

Desenvolvimento de uma aplicação para gestão da produção de produtos orgânicos utilizando uma arquitetura com a tecnologia Blockchain.

Richard Weber¹
Elder F. F. Bernardi²

RESUMO

Este trabalho apresenta o estudo da utilização da arquitetura e protocolo da tecnologia Blockchain, como possível solução para problemas de sustentabilidade prática de produção e processamento que promovam um equilíbrio aos problemas de qualidade, segurança e boa qualidade dos alimentos. Buscando relacionar a tecnologia Blockchain com a rastreabilidade da cadeia produtiva de alimentos, será desenvolvida uma aplicação para a gestão e controle de processos realizados na produção dos produtos orgânicos. Como resultado, percebeu-se uma possibilidade na utilização da Blockchain como meio de armazenamento destes dados, devido a sua capacidade de transparência, imutabilidade e segurança.

Palavras-chave: Blockchain. Produtos Orgânicos.

INTRODUÇÃO

A rastreabilidade da cadeia produtiva de alimentos surgiu como consequência da preocupação dos consumidores com a sanidade dos produtos, gerada após vários fatos que afetaram a saúde das pessoas em diversos países (SILVA, 2005).

Contaminações de alimentos por resíduos tóxicos, crises sanitárias e até mesmo a manipulação genética trouxeram a necessidade, por parte do consumidor final, de conhecer a origem, forma que os alimentos consumidos foram produzidos e quais os possíveis riscos que podem apresentar à saúde humana (NICOLOSO, 2013, p. 80).

Os padrões de consumo de alimentos estão mudando rapidamente hoje em dia, como resultado de questões de desenvolvimento e sustentabilidade, considerações com relação ao seu aspecto nutricional e também questões relacionadas à saúde.

Planos de ações de segurança dos alimentos estão sendo adotados pelos

¹Graduando em Tecnologia em Sistemas para Internet pela IFSUL. richardbvb123@gmail.com
²elder.bernardi@passofundo.ifsul.edu.br.

produtores, nas etapas de cultivo, colheita, classificação, embalagem, transporte até a distribuição, com a finalidade de aumentar a qualidade sanitária do produto comercializado.

Mesmo que tenha uma tendência de consumo sobre os alimentos orgânicos, ainda existem muitas dúvidas acerca do assunto, o que faz com que muitas pessoas hesitem em comprá-los. A auditabilidade da cadeia produtiva desses produtos pode ser uma solução para o problema, trazendo mais confiança ao consumidor.

O padrão Blockchain surge como alternativa para o armazenamento de dados, visto que, o uso da tecnologia garante nativamente proteção contra alteração de informações, proporcionando a centralização das informações, descentralização do armazenamento, transparência no processo de armazenamento e auditabilidade. Qualquer nó pode verificar o histórico das transações daquele protocolo sem a necessidade de nenhum tipo de autorização prévia, tornando o uso da tecnologia Blockchain uma solução para o armazenamento de dados referentes a registros de rastreabilidade, trazendo mais confiança ao consumidor que terá a possibilidade de acompanhar a cadeia produtiva do produto.

O artigo está estruturado da seguinte forma: na Seção 1 são descritas os conceitos e tecnologias utilizadas para o desenvolvimento da aplicação de rastreabilidade utilizando o Blockchain; na Seção 2 é detalhado a proposta da aplicação, quais os fundamentos e funcionamento ideal da aplicação; na Seção 3 é descrito como a aplicação foi desenvolvida e as suas principais funcionalidades; na Seção 3 denota-se às considerações finais, juntamente com o planejamento das melhorias futuras e possíveis novas funcionalidades para a aplicação.

1 REFERENCIAL TEÓRICO E TRABALHOS RELACIONADOS

No decorrer deste capítulo serão abordados os principais conceitos e tecnologias para a elaboração deste projeto, com foco em rastreabilidade da cadeia produtiva utilizando uma tecnologia Blockchain.

1.1 RASTREABILIDADE DA CADEIA PRODUTIVA

O consumidor global de alimentos está se tornando mais exigente, quanto ao

que consome. Alimentar-se no século XXI tornou-se um constante desafio (LIMA FILHO et al, 2006). Sob a nova economia agrícola, atributos de agricultura intensiva criam novos desafios para a sustentabilidade prática de produção e processamento que promovam um equilíbrio aos problemas de qualidade, segurança e boa qualidade dos alimentos.

Os consumidores modernos exigem alimentos frescos, saborosos, nutritivos e seguros (OPARA, 2002). Portanto, é necessário buscar a qualidade e transparência da produção de alimentos no mercado para buscar transparência nas condições de produção e comercialização

Segundo as informações que compõem a ISO 8402:1994, rastreabilidade é a capacidade de descrever uma história e localizar um produto por meio de informação registrada (FSA – FOOD STANDARD AGENCY, 2002).

A rastreabilidade é uma forma de simplificar a localização de problemas, reduzir o volume de devolução de produtos e estabelecer responsabilidade. A determinação das responsabilidades dos agentes sobre as condições de conformidade dos produtos em cada etapa da produção é uma das funções importantes da rastreabilidade (CHAPAVAL; ALVES, 2013).

Os padrões de consumo de alimentos estão mudando rapidamente hoje em dia, como resultado de questões de desenvolvimento e sustentabilidade, considerações com relação ao seu aspecto nutricional e também questões relacionadas à saúde. Consumir produtos orgânicos por si só não é suficiente para nos tornarmos mais saudáveis, porém, como afirmam Alföldi et al. (2006), produtos orgânicos são parte importante de uma dieta saudável e sustentável.

A rastreabilidade é hoje uma exigência para concretização dos contratos de importação de frutas e hortaliças por vários mercados em todo o mundo (OPARA, 2001).

1.2 BLOCKCHAIN

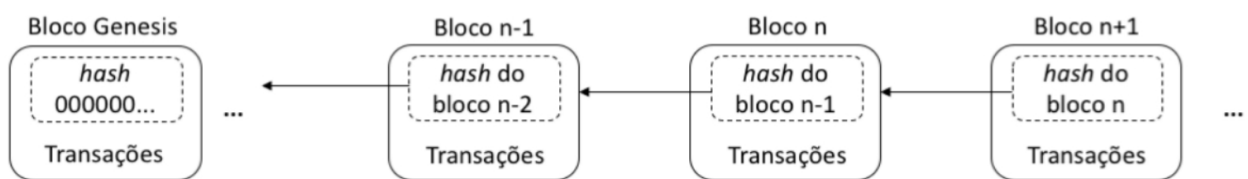
A blockchain é uma tecnologia que simula um livro-razão, que é utilizado na contabilidade das empresas para registrar as transações realizadas por essas. Porém, a blockchain pode ser mantida de forma pública de forma que todos possuem acesso às transações (PECK, 2017).

Satoshi Nakamoto (Nakamoto, 2008) introduziu o Blockchain como

mecanismo para garantir irretratabilidade, auditabilidade, e imutabilidade a fim de prover segurança a transações eletrônicas.

O conceito de Blockchain ou cadeia de dados, ilustrado na Figura 1, nada mais é do que um sistema de consenso distribuído, transparente e verificável. A tecnologia mudou a maneira como as pessoas pensam sobre trocar dinheiro, ações, produtos, serviços, o cumprimento de contratos e o compartilhamento de dados (PEREIRA, 2018).

Figura 1: Representação da cadeia de blocos.



Fonte: OLIVEIRA et al. (2018)

Instrumentos financeiros, como pagamentos, registros e contratos inteligentes podem ser construídos com a tecnologia Blockchain, pois evita comportamentos adversos e problemas, como gastos duplos e falsificações. Além disso, a tecnologia pode ser usada para registros legais e públicos, como títulos, certidão de nascimento, certificados, registros de votos ou judiciais. Também pode ser usado para criação de “propriedade inteligente”, caso em que o Blockchain se torna um mecanismo de inventário, rastreamento e compra de ativos, como diamantes ou carros.

Também pode ser usado para rastrear a cadeia produtiva de suprimentos e alimentos, que é proposta de utilização deste projeto, permitindo que participantes da cadeia possam inserir informações na rede de maneira identificada e abertamente auditável.

1.3 TECNOLOGIAS UTILIZADAS

Neste capítulo serão abordadas as principais tecnologias utilizadas para o desenvolvimento da aplicação, além do Blockchain.

1.3.1 Node.js

O Node.JS foi projetado como um servidor de Internet escalonável. De acordo com seu fabricante (NODE.JS, 2020) é um ambiente de execução Javascript leve e eficiente. Apesar de suportar vários protocolos, devido ao seu framework Express.js, que estende as capacidades do servidor padrão do Node.js adicionando middlewares e outras capacidades como views e rotas (ALMEIDA, 2015), é amplamente usado para construir aplicativos Web, com recursos que suportam protocolo HTTP.

Com o Node.js, os aplicativos Javascript podem ser criados para serem executados como processos independentes no computador, em vez de depender do navegador para executá-los, como no passado, e devido a essa capacidade, foi utilizado nesta aplicação para executar código JavaScript desenvolvido, a nível backend e frontend.

Segundo Pereira (2016), o Node.js possui as seguintes principais características:

- Single-Thread: Suas aplicações serão single-thread, ou seja, cada aplicação terá instância de uma thread principal por processo iniciado. A programação assíncrona e os recursos compartilhados são usados para fazer uso total da thread utilizada;
- Operações de entrada e saída assíncrono e não bloqueante – Facilita execução paralela e o aproveitamento de recursos;
- Event-loop: É orientado a eventos.

1.3.2 Angular

Segundo o site oficial, o Angular (Angular, 2020) é uma estrutura de design de aplicativo e plataforma de desenvolvimento para a criação de aplicativos eficientes e sofisticados de página única.

A linguagem HTML é ótima para declarar documentos estáticos, mas deixa a desejar quando tentamos usá-lo para declarar visualizações dinâmicas em aplicativos da web. Buscando sanar essas adversidades, a utilização do Angular para o desenvolvimento desta aplicação, permite estender o vocabulário HTML para

o desenvolvimento da mesma. Tornando o ambiente resultante, excepcionalmente expressivo, legível e rápido de desenvolver.

Segundo Freeman (2018), aplicações Angular são construídas em torno de um padrão de design chamado Model-View-Controller (MVC), que coloca uma ênfase na criação de aplicação que possuem as seguintes características:

- Extensível: Você pode facilmente aprimorar aplicativos para criar recursos novos e úteis para seus usuários;
- Manutenção: aplicativos desenvolvidos em Angular são fáceis de depurar e corrigir, o que significa que a longo prazo a manutenção é simplificada;
- Testável: o Angular tem bom suporte para testes, o que significa que você pode localizar e corrigir defeitos antes que seus usuários o façam;
- Padronizado: o Angular baseia-se nas capacidades inatas do navegador web sem ficar no seu caminho, permitindo que você crie aplicativos web em conformidade com os padrões que aproveitam os recursos mais recentes (como APIs HTML5) e ferramentas populares e frameworks.

1.3.3 MongoDB

O MongoDB (MONGODB, 2020) é um software de banco de dados orientado a documentos livre, de código aberto e multiplataforma. O MongoDB armazena os dados em documentos flexíveis do tipo JSON, o que significa que os campos podem variar de documento para documento e a estrutura de dados pode ser alterada com o tempo.

Como outros bancos de dados NoSQL, o MongoDB não requer esquemas predefinidos. Ele armazena qualquer tipo de dados. Isso dá aos usuários a flexibilidade de criar qualquer número de campos em um documento, tornando mais fácil escalar bancos de dados MongoDB em comparação com bancos de dados relacionais.

Segundo Rouse (2020), uma das vantagens de usar documentos é que esses objetos mapeiam para tipos de dados nativos em várias linguagens de programação. Além disso, ter documentos incorporados reduz a necessidade de associações de

dados.

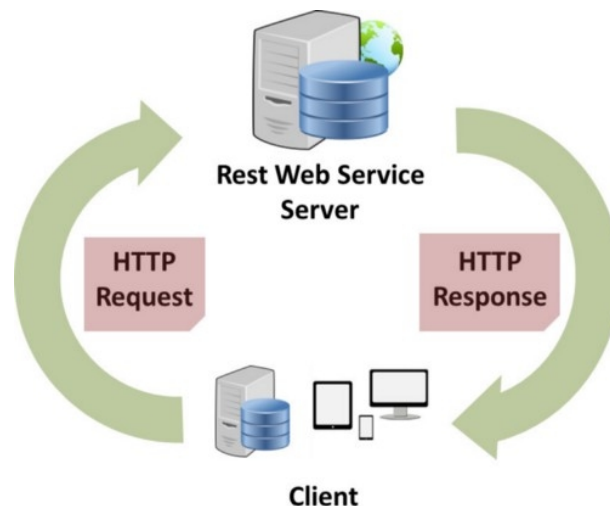
Para o desenvolvimento desta aplicação, é utilizado para guardar os dados de maneira temporária, e também para o armazenamento dos dados no formato de uma Blockchain.

1.3.4 Arquitetura REST

REST significa Representational State Transfer, proposto por Fielding (2000), é descrito como um estilo arquitetural que consiste em um conjunto de restrições aplicadas aos elementos dentro da arquitetura. Utilizado na aplicação como princípios e regras para a criação de um projeto com interfaces bem definidas, determinando como o HTTP e URIs devem ser utilizados.

Segundo Costa (2014), os sistemas REST são menos acoplados, mais leves, eficientes e flexíveis do que os sistemas baseados em outros protocolos Web Services. e podem ser facilmente reutilizados. Além disso, os princípios REST podem ser mapeados nos métodos básicos do protocolo HTTP (GET, POST, UPDATE e DELETE) para criar sistemas CRUD (Create, Read, Update, Delete) de uma aplicação RESTful capaz de aplicar os princípios de REST.

Figura 2: Arquitetura RESTful Web Service



Fonte: Mane (2017)

2 APLICAÇÃO PROPOSTA

A aplicação tem como propósito fazer a gestão dos processos realizados pelo produtor na sua produção de produtos orgânicos.

O produtor tem a possibilidade de separar sua produção em lotes ou produtos, e em cada lote, registra todos os processos realizados na produção, exemplo: adubação verde, irrigação, tratamento do solo, etc, para que seja possível visualizar de forma auditável os procedimentos adotados na produção, e servindo de instrumento para a verificação das etapas da mesma, como por exemplo, os processos realizados na produção dos produtos orgânicos.

Todos os processos registrados pelo produtor, são enviados a uma rede Blockchain, onde se tornam imutáveis e transparentes, assim dando a possibilidade ao consumidor final visualizar todas essas etapas realizadas na produção, garantindo a procedência do produto.

3 DESENVOLVIMENTO DA APLICAÇÃO

Para o desenvolvimento da aplicação foram utilizados: o banco de dados não relacional MongoDB para armazenamento temporário dos registros antes da persistência na rede Blockchain; da plataforma backend NodeJS; do framework Express; do framework Angular para o desenvolvimento frontend; o modelo de arquitetura REST para a integração Web; e da formatação de dados JSON para a comunicação com a API.

A Figura 3 exibe o fluxo de operações que envolveram tal aplicação. Para detalhar o processo, os componentes necessários para o funcionamento da aplicação foram ilustrados de forma simplificada. Dessa forma, os dados recebidos da aplicação por meio de requisições HTTP (Sequência 1), são tratados por uma API que possui um conjunto de definições e protocolos para a comunicação desses dados. O padrão REST é utilizado para que as requisições HTTP atendam as diretrizes definidas na arquitetura.

Após o tratamento dos dados, é feita a persistência destes(Sequência 2), utilizando o banco de dados MongoDB. Esses dados são armazenados de maneira temporária, para que então sejam inseridos de forma definitiva e imutáveis na rede Blockchain (Sequência 3).

Figura 3: Fluxo de operações

Fonte: Autor

3.1 MODELAGEM DA APLICAÇÃO

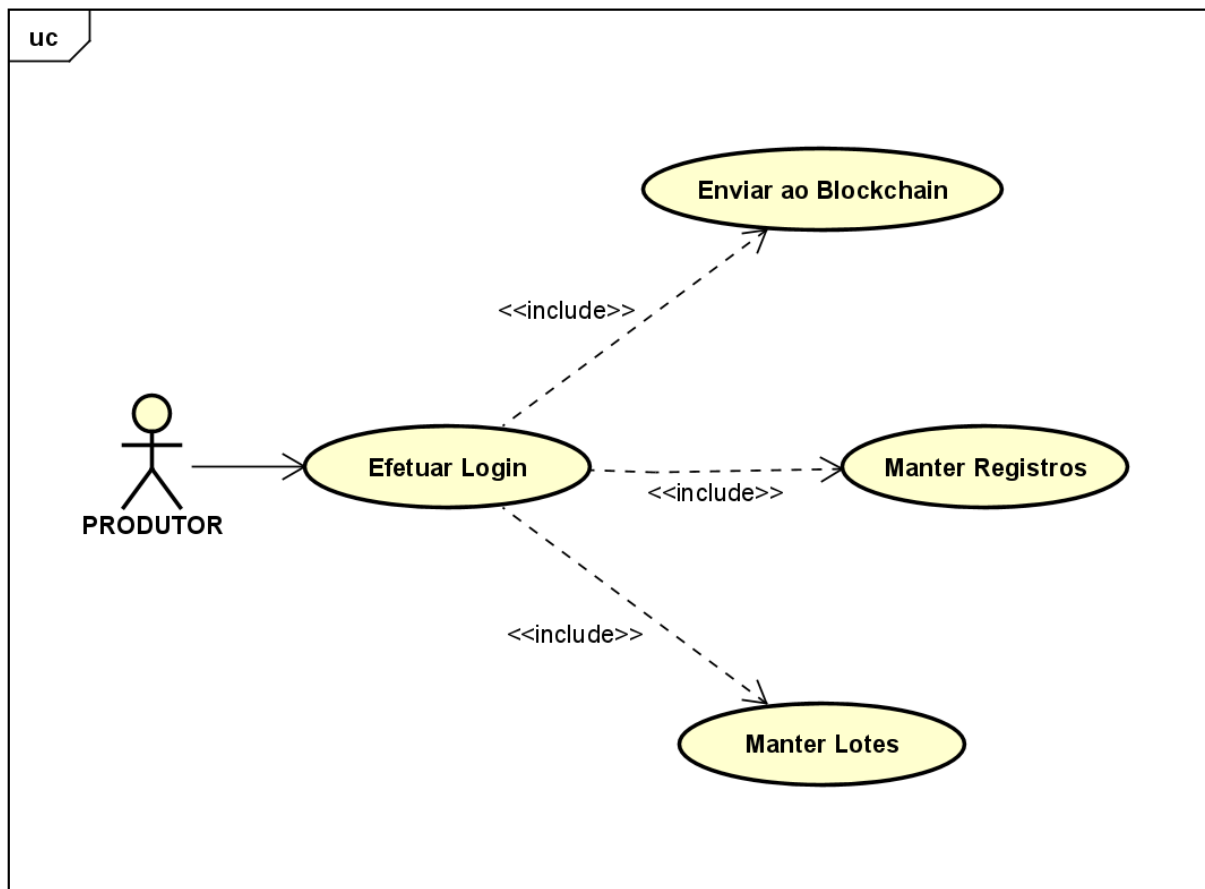
Nesta seção será demonstrado através de diagramas como será o fluxo de funcionamento do aplicativo, bem como uma perspectiva das principais funcionalidades que a aplicação terá para que ela funcione adequadamente.

3.1.1 Diagrama de Casos de Uso

O Diagrama de Casos de Uso tem como principal objetivo demonstrar o fluxo de funcionamento da aplicação através do ponto de vista do usuário, e isto fornece uma visão geral do funcionamento do sistema, além explicar as principais funcionalidades que o sistema deve disponibilizar para os usuários.

O diagrama apresentado na Figura 4 demonstra as principais funcionalidades que a aplicação proverá aos usuários envolvidos que utilizarão o sistema.

Figura 4: Diagrama de Casos de Uso



Fonte: Autor

3.1.2 Descrição dos Casos de Uso

A descrição dos casos de uso tem por objetivo fornecer um panorama detalhado das principais funcionalidades que o sistema fornecerá aos atores através dos casos de uso que foram desenvolvidos no diagrama da figura 3.

Produtor:

- *Enviar ao Blockchain*: O Produtor faz o envio dos registros feitos para a rede Blockchain.
- *Manter Registros*: O Produtor faz o registro com todos os processos da produção.
- *Manter Lotes*: O Produtor insere os lotes ou produto que serão registrados os processos feitos na produção.
- *Efetuar Login*: O Produtor realiza o login com seus dados para ter acesso à aplicação.

3.2 CARACTERÍSTICAS E FUNCIONALIDADES

Esta seção descreve as funcionalidades implementadas na aplicação, mostrando quais recursos e tecnologias foram utilizados.

3.2.1 Autenticação dos usuários

Este recurso foi desenvolvido fazendo uso do padrão JSON Web Token. É através desse padrão que se realiza o gerenciamento dos usuários cadastrados e o cadastro de novos usuários na aplicação através de e-mail e senha.

O usuário enviará uma requisição HTTP de autenticação para a API. Nela o usuário envia, no corpo da requisição dados como endereço de e-mail e senha.

Uma vez que os dados enviados pelo usuário tenham sido autenticados no servidor, este criará um token JWT assinado com um segredo interno da API e enviará este token de volta ao usuário

Figura 5: Tela de login



Fonte: Autor

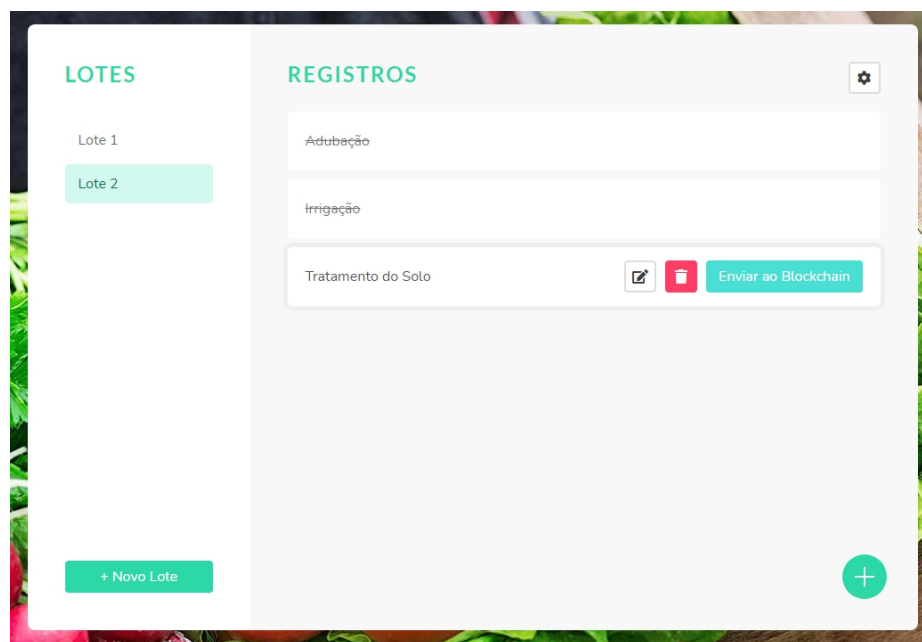
Provido com o token autenticado, o usuário possui acesso a recursos da aplicação que antes eram restritos. Para realizar esse acesso, conforme mostra a Figura 5 é preciso informar esse token no header Authorization da requisição.

3.2.2 Inserção de Lotes e Processos

O usuário tem a capacidade de fazer a inserção, edição e exclusão de Lotes, também de registrar, editar e excluir os processos realizados em cada Lote. Conforme mostra a Figura 6, os botões são apresentados quando o usuário passa com o cursor em cima do processo desejado para fazer a edição ou exclusão. Esses dados são inseridos de forma temporária em um banco de dados criado usando o MongoDB.

O botão “Enviar ao Blockchain” tem a função de enviar o processo registrado para a rede Blockchain de forma definitiva, já que depois de inserido na Blockchain, devido ser um banco criptografado e referenciado com o seu hash, que garante que esses dados não foram alterados, não pode ser mais modificado ou excluído.

Figura 6: Tela Principal



Fonte: Autor

3.2.3 Envio ao Blockchain

A rede Blockchain foi implementada seguindo o modelo proposto por Maboud (2020), desenvolvendo uma estrutura Blockchain utilizando o banco de dados MongoDB, adaptada para uma API capaz de receber requisições REST, e a

inserção dos dados usando a biblioteca de modelagem de dados de objeto (ODM) Mongoose, que fornece um ambiente de modelagem para seus dados, reforçando a estrutura conforme necessário, mantendo a flexibilidade tornando o MongoDB mais eficaz.

O Quadro 1 mostra o esquema de dados criado para ser feito a inserção na rede Blockchain.

Quadro 1: Modelo Blockchain MongoDB

```
const BlockchainSchema = new mongoose.Schema({  
  
  index: {  
    required: true,  
    type: Schema.Types.Number  
  },  
  timestamp: {  
    required: true,  
    type: Schema.Types.Date,  
    default: Date.now(),  
  },  
  lote: {  
    required: true,  
    type: Schema.Types.String,  
  },  
  processo: {  
    required: true,  
    type: Schema.Types.String,  
  },  
  prevHash: {  
    required: true,  
    type: Schema.Types.String,  
  },  
  hash: {  
    required: true,  
    type: Schema.Types.String,  
  }  
})
```

Fonte: Autor

Após ser feito o evento de clique no botão “Enviar ao Blockchain” descrito na seção anterior, é chamada a função “criarBlockchain”, passando o processo como parâmetro, conforme mostra o quadro 2.

Quadro 2: Função “criarBlockchain”

```
criarBlockchain(processo: string) {
  this.controladorService.criarBlockchain(processo,
  this.selecionaLoteld).subscribe((blockchain: Blockchain) => {
  },
  (err) => {console.log("err")})
}}
```

Fonte: Autor

Essa função tem o objetivo de enviar ao controlador o processo e o id do lote que pertence esse processo. O controlador recebe esses dados e faz uma requisição para o web service para serem inseridos do na Blockchain, conforme mostra o Quadro 3.

Quadro 3: Requisição Web Service

```
criarBlockchain(processo: string, selecionaLoteld: string) {
  return this.webReqService.postBlockchain('blockchain', { processo , selecionaLoteld });
}
```

Fonte: Autor

O controlador recebe essa requisição e então utilizando o protocolo HTTP, utiliza do método POST, se comunica com a API através de requisições REST, e passa como parâmetro o identificador de recurso e o objeto com os dados, conforme mostra no Quadro 4.

Quadro 4 : Comunicação API e Aplicação

```
constructor(private http: HttpClient) {
  this.ROOT_URL = 'http://localhost:3001';
}
postBlockchain(uri: string, payload: Object) {
  return this.http.post(`${this.ROOT_URL}/${uri}`, payload);
}
```

Fonte: Autor

A API recebe essa requisição POST, busca os dados passados como parâmetro e cria a variável “block” sendo o bloco a ser inserido na Blockchain, seguindo o esquema de dados mostrado na Figura 7. Então gera um hash para o bloco criado utilizando a biblioteca object-hash, e também busca o hash do bloco

anterior, assim um bloco dependendo do outro e formando uma cadeia de blocos.

Se tudo estiver de acordo, esse bloco é inserido na rede Blockchain, conforme mostra o quadro 5.

Quadro 5: Inserção do bloco na Blockchain

```
app.post('/blockchain', (req, res) => {
  let prevHash = 0;
  let index = 0;

  Blockchain.findOne().sort({ _id: -1 }).limit(1).exec((err, data) => {
    prevHash = data.hash;
    index = data.index + 1;

    let block = {
      index: index,
      timestamp: Date.now(),
      lote: req.body.selecionaLoteld,
      processo: req.body.processo,
      prevHash: prevHash,
      hash: hash(prevHash)
    };

    //Add a instância e Salvar no DB
    let newBlock = new Blockchain(block);
    newBlock.save((err) => {
      if (err) return console.log(chalk.red("Erro ao salvar BD ", err.message));
      console.log(chalk.green("Bloco salvo no BD!"));
    });

    console.log(block);

  });
});
```

Fonte: Autor

Figura 7: Exemplo de dados salvos na Blockchain

```

_id: ObjectId("5fc146924678d0221ce5038d")
timestamp: 2020-11-27T18:33:54.625+00:00
index: 0
lote: "5fb6bc7e710eca25c47b5f0b"
processo: "GENESIS"
prevHash: null
hash: "63d0b026e716c2f15878d2b9d7ff673afe0f9916"
__v: 0

```

```

_id: ObjectId("5fc146e8f3637d00bce4ad9e")
timestamp: 2020-11-27T18:35:20.554+00:00
index: 1
lote: "5fb6bc7e710eca25c47b5f0b"
processo: "Irrigação"
prevHash: "63d0b026e716c2f15878d2b9d7ff673afe0f9916"
hash: "37f0eb124eb134bf921797edc1d6a75291dccbf7"
__v: 0

```

```

_id: ObjectId("5fc146ebf3637d00bce4ad9f")
timestamp: 2020-11-27T18:35:23.697+00:00
index: 2
lote: "5fb6bc7e710eca25c47b5f0b"
processo: "Adubação"
prevHash: "37f0eb124eb134bf921797edc1d6a75291dccbf7"
hash: "ee6bd726a588810e9d7faebca50a40ab5c69b429"
__v: 0

```

Fonte: Autor

4 CONSIDERAÇÕES FINAIS

Utilizando a tecnologia Blockchain como forma de auxiliar na segurança da rastreabilidade da cadeia produtiva de alimentos, tendo como cadeia produtiva e produção de produtos orgânicos como exemplo, podemos concluir que o Blockchain se torna eficiente em consequência da sua capacidade de auditabilidade, transparência e imutabilidade devido a sua estrutura de designação de códigos hash, encadeando todos os blocos da cadeia.

Outra alternativa de empregar o Blockchain como um serviço de armazenamento, é utilizar algum dos serviços de Blockchain em nuvem disponíveis no mercado. Mas devido ao alto preço desses serviços e dificuldade de integração com uma aplicação web service, a criação de uma estrutura de Blockchain utilizando um banco de dados não relacional, adaptada a inserção dos dados com uma API REST desenvolvida para fazer a comunicação da aplicação com a estrutura

Blockchain.

Para trabalhos futuros abre-se um leque de possibilidades que vem ao encontro com este tema, como: geração de QRCode para a visualização dos processos com mais facilidade para o consumidor final, diferente aplicabilidade da aplicação em outros meios que a rastreabilidade possa ser uma solução, etc.

ABSTRACT

This work presents the study of the use of the architecture and protocol of Blockchain technology, as a possible solution to practical sustainability problems of production and processing that promote a balance to the problems of quality, safety and good quality of food. Seeking to relate Blockchain technology to the traceability of the food production chain, an application will be developed for the management and control of processes carried out in the production of organic products. As a result, avoid a major advantage in using Blockchain as a data storage medium, due to its great capacity for transparency, immutability and security.

Keywords: Blockchain. Organic products.

REFERÊNCIAS

ALMEIDA, Flávio. (2015). Mean Full stack JavaScript para aplicações web com MongoDB, Express, Angular e Node. Disponível em: <https://docero.com.br/doc/ses801>. Acesso em: 30 de novembro. 2020

ALFÖLDI, T.; GRANADO, J.; KIEFFER, E.; KRETZSCHMAR, U.; MORGNER, M.; NIGGLI, U.; SCHÄDELI, A.; SPEISER, B.; WEIBEL, F.; WYSS, G.; SCHMIDT, W.; SCHMIDT, G. 2006. Quality and safety of organic products: Food systems compared. Disponível em: https://www.researchgate.net/publication/232688680_Organic_Food_Quality_and_Safety_Perception_Throughout_Europe_Organic_Food_Quality_and_Safety_Perception. Acesso em: 10 de dezembro. 2020.

CHAPAVAL, L.; ALVES, F. S. F. (2008). Rastreabilidade na produção de leite de cabra: diferencial para um agronegócio sustentável. EMBRAPA Caprinos. Disponível em: <https://www.infoteca.cnptia.embrapa.br/infoteca/bitstream/doc/533940/1/MidiaRastreabilidadeNaProducaoDeLeitedeCabrapdf>. Acesso em: 24 de novembro. 2020.

COSTA, B. (2014). Avaliando uma Arquitetura Baseada no Estilo REST. Universidade Federal do Rio de Janeiro. Disponível em: <http://objdig.ufrj.br/15/teses/826945.pdf>. Acesso em: 27 de novembro. 2020.

FOOD STANDARDS AGENCY. (2002). Traceability in the food chain: a preliminary study. Food Chain Strategy Division, Food Standards Agency. Disponível em: https://www.adiveter.com/ftp_public/articulo361.pdf. Acesso em: 27 de novembro. 2020.

FREEMAN, Adam. Pro AngularJS 6. (2018). Disponível em: <https://www.pdfdrive.com/pro-angular-6-e176207319.html>. Acesso em: 22 de novembro. 2020.

LIMA FILHO, D. de O et al. (2006) O consumo de alimentos e a percepção de risco. XXVI Encontro Nacional de Engenharia de Produção (ENEGEP). Disponível em: http://www.abepro.org.br/biblioteca/enegep2006_TR530357_7055.pdf. Acesso em: 20 de novembro. 2020.

MABOUD, Islem. (2020). Let's Create A Blockchain On Node.js. Disponível em: <https://ipenywis.com/lets-create-a-blockchain-on-node-js-a326016cada5>. Acesso em: 03 de novembro. 2020.

MANE, Sagar. (2017). Understanding REST (Representational State Transfer). Disponível em: <https://medium.com/@sagar.mane006/understanding-rest-representational-state-transfer-85256b9424aa>. Acesso em: 30 de novembro. 2020.

MONGODB (2020). MongoDB. Disponível em: <https://www.mongodb.com/>. Acesso em: 20 de novembro. 2020.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. Disponível em: <https://bitcoin.org/bitcoin.pdf>. Acesso em: 24 de novembro. 2020.

OLIVEIRA, M. T. et al. (2018). Uma avaliação de desempenho de cadeias de blocos privadas permissionadas através de cargas de trabalho realísticas Disponível em: <https://sol.sbc.org.br/index.php/sbseg/article/view/4261>. Acesso em: 25 de novembro. 2020.

OPARA LU; MAZEAUD F. (2001). Food traceability from field to plate. Disponível em: https://www.researchgate.net/publication/233683672_Food_traceability_from_field_to_plate. Acesso em: 20 de novembro. 2020.

PECK, M. E. (2017). Blockchains: How they work and why they'll change the world. Disponível em: <https://spectrum.ieee.org/computing/networks/blockchains-how-they-work-and-why-theyll-change-the-world>. Acesso em: 24 de novembro. 2020.

PEREIRA, C. R. (2016). Construindo APIs Rest com Node.js. 1. ed. São Paulo: Casa do Código. Disponível em: <https://docero.com.br/doc/18exec>. Acesso em: 24 de novembro. 2020.

ROUSE, Margaret. MongoDB. Disponível em: <https://searchdatamanagement.techtarget.com/definition/MongoDB>. Acesso em: 24 de novembro. 2020.