UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS

Rafael Silva Montes

Tecnologias Blockchain e RFID aplicados na automação comercial

São Carlos

2018

Rafael Silva Montes

Tecnologias Blockchain e RFID aplicados na automação comercial

Monografia apresentada ao Curso de Engenharia Elétrica com Ênfase em Eletrônica, da Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Engenheiro Eletricista.

Orientador: Prof. Dr. Carlos Dias Maciel

São Carlos

2018

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da EESC/USP com os dados inseridos pelo(a) autor(a).

Silva Montes, Rafael
S779t Tecnologias Blockchain e RFID aplicados na automação comercial / Rafael Silva Montes; orientador Carlos Dias Maciel. São Carlos, 2018.

Monografia (Graduação em Engenharia Elétrica com ênfase em Eletrônica) -- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2018.

1. RFID. 2. Blockchain. 3. Automação Comercial. I. Título.

Eduardo Graziosi Silva - CRB - 8/8907

FOLHA DE APROVAÇÃO

Nome: Rafael Silva Montes

Título: "Tecnologia Blockchain e RFID aplicados na automação comercial"

Trabalho de Conclusão de Curso defendido e aprovado em 22 / 11 / 2018,

com NOTA 8,0 (oito, zero), pela Comissão Julgadora:

Prof. Associado Carlos Dias Maciel - Orientador - SEL/EESC/USP

Mestre Victor Hugo Batista Tsukahara - Doutorando - SEL/EESC/USP

Mestre Talysson Manoel de Oliveira Santos - Doutorando - SEL/EESC/USP

Coordenador da CoC-Engenharia Elétrica - EESC/USP:
Prof. Associado Rogério Andrade Flauzino

Este trabalho é dedicado aos meus pais e às minhas irmãs.

AGRADECIMENTOS

Agradeço aos meus pais, Paulo Sérgio e Maria Joana, por fazer da minha formação uma prioridade em suas vidas, garantindo, assim, que eu tivesse acesso a uma educação de qualidade, que foi o que permitiu cada conquista importante da minha vida.

As minhas irmãs, Priscila e Patrícia, eu agradeço pelo apoio, companheirismo, incentivo e exemplo. Pois elas são as fontes de motivação e espelho para eu mirar e seguir.

Ao Professor Doutor Carlos Dias Maciel pela paciência e por me oferecer um tema de caráter prático que foi uma importante fonte de aprendizado e fomento para seguir uma carreira voltada para engenharia.

Aos amigos que adquiri durante a graduação, cuja parceria foi importante para o nosso crescimento junto.

À Universidade de São Paulo, por oferecer um ambiente de formação repleto de oportunidades e grupos de extensão, que foram essenciais para minha formação humana e técnica.

A educação é a arma mais poderosa que você pode usar para mudar o mundo.

Nelson Mandela

RESUMO

MONTES, R. S. **Tecnologias Blockchain e RFID aplicados na automação comercial**. 2018. 76p. Monografia (Trabalho de Conclusão de Curso) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2018.

A partir do artigo publicado por Satoshi Nakamoto, o mundo se viu impactado pela novidade que este apresentava e começou a ter uma corrida para averiguar as possibilidades que a tecnologia Blockchain pode oferecer. Em um mundo globalizado, no qual as empresas devem investir na aceleração dos processos para ganhar competitividade torna-se necessário automatizar processos com a finalidade de torna-los mais rápidos. Novas tecnologias surgerem o estudo de aplicações originais afim de averiguar as possibilidades e a efetividade das mesma. Dessa forma este trabalho sugere o uso da tecnologia blockchain aliado ao undo de etiquetas RFID para realização da automação de lojas de varejo. A Blockchain é desenvolvida em linguagem python e a aplicação RFID utiliza da Raspberry Pi atrelado com o modulo leitor RFID MRFC522 Mifare. Neste trabalho foi desenvolvido as operações envolvidas em um nós e foi realizados abstrações sobre seu funcionamento em uma rede distribuída (P2P).

Palavras-chave: Blockchain. Etiquetas RFID. Automação Comercial.

ABSTRACT

MONTES, R. S. **Plant identification by applying the SIFT algorithm and color moment analysis to its leaves**. 2018. 76p. Monografia (Trabalho de Conclusão de Curso) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2018.

From the article published by Satoshi Nakamoto, the world was impacted by the novelty that it presented and began to have a race to investigate the possibilities that the technology Blockchain could offer. In a globalized world, in which companies must invest in accelerating processes to gain competitiveness, it becomes necessary to automate processes in order to make them faster. New technologies come from the study of original applications in order to ascertain the possibilities and the effectiveness of the same ones. In this way, this work suggests the use of blockchain technology along with the undo of RFID tags for the automation of retail store. Blockchain is developed in python language and the RFID application uses Raspberry Pi coupled with the MRFC522 Mifare RFID reader module. In this work the operations involved in a node were developed and abstractions were performed on its operation in a distributed network (P2P).

Keywords: Blockchain, Tags RFID, Commercial automation.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

RFID	*Radio Frequency Identification*
PoW	*Proof of Work*
PoS	*Proof of Stake*
PoC	*Proof of Capacity*
PoeT	*Proof of Elapsed Time*
P2P	*Peer to Peer*

SUMÁRIO

1 INTRODUÇÃO

Diante da atual conjutura social, dos padrões de exigência do mercado, do crescimento da humanidade e a busca por modos de produção viáveis, a automação surge como uma necessidade para que empresas aumentem a sua competitividade. No entanto deve-se entender os impactos desse processo na sociedade. Analisando sob o ponto de vista de um empresário, chega-se à conclusão sobre a viabilidade observando que é possivel obter resultados tão precisos quanto os obtidos com a utilização da mão de obra humana. Todavia num prazo reduzido. Outra vantagem, consiste na diminuição da necessidade de oferecer treinamento e gastar recursos com processos seletivos; menor investimento em trabalhadores e dessa forma redução de encargos trabalhistas; preocupação com segurança e índole do funcionário.

Entretanto, deve-se considerar o aspecto social ao implementar um processo de automação. Existe a ideia que a introdução das máquinas reduzirá o campo de trabalho, diminuindo, dessa maneira, a demanda de trabalho para pessoas menos qualificadas.

O objetivo desse trabalho é apresentar uma solução de automação de um supermercado convencional, preciso e menos custoso, utilizando de dois conceitos que estão em ênfase: o Blockchain e aplicações de internet das coisas (IOT). A introdução desses conceitos podem dar ao mercado características que seriam altamente vantajosas para o varejista e para os consumidores.

Com o surgimento da tecnologia Blockchain, a maneira de se fazer negócios está sofrendo mudanças. De um modo seguro e rápido é possível realizar transações comerciais utilizando esta estrutura de dados. Desta forma, pensar em aplicar tal inovação em novos contextos torna-se uma tendência. Segundo o ponto de vista de Mougayar (2017), o Blockchain é a maior revolução tecnológica desde a internet. Seu conceito se baseia no princípio de que é possível realizar operações mercantins e contratuais, de forma que a confiança não está centralizada em um banco de dados, mas sim dispostos numa rede de usuários conectados. Essa rede é o que denominamos de Blockchain.

A segurança do blockchain é baseada no fato de que as transações são registradas em blocos, que são datados, e um bloco sempre se refere ao bloco seguinte e eles são dotados de criptografia, que é a base da segurança do Blockchain. Outro fator que torna a Blockchain é a sua aplicação distribuída.

O termo criptografia traz consigo a noção de segurança. Pois há uma certa singularidade no decorrer dos processos que é possível saber se houve alterações de informações; há a presença de chaves, sendo uma de domínio público e uma fica de posse do usuário, o que dá privacidade para o mesmo; há como autenticar a assinatura de um documento ou

mensagem através de uma algorítimo matemático.

Dessa forma a implementação da tecnologia Blockchain no ambiente de um mercado traz vantagens tanto para o verejista como para o cliente.

O Varejista teria maior controle sobre as operações financeiras do mercado. Dessa forma grandes redes de supermercado terão maior confiança sobre a legalidade das informações passadas pelos colaboradores. As transações poderão ser mensuradas de forma instantânea. Permite o controle dos produtos em estoque

Os usuários terão como vantagem maior transparência sobre as transações e pederão ter privacidade de seus dados pessoais.

O blockchain por si só traz muitas vantagens para o supermercado, mas não garante uma automatização do supermercado. A essência da automação está no uso de etiquetas RFID (Radio Frequency Identification).

Tais etiquetas permitem dá unicidade para cada objeto. Essas características são a chave para a idealização desse trabalho. Identificar produtos semelhantes de forma única é poder associar cada um como uma propriedade inteligente em uma transação blockchain. Dessa forma, quando um consumidor identificar interesse em um produto ele será identificado como possivelmente dele até que a transação seja concluída, isto é, quando o usuário pagar pelo mesmo.

De forma resumida, idealizando um mercado em que todos os produtos possam ser identificados por etiquetas RFID, cada uma dessas terá um código único que diferenciará produtos, por mais idênticos que os mesmos sejam entre si. O usuário, quando identificar aquele produto como dele adicionará a esse código uma identificação única que corresponde a sua identidade de comprador. Ao ser pago o produto, será registrado no blockchain que aquele produto pertence aquele usuário.

O maior meio de verificação se o usuário está levando os produtos pelos quais pagou é que na saída tenha um leitor que lerá todos os códigos e se for identificado um que não contenha seu código de identificação, seja o produto registrado como um produto sem dono ou que sendo de outro usuário será emitido um sinal sonoro.

As vantagens desse método é que não haverá filas longas para a conferência e pagamento dos produtos, já que não será necessária uma conferência produto a produto. E como benefício há o fato de usuário ter privacidade e controle sobre as suas compras. Para o varejista esse procedimento reduz recursos de força de trabalho e terá maior conferência sobre o repasse financeiro que lhe cabe.

Para a realização desse trabalho, deve ser entendido a tecnologia RFID e a dinâmica de funcionamento de uma blockchain, mais especificamente, questões acerca da criptografia, definição dos blocos, àrvore de merkle e toda a dinâmica de funcionamento.

Para atender o objetivo proposto, será montato o algorítimo de uma blockchain que receberá como entrada a leitura de etiquetas RFID e será abstraído um modelo de consenso que possa ser aplicado no varejo.

2 CONCEITOS E BASE TEÓRICA

2.1 A importância do processo de automação comercial no Brasil

O propósito deste trabalho é propor uma alternativa de automação comercial utilizando do conceito de Blockchain, para o registro e controle das operações, atrelado ao uso de tags RFID, tecnologia chave para a automação a ser proposta, como uma tentativa de acelerar os processos dentro de um comércio. Para tanto, deve ser entendido a importância e as motivações para a busca por alternativas de automação comercial. Circunstâncias essas a serem exploradas nesse capítulo.

2.1.1 Automação

> Pela ferramenta que se interpõe entre o homem e a natureza, criação humana para aumentar sua capacidade de transformação do material, o homem faz com que o material se torne órgão humano. A ferramenta é uma extensão do corpo humano, da mão humana, é uma mão capaz de assumir múltiplas formas e funções. (REIS, 2015).

Seguindo a analogia da citação acima, podemos dizer que os computadores e as máquinas são extensões dos seres humanos, a fim de aumentar suas capacidades intelectuais e físicas. Desta forma, quando estas são aliadas a atividades industriais, aumenta-se o potencial competitivo de uma corporação. Tal processo é o que denominamos automação.

Segundo (ROCKENBACH, 2005), entende-se por automação os sistemas nos quais os processos operacionais são controlados e executados por meio de dispositivos mecânicos ou eletrônicos, substituindo o trabalho humano. Dessa forma, automatizar seria fazer o uso de máquinas para agilizar e otimizar a produção ou serviços.

De acordo com (BOTEGA, 2005)

> A automação é o conjunto das técnicas baseadas em máquinas e programas com o objetivo de executar tarefas previamente programadas pelo homem e de controlar sequências de operações sem a intervenção humana. Através de intertravamentos (sequências de programação) do sistema, o usuário consegue maximizar com qualidade e precisão seu processo produtivo, controlando, assim, variáveis diversas (temperatura, pressão, nível e vazão) e gerenciamento a distância de toda a cadeia produtiva.

(QUINTÃO et al., 2008) acrescenta a ideia de que “Deve-se acrescentar à máquina algum tipo de inteligência, através de procedimentos ou programas, para que ela execute a

sua tarefa, tarefa essa pré-definida, de modo mais eficiente e com vantagens econômicas e de segurança."

Dessa forma, pode ser aferido que a automação de uma empresa deve ser executada se os processos a serem otimizados compensarem e não manifestar nenhuma periculosidade para clientes e possíveis funcionários.

2.1.2 Vantagens da automação

Como destacado, o aumento da produtividade é conferido pela automação de um processo que antes necessitava de esforço humano. Afinal, erros devido a falhas humanas e o desgaste da máquina para o trabalho, principalmente se comparamos com trabalhos repetitivos, são menores em relação ao trabalho humano. Com o desenvolvimento da técnica do processo produtivo pode ser aferida uma maior produção, num determinado período de tempo.

Outro valor agregado ao processo de automação a ser destacado é a redução de custos. Segundo (QUINTÃO et al., 2008) a redução orçamentária está relacionada a redução de operadores, melhor aproveitamento da matéria-prima e diminuição do tempo de ociosidade dos equipamentos.

Como já pontuado, a automação traz como consequência o aumento da produtividade. Tal característica está associada à redução de tempo para execução dos processos, minimização de erros e maior precisão. Segundo (QUINTÃO et al., 2008), a diminuição de erros é consequência de toda a configuração dos equipamentos usados na automação ser mediante a fatos automáticos.

Diversos autores defendem que a automação elimina os funcionários do esforço para atividades repetitivas, que pode ocasionar erros. (BORTOLINI; CONTÍNUOS,):

> A automação oferece a racionalização dos processos; eliminação de tarefas repetitivas; minimiza erros em controles manuais; melhora o atendimento ao cliente interno e externo devido à alta qualidade e rapidez nas operações; além de permitir o aproveitamento das informações em tempo adequado para a tomada de decisão.

O mesmo raciocínio pode ser inferido a partir de e (BOTEGA, 2005). Segundo a autora, a automação é responsável pela eliminação de atividades desgastantes, insalubres ou de risco para o funcionário, que não agregam produtividade.

A automação também proporciona a criação de novas oportunidades. Segundo (QUINTÃO et al., 2008):

> Criação de oportunidades – com a eliminação de alguns operadores (que realizam atividades repetitivas e que pouco agregam valor a sua formação), há oportunidades para que operadores remanescentes

> se preocupem com outras questões importantes como a qualidade do produto.

Podemos inferir que a automação permite o melhor uso da mão-de-obra, aproveitando-a em atividades que fazem maior uso do intelecto.

A criação de oportunidades também está relacionada com o aumento da produtividade que pode trazer a conquista de novos clientes. Tal consequência foi pontuada por e (ANDRADE, 2008) ao analisar o processo de automação da empresa Garoto-SA.

Na sua análise, foi observado aumento no número de colaboradores. Do período de 2002 até 2006 a empresa gerou 783 oportunidades de emprego, o que resultou em um aumento de cinquenta e três por cento do valor da folha de pagamento. Esse crescimento está diretamente relacionado com o aumento da competitividade da empresa gerada pela automação de processos. Este aumento resultou da necessidade de expansão da produção, o que necessitou de um quadro maior de funcionários.

Em resumo, houve melhoria das condições de trabalho em alguns processos por conta da introdução de tecnologias automotivas. Na produção resultou um melhor aproveitamento dos trabalhadores para melhorar a qualidade dos produtos, o que resultou no crescimento da empresa.

Desta forma, podemos inferir que a automação também pode contribuir para um melhor aproveitamento dos funcionários. Segundo (ZUBOFF,): “É o conhecimento e a compreensão na cabeça das pessoas – suas qualificações intelectivas – que transformam as máquinas inteligentes numa máquina de melhorar fundamentalmente os negócios”.

Entretanto, não se pode inferir que toda mão de obra usada antes de um processo de automatização das operações será reaproveitada em outras atividades. Nem sempre é possível fazer esse reaproveitamento e nem sempre é do interesse da empresa, pois esta busca a redução de custos de produção, como os gastos com funcionários, para aumentar a sua competitividade.

Outras reduções de custos que o mencionado processo pode agregar é a possibilidade de um melhor gerenciamento do uso das matérias-primas, aliado à diminuição do tempo de ociosidade das máquinas, o que intensifica o processo produtivo.

A melhoria da qualidade dos resultados obtidos em um ambiente automatizado, segundo (QUINTÃO et al., 2008):

> Aumenta a confiabilidade das informações geradas no decorrer do processo, auxiliando também na qualidade dos produtos produzidos via aprimoramento dos dados que irão configurar os equipamentos. Sistemas automatizados garantem um grau de repetibilidade maior, melhorando o controle da qualidade – obtém-se menos desvios na produção.

(BOTEGA, 2005) defende que a partir de sequências de programação do sistema, a corporação consegue obter melhoras na precisão do processo produtivo a partir de um melhor controle das intempéries do sistema. Dessa forma obtém-se melhora na qualidade, o que resulta em maior competitividade para a empresa.

Alguns pontos importantes devem ser avaliados em relação a automação. O primeiro deles trata-se da dependência tecnológica, isto é, caso ocorra falha técnica em um equipamento pode haver consequências para toda a produção. Por tornar a produção mais rápida, um erro – de configuração, por exemplo – na produção pode gerar vários produtos defeituosos, o que seria um desastre devido ao gasto com energia e de matéria-prima.

Dessa forma, a empresa deve fazer manutenções constantes em seus equipamentos e estabelecer protocolos de identificação de erros e da forma como deve proceder, quando houver falhas técnicas, de forma a obter o menor prejuízo possível.

Um outro ponto a ser analisado é o investimento inicial. Os responsáveis devem avaliar a viabilidade de um processo de automação, avaliando se o tempo de retorno de investimento e a economia de gastos obtidos após o processo e as outas vantagens já mencionadas, justificam o investimento.

A automação pode ser associada a impactos negativos na comunidade onde a empresa atua, pois ela pode trazer como resultado o aumento da taxa de desemprego regional. (SANTOS et al., 2000):

> O desemprego contemporâneo está associado a diversas coisas. Fala-se do desemprego estrutural decorrente das mudanças tecnológicas ou da alteração dos padrões de consumo [...] . Há aumento dos índices de desemprego em razão do maior uso da técnica que dispensam mão-de-obra humana, gerando um excedente de trabalhadores que não são absorvidos pelo mercado de trabalho.

Entretanto, não investir em automação pode significar perda considerável de competitividade em relação a concorrentes. Tal conclusão pode ser inferida a partir do seguinte trecho de (MORAES; CASTRUCCI, 2000):

> [...]pode-se inferir que a automação deixou de ser opcional no atual mundo dos negócios. A automação na indústria decorre de necessidades, tais como maiores custos de trabalho, menores perdas de materiais e menores custos de capital, maior controle das informativas relativas ao processo, maior qualidade das informações e melhor planejamento e controle de produção.

2.1.3 A automação nos supermercados: evoluções e desafios

O primeiro indício de automação comercial surgiu com a invenção da caixa registradora mecânica, que tinha como principal finalidade o controle das operações financeiras,

para evitar o furto por parte dos funcionários. Ela registrava todas as operações e só era possível abri-la após ser realizada uma transação por intermédio dela.

Sua versão eletrônica surgiu na década de 1970. Nos anos que se seguiram surgiram novos equipamentos para auxiliar os operadores de caixa e tornarem as operações mais rápidas. Foram os PDVs. São exemplos desses equipamentos:

- Teclado;
- Leitor óptico;
- Monitor;
- Impressoras (de cheques e de cupom fiscal).

São equipamentos comumente encontrados em supermercados atualmente. A partir do uso do computador para o auxílio do operador, novos serviços começaram a ser oferecidos no caixa, devido ao contato direto com o banco. Isto foi tanto vantajoso para o comerciante como para os bancos, no que se refere à redução de custos.

As autoridades fiscais concluíram que a automação comercial pode ser usada como importantes instrumentos de fiscalização. A partir disso foram criados convênios fiscais para normatizar esses instrumentos. Segundo (MELO; JUNIOR, 1997) as principais exigências eram:

- Os equipamentos devem conter memória fiscal inviolável.
- Não deve existir algum recurso que iniba o registro de operações.
- Devem ter totalizador geral, totalizadores parciais e contador de ordem de operação.
- Bloqueio automático de funcionamento ante a perda de dados acumulados nos totalizadores e acumuladores.
- Os contadores e temporizadores deverão ser mantidos em memória residente por no mínimo 720 horas, mesmo na ausência de energia elétrica.

Na década de 90, ocorreu, no Brasil, o surgimento de incentivos por parte dos governos estaduais para que os comércios implementassem instrumentos de automação. Tratavam-se de descontos de investimentos ou parte destes no pagamento do ICMS do estabelecimento.

Um fato importante, principalmente no que se refere a proposta desse trabalho, é que as indústrias que possuem produtos, que passaram a ser comercializados em mercados, precisaram providenciar para que seus produtos fossem adequados a automação comercial.

Segundo (MELO; JUNIOR, 1997), foi crescente a pressão para que os fornecedores adotassem código em seus produtos. Em 1994, 6,4 mil indústrias já forneciam produtos com código de barras. Apenas 2 anos depois, 1996, esse número cresceu para 19 mil. Tal fato deve ser considerado em termos deste trabalho, pois a proposta de automação comercial sugerida depende que os fornecedores identifiquem seus produtos com etiquetas RFID.

Percebe-se que até o momento, os ditos processos de automação comercial possuem grande dependência de um operador de caixa. Entretanto, já existem soluções que eliminam o uso de operadores, como por exemplo, os supermercados em que são os clientes que fazem a identificação e a conferência dos produtos. O que dá alguma segurança para o comerciante sobre a idoneidade do cliente, é a conferência que é feita por meio de uma balança, que identificará se o peso do produto indicado está na faixa do que se espera de produtos com aquela característica. Essa metodologia não acarreta agilidade dos processos, pois seu objetivo único é a eliminação do operador.

Recentemente, a Amazon lançou a Amazon Pró, que é um mercado que faz demasiado uso de sensores e de visão computacional. O sistema de compra dessa loja tem como principal característica a agilidade do processo de compra.

A dinâmica de compra funciona da seguinte maneira: O cliente ao entrar na loja se identifica por meio de um aplicativo em seu próprio celular, recolhe os produtos que deseja levar e sai da loja. O boleto da compra realizada é enviado para o e-mail do cliente. As câmeras e os sensores da loja são utilizados para identificar o comprador e os produtos que o mesmo adquiriu. Assim, o processo de verificação dos produtos a serem comprados é imediato.

Existem protótipos de automação de supermercados utilizando Tags RFID, mas ainda não houve uma implantação efetiva. Observa-se que dessas alternativas apresentadas nenhuma traz novidade em relação ao registro irrefutável das transações, de forma a auxiliar na contabilidade e na comprovação de declarações fiscais.

Com a utilização do Blockchain pretende-se obter um registro fiel das transações, respeito a sua privacidade, facilidade de declaração fiscal, impossibilidade de algum funcionário adulterar o registro para benefício próprio. Assim haverá segurança fiscal para o Estado e segurança administrativa para o comerciante, principalmente para grandes redes de supermercado. A avaliação do uso da tecnologia Blockchain para a automação comercial, usando tags RFID, é o objetivo central desse trabalho.

2.2 Conceitos técnicos específicos

Este trabalho de conclusão de curso apresenta análises sobre a possibilidade de se implementar sistemas IoT com registro em uma Blockchain, em particular, com ênfase

em automação comercial. O uso do Blockchain implica em uma busca pela segurança no registro das informações e expressa o desejo de que essas não sejam alteradas e que haja uma forma de verificar a sua integridade, enquanto que a utilização das aplicações de internet das coisas (IoT) tendem a deixar os processos mais dinâmicos e ágeis. Dessa forma, é possível inferir que o estudo da complementaridade dessas tecnologias é benéfico e pode trazer transformações positivas para a sociedade.

A implementação IoT proposta nesse trabalho ocorre através da utilização de etiquetas RFID que serão utilizadas para identificação de produtos no mercado. Para que o estudo seja completo, deve ser compreender as fragilidades que tais etiquetas podem inferir para um sistema comercial e apresentar as soluções propostas até o momento acerca da leitura e da escrita das tags RFID. De forma análoga deve ser entendido os protocolos de execução de Blockchain e abstrair qual será o mais efetivo para a aplicação proposta e que corresponderá aos requisitos de segurança esperadas do registro de estabelecimentos comerciais.

Neste capítulo serão abordados conhecimentos específicos sobre as tecnologias envolvidas na resolução do objetivo principal deste trabalho.

2.2.1 RFID - *Radio Frequency Identification*

As etiquetas RFID (*Radio Frequency Identification*), isto é, as etiquetas cuja identificação é feita via radiofrequência, possuem uma ampla gama de aplicações no mercado e tem ganhado grande importância com o aumento das aplicações de IoT (Internet of Things). Podem ser utilizadas para identificação de objetos para controle de estoque, identificação dos dados de um gado, chaves para entrar em edifícios, entre outros.

Segundo (DIAS, 2016), uma etiqueta RFID é constituída por:

- Um microchip ou circuito integrado, no qual os dados ficam armazenados na memória deste. Ele também é responsável por diversos procedimentos imprescindíveis para ocorrer a comunicação com o leitor, ou seja, ele deve possuir uma capacidade de processamento.
- Antena. Ela é responsável por receber e enviar ondas de radiofrequência. O formato da antena depende da frequência de operação do sistema.
- Conectores. Responsáveis por ligar os CIs à antena.
- Substrato. Refere-se a base de sustentação da antena, do CI e dos conectores. Contém características elétricas que são consideradas no projeto da antena.

A autora, (DIAS, 2016), explica o funcionamento de um sistema RFID, o acoplamento de *backscatter*. Tal acoplamento é utilizado em frequência UHF e pode proporcionar leituras de até 15 metros. A seguir, foi elaborado um resumo, em passos, da exposição da autora.

- O leitor, por meio de uma antena, transmite sinais de radiofrequência.
- A etiqueta que os receber, obterá energia desses sinais que será utilizada para o CI executar internamente as suas funções
- A etiqueta retorna os dados armazenados em sua memória através de outro sinal de radiofrequência.

O leitor, ao receber os dados, deve amplificar os sinais recebidos, modular e organizar os dados recebidos da etiqueta RFID.

Segundo (ABRAHAO et al., 2007): "A comunicação é feita remotamente por rádio e frequência e esta característica permite que a etiqueta seja suscetível a uma série de ataques que tentam fraudar ou obter informações do usuário sem a sua autorização".O mesmo autor,(ABRAHAO et al., 2007), elencou os principais ataques aos quais o sistema RFID está suscetível, que são: negação de serviço, impersonificação, vazamento de informação e rastreamento malicioso.

A negação de serviço pode ser tanto conferida por meio da interferência entre ondas de rádio como, por exemplo, a partir do sobrecarregamento do CI com solicitações de leituras a uma taxa maior do que a suportada pelo mesmo. Impersonificar consiste na clonagem não autorizada das etiquetas RFID. Isto é, munir uma etiqueta com as configurações e informações de uma outra, podendo fazer uso fraudulento dessa nova etiqueta.

Como exemplo de impersonificação podemos citar os ataques *Cloning* e os do tipo *Replay*. Os ataques do tipo *Cloning* consitem na realização da escrita em uma tag de uma identificação obtida a partir da leitura de outra tag. Essa técnica pode ser danosa, por exemplo, a sistemas de segurança em prédio nos quais a entrada e saída dos moradores são realizadas a partir da identificação de tags RFID. O ataque do tipo *Replay* é parecido com o *Cloning*. Nessa técnica o atacante intercepta as informações de uma leitura legítima de uma tag RFID e utiliza as informações obtidas para burlar o sistema de leitura do leitor.

O vazamento de informações pode ocorrer caso não seja feito o uso de um protocolo de segurança para garantir que somente quem tem acesso ao protocolo possa ter acesso a informação. Ataques do tipo *Sniffing* é um exemplo de ataques desse tipo. O rastreamento malicioso consiste na identificação de uma falha em um protocolo de segurança a partir, por exemplo, de uma correlação de diversas leituras. Tais falhas sugerem um cuidado ao implementar a tecnologia RFID de forma a aplicá-la de forma segura. Para tanto, deve-se procurar ou elaborar um protocolo de segurança que dê suporte à aplicação desejada.

Outro tipo de ataque é o RFID *exploits*, segundo (FILHO, 2009), é utilizado para explorar vulnerabilidades de segurança em componentes de processamento de dados. Os sistemas RFID também estão sujeitos a virus, programas que criam cópias de si mesmos

com a finalidade de infectar outros dispositivos, podem impactar no desempenho, alterar os dados das tags, acessar informações confidenciais e monitorar a utilização de componentes.

Para proteger as etiquetas desses ataques, técnicas de criptografia foram aplicadas para auxiliar na autenticação de uma leitora. De forma que a *tag* identifique a leitora com uma autorizada a receber as informações antes de as envia-las e a leitora reconheça a tag como uma autentica dentro do sistema em que ela está inserida.

A autenticação de forma geral é obtida por meio de protocolos de desafio/resposta nos quais uma parte A envia para B um problema a ser resolvido e caso B devolva a resposta estipulada, A enviará para B suas informações. Deve-se tomar muito cuidado na elaboração desses protocolos, pois qualquer falha pode ser explorada por um atacante.

(TANENBAUM, 2011) cita quatro regras que o projetista de um mecanismo de autenticação deve seguir. Tais regras são:

- Fazer com que o transmissor prove quem é antes de o receptor responder.
- Fazer que o transmissor e o receptor utilizem chaves específicas para provar quem são, mesmo que isso signifique ter duas chaves compartilhadas
- Fazer com que o receptor e o transmissoar extraiam seus desafios de conjuntos distintos
- Tornar o protocolo resistente a ataques que envolvam uma segunda sessão paralela, nos quais as informações obtidas em uma uma sessão são usadas em uma sessão diferente.

(ABRAHAO et al., 2007) define alguns protocolos, dos quais podemos citar o Desafio-Resposta, *Hash Look* e o desafio resposta com aplicação de pseudônimos. No desafio resposta, um número gerado de forma pseudo aleatória é enviado para a etiqueta. A etiqueta responde com a sua identificação e por número que é gerado por uma função que tem como variáveis a identificação e o número inicial enviado.

No protocolo *Hash Look* o leitor envia solicitações de acessos às identificações da etiqueta que, ao recebe-la, retorna uma função chamada de metaID. A etiqueta compara a função hash armazenada como identificação com a que ele calcular a partir dessa chave enviada pelo leitor. Se os hash forem iguais, ele envia a resposta.

O funcionamento do desafio resposta com aplicação de pseudônimos consiste no envio de uma chave do leitor para a etiqueta. Em posse dessa chave, a etiqueta irá processar essa chave e verificar a autenticidade do leitor. Caso este esteja adequado, a *tag* envia uma chave para o leitor para que este a autentique.

2.2.2 Criptografia

Para compreender os assuntos tratados nesse trabalho deve-se ter um entendimento de alguns tópicos de criptografia, principalmente sobre o que é um *hash* e como obter a

assinatura digital de uma informação.

A cripitografia, ciência de codificação ou cifragem de mensagens, surgiu a partir da necessidade de enviar mensagens sensíveis através de um intermediário. Para garantir que, com a captura do mensageiro, a informação ficasse protegida, surgiram as primeiras formas de codificar os textos.

A exemplo disso, é possível citar a Cifra de César, criado pelo próprio imperador romano Júlio César, 58 a.C, para enviar mensagens para os seus comandantes, diminuindo assim o risco de outra pessoa conseguir ler a mensagem. Entretanto, a metodologia de César é muito simples, não garantindo assim uma segurança absoluta. Essa cifra consistia em substituir cada letra por uma que estivesse algumas posições na frente desta na ordem alfabética.

Na história da criptografia é possível encontrar inúmeros métodos de codificação de mensagem. Alguns usando métodos simples de substituição silábica, como por exemplos a Cifra de Cesar e o ROT13, e outros mais sofisticados, como exemplo o algoritimo *Playfair*, que surgiram para proteger dados sensíveis.

No caso, o *Playfair*, foi utilizado na primeira guerra mundial pelos governos dos Estados Unidos e Inglaterra. Nesse algoritmo a encriptação é realizada pela tradução de dois caracteres por vez e há todo uma metodologia de codificação que faz o uso de uma tabela de 5 linhas e 5 colunas formada a partir de uma palavra chave, no qual a codificação segue requisitos referente a posição do par de letras a serem codificadas.

A criptoanálise, a tentativa da descoberta da mensagem criptografada, pode trazer impactos negativos para corporações e governos, por isso a procura pelo algoritmo sem furos é uma constante na história da humanidade, o que justifica os inúmeros códigos existentes. Um exemplo do impacto da criptoanálise foi o impacto que o estudo de Allan Mathison Turing, a cerca da Enigma, máquina de rotação de um conjunto de cilindros de codificação utilizado pela Alemanha na segunda guerra mundial, que resultou na descoberta de mensagens críticas que auxiliaram o governo inglês nessa grande guerra.

A partir da criptografia é possível obter a integridade, autenticidade, confidencialidade e o não repúdio da informação. A criptografia faz uso de operações de substituição – que seria a troca de símbolos – e transposição, que consiste no rearranjo dos elementos do texto, podendo não necessariamente fazer o uso dos dois métodos. Deve-se considerar que todo processo de criptografia é reversível e que necessariamente não deve haver a perda de nenhuma informação.

A criptografia é classificada quanto ao número de chaves utilizadas. Dessa forma ela pode ser tida como simétrica ou assimétrica. A simétrica se refere as técnicas que fazem o uso de uma única chave, tanto para criptografar como para descriptografar. Já a criptografia assimétrica tem como característica o uso de duas chaves, sendo que uma é de

acesso público e uma de privado, que se complementam. Isto é, se for feita a criptografia de um conteúdo usando uma chave privada, todos que tiverem acesso à chave pública poderão decodifica-la.

De forma semelhante, se for utilizada a chave pública de alguém, apenas essa pessoa pode descriptografar com a sua própria chave. Apesar de as chaves públicas e privadas serem matematicamente relacionadas, não é possível obter a chave privada por meio da pública. A criptografia assimétrica exige mais em termos de processamento do que a simétrica. Entretanto, o uso de duas chaves deixa o processo mais seguro.

A assinatura digital é um recurso para conferir a integridade e a autenticidade. Ela é obtida pelo uso de criptografia assimétrica, que pode ser combinada com o uso de *hash*.

O *hash* é definido como um código de tamanho físico que representa uma informação, independentemente do tamanho da mesma. Cada mensagem deve ter seu *hash* exclusivo, ou seja, não deve existir um *hash* que identifique informações diferentes.

Para se fazer a assinatura digital de uma informação, primeiramente deve-se usar um algoritmo do *hash* para obter o correspondente da informação. Em seguida, o autor da informação deve usar a sua chave privada para criptografá-la. Nesse ponto a assinatura já foi implementada. Para conferir se a informação associada a ela corresponde mesmo ao que originalmente foi assinado, deve ser utilizada a chave pública para descriptografar a assinatura a fim de obter o *hash* do documento original. E comparar com o *hash* do documento ao qual ele estava sendo associado. Se forem iguais, tanto a autoria quanto a integridade da informação ficarão mantidas. Se o processo fosse realizado sem o uso do *hash* o receptor, interessado em conferir a autenticidade da informação, deverá conferir se toda a informação codificada corresponde a que ela está associada. Entretanto, dependendo do tamanho do conteúdo, este processo pode ser inviável. O uso do *hash*, como indicado anteriormente, é indicado pelo fato de a conferência ser simplificada, já que o *hash* é um resumo de tamanho fixo.

2.2.3 Blockchain

De acordo com (MOUGAYAR, 2016):

> Em sua essência, o Blockchain é uma tecnologia que grava transações permanentemente de uma maneira que não podem ser apagadas depois, somente podem ser atualizadas sequencialmente, mantendo um rastro sem fim [...].

Complementando a definição acima, o Blockchain é uma forma de registro de dados no qual o mesmo é feito em blocos sequenciados que obedecem a determinadas regras, que serão discutidas mais a frente deste tópico.

O conceito de Blockchain ganhou o devido destaque a partir da publicação do artigo de Satoshi Sakamoto que estipula uma forma de uso dessa tecnologia aliada com criptografia e redes distribuídas para criação de uma criptomoeda, o Bitcoin. As ideias de Sakamoto deram o impulso que era necessário para concretizar o uso dessa tecnologia, para encontrar soluções mais efetivas e seguras tanto para o setor financeiro como para outros setores econômicos da sociedade.

(TAPSCOTT; TAPSCOTT, 2016) enumeram 6 razões principais para considerar o Blockchain uma fonte de mudanças no setor bancário. Essas são: atestação, custo, velocidade, gestão de risco, inovação de valor e código aberto.

A atestação expressa a característica do blockchain de não necessitar de um intermediário para realizar uma transação. É possível, de forma imediata, saber se houve alguma alteração em algum bloco do registro. Os blockchains podem ser implementados de forma distribuída, numa rede P2P. Assim, todos os usuários da rede podem cooperar guardando o registro dos dados, fato esse que agrega maior segurança para a manutenção dos dados.

O custo referido pelos autores, (TAPSCOTT; TAPSCOTT, 2016), se trata dos custos de oportunidades, expressa pela não dependência dos processos dos intermediários que ocorre segundo as regras dos mesmos, de forma a agregar valor para eles, e da não obrigação de se efetuar pagamentos de taxas extras por serviços do intermediário.

Essa demora para realizar transações a níveis globais pode ser substituída por uma operação mais rápida a partir de uma rede blockchain. Essa velocidade para realizar as operações também é considerada como uma das razões pelas as quais os sistemas financeiros atuais sobrerão alterações.

A eliminação de processos demorados contribui para uma melhor gestão dos riscos envolvidos em transações bancárias, como os de liquidação – cancelamento da transação por inconformidade no processo – , o de inadimplência – se trata da eventualidade de uma parte ficar inadimplente antes do término da transação – e sistêmico, que se refere a possibilidade de uma das partes não cumprir o acordo estabelecido. A gestão desses riscos também é considerada por (TAPSCOTT; TAPSCOTT, 2016) como um dos fatores, segundo eles:

> *. Os contadores poderiam acompanhar a qualquer momento a situação de cada transação que estivesse acontecendo e como estariam sendo registradas na rede. Transações irrevogáveis e reconciliação financeira instantânea eliminariam um dos aspectos observados pelas agências de riscos – o risco de gestores irresponsáveis explorar brechas e esconder irregularidades.*

O fator inovação de valor se refere ao uso da tecnologia para transações que não

sejam na forma de moeda, mas a de outros ativos financeiros como propriedades, barras de ouro, barril de petróleo, entre outros. Enquanto que o fator código aberto se refere ao potencial de a rede estar sempre se inovando em aspectos técnicos como segurança e elaboração de novas possibilidade.

O Blockchain é a tecnologia que permitiu a criação de criptomoedas. Satoshi Sakamoto empregou tal tecnologia para resolver o problema dos gastos duplos. Isto é, ele implementou uma solução que dá a um ativo digital a característica de unicidade. Isto é, a moeda é única. Ao fazer uma compra você cede uma quantidade de dinheiro em troca de um benefício, que pode ser um produto ou um serviço. Ao final do processo, percebe-se que você possui menos dinheiro do que possuía antes. Com a mesma moeda que você adquiriu o benefício você não consegue comprar outra coisa, pois essa moeda financeira não lhe pertence mais.

Não se tinha um modo de garantir transações sem ter uma terceira parte, confiável, para conferir as transações. Pois, digitalmente, é possível fazer cópias de arquivos, programas, fotos, entre outros. Com aplicação direta de Blockchain obteve-se uma solução, pois ele registra todas as operações de forma que é possível verificar os ativos de cada usuário. As criptomoedas atrelam a essa tecnologia conceitos de criptografia e de redes distribuídas do tipo P2P. Tal aliança de conceitos torna complicada e inviável a realização da violação das plataformas Blockchains das criptomoedas.

Segundo (MOUGAYAR, 2016): “Uma das questões desafiadoras relativas as criptomoedas é a volatilidade de preços, que é suficiente para manter a maioria dos consumidores longe”. Tal volatilidade é associada à especulação devida a incerteza acerca do futuro. Apesar de estar aumentando o número de criptomoedas, alguns governos ainda apresentam uma resistência para aceitá-las. Entretanto, é inegável que a solução da aplicação de Blockchain pode ser usada como moeda e apresenta considerável segurança.

O mesmo autor, (MOUGAYAR, 2016), afirma que já existem governos que usam o Blockchain para oferecer serviços para a sua população. A Ucrânia, por exemplo, criou uma plataforma na Etherium, um meio onde é possível implementar soluções com Blockchain, que permitem a realização de eleições, petições *on line*, referendos, entre outros.

Nota-se aqui que esta tecnologia pode ser usada por governos para oferecer serviços que atualmente dependem de processos algumas vezes demorados e ainda garantir um registro inviolável. (MOUGAYAR, 2016) lista alguns exemplos que são apresentados a seguir:

- Registro de casamento
- Leilões de contratos
- Emissão de passaportes
- Coleta de benefícios

- Registro de terras
- Licenças
- Certidões de nascimento
- Direito de propriedade
- Registro de veículo
- Patentes
- Impostos
- Votos
- Títulos Públicos
- Arquivamentos e conformidades

As mudanças advindas com o uso do Blockchain podem vir de serviços que já são oferecidos na internet. Entretanto, a segurança é menor e não há garantias do controle da confidencialidade dos dados do usuário. Atualmente, dados captados por redes sociais ou sites de compras ficam a cargo das empresas. Com o uso da referida tecnologia é possível que num futuro tais informações fiquem em controle do usuário. Uma possibilidade adquirida é que ao invés de terceiros comercializarem seus dados, o usuário poderá fazê-lo. O mesmo controle poderá ser exercido sobre propriedades intelectuais, documentos, propriedades e outros.

O Blockchain vem tornar a internet mais democrática, no sentido de oferecer independência de grandes servidores que lucram com a centralização de dados. Imagine que essa tecnologia permite a existência de redes sociais em que o usuário controla quem acessa seu perfil, sites de trocas de valores que lucram com cobrança de taxas por fazer intermédio de operações vão concorrer com redes do mesmo tipo mas que fazem o uso de uma rede Blockchain, ou seja, sem uso de intermediação, já que nessa rede a confiança está embutida pela avaliação da reputação adquirida pelo registro inviolável que ela disponibiliza.

(TAPSCOTT; TAPSCOTT, 2016) simulam como seria uma concorrente da AIRBNB, um site de aluguel de quartos em residências, que utilizaria uma aplicação Blockchain distribuída. Esta nova plataforma seria “um conjunto de contratos inteligentes que armazenam dados em registros internos do blockchain”.

Verificaram a possibilidade de se obter os seguintes benefícios: verificação da reputação, verificação da identidade, proteção de privacidade, redução de riscos, seguros, liquidação de pagamentos, acesso a propriedade utilizando fechaduras inteligentes (aplicação de IoT).

O modelo proposto faz o uso de contratos inteligentes. É cabível dar uma breve explicação do que são contratos inteligentes, já que é uma aplicação comum de Blockchain.

Segundo (TAPSCOTT; TAPSCOTT, 2016):

> Para o propósito desta discussão, os contratos inteligentes são programas de computadores que protegem, fazem cumprir e executam a liquidação de acordos registrados entre pessoas e organizações. Como tal, eles ajudam na negociação desses acordos.

Tais contratos tem como característica o imediatismo e, por estarem associados ao Blockchain, a reputação das partes, conferida a partir do histórico de transações de cada uma, garante o cumprimento do contrato.

A tecnologia deve impactar diversos segmentos, incluindo serviços na internet, transações comerciais e financeiras. Existem projetos, como a ConsenSys, que buscam uma estrutura empresarial onde as definições das obrigações de cada funcionário são atribuídas pela própria rede, utilizando-se para isso as informações das qualificações dos usuários e de inteligência artificial. Dessa forma, eliminaria a figura de um gerente, ou mesmo de um CEO, segundo informações de (TAPSCOTT; TAPSCOTT, 2016). Tal estrutura tornará as dinâmicas de trabalho mais horizontais e cada um teria uma avaliação justa do quanto contribui.

Para finalizar a análise dos impactos do Blockchain, elencarei as possibilidades que tal tecnologia oferece, segundo, (MOUGAYAR, 2016):

- Ativos programáveis
- Confiança programável
- Propriedade programável
- Dinheiro programável
- Identidade programável
- Contratos programáveis

Complementando a definição acima, o Blockchain é uma forma de registro de dados no qual o mesmo é feito em blocos sequenciados, de forma que o bloco seguinte conterá a identificação do bloco anterior. Dessa maneira, qualquer modificação de um bloco causará modificações nos blocos seguintes.

Afinal, alterando as informações de um bloco a identificação *hash* desse bloco será alterada e, como ela é indicada no bloco seguinte e de forma indireta no conjunto de blocos seguintes, os *hash* de todos esses blocos posteriores serão alterados. Por conta dessa arquitetura, a verificação de autenticidade de um Blockchain é imediata.

Outra característica importante do Blockchain é o fato de o armazenamento dos dados serem descentralizados, isto é, eles estão impressos em mais de um servidor, numa rede de computadores. Caso um *craker* realize um ataque em um desses servidores ele não causará danos nos registros, pois, apesar de ele poder modificar os dados contidos nesse servidor, o blockchain irá desconsiderar essa alteração pelo fato de ele considerar como

verdadeiro as informações contidas no maior número de nós. Isso dependerá do modo como o protocolo do blockchain está configurado, dificultando a alteração do registro.

O fato de se utilizar hash aumenta a segurança da rede, pois para realizar a codificação de um dado em um hash precisa-se realizar operações matemáticas. Caso seja feita a alteração de algum dado de um bloco anterior terá que ser feita a recodificação de todos os dados posteriores, afinal cada bloco sempre indica o anterior o que envolverá inúmeros cálculos. Dessa forma a capacidade de processamento é um fato limitante.

Pensando em questões de vulnerabilidade de um Blockchain, imagine que se um grupo desejar realizar alterações em um registro do tipo Blockchain ele deverá ter capacidade de processar todas as alterações de hashs e fazer com que essa mudanças sejam simultâneas em mais da metade dos servidores da rede. Esse fato já expõe a grande dificuldade de realizar fraudes em um Blockchain.

Repare que quanto maior for o número de servidores utilizados como nós na rede e quanto mais anterior for o bloco a ser alterado aumenta-se a dificuldade dos fraudadores obterem êxito em uma tentativa de burlar um Blockchain, pois aumenta-se a capacidade de processamento.

2.2.4 Funcionamento Blockchain

Neste item do capítulo será detalhado como funciona, de uma forma genérica, um Blockchain sob o protocolo de funcionamento Proof of Work (POW).

Primeiramente deve ser entendido a estrutura de um bloco. O Bloco pode conter as seguintes informações: o número do bloco na sequência, *nonce*, o *hash* que resume as informações do bloco, o hash do bloco anterior e o hash gerado após encerrar o bloco.

- Número do Bloco: Refere-se ao número do bloco da sequência da Blockchain ao qual ele pertence

- Nonce : É um número de auxílio para quando for encerrar o bloco. Ele se modifica até se obter um hash que atenda as exigências do protocolo de funcionamento da Blockchain. Não é um campo obrigatório numa Blockchain. Ele é utilizado na blockchain do Bitcoin como auxilio para execução do seu modelo de execução.

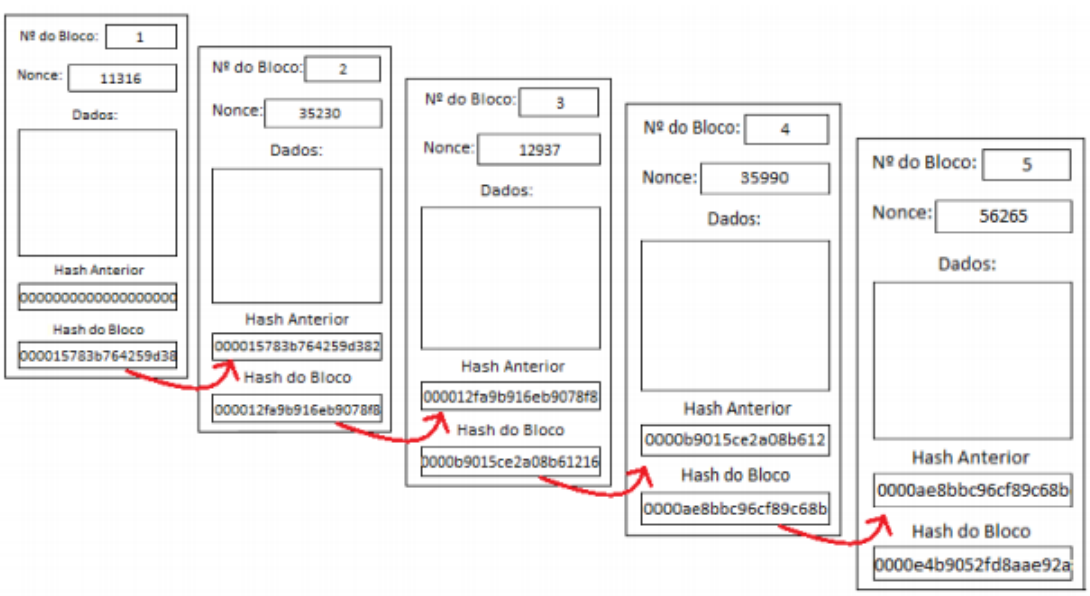
- Hash com as informações do bloco: Refere-se ao que for registrado na Blockchain. ele é obtido a partir da árvore de Merkle. Utiliza-se para esse campo a raiz dessa árvore. No item a seguir será explicado a árvore de Merkle.

- Hash anterior: É um código de tamanho fixo que resume estes mesmos campos do bloco anterior. No primeiro bloco da Blockchain ele indicará uma sequência de zeros.

- Hash do bloco: As informações de todos os campos campos a cima serão reduzidos a um código de tamanho fixo.

O encadeamento desses blocos forma o que chamamos de Blockchain. Confira o exemplo da Figura 1.

Figura 1: Ilustração da geração dos *hash* de informações.



Fonte: autoria própria.

Repare que, por se tratar do primeiro bloco, o campo referente ao bloco anterior está zerado. Ao finalizar os registros no bloco e fazer a mineração do mesmo é gerado um segundo bloco. Note a igualdade entre o *hash prev* do segundo com o hash do primeiro. Isto é feito sucessivamente ao longo de todo os registros dos blocos. Dessa forma assegura-se que a verificação das alterações seja imediata. Entretanto, deve-se pensar o que do Blockchain garante a não mudança dos dados e não só um modo de saber a sua autenticidade.

O segredo está no fato de o Blockchain ser descentralizado, isto é, o registro dos blocos é feito de modo descentralizado, isto é, ele é realizado em diversos servidores. Em um Blockchain é considerada como verdadeira a infomação contida na maior parcela de nós que tiverem informações iguais. Uma mudança em um bloco causa alterações em todos os dados posteriores, por menor que seja, conforme representado na figura abaixo.

Cada Blockchain tem o seu modelo de consenso entre os nós. Esse modelo é o que resolve as divergências entre os nós e permite que se chegue a um consenso sobre a informação a ser tomada como válida.

Existem diversos modelos de consensos aplicados a blockchain, dos quais podemos citar: PoW, PoS, POC e PoeT. O primeiro deles, é o utilizado na blockchain do Biticoin.

Nele, segundo (BALIGA, 2017), cada nó tem que mostrar que tem realizado uma quantidade mínima de trabalho. Sendo que esse trabalho seria encontrar um valor para o hash que seja menor que um determinado valor. O nó que gerar esse hash primeiro finaliza o bloco e ganha uma reconpensa por isso. Esse processo é o que chamamos de mineração. Esse modelo apresenta algumas falhas, entre elas podemos citar o ataque dos "51%", que seria o domínio de mais da metade do número de nós da rede Blockchain. Outra desvantagem é que esse modelo demanda muita energia, devido a enorme quantidade de calculos que os nós tem que executar para a realização do modelo de consenso.

Como consequência dessas desvantagens, novos modelos foram surgindo, como o PoS que, segundo (BALIGA, 2017), parte da ideia de ao invés de se gastar tempo e recursos computacionais, é estabelecido um critério para a eleição com base na posse que que cada nó tem na rede, isto é, a quantidade de recursos que cada um tem na rede. Outro exemplo seria o PoC, em que, segundo (GREVE et al.,) , a chance de um nó finalizar o bloco é baseado na capacidade que ele tem diponível para alocar as informações da Blockchain.

Todos os modelos tem suas vantagens e desvantagens, ao elaborar um projeto de Blockchain, o desenvolvedor deve pensar qual o modelo que melhor atente as necessidades de sua aplicação.

2.2.5 Árvore de Merkle

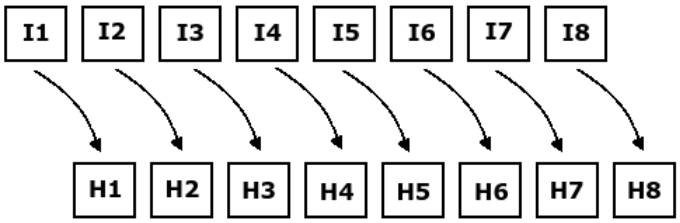
De acordo com (FERREIRA,), a Árvore de Merkle é uma estrutura de dados voltada para permitir a fácil verificação da presença de uma certa informação em um determinado local.

Em uma blockchain, por exemplo a do Bitcoin, esta árvore garante maior segurança a nível de bloco. Pois ela permite averiguar se uma determinada informação corresponde ao bloco que ela indica e se a ordem em que ela foi exposta a Blockchain está realmente na ordem que o verificador, usuário da blockchain, afirma estar.

Para garantir a ordem e a integridade das informações essa estrutura faz o uso dos *hashes* de cada informação e agrupa elas na ordem em que elas são inseridas, em pares, nos quais são expostos novamentes ao algorítimo de *hash*, sendo que esses resultados são novamente agrupado em pares, repetindo o processos de codificação, e assim sucessivamente, até não restar mais pares. Para ilustrar melhor a sua estrutura considere que temos um conjunto de oito informações, ordenadas de acordo com os seus próprios indices como indicado na Figura 2.

A medida que as informações de uma transação vão sendo finalizadas, são geradas o hash das informações de cada uma delas e estes são enviados, a medida que são gerados, para a árvore de Merkle. E a cada par é gerado um novo hash, que é representado, na Figura 3, em um nível acima. E a cada par desse nível é gerados os hash do nível superior

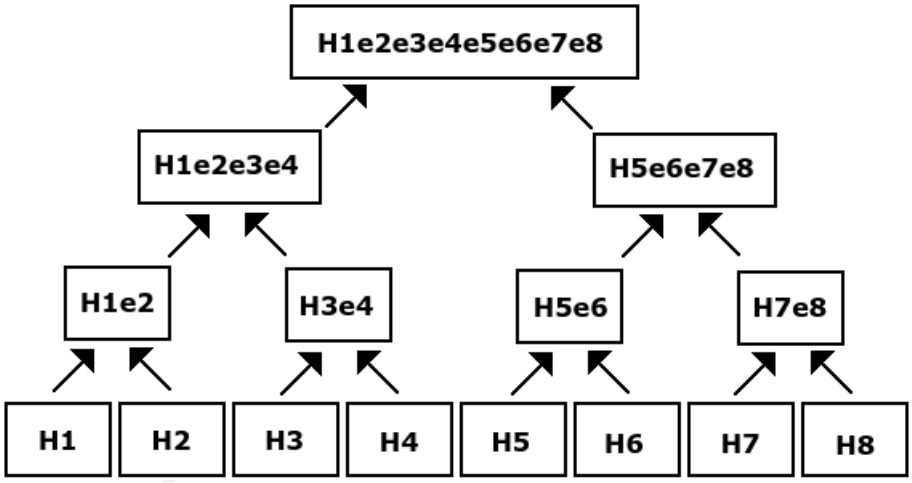
Figura 2: Ilustração da geração dos hash de informações.



Fonte: autoria própria.

e esse processo se repete até não restar mais pares. Assim, para o referido exemplo, no segundo andar notamos quatro *hashes*, no terceiro dois *hashes* e no útimo apenas um *hash*. A medida que novos elementos vão sendo enviados para a árvore os níveis existentes vão sendo remodelados e novos níveis vão ser formados.

Figura 3: Ilustração da estrutura da Árvore de Merkle.



Fonte: autoria própria.

O *hash* presente no último nível é conhecido como *hash* raiz. Esse *hash* é o utilizado na Blockchain para representar um determinado Bloco.

No exemplo, utilizamos um número par de elementos, entretando deve ser entendido qual é o comportamento da árvore de Merkle quando é adicionado um número ímpar de elementos. Quando isso ocorre, é realizado uma cópia desse elemento para que esta seja o par faltante. Note que, após esse processo o nível acima que irá ter número ímpar de elementos. Para resolver, é realizado a cópia desse elemento para gerar o par e isso vai ocorrendo sucessivamente até chegar ao nível com apenas um elemento.

Uma vantagem da utilização da arvore de Merkle para verificar a ordem e a veracidade de uma informação é a velocidade em que isso pode ser feito. Isso se deve ao fato de a taxa de crescimento do número de elementos da base da árvore é muito maior do que o de níveis.

Essa diferença na velocidade pode ser notada na Tabela a seguir. Nela observamos a eficiencia da àrvore de Merkle aplicada ao Biticoin.

Tabela 1: Eficiência do algorítimo da Árvore de Merkle no algorítimo do Bitcoin

Número de transações	Tamanho Aproximado do Bloco	Tamanho do Caminho de Merkle (hashes)	Tamanho do caminho de Merkle (bytes)
16 transações	4 kilobytes	4 hashes	128 bytes
512 transações	128 kilobytes	9 hashes	288 bytes
2048 transações	512 kilobytes	11 hashes	352 bytes
65525 transações	16 megabytes	16 hashes	512 bytes

Fonte: (AGNER,).

3 MATERIAIS E MÉTODOS

Neste trabalho de conclusão de curso será implementado uma Blockchain cujas funcionalidades vão atender as necessidades de um estabelecimento do tipo varejo. No software descrito não haverá a implementação do conceito de assinaturas digitais. Neste capítulo será descrito o desenvolvimento de um algorítimo de Árvore de Merkle, da implementação RFID a nível de protótipo, do banco de dados que surportará a Blockchain, das regras de concordância da rede P2P, a dinâmica e algorítimo relacionado ao funcionamento da Blockchain e a integração das partes para o funcionamento da Blockchain por completo.

3.1 Materiais

O ferramental nescessário para o desenvolvimento pretendido são de dois tipos: *software* e *hardware*. O que comporta a *hardware* foi desenvolvido em circuitos embarcados bem conhecidos no mercado. A implementação a nível de *software* fez uso de bibliotecas difundidas pela comunidade de programação. A seguir será detalhado o material que foi necessário para a implementação da Blockchain.

3.1.1 *Hardware*

- Raspberry Pi

O elemento principal de conexão entre a Blockchain desenvolvida e o módulo leitor RFID é a *Raspberry Pi*. Trata-se de um microcontrolador ARM embarcado em uma placa que contêm Pinos GPIO, Ethernet, HDMI e USB. Considerando as suas funcionalidades, ela é considerada de baixo custo e, nela, pode ser executado um sistema operacional, o Raspbian.

- Kit Módulo leitor RFID MRFC522 Mifare

É um módulo que contém o chip MF-RC522 que permite a leitura da tags RFID sem a necessidade de contato. Ele opera na frequência de 13,56 MHz. A Tecnologia do chip é denominada MIFARE, desta forma ela obedece o padrão ISO/IEC 14443 Tipo A.

- Etiquetas RFID

Para realização do projeto foram utilizados etiquetas RFID de 13,56 MHz para atender as características do leitor mencionado no item anterior.

3.1.2 Software

Para o desenvolvimento deste trabalho foi escolhido a linguagem de programação Python3, por ser uma linguagem de alto nível, orientada a objetos e devido a sua riquesa de bibliotecas e por ter uma comunidade ativa, fato este que torna esta linguagem ideal para desenvolvimento rápido de softwares.

- hashlib

Este módulo implementa diversos algorítimos seguros de hash. Este módulo foi utilizado para a implementação do algorítimo SHA256.

- sqlite3

Este módulo foi necessário para a implementação do banco de dados. Ele foi escolhido entre os inúmeros módulos existentes para esse fim, pois ele já vem instalado junto ao python, é intuitivo e as suas funcionalidades atendem ao que projeto necessita.

- datetime

O módulo datetime possui métodos para a obtenção da data e hora local no instante em que são chamados. Como a Blockchain exige o registro temporal, este módulo torna-se essencial.

- RPI.GPIO

Este módulo é utilizado para o controle dos pino GPIO da Raspbery pi. Neste trabalho ele foi utilizado para a implementação do módulo leitor RFID.

- MFRC522

Este é um módulo que contém os métodos referentes as funcionalidades do leitor RFID MRFC522 Mifare.

3.2 Métodos

Para a realização do projeto, foram realizados os desenvolvimento das partes relevante em separado e ao final, realizar a aplicação conjuta. Dessa forma, nesta seção, serão descrito, primeiramente, o desenvolvimento das partes essenciais da blockchain e da implementação RFID na Raspberry pi e em seguida se discutida a abordagem de integração. As partes relevantes para o projeto, são:

- Implementação RFID
- Árvore de Merkle
- Banco de dados
- Blockchain
- Abstração P2P

3.2.1 Árvore de Merkle

Assim como no Bitcoin, a blockchain desenvolvida neste trabalho também é estruturado por uma Árvore de Merkle. No algoritimo desenvolvido nesse trabalho foi considera que o arranjo das raízes serão organizados em duplas. Assim se faltar elementos para formar uma dupla símétrica com os *hash* da esquerda da árvore, deve ser feita a duplicação de elementos em que há correspondencia, em relação a posição na arvore, de forma a completar a árvore.

Para a construção do algorítimo de merkle que atendesse foram necessário onze, 11, métodos, que são:

Quadro 1: Métodos referentes a Árvore de Merkle

Métodos referentes a Árvore de Merkle
n_níveis_merkle
add_nivel_merkle
n_elementos_merkle
add_elementos_merkle
preenche_merkle
controle_clone_merkle
delimitando_clonagem
atribuindo_clones
clones_niveis_alto
processa_merkle
percorre_merkle

Fonte: Autoria própria

Cada um dos métodos citados tem um papel importante na montagem da árvore de Merkle. A seguir você vai encontrar uma explicação de cada um deles.

- n_níveis_merkle

Neste trabalho é atribuído como nível na arvore de merkle cada etapa de elaboração de *hash*. Dessa forma, o primeiro nível corresponde aos *hashes* gerados a partir dos dados

!h]

Figura 4: Algorítimo do método n_niveis_merkle.

```
def n_níveis_merkle(indice_transação_atual):
    ''' Este método avalia quanto niveis deve
    ter a arvore de Merkle baseado no indice de
    transação atual '''

    if indice_transação_atual <= 2:
        n_niveis = 2
    else:
        n_niveis = 2
        a = indice_transação_atual
        c = 0
        while a != 1:
            b = a % 2
            a = a // 2
            c += b
            n_niveis += 1
        if c == 0:
            n_niveis -= 1

    return n_niveis
```

Fonte: autoria própria.

de uma transação, o segundo degrau é o formado pelos *hashes* gerados tendo como entrada dois *hashes* do primeiro, e assim em diante até chegar ao nível que vai ter apenas um *hash*.

Este método avalia quantos níveis deve ter a arvore de merkle tendo como referência o índice da transação atual.

Nele é admitido como dois degrais quando o número de transações é menor ou igual a dois. Quando o número de degraus é maior do que dois o valor do índice da transação é dividido por dois até o seu valor ficar igual a um, 1, e em cada interação é somado um ao número de níveis. Há os casos especiais, nos quais o indice é igual ao ultimo elemento da árvore naquele instante. Nesses casos deve ser subtraído um do número de níveis. O Algorítimo é apresentado na Figura 4.

- add_nivel_merkle

Este método adiciona novos níveis na árvore de merkle caso tenha a necessidade. Isso vai depender no indice de transação, pois, como o indice de transação representa o tamanho da base da árvore, o acréscimo dele interfere na estrutura da árvore de forma que em alguns casos é necessário aumentar o tamanho da árvore em mais um nível. Esses

casos ocorrem quando o índice da transação fica maior que dois elevado a n, sendo n um número natural.

Para decidir se deve ou não adicionar um novo nível, este método compara o número de níveis exigidos para o indice de transação atual, o resultado do método anterior, com o número de níveis que a Árvore de Merkle possui no momento em que add_nível_merkle é chamado. Caso tenha diferença entre os valores, é adicionado mais um nível. O algorítimo do método é apresentado na figura 5.

Figura 5: Algorítimo do método add_nivel_merkle.

```
def add_nivel_merkle(n_niveis, arvore_merkle):
    ''' Essa função adiciona novos niveis na árvore
    caso seja nescessário '''

    aux1 = len(arvore_merkle)
    aux2 = n_niveis - aux1
    while aux2 != 0:
        arvore_merkle.append([])
        aux2 -= 1
```

Fonte: autoria própria.

- n_elementos_merkle

Considere que elemento seja o número de itens que deve conter na base da Árvore de Merkle considerando o número de transação atual. Não se deve confundir elementos com o indice pois eles não são os mesmo. O número de elemento é maior ou igual indice da transação. Os casos em que ele é maior são os que exigem que sejam realizadas cópias de alguns elementos da árvore para se obter a arvore simérica que a de Merkle exige. O modo como são realizadas essas cópias será explicado em outro método.

Para se obter o número de elementos é feito o uso de uma variável global N, que tem seu valor somado a um sempre que for acrescentado novos elementos. A quantidade de elementos no momento em que este método é chamado é igual a dois elevado a N. Caso o indice de transação seja maior que o número de elementos, é somado um, 1, na variável global N e é calculado novamente o número de elemento elevando dois a N. O Algorítimo descrito pode ser visto na figura 6 a seguir:

- add_elementos_merkle

Este método adiciona novos elementos aos níveis caso seja nescessário. Ele compara se o número de elementos na base da arvore, no momento em que o método foi chamado,

Figura 6: Algorítimo do método n_elementos_merkle.

```
def n_elementos_merkle(indice_transação_atual):
    ''' Essa função informa quantos elementos devem ter no
    primeiro nível da arvore de merkle '''

    global N
    quantidade_de_elementos = 2**N

    if indice_transação_atual > quantidade_de_elementos:
        N += 1
        quantidade_de_elementos = 2**N

    return quantidade_de_elementos
```

Fonte: autoria própria.

é menor que o nescessáro, valor calculado pela função descrita anteriormente. Caso seja menor, ele adiciona novos elementos. Sendo que em cada nível superior da árvore são adicionados menos elementos. A razão de elementos adicionado entre um nível e algum anterior é expresso por uma potência de dois. O código do método exposto na figura 7.

Figura 7: Algorítimo do método add_elementos_merkle.

```
def add_elementos_merkle(n_niveis, quantidade_de_elementos, indice_transação_atual):
    ''' Este método adiciona novos elementos nos níveis caso seja nescessário '''
    n_elementos_atual = len(arvore_merkle[0])
    if n_elementos_atual < quantidade_de_elementos:
        a = quantidade_de_elementos - n_elementos_atual

        b = a
        i = 0
        n = 1
        c = 2**n
        while n_niveis >= 0:
            while a >= 1:
                arvore_merkle[i].append([])
                a -= 1
            a = b/c
            n += 1
            c = 2**n
            i += 1
            n_niveis -= 1
        if N != 1:
            arvore_merkle[n_niveis].append([])
    print()
    print()

    return arvore_merkle
```

Fonte: autoria própria.

- preenche_merkle

Este método preenche a tabela de merkle com um *hash* recebido e caso este seja referente a um índice par, os níveis acimas que tiverem índice par dento do próprio nível serão preenchido com o *hash* gerado a partir de dois nós no nível logo abaixo.

Ao final deste método, a Arvore de Merkle pode estar incompleta, necessitando que sejam feitas cópias de alguns elementos. A geração de cópias será definida em um outro método.

Figura 8: Algorítimo do método preenche_merkle.

```
def preenche_merkle(hash_conteudo, indice_transação_atual, arvore_merkle):
    ''' Este método preenche a arvore de merkle com o elemento recebido '''

    arvore_merkle[0][indice_transação_atual-1] = hash_conteudo
    i = 1
    a = indice_transação_atual
    if a % 2 == 0:
        while a % 2 == 0:
            new_conteudo = str(arvore_merkle[i-1][a-2]) + str(arvore_merkle[i-1][a-1])
            hash_new_conteudo = calcula_hash(new_conteudo)
            a = int(a/2)
            arvore_merkle[i][a - 1] = hash_new_conteudo
            i += 1

    return arvore_merkle
```

Fonte: autoria própria.

- controle_clones_merkle

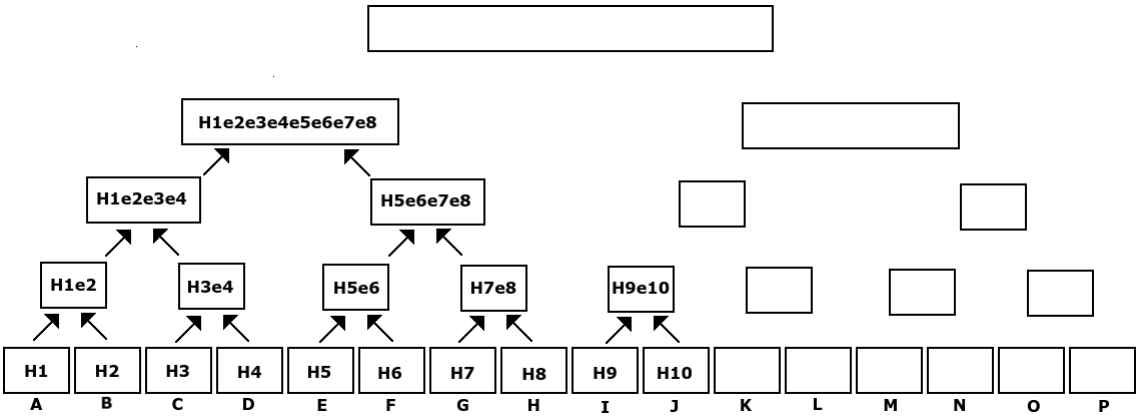
Este método é responsável por estipular a referência que o algorítimo como todo da Árvore de Merkle deve considerar para completar a estrutura com as cópias de alguns dos elementos quando nescessário. Essa refência vai ser utilizado pelos próximos dois métodos.

Este método torna-se necessário pois a referência tem um corportamento dinâmico, isto é, ela muda a medida que o indice da transação for aumentando. E, quando for feita a montagem da árvore com os elementos em aberto, esta função pode ser chamada mais de uma vez devido as aterações necessárias para um mesmo índice de transação.

Este comportamento ocorre, pois o preenchimento da Árvore de Merkle respeita uma simetria em relação a posição. Para ilustrar melhor, vamos tomar como exemplo como estaria a Árvore de Merkle imediatamente depois de ter ocorrido a transação de número dez, 10. Antes de ser realizada a atribuição das cópias, encontraremos a Árvore de Merkle atualizada da seguinte maneira:

Para que a Árvore deste exemplo fique completa, os espaçoes em branco nessa representação, deverá receber uma informação. Como relatado no capítulo de Revisão bibliográfica, quando há um nó da árcore que não forma duplas, o nó anterior é duplicado e é atribuído como a parte faltante. Assim, seguindo o exemplo, o nó logo a cima de

Figura 9: Ilustração da estrutura daÁrvore de Merkle ao receber a transação 10.



Fonte: autoria própria.

"H9e10"não pode ser gerado pois não tem a informação do nó que faria par com ele. Para resolver o problema, deve ser duplicada a informação do nó I em K e a informação de J em L, e assim é gerado o par de H9e10, que possui o mesmo conteúdo que este, e assim o nó superior pode ser gerado. Tal procedimento é executado novamente, de forma que os *hashes* em I, J, K e L são copiados em M, N, O e P, e assim a árvore pode ser completada.

O método controle_clones_merkle dá a informação dos parâmetros para que possa ser estipulado quais elementos vão ser copiados e onde serão atribuidos. Repare que as cópias são feitas em mútiplos de dois, 2. Desse método é extraído o índice de preenchimento relativo que delimita o mútiplo de dois, 2, que os métodos a seguir usam para atribuir as cópias, ou seja, delimita se no ciclo, considere um ciclo como uma passagem por este método, serão copiados um, dois, quatro ou outro múltiplo do número dois elementos. O algoritimo deste método pode ser consultado na figura 10 a seguir:

Figura 10: Algorítimo do método controle_clones_merkle.

```
def controle_clones_merkle(quantidade_de_elementos, indice_preenchimento, arvore_merkle):
    ''' No início da clonagem a variável "indice_preenchimento" corresponde a variável
        "indice_transação_atual" mas ele é um parametro para completar a árvore de merkle
        com os devidos elementos correspondentes a posição de cada clone '''

    if indice_preenchimento < quantidade_de_elementos:
        aux = quantidade_de_elementos - indice_preenchimento
        n =1
        a = 2**n
        while aux >= a:
            a = 2**n
            n +=1
        Total = quantidade_de_elementos - a
        indice_preenchimento_relativo = indice_preenchimento - Total
        delimitando_clonagem(indice_preenchimento, indice_preenchimento_relativo,
                                                    quantidade_de_elementos, arvore_merkle)
```

Fonte: autoria própria.

Repare, no algorítimo da figura 10, que “a” é o parâmetro que representa um bloco de indefinição. Entenda como bloco de indefinição o conjunto de elemento do primeiro nível da Árvore de Merkle que tem os elementos com conteúdo não defidido e os que serão utilizados, vão ser copiados, para preencher esse primeiros.

- delimitando_clonagem

Este método interpreta o número de cópias que serão executados no ciclo. Utilizando o exemplo do item anterior, no qual consideramos que o índice de transação é dez (10), temos que no primeiro ciclo este método interpreta que devem ser realizadas duas cópias, que seriam a cópias de I em J e de K em L, e no segundo ciclo ele entende que devem ser realizadas 4 cópias, que seriam as cópias de I em M, J em N, K em O e L em P.

Neste método, se for entendido que o indece de preenchimento, que no primeiro ciclo corresponde ao índice de transação, o número de cópias é necessáriamente um, já que os nós da Ávore se fecham em pares.

O método está representado na Figura 11

Figura 11: Algorítimo do método delimitando_clonagem.

```
def delimitando_clonagem(indice_preenchimento, indice_preenchimento_relativo, quantidade_de_elementos, arvore_merkle):
    ''' Este método delimita o número de cópias que o método atribuindo_clones deve executar '''
    N = 0
    if (indice_preenchimento % 2) != 0:
        atribuindo_clones(indice_preenchimento, 1, arvore_merkle, quantidade_de_elementos)
    else:
        if indice_preenchimento != quantidade_de_elementos:
            a = indice_preenchimento_relativo
            n = 1
            if a != 2**n:
                #n = 1
                while 2**n < a:
                    n += 1

                if 2**n != a:
                    N = a - 2**(n-1)
                else:
                    N = a
            else:
                N = a
            atribuindo_clones(indice_preenchimento, N, arvore_merkle, quantidade_de_elementos)
```

Fonte: autoria própria.

Repare no algorítimo da figura acima, que N representa o número de cópias e que a expressão “2**(n-1)” representa os elementos vazios no primeiro nível que não receberão cópias no ciclo.

- atribuindo_clones

Como o próprio nome diz, este método é responsável por atribuir as cópias para os elementos vazios. Ele leva em consideração o índice de elementos e o número de cópias, N,

que ele deve executar no ciclo. Assim ele atribui o conteúdo referente ao índice atual ao elemento N posições a frente. Caso N seja maior que um, 1, ele atribui o elemento anterior ao que está a N posições a frente deste e assim por diante, até que esse processo tenha sido executado N vezes. O método pode ser visto na figura 17.

Figura 12: Algorítimo do método atribuindo_clones.

```
def atribuindo_clones(indice_preenchimento, N, arvore_merkle,
                                                    quantidade_de_elementos):
    ''' Este método atribui aos elementos vazios as devidas cópias '''
    valor_retorno = indice_preenchimento + N
    a = N
    contagem = N-1
    while a >= 1:
        arvore_merkle[0][indice_preenchimento -1 + N - contagem] =
                                        arvore_merkle[0][indice_preenchimento - 1 - contagem]
        if (indice_preenchimento + N - contagem)%2 ==0:
            indice = indice_preenchimento + N - contagem
            clones_niveis_altos(indice)
        contagem -= 1
        a -= 1
    controle_clones_merkle(quantidade_de_elementos, valor_retorno, arvore_merkle)
```

Fonte: autoria própria.

Ao final do processo ele chama o método controle_clones_merkle, pois se o valor de retorno, representado no algorítimo como valor_retorno, não for igual a quantidade de elementos, este método vai iniciar um novo ciclo percorrendo os métodos delimitando_clonagem e atribuindo_clones. O valor de retorno corresponde ao último item que recebeu uma cópia em atribuindo_clones. No caso do exemplo utilizado para explicar os métodos anteriores, no final do primeiro ciclo ele coresponderia ao elemento L, assim o seu valor seria doze, 12, e no segundo ciclo ele seria dezesseis, 16, e assim não haveria mais ciclos pois o primeiro nível da Árvore de merkle estaria completo, conforme mostra a figura 13.

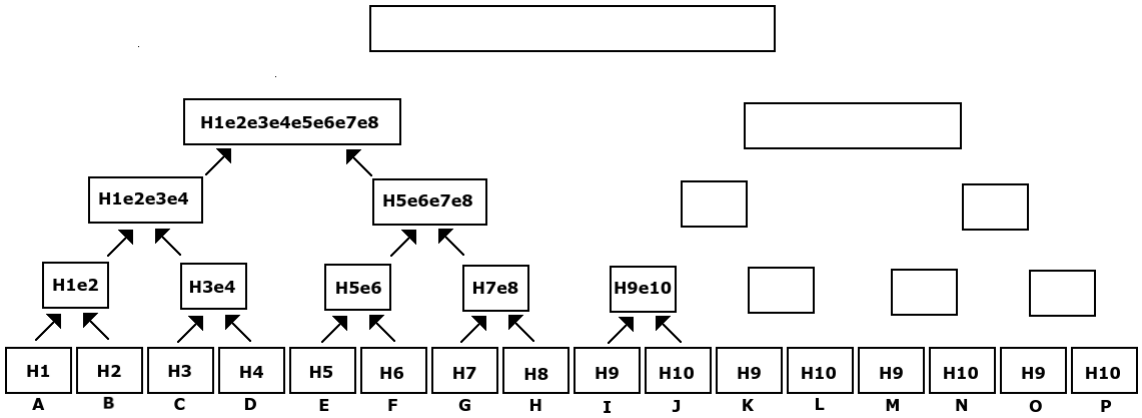
- clones_niveis_altos

Este método, simplesmente , calcula os hash de todas as duplas que foram formadas pelos métodos anteriores e assim ele obtém os nós do segundo nível e vai subindo na árvore, calculando os devidos *hashes*, até o nó raiz. Ou seja, é o método responsável por completar a Árvore de Merkle. Assim, a árvore do exemplo ficará do seguinte aspecto apresentado na figura 14.

O algorítimo do descrito método pode ser conferido na figura 15 a seguir:

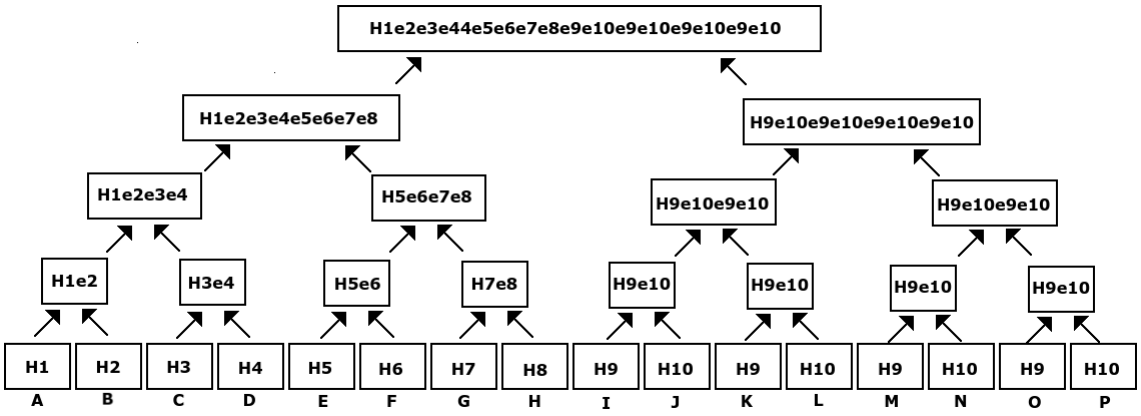
- processa_merkle

Figura 13: Ilustração da Árvore de Merkle no estado de clonagem dos elementos para formar a base.



Fonte: autoria própria.

Figura 14: Estrutura completa da Arvore de Merkle na transação de número 10.



Fonte: autoria própria.

Figura 15: Algorítimo do método clones_níveis_merkle.

```
def clones_niveis_altos(indice):
    ''' Calcula e preenche os campos indefinidos da Árvore de Merkle '''
    a = indice
    i = 1
    while (a % 2) == 0:
        b = int(a/2)
        a = int(a)
        dados = str(arvore_merkle[i-1][a - 2]) + str(arvore_merkle[i-1][a - 1])
        hash_dados = calcula_hash(dados)
        arvore_merkle[i][b-1] = hash_dados
        i += 1
        a = a/2
```

Fonte: autoria própria.

Este método apenas ordena os métodos anteriores para que ocorra a montagem correta da Árvore de Merkle, considerando o índice de transação. O método está exposto na figura 16 a seguir:

Figura 16: Algorítimo do método processa_merkle.

```
def processa_merkle(indice_transação_atual, hash_conteudo, arvore_merkle):
    ''' Esse método sequencia os métodos necessários para completar uma arvore de Merkle com o
        conteúdo "hash_conteudo" fornecido para ele'''

    n_niveis = n_níveis_merkle(indice_transação_atual)
    add_nivel_merkle(n_niveis, arvore_merkle)
    quantidade_de_elementos = n_elementos_merkle(indice_transação_atual)
    A = add_elementos_merkle(n_niveis, quantidade_de_elementos, indice_transação_atual)
    B = preenche_merkle(hash_conteudo, indice_transação_atual, arvore_merkle)
    controle_clones_merkle(quantidade_de_elementos, indice_transação_atual, arvore_merkle)
    print(arvore_merkle)
```

Fonte: autoria própria.

- percorre_merkle

Este método recebe o índice e o *hash* a ser conferido para verificar se o conteúdo que ele registra pertence a aquela posição que o indice indica ou se aquela informação é a mesma que foi registrada no bloco, considerando que a Árvore de Merkle está registrando os dados de um bloco da Blockchain.

O algorítimo apresentado na figura a seguir percorre nível a nivel juntando o *hash* recebido pelo método e os nós nos níveis acima com os pares correspondentes e verifica se há alguma diferença com o que está registrado na Árvore de Merkle. Caso seja identificado uma disparidade, é somado um, 1, na variável "c". Se, ao final do método, c for um valor diferente de zero a função vai alertar que o conteúdo não pertence a árvore ou está na posição errada.

3.2.2 Banco de dados

O banco de dados desta aplicação dá o suporte necessário para que o algorítimo da Blockcain consiga atribuir informações aos parâmetros da blockchain e tenha facilidade para buscar informações específicas. Dessa forma o banco de dados, implementado, é formado pelos campos expostos no Quadro 2.

- id

Trata-se da identificação progressiva das informações da tabela. Ordena as informações a medida que são inseridas no Banco de Dados

Figura 17: Algorítimo do método percorre_merkle.

```
def percorre_merkle(indice_transação_requisitada, hash_para_verificacao, arvore_merkle):
    ''' Este método verifica a integridade ou se a posição esta certa dentro de um bloco '''
    a = len(arvore_merkle)
    i = 0
    b = indice_transação_requisitada
    c = 0
    while a > 0:
        if b % 2 == 0:
            dados = str(arvore_merkle[i][b - 1]) + str(hash_para_verificacao)
            hash_dados = calcula_hash(dados)
            if arvore_merkle[i+1][int(b/2)] != hash_dados:
                c +=1
            b = b / 2
        else:
            dados = str(hash_para_verificacao) + str(arvore_merkle[i][b + 1])
            hash_dados = calcula_hash(dados)
            if arvore_merkle[i+1][int((b+1)/2)] != hash_dados:
                c +=1
            b = (b + 1)/2
    if c == 0:
        print("Dado integro")
    else:
        print("Não pertence ao bloco ou não está no local certo")
```

Fonte: autoria própria.

- data e hora

Essse campos são importantes para o registro do tempo. São campos importantes para a Blockchain pois auxiliam na caracterização de uma transação como única.

- tipotrasacao

Pode ser de dois tipos: “ENTRADA” e “SAÍDA”. O primeiro se refere as transações que se dedicam a acrescentar produtos como ativos para o mercado, isto é, produtos que podem ser comprados por um consumidor. O segunto tipo, se refere aos produtos adquiridos por um cliente, isto é, ativos vendidos pelo mercado. A diferenciação das transações é importante para auxiliar na obtenção da relação de produtos que o mercado tem disponível para venda.

- rfid

É o campo com a identificação única de cada produto. Este campo é completado a partir da leitura de etiqueta RFID

- ntransação e hashtransacao

São campos importantes para completar a Árvore de Merkle, o que facilita na ordenação dos dados do bloco e na verificação de sua autenticidade.

Quadro 2: Campos do banco de dados auxiliar

Campos do banco de dados auxiliar
id
data
hora
tipotransacao
rfid
nome
lote
validade
bloco
ntransacao
hashtransacao
hashblocoanterior
hashbloco
auxhash

Fonte: Autoria própria

- hashbloco e hashblocoanterior

Esses campos são importantes para o algorítimo da blockchain, pois são os responsáveis pela integração dos blocos, o que torna a verificação da integridade rápida e a identificação de falha é instantânea.

- auxhash

Este campo é utilizado apenas para auxiliar na atribuição dos campos ntransação, hashtransacao, hashbloco e hashblocoanterior ao se finalizar a listagem dos produtos que serão adicionados ou retirados do repositório do mercado ou quando o bloco for finalizado.

3.2.2.1 Métodos do banco de dados auxiliar

Os principais métodos relacionado ao Banco de dados estão registrados no Quadro 3.

- conecta e desconecta

Este métodos saõ os responsáveis por conectar, quando for necessário adicionar ou alterar informações, e desconectar o banco de dados.

- estrutura_do_banco

Quadro 3: Métodos do banco de dados auxiliar

Métodos do banco de dados auxiliar
conecta
desconecta
estrutura_do_banco
interface_estoque
interface_caixa
acrescentar_ao_banco_de_dados
busca_RFID
busca_Bloco
busca_hash
fabrica_hash
define_hora
define_data
atribui_hash_transação
atribui_hash_bloco
atribui_numero_transacao
atribui_hash_bloco_anterior
atribui_numero_bloco

Fonte: Autoria própria

É o método responsável por montar a tabela com os campos mencionados anteriormente.

- interface_estoque

É o método responasável por receber leituras do módulo RFID e enviar as informações fornecidas para o banco de dados e atribuir os produtos registrados como produtos disponíveis para a venda. Foram feitas algumas considerações. Cada transação registrada por esse bloco corresponde a um conjunto de etiquetas RFID de um mesmo tipo de produto, sendo que estes vão ser do mesmo lote e terão a mesma data de vencimento. Por se tratar de produos do mesmo tipo, eles podem corresponder há um mesmo cabeçalho. Considera-se como cabeçalho as informações comuns aos produtos do mesmo tipo, confira o Quadro 4 e 5.

Repare, que as informações comuns são atribúidas em cabeçalhos diferentes. Cada um desses cabelhalhos tem as suas informações atribuídas a todos os produtos, a diferenças entre eles é o momento em são atribuídos. O cabeçalho inicial é atribuído no início início do método, no qual o usuário informa as informações deste. Já o segundo cabeçalho é atribuído pelo próprio algorítimo em um outro método, sendo que essas informações são obtidas pelo próprio software no final da transação.

Quadro 4: Cabeçalho inicial

Cabeçalho inicial
tipotransacao
nome
lote
validade

Fonte: Autoria própria

Quadro 5: Cabeçalho final

Cabeçalho final
data
hora
bloco
ntransacao
hashtransacao
hashblocoanterior
hashbloco

Fonte: Autoria própria

Verifique a atribuição do cabeçalho no algorítimo que está exposto na Figura 18.

Figura 18: Algoritimo do método interface_estoque.

```
def interface_estoque(self):
    ''' Método referente a entrada de produtos. '''
    tipotransacao = "ENTRADA"
    nome = str(input("Nome do produto: "))
    lote = str(input("Lote do produto: "))
    validade = str(input("Validade: "))
    data = self.define_data()
    hora = self.define_hora()
    dados = str(tipotransacao) + str(nome) + str(lote) + str(validade) + str(data) + str(hora)
    auxhash = self.fabrica_hash(dados)
    bloco = "a definir"
    ntransacao = "a definir"
    hashtransacao = "a definir"
    hashblocoanterior= "a definir"
    hashbloco = "a definir"
    RFID = 0
    while RFID != "FIM":
        RFID = input("RFID: ")
        #RFID = Método de leitura
        if RFID != "FIM":
            self.acrescentar_ao_banco_de_dados(data, hora, tipotransacao, RFID, nome, lote, validade, bloco,
                                               ntransacao, hashtransacao, hashblocoanterior, hashbloco, auxhash)
```

Fonte: autoria própria.

Note que as variáveis correspondentes as informações do cabeçalho final recebem informações temporárias que devem ser alteradas por um outro método posteriormente

- interface_caixa

Este método é o responsável por receber as leituras do módulo RFID e enviar as informações ao banco de dados, além de atribuir essas leituras como produtos vendidos pelo mercado. Este método também faz uso dos dois tipos de cabeçalho, porém nesse método todas as informações são obtidas pelo próprio *software*. Sendo que no primeiro elas são obtidas após a leitura de uma tag RFID, a partir da chamada de métodos de busca de cada uma das outras informações desse cabeçalho.

As infomações do segundo, da mesma forma que no método anterior, são atribuídas após o término do registro dos códigos RFID a partir de chamadas de métodos específicos de buscas para cada uma das informações do cabeçalho. O algorítimo citado está exposto na Figura 19.

Figura 19: Algorítimo do método interface_caixa.

```
def interface_caixa(self):
    ''' Método referente a saída de produtos '''
    tipotransacao = "SAIDA"
    data = self.define_data()
    hora = self.define_hora()
    dados = str(data) + str(hora) + str(tipotransacao)
    auxhash = self.fabrica_hash(dados)
    Bloco = "a definir"
    ntransacao = "a definir"
    hashtransacao = "a definir"
    hashblocoanterior= "a definir"
    hashbloco = "a definir"

    while RFID != "FIM":
        RFID = input("RFID: ")
        #RFID = método específico do RFID
        if RFIF != "FIM":
            nome, lote, validade = self.busca_RFID(RFID)
            self.acrescentar_ao_banco_de_dados(data, hora, tipotransacao, RFID, nome,
                                               lote, validade, Bloco, ntransacao, hashtransacao,
                                               hashblocoanterior, hashbloco, auxhash)
```

Fonte: autoria própria.

Repare que ocorre a chamada do método de busca pelo código RFID para obtesão do nome, lote e validade do produto que a etiqueta RFID representa.

- acrescentar_ao_banco_de_dados

Este é o método responsável por adicionar as informações vindas dos métodos interfaxe_caixa oi do método interface_estoque no banco de dados da aplicação.

- busca_RFID, busca_Bloco e busca_hash

Estes métodos são utilizados para executarem buscas no banco de dados para procurar por um dado em específico. O primeiro dos métodos, busca_RFID, realiza a

Figura 20: Algorítimo do método acrescenta_ao_*banco_de_dados*.

```
def acrescentar_ao_banco_de_dados(self, data, hora, tipotransacao, rfid, nome, lote, validade, bloco,
                                  ntransacao, hashtransacao, hashblocoanterior, hashbloco, auxhash):
    ''' Este método acrescenta novos produtos ao banco de dados'''
    try:
        cursor = self.conexao.cursor()
        try:
            cursor.execute("""
            INSERT INTO blockchain (data, hora, tipotransacao, rfid, nome, lote, validade, bloco, ntransacao,
                                    hashtransacao, hashblocoanterior, hashbloco, auxhash) VALUES
                                    (?,?,?,?,?,?,?,?,?,?,?,?,?)
            """, (data, hora, tipotransacao, rfid, nome, lote, validade, bloco, ntransacao, hashtransacao,
                                                                    hashblocoanterior, hashbloco, auxhash))
        except sqlite3.IntegrityError:
            print("Há dados repetidos")

        self.conexao.commit()
    except AttributeError:
        pass
```

Fonte: autoria própria.

busca a procura por um determinado código RFID e retorna o nome, lote e a validade do produto que o representa. O segundo retorna o *hash* do bloco quando se inicia a procura a partir deste. O método busca_hash, ao receber o *hash* de um bloco, retorna o bloco que ele resume.

Na Figura 21, você encontrará o algoritimo do método busca_RFID.

Figura 21: Algorítimo do método busca_RFID.

```
def busca_RFID(self, RFID):
    ''' Método para buscar um produto pelo RFID. '''
    try:
        cursor = self.conexao.cursor()
        cursor.execute("""SELECT * FROM blockchain;""")

        for i in cursor.fetchall():
            if i[4] == RFID:
                print("RFID encontrado")
                nome = i[5]
                lote = i[6]
                validade = i[7]

    except AttributeError:
        pass

    return nome, lote, validade
```

Fonte: autoria própria.

- fabrica$_hash$

Este método utiliza do módulo hashlib para utilizar o algorítimo de *hash* SHA256 para obter os *hashes* utilizados por este trabalho.

O método encontara-se exposto na Figura 22.

Figura 22: Algoritimo do método fabrica_hash.

```
def fabrica_hash(self, dados):
    hash_bloco = hashlib.sha256(dados.encode()).hexdigest()
    return hash_bloco
```

Fonte: autoria própria.

- define_hora e define_data

Estes métodos, quando acionados, retornam a data e a hora atual. Estes métodos utilizam do módulo datetime para este fim.

- atribui_hash_transação, atribui_hash_bloco, atribui_numero_transacao, atribui_hash_bloco_ e atribui_numero_bloco

Estes métodos alteram os dados referentes, respectivamente, a hashtransacao, hashbloco, ntransacao, hashblocoanterior, hashbloco. Eles são usados após o uso do método interface_estoque ou interface_caixa para substituir as informações provisórias pelas definitivas que são obtidos apenas quando a transação vai ser finalizada.

O algoritimo de atribui_hash_transacao pode ser vizualizado na Figura 23. Ele é muito semelhante aos outros métodos de atribuição. A diferença está na localização da informação que ele altera.

3.2.3 Implementação RFID

Para realização da leitura das Tags RFID foi utlizado o módulo leitor RFID MRFC522 Mifare e uma *Raspberry Pi* modelo B+. A ligação entre os pinos foi realizado conforme a Tabela 2, conforme indicado na documentação da biblioteca MFRC522.

A comunicação entre o leitor e a Blockchain é realizado por meio do étodo leitura_RFID. Quando chamado ele aguarda a identificação pelo leitor da tag RFID e ao fazer a leitura ele grava ela em uma variável e a retorna para a Blockchain. O agorítimo descrito é encontrado na Figura 24.

Figura 23: Algorítimo do método atribui_hash_transacao.

```
def atribui_hash_transação(self, hashtransacao, auxhash):
    try:
        cursor = self.conexao.cursor()
        cursor.execute("""
        UPDATE blockchain
        SET hashtransacao = ?
        WHERE auxhash = ?
        """, (hashtransacao, auxhash))

    except AttributeError:
        pass
```

Fonte: autoria própria.

Tabela 2: Relação entre os pinos GPIO da Raspberry Pi com os do módulo MRFC522

Raspberry Pi	Módulo leitor RFID MRFC522 Mifare
Pino 24 (GP IO8)	SDA
Pino 23 (CLK)	SCK
Pino 19 (MOSI)	MOSI
Pino 21 (MISO)	MISO
Pino 20 (GND)	GND
Pino 22 (GPIO25)	RDS
Pino 1 (3V3)	3V3

Figura 24: Método para leitura de tags RFID.

```
leitura = True

def leitura_RFID():
    global leitura
    MifareReader = MFRC522.MFRC522()
    lista = []
    A = 0
    while = leitura:
        (status,TagType) = MifareReader.Request(MIFAREReader.PICC_REQIDL)
        if status == MifareReader.MI_OK:
            print("RFID detectado")
        (status, uid) = MifareReader.Anticooll()
        if status == MifareReader.MI_OK():
            A = str(uid[0]) + "," + str(uid[0]) + "," + str(uid[1]) + "," + str(uid[2]) + "," + str(uid[3])
            leitura = False
        return A
```

Fonte: autoria própria.

3.2.3.1 Dinâmica Blockchain

A integração entre todas as partes relatadas anteriormente se deu conforme o algorítimo da Figura 25.

Figura 25: Integração entre as partes da Blockchain.

```
hash_zero = "0000000000000000000000000000000000000000000000000000000000000000"
arvores_de_merkle = []
arvore_merkle = []
N = 1
N_bloco = 1
hash_bloco_anterior = "0"
hash_bloco = "0"

def calcula_hash(dados):
    hash = hashlib.sha256(dados.encode()).hexdigest()
    return hash

if __name__ == "__main__":
    arvore = Arvore_Merkle()
    banco = Blockchain()
    banco.conecta()
    banco.estrutura_do_banco()
    while True:
        N_transacao = 0
        while N_transacao < 1024:
            auxhash = banco.executa_itegração()
            Informacoes = banco.busca_auxhash(auxhash, N_transacao, banco)
            hash_informacoes = calcula_hash(informacoes)
            arvore_merkle = arvore.processa_merkle(N_transacao, hash_informacoes, arvore_merkle)
            N_transacao += 1
        arvores_de_merkle.append(arvore_merkle)
        if N_bloco == 1:
            hash_bloco_anterior = hash_zero
        else:
            hash_bloco_anterior = arvores_de_merkle[N_bloco-2][-1]
        hash_bloco = arvore_merkle[-1]
        banco.busca_Bloco(N_bloco, hash_bloco_anterior, hash_bloco, banco)
        N_bloco +=1
        arvore_merkle = []
```

Fonte: autoria própria.

Repare que atribuímos um valor limite para o número de transações dentro de um bloco. Este valor, 1024, foi escolhido por ser maior que mil, que é um número considerável de transações e completa a estrutura da Árvore de Merkle, isto é, não são encontrados cópias de elementos.

O método chamado executa_integracao leva o usuário a escolher qual a interface que ele quer operar. Ou seja, se ele vai registrar produtos novos para o mercado, e assim atuar como interface de estoque, ou se ele deseja vender produtos e, dessa forma, operar como caixa. Esse método encaminha o usuário para a interface escolhida e registra, de forma organizada, as informações da transação executada. A seguir é gerado o número e o *hash* dessa transação, que é repassado para a árvore de merkle, e, a partir de métodos de atribuição, o *hash* e o número de transação são atribuídos a cada RFID da transação em

campos reservados no banco de dados.

Terminando o ciclo das transações dentro do bloco, o algorítimo aglomera as informações correspondentes ao hash do bloco anterior, hash do bloco e o número do bloco e atribui essas informações aos campos correspondentes no banco de dados, pois o método busca_Bloco chamas os métodos que atribui essas informações aos seus reespectivos campos. Ao final do algorítimo é preparado o cenário para a geração do novo bloco

3.2.3.2 Abstração P2P

Os modelos de consenso do Blockchain, no geral, giram em torno de uma competição entre os nós para finalizar o bloco, afinal, quem o finalizar ganha uma recompensa. Entretanto, pensando no contexto de uma aplicação dentro de um Varejo, não há necessidade de uma competição entre os nós, já que todos estão sobre os cuidados da mesma administração. Ou seja não há vantagens nessa competição.

Dessa forma o modelo de consenso aplicado ao varejo deve levar em conta a gestão eficiente da Blockchain. Dessa forma o protocolo de consenso proposto neste trabalho vai levar em conta o tempo de ociosidade do caixa em questão e o tempo de operação do mesmo. Dessa forma foram criadas as seguintes regras para a escolha do nó que finalizará o bloco:

- Não ter realizado a validação dos blocos anteriores. Esta regra visa dificultar que seja previsto o nó que realizará a validação
- O Bloco que estiver há mais tempo sem validar uma transação irá, a não ser que tenha realizado uma das ultimas operações, validar o próximo bloco que se finalizará
- A quantidade de vezes que um nó ficar sem finalizar um bloco após ter validado o último deve ser aleatório
- Nós que não tiverem realizado um número mínimo de operações no dia, sendo este variável em relação ao período do dia, não poderá realizar a finalização de um bloco.

Tais regras, não são suficientes para assegurar um modelo de consenso entre os nós. Deve ter regras para resolver problemas relacionados a divergência de transações. Adiciona-se a esses termos os a seguir, que foram importados do protocolo de consenso do biticoin, conforme descrito por (MOUGAYAR, 2016):

- Caso ocorra uma divergência do número de blocos entre os nós da Blockchain, mas for conferido que se tratam das mesmas transações, deve ser considerado o que tiver o maior caminho, isto é, o que possuir o maior número de blocos. Pois maior será o número de operações matemáticas realizadas e assim maior será a segurança envolvida.

- O nó que entrar na rede após o início do seu funcionamento, deve aceitar as informações contidas nela como verdadeiras
- O nó que tiver alguma informação que não tenha consenso com a Árvore de Merkle do bloco distribuído deve ser excluída da Blockchain.

4 RESULTADOS E DISCUSSÃO

Para aferir a eficiência da Blockchain elaborada, considerou que o algorítimo Árvore de Merkle é o que expressa a maior sensibilidade para este quesito. Afinal nele é feito todo o processamento das informações dos blocos.

Dessa forma, foi realizado um teste que permitiu verificar a velocidade da formação de uma árvore completa para determinadas quantidades de transações. Optou-se em realizar o teste em relação a um número definido, pois não há precisão necessária para estabelecer a conferência na velocidade de formação para valores seguidos. Realizando o teste para valores específicos já garante uma boa análise do comportamento do algorítimo elaborado.

Figura 26: Algorítimo de teste da Árvore de Merkle.

```
def define_hora():
    hora_atual = datetime.now()
    hora_em_texto = hora_atual.strftime("%H:%M:%S")
    print(hora_em_texto)

AUX = Arvore_Merkle()
arvore_merkle = []
N = 1
i = 1
define_hora()
while i <= QUANTIDADE_DE_TRANSAÇÕES:
    arvore_merkle = AUX.processa_merkle(i, "00000000...0000", arvore_merkle)
    i += 1

define_hora()
```

Fonte: autoria própria.

No teste ilustrado na Figura 26 verifica-se o uso de uma função que define a hora atual. Dessa forma, ao aplicar esta função no início e após a montagem de uma Árvore de Merkle com número de transações definidos, tem-se uma ideia aproximada do tempo que a estrutura demora para ser executada. Na tabela 3 encontam-se os resultados obtidos.

Repare que neste teste não foi verificado diretamente o tempo de retorno, isto é, a resposta de uma arvore de merkle ao que foi solicitado na transação X depois de ocorrer a transação X-1. Entretanto essa relação pode ser obtida a partir desses dados registrados. O que esses resultados expressam diretamente é o quanto o número de transações afeta a taxa de resposta média da Árvore de Merkle.

Considera-se que as aplicações relacionadas ao banco de dados e RFID, em média,

Tabela 3: Eficiência do algorítimo da Árvore de Merkle em relação ao número de transações

Número de transações	Tempo [s]	(Número de transações)/Tempo
1000	1	1000
2000	5	400
3000	13	230,76
4000	17	235,29
5000	35	142,85
6000	52	115,38
7000	61	114,75
8000	67	119,40
9000	103	87,37
10000	150	66,66
15000	258	58,13
20000	252	79,36
25000	870	28,73
30000	1433	20,93
35000	1983	17,65
40000	3047	13,12
45000	3744	12,01
50000	4236	11,80
100000	14765	6,77

não aumentam de volume, em relação ao tempo, a medida que se aumenta o número de transações. dessa forma, o que limita a performace da Blockchain em relação ao número de transações é o algorítimo da Árvore Merkle.

5 CONCLUSÃO

Por se tratar de um estudo que relaciona uma busca contínua do mercado para melhoria dos processos dentro de um varejo, o que é bem visto pelo ponto de vista do lojista, e fazer o uso de tecnologias que estão atualmente em evidência, este trabalho de conclusão de curso se mostra atual e de interesse da sociedade. Neste, considerou-se que com a aplicação das tags RFID os varejistas poderiam conferir tosdos os produtos que estão entrando e saindo do mercado, fato esse que pode ser de extrema importância para os setores de prevenção de perdas, cuja função é ter o maior controle possível sobre os processos para poder reduzir as perdas do estabelecimento.

Neste estudo, foi utilizado um módulo RFID que não permite que seja realizada a leitura em longas distâncias, no entanto o sistema desenvolvido pode operar com leitores conseguem realizar leituras de etiquetas RFID em distancias maiores. Com o tempo, a medida que for reduzido os custos de produção sobre as etiquetas RFID e as antenas, a tecnologia RFID vai ser amplamente difundida no mercado pois tornará os processos mais rápidos e este sistema poderão ser utilizado como uma importante ferramenta para a área de prevenção de perdas.

A partir dos dados averiguados no capítulo de resultados, percebe-se que a medida que se aumenta o número de transações dentro de um bloco , vai se reduzindo a velocidade média das transações. Desse resultado, entende-se que a taxa de resposta do algorítimo de Árvore de Merkle diminui a medida que as transações vão ocorrendo e a Árvore de Merkle vai se expandindo. O que limita a eficiência do algorítimo é o número de transações por bloco da Blockchain. Dessa forma, deve ser estipulado o número de transações ou o tempo de duração do bloco com a finalidade de controlar o efeito de ineficiência gerado pelo trabalho em altas quantidades de transações. Assim, a quantidade de transações estipulada no desenvolvimento deste trabalho, permite a utilização do algorítimo em sua faixa eficiente.

Um ponto importante em relação ao teste é que se forem realizados transações em uma velocidade maior do que a que foi notada em cada um dos casos, haverá obstrução de dados, isto é, alguns dados serão perdidos. De forma que quanto maior o número de transações, mais provável será a ocorrência de obstrução pois a relação entre o número de transações operadas pela Árvore de Merkle em relação ao tempo é menor.

A implementação RFID não foi a quetão de maior dificuldade no trabalho. Entretanto, como futuros passos, o estudo de protocolos de segurança, como os vistos na revisão bibliográfica, são focos importantes para a proteção do patrimônio no mercado. Se for aplicado o algorítimo de leitura da etiqueta RFID como foi estipulado neste trabalho,

os produtos do mercado estarão sujeito a diversos ataques, como negação de serviço, impersonificação, vazamento de informações e rastreamento malicioso.

Em relação a implementação do Blockchain, o algorítimo elaborado representa bem a dinâmica de funcionamento de uma Blockchain. Para a continuação da pesquisa, deve ser realizada o estudo para desenvolver o modelo de consenso mencionado em uma rede distribuída. Isto é como o algorítimo deve se comportar de forma que se tenha a concodância das informações contidas na Blockchain e como adaptar o que foi desenvolvido neste trabalho de conclusão de curso para se adequar a uma blockchain distribuída.

E por fim, deve ser entendido se o estudo da blockchain aplicada ao varejo é uma necessidade. Isto é, as aplicações do blockchain são voltadas em aplicações em que há uma necessidade de se buscar segurança em um ambiente não confiável. Dessa forma deve ser avaliado os processos do mercado nos quais a perda esta envolvida e verificar se são casos que clamem pelo uso de uma tecnologia como a Blockchain.

REFERÊNCIAS

ABRAHAO, E. et al. Transferência de propriedade de etiquetas rfid e utilização de matrizes mds 16 x 16 na cifra de bloco aes. Universidade Católica de Santos, 2007.

AGNER, M. Biticoin para programadores.

ANDRADE, T. M. **A Questão da Automação na Perspectiva do Trabalho como um Direito Fundamental**. [S.l.: s.n.], 2008.

BALIGA, A. Understanding blockchain consensus models. **Persistent**, 2017.

BORTOLINI, S. C.; CONTÍNUOS, I. D. P. Universidade tecnológica federal do paraná–utfpr programa de pós-graduação em tecnologia.

BOTEGA, J. V. L. **Diagnóstico da automação na pecuária leiteira**. 2005. Tese (Doutorado) — Universidade Federal de Lavras., 2005.

DIAS, R. R. d. F. **Internet das Coisas sem mistérios: uma nova inteligência para os negócios**. [S.l.]: São Paulo: Netpress, 2016.

FERREIRA, F. L. Blockchain e ethereum aplicações e vulnerabilidades.

FILHO, J. M. F. **Implementação e análise de desempenho dos protocolos de criptografia neural e Diffie-Hellman em sistemas RFID utilizando uma plataforma embarcada**. 2009. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Norte, 2009.

GREVE, F. et al. Blockchain e a revolução do consenso sob demanda.

MELO, P. R. d. S.; JUNIOR, O. M. Panorama da automação comercial no brasil. Banco Nacional de Desenvolvimento Econômico e Social, 1997.

MORAES, C. C. D.; CASTRUCCI, P. de L. **Engenharia de Automação Industrial .** [S.l.]: Grupo Gen-LTC, 2000.

MOUGAYAR, W. **The business blockchain: promise, practice, and application of the next Internet technology**. [S.l.]: John Wiley & Sons, 2016.

QUINTÃO, H. C. M. et al. Especificação de um sistema multiagente de recomendação de ações em caso de falhas de sistemas de automação e controle. Universidade Federal do Maranhão, 2008.

REIS, J. C. **Teoria & História: tempo histórico, história do pensamento histórico ocidental e pensamento brasileiro**. [S.l.]: Editora FGV, 2015.

ROCKENBACH, S. Arquitetura, automação e sustentabilidade. 2005.

SANTOS, B. M. d. et al. Dilemas da globalização: teoria liberal e ordem jurídica no mundo contemporâneo. **São Paulo: Cultural Paulista. 140p**, 2000.

TANENBAUM, A. S. **Computer Networks, /Andrew S. Tanenbaum, David J. Wetherall**. [S.l.: s.n.], 2011.

TAPSCOTT, D.; TAPSCOTT, A. **Blockchain revolution: how the technology behind bitcoin is changing money, business, and the world**. [S.l.]: Penguin, 2016.

ZUBOFF, S. **Organizalional Dynamics**.