UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE QUIXADÁ
ENGENHARIA DE SOFTWARE

GABRIEL OLIVEIRA MENDANHA

ASSEGURANDO A PROPRIEDADE E IMUTABILIDADE DE DOCUMENTOS DIGITAIS: UMA PROVA DE CONCEITO UTILIZANDO BLOCKCHAIN

QUIXADÁ - CE
2017

GABRIEL OLIVEIRA MENDANHA

ASSEGURANDO A PROPRIEDADE E IMUTABILIDADE DE DOCUMENTOS DIGITAIS: UMA PROVA DE CONCEITO UTILIZANDO BLOCKCHIAN

Trabalho de conclusão de curso submetido à Coordenação do Curso Bacharelado em Engenharia de *Software* da Universidade Federal do Ceará como requisito parcial para obtenção do grau de bacharel.
Área de concentração: Engenharia de *Software* / Desenvolvimento / *Blockchain.*

Orientadora: Profa. Ma. Lívia Almada Cruz Rafael.
Coorientador: Prof. Me. Régis Pires Magalhães.

QUIXADÁ - CE
2017

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

M488a Mendanha, Gabriel Oliveira.
Assegurando a propriedade e imutabilidade de documentos digitais : uma prova de conceito utilizando blockchain / Gabriel Oliveira Mendanha. – 2017.
55 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Engenharia de Software, Quixadá, 2017.
Orientação: Profa. Ma. Lívia Almada Cruz Rafael.
Coorientação: Prof. Me. Régis Pires Magalhães.

1. Software-Desenvolvimento. 2. Blockchain. 3. Engenharia de Software. I. Título.

CDD 005.1

GABRIEL OLIVEIRA MENDANHA

ASSEGURANDO A PROPRIEDADE E IMUTABILIDADE DE DOCUMENTOS DIGITAIS: UMA PROVA DE CONCEITO UTILIZANDO BLOCKCHIAN

Trabalho de conclusão de curso submetido à Coordenação do Curso Bacharelado em Engenharia de *Software* da Universidade Federal do Ceará como requisito parcial para obtenção do grau de bacharel.
Área de concentração: Engenharia de *Software* / Desenvolvimento / *Blockchain.*

Aprovado em: __/__ /____.

BANCA EXAMINADORA

Profa. Ma. Lívia Almada Cruz Rafael

Orientadora

Prof. Me. Régis Pires Magalhães

Coorientador

Prof. Me. Victor Aguiar Evangelista de Farias

Universidade Federal do Ceará

Dedico este trabalho à minha família, namorada e amigos.

RESUMO

Após a criação da criptomoeda Bitcoin, com o estudo e pesquisa da tecnologia que tornou-a possível, a *blockchain,* viu-se a possibilidade de utiliza-lá para outros casos de uso não relacionados a transações financeiras ou dinheiro digital. Motivado por uma realidade que pessoas facilmente adulteram, copiam e corrompem informações digitais, este trabalho apresenta uma prova de conceito utilizando a *blockchain,* tirando proveito das características de imutabilidade e propriedade dos dados. Voltado para documentos digitais, abre uma gama de casos de uso e possibilidades para uma variedade de consumidores que desejam uma maneira de provar a autenticidade e posse, assim como transferi-la a outrem.

Palavras chave: Desenvolvimento de *Software*, *Blockchain*, Engenharia de *Software*.

ABSTRACT

The blockchain is the technology that made Bitcoin cryptocurrency possible. Its study and research have grown up after the creation of Bitcoin. There was the possibility to use the blockchain for other use cases that are not related to financial transaction ou digital cash. Motivated by a reality where people easily adulterate, copy and corrupt digital information, this work presents a proof of concept utilizing blockchain, taking advantage of the tamper resistant and property of data characteristics. Focused on digital documents, it opens a variety of use cases and possibilities for those who wish a manner to prove the authenticity and ownership of digital documents or transfer them to someone else.

Keywords: Software Development, Blockchain, Software Engineering.

LISTA DE FIGURAS

LISTA DE QUADROS

SUMÁRIO

1. INTRODUÇÃO

Em 2008 foi publicado o primeiro dinheiro digital bem sucedido, o Bitcoin (NAKAMOTO; SATOSHI[1], 2008). Uma combinação de fatores permitiu essa moeda ser difundida, além de ser um ativo digital, ela também é um sistema de pagamento completamente descentralizado e que permite transações financeiras seguras em uma rede *peer-to-peer* com participantes não-confiáveis, sem depender de uma autoridade central. A descentralização é uma dentre muitas características que foram incorporadas graças a tecnologia *blockchain*, que é fundamentalmente um banco de dados transparente e descentralizado que contém o registro de todas as transações. O Bitcoin é o protocolo que opera sobre este banco de dados.

Com a popularização do Bitcoin, viu-se o potencial da *blockchain* para remodelar a sociedade, pois antes confiávamos em pessoas para intermediar transações, mas agora é possível realizar tal tarefa com o uso da matemática. Logo, outras organizações adaptaram a tecnologia de acordo com as suas necessidades, surgindo uma gama de aplicações que garantem o seu bom funcionamento, integralidade, imutabilidade e disponibilidade dos dados, sem a dependência de um supervisor humano ou servidor centralizado. A *blockchain* pode ser utilizada de forma pública, que permite que todos olhem os dados de todos, além de possibilitar que qualquer um solicite a inclusão de um dado à *blockchain;* ou de forma privada, possibilitando um maior controle na inclusão de novas informações, restringindo o acesso a leitura e escrita a determinados grupos ou indivíduos de interesse.

Este trabalho tem como objetivo desenvolver, utilizando a tecnologia da *blockchain* pública, uma ferramenta que permitirá às pessoas uma maneira irrefutável de provar a posse e autenticidade de determinado documento digital, assim como transferi-lo a outra pessoa. Oferecerá também um meio para armazenar o documento com a garantia de que o mesmo não poderá ser modificado ou removido pelo dono, por outra pessoa, ou pelo administrador do sistema. A plataforma proposta tem em seu núcleo o BigchainDB, um banco de dados distribuído e descentralizado que incorporou as características da *blockchain* sem perda de escalabilidade (TRENT MCCONAGHY, ET AL. 2016.), que garantiria para o sistema bom desempenho, mesmo que fosse utilizado em larga escala.

---

[1] Satoshi Nakamoto é um pseudônimo. Até a conclusão deste trabalho não se sabe a identidade da pessoa ou organização responsável pela criação do Bitcoin.

O sistema proposto tem como público-alvo pessoas que querem uma forma de garantir o reconhecimento como autor de uma obra intelectual, como: artistas, compositores, pesquisadores, etc. Assim, bem como pessoas de negócio, que podem transferir, por exemplo, a escritura de uma casa para outra pessoa de forma ágil e segura. Também pode ser usado por entidades que desejam combater fraude de documentos sensíveis, como: documentos de identidade pessoal, prontuários médicos, diplomas, etc.

Este trabalho está organizado na seguinte forma. No Capítulo 2 são apresentados os objetivos gerais e específicos. No Capítulo 3 são apresentados os conceitos e tecnologias fundamentais deste trabalho. Os trabalhos que possuem uma proposta de alguma forma relacionada com este trabalho estão no Capítulo 4. O sistema desenvolvido está descrito em detalhes no Capítulo 5. O Capítulo 6 detalha a avaliação realizada do *software.* A conclusão está no Capítulo 7.

2. OBJETIVOS

Neste capítulo será apresentado os principais objetivos que este trabalho pretende satisfazer.

2.1. Objetivo geral

Este trabalho tem como objetivo apresentar uma plataforma que permita rastrear qual arquivo digital pertence a qual pessoa, utilizando a tecnologia *blockchain*.

2.2. Objetivos específicos

Para satisfazer o objetivo geral, será necessário atingir os seguintes objetivos específicos:

- Viabilizar o armazenamento de documentos;
- Estabelecer um vínculo de posse entre indivíduo e documentos;
- Viabilizar a transferência de posse de documentos;
- Manter a integridade e imutabilidade dos documentos armazenados;
- Permitir a validação de documentos.

3. FUNDAMENTAÇÃO TEÓRICA

Nas seções a seguir são apresentados os conceitos e tecnologias principais deste trabalho. A Seção 3.1 explica a *blockchain* e suas principais características. Nas seções 3.2 e 3.3 estão detalhados as duas principais tecnologias utilizadas neste trabalho, o BigchainDB e o IPFS, respectivamente.

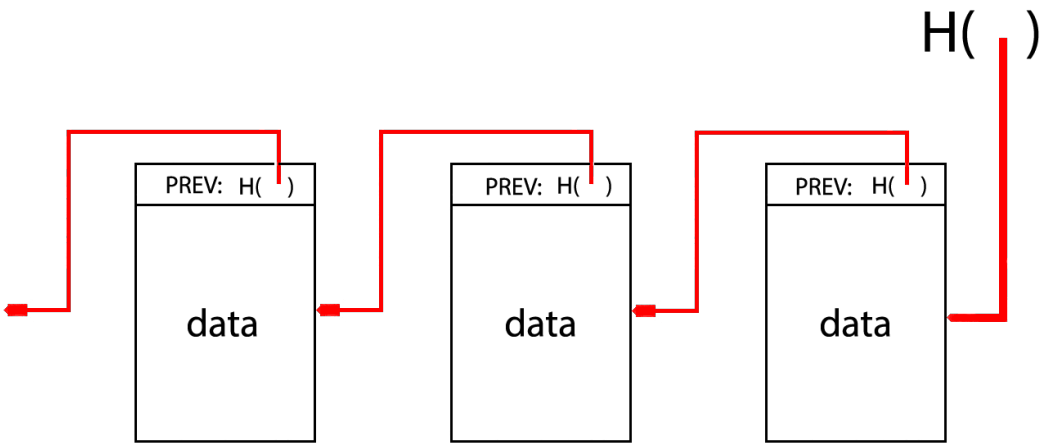
3.1. Blockchain

De acordo com Antonopoulos (2016), *blockchain* é uma lista encadeada e ordenada de blocos que contém transações como dado principal. Neste trabalho, entende-se por “transação" um objeto JSON que é utilizado para criar ou transferir um ativo digital qualquer. Cada bloco referencia o seu anterior e os mesmos são identificados pelo resultado de uma função *hash* criptográfica que mapeia um dado de tamanho variável para um dado de tamanho fixo. Para a *blockchain,* é desejável que a função *hash* produza dois resultados iguais se, e somente se, as entradas forem as mesmas. Portanto, deve existir um valor *hash* diferente para cada bloco.

A Figura 1 ilustra este conceito, observe que o bloco é representado pelo retângulo. H é uma função *hash* qualquer que recebe como entrada o bloco anterior, indicado pela seta vermelha. Logo, cada bloco sabe quem é o seu anterior através da *hash* obtida como resultado de H.

O objetivo de cada bloco armazenar a *hash* do bloco anterior é para possibilitar a detecção de modificações em qualquer ponto da *blockchain*, pois qualquer alteração no bloco resultará em uma *hash* diferente da que está armazenada no bloco posterior.

Figura 1 - Estrutura da blockchain



Fonte: Adaptado de Cryptoccurencies for Everyone (2016)

O Institute of International Finance (2015) define a *blockchain* como um sistema de consenso distribuído que permite verificar e manter transações de forma rápida e segura. Para isso, a *blockchain* utiliza criptografia e o poder computacional das máquinas conectadas à rede, sendo assim desnecessária a confiança em uma autoridade central. Devido ao fato da *blockchain* conter informações com data e hora, ser irreversível e replicada em computadores ao redor do mundo, não existe um ponto único de falha[2] (INSTITUTE OF INTERNATIONAL FINANCE, 2015). É importante ressaltar que a *blockchain* é muito conhecida no contexto de moedas digitais. Porém, ela não é tecnicamente dependente destas. Outras aplicações já estão utilizando-a de acordo com as suas necessidades, como: Ethereum[3] ou Hyperledger[4].

François Zaninotto (2016) explica que a *blockchain* é um registro de fatos que estão replicados em diversos computadores em uma rede *peer-to-peer*. Os membros da rede são indivíduos de identidade desconhecida, denominados de nós. Toda comunicação entre um nó emissor e um nó receptor dentro da rede utiliza-se de criptografia para identificá-los de forma segura.

De acordo com o Dicionário Online de Português, fato é algo que aconteceu ou que está prestes a ocorrer. Na *blockchain,* um nó pode desejar adicionar um fato, então um acordo

[2] Tradução livre do inglês, *single point of failure,* que designa um ponto em um sistema que caso falhe, resulta na falha de todo o sistema.

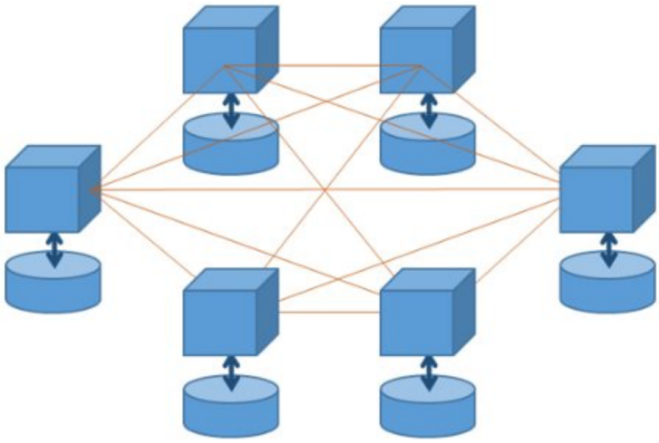
[3] https://www.ethereum.org

[4] https://www.hyperledger.org

é formado na rede para determinar onde ele deve aparecer na *blockchain.* Este fato recebe o nome de bloco, pois assim como um fato, um bloco não pode ser alterado, seja o conteúdo que carrega, as transações, ou os metadados.

A Figura 2 representa visualmente a definição de François Zaninotto (2016). Observe que cada nó é representado como um cubo e está conectado com vários outros nós, assim como em uma rede *peer-to-peer*, e que cada nó possui a sua cópia da *blockchain*, que é representado por um cilindro. O tipo de replicação, total ou parcial, depende dos requisitos específicos do sistema.

Figura 2 - *Blockchain* em uma rede *peer-to-peer*



Fonte: Can We Reach Consensus on *Blockchain* (2016)?

Ao longo desta seção, serão discutidos os conceitos fundamentais de toda *blockchain*, e para fins didáticos a *blockchain* será exemplificada da forma que foi implementada por Satoshi Nakamoto, devido ao fato de ser a implementação mais popular e consolidada (JOSEPH YOUNG, 2016). Detalhes específicos da implementação foram omitidos.

3.1.1. Características

Nas seções a seguir serão discutidos alguns aspectos inerentes a todas as *blockchains.*

3.1.1.1. Distribuição, integridade e segurança

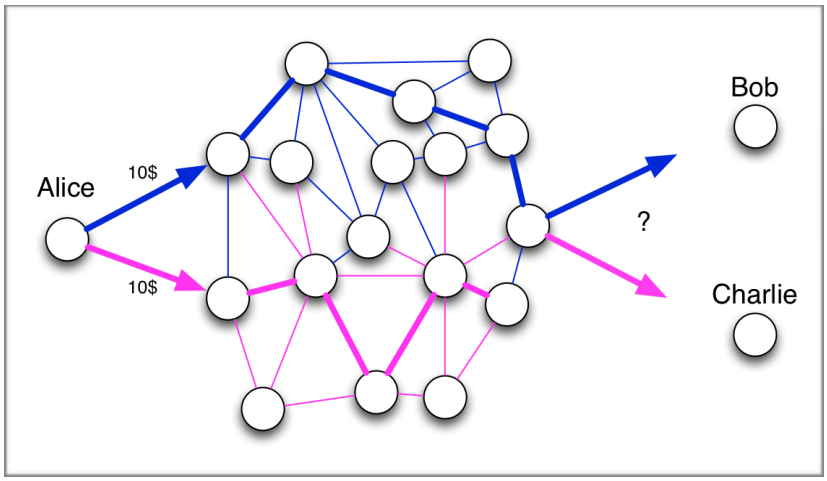
Quanto à distribuição da *blockchain* na rede, Peter Evans-Greenwood et al. (2016) afirmam que, considerando o contexto da tecnologia da informação, um livro-razão é um

mecanismo de armazenamento que permite apenas inserções. Em um livro-razão, as informações são imutáveis e podem conter dados genéricos, portanto a *blockchain* pode ser vista como uma tecnologia de livro-razão. Na *blockchain*, não existe um agente central responsável pelo gerenciamento do sistema. Portanto, torna-se necessário que a *blockchain* seja replicada entre vários nós em uma rede *peer-to-peer* (CLEARMATICS, 2015).

O que está sendo distribuído é a responsabilidade de decidir o que incluir, em que ordem incluir e de garantir que uma vez incluído, o registro não possa ser alterado. Um grupo de nós, através de um consenso, dividem essa responsabilidade.

A *blockchain* deve manter as informações íntegras, porém assim como em qualquer outro sistema distribuído, a *blockchain* possui o problema de resolução de conflitos. Se dois fatos incompatíveis chegarem no mesmo instante, o sistema deve possuir regras que determinem qual dos fatos será considerado válido. A Figura 3 exemplifica o problema de resolução de conflitos. Nesta Figura, Alice envia $10 para Bob e os mesmos $10 para Charlie. O problema está no fato de que Alice possui somente $10 e está tentando gastar duas vezes este valor. Uma maneira de resolver este problema é ordenando os fatos, o primeiro que for registrado é o vencedor.

Figura 3 - O problema do gasto duplo



Fonte: The Blockchain Explained to Web Developers (2016)

Porém, ambos os fatos podem aparecer em ordens diferentes em nós distantes um do outro. Para que toda a rede concorde na ordem dos fatos e preserve sua integridade é

necessário um sistema de sincronização de dados, um algoritmo de consenso, que será discutido em mais detalhes na Seção 4.1.2 (FRANÇOIS ZANINOTTO, 2016).

No quesito segurança na *blockchain*, Tinker (2013) explica que a criptografia de chave pública/privada é um dos fundamentos da segurança moderna, pois este sistema possibilita que as pessoas assinem digitalmente documentos como: arquivos de texto, imagens, etc. De forma que esta assinatura digital é muito difícil de ser replicada por outra pessoa. Audrey Watters (2016) diz que a *blockchain* utiliza-se deste sistema de criptografia para assinar digitalmente as transações. Gareth Peters e Efstathios Panayi (2015) afirmam que para qualquer *blockchain* o acesso e a utilização dos ativos digitais não é possível sem o conhecimento da chave privada do dono atual. Está descrito na Seção 5.2.1 detalhes práticos de como a chave privada garante que somente o dono atual do ativo possa transferi-lo.

3.1.1.2. Consenso

De acordo com o Dicionário Online de Português, consenso significa pensamento comum, consentimento ou ação de aprovar. Um algoritmo de consenso é uma sequência de passos que permite a um sistema distribuído alcançar um acordo, ou seja, um consenso (DIGITAL ASSETS HOLDINGS, 2016).

De acordo com David Schwartz et al. (2014) os sistemas livro-razão distribuídos possuem três categorias de problemas: corretude, acordo e utilidade. Corretude, significa que o sistema deve discernir entre transações fraudulentas e transações legítimas. Acordo, significa que o sistema deve manter-se íntegro e único. Utilidade remete ao quanto o sistema é útil para quem o utiliza, por exemplo: um sistema que demora um ano para processar as transações não é viável para ser usado em aplicações reais. O problema da resolução de conflitos descrito na Seção 3.1.1.1 encontra-se nesta penúltima categoria, pois ambas as transações são corretas, porém incompatíveis entre si. Os nós participantes do sistema devem entrar em acordo sobre qual transação será considerada a verdadeira e será registrada, ao passo de que a outra será descartada.

Tais problemas foram explorados antes mesmo da invenção dos sistemas distribuídos, conhecido como o Problema dos Generais Bizantinos (LESLIE LAMPORT, ET AL. 1982). Neste problema, os generais estão em território inimigo e desconhecido e têm o objetivo de coordenar um ataque, enviando somente mensagem uns aos outros, da mesma forma que os

nós em um sistema distribuído se comunicam. Porém, como estão em um ambiente desconhecido, as mensagens podem não chegar. O mesmo ocorre com os nós que se comunicam em uma rede não segura que pode perder os pacotes, ou corrompê-los ao longo do caminho. Um outro aspecto que adiciona complexidade ao problema é a possibilidade dos generais serem traidores, individualmente ou coletivamente, assim como um ou vários nós na rede podem tentar enganar o sistema emitindo transações fraudulentas ou incompatíveis entre si.

Existem vários algoritmos de consenso e, embora cada um possa ser mais adequado do que outro para determinado sistema distribuído, a depender dos requisitos, eles devem ser robustos o suficiente para tolerar falhas como as descritas no "Problema dos Generais Bizantinos", além de atingir corretude e acordo (DAVID SCHWARTZ, et al. 2014).

3.1.2. Estrutura da *blockchain*

Nas próximas seções será discutida a estrutura da *blockchain* com ênfase na implementação do *Bitcoin*.

3.1.2.1. Estrutura do bloco

Um bloco é uma estrutura de dados que tem como carga principal uma lista de transações e que, posteriormente, será incluído na *blockchain*. A estrutura do bloco representado no Quadro 1 consiste em um cabeçalho, que contém metadados e uma lista que armazena as transações. Além disso, o bloco contém outros dois campos: Tamanho do Bloco, que informa o tamanho do bloco em bytes e Contador de Transações, que é quantidade de transações contida no bloco, representado por um número inteiro (ANDREAS ANTONOPOULOS, 2015).

Quadro 1 - Estrutura de um bloco da *blockchain* do Bitcoin

Tamanho	Campo	Descrição
4 bytes	Tamanho do Bloco	Tamanho do bloco em bytes
80 bytes	Cabeçalho	Contém diversos metadados
1-9 bytes (Varint)	Contador de Transações	Quantidade de transações contidas no bloco
Variável	Transações	Transações contidas neste bloco

Fonte: Adaptado de Antonopoulos (2015)

3.1.2.2. Cabeçalho

O cabeçalho guarda a referência para o valor *hash* do bloco anterior, a raiz da árvore de Merkle, e *timestamp* (data e hora). A referência para o bloco anterior permite conectar o bloco à *blockchain.* A raiz da árvore de Merkle, que será apresentada em mais detalhes na Seção 3.1.2.3, contém um resumo de todas as transações do bloco. O Quadro 2 abaixo representa a estrutura geral do cabeçalho.

Quadro 2 - Estrutura do cabeçalho de um bloco da *blockchain*

Tamanho	Campo	Descrição
32 bytes	Hash do Bloco Anterior	Referencia ao valor hash do bloco anterior
32 bytes	Raiz da Árvore de Merkle	A hash para as transações do bloco na Raiz da Árvore de Merkle
4 bytes	Timestamp	A data e hora aproximada em que o bloco foi criado

Fonte: Adaptado de Antonopoulos (2015)

O valor *hash* de um bloco é o seu identificador único, obtido ao submeter o cabeçalho a função *hash* criptográfica duas vezes[5]. Nota-se que ela não é inclusa no cabeçalho do bloco, nem é transmitida na rede devido ao fato de ser calculada por cada nó no instante em que o bloco é recebido (ANDREAS ANTONOPOULOS, 2015).

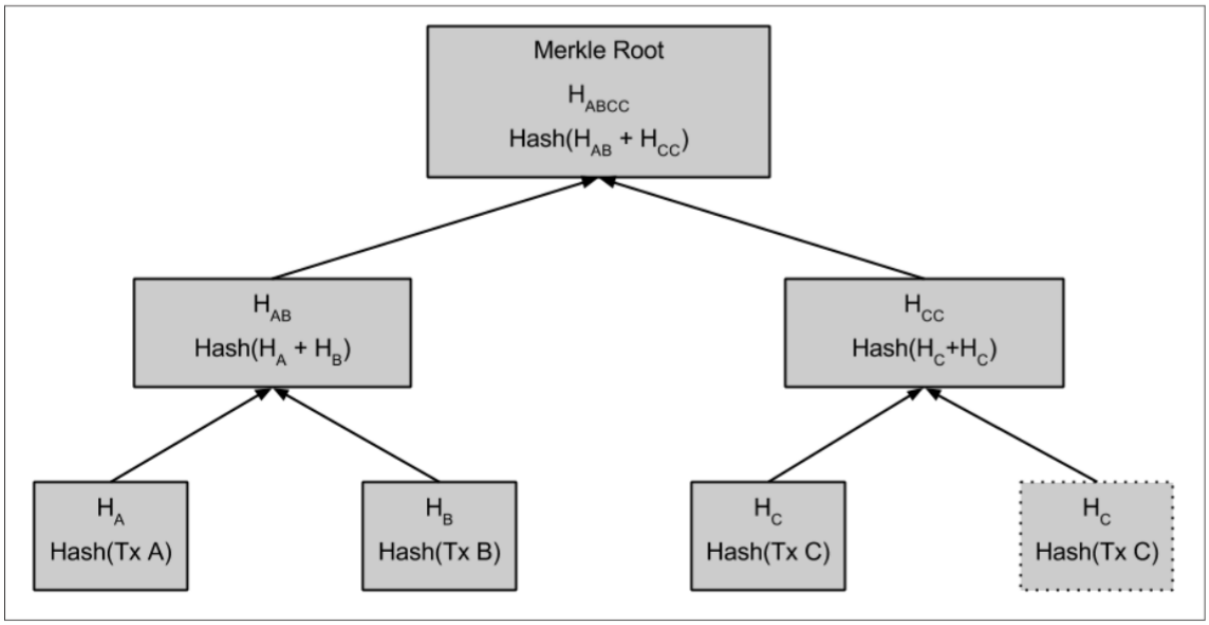
[5] Esta técnica de programação foi proposta por Ferguson e Schneier (2013) a fim de tornar a função *hash* invulnerável ao ataque *length-extension.*

3.1.2.3. Árvore Merkle

Uma árvore de Merkle é uma estrutura de dados utilizada para verificar a integridade de grandes volumes de dados. Dentro deste contexto, elas servem para resumir todas as transações de um bloco em uma *hash*, garantindo a integridade das transações e a possibilidade de verificar se determinada transação está contida no bloco.

Uma única hash de 32 bytes que represente todas as transações de um bloco é obtida ao submeter recursivamente um par de nós da árvore a um algoritmo criptográfico até que reste apenas uma única *hash*, que é a raiz da árvore de Merkle, contida no cabeçalho do bloco.

Figura 4 - Cálculo dos nós de uma árvore Merkle com número ímpar de elementos



Fonte: Antonopoulos (2015)

A Figura 4 ilustra a estrutura de uma árvore Merkle que contém três transações: A, B e C, representadas no nível inferior da árvore acima. O conteúdo das transações é submetido a uma função *hash* onde o resultado de cada par de folhas é concatenado para formar o nó “pai" e assim em diante. Como a estrutura é uma árvore binária é preciso um número par de nós, duplicando determinada transação, se necessário, para balancear a árvore como demonstrado na Figura 4 onde a transação C foi duplicada (ANDREAS ANTONOPOULOS, 2015).

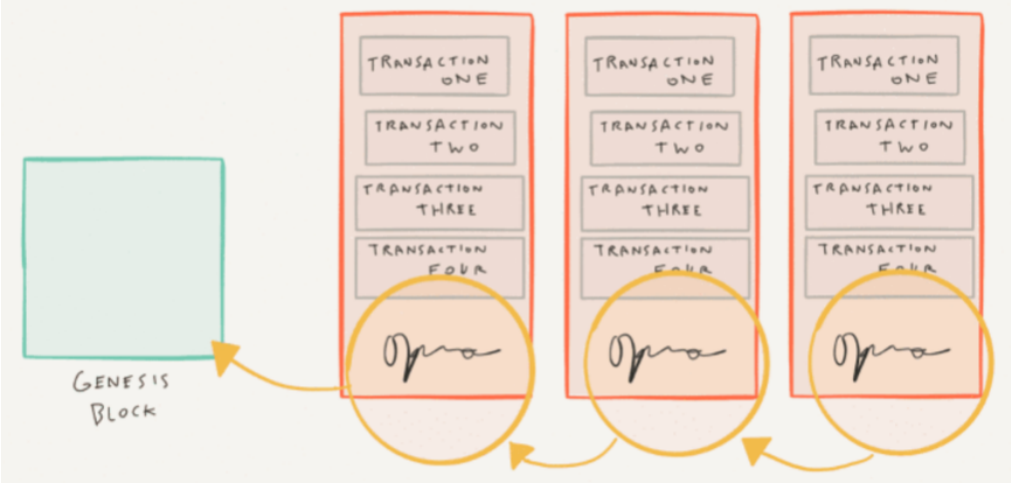
3.1.2.4. Bloco gênese

A partir de qualquer bloco na *blockchain* é possível rastrear o bloco anterior a partir do campo *previous hash*. Eventualmente chegar-se-à ao bloco gênese, que é o nome dado ao primeiro bloco criado na *blockchain*, que provê um ponto de partida para que a mesma possa ser construída (ANDREAS ANTONOPOULOS, 2015). O bloco gênese é representado na Figura 5 em forma de quadrado.

3.1.2.5. Vinculação dos blocos

Antonopoulos (2015) explica que existe uma cópia completa da *blockchain* em cada nó que está conectado a rede do Bitcoin que é constantemente atualizada conforme novos blocos vão chegando. Antes de vincular os blocos à *blockchain* do Bitcoin, os mesmos são validados e após este processo, o nó procura dentro do cabeçalho do bloco pela *hash* do bloco anterior. Se for conhecida, o bloco é adicionado ao fim da *blockchain*. A Figura 5 representa os blocos ligados entre si.

Figura 5 - Blocos ligados um ao outro em forma de corrente



Fonte: What is Blockchain? (2016)

3.2. BigchainDB

De acordo com Trent McConaghy et al. (2016) BigchainDB é um banco de dados descentralizado e distribuído e que, por incorporar as melhores características da *blockchain* e de bancos de dados distribuídos (BDD), ilustrado no Quadro 3, foi escolhido para armazenar os documentos e transferi-los entre as pessoas que utilizarão a plataforma proposta.

No BigchainDB a posse de determinado ativo digital é garantida através do par de chaves pública e privada. Para todas as transações é necessário gerar uma assinatura digital que é calculada com base na chave privada e na chave pública da pessoa. Logo, ao informar a sua chave privada a pessoa expressa o seu consentimento em querer transferir o ativo a outra pessoa. O Quadro 3 abaixo compara os aspectos do BigchainDB com os aspectos das *blockchains* e dos BDD tradicionais.

Quadro 3 - Blockchain e BDD comparados com o BigchainDB

	Blockchain Tradicional	BDD Tradicional	BigchainDB
Alta vazão	×	✓	✓
Baixa latência	×	✓	✓
Alta capacidade de armazenamento	×	✓	✓
Controle descentralizado	✓	×	✓
Imutabilidade	✓	×	✓
Criação e movimento de ativos digitais	✓	×	✓

Fonte: Adaptado de Trent McConaghy, et al. (2016)

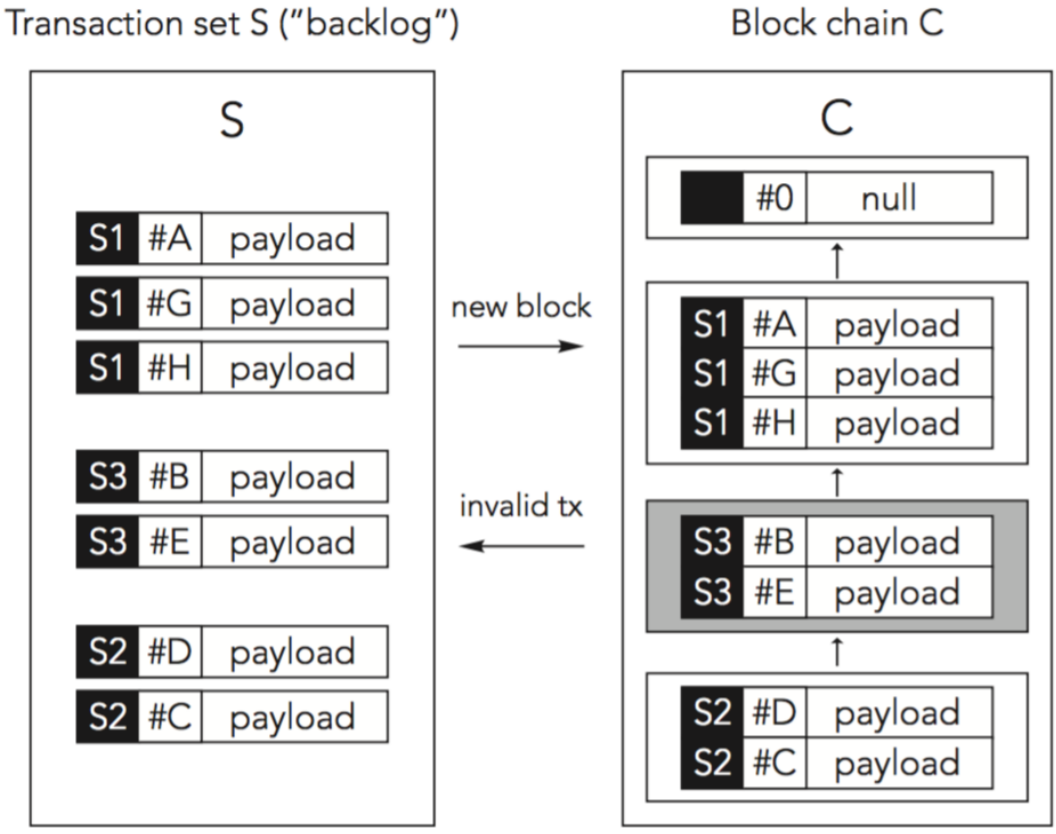
O Quadro 3 demonstra que o BigchainDB incorporou os melhores aspectos da *blockchain* e dos BDD. Os dados inseridos são imutáveis, o controle é descentralizado e permite criação e movimento de ativos digitais, que é algo que pode ser representado digitalmente e atribuído a uma pessoa, no caso do sistema proposto o ativo digital é o próprio documento da pessoa. Possui também alta capacidade de armazenamento, baixa latência e alta vazão.

Ao longo das próximas seções é explicado de forma breve como as características da *blockchain* foram incorporadas a um banco de dados distribuído sem perda em escalabilidade.

3.2.1. Arquitetura do Banco de Dados

O BigchainDB foi inicialmente construído a partir de um banco de dados distribuído, o RethinkDB[6], do qual herda suas características de alta escalabilidade. A partir dele, as características da *blockchain* foram incorporadas. A Figura 6 ilustra a arquitetura do BigchainDB. Apesar de ser utilizado como um único sistema, ele possui dois bancos de dados distribuídos, que são representados como duas tabelas: **S** (*backlog*) e **C** (*blockchain*), que serão discutidos a seguir.

Figura 6 - Arquitetura do BigchainDB



Fonte: Trent McConaghy, et al. (2016)

[6] https://www.rethinkdb.com

O banco de dados **S** serve para armazenar um conjunto de transações desordenado. Cada transação ao ser recebida por um nó é validada, e se de acordo com aquele nó, for uma transação válida, ela é armazenada em **S**. Os nós do BigchainDB consideram uma transação válida se ela atingir os seguintes requisitos:

- Todas as assinaturas devem ser válidas. Complementando o que foi discutido na Seção 3.1.1, o acesso a um ativo digital na *blockchain* só é possível pois toda transação deve conter uma assinatura digital que é feita utilizando a chave privada do dono atual que representa o seu consentimento na ação;
- Se for uma transação que cria um novo ativo digital, o que está sendo criado não deve existir previamente;
- Caso seja uma transferência de posse, o ativo digital deve existir e a transferência deve ter sido requisitada pelo dono atual, e não por donos anteriores.

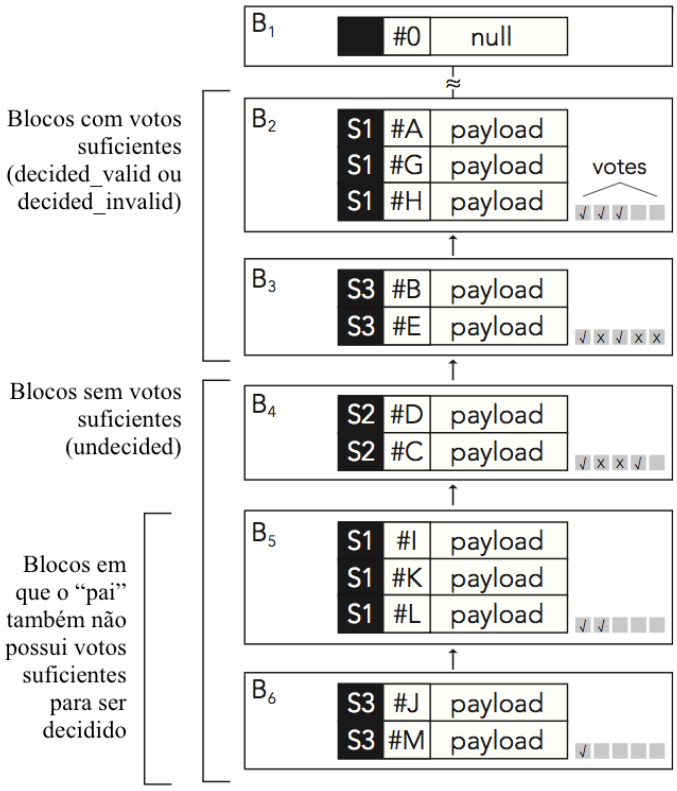
Considere que existem N nós e que $S_k = \{t_{k,1}, t_{k,2}, \ldots\}$ é o conjunto desordenado de transações em **k**. As transações em **S** são processadas por **k** da seguinte maneira:

1. As transações de S_k são ordenadas em uma lista;
2. Um bloco é criado e as transações são armazenadas neste bloco;
3. O bloco recém criado é movido para **C**.

A função de **C** é armazenar de forma ordenada os blocos, sendo que cada bloco possui uma referência para o bloco anterior. Os blocos em **C** possuem três estados distintos: não-decidido, válido e inválido. Os nós que possuem permissão de voto devem votar se um bloco que ainda não foi decidido é válido ou inválido. O novo estado é decidido por maioria, e um bloco só pode receber votos positivos se todas as transações contidas dentro do bloco forem válidas.

Os blocos vão sendo inseridos em **C** mesmo que ainda existam blocos não decididos, como é demonstrado na Figura 7, no bloco B6 que foi inserido depois do B5, ainda que B5 não tenha recebido votos o suficiente para ser considerado válido ou inválido. Este conceito chama-se *blockchain pipelining* e possibilita ganhos em escalabilidade ao incorporar características da *blockchain* a banco de dados distribuídos (TRENT MCCONAGHY, et al. 2016).

Figura 7 - *Blockchain pipelining*



Fonte: Trent McConaghy, et al. (2016)

3.3. *Inter Planetary File System* (IPFS)

Juan Benet (2016) explica que o IPFS é um sistema *peer-to-peer* que tem como objetivo conectar vários dispositivos ao mesmo sistema de arquivo, provendo um modelo de armazenamento endereçado ao conteúdo ao mesmo tempo em que os nós não precisam confiar uns nos outros e todos possuem a mesma influência na rede. Os nós se conectam entre si para transferir os arquivos.

O IPFS gera uma *hash* única e imutável para cada arquivo digital, além de possibilitar encontrar o arquivo na rede *peer-to-peer* a partir desta mesma *hash.* Este sistema replica o arquivo nos nós que o requisitam, diminuindo assim as chances de determinado arquivo ficar indisponível temporariamente ou permanentemente em caso de catástrofe (falha no disco rígido, quedas de energia ou interrupção do serviço de internet do servidor que provê os documentos, por exemplo), sendo inversamente proporcional à quantidade de pessoas interessadas em visualizar o documento digital.

O IPFS mostrou-se uma ótima ferramenta para utilizar em conjunto com a *blockchain*, tornando desnecessário armazenar no banco de dados o arquivo de fato, o que implica em ganhos de desempenho uma vez que não está mais limitado ao tipo e tamanho dos dados aceito pelo BigchainDB. Portando, não será gasto processamento com conversões toda vez que um documento for consultado ou inserido.

Na seção subsequente será explanado o principal sub-protocolo do IPFS que é responsável por diferentes funcionalidades essenciais para este projeto.

3.3.1. Objetos

Juan Benet (2016) explica que o IPFS constrói um grafo direcionado e acíclico, capaz de representar arquivos e diretórios. Tal estratégia é chamada de Merkle DAG e provê as seguintes propriedades para o IPFS:

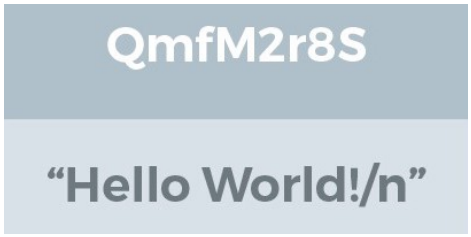
1. Endereçamento a conteúdo: todos os dados são unicamente identificados por uma *hash,* portanto para esta aplicação basta salvar no BigchainDB a *hash* do documento gerada pelo IPFS, resultando em um sistema mais eficiente e removendo restrições de tamanhos em bytes e tipos de arquivos, embora neste trabalho seja considerado somente formato PDF.
2. Resistência contra adulteração: todo o conteúdo é passível de verificação de integridade. É possível detectar erros, corrupções ou adulterações nos dados. Este ponto é especialmente útil para o sistema pois pode-se confiar que o arquivo no IPFS não sofrerá mudanças.
3. Eliminação de duplicatas: o sistema considera que todos os objetos que possuem exatamente o mesmo conteúdo são idênticos, logo devem ser armazenado somente uma vez ainda que possuam nomes distintos. Se um arquivo sofrer alterações, o IPFS reconhecerá como um novo objeto, com uma *hash* diferente. O arquivo original não é afetado.

De acordo com Christian Lundkvist (2015), a estrutura que representa os arquivos no IPFS é composta por um campo *Data*, que armazena dados binários não-estruturados de tamanho até 256 kB. E o campo *Links*, que é um vetor de estruturas *Link*, que por sua vez possui os campos:

- *Name*: armazena o nome do Link;
- *Hash*: guarda a hash do objeto relacionado;
- *Size*: informa o tamanho total do objeto;

Logo, um arquivo de tamanho menos que 256 kB que possui a hash *QmfM2r8s*... pode ser visualizado como um único nó. A Figura 8 representa este arquivo como um único nó no grafo.

Figura 8 - Representação de um arquivo < 256 kB no IPFS

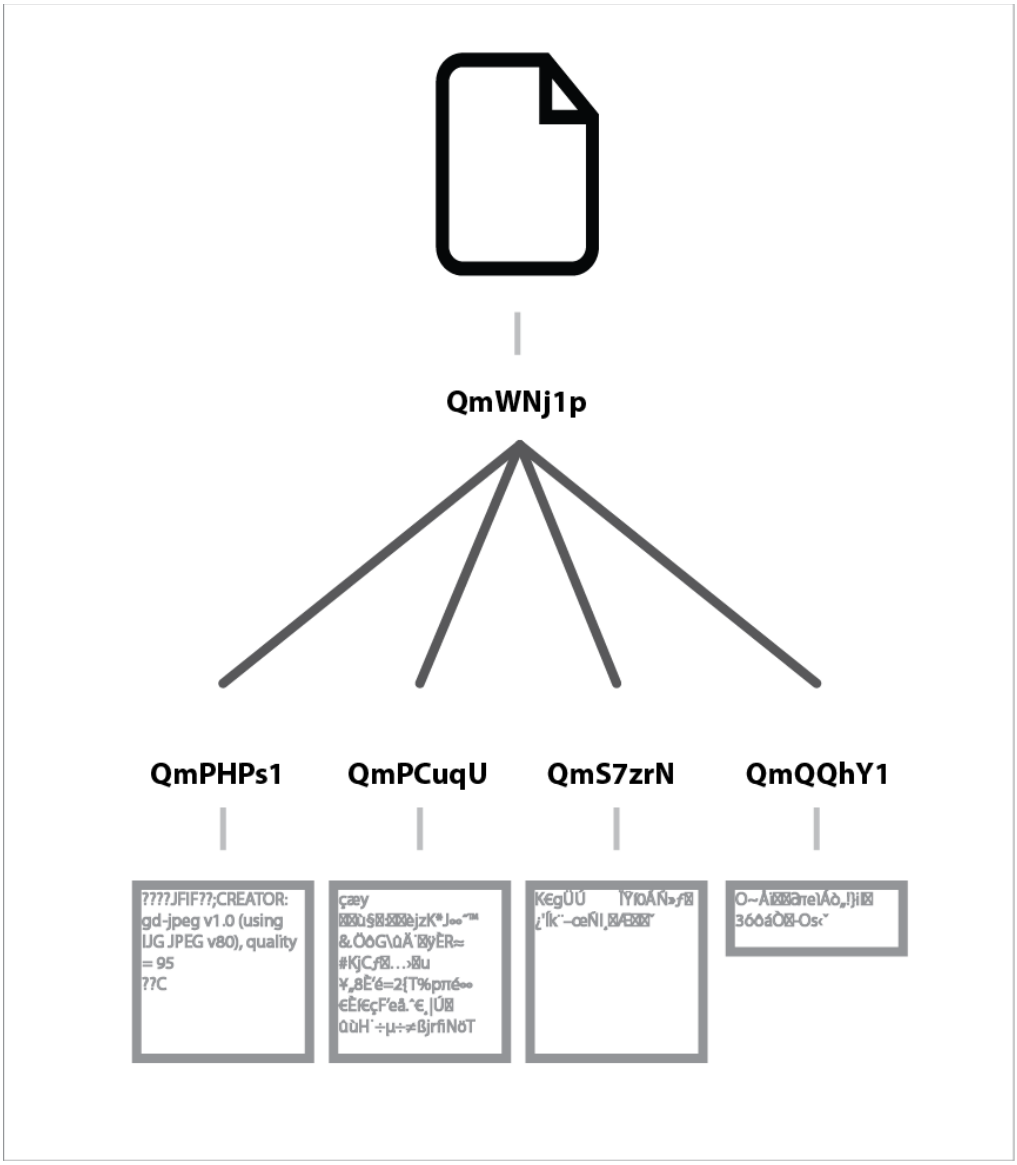


Fonte: Christian Lundkvist (2015)

Neste caso, o campo *Data* armazena o conteúdo "Hello World!\n” além de um pequeno cabeçalho, o vetor *Links* é vazio. Percebe-se que esta estrutura não contém o nome do arquivo, portanto documentos que possuem o mesmo conteúdo mas nomes diferentes são o mesmo objeto no IPFS e terão a mesma *hash.*

Em arquivos maiores que 256 kB o vetor *Links* é preenchido com pedaços do arquivo, de tamanho menor que 256 kB. O campo *Name* é vazio para os sub-blocos e *Data* especifica que este objeto representa um arquivo maior. Um documento que possui a hash *QmWNj1p*… pode ser visualizado na Figura 9:

Figura 9 - Representação de um arquivo > 256 kB no IPFS



Fonte: Matt Zumwalt et al. (2017)

Vale ressaltar que os sub-blocos podem também conter os seus próprios sub-blocos, formando um grafo direcionado e sem ciclos.

4. TRABALHOS RELACIONADOS

Este capítulo descreve alguns trabalhos que possuem uma proposta relacionada com este trabalho. Dentre os trabalhos citados nas seções, o Bitnation destaca-se, por ser o único sistema de armazenamento de documentos baseado na *blockchain*.

4.1. Bitnation

De acordo com Susanne (2015), governança 1.0 é a combinação involuntária de governança e território geográfico, de forma que uma entidade reclama para si o monopólio da violência sob determinada região do globo terrestre e em troca tal entidade provê e normalmente também possui o monopólio de serviços, como: segurança, resolução de disputas e aplicação da lei. A autora afirma que dentro deste modelo antigo e ultrapassado não existe liberdade de movimento para os cidadãos, o que não faz sentido em um mundo cada vez mais globalizado, onde o desejo de escolha e liberdade é cada vez crescente.

Susanne (2015) explica que estamos na "era da tecnologia *blockchain*", pois agora temos sistemas com propriedades requeridas para sistemas de governança. De acordo com a autora, a *blockchain* pode armazenar certidões de nascimento, casamento, morte, propriedade, contratos, entre uma variedade enorme de registros que são normalmente criados e estão sob a posse de governos. Além disso, também é possível identificar cada indivíduo na rede por uma "assinatura eletrônica". Aliado a isso, o consenso proporcionado pela *blockchain* possibilita que os indivíduos assinem e verifiquem transações, ao invés de um agente governamental ou outra autoridade terceira. Com base nisto, ela afirma que novas nações sem fronteira estão surgindo, denominadas *Decentralized Borderless Voluntary Nations* (DBVNs). Assim, Susanne (2015) propõe o Bitnation, que é uma plataforma que permite a emergência das DBVNs. O Bitnation oferece a princípio, mas não está limitado a: um sistema de identificação de cidadãos, uma biblioteca de aplicativos descentralizados, um mecanismo de resolução de disputas, registro de propriedades de terras, certificados de nascimento e morte, certidão de casamento, divórcios e serviços de segurança.

O Bitnation relaciona-se com o sistema proposto neste trabalho no ponto em que oferece uma solução baseada na *blockchain* para as pessoas armazenarem registros. No entanto este trabalho diverge do proposto por Susanne (2015) no ponto em que não está

limitado somente a estes tipos de documento, não limita o público somente a “cidadãos”, não emite tais documentos e permite a transferência da posse.

4.2. BRIGHT

Shigeru et al. (2015) explicam que o alto custo de proteger os atuais sistemas de controle de direitos, do inglês, *digital rights management* ou DRM, para vídeos de *hackers* afeta diretamente o preço final do produto, então, os autores sugerem que um sistema DRM baseado na tecnologia da *blockchain* é seguro e muito menos dispendioso financeiramente. Diante disso eles conceituam e realizam uma implementação inicial do BRIGHT, um sistema de gerenciamento de direitos descentralizado baseado na *blockchain*.

Entretanto, é inviável armazenar na *blockchain* o vídeo completo devido ao seu tamanho que pode ser enorme. Logo, é necessário uma forma de acoplar a informação de direito que está na *blockchain* a um determinado vídeo. Além disso, a latência entre um bloco e outro não pode ser muito grande, pois isso implicaria em uma espera para a pessoa, o que pode tornar-se um incomodo.

Tendo em vista esses problemas, Shigeru et al. (2015) propõem que o método de entrega do vídeo seja separado do método de entrega das informações de direitos ou licença. Logo, eles modificam o Media Player Classic - Home Cinema[7] para que ele consiga pegar as informações de direito que estão na *blockchain*. Para que não seja necessário esperar tempo demais para assistir ao vídeo, o intervalo de tempo entre a criação de um bloco e a inserção na *blockchain* foi ajustado para cinco segundos. Como resultado, o preço deste tipo de serviço deve cair e tornar-se mais acessível a semi-profissionais.

Este trabalho está relacionado com Shigeru et al. (2015) no ponto em que ambos utilizam a *blockchain* para armazenamento de dados, mas divergem quanto à informação carregada no bloco da *blockchain* e ao público alvo.

4.3. Rep on the Block

Richard Dennis e Gareth Owen (2015) afirmam que a reputação mede o quanto a comunidade confia em um determinado indivíduo e citam o exemplo do eBay, um mercado

[7] https://github.com/mpc-hc

online que possui um dos sistemas de reputação mais usados no mundo, com mais de um bilhão de transações todos os dias. Os autores afirmam que o cálculo da reputação ser realizado de forma centralizada pelo eBay produz efeitos negativos, pois a empresa pode mudar a forma que o cálculo é feito sem o consentimento dos clientes. Os autores citam o exemplo em que o eBay impediu que vendedores deixassem um feedback negativo a respeito dos compradores.

Existem outros casos de sistemas de reputação de grande sucesso além do eBay, mas eles são centralizados e portanto não servem para uma rede *peer-to-peer*, apesar de existirem sistemas de reputação voltados para impedir *freeloaders*[8] em redes *peer-to-peer*, todos eles possuem um problema em comum: a impossibilidade de ligar uma identidade a uma única pessoa, e impedir que um indivíduo obtenha várias identidades.

Tendo em vista os problemas citados, os autores conceituam um novo sistema de reputação baseado na *blockchain*, o Rep on the Block, que resolve esses problemas assim como previne possíveis ataques, apesar do foco ser em redes *peer-to-peer*, o sistema também pode ser implementado em um *e-commerce*[9] comum. O sistema não incluirá opinião humana,. Cada usuário poderá apenas dizer se a transação foi positiva ou não-satisfatória. Entende-se por transação positiva quando o usuário recebeu o arquivo que foi requisitado, e por transação, um pedaço de informação assinado pela chave privada do emissor para a pessoa que a requisitou.

As transações seriam verificadas pelos próprios usuários do sistema, cada pessoa envolvida na transação seria contatada a fim de obter uma prova, que consiste em uma *hash* do arquivo e um número qualquer enviado por quem está verificando a transação. Isto verifica se cada pessoa enviou/recebeu o arquivo, mas apenas pode ser realizado se as pessoas ainda estiverem online na rede. Para dificultar que sejam geradas várias identidades por uma única pessoa, o endereço IP será vinculado a identidade a fim de aumentar os custos de se realizar um ataque.

A reputação seria calculada como a média de toda a sua reputação, desta forma dois nós receberiam a mesma pontuação se trocassem um arquivo ou mil arquivos entre si além de poder ser avaliado durante um curto período de tempo pois, segundo os autores, o

---

[8] Neste caso, são pessoas que realizam *download* em uma rede *peer-to-peer* mas não fazem *upload*.

[9] Site de comércio *online*.

comportamento de um indivíduo pode ser mais precisamente previsto se comparado com o seu comportamento nos últimos dias, portanto os autores afirmam que não é necessário analisar todo o comportamento anterior.

Com base no volume médio de 23,148 avaliações de reputação processadas pelo eBay todos os dias, os autores prevêem que a *blockchain* cresceria 53GB ao ano. Seria impossível que dispositivos de poucos recursos como celulares, pudessem usufruir do sistema. Para resolver estes problemas é proposto que o tempo entre a inserção de um bloco e outro seja curto, e que somente as pessoas que desejam verificar as transações tenham uma cópia completa da *blockchain*.

O sistema conceituado por Richard Dennis e Gareth Owen (2015) assemelha-se com o proposto neste trabalho no fato em que ambos utilizam a *blockchain* para armazenar um dado, mas divergem quanto à informação que cada bloco carrega.

5. O SISTEMA PROPOSTO

Este sistema[10] tem o propósito de apresentar uma maneira de assegurar a propriedade e imutabilidade de documentos digitais, de forma limpa, enxuta e centrada nas funcionalidades a fim de que as técnicas utilizadas neste trabalho sirvam de base para a construção de outros aplicativos melhores, mais completos e sofisticados.

Na Seção 5.1 serão detalhados os requisitos elicitados, na Seção 5.2 a arquitetura do sistema desenvolvido e por fim na Seção 5.3 detalhes da implantação do sistema.

5.1. Especificação

Primeiramente, iniciou-se a especificação dos requisitos, após uma breve documentação, deu-se a pesquisa para escolher as tecnologias mais apropriadas para utilizar no sistema. Um esboço da arquitetura foi planejada e as funcionalidades implementadas. Este processo de planejamento e implementação repetiu-se, onde cada iteração resultava em um sistema mais robusto, com mais funcionalidades e outras melhorias.

5.1.1. Requisitos

Nesta seção serão apresentados os requisitos funcionais (RF) do sistema. No Quadro 4 estão listados todos os requisitos funcionais do sistema.

Quadro 4 - Requisitos funcionais

	Requisitos Funcionais
RF001	O sistema deverá permitir a pessoa realizar o upload de um arquivo PDF e vinculá-lo a um par de chaves criptográficas.
RF002	O sistema deverá permitir a pessoa gerar um par de chaves criptográficas.
RF003	O sistema deverá permitir a pessoa transferir a posse do documento a outra pessoa.
RF004	O sistema deverá permitir a pessoa consultar um documento.
RF005	O sistema deverá permitir a pessoa realizar o download do documento.
RF006	O sistema deverá permitir a pessoa consultar se determinado documento pertence a determinada pessoa.
RF007	O sistema deverá permitir a pessoa realizar o download do comprovante de upload.

Fonte: elaborado pelo autor

[10] Código-fonte disponível em: https://github.com/gabrielmendanha/tcc2

Quadro 5 - Descrição dos requisitos funcionais

	Descrição
RF001	Permitir o upload de um arquivo PDF para o sistema de arquivo IPFS e dar a posse do documento ao par de chaves criptográficas fornecido pela pessoa. Entrada: um par de chaves criptográficas e um arquivo PDF. Saída: ID da transação.
RF002	Deve possibilitar a pessoa que utiliza o sistema criar um novo par de chaves criptográficas. Entrada: n/a. Saída: arquivo TXT contendo o par de chaves criptográficas.
RF003	Permitir que a posse de um documento seja transferida a outra pessoa. Entrada: o par de chaves pública e privada do dono atual e a chave pública do futuro dono. Saída: n/a.
RF004	Ao informar o ID de uma transação deve ser possível consultar os seguintes dados: dono nesta transação, status, data/hora, nome do documento. Entrada: ID de uma transação. Saída: dono do documento nesta transação, status, nome do documento, data e hora do upload.
RF005	Após informar o ID de uma transação, na mesma página da saída do RF004 deve ser possível realizar o download do documento contido na transação. Entrada: ID de uma transação. Saída: download do documento.
RF006	Ao consultar uma transação, a pessoa pode informar uma chave pública caso deseje saber se o documento contido na transação pertence a determinada pessoa (representada pela chave pública). Entrada: ID de uma transação, chave pública. Saída: dono do documento nesta transação, status, nome do documento, data e hora do upload e se o documento pertence ou não.
RF007	Após realizar o upload de um documento, a pessoa tem a opção de realizar o download do comprovante, que é um documento TXT que contêm as seguintes informações: data e hora da emissão, referência do documento (id da transação), chave pública utilizada e o nome do documento. Entrada: n/a. Saída: arquivo TXT.

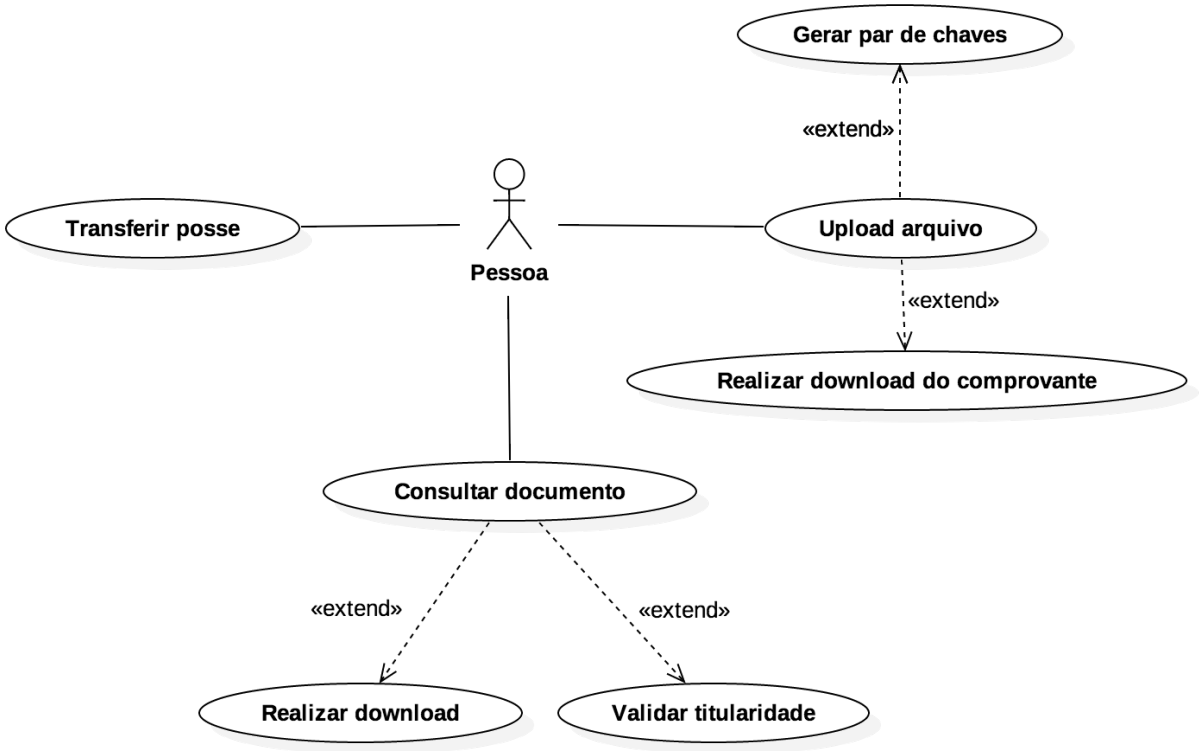
Fonte: elaborado pelo autor

5.1.2. Diagrama de Caso de Uso

De acordo com Sommerville (2007), os casos de uso possibilitam identificar as possíveis interações com o sistema e os papéis relacionados. Neste sistema, eles foram documentados na forma de diagramas em linguagem UML (*Unified Modeling Language*), disposto em uma única visão na Figura 10.

As funcionalidades descritas neste diagrama estão documentadas no Quadro 5 na Seção 5.1.1. Neste diagrama explica-se que existe somente um tipo de usuário no sistema, e que algumas das funcionalidades podem extender-se a outros casos de uso.

Figura 10 - Caso de Uso



Fonte: Elaborado pelo autor

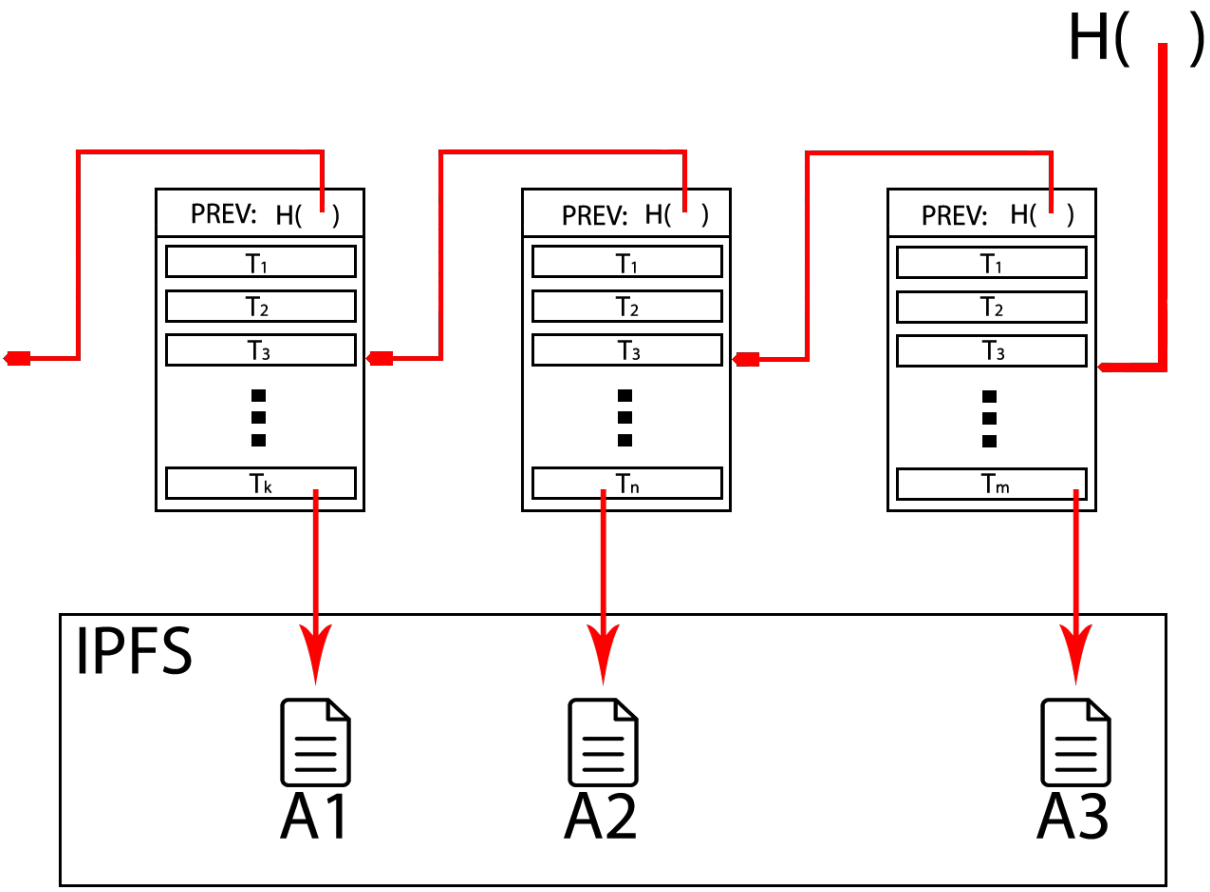
5.2. Projeto e Implantação

Na parte de projeto da arquitetura do sistema, foi realizada uma pesquisa para estudar, compreender e definir as melhores tecnologias que pudessem ser utilizadas junto com a *blockchain*, dentre as disponíveis no mercado. Na parte de implantação está descrita e documentada a estratégia utilizada para implantar o sistema em um serviço de computação em nuvem.

5.2.1. Arquitetura do Sistema

Nesta seção, está descrita a arquitetura principal deste sistema. Será explicado a integração do banco de dados BigchainDB com o IPFS, a dependência das funcionalidades do sistema para com as tecnologias utilizadas, assim como o fluxo de troca de mensagens e execução de procedimentos.

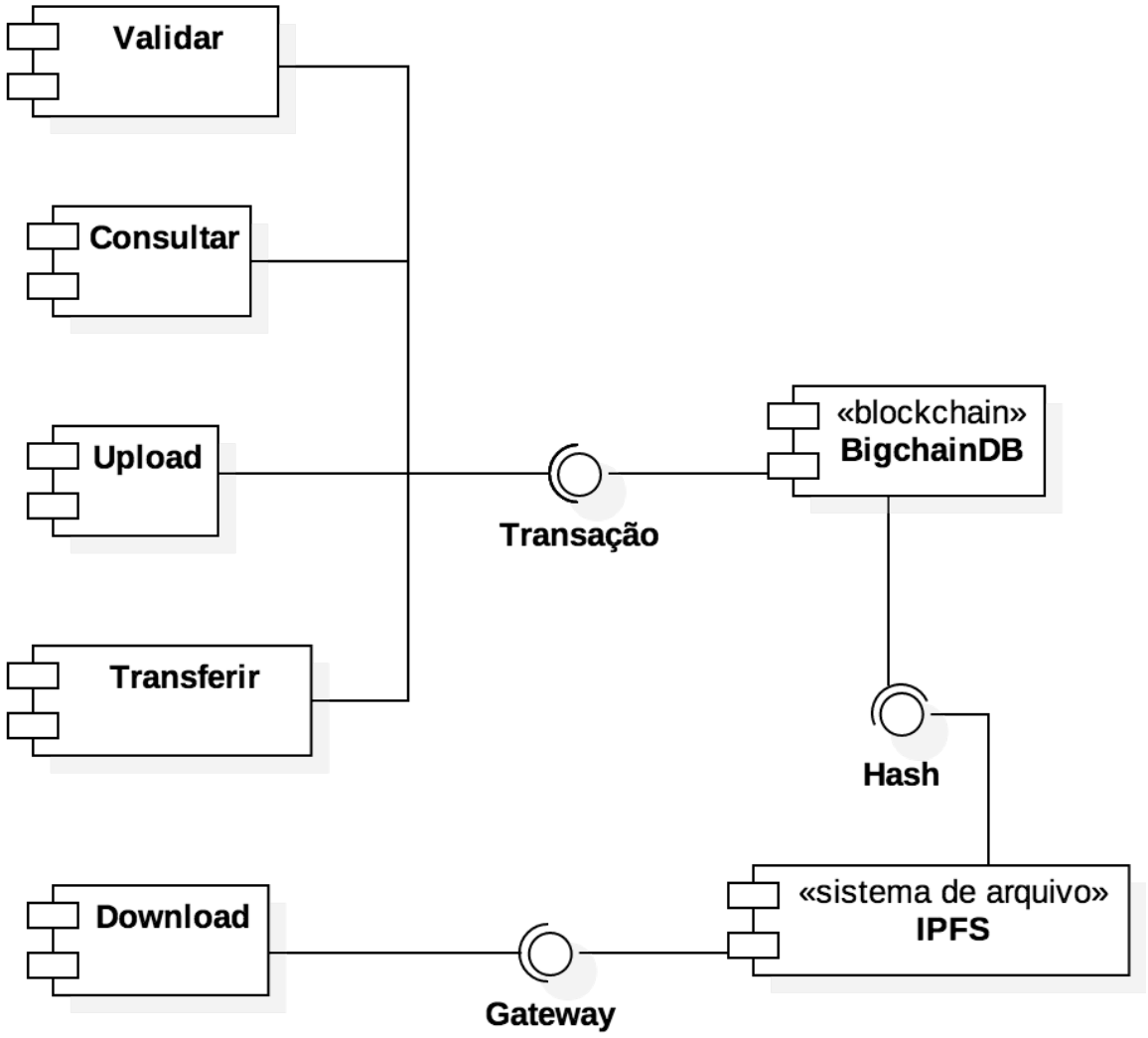
Figura 11 - Integração entre a *blockchain* e o IPFS



Fonte: Elaborado pelo autor

A Figura 11 ilustra como a blockchain provida pelo BigchainDB integra-se aos objetos no IPFS. Como o valor *hash* dos objetos no IPFS é única, imutável e basta para localizar determinado arquivo no sistema, é viável e seguro armazena-lá como uma referência para o documento binário. Embora omitido na imagem, cada transação dentro de um bloco armazena o valor *hash* do nó inicial, ou raiz, de uma Merkle DAG. Quaisquer modificações em documentos no IPFS são reconhecidas como um novo objeto que não está incluso na *blockchain.* Entretanto, o objeto original permanece intacto e seu valor *hash* continua vinculado à *blockchain.* O novo objeto, com as modificações, não é reconhecido pelo sistema.

Figura 12 - Componentes do sistema



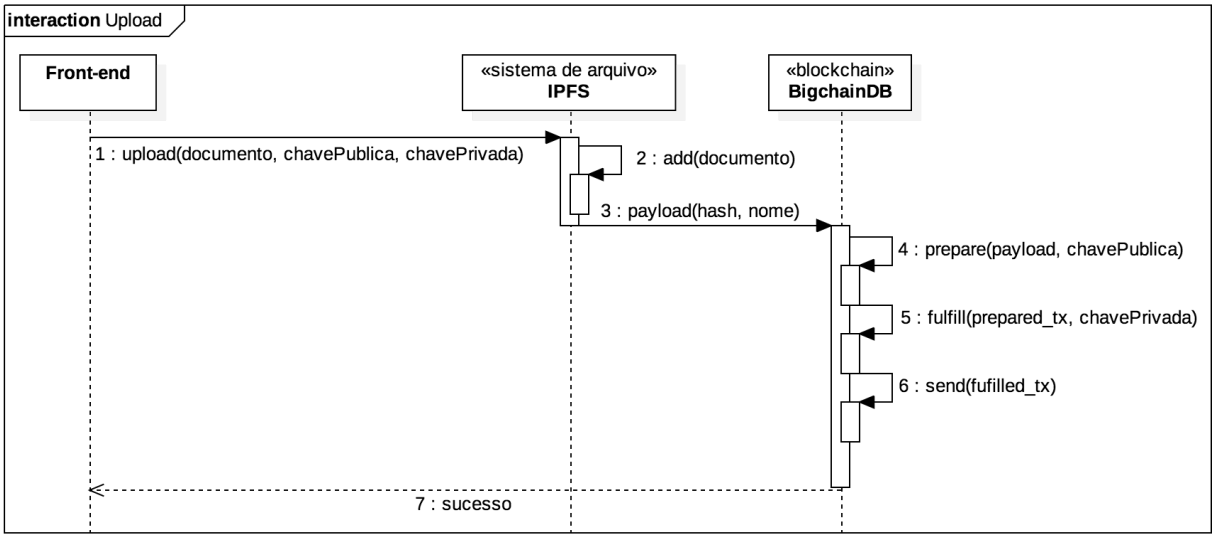
Fonte: Elaborado pelo autor

Na Figura 12 podemos observar o diagrama de componentes do sistema, o IPFS provê para o BigchainDB o endereço *hash* de cada arquivo que será armazenado no sistema, este endereço *hash* é associado a um par de chaves criptográficas dentro de uma transação.

Os componentes de Consulta, Upload e Transferência todos dependem da transação provida pelo BigchainDB pois é nela que estão contidos dados que serão consultados, assim como é armazenado na transação o endereço *hash* de um documento no instante em que o upload ou a transferência de posse é realizada.

O único componente que não consome funcionalidades diretamente relacionadas ao BigchainDB é o Download, que é realizado através do *gateway* que o IPFS provê. Da forma que foi implementado é possível qualquer um realizar o download do documento, mas isso pode ser contornado realizando modificações a nível de aplicação e infra-estrutura, exigindo cadastro com dados pessoais e restringindo a conexão dos nós no IPFS somente àqueles que a organização possui controle, respectivamente.

Figura 13 - Upload de documento

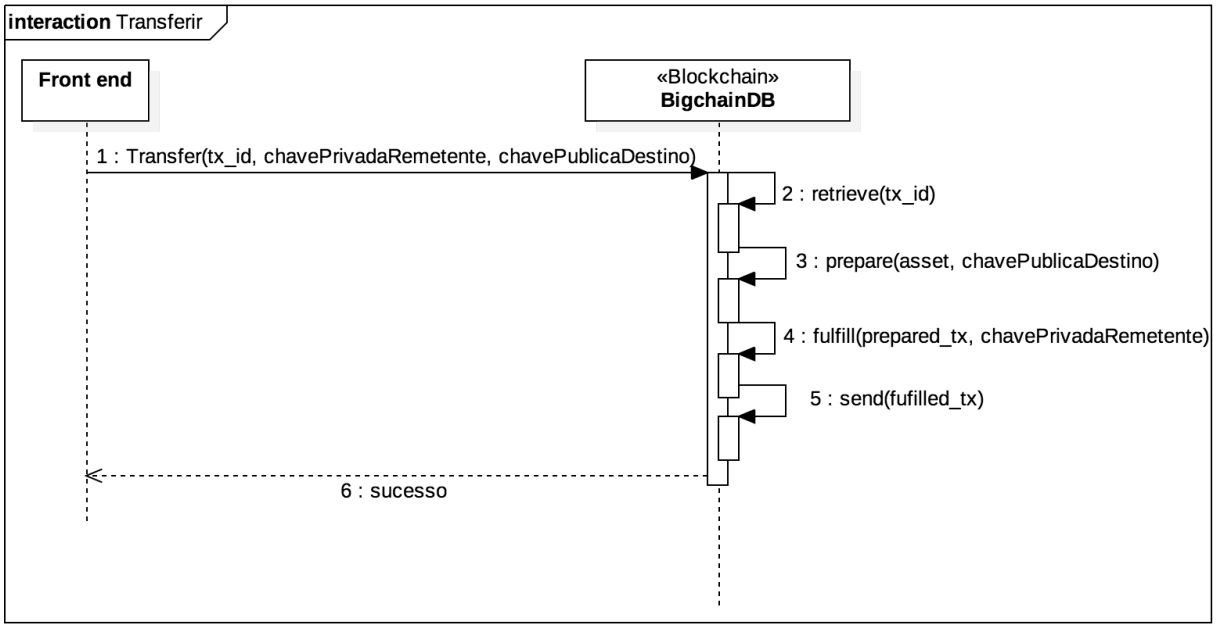


Fonte: Elaborado pelo autor

A Figura 13 representa o diagrama de sequência para o caso de uso de upload arquivo, onde os dados fornecidos pela pessoa são: o documento e um par de chaves criptográficas. Primeiramente é adicionado no sistema de arquivo IPFS o arquivo fornecido pela pessoa, é retornado a *hash* e o nome do documento.

É necessário realizar dois procedimentos antes de enviar a transação para ser incluída na *blockchain.* Primeiro, a transação é preparada no método *prepare,* que recebe como parâmetro o *payload* e a chave pública e retorna uma transação no formato JSON com seus campos parcialmente concluídos. Depois, a transação é assinada digitalmente com a chave privada no método *fulfill.* Em outras palavras, serão escritas no objeto JSON as condições para que ela possa ser futuramente transferida a outra pessoa. Após esse processo, a transação é enviada para o BigchainDB para ser validada e incluída na *blockchain*. O sistema então encaminha a pessoa para uma página onde ela pode realizar o download do comprovante e visualizar informações relacionadas a transação submetida.

Figura 14 - Transferência de posse



Fonte: Elaborado pelo autor

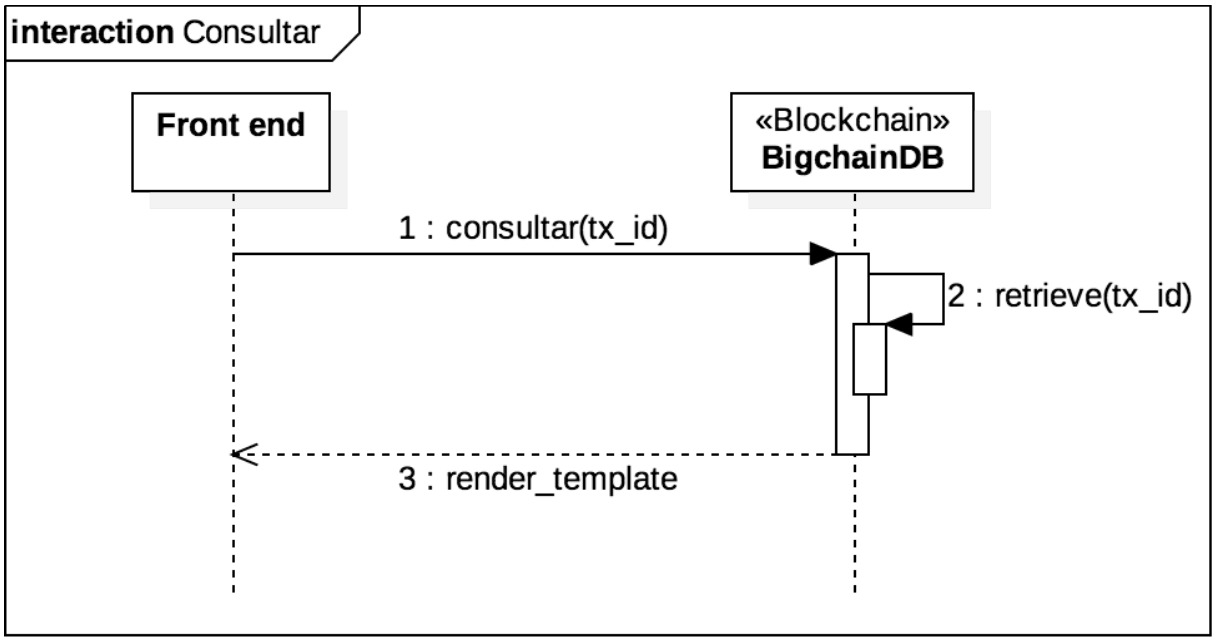
Na Figura 14 observa-se o diagrama de sequência para uma transferência de posse. A pessoa fornece a identificação da transação (*tx_id*), a chave privada do remetente e a chave pública do destinatário.

O BigchainDB captura esses dados e, em primeiro momento recupera o objeto (uma transação) através do *tx_id*, pois neste caso o método *prepare* exige metadados presente neste mesmo objeto, além da chave pública do destinatário.

Realizados estes procedimentos, o fluxo segue muito parecido com o descrito anteriormente na Figura 14. A única diferença é que o método *fulfill* recebe a chave privada do remetente. Reunindo todos os dados, a transferência de posse é submetida e se todos os campos tiverem sido preenchidos corretamente, uma mensagem de sucesso é enviada para a pessoa, ou insucesso caso contrário.

O diagrama de sequência para a consulta de uma transação está descrito na Figura 15, é o mais simples, uma vez que apenas é necessário informar a identificação da transação (*tx_id*) desejada.

Figura 15 - Consultar documento



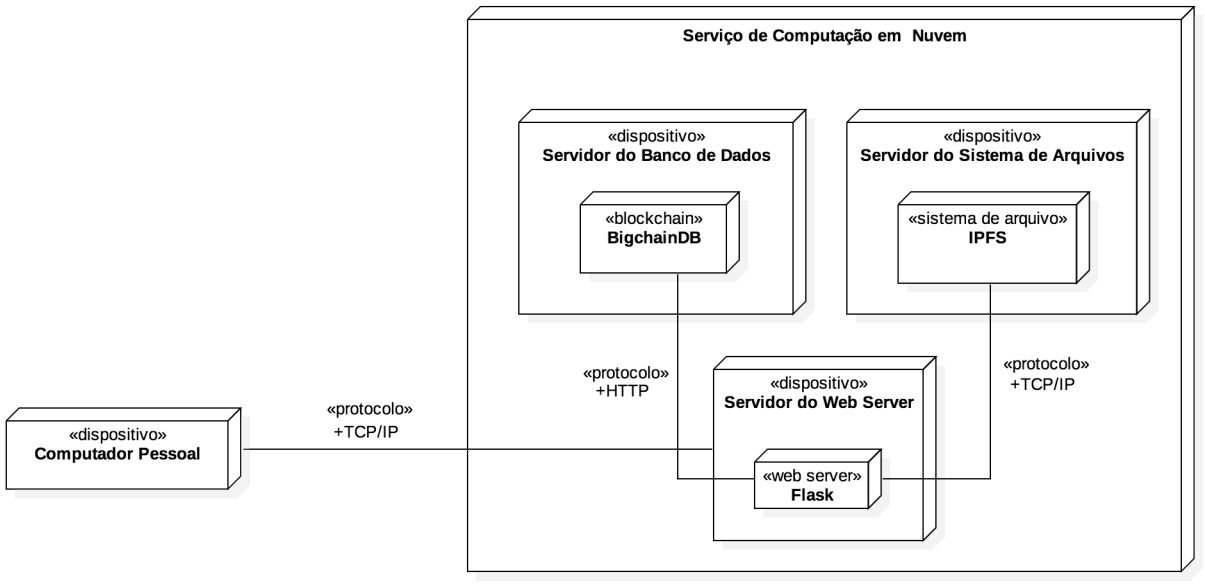
Fonte: Elaborado pelo autor

5.3. Implantação

Decidiu-se implantar o sistema por dois motivos: para ser utilizado pelos voluntários na avaliação, que será explicada em maiores detalhes no Capítulo 5 e exemplificar as possibilidades de escalabilidade. Primeiro será explicada a implantação realizada e depois a discussão teórica quanto à escalabilidade.

Para implantar o sistema foi utilizado o serviço de computação em nuvem Amazon Web Service[11]. Foram criadas três instâncias, onde cada uma assumiu uma responsabilidade. A Figura 16 representa a implantação do serviço.

Figura 16 - Implantação do *software* utilizando serviço de computação em nuvem



Fonte: Elaborado pelo autor

O cliente conecta-se a instância que está executando o servidor web. Todas as interações com a *blockchain* são realizadas através de uma API HTTP, pois ela está separada da instância que executa o servidor web. Assim como a instância que executa o IPFS está separada da instância que executa a *blockchain* e o servidor web.

Neste padrão, também é possível escalar os componentes de maneira mais fácil, uma vez que a grande vantagem do BigchainDB é ser escalável. Para aumentar o volume de transações escritas e lidas por segundo, basta utilizar o modelo de escalabilidade horizontal,

[11] https://aws.amazon.com

descrito por Galante e Bona (2012), replicar a instância que executa a *blockchain* e conecta-lá a qualquer uma das existentes, que também estão executando o processo do BigchainDB. Desta forma, o processamento é distribuído entre as instâncias, o que resulta em ganhos em desempenho. Pode-se utilizar a mesma técnica para o servidor web, porém é necessário adicionar um balanceador de carga entre o cliente e as instâncias do servidor web.

Embora o BigchainDB seja compatível com ambos RethinkDB e MongoDB, neste trabalho utilizamos o RethinkDB pois quando o desenvolvimento iniciou-se era a única opção disponível.

6. AVALIAÇÃO

Neste capítulo, será apresentada a avaliação do *software* desenvolvido. Para isso, foi utilizado um modelo adaptado do *User Experience Questionnaire*, proposto por Martin Schrepp et al. (2011). Este questionário tem o objetivo de medir a experiência com o usuário de maneira ágil e imediata através de aspectos de qualidade pragmático e prazeroso. Como este trabalho é uma prova de conceito, e não foi desenvolvido para ser utilizado em um ambiente de produção real. Foram levados em consideração apenas os aspectos pragmáticos, que são diretamente relacionados às funcionalidades oferecidas pelo *software* e um único atributo do aspecto prazeroso.

Os atributos de qualidade que foram considerados nessa avaliação, foram: *perspicuity, novelty, efficiency.* Cada uma refere-se respectivamente à: facilidade da pessoa em entender e manusear o sistema; inovação e criatividade entregues e a eficiência do *software* no que ele se propõe a resolver.

6.1. Planejamento

A fase de planejamento teve o objetivo de avaliar o sistema desenvolvido a fim de verificar se o mesmo encontra-se em conformidade com seus requisitos e seu propósito. Os participantes da avaliação são alunos de diversos cursos do Campus UFC Quixadá. A execução do sistema consistiu em uma série de atividades propostas aos voluntários pelo autor.

6.2. Execução

Esta etapa foi dividida em três partes: primeiro, uma introdução teórica à criptografia, autenticação por pares de chaves pública/privada, *blockchain* e como seus atributos podem ser utilizados para garantir a imutabilidade e propriedade de documentos digitais neste sistema, além do objetivo da plataforma em si. Segundo, a execução das tarefas propostas pelo autor, que aconteceu no laboratório de redes de computadores da Universidade Federal do Ceará em Quixadá, para que cada pessoa tivesse um computador para a avaliação. E por fim, a aplicação de um questionário *online* para coletar os resultados. As atividades propostas aos voluntários estão detalhadas nos próximos parágrafos.

Na interface apresentada na Figura 17, a pessoa deve gerar um novo par de assinaturas criptográficas (1) e usar os dados gerados para realizar o upload de um documento qualquer desde que tenha formato PDF (2).

Figura 17 - Upload de documento

Fonte: Elaborado pelo autor

Após realizado o upload, a pessoa deve fazer o download do comprovante (3), como ilustrado na Figura 18.

Figura 18 - Detalhes da transação submetida

Fonte: Elaborado pelo autor

Em posse do comprovante, a pessoa deve então consultar (4) o seu documento utilizando a referência do documento e opcionalmente a sua chave pública, ambos são informações disponíveis no comprovante. A Figura 19 ilustra esse passo.

Figura 19 - Consulta de documento

Fonte: Elaborado pelo autor

Realizada a consulta, a pessoa deve acessar a página provida pelo autor e fornecer a sua assinatura pública no formato "PRIMEIRO_NOME : ASSINATURA_PÚBLICA". Nesta mesma página, o voluntário deve escolher a chave pública de outra pessoa, e transferir (5) o seu documento à ele(a), assim como ilustrado na Figura 20.

Figura 20 - Transferência de documento

Fonte: Elaborado pelo autor

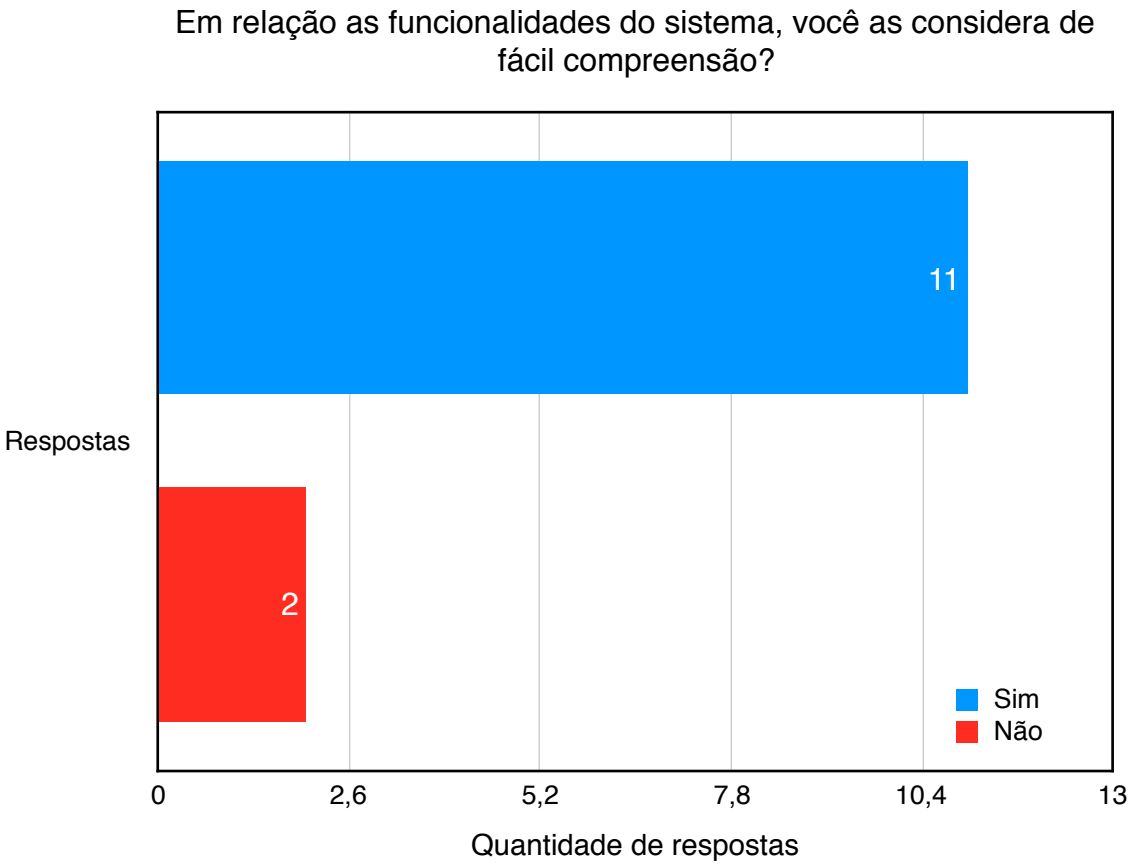
6.3. Resultados

A avaliação reuniu um total de treze voluntários, entre homens e mulheres. Todos eram alunos da Universidade Federal do Ceará em Quixadá. As questões foram elaboradas para que cada uma avaliasse um atributo na ordem: *perspicuity, novelty, efficiency* e estão dispostas na respectiva ordem.

Após a utilização do *software* cada voluntário respondeu o questionário eletrônico e as respostas foram analisadas. O resultado está apresentado a seguir:

Questão 1. Em relação as funcionalidades do sistema, você as considera de fácil compreensão?

Gráfico 1 - Respostas da Questão 1

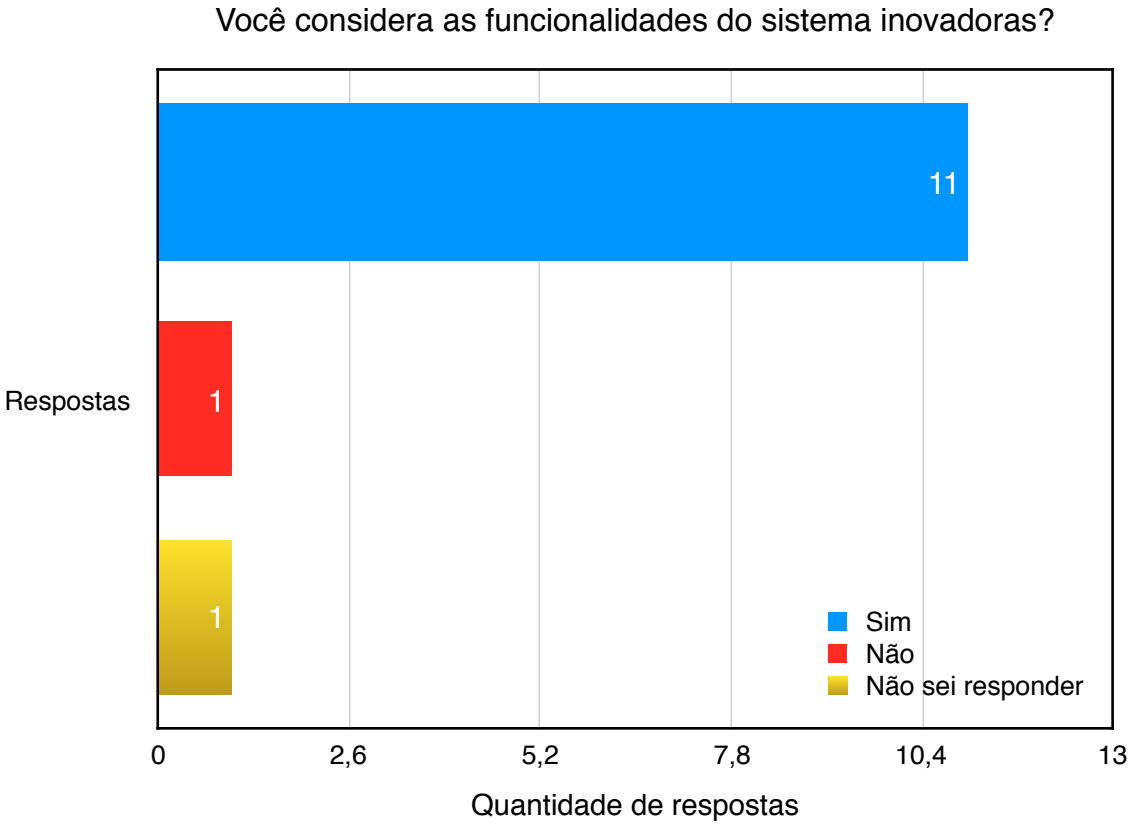


Fonte: Elaborado pelo autor

De acordo com o Gráfico 1, a grande maioria dos voluntários respondeu SIM, indicando que o sistema desenvolvido não é difícil de entender, mesmo para as pessoas que possuem pouco ou nenhum conhecimento de criptografia ou *blockchains*, uma vez que uma explicação teórica e não aprofundada em formalidades acadêmicas foi suficiente.

Questão 2. Você considera as funcionalidades do sistema inovadoras?

Gráfico 2 - Respostas da Questão 2

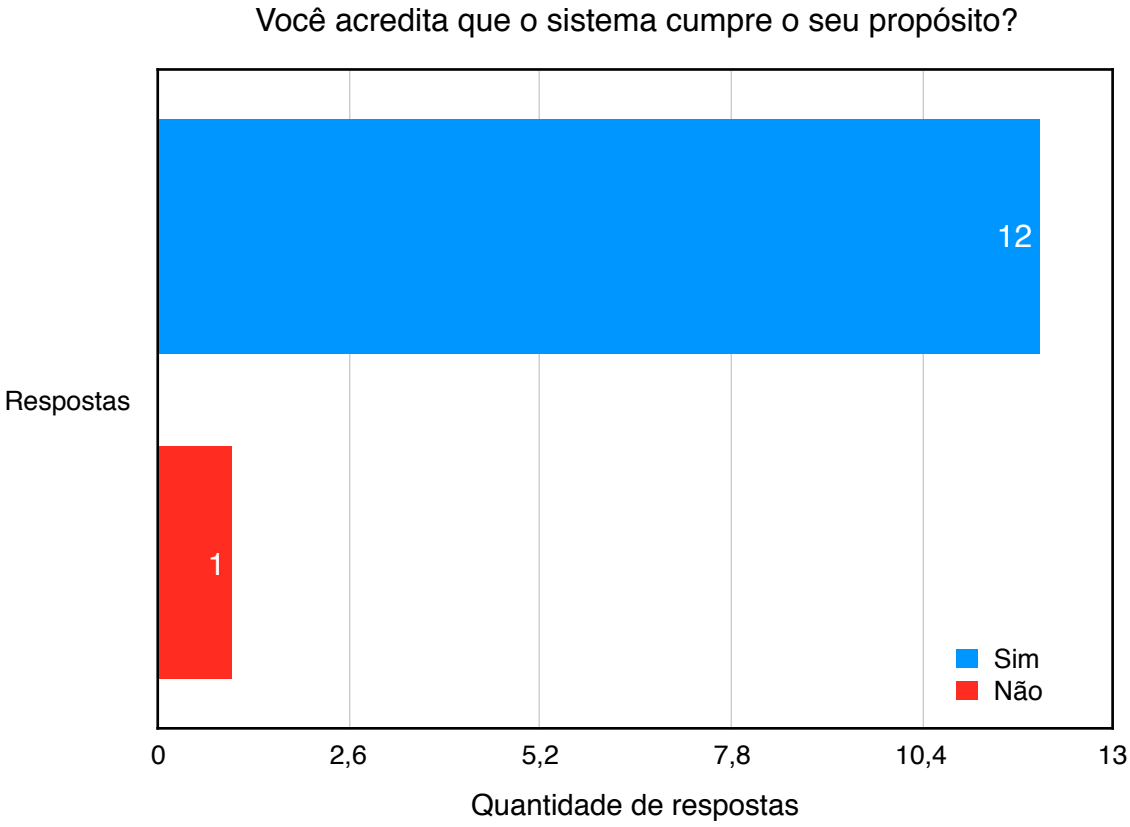


Fonte: Elaborado pelo autor

Como podemos observar no Gráfico 2, a maioria dos voluntários considerou o sistema desenvolvido inovador, o que confirma o atributo de qualidade *novelty*.

Questão 3. Você acredita que o sistema cumpre o seu propósito?

Gráfico 3 - Respostas da Questão 3



Fonte: Elaborado pelo autor

Conforme ilustrado no Gráfico 3, doze dos treze voluntários acreditam que o sistema de fato consegue assegurar a propriedade e imutabilidade dos documentos digitais.

7. CONCLUSÃO

Neste trabalho foram apresentados os conceitos e características principais da tecnologia *blockchain.* Apresentamos casos de uso da *blockchain* fora do contexto de criptomoedas e propomos um sistema que agregasse imutabilidade e propriedade à documentos digitais com ganhos em escalabilidade, abrindo um leque de possibilidades para diversos tipos de aplicações e consumidores. Utilizando o banco de dados descentralizado e distribuído BigchainDB junto com o IPFS foi possível desenvolver uma estratégia onde o documento binário foi separado da *blockchain*, o arquivo é armazenado no sistema de arquivos enquanto a *blockchain* registra dentro de uma transação apenas a referência para o documento no IPFS implicando em maior flexibilidade e desempenho para o sistema.

Foram explicadas as motivações, objetivos, técnicas, ferramentas e tecnologias utilizadas nesta prova de conceito. Este *software* foi documentado, implantado no serviço de computação em nuvem da Amazon e avaliado por estudantes de diversos cursos do Campus UFC Quixadá, obtendo um resultado positivo e satisfatório.

REFERÊNCIAS

10 criteria for choosing the correct framework. **Symfony.** Disponível em: <http://symfony.com/ten-criteria>. Acesso em: 08 mai. 2016.

SZABO, Nick. **Formalizing and securing relationships on public networks**. *First Monday* 2.9 (1997). Disponível em: <http://szabo.best.vwh.net/formalize.html#Building%20Blocks%20of%20Smart%20Contract%20Protocols>. Acesso em: 22 abr. 2016.

NAKAMOTO, Satoshi. **Bitcoin**: A Peer-to-Peer Eletronic Cash System. Disponível em: <https://bitcoin.org/bitcoin.pdf>. Acesso em: 22 abr. 2016.

WINTERS, Tristan. **The Greater Promise of a Blockchian.** Disponível em: <https://bitcoinmagazine.com/articles/greater-promise-blockchain-1391035696/> Acesso em: 17 jul. 2017

FERGUSON, Niels; SCHNEIER, Bruce. **Pratical Cryptography**. Wiley, 2013.

ANTONOPOULOS, Andreas. **Mastering Bitcoin:** unlocking digital cryptocurrencies. O'Reilly Media, 2014.

BENET, Juan. **Ipfs-content addressed, versioned, p2p file system**. *arXiv preprint arXiv:1407.3561,* 2014.

SOMMERVILE, I. **Engenharia de Software**. 8.ed. São Paulo: Addison Wesley, 2007.

TARKOWSKI, Susanne. **Bitnation**. Disponível em: <https://docs.google.com/document/d/1ZiIZ-rmI79HPNbfJ1AXwcgoe8TKMoUMatDf7YfO5LZw/edit>. Acesso em: 05 mai. 2016.

FUJIMURA, Shigeru. et al. **BRIGHT**: A concept for a decentralized rights management system based on blockchain. *Consumer Electronics-Berlin (ICCE-Berlin), 2015 IEEE 5th International Conference on*. IEEE, 2015.

LAMPORT, Leslie; SHOSTAK, Robert; PEASE, Marshall. **The Byzantine Generals Problem**. *Transactions on Programming Languages,* v.4, n. 3, p. 382-401, jul. 1982.

WOOD, Gavin. **Ethereum:** A secure decentralized generalised transaction generalised transaction ledger. Ethereum Project Yellow Paper 151 (2014).

MCCONAGHY, Trent. et al. **BigchainDB**: A Scalable Blockchain Database. Disponível em: <https://www.bigchaindb.com/whitepaper/>. Acesso em: 10 jun. 2016.

EVANS-GREENWOOD, Peter. et al. **Bitcoin, Blockchain & Distributed Ledgers**: Caught between promise and reality. Disponível em: <http://www2.deloitte.com/au/en/pages/technology/articles/distributed-ledgers.html>. Acesso em: 22 jun. 2016.

SCHWARTZ, David. et al. **The Ripple Protocol Consensus Algorithm**. 2014. Disponível em: <https://ripple.com/consensus-whitepaper/>. Acesso em: 21 jun. 2016

LUNDKVIST, Christian. **IPFS Introduction by Example.** 2014. Disponível em: <http://whatdoesthequantsay.com/2015/09/13/ipfs-introduction-by-example>. Acesso em: 17 jul. 2017.

DENNIS, Richard; OWEN, Gareth. **Rep on the block:** A next generation reputation system based on the blockchain, In: **The 10th International Conference for Internet Technology and Secured Transactions**, 2015.

GALANTE, Guilherme; BONA, Luis. **A Survey on Cloud Computing Elasticity.** *Fifth International Conference on Utility and Cloud Computing.* IEEE, 2012.

SCHREPP, Martin. et al. **Construction and Evaluation of a User Experience Questionnaire.** Alemanha, 2008.

ZUMWALT, Matt. et al. **The Decentralized Web Primer**. Disponível em: <https://flyingzumwalt.gitbooks.io/decentralized-web-primer/>. Acesso em: 05 jul 2017.

The Blockchain Explained to Web Developers, Part 1: The Theory. **Marmelab.** Disponível em: <http://marmelab.com/blog/2016/04/28/blockchain-for-web-developers-the-theory.html>. Acesso em: 15 mai. 2016.

An Introduction to IPFS. **Medium.** Disponível em: <https://medium.com/@ConsenSys/an-introduction-to-ipfs-9bba4860abd0>. Acesso em: 05 jul 2017.

Why No One Else Can Spend Your Bitcoins. **Tinker Coin.** Disponível em: <http://www.tinkercoin.com/blog/2013/07/why-no-one-else-can-spend-your-bitcoins>. Acesso em: 29 mai. 2016.

What is Blockchain. **Medium**. Disponível em: <https://medium.com/badge-chain/what-is-blockchain-5e4498f05c20#.iz6m0nyyw>. Acesso em: 29 mai. 2016.

The Blockchain for Education: An Introduction. **Hacker Education.** Disponível em: <http://hackeducation.com/2016/04/07/blockchain-education-guide>. Acesso em: 29 mai. 2016.

The MD5 Message-Digest Algorithm. **Internet Engineering Task Force.** Disponível em: <https://tools.ietf.org/html/rfc1321>. Acesso em: 18 jun. 2016.

Distributed Asset Ledgers. **Clearmatics.** Disponível em: <http://www.clearmatics.com/solution/distributed-ledgers/>. Acesso em: 22 jun. 2016.

Digital Asset. Disponível em: <https://digitalasset.com/faqs.html>. Acesso em: 21 jun. 2016.

Cryptocurrencies for Everyone. **Dmytro Pershyn.** Disponível em: <http://www.slideshare.net/DmytroPershyn/bitcoinpresentation-54663658>. Acesso em: 21 jun. 2016.

Can We Reach Consensus on Blockchain?. **Robert Eriksson.** Disponível em: < https://www.linkedin.com/pulse/can-we-reach-consensus-blockchain-robert-eriksson >. Acesso em: 21 jun. 2016.

Dicionário Online de Português. Disponível em: <http://www.dicio.com.br/fato/>. Acesso em: 17 jul. 2016.