarter**. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. [S.l.]: [s.n.]. 2016. p. 254-269.

METAMASK. https://metamask.io/. **Metamask**, 2017. Disponivel em: <https://metamask.io/>. Acesso em: 01 Outubro 2018.

MOECKE, C. T. **Assinatura Digital de Documentos Eletrônicos na ICP-Brasil.** Florianópolis: [s.n.], 2008.

NAKAMOTO, S. A Peer-to-Peer Electronic Cash System. **Bitcoin.org**, 2008. Disponivel em: <https://bitcoin.org/bitcoin.pdf>.

NUNES, J. M. G.; CASTELLO BRANCO, A. Impactos da Plataforma P2P na Economia do Compartilhamento. **Revista Brasileira de Opinião de Marketing, Opinião e Mídia - PMKT**, Rio de Janeiro, v. 11, p. 222-235, 08 Fevereiro 2018. Disponivel em: <http://www.revistapmkt.com.br/Portals/9/Revistas/v11n2/7%20-%20Impactos%20das%20plataformas%20P2P%20na%20economia%20do%20compartilhamento%20-%20Ensaio.pdf>. Acesso em: 02 out. 2018.

OLIVEIRA, R. R. Criptografia simétrica e assimétrica-os principais algoritmos de cifragem. **Segurança Digital [Revista Online]**, v. 31, p. 11-15, 2012.

PINHEIRO, P. P. **Direito Digital**. 4º. ed. São Paulo: Saraiva, v. 2º , 2010.

RIBEIRO, S. L. Tecnologia Blockchain : Aplicações e Iniciativas. **Centro de Pesquisa e Desenvolvimento em Telecomunicações - CPqD**, Campinas - São Paulo, 2017. Disponivel em: <https://www.cpqd.com.br/wp-content/uploads/2017/09/whitepaper_aplicacoes_e_iniciativas_final.pdf>. Acesso em: 19 Março 2018.

SOLIDITY. **Solidy 0.4.25 Documentation**, 2016. Disponivel em: <https://solidity.readthedocs.io/en/v0.4.25/index.html>. Acesso em: 05 Setembro 2018.

SOMMERVILLE, I. **Engenharia de Software**. 9º. ed. São Paulo: Pearson Prentice Hall, 2011.

STACK, T. N. **JavaScript's History and How it Led To ReactJS**. Disponivel em: <https://thenewstack.io/javascripts-history-and-how-it-led-to-reactjs/>.>. Acesso em: 15 Outubro 2018.

STALLINGS, W. **Criptografia e Segurança de Redes:** Principios e Práticas. 4ª. ed. São Paulo: Pearson Education, v. 3, 2008.

SZABO, N. The Idea of Smart Contracts. **Nick Szabo's Papers and Concise Tutorials**, 1997. Disponivel em: <http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/idea.html>. Acesso em: 10 Setembro 2018.

VIEIRA, R. M.; ARAÚJO, W. J. D. **Assinatura de documentos eletrônicos utilizando certificados digitais:** estudo de caso de assinaturas digitais aplicadas em atas de reuniões. Encontro Regional dos Estudantes de Biblioteconomia,Documentação, Ciência e Gestão da Informação - EREBDN/NE. Juazeiro do Norte: [s.n.]. 2012.

VIVIAN, D. Carimbo de Tempo versus Timestamp. **BRY Tecnologia**, 2018. Disponivel em: <https://www.bry.com.br/blog/carimbo-do-tempo-timestamp/>. Acesso em: 05 Outubro 2018.

WEB3.JS. **web3.js - Ethereum JavaScript API**, 2016. Disponivel em: <https://web3js.readthedocs.io/en/1.0/>. Acesso em: 05 Setembro 2018.