UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

BRUNA SCHRÖDER

RASTREAMENTO DE MEDICAMENTOS COM IDENTIFICAÇÃO POR RADIOFREQUÊNCIA E ARMAZENAMENTO EM *BLOCKCHAIN*

Porto Alegre

2019

BRUNA SCHRÖDER

RASTREAMENTO DE MEDICAMENTOS COM IDENTIFICAÇÃO POR RADIOFREQUÊNCIA E ARMAZENAMENTO EM *BLOCKCHAIN*

Projeto de Diplomação apresentado ao Departamento de Engenharia Elétrica da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para a obtenção do título de Engenheiro Eletricista.

ORIENTADOR: Prof. Dr. Ronaldo Husemann

Porto Alegre

2019

BRUNA SCHRÖDER

RASTREAMENTO DE MEDICAMENTOS COM IDENTIFICAÇÃO POR RADIOFREQUÊNCIA E ARMAZENAMENTO EM *BLOCKCHAIN*

Este Projeto de Diplomação foi analisado e julgado adequado para a obtenção do grau de bacharel em Engenharia Elétrica e aprovado em sua forma final pelo Orientador e pela Banca Examinadora.

_____

Prof. Dr. Ronaldo Husemann

Orientador - UFRGS

_____

Prof. Dr. Roberto Petry Homrich

Chefe do Departamento De Engenharia Elétrica - UFRGS

Aprovado em: ____ de _____ de _____

Banca Examinadora:

_____

Prof. Dr. Marcelo Soares Lubaszewski - UFRGS

_____

Prof. Dr. Raphael Martins Brum - UFRGS

RESUMO

Em 2017 a Agência Nacional de Vigilância Sanitária, Anvisa, determinou que até 2022 todos os medicamentos que circulam no Brasil devem ser rastreados desde o ponto de fabricação ou importação até a venda ao consumidor. Todos os medicamentos devem possuir um Identificador Único de Medicamentos (IUM), gravado na embalagem no formato de um código de barras bidimensional, e os lotes devem conter um número de identificação próprio que permita relação com o IUM dos medicamentos nele contidos. Como não há especificação legal de como os lotes devam ser identificados, propomos neste trabalho a utilização de etiquetas de identificação por radiofrequência (RFID), cujo número de identificação (ID) pode ser lido por leitores presentes em determinados pontos de verificação ao longo da cadeia de movimentação do medicamento. Foi construído um protótipo de leitor RFID conectado ao wi-fi para enviar o ID das etiquetas lidas para uma *blockchain* privada, uma base de dados distribuída que utiliza algoritmos criptográficos para assegurar a integridade das informações ali escritas. Na *blockchain* foram gravadas as datas em que os lotes passaram pelos pontos de verificação, além de informações sobre os lotes, incluindo uma lista de IUM dos medicamentos e o prazo de validade. Testes de *software* e testes de sistemas mostraram que o sistema criado funciona e atende a todos os requisitos.

Palavras-chave: *blockchain*, rastreamento, RFID.

ABSTRACT

In 2017, the Brazilian National Health Surveillance Agency determined that until 2022 every medicine in Brazil must be tracked from source to end consumer. Furthermore, all medicines must have a unique identification number (called IUM) stamped on the packaging as a 2D barcode. Drug lots must also be identified allowing making the link with the IUM of the medication inside. Since there is no legal specification of how the lots must be identified, we propose in this paper the use of Radio Frequency Identification (RFID) tags. Each tag has an identification number (ID) that can be read by RFID readers as the lots pass checkpoints in the supply chain. A prototype RFID reader was built and connected to wi-fi to send the tags IDs to a private blockchain, a distributed ledger where cryptographic algorithms ensure data integrity. In the blockchain, we store the date and time the lot passed by each checkpoint in addition to information about the drugs inside, including the list of IUM and the expiration date. Software tests and system tests showed that the system works and meets all the requirements.

Keywords: blockchain, tracking, RFID.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1. INTRODUÇÃO

As leis federais 11.903 (BRASIL, 2009) e 13.402 (BRASIL, 2016) bem como a Resolução 157 (ANVISA, 2017) estabelecem que a movimentação dos medicamentos deve ser acompanhada desde a fabricação até o consumo pela população, de modo que se possa "traçar o histórico, a custódia atual ou a última destinação conhecida de medicamentos" e que "Cada membro da cadeia de movimentação de medicamentos deverá registrar e comunicar eletronicamente os dados correspondentes às instâncias de eventos ocorridas com o medicamento sob sua custódia" (ANVISA, 2017).

Para traçar a origem e as movimentações dos produtos ao longo de uma cadeia, o sistema de gerenciamento de dados deve ser operado pelos múltiplos parceiros comerciais envolvidos nessa cadeia (RAKIC, 2017). Entretanto, de acordo com Rakic (2017), os sistemas de gerenciamento atuais operam isolados dos outros participantes e não são capazes de prover de forma completa a procedência dos bens. Ambrosus (2018) sinaliza para o potencial da *blockchain* para o aperfeiçoamento da gestão de dados:

> Até agora tem sido difícil reunir dados precisos sobre o estado e a integridade dos materiais e produtos ao longo de toda a cadeia. Mesmo quando os dados estão disponíveis, é um desafio coletá-los, agregá-los, disseminá-los e, acima de tudo, garantir a confiabilidade e a integridade. Novos avanços tecnológicos [...] particularmente na área dos sensores e da Internet das Coisas, em paralelo com as potencialidades da tecnologia *blockchain*, contratos inteligentes e aplicações descentralizadas, permitem-nos repensar como as cadeias produtivas operam. (AMBROSUS, 2018, p.6, tradução nossa).

Blockchain é uma tecnologia que surgiu em 2008 com o Bitcoin, mas cujas aplicações vão muito além das criptomoedas. Na *blockchain* os bens são representados de forma digital, em uma base de dados segura e confiável à qual múltiplos participantes podem ter acesso para escrita e/ou leitura de dados (GREVE, 2018).

Porém, de nada adianta uma base segura se a origem dos dados é duvidosa. Em vista disso, é proposto um sistema de identificação por radiofrequência (RFID) onde um leitor lê etiquetas fixas nos lotes de medicamento, as quais possuem um número de identificação único que é enviado pelo leitor à *blockchain*. Esta, por sua

vez, verifica se a informação veio de uma fonte confiável antes de gravar permanentemente a data e a hora da leitura.

Os dados de rastreamento são armazenados em uma rede descentralizada formada pelos parceiros comerciais que constituem a cadeia de movimentação dos medicamentos. Assim, ao invés de concentrar as informações em um único servidor, todos os membros da rede possuem a sua cópia da *blockchain*, sendo capazes de enviar e ler as informações sobre o contexto dos medicamentos. Isto torna a rede mais robusta, assegurando que as informações estejam sempre disponíveis a todos e dificultando a manipulação dos dados (TANENBAUM, 2011).

A fim de verificar a proposta, foi construído um protótipo de leitor RFID, com uma placa Arduíno Uno em conjunto com um módulo de leitura e um módulo wi-fi, capaz de ler as etiquetas RFID dos lotes e enviar a informação para a *blockchain*. Contratos inteligentes foram desenvolvidos para implementar a lógica da *blockchain*, e aplicativos foram criados para permitir o registro de lotes e a consulta de informações. A validação foi feita com testes de *software* e testes de sistema, e mostrou que o sistema atendia aos requisitos propostos.

2. CONCEITOS FUNDAMENTAIS

Este capítulo traz duas seções, uma sobre criptografia e outra sobre redes, e tem por objetivo facilitar o entendimento deste trabalho pelo leitor que não está familiarizado com os temas.

2.1. Criptografia

2.1.1. Chaves simétricas e assimétricas

Segundo Oliveira (2006), o modelo mais antigo de criptografia é o de chave simétrica, em que a chave - o elemento que dá acesso à mensagem oculta trocada entre duas partes - é igual (simétrica) para ambas as partes e deve permanecer em segredo (privada). Tipicamente, esta chave é representada por uma senha, usada tanto pelo remetente para codificar a mensagem numa ponta, como pelo destinatário para decodificá-la na outra. A principal vantagem deste modelo é a simplicidade e rapidez para executar os processos criptográficos. A desvantagem é que a chave precisa ser compartilhada entre origem e destino antes de se estabelecer o canal criptográfico desejado, e durante o processo de compartilhamento a senha pode ser interceptada (OLIVEIRA, 2006).

Já nos algoritmos de criptografia assimétrica, também chamada de criptografia de chave pública, onde são utilizadas duas chaves distintas, sendo uma pública, que pode ser divulgada, e a outra privada, que deve ser mantida em segredo. Se uma mensagem for cifrada com a chave privada ela somente será decifrada pela chave pública correspondente e vice-versa. A grande vantagem deste sistema é permitir a qualquer um enviar uma mensagem secreta, apenas utilizando a chave pública de quem irá recebê-la. Como a chave pública está amplamente disponível, não há necessidade do envio de chaves como feito no modelo simétrico. A confidencialidade da mensagem é garantida, enquanto a chave privada estiver segura (OLIVEIRA, 2006).

2.1.2. Hashes

Uma função de *hashing* é uma função criptográfica que gera uma saída de tamanho fixo (geralmente 128 a 256 bits) independentemente do tamanho da entrada GTA/UFRJ (2009). A esta saída se denomina de *hash* da mensagem. De acordo com GTA/UFRJ (2009), para ter utilidade criptográfica a função de *hashing* deve ter as seguintes características:

- Unidirecionalidade: conhecido um *hash* h(M), deve ser muito difícil encontrar M a partir de h(M).
- Compressão: a partir de uma mensagem de qualquer tamanho, h(M) deve ter um tamanho fixo. O normal é que o tamanho de h(M) seja menor do que a da mensagem M.
- Facilidade de cálculo: deve ser fácil calcular h(M) a partir de uma mensagem M.
- Difusão: o *hash* h(M) deve ser uma função complexa de todos os bits da mensagem M: se modificado um só bit da mensagem M, o *hash* h(M) deveria mudar a metade dos seus bits aproximadamente.

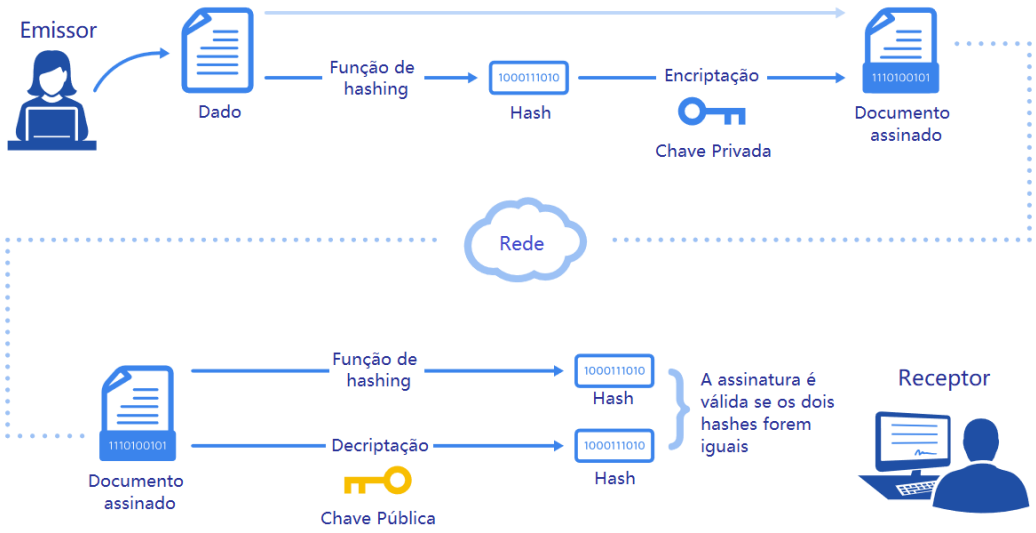
2.1.3. Assinatura Digital

Segundo Maciel (2010), as assinaturas digitais são implementadas através de criptografia de chave assimétrica. A chave privada deve ser mantida secreta e é utilizada para assinar os documentos, enquanto que a chave pública deve ser revelada para qualquer um que queira atestar a autenticidade da assinatura. Para assinar um documento, o autor deve calcular o *hash* da mensagem a ser assinada e, após, criptografá-lo utilizando sua chave privada, gerando assim uma assinatura digital, que é anexada ao documento original e disponibilizada para as partes interessadas (MACIEL, 2010).

O receptor do documento assinado pode facilmente verificar sua autenticidade, bastando encontrar o *hash* da mensagem recebida utilizando o mesmo

algoritmo de *hash* utilizado na assinatura do documento, e em seguida, decriptar a assinatura digital utilizando a chave pública do emissor, obtendo o *hash* original da mensagem. Caso os *hashes* encontrados sejam idênticos, a assinatura é considerada válida para o documento, e a autenticidade e a integridade são garantidas. Na Figura 1 pode-se observar o processo de assinatura e verificação.

Figura 1: Assinatura digital e verificação.



Fonte: Traduzida de Selvamanikkam (2018).

2.2. Redes

2.2.1. Arquitetura cliente-servidor

Redes de computadores são aquelas onde um conjunto de computadores autônomos estão interconectados. A internet não é uma rede, mas um conjunto de redes, a maioria seguindo o modelo cliente-servidor (TANENBAUM, 2011).

Neste modelo, existem os fornecedores de recursos e serviços, que são chamados de servidores, e existem os requerentes dos recursos e serviços, chamados clientes. O cliente não compartilha nenhum de seus recursos com o servidor, mas solicita serviços do servidor. O cliente é o responsável por iniciar a

comunicação através de requisições que são respondidas pelo servidor (TANEMBAUM, 2011).

Tanenbaum (2011) destaca dois problemas inerentes a esta arquitetura. O primeiro diz respeito à sobrecarga: quando o número de requisições dos clientes ultrapassa a capacidade computacional do servidor algumas requisições não serão atendidas. Explorando essa vulnerabilidade, existe um ataque conhecido como DoS (*Denial Of Service*, em inglês) ou ataque de negação de serviço, onde causa-se uma sobrecarga no servidor visando indisponibilizar o serviço para os usuários. O segundo é um problema de falta de robustez: se um servidor falha, seja por motivos físico ou devido a um ataque, as requisições dos clientes ficam desatendidas e dados podem ser perdidos.

2.2.2. Arquitetura par-a-par

Uma rede par-a-par, ou *peer-to-peer* em inglês, surge quando dois ou mais computadores compartilham recursos que não passam por uma entidade central. Esta arquitetura não é hierárquica, todos os participantes são considerados iguais e atuam como fornecedores e consumidores de recurso ao mesmo tempo (TANENBAUM, 2011).

Nesta arquitetura, a informação é replicada em diversos computadores da rede, também chamados de nós. Assim, este modelo é mais resistente à sobrecargas e ataques de negação de serviço quando comparado ao cliente-servidor. Segundo Rocha et al. (2004), grande vantagem deste modelo em relação ao cliente-servidor, é possibilitar a colaboração direta entre os usuários, sem depender de servidores administrados por terceiros. O modelo par-a-par apresenta o benefício da escalabilidade, para tratar de crescimentos incontroláveis no número de usuários e equipamentos conectados, capacidade de rede, aplicações e capacidade de processamento (ROCHA et al, 2004).

Tanenbaum (2011) destaca duas desvantagens desta arquitetura: a dificuldade em garantir a replicação sem adulteração de conteúdo e a dificuldade em rastrear um nó que tenha compartilhado informações falsas ou conteúdo impróprio.

3. RASTREABILIDADE DE MEDICAMENTOS

3.1. Definições

A Resolução 157 define alguns conceitos importantes que serão utilizados ao longo deste trabalho (ANVISA, 2017):

- Cadeia de movimentação de medicamentos: fluxo da origem ao consumo de medicamentos abrangendo as etapas de fabricação, importação, distribuição, transporte, armazenagem e dispensação, bem como os demais tipos de movimentação previstos pelos controles sanitários.
- Membros da cadeia de movimentação de medicamentos: responsáveis pelo registro de instâncias de evento e sua comunicação ao banco de dados centralizado, os quais sejam: fabricantes, importadores, distribuidores, atacadistas, varejistas, hospitais, estabelecimentos de saúde, armazenadores, comerciantes e dispensadores de medicamento.
- Rastreamento de medicamentos: conjunto de mecanismos e procedimentos que permitem traçar o histórico, a custódia atual ou a última destinação conhecida de medicamentos.

3.2. Legislação

A Lei Federal 11.903 (BRASIL, 2009) criou o Sistema Nacional de Controle de Medicamentos (SNCM) com o objetivo de acompanhar os medicamentos em toda a cadeia produtiva, desde a fabricação até o consumo pela população. A Lei Federal nº 13.410 (BRASIL, 2016) altera a lei anterior especificando como o controle deve ser feito.

> Art. 3º O controle será realizado por meio de sistema de identificação individualizado de medicamentos, com o emprego de tecnologias de captura, armazenamento e transmissão eletrônica de dados.
> § 1o As embalagens de todos os medicamentos registrados receberão identificação específica [...] contendo minimamente as seguintes informações:
> I - número de registro do medicamento no órgão de vigilância sanitária federal competente;
> II - número de série único do medicamento;
> III - número do lote ou da partida do medicamento;
> IV - data de validade do medicamento; (BRASIL, 2016)

A legislação diz ainda que "O Sistema Nacional de Controle de Medicamentos deverá contar com banco de dados centralizado em instituição do governo federal, para armazenamento e consulta das movimentações dos medicamentos " (BRASIL, 2016).

A resolução 157 define o Identificador Único de Medicamentos (IUM) como uma "série de caracteres [...] que permita a identificação individualizada, exclusiva e inequívoca de cada embalagem comercial do medicamento" (ANVISA, 2017). A mesma resolução estabelece ainda que as embalagens dos medicamentos devem conter o IUM no formato de um código bidimensional conhecido por Datamatrix. "Toda embalagem de transporte contendo ao menos um medicamento [...] deverá ter um código identificador único próprio, que permita a relação com o IUM dos medicamentos nela contida" (ANVISA, 2017).

3.3. Motivação

Segundo dados do boletim de Saúde e Segurança do Consumidor (SECRETARIA NACIONAL DO CONSUMIDOR, 2015), a quantidade de produtos defeituosos colocados no mercado de consumo em 2014 foi de 161.041 para medicamentos e 158.328 para equipamentos para saúde.

> A importância da lei 11.903/2009 pode ser avaliada pelo problema recorrente da venda de medicamentos falsos no Brasil. Anualmente, um terço de todos os medicamentos vendidos é falso segundo dados de pesquisa realizada pelo Fórum Nacional contra a Pirataria e a Ilegalidade. O sistema também permitirá a rápida identificação e retirada do mercado dos lotes de produtos com algum problema de fabricação, o que é quase impossível atualmente por causa do tamanho do mercado. (MENCK, 2014).

O medicamento falso, ou também conhecido como clandestino, pode causar inúmeros problemas, principalmente às pessoas que estão gravemente enfermas. Sem garantias da procedência ou se são verdadeiros, esses remédios podem não só causar uma piora na doença, como, em casos extremos, até mesmo a morte do paciente (MENK, 2014).

Com o sistema de rastreabilidade é possível realizar o *recall* dos produtos falsos e defeituosos, através do seu número série, junto aos locais comercializados e pacientes destinados, possibilitando assim ações de reversão e a proteção da saúdo de consumidor (MENK, 2014).

3.4. Implementação

Segundo Müller e Leonardi (2017), devido à complexidade do processo, a Anvisa decidiu criar um projeto-piloto de rastreabilidade. No fim de agosto de 2017, a Agência definiu cinco empresas escolhidas para os testes, e posteriormente foram acrescentadas outras duas.

Ainda de acordo com Müller e Leonardi (2017), a Anvisa definiu que projeto-piloto iria até outubro de 2018, após o qual haveria um prazo de um ano para determinar adequações ou fazer alterações necessárias. Então, os laboratórios teriam ainda 36 meses para se adequar às exigências da Anvisa.

Na tentativa de agilizar o processo de implantação da rastreabilidade, foi promulgada a Lei Federal 13.410, dando prazo máximo de cinco anos para que a cadeia farmacêutica implantasse de maneira integral o SNCM (MÜLLER E LEONARDI, 2017).

> Diante do novo prazo e dos apelos do setor, a Anvisa decidiu rever as regras estabelecidas e publicou em 2017 a Resolução 157, determinando novos mecanismos e procedimentos para o rastreamento de medicamentos. De acordo com a decisão da Agência, a implantação do SNCM será dividida em três etapas. Na primeira delas, será feita a serialização da linha de produção, ou seja, as embalagens começarão a ser produzidas com o código bidimensional impresso. Em seguida, será a vez de testar o sistema de troca de informações e o funcionamento do banco de dados da Anvisa. Em um terceiro momento, o sistema de rastreabilidade será implementado de maneira integral, incluindo distribuidores e varejistas. (MÜLLER E LEONARDI, 2017).

Após um ano de testes e participação de mais de 15 empresas, entre indústrias, distribuidoras, farmácias e hospitais, foi concluída em 28 de abril de 2019, a fase experimental de implantação do SNCM (ANVISA, 2019). A legislação prevê mais três anos para a completa implementação desse sistema, período que o setor terá para se adaptar às novas regras (BRASIL, 2016).

4. BLOCKCHAIN

4.1. Visão geral

Blockchain é uma tecnologia que surgiu em 2008 para permitir o funcionamento do Bitcoin, uma rede que permite aos usuários possuir e transferir moedas digitais, também conhecidas como criptomoedas. Por essa razão histórica, a unidade de informação na *blockchain* é chamada de transação, apesar de não necessariamente representar ativos financeiros (GREVE, 2018). Greve (2018), define a *blockchain* como uma máquina de estados, onde as transações são o que desencadeiam as mudanças de estado. O estado atual armazena o histórico de todas as transações já inseridas na *blockchain*.

De acordo com Nakamoto (2008), as transações são agrupadas em blocos que são escritos sequencialmente; cada bloco contém uma referência ao anterior, não sendo possível alterar um dos blocos sem que ele fique incompatível com o resto da cadeia. Assim, para alterar uma transação após sua escrita na *blockchain*, seria necessário não apenas modificar o bloco em que ela está inserida, mas também reescrever todos os blocos subsequentes.

Greve (2018) ainda afirma que a *blockchain* é replicada em uma rede par-a-par, isto é, todos os nós participantes da rede possuem uma cópia do estado atual. Para que um nó participe da rede ele deve estar sincronizado, ou seja, deve possuir o mesmo estado que os demais nós. Se alguém adulterar os blocos acabará ficando com uma versão diferente da *blockchain*, a qual não será aceita pelos demais participantes. Assim, a única maneira de modificar uma informação escrita na *blockchain* seria modificando simultaneamente o estado da maioria dos nós.

4.2. Estrutura dos blocos

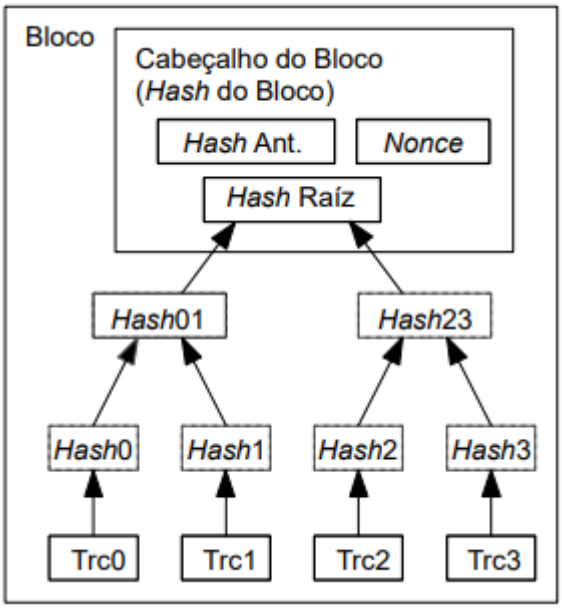
De acordo com Nakamoto (2008), os blocos de uma *blockchain* (como o Bitcoin) são formados por um conjunto de transações e um cabeçalho. Como pode-se observar na Figura 2, no cabeçalho do bloco estão contidos:

● O *hash* raiz: funciona como um resumo criptográfico das transações inseridas no bloco;

● O *hash* do bloco anterior: funciona como uma ligação entre os blocos da cadeia, de modo que se um bloco for alterado seu *hash* vai mudar e ele fica incompatível com o resto da *blockchain*;

● O *nonce*: um número único que identifica os blocos e é crescente ao longo da cadeia.

Figura 2. Estrutura dos blocos do Bitcoin



Fonte: Traduzida de Nakamoto (2008).

Observa-se na Figura 2 que as transações Trc0, …, Trc3 passam por funções de *hashing* e os *hashes* gerados são concatenados aos pares (Hash0+Hash1, Hash2+Hash3), gerando Hash01 e Hash23. Novamente os *hashes* são concatenados aos pares (Hash01+Hash23) e passam por funções de *hashing.* O processo é repetido até que se chegue ao *hash* raiz. Segundo Bergamo Filho (2019), essa estrutura é chamada árvore de Merkle, e é utilizada para resumir e verificar de maneira eficiente a integridade de grandes conjuntos de dados. Se uma pequena modificação fosse feita em uma das transações após a escrita do bloco, seja devido à fraude ou erro, o *hash* raiz mudaria totalmente. Como todos os participantes da rede tem a sua cópia

do estado da *blockchain*, essa mudança no bloco ficaria evidente e não seria aceita pelos demais participantes.

4.3. Tipos de blockchains

Segundo Massessi (2018), as *blockchains* podem ser classificadas quanto às restrições impostas tanto na leitura quanto na escrita dos dados, existindo, então, *blockchains* públicas e privadas.

Blockchain pública:

As características deste tipo de *blockchain* são (MASSESSI, 2018):

- Sem restrições para leitura e escrita de dados: qualquer um pode participar da rede, enviar transações e visualizar as transações que já foram inseridas.
- As informações são replicadas em todos os nós, tornando a rede mais resistente contra adulteração e contra indisponibilidade de serviço.
- Como não há restrições de acesso e todos os participantes são tratados da mesma forma, pode-se dizer que a blockchain pública é democrática e transparente.

Na maioria das *blockchains* públicas, novos blocos são inseridos pelos chamados mineradores (MASSESSI, 2018). Segundo Tschorsch (2016), qualquer nó de uma rede pública pode ser um minerador e competir com os demais em uma tarefa computacional difícil conhecida como Prova de Trabalho. Assim, a habilidade de inserir transações na *blockchain* depende do poder computacional, e não apenas do número de nós na rede, onde a maioria seria mais facilmente obtida em caso de ataque. Além disso, a Prova de Trabalho garante o surgimento de novas criptomoedas, que são pagas como recompensa aos mineradores que resolveram a tarefa primeiro (TSCHORSCH, 2016).

Blockchain privada:

Destaca-se três características deste tipo de *blockchain* (MASSESSI, 2018):

- Há controle sobre quem pode participar da rede, enviar transações e visualizar as informações que foram inseridas.
- Costumam ser utilizadas por empresas que precisam proteger informações confidenciais.
- Maior performance: como todos os participantes são identificados e confiáveis, esse tipo de blockchain permite a utilização de algoritmos de consenso mais rápidos, como a Prova de Autoridade, onde o grupo de validadores é pré-fixado e 2/3 destes devem votar para validar uma transação (ANGELIS, 2017).

4.4. Ethereum

O Bitcoin é limitado a transações com uma criptomoeda, pois foi desenvolvido como um conjunto de scripts que não permite outras regras de negócio (GREVE, 2018). O surgimento do Ethereum, em 2013, introduziu o conceito de contratos inteligentes, códigos onde qualquer regra pode ser implementada, permitindo que a tecnologia *blockchain* fosse utilizada para aplicações muito mais amplas (AVRAMENKO, 2017).

O Ethereum é uma *blockchain* pública, onde os contratos são armazenados nos blocos juntamente com as transações, e possuem um endereço, de modo semelhante às contas de usuários. Para executar métodos definidos em um contrato, o usuário da *blockchain* envia uma transação para o endereço daquele contrato. A lógica definida nos contratos é executada na Máquina Virtual Ethereum, conhecida como EVM, que é executada no computador de todos os mineradores da rede Ethereum (AVRAMENKO 2017).

Cada transação executada na rede *Ethereum* exige o pagamento de uma taxa, conhecida como gás. O gás limita o número de computações executados na

EVM: os blocos possuem um limite de gás, de modo que a soma do gás utilizado pelas transações naquele bloco não pode superar esse limite. Com exceção do limite de gás, a EVM é uma máquina de estados *Turing*-completa, ou seja, capaz de resolver qualquer problema computacionalmente solucionável (AVRAMENKO 2017).

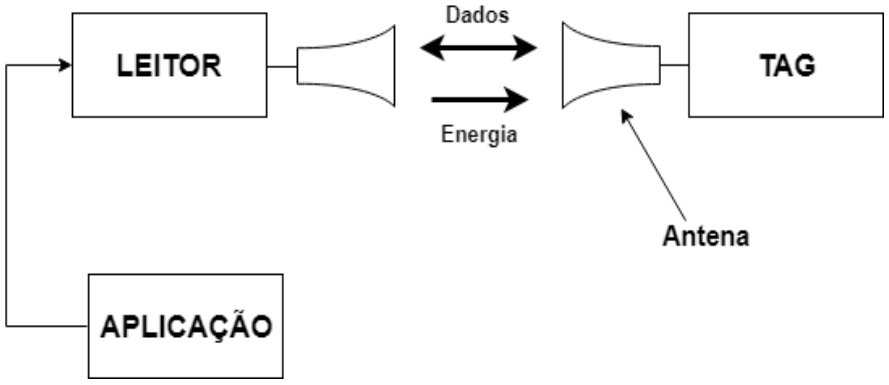
5. IDENTIFICAÇÃO EM RADIOFREQUÊNCIA

5.1. Funcionamento

A tecnologia de identificação em radiofrequência, ou RFID (*Radio Frequency Identification*), é baseada no desenvolvimento do rádio e dos radares entre 1900 e 1940. Nos últimos anos, o RFID vem substituindo o código de barras para rastreamento de produtos em cadeias produtivas (COLBACH, 2008).

Segundo Sousa (2010), a tecnologia RFID fornece um mecanismo para identificação à distância. Os sistemas RFID usam transmissão de rádio para comunicação com entre leitor e etiqueta, também conhecida como *tag*. Os dados recebidos podem ser enviados diretamente a um computador ou armazenados no leitor para envio posterior. Um sistema RFID típico consiste de etiquetas, leitores e aplicação. O sistema de aplicação, também chamado de sistema de processamento, pode ser um aplicativo ou uma base de dados, dependendo da aplicação. Na Figura 3 observa-se um sistema RFID típico.

Figura 3: Sistema RFID



Fonte: adaptado de GTA/UFRJ, 2007.

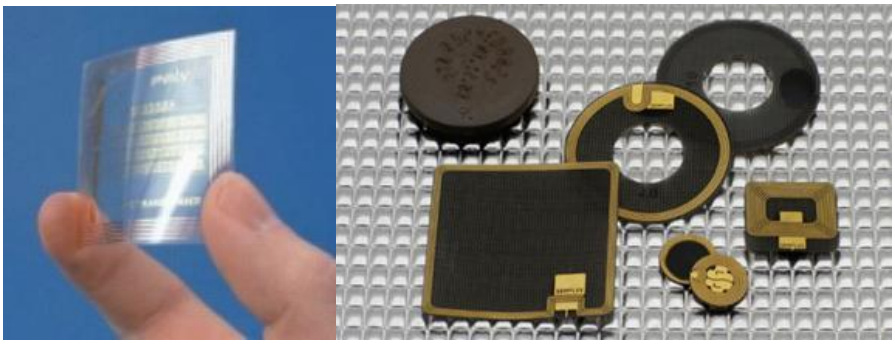
As ondas eletromagnéticas utilizadas na tecnologia RFID usualmente operam em frequências entre as 30 kHz e 5.8 GHz (SOUSA, 2010). As frequências mais baixas são ideais em situações onde a etiqueta precisa ser lida através de materiais que contenham líquidos ou partes metálicas, e a distância da leitura não ultrapassa 10 cm. Com o aumento da frequência, as ondas de rádio passam a ter um comportamento semelhante ao da luz, não passando com facilidade por certos materiais. No entanto, frequências altas permitem leitura em distâncias maiores, maior

taxa de transferência de dados e o custo das etiquetas costuma ser menor (SOUSA, 2010).

5.2. Etiquetas ativas e passivas

Segundo Jia (2012), as etiquetas, também conhecidas como *transponders* (*transmitter*/*responder*) são fixadas em objetos para contagem ou identificação. Etiquetas ativas são alimentadas parcial ou totalmente por baterias, e são capazes de se comunicar com outras etiquetas bem como iniciar a comunicação com o receptor. As etiquetas passivas não possuem fonte interna de energia e são carregadas pelo leitor, enquanto que as semi-ativas - ou semi-passivas - usam uma bateria interna para alimentar o circuito do *microchip*, mas para comunicação, retiram energia do sinal originado no leitor. Na Figura 4 observa-se fotos de etiquetas passiva e ativas.

Figura 4: Etiqueta passiva (esquerda) e etiquetas ativas (direita)



Fonte: GTA/UFRJ, 2007.

Fisicamente uma etiqueta passiva é formada por um *microchip* conectado a uma pequena antena. Tipicamente, este arranjo é montado em um substrato de plástico ou papel, e posteriormente pode ser transformada em etiquetas, cartões ou outro formato que melhor se adapte ao tipo de aplicação final (SOUSA, 2010).

As *tags* ativas possuem, além dos componentes da *tag* passiva, uma fonte de alimentação interna que é usada para ativar os circuitos do *microchip* e para enviar o sinal de resposta à leitora. Esse tipo de *tag* pode ser lido a distâncias maiores e consegue responder a sinais bastante atenuados. Algumas *tags* ativas podem atingir um alcance de centenas de metros (SOUSA, 2010).

Segundo Colbach (2008), a etiqueta é capaz de alterar as características das ondas de rádio refletidas por meio da mudança da impedância entre a antena e o circuito do *microchip*. Com isso, as informações enviadas pela etiqueta como resposta ao leitor são moduladas na onda refletida por chaveamento de amplitude, de frequência ou de fase. Sousa (2010) explica que as antenas são construídas com uma tira de metal que apresenta ressonância mecânica na mesma frequência de operação do leitor.

Abaixo apresentamos uma tabela comparativa entre as etiquetas passivas e ativas, traduzido livremente de Farooq et al (2014).

Tabela 1: Comparação entre tags ativas e passivas.

ATRIBUTO	TAG ATIVA	TAG PASSIVA
Fonte de energia	Bateria interna	Indução eletromagnética
Distância de leitura	Longa (20 a 100m)	Curta (até 5m)
Custo	Alto (dólares)	Baixo (centavos)
Memória	Grande (128kb ou mais)	Pequena (até 128b)
Vida útil	Curta (3 a 8 anos)	Longa (10 anos)

Fonte: Farooq et al (2014).

5.3. Leitores

Os leitores, ou *transceiver* (*transmitter*/*receiver*) têm as funções de ativar as etiquetas, estruturar a sequência de comunicação com as mesmas e transferir dados entre o *software* de aplicação e as etiquetas. Os componentes básicos de um leitor são o módulo de controle e o módulo de radiofrequência, formado pelo receptor e pelo transmissor. Operações de leitura e escrita na etiqueta são executadas usando o princípio mestre-escravo: o leitor assume o papel de mestre e a etiqueta apenas responde aos seus comandos (SOUSA, 2010).

Um leitor de RFID que se comunica com etiquetas passivas deve operar no modo *full-duplex* (transmissão de dados nos dois sentidos simultaneamente), pois

deve transmitir uma onda contínua - para que a etiqueta possa enviar o sinal refletido - enquanto simultaneamente recebe a resposta. Para minimizar a interferência do transmissor no receptor do leitor, antenas distintas podem ser usadas para a transmissão do campo do leitor e para a recepção do sinal da etiqueta (SOUSA, 2010).

Depois de recebido, o sinal emitido pela etiqueta passa do módulo de radiofrequência para o módulo de controle, onde é digitalizado e submetido a um processador digital de sinais para que seja demodulado.

5.4. Comparação com outras tecnologias

Métodos ópticos de identificação, como o código de barras e o QR code, ainda dominam o mercado, principalmente devido ao baixo custo das etiquetas e leitores. No entanto, obstáculos visuais impossibilitam a leitura das etiquetas, e mesmo sujeiras presentes nas etiquetas podem inviabilizar a identificação. Além disso, a quantidade de informação que pode ser armazenada é baixa quando comparada a certas etiquetas RFID, assim como a velocidade de leitura (em torno de 4s) e o alcance (até 50 cm) (SOUSA, 2010).

O custo das etiquetas RFID caiu imensamente nos últimos anos, e essa tecnologia vêm substituindo os códigos de barra. A possibilidade de identificação mesmo que não haja contato visual aliada à alta velocidade de leitura permite escanear ao mesmo tempo dezenas ou centenas de produtos à distância ou dentro de caixas ou caminhões, por exemplo (SOUSA, 2010).

6. METODOLOGIA

6.1. Sistema proposto

Foi desenvolvido um sistema de rastreamento de medicamentos em uma cadeia simplificada composta por três participantes: "Fabricante", "Transportadora" e "Farmácia". Cada participante foi considerado uma entidade única, juridicamente independente das demais. Considerando isto, nesta cadeia constam três pontos de verificação: o primeiro ao final da etapa de fabricação, o segundo onde o lote é recebido pela transportadora e o terceiro onde o lote chega à farmácia. Tais pontos são chamados, nessa ordem, de "Ponto A", "Ponto B" e "Ponto C", como observa-se na Figura 5.

Figura 5: Cadeia com os pontos de leitura

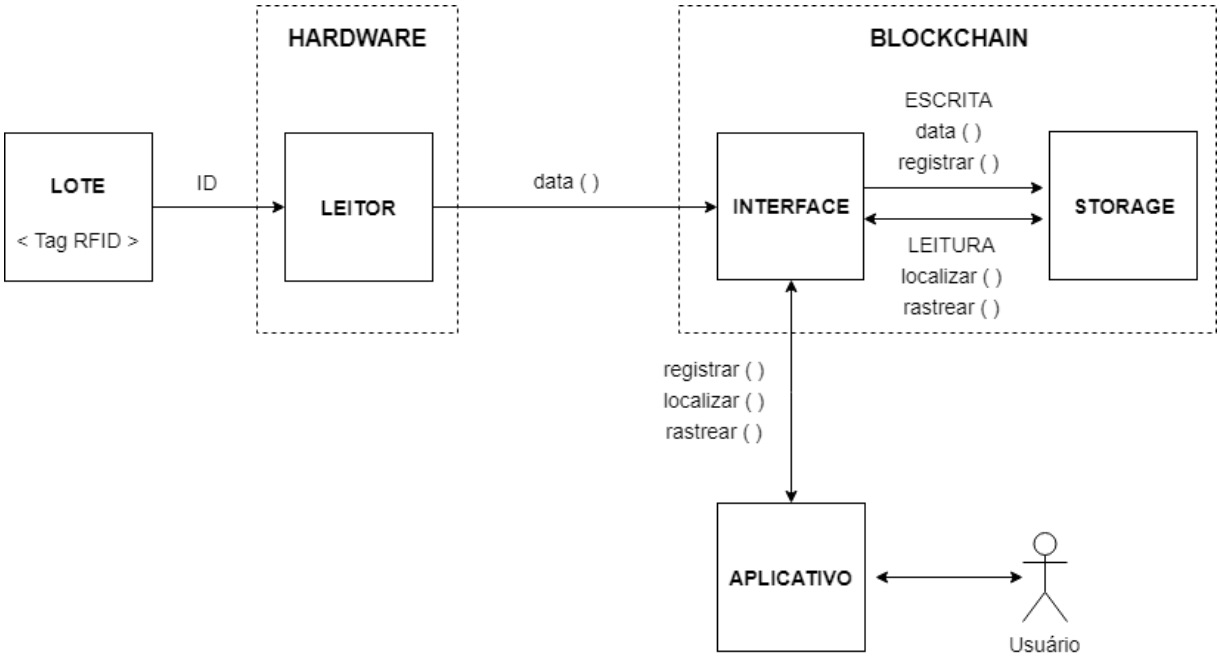
Fonte: A autora (2019).

Uma vez que a legislação não especifica como os lotes devam ser identificados, escolheu-se utilizar a tecnologia RFID, que permite identificação dos lotes com interferência humana mínima, aumentando a confiabilidade das informações. Em cada lote foi afixada uma etiqueta RFID, a qual traz um número de identificação único, doravante chamado apenas de ID. Os medicamentos dentro do lote têm o seu IUM gravado na embalagem em forma de Datamatrix como estabelece a legislação. A relação entre o ID do lote e os IUM dos medicamentos nele contidos fica armazenada na *blockchain*.

O lote deve passar pelos pontos A, B e C, nesta ordem, e em cada ponto há um equipamento capaz de ler etiquetas RFID e enviar dados para a *blockchain*, chamados "Leitor A", "Leitor B" e "Leitor C". Os lotes são lidos individualmente conforme passam pelos leitores. Na *blockchain* ficam registradas as datas de passagem por cada um dos pontos, permitindo traçar o histórico da movimentação, como determina a legislação.

Na Figura 6 observa-se o diagrama do sistema construído. As etiquetas são lidas pelo leitor e o ID é enviado para a *blockchain*, que contém dois contratos: "Interface" e "Storage". O usuário será capaz de interagir com a *blockchain* realizando consultas e registrando lotes a partir do aplicativo.

Figura 6: Diagrama do sistema desenvolvido



Fonte: A autora (2019).

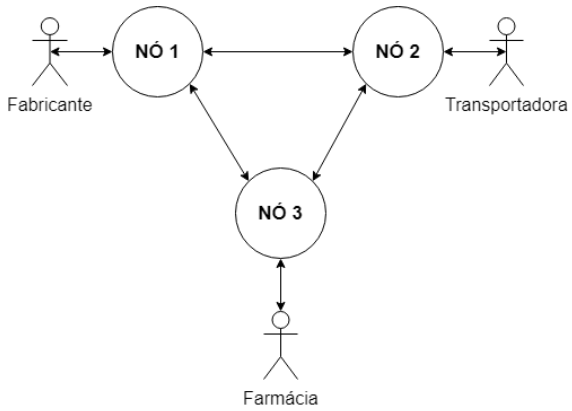
Os dados são armazenados na *blockchain* como objetos chamados "lote", cada um contendo as informações listadas a seguir. Os itens 1, 5 e 6 foram incluídos para obedecer à Lei Federal 13.410 e a Resolução 157 da Anvisa. Os itens 7, 8 e 9 visam identificar os participantes da cadeia e os itens 10, 11 e 12 registram a movimentação do medicamento ao longo da cadeia. Os demais itens foram acrescentados como informação adicional.

1. ID do lote: número da etiqueta RFID que identifica o lote;
2. Medicamento: nome do medicamento;
3. Fabricante: nome do fabricante dos medicamentos;
4. Data de fabricação: data em que o medicamento foi fabricado;
5. Prazo de validade: data em que o medicamento expira;
6. Conteúdo: lista dos IUM de cada medicamento contido no lote;
7. CNPJA: CNPJ do "Fabricante";
8. CNPJB: CNPJ da "Transportadora";
9. CNPJC: CNPJ da "Farmácia";
10. DataA: data e hora da passagem do lote pelo "Ponto A";
11. DataB: data e hora da passagem do lote pelo "Ponto B";
12. DataC: data e hora da passagem do lote pelo "Ponto C".

6.2. Rede

Somente os participantes da cadeia podem enviar e ler dados na *blockchain*, uma vez que a divulgação da posição dos medicamentos pode levar a roubo de carga. Por este motivo escolheu-se utilizar uma *blockchain* privada, constituindo uma rede formada por três nós, um para cada participante da cadeia, como observa-se na Figura 7.

Figura 7: Rede com os três nós



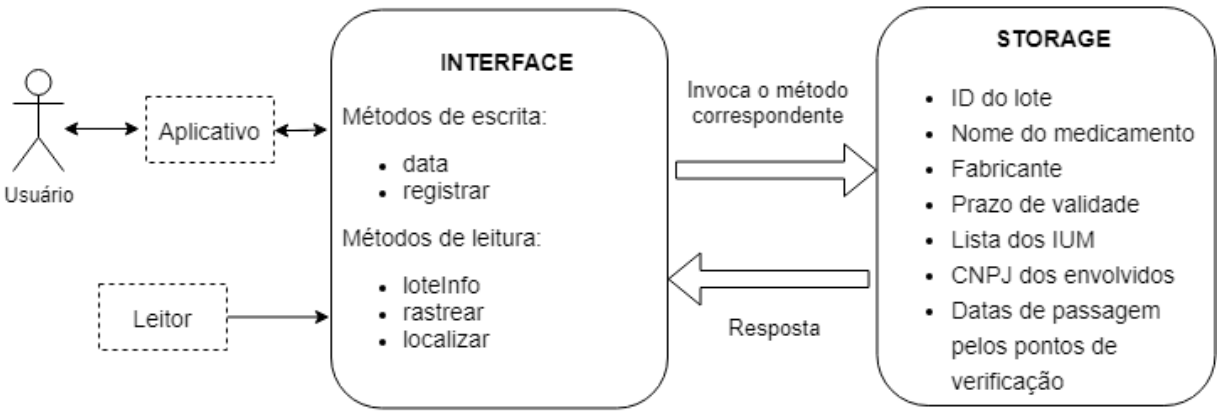
Fonte: A autora (2019)

Para construir a rede foi utilizado o Parity, um *software* que permite aos usuários tornarem-se nós da *blockchain* pública Ethereum, mas que também pode ser utilizado para criar uma *blockchain* privada utilizando a Máquina Virtual Ethereum (PARITY INC, 2019). Em uma rede privada com mecanismo de validação do tipo Prova de Autoridade, 2/3 dos nós validadores deve validar uma transação para que esta seja incluída em um bloco. Os três nós criados são validadores, assim, quando um nó da rede envia uma transação, apenas mais um nó é necessário para validá-la. A validação é feita automaticamente pelos nós, e consiste em verificar que a transação cumpra um conjunto de requisitos técnicos (PARITY INC, 2019).

6.3 Blockchain

Foram criados dois contratos inteligentes: "Interface" e "Storage". O contrato de "Interface" recebe os dados do leitor e possui métodos que podem ser invocados pelo usuário para registrar lotes e realizar consultas. Todos os dados ficam armazenados no contrato "Storage", que, por questões de segurança, não é exposto a agentes externos. Assim, seus métodos só podem ser invocados pelo contrato "Interface", que antes deve verificar a autenticidade e a competência de quem está enviando os dados ou requisitando consulta. Fala-se aqui em competência pois os métodos são restritos e só podem ser invocados por determinados agentes. Na Figura 8 observa-se o diagrama dos contratos.

Figura 8: Contratos



Fonte: A autora (2019).

O método "registrar" do contrato "Interface" só pode ser invocado pelo "Fabricante" e recebe como parâmetros as informações do lote conforme citado na seção 6.1, exceto as datas de leitura. Os parâmetros são passados para o contrato "Storage" que persiste os dados indexando-os pelo ID do lote.

Existem três métodos de consulta: "infoLote", "rastrear" e "localizar". Os três recebem como parâmetro o ID do lote e podem ser invocados pelo "Fabricante", pela "Transportadora" e pela "Farmácia". O método "infoLote" retorna os parâmetros passados no registro do lote, exceto o ID. O método "rastrear" retorna a "DataA", "DataB" e "DataC". O método "localizar" retorna a última posição conhecida do lote ("Ponto A", "Ponto B" ou "Ponto C") e a data de passagem do lote por aquele ponto.

O método "data" recebe o ID do lote lido pelo leitor. Se o nó do "Fabricante" tiver enviado o ID, a data do recebimento da informação pela *blockchain* é registrada como "DataA". A mesma lógica segue para os nós da "Transportadora" e "Farmácia", onde as datas são registradas como "DataB" e "DataC", respectivamente. Caso um nó desconhecido tenha invocado o método, este retornará sem registrar data. No contrato de "Storage" há ainda três verificações:

- Não há registro prévio daquela data, ou seja, não é possível registrar a mesma data duas vezes;
- Para o registro da "DataB", a "DataA" já deve ter sido registrada, enquanto que para o registro da "DataC", a "DataB" já deve ter sido registrada, garantindo que as datas são registradas na mesma sequência dos pontos de verificação;
- A "DataB" só é registrada se for maior que a "DataA", e a "DataC" só é registrada se for maior que a "DataB", garantindo que o lote passou primeiro pelo "Ponto A", depois pelo "Ponto B" e só então pelo "Ponto C".

6.4. Hardware

As informações sobre os medicamentos são registradas na *blockchain* e as etiquetas contêm apenas um número de identificação, não necessitando ter grande

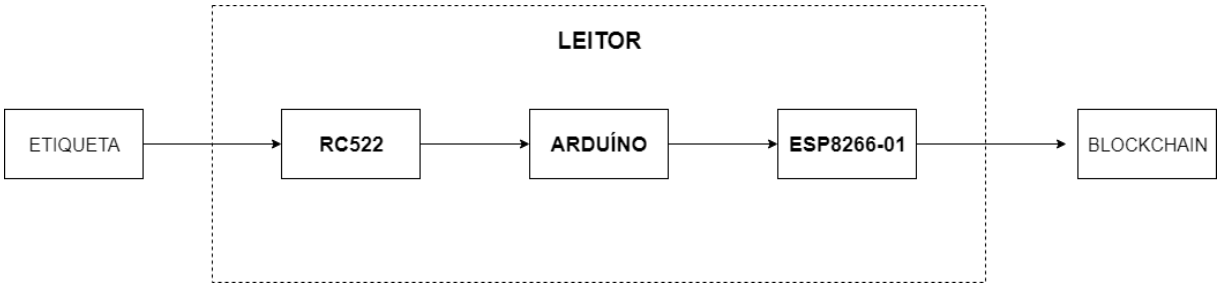
capacidade de memória. As etiquetas são fixadas na parte externa dos lotes de medicamentos e a leitura é feita em curta distância.

Em vista disto, optou-se por utilizar etiquetas passivas que, como visto no capítulo 5 possuem menor memória e menor distância de leitura, no entanto são mais baratas e têm vida útil maior. Como não há metal entre a etiqueta e o leitor, pode-se utilizar etiquetas de alta frequência, que costumam ter custo menor, como mencionado no capítulo 5.

Dentre as opções disponíveis no mercado, escolheu-se a etiqueta NTAG213 da NXP com formato circular, diâmetro de 25mm e superfície adesiva para facilitar a fixação. A etiqueta opera com frequência de 13,56 MHz, a uma distância máxima de 100mm e cuja interface de radiofrequência segue o padrão ISO/IEC 14443. O padrão ISO/IEC 14443 de comunicação de campo próximo define os requisitos e os protocolos de comunicação para cartões de proximidade comumente usados para fins de identificação (NXP, 2015).

Para construção do leitor utilizou-se o módulo RC522 baseado no chip MFRC522 da NXP, capaz de ler etiquetas NTAG213. Uma placa Arduíno Uno recebe os dados do RC522 e os envia para a *blockchain* através do módulo wi-fi ESP8266 ESP-01. Apenas um protótipo foi construído, mas o *software* embarcado no Arduíno possui três versões diferentes, permitindo emular os três leitores da cadeia. Na Figura 9 observa-se o diagrama do leitor e em seguida há o detalhamento dos módulos.

Figura 9: Diagrama do leitor



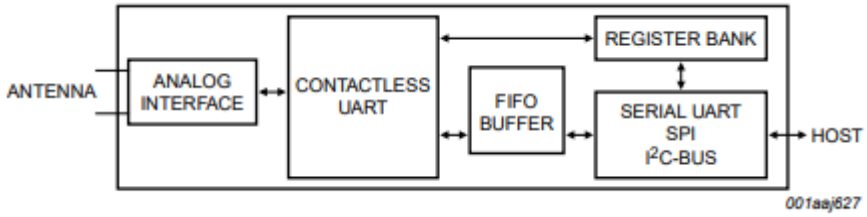
Fonte: A autora (2019).

6.4.1. O módulo RC522

Segundo o fabricante, o módulo RC522 é baseado no *chip* MFRC522 fabricado pela NXP. O *chip* suporta leitura e escrita de etiquetas RFID que seguem o padrão ISO/IEC 14443, a frequência de operação é de 13,56 Mhz e a distância de operação típica é de até 50 mm (NXP, 2016).

Como observa-se no diagrama de blocos da Figura 10, o MFRC522 possui uma antena e um bloco que faz a interface analógica para recebimento e envio de sinais. As interfaces digitais disponíveis permitem comunicação serial com o *host* de forma síncrona - utilizando SPI (*serial peripheral interface*) ou I2C (*inter-integrated circuit*) - ou assíncrona utilizando UART (*universal asynchronous receiver-transmitter*). Os dados de entrada e de saída são bufferizados em um *buffer* FIFO (*first in first out*) de 64 bytes (NXP, 2016).

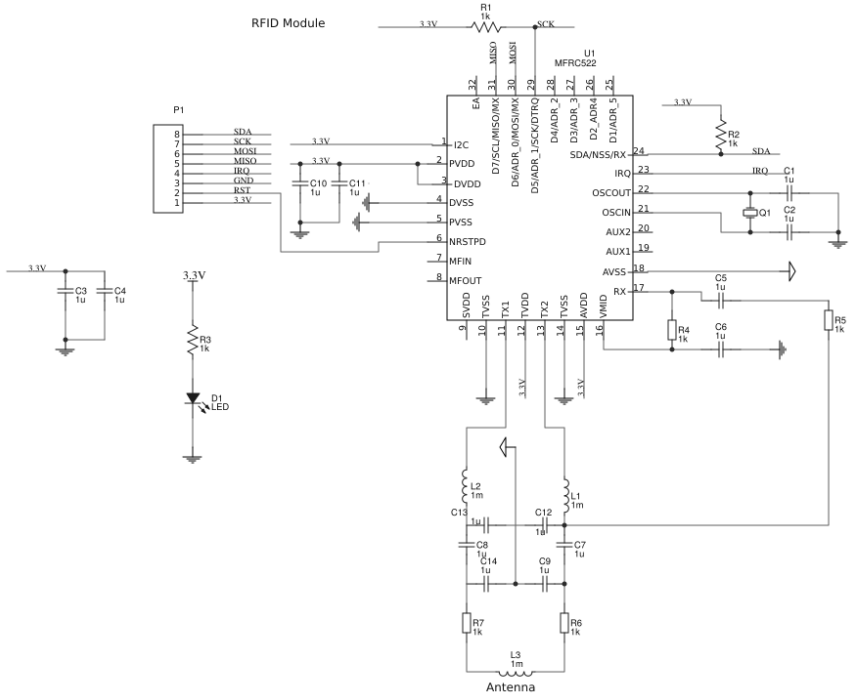
Figura 10: Diagrama de Blocos MFRC522



Fonte: NXP, 2016.

Na Figura 11 observa-se que o módulo RC522 utiliza componentes passivos ligados ao chip MFRC522 para acoplar uma antena. A comunicação com o microcontrolador é feita por SPI, com taxa de transmissão de 10 Mbits/s (NXP, 2016).

Figura 11: Circuito do módulo RC522



Fonte: EasyEDA, 2016.

6.4.2. O módulo ESP8266-01

O módulo ESP8266-ESP01 suporta conexão com a rede Wi-Fi, possui alcance de até 90m e suporta comunicação serial com um microcontrolador através dos pinos RX (recepção) e TX (transmissão). O chip deve ser alimentado com tensão de 3.3V e consome tipicamente 170mA de corrente para transmissão (ESPRESSIF, 2019).

O módulo ainda possui duas portas GPIO (*general purpose input/output*). O pino de *Chip Enable* (CH_PD) quando ligado em nível alto faz o chip funcionar normalmente, e quando ligado em nível baixo faz o chip entrar em modo de baixo consumo. O pino RST reseta o chip quando ligado em nível baixo (ESPRESSIF, 2019).

6.5. Validação

Para validação dos resultados foram realizados testes unitários de *software* e testes de sistema.

> O teste é destinado a mostrar que um programa faz o que é proposto a fazer e para descobrir os defeitos do programa antes do uso. Quando se testa o software, o programa é executado usando dados fictícios. Os resultados do teste são verificados à procura de erros, anomalias ou informações sobre os atributos não funcionais do programa. (SOMMERVILLE, 2007).

Foram escritos diversos testes unitários para mostrar que os contratos desenvolvidos cumprem os requisitos mencionados neste capítulo. Testes unitários são aqueles em que as unidades individuais de programa, neste caso os métodos, são testados individualmente. Apesar de ser impossível provar que um *software* está absolutamente correto por meio de testes, a sua utilização fornece evidências da conformidade com as funcionalidades especificadas (SOMMERVILLE, 2007).

Com o teste de sistema, procura-se verificar se todos os elementos do sistema - e.g. *hardware*, bases de dados, interfaces com usuário - combinam adequadamente e se a função/desempenho global do sistema e alcançada (SOMMERVILLE, 2007). Para execução dos testes de sistema, lotes fictícios foram registrados na *blockchain* através da aplicação do "Fabricante", tags com o mesmo ID dos lotes registrados foram lidas pelo protótipo construído e, em seguida, foram realizadas consultas aos lotes através das aplicações para verificar que os dados armazenados na blockchain são retornados corretamente.

7. IMPLEMENTAÇÃO

7.1. Hardware

Como discutido no capítulo de metodologia, o *hardware* é composto por dois módulos ligados ao Arduíno: o módulo RC522 para leitura das etiquetas RFID e o módulo ESP8266-01 para envio dos dados à *blockchain* por wi-fi. A conexão do módulo RC522 com o Arduíno foi feita segundo a Tabela 2.

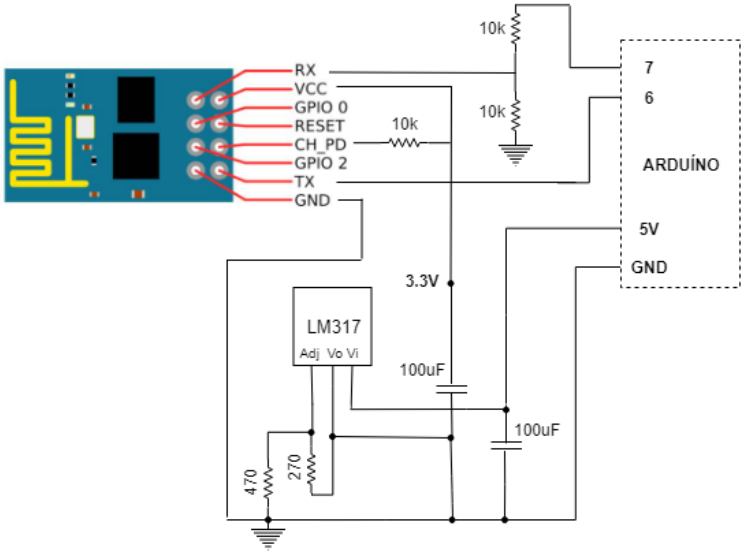
Tabela 2: Conexão RC522-Arduíno

RC522	ARDUÍNO
1- SDA	D10
2 - SCK	D13
3 - MOSI	D11
4 - MISO	D12
5 - IRQ	-
6 - GND	GND
7 - RST	9
8 - 3.3V	3.3V

Fonte: A autora (2019).

O módulo ESP8266-01 deve ser alimentado em 3.3V e consome corrente maior que a oferecida pelo pino 3.3V do Arduíno. Em vista disto, foi montado um circuito de alimentação ligado ao pino de 5V do Arduíno e utilizando o regulador de tensão LM317. A tensão de saída Vs do LM317 pode ser aproximada pela expressão Vs = 1.25 * (1 + R2/R1) (TEXAS INSTRUMENTS, 2016). Portanto, foram utilizados resistores de 470 e 270 ohms, obtendo uma saída de 3,4V. Na Figura 12 observa-se o circuito montado para a ligação do módulo ao Arduíno.

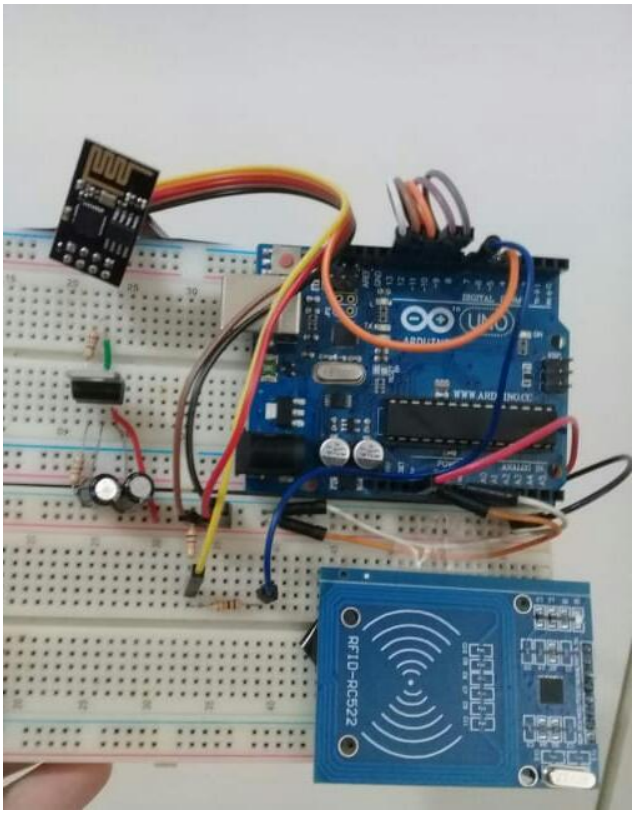
Figura 12: Ligação entre o ESP8266-01 e o Arduíno



Fonte: A autora (2019).

Na Figura 13 observa-se a foto do *hardware* montado na *protoboard*.

Figura 13: Foto do hardware construído



Fonte: A autora (2019).

7.2. Software

7.2.1. Rede

Para criação da rede foi necessário utilizar um cliente Ethereum, um programa que implementa os protocolos para interação com a rede Ethereum, permitindo conectar nós, validar e executar transações (IYER e DANNEN, 2018). O cliente escolhido foi o Parity (versão 2.5.5).

Na Figura 14 observa-se os três nós sendo executados no mesmo computador, com processador i5-8400, 8GB de memória RAM e sistema operacional Ubuntu 18.04. Observa-se em "2/25 peers" que cada nó está conectado a dois outros nós, ou *peers*, num total de 25 conexões permitidas pelo Parity.

Figura 14: Três nós sendo executados

Fonte: A autora (2019).

Cada nó possui seu arquivo de configuração, que, entre outras coisas, define que aquele nó deve conectar-se automaticamente com uma lista de nós específicos sempre que estes estejam disponíveis, e não deve aceitar conexões com nenhum outro nó. A lista de nós para conexão contém o endereço dos três nós criados.

7.2.2. Contratos

Os contratos "Interface" e "Storage" foram implementados em linguagem Solidity (versão 0.5.0), a principal linguagem de programação para Ethereum (IYER e DANNEN, 2018). Os códigos foram escritos no Visual Studio Code, um editor de texto com suporte para Solidity. Para compilação e publicação (*deploy*) dos contratos utilizou-se o Truffle (versão 5.0.26), um popular framework de desenvolvimento para Ethereum (IYER e DANNEN, 2018).

Durante o desenvolvimento dos contratos foi utilizada a metodologia de desenvolvimento orientado a testes. Testes unitários de *software* foram desenvolvidos antes de qualquer código ser escrito. Seguindo a metodologia, métodos eram criados para que os testes passassem e em seguida o código era refatorado e os testes eram executados novamente. O processo se repete até que o código passe em todos os testes. O objetivo da metodologia é integrar o desenvolvimento do código e dos testes, facilitando a identificação e alcance dos requisitos (KENT, 2002).

7.2.3. Aplicativo

Foram criados três aplicativos muito semelhantes, uma para cada participante da cadeia, utilizando linguagem Javascript com a biblioteca Web3, que é a biblioteca padrão para comunicação com o Ethereum (IYER e DANNEN, 2018). O aplicativo de um determinado participante conecta-se ao nó daquele participante para acessar métodos do contrato “Interface” na *blockchain*.

Todos os aplicativos possuem a opção "Pesquisar Lote", que ao ser selecionada lista três opções: "Informações do lote", "Localizar lote" e "Rastrear lote". Essas opções invocam na *blockchain* os métodos "infoLote", "localizar" e "rastrear", respectivamente.

Na Figura 15 observa-se o aplicativo sendo utilizado para uma consulta de "Informações do lote", onde é passado o ID de um lote previamente registrado na *blockchain* e como resultado mostra na tela o ID do lote, nome do medicamento, nome

do fabricante, data de fabricação, prazo de validade, conteúdo (lista do IUM dos medicamentos dentro do lote) e o CNPJ dos participantes da cadeia.

Figura 15: Consulta de informações do lote

```
? Selecione a opcao: Pesquisar lote
? Pesquisar: Informacoes do lote
? ID do lote: 1286936256074369


** INFORMACOES DO LOTE **
-------------------------
ID do lote: 1286936256074369
Medicamento: Aspirina
Fabricante: Laboratorio X
Data de fabricacao: 31/07/2019
Prazo de validade: 31/07/2020
Conteudo: [ 'code-01', 'code-02', 'code-03' ]
CNPJ do fabricante: 123.456.789-01
CNPJ da transportadora: 369.258.147-02
CNPJ da farmacia: 159.263.487-03

******************************************************
```

Fonte: A autora (2019).

Na Figura 16 observa-se uma consulta do tipo "rastrear" pelo aplicativo, onde é passado o ID do lote e como resultado mostra na tela os registros que a *blockchain* possui da "DataA", "DataB" e "DataC". Ainda não há registro de "DataB" e "DataC" para este lote.

Figura 16: Consulta de rastreamento pela farmácia

```
bruna@bruna:~/Documents/pd/tracking/cli/farmacia$ ./cli-farmacia
? Selecione a opcao: Pesquisar lote
? Pesquisar: Rastrear lote
? ID do lote: 1281438697935489


** HISTORICO DO LOTE **
-----------------------
ID do lote: 1281438697935489
DataA: 2019-7-31 18:35:58
DataB: Nao ha registro
DataC: Nao ha registro

******************************************************
```

Fonte: A *autora* (2019).

Na Figura 17 observa-se uma consulta do tipo "localizar" onde é passado o ID do lote e como resultado mostra na tela a última posição conhecida do lote, neste caso o "Ponto A", bem como a data registrada de leitura nesse ponto.

Figura 17: Consulta de posição pela transportadora

```
bruna@bruna:~/Documents/pd/tracking/cli/transportadora$ ./cli-transportadora
? Selecione a opcao: Pesquisar lote
? Pesquisar: Localizar lote
? ID do lote: 1277040651424385


** POSICAO DO LOTE **
---------------------
ID do lote: 1277040651424385
Ponto A:
2019-7-31 18:37:45

******************************************************
```

Fonte: A autora (2019).

O aplicativo do "Fabricante" possui ainda a opção "Registrar lote" que, ao ser selecionada, permite ao usuário inserir as informações do lote a ser registrado. Enquanto a transação está sendo validada pela *blockchain* lê-se na tela "Registrando lote…", e após o registro lê-se "LOTE REGISTRADO COM SUCESSO!", como observa-se na Figura 18.

Figura 18: Lote registrado pelo fabricante

```
** REGISTRO DE LOTE **
----------------------
? ID do lote: 1286936256074369
? Nome do medicamento Aspirina
? Nome do fabricante: Laboratorio X
? Data da fabricacao: 31/07/2019
? Prazo de validade: 31/07/2020
? Codigo dos medicamentos (separados por espaco): code-01 code-02 code-03
? CNPJ do fabricante: 123.456.789-01
? CNPJ da transportadora: 369.258.147-02
? CNPJ da farmacia: 159.263.487-03

Registrando lote...


** LOTE REGISTRADO COM SUCESSO! **
```

Fonte: A autora (2019).

7.2.4. Software embarcado no leitor

O *software* embarcado no Arduíno foi desenvolvido na IDE Arduíno, e começa com a inicialização da serial para comunicação com o módulo wi-fi e a configuração dos módulos RC522 e wi-fi. O método "rfid_read", que é executado em um laço de repetição infinito, identifica quando uma tag foi aproximada, lê o ID e o

envia para o método "send_transaction". O método "send_transaction" envia o ID via wi-fi para um nó do Parity como uma transação.

As bibliotecas utilizadas foram:

- MFRC522: permite a configuração e a comunicação com o módulo RFID RC522;
- Wi-fi Esp: permite a configuração e comunicação com o módulo ESP8266.
- SPI: utilizada para comunicação serial com o módulo RFID RC522.

Para invocar o método "data" do contrato "Interface" enviando como parâmetro o ID da etiqueta lida, utilizamos transações. Uma transação pode ter vários parâmetros, dos quais três foram utilizados (PARITY INC, 2019):

- "From": endereço de quem envia a transação;
- "To": endereço de quem recebe a transação; neste caso é o endereço do contrato que contém o método a ser chamado;
- "Data": o *hash* da assinatura do método a ser chamado seguido pelos parâmetros que serão passados codificados em hexadecimal e completados com zeros à esquerda até chegar a 32 bytes.

A assinatura do método trata-se do nome do método seguido pelo tipo dos parâmetros entre parênteses, separados por vírgula (PARITY INC, 2019).

Como dito na metodologia, três versões do software embarcado foram criadas para emular três leitores diferentes. A diferença entre as versões é o nó para o qual vai ser enviada a transação e o campo "From" da transação. O leitor do "Ponto A" envia para o nó do "Fabricante" e o campo "From" contém o endereço do "Fabricante". A mesma lógica segue para os leitores dos pontos B e C, associados à "Transportadora" e à "Farmácia", respectivamente.

8. RESULTADOS

8.1 Testes de software

Foram desenvolvidos testes de software para verificar que o sistema desenvolvido atendia aos requisitos. Para tanto, criou-se um nó chamado "Nó 0" que possui as chaves do "Fabricante", "Transportadora" e "Farmácia". Todos os testes foram executados com sucesso no dia 31 de julho de 2019 em um computador com processador i5-8400, 8GB de memória RAM e sistema operacional Ubuntu 18.04. Os testes foram escritos em Javascript utilizando-se o editor de texto Visual Studio Code e foram executados com o Truffle versão 5.0.26, gerando resultados consistentes com os requisitos definidos na metodologia.

8.1.1. Testes de registro de lote

Foram escritos três testes para o método registrar, que deve persistir na *blockchain* as informações do lote apenas quando invocado pela conta do fabricante. Os testes são descritos na Tabela 3.

Tabela 3. Testes de registro de lote

NOME DO TESTE	DESCRIÇÃO	RESULTADO
"Fabricante" deve conseguir registrar lote	Invoca-se o método "registrar" do contrato "Interface" através da conta do "Fabricante".	O lote é registrado com sucesso.
"Transportadora" não deve conseguir registrar lote	Invoca-se o método "registrar" do contrato "Interface" através da conta da "Transportadora".	Ocorre um erro e o lote não é registrado.
"Farmácia" não deve conseguir registrar lote	Invoca-se o método "registrar" do contrato "Interface" através da conta da "Farmácia".	Ocorre um erro e o lote não é registrado.

Fonte: A autora, 2019.

8.1.2. Testes de consulta

Desenvolveu-se um teste para cada um dos métodos de consulta: “infoLote”, “rastrear” e “localizar”. No momento da execução destes testes ainda não havia registro de nenhuma data na *blockchain*. Os testes são descritos na Tabela 4.

Tabela 4. Testes de pesquisa

NOME DO TESTE	DESCRIÇÃO	RESULTADO
Deve consultar informações do lote pelo ID	Invoca-se o método "infoLote" do contrato "Interface".	Retorna as informações do lote consistente com o que foi registrado.
Deve consultar as datas pelo ID do lote	Invoca-se o método "rastrear" do contrato "Interface".	Retorna as três datas vazias.
Deve retornar a última posição conhecida	Invoca-se o método "localizar" do contrato "Interface".	Retorna "Nenhuma data registrada".

Fonte: A autora, 2019.

8.1.3. Testes de registro de data

Nestes testes foi simulado o registro das datas de leitura de lotes na *blockchain*. Para tanto, o método "data" é invocado a partir de contas diferentes ("Fabricante", "Transportadora" e "Farmácia") passando como parâmetro um ID que identifica um suposto lote. Espera-se que a data enviada pelo "Fabricante" seja registrada como "DataA", a enviada pela "Transportadora" seja registrada como "DataB" e a enviada pela "Farmácia" seja registrada como "DataC".

As datas registradas são as datas da execução do teste. Portanto, nos testes em que se registra data, declarou-se uma variável onde será armazenada a data e hora da execução do teste para posterior verificação: os métodos "rastrear" e "localizar" devem retornar datas consistentes com o que foi registrado. Os testes são descritos nas Tabelas 5, 6 e 7.

Tabela 5. Testes de registro e consulta da DataA

NOME DO TESTE	DESCRIÇÃO	RESULTADO
Leitor do "Fabricante" deve conseguir registrar a "DataA"	Invoca-se o método "data" do contrato "Interface" através da conta do "Fabricante".	A "DataA" é registrada com sucesso.
Consulta à "DataA" deve retornar o que foi registrado	Invoca-se o método "rastrear" do contrato "Interface".	Retorna as três datas e a "DataA" idêntica àquela registrada no teste anterior.
Consulta de posição deve retornar "DataA"	Invoca-se o método "localizar" do contrato "Interface".	Retorna as três datas e a "DataA" idêntica àquela registrada no teste anterior.

Fonte: A autora, 2019.

Tabela 6. Testes de registro e consulta da DataB

NOME DO TESTE	DESCRIÇÃO	RESULTADO
Leitor da "Transportadora" deve conseguir registrar a DataB	Invoca-se o método "data" do contrato "Interface" através da conta da "Transportadora".	A "DataB" é registrada com sucesso.
Consulta à "DataB" deve retornar o que foi registrado	Invoca-se o método "rastrear" do contrato "Interface".	Retorna as três datas e a "DataB" idêntica àquela registrada no teste anterior.
Consulta de posição deve retornar "DataB"	Invoca-se o método "localizar" do contrato "Interface".	Retorna as três datas e a "DataB" idêntica àquela registrada no teste anterior.

Fonte: A autora, 2019.

Tabela 7. Testes de registro e consulta da DataC

NOME DO TESTE	DESCRIÇÃO	RESULTADO
Leitor da "Farmácia" deve conseguir registrar a "DataC"	Invoca-se o método "data" do contrato "Interface" através da conta da "Farmácia".	A "DataC" é registrada com sucesso.
Consulta à "DataC" deve retornar o que foi registrado	Invoca-se o método "rastrear" do contrato "Interface".	Retorna as três datas e a "DataC" é idêntica àquela registrada no teste anterior.
Consulta de posição deve retornar "DataC"	Invoca-se o método "localizar" do contrato "Interface".	Retorna as três datas e a "DataC" idêntica àquela registrada no teste anterior.

Fonte: A autora, 2019.

Três outros testes foram desenvolvidos para garantir que a mesma data não possa ser registrada duas vezes, e as datas devem ser registradas em ordem: primeiro a "DataA", seguida da "DataB" e por último a "DataC". Os testes são descritos na Tabela 8.

Tabela 8. Outros testes de registro de data

NOME DO TESTE	DESCRIÇÃO	RESULTADO
Não deve registrar a mesma data duas vezes	Invoca-se o método "data" do contrato "Interface" através da conta do “Fabricante” passando o ID de um lote em que já há registro da "DataA".	A nova "DataA" não é registrada.
Não deve registrar a "DataB" antes da "DataA"	Invoca-se o método "data" do contrato "Interface" através da conta da “Transportadora” passando o ID de um lote em que não há registro da "DataA".	A "DataB" não é registrada.
Não deve registrar a "DataC" antes da "DataB"	Invoca-se o método "data" do contrato "Interface" através da conta da “Farmácia” passando o ID de um lote em que não há registro da "DataB".	A "DataC" não é registrada.

Fonte: A autora, 2019.

8.1.4. Testes de segurança

Foram desenvolvidos dois testes para garantir a segurança dos dados, descritos na Tabela 9.

Tabela 9. Testes de segurança

NOME DO TESTE	DESCRIÇÃO	RESULTADO
Não deve ser possível acessar métodos pelo contrato de "Storage"	Invoca-se o método "data" do contrato Storage.	Ocorre um um erro e o registro não é efetuado.
Não deve ser possível pesquisar lote por outra conta	Invoca-se o método "infoLote" do contrato "Interface" através de uma conta não registrada na blockchain.	Ocorre um um erro e as informações do lote não são retornadas.

Fonte: A autora, 2019.

8.2 Testes de sistema

Para validar o funcionamento do sistema como um todo, foram realizados testes no dia 31 de julho de 2019. Os testes foram realizados utilizando um computador com processador i5-8400, 8GB de memória RAM e sistema operacional Ubuntu 18.04. Foram utilizadas três etiquetas de ID's 1286936256074369, 1281438697935489, 1277040651424385, doravante chamadas primeira, segunda e terceira etiqueta, respectivamente. Lotes com esses ID's foram registrados na *blockchain* a partir do aplicativo do "Fabricante".

O Arduíno foi configurado como "Leitor A" e as três etiquetas foram lidas. Na Figura 19 observa-se que uma consulta do tipo "rastrear" pelo aplicativo do "Fabricante", passando o ID da primeira etiqueta retorna que a "DataA" é igual ao instante de leitura da primeira etiqueta, dia 31/7/2019 às 18:20:05. Ainda não há registro de "DataB" e "DataC".

Figura 19: Consulta de rastreamento lote 1286936256074369

```
bruna@bruna:~/Documents/pd/tracking/cli/fabricante$ ./cli-fabricante
? Selecione a opcao: Pesquisar lote
? Pesquisar: Rastrear lote
? ID do lote: 1286936256074369


** HISTORICO DO LOTE **
-----------------------
ID do lote: 1286936256074369
DataA: 2019-7-31 18:20:05
DataB: Nao ha registro
DataC: Nao ha registro

******************************************************
```

Fonte: A autora (2019).

Em seguida realizou-se uma consulta do tipo "rastrear" pelo aplicativo da "Farmácia" passando o ID da segunda etiqueta. Observa-se na Figura 20 que a "DataA" é a data de leitura daquela etiqueta, dia 31/7/2019 às 18:35:58, e não há registro de "DataB" e "DataC".

Figura 20: Consulta de rastreamento do lote 1281438697935489

```
bruna@bruna:~/Documents/pd/tracking/cli/farmacia$ ./cli-farmacia
? Selecione a opcao: Pesquisar lote
? Pesquisar: Rastrear lote
? ID do lote: 1281438697935489


** HISTORICO DO LOTE **
-----------------------
ID do lote: 1281438697935489
DataA: 2019-7-31 18:35:58
DataB: Nao ha registro
DataC: Nao ha registro

******************************************************
```

Fonte: A autora (2019).

Por fim, realizou-se uma consulta do tipo "localizar" pelo aplicativo da "Transportadora", passando o ID da terceira etiqueta. Observa-se na Figura 21 que a última posição conhecida para este lote é o "Ponto A", e a data de passagem por esse ponto é a data em que se leu a etiqueta com este ID, dia 31/7/2019 às 18:37:45.

Figura 21: Consulta de posição do lote 1277040651424385

```
bruna@bruna:~/Documents/pd/tracking/cli/transportadora$ ./cli-transportadora
? Selecione a opcao: Pesquisar lote
? Pesquisar: Localizar lote
? ID do lote: 1277040651424385


** POSICAO DO LOTE **
---------------------
ID do lote: 1277040651424385
Ponto A:
2019-7-31 18:37:45

******************************************************
```

Fonte: A autora (2019).

Em seguida o Arduíno foi configurado como "Leitor B" e novamente as três etiquetas foram lidas. Na Figura 22 observa-se que uma consulta do tipo "rastrear" pelo aplicativo da "Farmácia" passando o ID da primeira etiqueta. Agora há registro da "DataB" como dia 31/7/2019 às 19:03:50. A "DataA" continua idêntica àquela observada na Figura 19 e ainda não registro da "DataC"

Figura 22: Consulta de rastreamento lote 1286936256074369

```
bruna@bruna:~/Documents/pd/tracking/cli/farmacia$ ./cli-farmacia
? Selecione a opcao: Pesquisar lote
? Pesquisar: Rastrear lote
? ID do lote: 1286936256074369


** HISTORICO DO LOTE **
-----------------------
ID do lote: 1286936256074369
DataA: 2019-7-31 18:20:05
DataB: 2019-7-31 19:03:50
DataC: Nao ha registro

******************************************************
```

Fonte: A autora (2019).

Na Figura 23 observa-se uma consulta do tipo "localizar" pelo aplicativo passando o ID da segunda etiqueta. A última posição conhecida para este lote é o "Ponto B" e a data de passagem por este ponto é dia 31/7/2019 às 19:04:33.

Figura 23: Consulta de posição do lote 1281438697935489

```
? Selecione a opcao: Pesquisar lote
? Pesquisar: Localizar lote
? ID do lote: 1281438697935489


** POSICAO DO LOTE **
---------------------
ID do lote: 1281438697935489
Ponto B:
2019-7-31 19:04:33

*****************************************************
```

Fonte: A autora (2019).

A consulta do tipo "rastrear" pelo aplicativo do "Fabricante" passando o ID da terceira etiqueta retorna que a "DataB" é dia 31/7/2019 às 19:05:18, a "DataA" é idêntica àquela consultada anteriormente e ainda não há registro da "DataC", como observa-se na Figura 24.

Figura 24: Consulta de rastreamento do lote 1277040651424385

```
bruna@bruna:~/Documents/pd/tracking/cli/fabricante$ ./cli-fabricante
? Selecione a opcao: Pesquisar lote
? Pesquisar: Rastrear lote
? ID do lote: 1277040651424385


** HISTORICO DO LOTE **
-----------------------
ID do lote: 1277040651424385
DataA: 2019-7-31 18:37:45
DataB: 2019-7-31 19:05:18
DataC: Nao ha registro

*************************************************
```

Fonte: A autora (2019).

Por fim o Arduíno foi configurado como "Leitor C" e mais uma vez as três etiquetas foram lidas. Na Figura 25 observa-se que a consulta do tipo "localizar" realizada pelo aplicativo da "Farmácia" passando o ID da primeira etiqueta retorna que a última posição conhecida do lote é o "Ponto C", e a data de passagem por este ponto é dia 31/7/2019 às 19:10:24.

Figura 25: Consulta de posição do lote 1286936256074369

```
bruna@bruna:~/Documents/pd/tracking/cli/farmacia$ ./cli-farmacia
? Selecione a opcao: Pesquisar lote
? Pesquisar: Localizar lote
? ID do lote: 1286936256074369


** POSICAO DO LOTE **
---------------------
ID do lote: 1286936256074369
Ponto C:
2019-7-31 19:10:24

******************************************************
```

Fonte: A autora (2019).

Uma consulta do tipo "rastreamento" passando o ID da segunda etiqueta retorna que a "DataC" é dia 31/7/2019 às 19:11:17, como observa-se na Figura 26. "DataA" e "DataB" estão coerentes com as consultas anteriores.

Figura 26: Consulta de rastreamento lote 1281438697935489

```
? Selecione a opcao: Pesquisar lote
? Pesquisar: Rastrear lote
? ID do lote: 1281438697935489


** HISTORICO DO LOTE **
-----------------------
ID do lote: 1281438697935489
DataA: 2019-7-31 18:35:58
DataB: 2019-7-31 19:04:33
DataC: 2019-7-31 19:11:17

******************************************************
```

Fonte: A autora (2019).

Na Figura 27 observa-se que uma consulta do tipo "localizar" passando o ID da terceira etiqueta retorna que a última posição conhecida do lote é o "Ponto C" e data de passagem por este ponto é dia 31/7/2019 às 19:11:52

Figura 27: Consulta de rastreamento do lote 1277040651424385

```
? Selecione a opcao: Pesquisar lote
? Pesquisar: Localizar lote
? ID do lote: 1277040651424385


** POSICAO DO LOTE **
---------------------
ID do lote: 1277040651424385
Ponto C:
2019-7-31 19:11:52

*********************************************************
```

Fonte: A autora (2019).

9. CONCLUSÃO E TRABALHOS FUTUROS

O objetivo deste trabalho era implementar um sistema de rastreamento de medicamentos onde os lotes eram identificados por etiquetas RFID cujo ID era lido por leitores em três pontos de verificação e os dados de rastreamento eram armazenados em uma *blockchain* privada. Os testes de *software* mostram que o sistema desenvolvido atende aos requisitos propostos, e os testes de sistema mostram que o objetivo global foi alcançado.

Como determina a legislação, é possível relacionar o número único de identificação dos lotes com a lista dos IUM dos medicamentos nele contidos, que fica armazenada na *blockchain* e disponível para todos os participantes da cadeia. É possível traçar o histórico de cada lote, bem como sua última posição conhecida através dos aplicativos desenvolvidos. Além disso, o sistema desenvolvido garante confiabilidade nos registros pois:

- A utilização da tecnologia RFID reduz a interferência humana na coleta dos dados e possibilita a redução de fraude e falhas;
- A *blockchain* assegura que a fonte dos dados é confiável: a data somente é registrada quando o ID foi enviado por um leitor registrado na *blockchain*;
- A criptografia e os conceitos de rede distribuída utilizados na *blockchain* tornam muito difícil a manipulação dos dados após o registro.

Destaca-se ainda que o caráter descentralizado da *blockchain* leva a uma maior robustez contra indisponibilidade de informações.

O sistema foi desenvolvido considerando-se uma cadeia de movimentação simplificada, constituída por apenas três participantes e limitada a três pontos de verificação. No entanto, poderia ser expandido para cadeias mais complexas, com diversos participantes envolvidos na fabricação ou importação, distribuição e venda dos medicamentos. Todos seriam registrados na *blockchain*, e a movimentação dos medicamentos seria controlada através do registro dos lotes em diversos pontos. A base de dados do governo, que segundo determinação da Anvisa deve reunir os dados do rastreamento de todos os medicamentos produzidos ou importados pelo

Brasil, também poderia ser uma *blockchain*, capaz de receber os dados de diversas *blockchains* de rastreamento formada por diferentes parceiros comerciais.

Entre as dificuldades encontradas durante a realização deste trabalho, destaca-se a precariedade da documentação das bibliotecas de código aberto. A ideia inicial era utilizar uma biblioteca Web3 para Arduíno, semelhante àquela utilizada para criação dos aplicativos em Javascript. Segundo à documentação existente, a biblioteca era compatível com o Arduíno e podia ser utilizada para acessar métodos de contratos enviando parâmetros com assinatura digital. No entanto, ao tentar utilizar a biblioteca descobriu-se que esta não era compatível com o Arduíno, uma vez que os desenvolvedores a testaram em outro microcontrolador e nunca concluíram a compatibilização. Após extensa pesquisa tentando encontrar algum substituto para a biblioteca, decidiu-se utilizar uma comunicação de baixo nível com a *blockchain*, por meio de transações e sem assinatura.

REFERÊNCIAS

AMBROSUS, Team. Ambrosus white paper. 2018. Disponível em: https://ambrosus.com/assets/en/-White-Paper-V8-1.pdf. Acesso em: 4 de abril de 2019.

ANGELIS, Stefano De. Assessing Security and Performances of Consensus algorithms for Permissioned Blockchains. Dissertação de Mestrado em Engenharia da Computação - Universidade Sapienza, Roma, 2016.

ANVISA. Resolução da Diretoria Colegiada – RDC n° 157. 2017. Disponível em: http://portal.anvisa.gov.br/documents/10181/2724161/RDC_157_2017_.pdf/a91d19ef-937e-432b-97b0-4bf9cb75062e. Acesso em: 11 de agosto de 2019.

AVRAMENKO, Alexey. Ethereum and smart contracts basics. 2017. Disponível em: https://medium.com/@olxc/ethereum-and-smart-contracts-basics-e5c84838b19. Acesso em: 5 de julho de 2019.

BERGAMO FILHO, Clóvis et al. Ruptura No Modelo Tradicional Das Empresas. Rio de Janeiro: Brasport, 2019.

BRASIL. Lei nº 11.903 de 14 de janeiro de 2009. Disponível em: http://www.planalto.gov.br/ccivil_03/_Ato2007-2010/2009/Lei/L11903.htm. Acesso em: 11 de agosto de 2019.

BRASIL. Lei nº 13.410 de 28 de dezembro de 2016. Disponível em: http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2016/lei/L13410.htm. Acesso em: 11 de agosto de 2019.

COLBACH, Gordon. RFID Handbook: Technology, Applications, Security and Privacy. Miami: Crc, 2008.

EASYEDA. RFID MFRC522. 2016. Disponível em: https://easyeda.com/gerrychen/RFID_MFRC522-lHBSasmEW. Acesso em 21 de agosto de 2019.

ESPRESSIF. ESP8266EX Datasheet. 2019. Disponível em: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf. Acesso em: 11 de agosto de 2019.

FAROOQ, U. et al. RFID Based Security and Access Control System. IACSIT International Journal of Engineering and Technology, vol. 6, no. 4, 2014.

GREVE, Fabíola et al. Blockchain e a Revolução do Consenso sob Demanda. In: Minicursos do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos. Ed. 36. São Paulo: Sociedade Brasileira de Computação, 2018.

IYER, KEDAR; DANNEN CHRIS. Building Games with Ethereum Smart Contracts: Intermediate Projects for Solidity Developers. Nova York: Apress, 2018.

JIA, Xiaolin et al. In: International Conference on Consumer Electronics, Communications and Networks (CECNet), 2, 2012, China. RFID technology and its applications in Internet of Things (IoT). Estados Unidos: IEEE, 2012.

KENT, Beck. Test-Driven Development by Example. Boston: Addison Wesley: 2002.

MACIEL, Marco Antonio. Segurança e Certificado Digital. 2010. Disponível em: http://www.mmaciel.com.br/tags/assinatura-digital/. Acesso em: 10 de abril de 2019.

MASSESSI, Demiro. Public Vs Private Blockchain In A Nutshell. 2018. Disponível em: https://medium.com/coinmonks/public-vs-private-blockchain-in-a-nutshell-c9fe284fa39f. Acesso em: 17 de abril de 2019.

MENCK, Denise. Entenda a lei para rastreabilidade de medicamentos. 2014. Disponível em: https://www.quebeckautomacao.com.br/entenda-a-lei-para-rastreabilidade-de-medicamentos. Acesso em 4 de julho de 2019.

MÜLLER, M. LEONARDI, E.E.. Qual será o final da novela da rastreabilidade de medicamentos? 2017. Disponível em https://www.ictq.com.br/industria-farmaceutica/761-qual-sera-o-final-da-novela-da-rastreabilidade-de-medicamentos. Acesso em 4 de julho de 2019.

NAKAMOTO, S. Bitcoin: a peer-to-peer electronic cash system. 2008. Disponível em: https://bitcoin.org/bitcoin.pdf. Acesso em: 5 de abril de 2019.

NXP. MFRC522: Standard performance MIFARE and NTAG frontend. 2016. Disponível em: https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf. Acesso em: 11 de agosto de 2019.

NXP. NTAG213/215/216: NFC Forum Type 2 Tag compliant IC with 144/504/888 bytes user memory. 2015. Disponível em: https://www.nxp.com/docs/en/data-sheet/NTAG213_215_216.pdf. Acesso em: 11 de agosto de 2019.

OLIVEIRA, Ronielton R. Criptografia tradicional simétrica de chave privada e criptografia assimétrica de chave pública: análise das vantagens e desvantagens. Trabalho de pós-graduação em criptografia e segurança de redes - Universidade Federal Fluminense, 2006.

PARITY INC. Parity Tech Documentation. 2019. Disponível em: https://wiki.parity.io/. Acesso em: 16 de agosto de 2019.

RAKIC, B. et al. First purpose built protocol for supply chains based on blockchain. 2017. Disponível em: https://origintrail.io/storage/documents/OriginTrail-White-Paper.pdf. Acesso em: 5 de abril de 2019.

ROCHA, J. et al. Peer-to-Peer: Computação Colaborativa na Internet. Minicursos, XXII SBRC, Gramado-Rs, Maio 2004.

SOUSA, Marcelo Ferreira. RFID e suas aplicações - um estudo de caso com prateleiras inteligentes. Dissertação de Mestrado em Engenharia de Teleinformática - Universidade Federal do Ceará, 2010.

SELVAMANIKKAM, M. Digital Signature Generation. 2018. Disponível em: https://medium.com/@meruja/digital-signature-generation-75cc63b7e1b4. Acesso em 26 de julho de 2019.

SOMMERVILLE, I. Engenharia de Software. 8. ed. São Paulo: Pearson Prentice Hall, 2007.

TANENBAUM, Andrew S. Redes de computadores. 5ª edição. Londres: Pearson Universidades, 2011.

TEXAS INSTRUMENTS. LM317 3-Terminal Adjustable Regulator. 2016. Disponível em: http://www.ti.com/lit/ds/slvs044x/slvs044x.pdf. Acesso em 26 de julho de 2019.

TSCHORSCH, F.; SCHEUERMANN, B. Bitcoin and beyond: a technical survey on decentralized digital currencies. IEEE Communications Surveys and Tutorials, vol. 18, no. 3, pp. 2084–2123, 2016.

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO. Grupo de Teleinformática e Automação. Função Hashing. 2009. Disponível em: https://www.gta.ufrj.br/grad/09_1/versao-final/assinatura/hash.htm. Acesso em 30 de março de 2019.

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO. Grupo de Teleinformática e Automação. RFID: Como funciona. 2007. Disponível em: https://www.gta.ufrj.br/grad/07_1/rfid/RFID_arquivos/como%20funciona.htm. Acesso em 28 de agosto de 2019.