



BLOCKCHAIN APLICADO NO MONITORAMENTO E RASTREABILIDADE DA CADEIA DE SUPRIMENTOS FARMACÊUTICA

Palavras-Chave: BLOCKCHAIN, CADEIA DE SUPRIMENTOS FARMACÊUTICA, RASTREABILIDADE, MONITORAMENTO, SIMULAÇÃO

Autores:

MANUELA VERIDIANA PUGLIESI [UNIVERSIDADE ESTADUAL DE CAMPINAS]

Prof. Dr. PAULO SERGIO DE ARRUDA IGNÁCIO (orientador) [UNIVERSIDADE ESTADUAL DE CAMPINAS]

RESUMO

Inicialmente utilizado como anteparo de uma moeda digital, hoje o *blockchain* tem se inserido nos mais diversos tipos de indústria, haja vista a transparência, confiabilidade, eficiência e segurança consequentes de sua implementação. A simulação da aplicação do *blockchain* em uma cadeia de suprimentos farmacêutica simplificada teve por objetivo apontar as inúmeras vantagens da tecnologia. A simulação foi realizada em uma plataforma que emula o funcionamento do *Ethereum*, o mais famoso *blockchain* atualmente.

INTRODUÇÃO

O *blockchain* pode ser definido como uma base de dados (*database*) descentralizada e criptografada. A essa base de dados também se dá o nome de *ledger*. Ao se distribuir e, assim, descentralizar o poder sobre informações sensíveis contidas nessa base de dados, é possível reduzir custos e aumentar a eficiência de uma série de processos, em diversos setores da indústria (AQUINO, 2019).

Na literatura (REDA et al, 2020), (DAVE; PARIKH, 2019)(DWIVEDI; AMIN; VOLLALA, 2020), (AZZI; CHAMOON; SOKHN, 2019), existem algumas definições de *blockchain*. Este nada mais é do que uma cadeia de blocos, cada qual com sua identificação (a *hash*); todo bloco é vinculado ao anterior por meio da *hash*, de modo que é possível rastrear, bloco a bloco, determinada informação. Destacam-se algumas de suas características: descentralização, persistência, anonimidade, confiabilidade, transparência e segurança. Por essas características, o *blockchain* pode ser aplicado de modo bastante satisfatório para rastreabilidade e monitoramento em cadeias de suprimentos. A rastreabilidade e o monitoramento são viabilizados por meio da dita tecnologia, implementando-se sua transparência (SHILLING, 2018).

Os objetivos deste trabalho são, primariamente, avaliar a aplicação das tecnologias de *blockchain* no monitoramento e rastreabilidade de produtos e processos de cadeias produtivas e, secundariamente, identificar as principais características das estruturas de *blockchain* em cadeias produtivas, entender as principais dificuldades na implementação dessas tecnologias, verificar as melhorias no desempenho das cadeias produtivas, analisar os principais modelos matemáticos aplicados nas cadeias produtivas, identificar os segmentos de maior evolução sobre a aplicação de *blockchain* e, por fim, simular a aplicação do *blockchain* em uma cadeia de suprimentos.

SIMULAÇÃO

Remix IDE é um aplicativo *open source* de *web*, ou seja, que pode ser utilizado diretamente do navegador. Ele é usado para toda a jornada de desenvolvimento de um *smart contract*, dentro de um ambiente que simula a plataforma *Ethereum*. No *Remix IDE*, é possível escrever em uma linguagem de programação própria, o *Solidity*, que é muito similar a *Java*, mas com especificidades de *smart contracts*.

Assim, um contrato inteligente foi feito de modo a simular o armazenamento de informações em cada etapa da cadeia de suprimentos farmacêutica. Na Figura 1, há a visualização do ambiente de simulação com o início do contrato inteligente realizado. O contrato em sua integridade pode ser visualizado no código em nuvem no *Github* (PUGLIESI, 2021).

Figura 1: Smart contract produzido no Remix IDE.

```

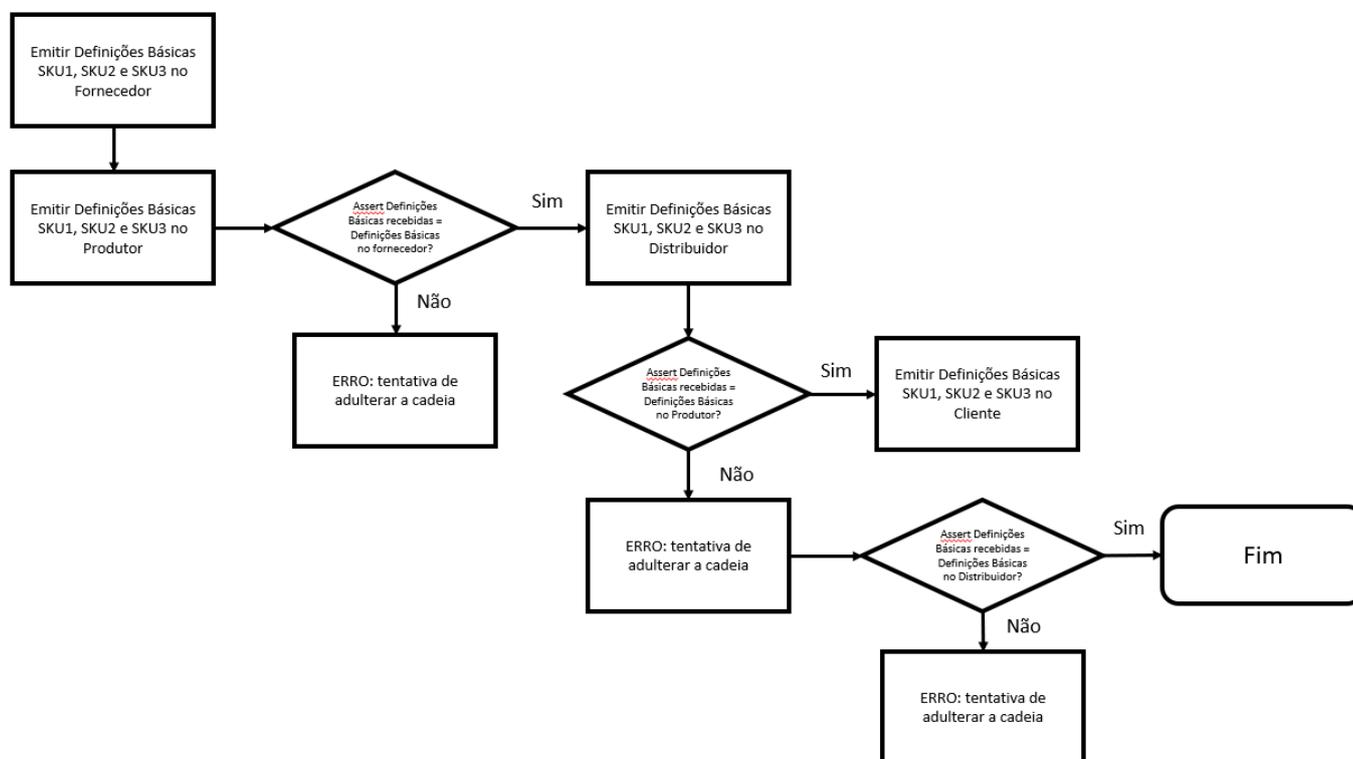
1 // SPDX-License-Identifier: Manu
2
3 pragma solidity >=0.4.0 <0.8.4;
4
5
6
7 contract cadeia {
8
9
10
11 event EmissaoPedidoFornecedor1 (address Fornecedor1, int16 DefBasicas1, int16 DefBasicas2, int16 DefBasicas3);
12 event DespachoPedidoFornecedor1 (int16 NSU1, int16 Quantidade1, int16 Lote1, int16 NSU2, int16 Quantidade2, int16 Lote2, int16 NSU3, int16 Quantidade3, int16 Lote3);
13 event EmitirNotaFornecedor1 (address Operador, int16 NSU1, int16 Quantidade1, int16 Precos1, int16 NSU2, int16 Quantidade2, int16 Precos2, int16 NSU3, int16 Quantidade3, int16 Precos3);
14
15 event EmissaoPedidoFornecedor2 (address Fornecedor2, int16 DefBasicasA, int16 DefBasicasB, int16 DefBasicasC);
16 event DespachoPedidoFornecedor2 (int16 NSUA, int16 QuantidadeA, int16 LoteA, int16 NSUB, int16 QuantidadeB, int16 LoteB, int16 NSUC, int16 QuantidadeC, int16 LoteC);
17 event EmitirNotaFornecedor2 (address Operador, int16 NSUA, int16 QuantidadeA, int16 PrecosA, int16 NSUB, int16 QuantidadeB, int16 PrecosB, int16 NSUC, int16 QuantidadeC, int16 PrecosC);
18
19 event ReceberProdutoProdutor (string hash);
20 event ReceberNotaProdutor (address Fornecedor1, address Fornecedor2, int16 DefBasicas1, int16 DefBasicas2, int16 DefBasicas3, int16 DefBasicasA, int16 DefBasicasB, int16 DefBasicasC);
21 event FabricaProdutor (address Operador, int16 DefBasicas1, int16 DefBasicas2, int16 DefBasicas3, int16 DefBasicasA, int16 DefBasicasB, int16 DefBasicasC);
22 event ProducaoProduto1Produtor (string Produto1, address Distribuidor, int16 S, int16 Quantidade12B, int16 Precos12B);
23 event ProducaoProduto2Produtor (string Produto2, int16 R, int16 Quantidade3AC, int16 Precos3AC);
24 event DespachoPedidoProdutor1 (string Produto1, address Distribuidor, int16 S, int16 Quantidade12B, int16 Precos12B);
25 event DespachoPedidoProdutor2 (string Produto2, address Distribuidor, int16 R, int16 Quantidade3AC, int16 Precos3AC);
26 event EmitirNotaProdutor1 (string Produto1, address Distribuidor, int16 S, int16 Quantidade12B, int16 Precos12B);
27 event EmitirNotaProdutor2 (string Produto2, address Distribuidor, int16 R, int16 Quantidade3AC, int16 Precos3AC);
28
29
30 event ReceberProdutoDistribuidor (string hash);
31 event ReceberNotaDistribuidor (address Operador, address Fornecedor1, address Fornecedor2, int16 DefBasicas1, int16 DefBasicas2, int16 DefBasicas3, int16 DefBasicasA, int16 DefBasicasB, int16 De
32 event DespachoPedidoDistribuidor (int16 DefBasicas1, int16 DefBasicas2, int16 DefBasicas3, int16 DefBasicasA, int16 DefBasicasB, int16 DefBasicasC);

```

Fonte: Autoria própria no Remix IDE

O smart contract produzido armazena informações dos SKUs e, a cada etapa da cadeia de suprimentos, valida se os *inputs* com as informações dos produtos recebidos são iguais aos dados armazenados na etapa prévia da cadeia. A lógica simplificada do código está na Figura 2, assumindo um fornecedor.

Figura 2: Fluxograma representativo da lógica do smart contract.



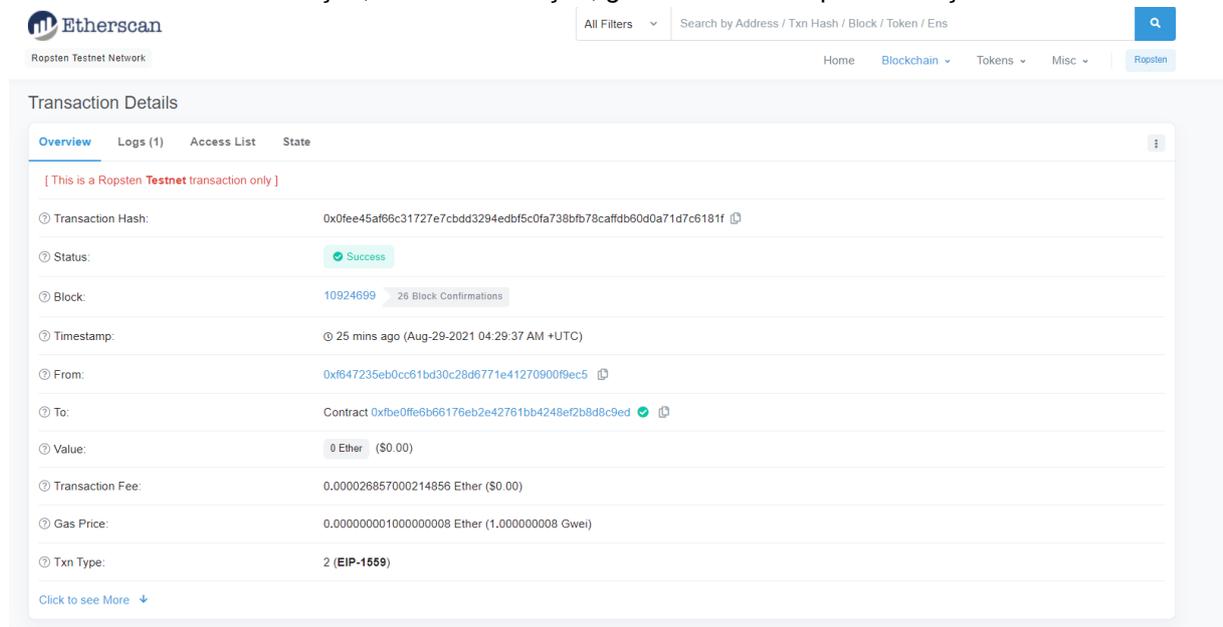
Fonte: Autoria própria

Em termos simples, o código é composto por diversos eventos que são acionados por meio das funções. A maioria das funções têm efeito de armazenamento dos *inputs* de cada uma das definições básicas dos SKUs, e esse tipo de função se chama *emit*. Vale notar que existem diversas funções *emit* dentro de cada componente da cadeia de suprimentos e também como comunicação entre esses componentes; ou seja, as informações armazenadas em dado *checkpoint* dentro de um dado componente são verificadas em outros *checkpoints* dentro desse mesmo componente e, a seguir, no componente seguinte. Para exemplificar, no caso do Fornecedor, há três funções *emit*, ou seja, três vezes em que são feitos *inputs*: emissão do pedido do fornecedor, despacho do pedido e emissão de nota fiscal.

Também existem funções *assert* que servem para validar informações entre os componentes da cadeia; esse tipo de função serve para, caso as informações forem incompatíveis, um erro ser acionado à *blockchain*. No caso da cadeia de suprimentos farmacêutica em específico, essa ferramenta é útil para verificar itens falsos, mal manipulados ou formulados.

A cada transação realizada no Remix IDE, um bloco é gerado. Suas informações podem ser visualizadas no Etherscan (2021), um suplemento do *Metamask*, a carteira de criptomoedas utilizada para gerar as transações; nenhum valor monetário real foi gasto para gerar a simulação, dado que é possível adquirir *Ethers* falsos na rede testes *ropsten faucet*. A visualização com as informações da transação e do bloco gerado pode ser vista no exemplo da Figura 3 abaixo.

Figura 3: Detalhes da transação final do Fornecedor 1, com informações da *hash* da transação, *status*, número do bloco, momento da transação, *addresses* do realizador da transação e do receptor da transação, valor da transação, taxa da transação, *gas* utilizado e o tipo de taxação.



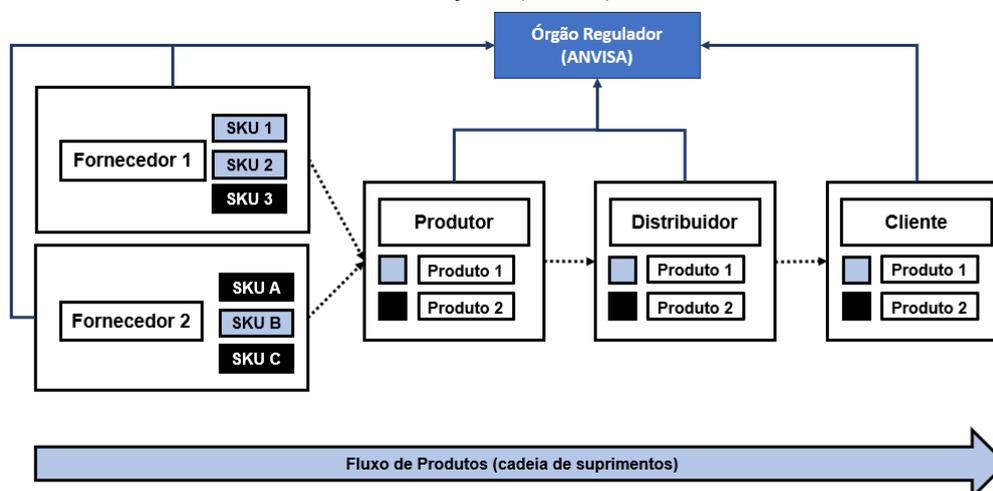
Fonte: Etherscan (2021)

RESULTADOS E DISCUSSÕES

Na Figura 4, pode-se observar uma cadeia de suprimentos farmacêutica simplificada sem uso do *blockchain*, isto é, com métodos de rastreabilidade e monitoramento convencionais.

Nessa cadeia de suprimentos farmacêutica, haverá dois Produtos, que são compostos por SKUs específicos. Inicialmente, os Fornecedores 1 e 2 enviam seis tipos diferentes de SKUs para o Produtor, que vai reuni-los para gerar os Produtos 1 e 2, dois tipos diferentes de medicamentos. O Produto 1 é composto pelos SKU 1, SKU 2 e SKU B e o Produto 2 é composto pelo SKU 3, SKU A e SKU C. Do Produtor, ambos os produtos são transportados em lotes para o Distribuidor, que por sua vez os transporta para o Cliente, uma rede de farmácias.

Figura 4: Representação da cadeia de suprimentos farmacêutica no Brasil, com um órgão de fiscalização das informações (Anvisa).

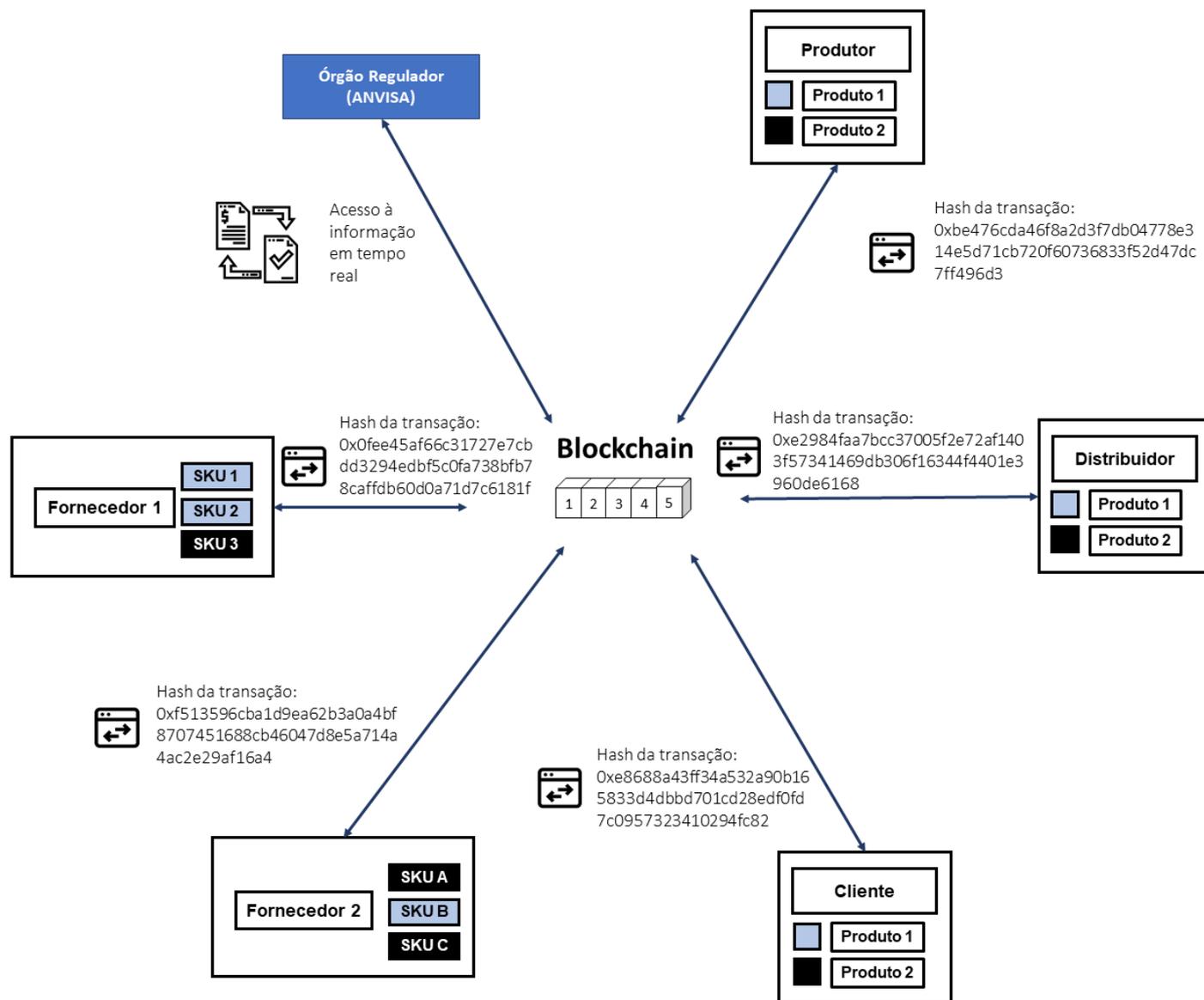


Fonte: Autoria própria

Um sistema convencional de rastreamento e monitoramento impescinde de identificadores. Idealmente, os Fornecedores 1 e 2 indicados e o Produtor utilizarão Notas Fiscais para implementarem seus sistemas; o Distribuidor se organiza com número de lotes de saída.

Já na Figura 5, há a representação de uma cadeia de suprimentos farmacêutica simplificada com a aplicação de *blockchain* para rastrear e monitorar as informações referentes aos produtos, isto é, desde incipiente e princípio ativo brutos (matéria prima) até o produto acabado nos canais de venda (farmácias).

Figura 5: Representação da cadeia de suprimentos farmacêutica com a implementação do *blockchain* para armazenamento das informações.



Fonte: Autoria própria

A diferença entre os modelos apresentados nas Figuras 4 e 5 é a descentralização do armazenamento da informação. No modelo de cadeia sem aplicação do *blockchain*, o Órgão Regulador é o responsável por armazenar e fiscalizar as informações entre os agentes da cadeia de suprimentos.

Já no modelo de cadeia em que se utiliza o *blockchain* como *ledger*, cada participante tem acesso a informações dos SKUs da etapa anterior e da seguinte, de modo que essa transparência viabiliza maior agilidade na verificação e validação dos dados compartilhados entre os participantes. Em outras palavras, o Órgão Regulador não mais seria responsável por armazenar as informações e não mais deteria poder sobre os dados a respeito dos produtos, pois os próprios agentes da cadeia fariam os *inputs* (NSU, Quantidade, Lote, Turno, Operador etc), que, ao se diferirem dos *inputs* feitos pelo agente anterior da cadeia, são

rejeitados e o sistema aciona para os participantes que houve algum erro, falsificação ou adulteração dos dados.

CONCLUSÃO

Verifica-se que é possível melhorar muitos aspectos da rastreabilidade e do monitoramento. Na cadeia de suprimentos farmacêutica, em específico, a burocracia existente para fiscalização e regulamentação de diversos medicamentos entre os fabricantes, distribuidores e vendedores desses produtos e órgãos reguladores pode ser reduzida. A redução dessa burocracia, acompanhada de maior transparência, confiabilidade das informações e segurança proporcionada pela criptografia, tornaria a rastreabilidade e monitoramento menos oneroso para as diversas partes envolvidas.

Ademais, o resultado da pesquisa indica que a aplicação pode ser implementada na realidade. O sistema de rastreabilidade e monitoramento atual, apesar de funcional, tem diversas desvantagens se comparado à proposta deste trabalho. A cadeia utilizada é teórica e simplificada, mas é possível trazer mais complexidade conforme a necessidade e existência de investimentos reais. Atualmente, no Brasil, a tecnologia não é usada para este fim específico, mas os agentes que tomarem a iniciativa de fazê-lo, claramente, terão consideráveis vantagens competitivas a seu dispor.

BIBLIOGRAFIA

AQUINO, Magali. **Adoção de *blockchain* na gestão de cadeias de suprimentos do Brasil**. Orientador: Priscila Laczynski de Souza Miguel. 2019. 86 f. Tese de Mestrado (Gestão para a Competitividade) - Fundação Getúlio Vargas Escola de Administração de Empresas de São Paulo, São Paulo, 2019. *E-book* (86p.).

AZZI, Rita; CHAMOUN, Rima; SOKHN, Maria. **The power of a blockchain-based smart contract**. Elsevier, [s. l.], v. 135, ed. 1, p. 582-592, 18 jun. 2019.

DAVE, Dharmin; PARIKH, Shalin; PATEL, Rema; DOSHI, Nishant. **A survey on blockchain technology and its proposed solutions**. Procedia Computer Science, Coimbra, 7 nov. 2019. 3rd International Workshop on Recent Advances on Internet of Things: Technology and Application Approaches, p. 740-745.

DWIVEDI, Sanjeev Kumar; AMIN, Ruhul; VOLLALA, Satyanarayana. **Blockchain based secured information sharing protocol in smart contract management system with key distribution mechanism**. Journal of Information Security and Applications, Naya Raipur, p. 1-15, 29 fev. 2020. DOI 10.1016. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S2214212620308255>. Acesso em: 20 out. 2020.

ETHERSCAN. Transaction details. In: **Etherscan** (ETH). [S. l.], Janeiro 2021. Disponível em: <https://etherscan.io/>. Acesso em: 10 maio 2021.

PUGLIESI, Manuela. **Github**. [S. l.], 9 ago. 2021. Disponível em: <https://github.com/ManuelaPugliesi/blockchain/blob/main/VF>. Acesso em: 9 ago. 2021.

SHILLING Rose. **The Smart contract Safe**. Food Engineering Magazine. Issue October 2018. pp.54-59.