

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

ANDERSON DOMENEGUETTE FELIPPE

**MODELO DE RASTREABILIDADE VINCULADO AO DNA PARA A CADEIA DA
CARNE BOVINA BASEADO EM *BLOCKCHAIN* E *SMART CONTRACTS***

CAMPINAS

2020

ANDERSON DOMENEGUETTE FELIPPE

**MODELO DE RASTREABILIDADE VINCULADO AO DNA PARA A CADEIA DA
CARNE BOVINA BASEADO EM *BLOCKCHAIN* E *SMART CONTRACTS***

Dissertação apresentada como exigência para obtenção do título de Mestre em Gestão de Redes de Telecomunicações ao Programa de Pós-Graduação em Engenharia Elétrica do Centro de Ciências Exatas, Ambientais e de Tecnologias.
Área de Concentração: Engenharia Elétrica.
Orientador(a): Prof. Dr. Juan Manuel Adan Coello

PUC-CAMPINAS

2020

Ficha catalográfica elaborada por Vanessa da Silveira CRB 8/8423
Sistema de Bibliotecas e Informação - SBI - PUC-Campinas

621.3 Felipe, Anderson Domeneguetto

F319m

Modelo de rastreabilidade vinculado ao DNA para a cadeia da carne bovina baseado em Blockchain e Smart Contracts / Anderson Domeneguetto Felipe. - Campinas: PUC-Campinas, 2021.

94 f.: il.

Orientador: Juan Manuel Adan Coello.

Dissertação (Mestrado em Gestão de Rede de Telecomunicações) - Programa de Pós-Graduação em Engenharia Elétrica, Centro de Ciências Exatas, Ambientais e de Tecnologia, Pontifícia Universidade Católica de Campinas, Campinas, 2021.

Inclui bibliografia.

1. Telecomunicações . 2. Blockchains (Base de dados). 3. Carne bovina. I. Coello, Juan Manuel Adan . II. Pontifícia Universidade Católica de Campinas. Centro de Ciências Exatas, Ambientais e de Tecnologia. Programa de Pós-Graduação em Engenharia Elétrica. III. Título.

CDD - 22. ed. 621.3

ANDERSON DOMENEGUETTE FELIPPE

**MODELO DE RASTREABILIDADE VINCULADO AO
DNA PARA A CADEIA DA CARNE BOVINA
BASEADO EM BLOCKCHAIN E SMART CONTRACTS**

Dissertação apresentada como exigência para obtenção do título de Mestre em Gestão de Redes de Telecomunicações ao Programa de Pós-Graduação em Engenharia Elétrica do Centro de Ciências Exatas, Ambientais e de Tecnologias.
Área de Concentração: Engenharia Elétrica.
Orientador (a): Prof. Dr. Juan Manuel Adan Coello

Dissertação defendida e aprovada em 17 de dezembro de 2020 pela Comissão Examinadora constituída dos seguintes professores:



Prof. Dr. Juan Manuel Adán Coello
Orientador da Dissertação e Presidente da Comissão Examinadora
Pontifícia Universidade Católica de Campinas



Prof. Dr. Frank Herman Behrens
Pontifícia Universidade Católica de Campinas



Prof. Dr. Wilfredo Jaime Puma Villanueva
Robert Bosch Ltda.

AGRADECIMENTOS

O presente trabalho foi realizado com o apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

This study was financed in party by Coordination and Improvement of Higher Level or Education Personnel – Brazil (CAPES) - Finance Code 001.

À PUC-Campinas pela infraestrutura concedida e o fornecimento dos equipamentos para a realização dos ensaios.

Ao meu orientador, Prof. Dr. Juan Manuel Adán Coello, que, com um conhecimento técnico e acadêmico extraordinário, forneceu apoio e orientação para a finalização deste trabalho com muita qualidade.

Ao Prof. Dr. Antonio Carlos Demanboro pela orientação inicial e auxílio na transformação de uma ideia bruta em um projeto de pesquisa com um objetivo definido.

Aos meus colegas de trabalho que me ajudaram a compreender e admirar a pecuária ao enxergar sua dimensão e importância na sociedade, sempre buscando soluções que possam melhorar a vida das pessoas.

À minha esposa, Milena Dias de Paula, pela minuciosa revisão do texto da dissertação e, principalmente, pela compreensão e suporte em mais uma difícil jornada em nossas vidas.

RESUMO

FELIPPE, Anderson Domeneguetto. **Modelo de rastreabilidade vinculado ao DNA para a cadeia da carne bovina baseado em *Blockchain* e *Smart Contracts***. 2020. Dissertação (Mestrado). Programa de Pós-Graduação em Gestão de Redes de Telecomunicações, Pontifícia Universidade Católica de Campinas, Campinas, 2020.

Como consequência do incessante crescimento populacional, a demanda por alimento no mundo enfrentará diversos desafios nos próximos anos, como rastreabilidade, garantia de procedência e qualidade. Em especial, destaca-se a cadeia da carne bovina, que continuará crescendo através das décadas e sofrerá um forte impacto da demanda global de alimentos. Este trabalho tem como foco analisar os problemas presentes na cadeia de produção da carne bovina e suas inúmeras fragilidades em prover uma alternativa confiável para rastrear o produto de maneira completa através de todas as etapas do processo logístico. Como solução para os problemas apresentados, é proposta a utilização da tecnologia de *Smart Contracts* vinculada às redes de *Blockchain*, as quais têm sido largamente utilizadas em soluções logísticas e em diversos outros mercados similares. Este trabalho apresenta o algoritmo do *Smart Contract* utilizado na solução e também o desenho detalhado do modelo de aplicação em uma rede *Blockchain* para que seja possível desenvolver um sistema de rastreabilidade imutável, inviolável e de fácil aplicação. A implementação prática de um estudo de caso do modelo proposto em uma rede *Blockchain* com um *Smart Contract* e a análise dos seus impactos técnicos mostraram que, por meio do acompanhamento de animais e produtos criados na rede com base no DNA, é factível a aplicação da tecnologia para obter uma rastreabilidade completa, imutável e padronizada para toda a cadeia da carne bovina.

Palavras-chaves: *Blockchain*, *Smart Contracts*, Cadeia de Suprimentos, Carne Bovina.

ABSTRACT

FELIPPE, Anderson Domeneguetto. **DNA-linked traceability model for the beef supply chain based on Blockchain and Smart Contracts**. 2020. Master's Thesis. Postgraduate Program in Telecommunications Network Management, Pontifical Catholic University of Campinas, Campinas, 2020.

Due to the continuous world population growth, world's demand for food will face a variety of challenges in the upcoming years, such as traceability, source reliability and quality. The beef supply chain will be in the spotlight as it keeps growing over decades and will suffer a strong impact of high global food demand. This work focuses on analyzing current issues faced by the beef supply chain and its various weaknesses to provide a reliable way to fully trace the product throughout the logistics involved in the process. As a solution to the presented issues regarding the beef supply chain, this work proposes the application of Smart Contracts technology running on Blockchain networks, which have been largely used in solutions for logistics and many other similar markets. This work presents the Smart Contracts algorithm used in the solution as well as the design and details of the application model in a Blockchain network in order to build a traceability system that is immutable, unbreakable and easy to deploy. The practical implementation of a case study of the proposed model in a Blockchain network alongside a Smart Contract and the analysis of its technical impacts showed that, by tracing animals and products created in the network and linked to DNA, the application of the technology is feasible in order to obtain complete, immutable and standardized traceability to the entire beef supply chain.

Keywords: Blockchain, Smart Contracts, Supply Chain, Beef.

LISTA DE SIGLAS E ABREVIações

ABIEC – Associação Brasileira das Indústrias Exportadoras de Carnes

API – *Application Programming Interface*

BAC – *Batch Addition Contract*

CPU – *Central Processing Unit*

DApp – *Decentralized App*

DNA – *Deoxyribonucleic Acid*

DPOS – *Delegated Proof-of-Stake*

ECDSA – *Elliptic Curve Digital Signature Algorithm*

EUA – Estados Unidos da América

FAO – *Food and Agriculture Organization of the United Nations*

GB – *Gigabytes*

GTA – Guia de Trânsito Animal

HD – *Hard Disk*

IBOPE – Instituto Brasileiro de Opinião Pública e Estatística

ID – Identificação

IoT – *Internet of Things*

ISO – *International Organization for Standardization*

KB – *Kilobytes*

LF – *Low Frequency*

MAPA – Ministério da Agricultura, Pecuária e Abastecimento

MB – *Megabytes*

ONU – Organização das Nações Unidas

P2P – *Peer-to-Peer*

PBFT – *Practical Byzantine Fault Tolerance*

PCR – *Polymerase Chain Reaction*

POA – *Proof-of-Authority*

POAC – *Proof-of-Activity*

POB – *Proof-of-Burn*

POC – *Proof-of-Capacity*

POET – *Proof-of-Elapsed Time*

POI – *Proof-of-Importance*

POL – *Proof-of-Luck*

POS – *Proof-of-Stake*

POT – *Proof-of-Trust*

POV – *Proof-of-Vote*

POW – *Proof-of-Work*

POWEIGHT – *Proof-of-Weight*

PRC – *Product Registration Contract*

RAM – *Random Access Memory*

RFID – *Radio-frequency Identification*

SISBOV – *Sistema Brasileiro de Identificação Individual de Bovinos e Búfalos*

SQL – *Structured Query Language*

SSD – *Solid State Drive*

TB – *Terabytes*

TI – *Tecnologia da Informação*

TUC – *Transaction Update Contract*

UHF – *Ultra High Frequency*

USDA – *United States Department of Agriculture*

LISTA DE FIGURAS

Figura 1 – Crescimento populacional em bilhões de pessoas por ano	15
Figura 2 – Crescimento anual da produção mundial de carne bovina em milhões de toneladas.....	16
Figura 3 – Exemplo de brincos LF e UHF	18
Figura 4 – <i>Tag</i> RFID subcutânea	18
Figura 5 – Crescimento da utilização do termo rastreabilidade em artigos	20
Figura 6 – Visão geral da cadeia de suprimento da carne bovina.....	21
Figura 7 – Sistema de rastreabilidade para a cadeia de alimentos de origem agrícola baseado em RFID e <i>Blockchain</i>	24
Figura 8 – Modelo de rastreabilidade de salmão norueguês.....	25
Figura 9 – Estrutura geral do <i>Smart Contract</i> proposto por Yano et al. (2018)	27
Figura 10 – Aplicação de uma função de <i>hash</i> de 160 bits para criar uma saída única.....	28
Figura 11 – Exemplo de uma cadeia de blocos na <i>Blockchain</i>	30
Figura 12 – Fluxo da transação dentro da rede de <i>Blockchain</i>	31
Figura 13 – Fluxo de criação das chaves criptografadas no <i>Bitcoin</i>	33
Figura 14 – Aplicação de funções criptográficas de mão única.....	33
Figura 15 – Rastreabilidade de grãos de soja utilizando <i>Smart Contracts</i>	40
Figura 16 – <i>Smart Contract</i> e a interação com os atores e a <i>Blockchain</i>	42
Figura 17 – Atores da cadeia da carne bovina na rede <i>Blockchain</i>	43
Figura 18 – Modelagem dos bens e seu relacionamento com os atores da cadeia da carne bovina.....	46
Figura 19 – Amostras de sangue e pelo de um animal armazenadas em papéis com filtro especial para conservação de amostras genéticas	49
Figura 20 – Exemplo de conversão do resultado do número de alelos presentes para cada marcador de microssatélite em um <i>fingerprint</i> genético	51
Figura 21 – Estrutura organizacional do <i>Hyperledger Composer</i>	58
Figura 22 – Ambiente de configuração e definição dos arquivos do <i>Hyperledger Composer</i>	59
Figura 23 – Ambiente de testes da <i>Blockchain</i> do <i>Hyperledger Fabric</i> com os atores da cadeia da carne, <i>assets</i> e botão de submissão de transações	60

Figura 24 – Ferramenta de submissão das transações à <i>Blockchain</i> no ambiente de testes do <i>Hyperledger Composer</i>	62
Figura 25 – Registro do animal na <i>Blockchain</i> e sua visualização com as informações atuais do menu “Animal” na ferramenta	63
Figura 26 – Informações da transação de atualização do animal	64
Figura 27 – Animal já atualizado na <i>Blockchain</i> com as informações enviadas pela transação de atualização	64
Figura 28 – Animal já atualizado com as informações na etapa de engorda	65
Figura 29 – Dados do animal já registrados na <i>Blockchain</i> após o abate	65
Figura 30 – Transação de criação do produto vinculado à identificação genética de um animal já existente na <i>Blockchain</i>	66
Figura 31 – Dados do produto já registrados na <i>Blockchain</i>	67
Figura 32 – Produto à venda no varejo derivado da peça maior com as informações atualizadas	67
Figura 33 – Registros históricos das transações na <i>Blockchain</i>	68
Figura 34 – Opção de visualização dos detalhes das transações na <i>Blockchain</i>	68
Figura 35 – Registro histórico de uma transação	69
Figura 36 – Tamanho do banco de dados em relação ao número de registros inseridos na <i>Blockchain</i>	70

LISTA DE TABELAS

Tabela 1 – Percentual de exportação de carne bovina por país	16
Tabela 2 – Probabilidade de colisão, em anos.....	29
Tabela 3 – Resumo dos mecanismos de consenso	38
Tabela 4 – Esquema de registro de identificação individual dos participantes (atores) da cadeia da carne bovina na <i>Blockchain</i>	44
Tabela 5 – Definição das enumerações do status dos bens	46
Tabela 6 – Quantidade máxima de alelos observados por microssatélite.....	50

LISTA DE PSEUDOCÓDIGOS

Pseudocódigo 1 – Função de transação de criação de um ator da cadeia da carne bovina na <i>Blockchain</i>	52
Pseudocódigo 2 – Função de transação de criação de um animal na <i>Blockchain</i>	53
Pseudocódigo 3 – Função de transação de atualização de um animal existente na <i>Blockchain</i>	54
Pseudocódigo 4 – Função de transação de criação de um produto na <i>Blockchain</i> ..	55
Pseudocódigo 5 – Função de transação de atualização de um produto existente na <i>Blockchain</i>	56

LISTA DE APÊNDICES

Apêndice A – Código-fonte do <i>Smart Contract</i> em <i>JavaScript</i> utilizado no estudo de caso feito com o <i>Hyperledger Fabric</i>	84
Apêndice B – Código-fonte do modelo dos participantes e <i>assets</i> da rede <i>Blockchain</i> no <i>Hyperledger Fabric</i>	90
Apêndice C – Código-fonte do <i>script</i> em <i>Python</i> para inserção automática de animais na <i>Blockchain</i>	93
Apêndice D – Link para o repositório no <i>Github</i> com o projeto completo da <i>Blockchain</i> com o <i>Hyperledger Fabric</i> e instruções de instalação.....	94

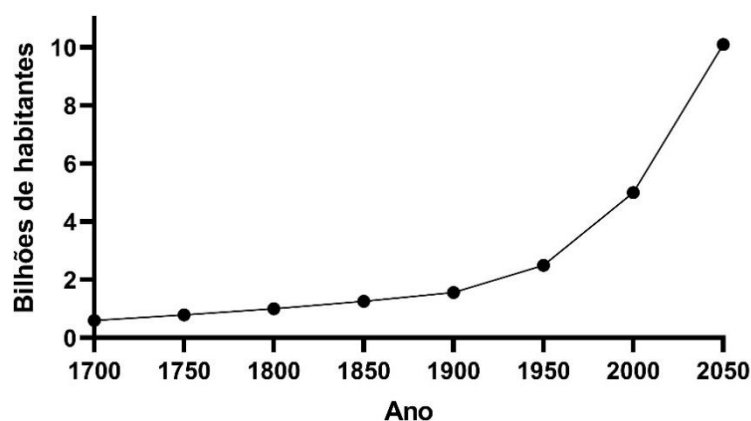
SUMÁRIO

1. INTRODUÇÃO	15
2. REVISÃO BIBLIOGRÁFICA.....	20
2.1 A rastreabilidade na cadeia de suprimento da carne bovina.....	20
2.2 <i>Blockchain</i> como solução de rastreabilidade em cadeias de suprimentos e logística.....	22
2.3 <i>Blockchain</i> como ferramenta na rastreabilidade da carne bovina	26
2.4 A tecnologia de <i>Blockchain</i>	28
2.5 Segurança da tecnologia <i>Blockchain</i>	32
2.6 <i>Smart Contracts</i>	39
3. MODELO DE RASTREABILIDADE PARA A CADEIA DA CARNE BOVINA	41
3.1 Definição dos atores e da rede <i>Blockchain</i>	42
3.2 Definição dos bens rastreáveis	45
3.3 O DNA animal como identificação inicial obrigatória	47
3.4 O <i>Smart Contract</i>	51
3.5 Protótipo funcional de um sistema de rastreabilidade baseado em <i>Blockchain</i>	57
4. VALIDAÇÃO DO MODELO PROPOSTO.....	61
4.1 Estudo de caso do modelo.....	61
4.2 Impactos do tamanho do banco de dados	69
4.3 Impactos financeiros	71
5. ANÁLISE E CONCLUSÃO	73
REFERÊNCIAS BIBLIOGRÁFICAS	76
APÊNDICES.....	84

1. INTRODUÇÃO

Nos mais recentes prospectos de crescimento populacional publicados no *World Population Prospects*, com uma probabilidade de acerto de 95%, a população mundial deve atingir a marca de 8,6 bilhões de pessoas até 2030 e de 10,1 bilhões até 2050 (Figura 1), quando se estima que, como consequência, a demanda por alimentos deve dobrar (ONU, 2019). Diante desse desafio, a produção de alimentos de origem animal é vista como um assunto central, principalmente devido ao seu alto impacto ambiental e à avaliação de que sua demanda pode se tornar insustentável (BAKKER; DAGEVOS, 2011).

Figura 1 – Crescimento populacional em bilhões de pessoas por ano

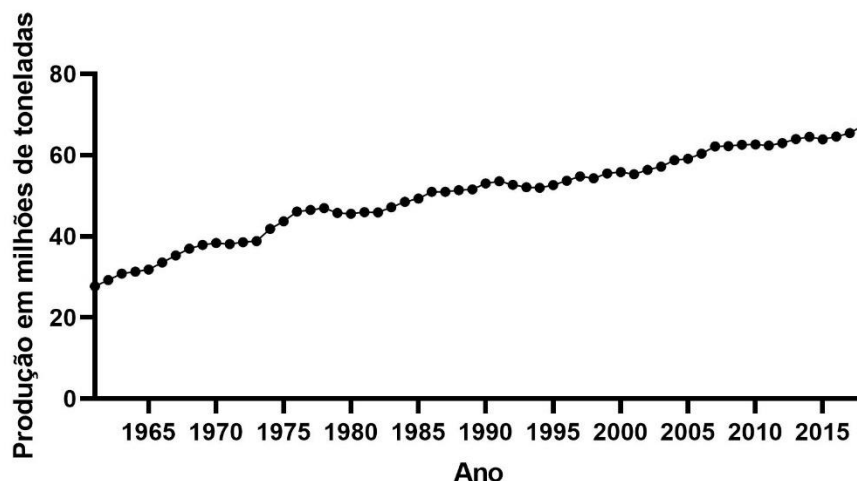


Fonte: Adaptado de ONU (2019)

Apesar de haver uma tendência ao aumento da parcela da população que mantém uma dieta livre de proteína ou produtos de origem animal (dietas veganas ou vegetarianas), como mostrou uma pesquisa do IBOPE (2018) e foi apontado por Valle (2018), bem como diversas iniciativas para integrar os insetos como uma fonte viável de proteína animal (CASTRO et al., 2018), a carne bovina continua sendo a protagonista na mesa da população e, nos próximos quatro anos, estima-se que o seu consumo doméstico deva aumentar até 16%, impulsionando também a demanda por exportação de carne dos grandes produtores. Somente no Brasil, o maior exportador de carne bovina do mundo (Tabela 1), o aumento da demanda de exportação pode chegar a 40% até 2024 (SAATH; FACHINELLO, 2018). Vale também ressaltar que,

do início da década de 60 até o ano de 2018, a produção mundial de carne bovina já havia mais do que dobrado (Figura 2).

Figura 2 – Crescimento anual da produção mundial de carne bovina em milhões de toneladas



Fonte: Food and Agriculture Organization of the United Nations (FAO)¹

Tabela 1 – Percentual de exportação de carne bovina por país

País	Milhões de ton.	Participação (%)
Brasil	2,02	19,29
Índia	1,90	18,15
Austrália	1,61	15,38
EUA	1,37	13,09
Nova Zelândia	0,56	5,35
Canadá	0,48	4,58
Uruguai	0,42	4,01
Paraguai	0,40	3,82
União Europeia	0,37	3,53
Argentina	0,35	3,34

Fonte: Dados do USDA²

Conforme mostra Baruselli (2019), mesmo com um número tão expressivo de exportação e tendo hoje o maior rebanho de gado do mundo, o Brasil ainda tem desempenho inferior ao dos EUA na produção de quilos de carne por animal/ano. Os EUA, com aproximadamente 89 milhões de cabeças de gado, detêm uma produção

¹ Adaptado por Our World in Data. Disponível em: <<https://ourworldindata.org/meat-and-seafood-production-consumption>>. Acesso em: 28 abr. 2019.

² Adaptado pelo autor com base na adaptação de farmnews. Disponível em: <<http://www.farmnews.com.br/mercado/principais-exportadores-de-carne-bovina-2/>>. Acesso em: 27 abr. 2019.

média de 133,2 kg de carne por animal/ano, contra apenas 45,8 kg de carne por animal/ano no Brasil (BARUSELLI, 2019). Isso ocorre principalmente porque o pasto ainda é o cenário majoritário para a criação de gado no Brasil, e apenas 12,6% do rebanho é criado em confinamento, onde as tecnologias empregadas aumentam a produção de carne por animal (ABIEC, 2019).

As regras para exportação de carne bovina são diversas, complexas e variam conforme o país importador. Segundo Sander, Semejin e Mahr (2018), o tema da regulamentação da exportação e da rastreabilidade da carne vem ganhando cada vez mais importância, principalmente após o impacto causado no mercado de carnes da União Europeia pelo escândalo da Operação Carne Fraca, que ocorreu em 2017 no Brasil. Essa e diversas outras crises envolvendo o fornecimento de carne ao longo dos anos, como os casos da doença da vaca louca ao redor do mundo e da carne de cavalo na rede de varejo europeia Tesco, mostram a necessidade de um sistema confiável de rastreamento na cadeia da carne bovina (SANDER; SEMEJIN; MAHR, 2018).

Além das exigências regulatórias, tanto Sander, Semejin e Mahr (2018) como Galvez, Mejuto e Simal-Gandara (2018) mostram uma crescente demanda por parte do próprio consumidor quanto à segurança da origem e da qualidade dos alimentos que consome, sendo a possibilidade de obter informações completas de rastreabilidade dos produtos um fator decisivo na hora da compra.

Apesar da complexidade e dos problemas envolvendo tanto o tema da exportação como o fornecimento dos mercados internos de carne bovina, não existe ainda um consenso ou uma padronização no setor, pois faltam ferramentas de rastreabilidade que possuam um método comum e acessível por todos os atores da cadeia para a verificação e acesso às informações com alta confiabilidade. Essa dificuldade é gerada principalmente pela obrigatoriedade legal da adoção de técnicas de rastreio individual dos animais para exportação, fazendo com que os produtores adotem diversas tecnologias, como os brincos eletrônicos RFID (PIZZUTI et al., 2017). Scarvada, Batalha e Ribeiro (2014, p. 269) também afirmam que existe uma grande dificuldade na aplicação de tecnologias de rastreabilidade e identificação na cadeia de carne bovina, em especial na brasileira, por conta do uso de diferentes tecnologias por parte dos atores envolvidos no processo. É muito comum que diferentes produtores utilizem tecnologias distintas de identificação, como *tags* RFID intra e

extracutâneas ou brincos de identificação RFID de baixa (LF) e ultra-alta frequência (UHF), exigindo equipamentos diferentes para cada caso (Figuras 3 e 4).

Figura 3 – Exemplo de brincos LF e UHF



Fonte: Tags Universe³ e Kimeery Intelligent Technology Co. LTD⁴

Figura 4 – Tag RFID subcutânea



Fonte: AgriExpo⁵

Baseado nas dificuldades apresentadas, este trabalho propõe um modelo de rastreabilidade para a cadeia da carne bovina que, quando comparado com os modelos atuais, seja:

- padronizado;
- completamente rastreável;
- financeiramente viável;

³ Disponível em: <<https://mbf-eu.info/uhf-cattle-tags/uhf-cattle-tags-rafid-tag-section-web/>>. Acesso em: 27 abr. 2019.

⁴ Disponível em: <<https://www.kimeeryrfidtag.com/lf-rfid-tag/lf-animal-rfid-tag/livestock-rfid-allflex-ears-tags-custom-made.html>>. Acesso em: 27 abr. 2019.

⁵ Disponível em: <http://www.agriexpo.online/pt/prod/i-d-ology/product-173571-52418.html#product-item_52408>. Acesso em: 27 abr. 2019.

- inviolável e imutável;
- concordante com as demandas do setor.

No capítulo 2, será realizada uma revisão bibliográfica do estado-da-arte de modelos de rastreabilidade na cadeia da carne bovina, bem como de cadeias logísticas similares. No capítulo 3, será apresentado o modelo de rastreabilidade proposto utilizando *Blockchain* e *Smart Contracts*, detalhes de implementação e a discussão de sua validação. Por fim, no capítulo 4, serão feitas as considerações finais.

2. REVISÃO BIBLIOGRÁFICA

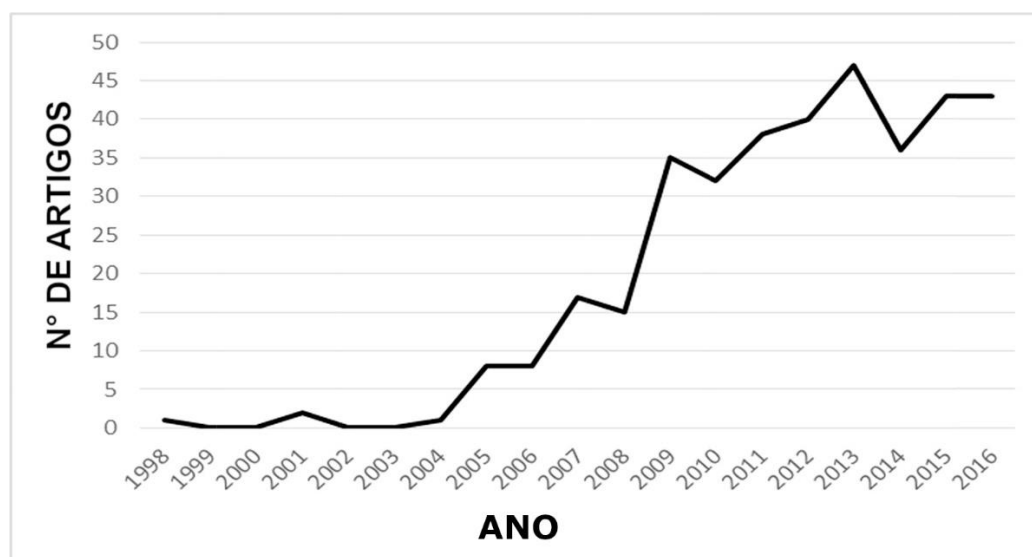
2.1 A rastreabilidade na cadeia de suprimento da carne bovina

A definição mais comum de rastreabilidade é a fornecida pela *International Organization for Standardization* (ISO). Ela define rastreabilidade como:

a capacidade de rastrear o histórico, aplicação ou localização de um objeto. Quando considerados produtos ou serviços, a rastreabilidade pode ser relacionada à origem dos materiais e peças, o histórico de processamento e a distribuição e localização do produto após a entrega (ISO, 2015, tradução nossa).

Segundo Olsen e Borit (2018), o termo “rastreabilidade” teve um aumento significativo em sua utilização em publicações científicas. Portanto, se faz necessária uma análise da situação atual da cadeia de suprimentos da carne bovina com relação ao tema da rastreabilidade para que uma solução alternativa se mostre viável (Figura 5).

Figura 5 – Crescimento da utilização do termo rastreabilidade em artigos



Fonte: Olsen e Borit (2018, tradução nossa)

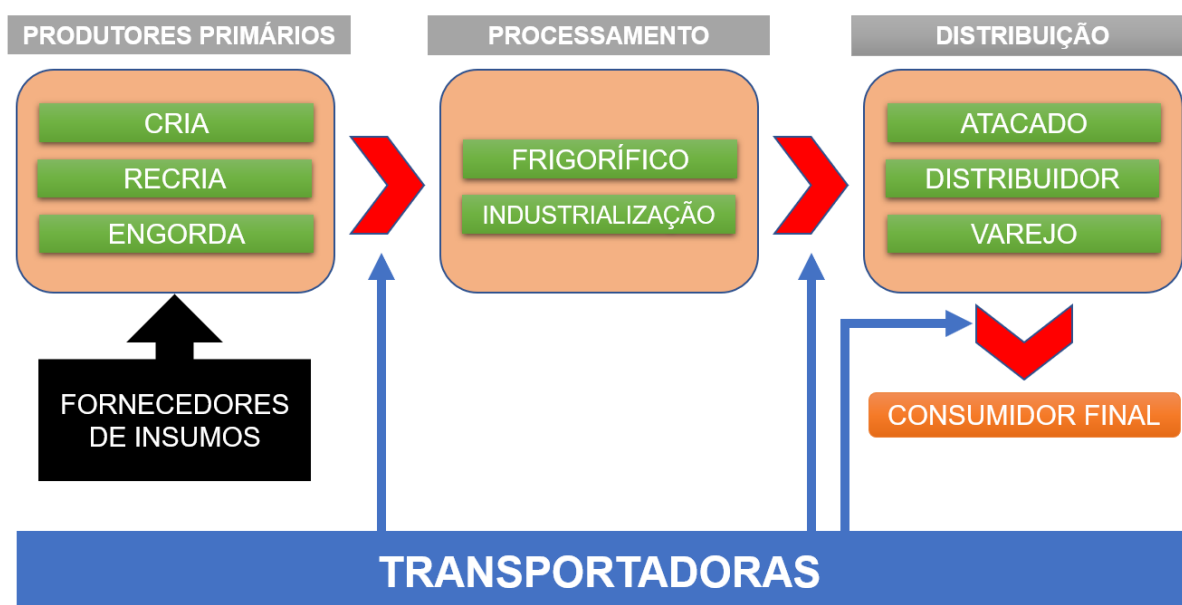
Segundo Pizzuti et al. (2017), a cadeia da carne bovina pode ser resumida nos seguintes principais atores:

- fornecedores de produtos e ração;
- produtores primários e fazendeiros;

- frigoríficos e abatedouros;
- indústrias de processamento e/ou transformação;
- atacados;
- distribuidores e transportadoras;
- consumidores finais.

A Figura 6 mostra uma visão geral da cadeia de suprimento da carne bovina.

Figura 6 – Visão geral da cadeia de suprimento da carne bovina



Fonte: O autor

No Brasil, a fiscalização do cumprimento das regras e a homologação de fazendas e produtores para exportação são feitas por empresas certificadoras credenciadas. Essas regras são definidas pelo Ministério da Agricultura, Pecuária e Abastecimento (MAPA) e delimitam as obrigações para a adoção do Sistema Brasileiro de Identificação Individual de Bovinos e Búfalos (SISBOV). O SISBOV é de adesão voluntária por produtores locais e obrigatório para os que queiram exportar, principalmente para a União Europeia, pois as regras visam atender às exigências desse setor (MAPA, 2017). O próprio SISBOV apresenta falhas estruturais se for avaliada a confiabilidade dos dados, conforme apontado por Silva (2018), que mostrou que as informações fornecidas ao sistema têm origem no próprio produtor e são enviadas e aceitas sem a fiscalização adequada.

Seguindo o caminho da produção da carne bovina, quando o animal deixa a propriedade do produtor para o abate, deve possuir um Guia de Trânsito Animal (GTA)

para o transporte, que é emitido por um médico veterinário credenciado pelo MAPA. O GTA contém informações de rastreabilidade do animal, como origem, destino, finalidade, espécie, vacinas, etc. Outras informações são exigidas nesse processo, como o aviso ao Serviço de Inspeção Federal (SIF), que carrega informações sobre lote, curral, origem, meio de transporte, tempo de viagem e o número do GTA (SILVA et al., 2018). Dessa maneira, é possível identificar que a quantidade de informações necessárias para a rastreabilidade até o ponto do abate é grande, dificultando consideravelmente esse processo.

Os métodos de rastreabilidade na saída do frigorífico (ou abatedouro) também não seguem um padrão definido, variando em fonte, modelo e quantidade de informações, dependendo de fatores como condições, origem e destino do animal, além de diferenças no processo. Para animais que serão utilizados no mercado interno, apenas os doentes são descartados, ou seja, separados do rebanho de corte. Já para os que serão abatidos, as informações de rastreabilidade, quando existem, são desconsideradas no processo. No caso dos animais cuja carne será exportada, a rastreabilidade individual permanece vinculada ao SISBOV, conforme exigido por lei. Ainda dentro do frigorífico, informações em brincos e *tags* RFID, marcações, etc. podem ser perdidas ou mal processadas, causando a perda da rastreabilidade individual do animal e impondo um desafio ainda maior desse ponto da cadeia de suprimentos em diante (GRANDE; VIEIRA, 2013).

Uma vez fora dos frigoríficos, a carne bovina passa pelos canais de distribuição em direção a diversos fins, tais como industrialização (carne processada e embutidos), setor atacadista, distribuidores e venda no varejo ao consumidor final. Durante o transporte, armazenamento, empacotamento e rotulagem da carne, não existe padronização ou consenso entre os atores, que buscam otimizar esse processo e reduzir custos para obter lucro máximo em cada uma de suas etapas (FERRAREZE; JUNIOR; BAPTISTA, 2018).

2.2 *Blockchain* como solução de rastreabilidade em cadeias de suprimentos e logística

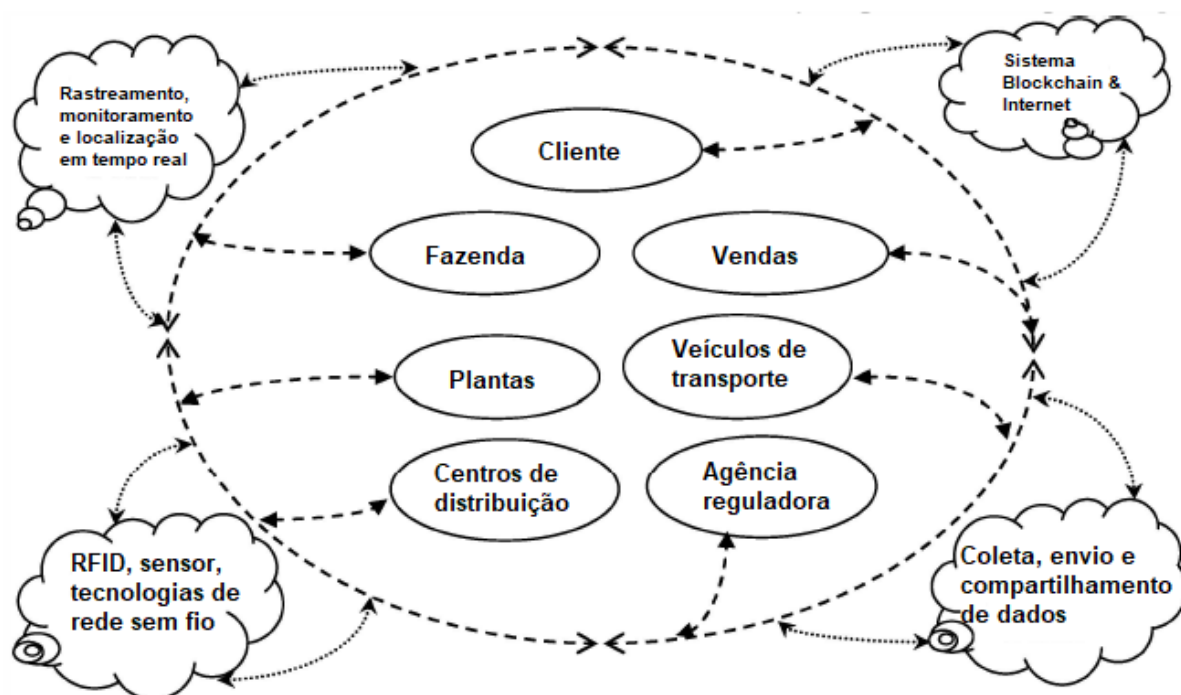
A tecnologia de *Blockchain* nasceu e cresceu em seus primeiros anos com foco em mercados financeiros e moeda, devido às suas características de transparência, descentralização e imutabilidade. Entretanto, foram essas mesmas características

que chamaram a atenção para sua utilização em outras áreas, como logística e cadeia de suprimentos. Atualmente, são esses os mercados mais promissores e que estão dando mais foco à aplicação da *Blockchain* (PERBOLI; MUSSO; ROSANO, 2018).

Uma pesquisa realizada com diversos executivos e profissionais dos setores financeiros, TI, manufatura, saúde, automotivo, alimentação, público e óleo e gás mostrou que o mercado enxerga a *Blockchain* como uma tecnologia que vive seu *momentum* e tem um grande potencial de impacto na economia e nos processos de controle de bens. Dos entrevistados, 72% acreditam que a tecnologia de *Blockchain* é crítica (estando entre as 5 maiores prioridades) ou importante (mas não entre as 5 prioridades) para os negócios (PAWCZUK; MASSEY; SCHATSKY, 2018). Perboli, Musso e Rosano (2018) destacam também que a *Blockchain* tem uma característica de tecnologia disruptiva, assegurando imutabilidade e acesso público aos dados, bem como uma infraestrutura descentralizada, evitando problemas de confiabilidade das informações e as vulnerabilidades inerentes a sistemas centralizados, comuns nas implementações de cadeias de suprimentos e logística. Essas características, aliadas ao potencial da tecnologia de *Blockchain*, têm chamado a atenção principalmente dos mercados de agricultura e alimentação, devido à sua capacidade de oferecer segurança e imutabilidade nas informações para fins regulatórios e de qualidade do produto, evitando problemas de contaminação, por exemplo (PERBOLI; MUSSO; ROSANO, 2018).

Como solução de rastreabilidade na agricultura, Tian (2016) propõe uma implementação que une a tecnologia de RFID à *Blockchain*, de maneira que seja possível criar um monitoramento em tempo real de alimentos de origem agrícola, aliado às informações já presentes nas etiquetas de RFID. Em sua proposta, Tian (2016) detalha que as leituras realizadas nas etiquetas RFID, seja individualmente nos produtos ou em embalagens maiores, como caixas e paletes, devem estar vinculadas a sistemas de *Blockchain*, inserindo as informações contidas na memória das etiquetas, como origem, trajeto, peso, preço, etc. (Figura 7). Galvez, Mejuto e Simal-Gandara (2018) também reforçam que a utilização de etiquetas RFID combinadas a sistemas de *Blockchain* é uma das aplicações mais promissoras para a rastreabilidade de alimentos.

Figura 7 – Sistema de rastreabilidade para a cadeia de alimentos de origem agrícola baseado em RFID e *Blockchain*



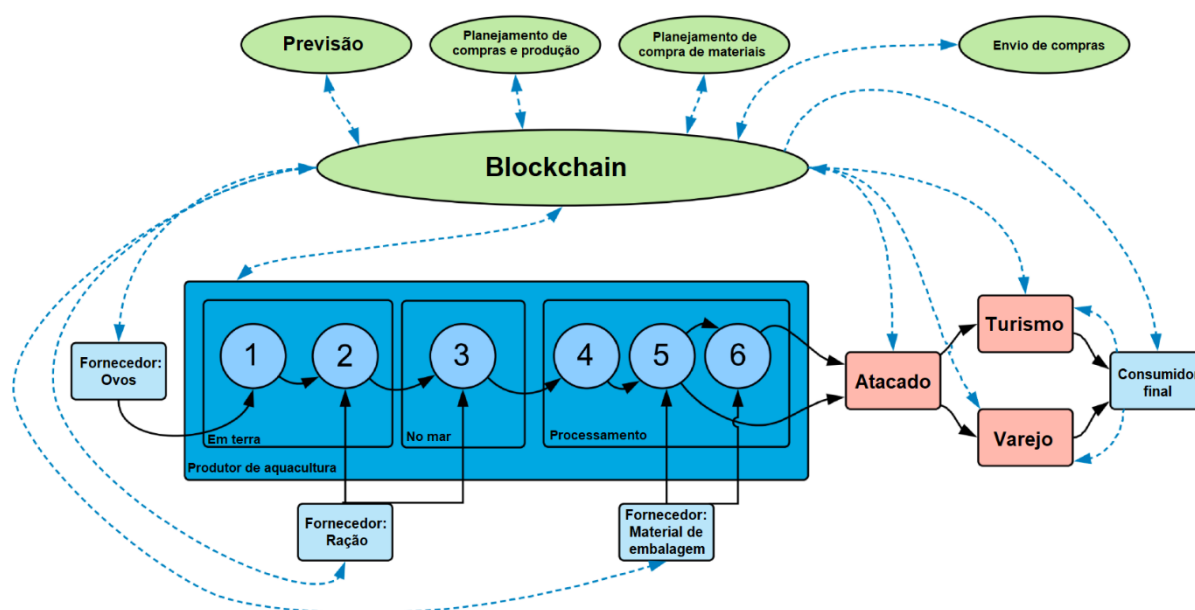
Fonte: Tian (2016, tradução nossa)

Um dos principais problemas envolvendo a rastreabilidade de alimentos é a segurança dos produtos com relação a eventuais riscos de saúde pública. Tse et al. (2017) apontam que as tecnologias atuais de rastreabilidade têm falhado nesse quesito, destacando o crescente aumento de problemas de saúde pública ocorridos na China e em outros países. Tse et al. (2017) sugerem uma implementação para a segurança da informação na cadeia de suprimentos de alimentos, apontando a utilização da *Blockchain* como sendo a escolha mais adequada para suprir os problemas atuais, pontuando que a autenticação de documentos no processo se torna mais confiável, uma vez que o processo é descentralizado e as transações são verificadas e amarradas umas às outras pela própria característica da tecnologia. Em seu modelo de aplicação, as chaves públicas e privadas necessárias para a criptografia dos dados nas transações são controladas por órgãos públicos do governo, que concede a capacidade de interagir com a *Blockchain* para os diversos atores, de fornecedores a consumidores (TSE et al., 2017). Apesar de não ser mencionado no trabalho de Tse et al. (2017), essa aplicação faz parte de um conceito de *Blockchain* permissionada, pois, apesar de o sistema ser distribuído e descentralizado como a rede de transmissão de dados, existe um controle de

participação de nós na rede por uma autoridade central que expede as chaves de criptografia (SHARMA, 2019).

As implementações de *Blockchain* para a rastreabilidade na cadeia logística de alimentos de origem animal também se mostram viáveis em diversos trabalhos. Mathisen (2018) apresenta uma solução de substituição dos modelos atuais de rastreabilidade de salmão na Noruega utilizando *Blockchain* desde os fornecedores de ovos, passando pelos produtores em aquacultura, abate, processamento e varejo até o consumidor final (Figura 8). Nesse modelo, além dos benefícios da rastreabilidade, projeções futuras e planejamento de produção de salmão baseados em dados confiáveis do passado também são propostos, mostrando que a confiabilidade da informação pode ainda agregar valor adicional com previsões de mercado (MATHISEN, 2018).

Figura 8 – Modelo de rastreabilidade de salmão norueguês



Fonte: Mathisen (2018, tradução nossa)

Na mesma linha de alimentos de origem animal, IBM e Walmart fecharam uma parceria para um piloto de sistema de rastreabilidade de carne de porco e manga através da utilização do sistema de *Blockchain* permissionado da própria IBM, o *Hyperledger*. Formando um conjunto de informações coletadas a partir de sensores no transporte da carne, câmeras e RFID na etapa de produção nas fazendas, foi possível criar um sistema robusto de rastreabilidade dos animais até o consumidor

final nas lojas de varejo. Ressalta-se que um dos principais motivos para a proposição desse piloto diz respeito aos diversos escândalos recentes envolvendo a contaminação de alimentos, principalmente os de origem animal (KAMATH, 2018). Atualmente, a IBM (2019) mantém um serviço de rastreabilidade de alimentos chamado *IBM Food Trust*, que é um dos resultados práticos obtidos a partir da implementação piloto estudada por Kamath (2018). Ainda nesse contexto, Leng et al. (2018) avaliam que as informações dos atores envolvidos em uma cadeia de distribuição do mercado agrícola configuram um tema sensível, envolvendo critérios como influência nos negócios, alcance de mercado de determinados produtos e até a reputação de empresas. Em redes *Blockchain* públicas, essas informações podem ser facilmente obtidas por qualquer participante e, partindo desse problema, Leng et al. (2018) sugerem uma implementação utilizando uma cadeia-dupla, combinando um sistema permissionado a um sistema público. No sistema público, acontecem os processos abertos de transação e rastreabilidade e, no sistema permissionado, mantêm-se os dados privados das instituições envolvidas. Essa implementação também visa desafogar um sistema único de *Blockchain* que pode sofrer com problemas de lentidão quando houver muitas transações pendentes em um curto intervalo de tempo (LENG et al., 2018).

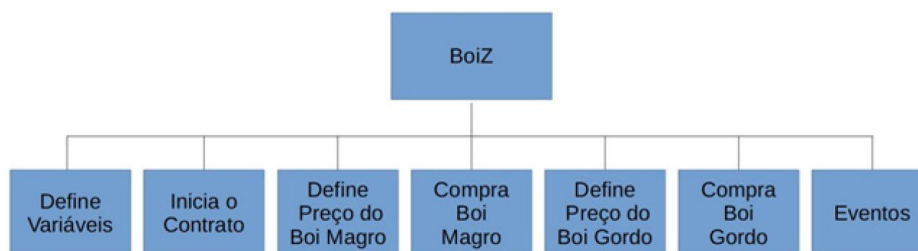
2.3 *Blockchain* como ferramenta na rastreabilidade da carne bovina

No mercado da carne bovina, Sander, Semejin e Mahr (2018) fazem uma análise da aceitação da tecnologia de *Blockchain* como uma ferramenta de rastreabilidade e transparência por meio de uma pesquisa com consumidores e potenciais atores envolvidos no negócio, como produtores, frigoríficos, açougues, varejo, consumidores e governo. O resultado da pesquisa demonstrou que existe uma divergência entre o que os atores de distribuição da cadeia da carne enxergam como sendo a perspectiva do consumidor e a visão real que o consumidor possui. Os consumidores apontaram que a quantidade de rótulos e certificações que acompanham os produtos de carne bovina é alta, que eles não oferecem segurança e que um sistema de rastreabilidade completa poderia influenciar positivamente sua decisão na hora da compra. Do ponto de vista dos produtores, o custo é a principal preocupação do consumidor e tanto a disponibilidade de um sistema de rastreabilidade completo como nenhum disponível não são opções desejáveis. Já os

entrevistados de órgãos governamentais demonstraram uma preocupação com a factibilidade e o custo de implementações com *Blockchain* para o mercado da carne bovina (SANDER; SEMEJIN; MAHR, 2018).

Yano et al. (2018) apresentam uma solução para um modelo de rastreamento bovino utilizando *Smart Contracts* em *Blockchain* com foco no preço do boi (Figura 9). Nesse modelo, é utilizada uma implementação prática a partir de uma rede privada da *Blockchain* da *Ethereum*. O contrato confere dados básicos de rastreabilidade e origem e permite transações financeiras entre os atores da cadeia a partir dos *Smart Contracts*, utilizando o *token* financeiro *ether* e a sua cotação atual frente à moeda do país. Essa implementação tem o objetivo de criar um ambiente comum de negociação entre os atores envolvidos na cadeia da carne bovina e prover um nível aperfeiçoado de rastreabilidade, com segurança nas transações, em comparação a modelos comuns nesse mercado (YANO et al., 2018).

Figura 9 – Estrutura geral do *Smart Contract* proposto por Yano et al. (2018)



Fonte: Yano et al. (2018)

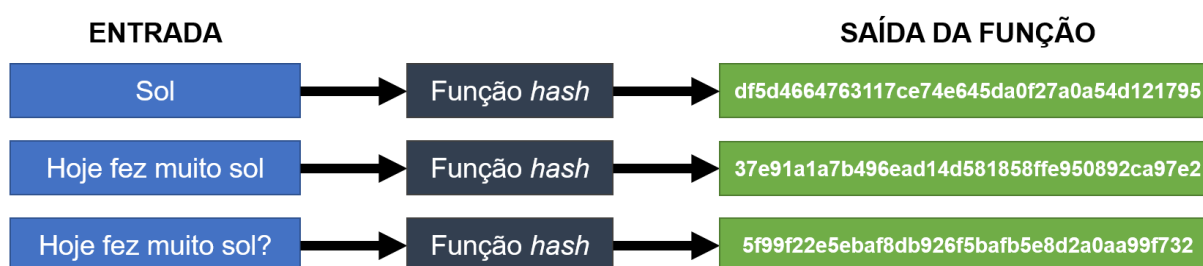
No Brasil, a empresa *SafeTrace* oferece um serviço de *Blockchain* que foi inicialmente proposto para o mercado da carne bovina, mas que atualmente pode ser utilizado para qualquer produto de origem agrícola ou pecuária. É um sistema permissionado de *Blockchain* que também utiliza o sistema *Hyperledger* da IBM. O objetivo é fornecer informações gerais de rastreabilidade do gado que incluem lotes de produção, movimentação, dados sanitários, qualidade, etc. (BLOCKCHAIN..., 2019). Ainda no campo das aplicações práticas, um marco importante foi o primeiro produto de carne bovina dos Estados Unidos a ser completamente rastreado por *Blockchain*. O projeto foi desenvolvido pela Universidade de Wyoming e, utilizando informações do animal associadas a uma etiqueta RFID, os cortes de carne foram enviados até um estabelecimento em Taiwan com rastreabilidade completa feita por *Blockchain*. A demanda nasceu da parceria entre a universidade e a iniciativa privada,

dando origem a uma empresa de rastreamento de carne bovina com *Blockchain* chamada *BeefChain* (UW..., 2019). Um dos mais renomados mercados de carne de qualidade e que movimentam em torno de 13 bilhões de dólares anuais é o da Austrália, no qual a empresa *BeefLedger* fornece um serviço de rastreabilidade para o mercado local utilizando *Blockchain* vinculada a um *token* financeiro chamado *BEEF Token*. O sistema é apoiado e desenvolvido em parceria com uma empresa de seguro para caminhões, com o objetivo de minimizar perdas (AUSTRALIAN..., 2019 e BEEFLEDGER..., 2020).

2.4 A tecnologia de *Blockchain*

A *Blockchain* nasceu como o protocolo por trás de um sistema descentralizado de pagamentos financeiros através de uma criptomoeda digital denominada *Bitcoin*. Ela descreve o funcionamento da rede e das transações financeiras do *Bitcoin*. O trabalho original buscou resolver o problema do gasto duplo, que ocorre quando um usuário consegue gastar a mesma moeda duas vezes, sendo proposta uma solução com um algoritmo de *Proof-of-Work* (POW) através de um método de consenso e assinaturas criptografadas com *hashing* (NAKAMOTO, 2008). As funções *hash*, quando aplicadas a quaisquer conjuntos de dados, produzem uma saída de tamanho igual e valor único a partir de sua condição de resistência às colisões (Figura 10), que é a dificuldade de se encontrar duas saídas iguais a partir de duas entradas diferentes. (DWORKIN, 2015).

Figura 10 – Aplicação de uma função de *hash* de 160 bits para criar uma saída única



Fonte: O autor

Conforme descrito por Kahn e Salah (2018), os endereços (carteiras individuais) do *Bitcoin* na *Blockchain* têm um tamanho de 160 bits, resultando em uma

quantidade máxima de 2^{160} endereços, o mesmo que $1,46 \times 10^{48}$. Considerando o tamanho do rebanho de bovinos e bubalinos no mundo de aproximadamente $1,65 \times 10^9$ cabeças (ABIEC, 2019), é possível estimar a probabilidade P de colisão dos *hashes* ao longo dos anos a partir da Equação 1, sendo N a quantidade máxima de endereços na *Blockchain*, Q o tamanho do rebanho mundial e a o tempo em anos decorridos desde a criação do sistema:

$$P = \frac{Q \times a}{N} \quad (1)$$

Utilizando a Equação 1 como base de cálculo, a Tabela 2 mostra que não é significativo o aumento da probabilidade de colisão em 100 anos de utilização de *hashes* únicos:

Tabela 2 – Probabilidade de colisão, em anos

ANO	PROBABILIDADE DE COLISÃO (%)
1º	$1,13 \times 10^{-39}$
2º	$2,26 \times 10^{-39}$
3º	$3,39 \times 10^{-39}$
50º	$5,64 \times 10^{-38}$
100º	$1,13 \times 10^{-37}$

Fonte: O autor

Considerando ainda um rebanho de tamanho S e uma rede *Blockchain* formada por N endereços, é possível calcular o tempo T em anos decorridos até que se chegue a uma probabilidade P de colisão a partir da Equação 2:

$$T = \frac{N \times P}{S} \quad (2)$$

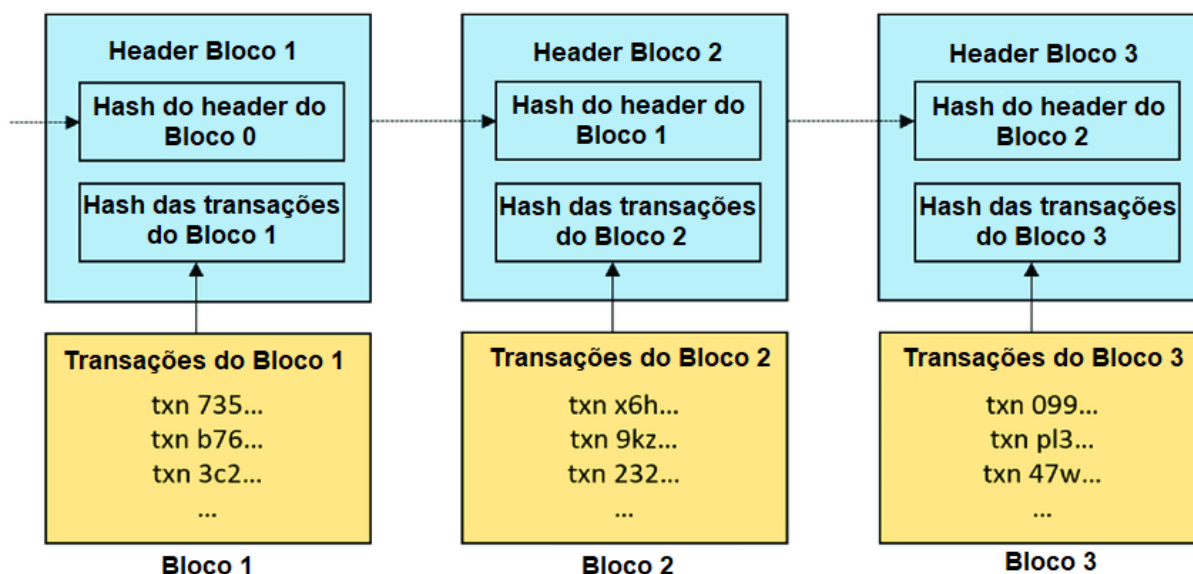
Levando em consideração uma probabilidade alta de colisão de 0,01, um rebanho de 1,65 milhões de cabeças de gado e uma rede *Blockchain* com 2^{160} endereços, conforme apresentados anteriormente, obtém-se um período T de **8,86 x 10³⁹** anos, o qual se julga ser longo e seguro o suficiente para o caso apresentado.

A ideia da *Blockchain* é que, além ser um sistema totalmente descentralizado, sirva também como um acordo confiável para ambientes em que haja desconfiança entre as partes, como é notoriamente o mercado financeiro (TIAN, 2016). Alguns

autores a consideram como uma das maneiras mais eficientes de se emitir transações financeiras pelas instituições bancárias (CROSBY et al., 2016).

Pode-se entender a *Blockchain* como sendo, basicamente, um banco de dados distribuído que contém as informações de um livro-caixa ou registros financeiros. Tian (2016) menciona inclusive que a *Blockchain* é tecnicamente similar ao *NoSQL (Not Only SQL)*, conhecido como um banco de dados não relacional. A *Blockchain* é vista, ainda, como um banco de dados empilhados em forma vertical, sendo o primeiro bloco, conhecido como bloco gênese, a base de toda a estrutura. Cada bloco contém seu próprio tamanho, um cabeçalho de metadados em formato de *hash* que carrega o *hash* do bloco anterior, um contador de transações e todas as transações que foram escritas no bloco, após aprovadas pelo mecanismo de consenso (ANTONOPOULOS, 2017, p. 174). A Figura 11 mostra um exemplo de cadeia de blocos utilizados na *Blockchain*.

Figura 11 – Exemplo de uma cadeia de blocos na *Blockchain*



Fonte: Agbo, Mahmoud e Eklund (2019, tradução nossa)⁶

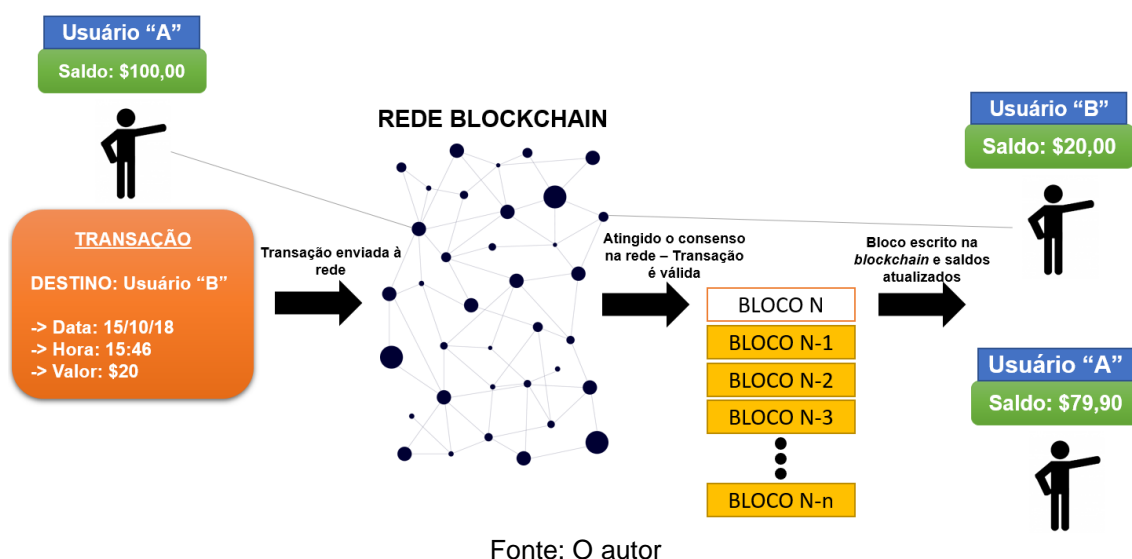
É possível compreender melhor o funcionamento dessa tecnologia quando se analisa o funcionamento de uma rede *Blockchain*. Conforme descrito por Nakamoto (2008), a rede é composta por nós que se comunicam *peer-to-peer*, ou P2P, e operam

⁶ AGBO, C. C.; MAHMOUD, Q. H.; EKLUND, J. M. Blockchain Technology in Healthcare: A Systematic Review. *Healthcare*, Basileia, v. 7, n. 2, p. 1-30, 2019, 04 abr. 2019. Disponível em: <<https://doi.org/10.3390/healthcare7020056>>. Acesso em: 23 ago. 2020.

na mesma rede. Um nó pode conter uma cópia completa da *Blockchain* (o banco de dados em si) ou ser uma versão mais leve, que apenas processa e armazena transações geradas localmente, mas é possível assumir que todos os nós possuem uma cópia completa. A Figura 12 exemplifica o seu funcionamento, utilizando o envio de valores financeiros para facilitar a compreensão de todo o processo, entretanto, Christidis e Devetsikiotis (2016) apontam que uma rede de *Blockchain* pode operar por si só, sem a necessidade de haver uma criptomoeda envolvida.

No exemplo proposto a seguir, toma-se um cenário em que o usuário “A”, com um saldo inicial de \$100 quer enviar \$20 ao usuário “B” com um saldo inicial de \$0 (Figura 12).

Figura 12 – Fluxo da transação dentro da rede de *Blockchain*



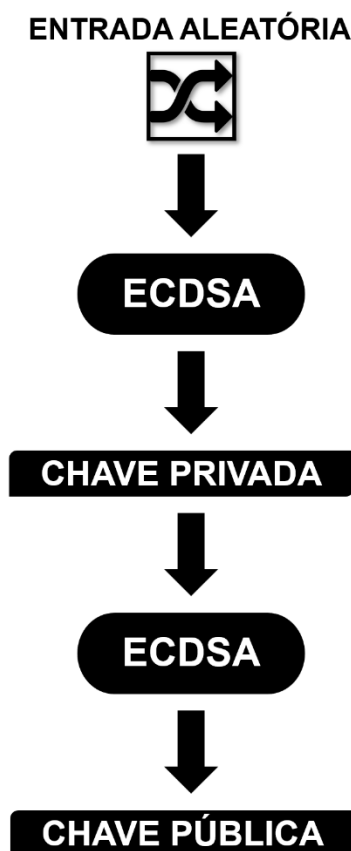
No exemplo da Figura 12, é possível observar que o usuário “A” terminou com menos do que o esperado em seu saldo final. Isso acontece porque é comum que as redes de *Blockchain* das criptomoedas atuais cobrem uma taxa de transação. Essa taxa é de fato o estímulo para que os nós continuem existindo na rede, mantendo o sistema em funcionamento para garantir a sua segurança e usabilidade. Outras recompensas podem ser pagas em forma de unidades de criptomoeda para os nós que foram responsáveis pela mineração do bloco (ANTONOPOULOS, 2017, p. 189).

2.5 Segurança da tecnologia *Blockchain*

O protocolo que descreve a *Blockchain* foi desenvolvido com o intuito de ser, em termos práticos, inviolável. Para compreender como isso é possível, serão analisadas as camadas de segurança utilizadas pela maioria dos sistemas de *Blockchain* existentes atualmente, focando principalmente nas implementações do *Bitcoin* e *Ethereum*, as duas criptomoedas mais utilizadas e consolidadas até o momento (KING, 2019).

Antonopoulos (2017, p. 68) descreve que o componente básico de um sistema de *Blockchain* nos moldes das criptomoedas é a carteira. O nome é apenas uma analogia, pois, em um cenário de transações financeiras, a moeda é o elemento central, entretanto, esse componente é de fato apenas um aplicativo que armazena duas informações importantes: a chave pública e a chave privada do usuário. Maesa, Marino e Ricci (2016) apontam que ambas as chaves são criadas utilizando o já conhecido método do Algoritmo de Assinatura Digital de Curvas Elípticas (*Elliptic Curve Digital Signature Algorithm*, ECDSA), sendo a chave privada criada de maneira aleatória e a chave pública criada a partir da aplicação do ECDSA sobre a própria chave privada (Figura 13). A chave privada é a prova da propriedade das informações da carteira (quantidade de moeda, dados, etc.) e uma espécie de carimbo digital para as transações na *Blockchain*, enquanto a chave pública é o endereço da carteira, que deve ser compartilhado entre as partes envolvidas em uma transação (ANTONOPOULOS, 2017, p. 68).

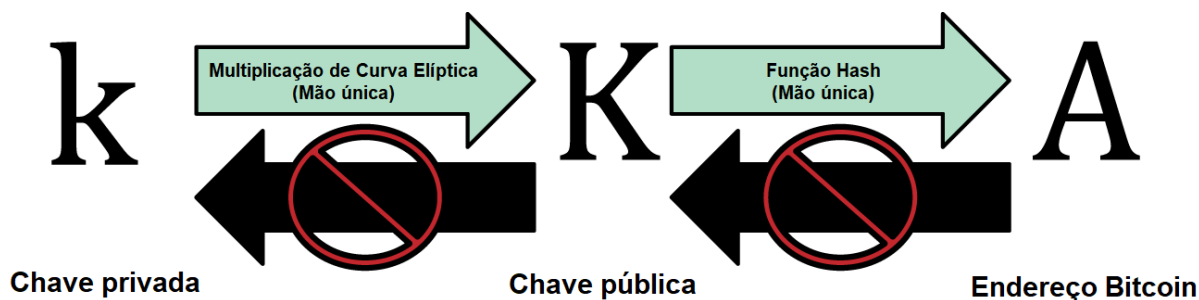
Figura 13 – Fluxo de criação das chaves criptografadas no *Bitcoin*



Fonte: O autor

As funções de criptografia utilizadas na *Blockchain* são conhecidas como funções de mão única (Figura 14), baseadas no conceito de criptografia assimétrica e assinaturas digitais com o ECDSA. Essas funções têm como característica serem calculadas facilmente a partir da fonte, mas, até o presente momento, sendo consideradas computacionalmente impossíveis de serem calculadas ao contrário, revelando a informação original (FEI; SHUI-SHENG; MIN, 2005).

Figura 14 – Aplicação de funções criptográficas de mão única



Fonte: Antonopoulos (2017, p. 68, tradução nossa)

Conforme descrevem Antonopoulos e Wood (2019), as chaves privadas devem ser geradas a partir de fontes seguras de aleatoriedade e entropia, como o usuário movendo o *mouse* em direções aleatórias ou o ruído presente no microfone do dispositivo que está sendo usado. Sem seguir os padrões definidos pela implementação do protocolo do sistema de *Blockchain* utilizado para a criação de chaves privadas e públicas, expõe-se uma falha de segurança na implementação. Uma dessas falhas foi observada recentemente com a criação de chaves privadas inseguras em carteiras digitais na rede *Ethereum*. Essas carteiras foram invadidas com técnicas de força bruta, ocasião em que foram descobertas chaves privadas geradas a partir de métodos frágeis (DUNN, 2019).

Uma vez estabelecidas as chaves privadas e públicas, qualquer aplicação voltada para funcionar em uma rede *Blockchain* está pronta para enviar e receber transações. As transações são criadas nos nós de origem e enviadas à rede para serem acrescentadas a um bloco, que precisa ser validado para garantir sua autenticidade, em especial que não haja gasto duplo ou alteração do estado da *Blockchain* atual. Esse é o trabalho dos métodos de consenso, por exemplo, o POW (NAKAMOTO, 2008).

Segundo Christidis e Devetsikiotis (2016), os mecanismos de consenso variam de acordo com cada implementação de *Blockchain*, mas todos derivam da ideia original criada no *Bitcoin* para aplicação de POW na rede P2P. De acordo com Douceur (2002), as redes P2P estão sujeitas ao ataque Sybil, que acontece quando uma única entidade pode criar diversas entidades adicionais de modo a controlar mais da metade dos nós da rede. Esse cenário seria problemático se o conceito de consenso na rede *Blockchain* fosse dependente de uma votação baseada em maioria (CHRISTIDIS; DEVETSIKIOTIS, 2016). Para evitar esse problema, Nakamoto (2008) apresentou o conceito de mineração, que é o trabalho computacional realizado para se encontrar o *hash* equivalente de um bloco candidato a entrar na *Blockchain*. Esse trabalho computacional é tão grande que, quando multiplicado pelo número de blocos já minerados, a criação de entidades adicionais, como as descritas no ataque Sybil, se torna impraticável (CHRISTIDIS; DEVETSIKIOTIS, 2016).

No trabalho original de Nakamoto (2008), a dificuldade para minerar um bloco aumenta a cada N blocos encontrados (2.016 blocos no *Bitcoin*), criando um sistema que aumenta sua segurança ao longo do tempo. O objetivo dos mineradores é encontrar um valor inteiro chamado de *nonce*, tal que:

$$H(B, n) < A_d \quad (3)$$

Onde $H(B, n)$ é a função *hash* do bloco B e n (*nonce*) e A_d é o objetivo com dificuldade d (OLIVER; RICOTTONE; PHILIPPOPOULOS, 2017).

Atualmente, devido à grande quantidade de sistemas de *Blockchain* existentes, diversos mecanismos de consenso que se adequam aos mais diferentes cenários foram criados. Muitas implementações nas quais a *Blockchain* tem sido empregada demandam um alto *throughput* de dados e, conseqüentemente, precisam ser altamente escaláveis. Essa escalabilidade pode ser definida como uma relação entre *throughput*, custo e capacidade (SHARMA; JAIN, 2019). Xiao et al. (2020) reforçam a preocupação com a escalabilidade e manutenção a longo prazo do algoritmo de POW na rede *Bitcoin*, destacando os principais problemas desse mecanismo, como o consumo insustentável de energia elétrica e o seu baixo número de transações por segundo.

A seguir, pode-se observar alguns detalhes de outros algoritmos além do POW que são muito utilizados nos sistemas de *Blockchain* atuais:

- **Proof-of-Stake (POS):** Esse algoritmo nasceu em 2011, na tentativa de se sobrepor ao problema do consumo de energia do POW. Bastiani (2019) pontua que, no POS, os blocos não são minerados, mas sim construídos. Nele, os nós que desejam participar da seleção dos blocos a serem escritos na *Blockchain* devem bloquear (ou colocar em jogo) uma certa quantidade de criptomoedas. O nó vencedor é escolhido baseado em um algoritmo pseudorrandômico e aqueles que apostaram mais valor têm mais chances de serem escolhidos. Apesar de corrigir o problema do gasto de energia, o POS traz novos desafios, como nós com muito dinheiro tomando controle da rede e destruindo a descentralização, por exemplo. Salimitari e Chatterjee (2020, no prelo) também mencionam que o POS pode ocasionar situações de *nothing at stake*, quando nenhum nó aposta nada e os blocos não podem ser construídos.
- **Delegated Proof-of-Stake (DPOS):** O DPOS é uma variação do POS. Nesse algoritmo, de 21 a 100 nós são selecionados, ou delegados. De acordo com uma ordem definida, os nós delegados podem entregar blocos à *Blockchain* e essa ordem muda com o tempo. Um nó que publica transações inválidas ou perde blocos é retirado da rede. Os usuários da rede podem usar seu poder de

voto para eleger os nós mais confiáveis. Esse algoritmo é considerado rápido, eficiente e tem um excelente balanço entre risco e nível de proteção quando comparado ao POS (SHARMA; JAIN 2019).

- **Proof-of-Activity (POAC):** O POAC é basicamente uma junção do POW e do POS. O algoritmo se inicia com uma mineração utilizando POW, mas apenas de um pedaço, ou *template*, do bloco. Uma vez que a mineração é finalizada, o algoritmo funciona como o POS e um grupo de validadores com a maior quantidade de moedas assina o novo bloco para ser acrescentado à *Blockchain*. (SHARMA; JAIN, 2019). Um dos métodos de ataque conhecidos é o ataque de 51%, que é quando a maioria dos nós da rede são maliciosos e podem influenciar no mecanismo de consenso. Entretanto, Bashar et al. (2019) mencionam que a chance de ataques de 51% é reduzida consideravelmente no POAC, uma vez que um nó precisaria controlar tanto o poder de *hash* da rede quanto a maioria das criptomoedas.
- **Practical Byzantine Fault Tolerance (PBFT):** O PBFT foi desenvolvido para atuar como uma solução em sistemas que precisam ser tolerantes a falhas bizantinas. Esses algoritmos resolvem o problema dos generais bizantinos, que propõe um cenário em que um grupo de generais que se comunicam apenas por mensagens precisa decidir entre atacar ou não uma cidade cercada pelo grupo e, entre eles, pode haver traidores que tomarão a decisão contrária, prejudicando o grupo (ZHENG et al., 2017). Conforme descreve Veronese et al. (2011), para atingir um consenso dentro do problema dos generais bizantinos, é preciso que haja no mínimo $3f + 1$ nós de acordo, sendo f a quantidade de nós maliciosos. Sharma e Jain (2019) apontam que, apesar dos algoritmos PBFT terem um baixo consumo de energia e serem bons no processo de decisão, são passíveis de ataques do tipo Sybil e não são escaláveis.

Esses são alguns algoritmos mais comuns na atualidade, porém existem outros que podem ganhar mais ou menos importância, como o *Proof-of-Authority* (POA), *Proof-of-Burn* (POB), *Proof-of-Weight* (POWEIGHT), *Proof-of-Trust* (POT), *Proof-of-Capacity* (POC), *Proof-of-Importance* (POI), *Proof-of-Elapsed Time* (POET), *Proof-of-Vote* (POV), *Proof-of-Luck* (POL), entre outros. A Tabela 3 apresenta um resumo de todos esses mecanismos, apontando suas principais vantagens e desvantagens.

Seguindo com a ideia dos blocos na *Blockchain*, por serem calculados utilizando o *hash* do bloco anterior, eles trazem uma segurança intrínseca ao protocolo, pois se um *hash* de um bloco anterior a outro escrito na cadeia sofrer qualquer alteração de valor, todos os blocos subsequentes passarão a ser inválidos, uma vez que seus *hashes* deverão ser recalculados (ou minerados), exigindo um esforço computacional tão grande quanto o já gasto até o ponto da alteração, mais sua dificuldade atual, considerando o algoritmo de POW. Ao longo do tempo, para cada novo bloco inserido, tanto o tamanho do banco de dados quanto a dificuldade imposta pela regra de mineração tornam a *Blockchain* inviolável considerando a capacidade computacional atual (OLIVER; RICOTTONE; PHILIPPOPOULOS, 2017). Outros algoritmos carregam uma dificuldade diretamente proporcional à sua característica de implementação.

Akita (2017) mostrou a inviabilidade financeira de se vencer o consenso do *Bitcoin* em seu estado e dificuldade atual, demonstrando que seria necessário um investimento de aproximadamente 1,5 bilhões de dólares para adquirir 1 milhão de unidades do computador AntMiner s9, especializado em mineração, sem considerar todo o gasto de energia com o processo.

Tabela 3 – Resumo dos mecanismos de consenso

Algoritmo	Vantagens	Desvantagens
Proof-of-Work	Segurança Fácil implementação	Alto consumo de energia Não escalável
Proof-of-Stake	Baixo consumo de energia Transações mais rápidas	Suscetível à descentralização se houver nós com muitas moedas Situações de “ <i>nothing at stake</i> ”
Delegated Proof-of-Stake	Baixo consumo de energia Transações mais rápidas Algoritmo eficiente	Risco de mais centralização nas decisões
Proof-of-Activity	Mais resistente a ataques de 51%	Situações de “ <i>nothing at stake</i> ” Consumo moderado de energia
PBFT	Menos números de confirmações para validar um bloco Baixo consumo de energia	Suscetível a ataques Sybil Baixa escalabilidade
Proof-of-Authority	Dificuldade alta para nós não confiáveis entrarem na rede	Os nós de validação não são anônimos Centralizado
Proof-of-Burn	Estabilidade	As moedas utilizadas para validação não são recuperáveis
Proof-of-Weight	Altamente escalável Customizável	Sem recompensas aos blocos validadores
Proof-of-Trust	Muito rápido Alta escalabilidade Eficiente	Suscetível a ataques Sybil
Proof-of-Capacity	Baseado no POW Mais rápido do que o POW	Utilização de espaço de disco rígido
Proof-of-Importance	Similar ao POS Método de colheita nos nós	Nós já ricos ficam mais ricos, pois são mais escolhidos
Proof-of-Elapsed Time	Alta segurança Somente nós autorizados entram na rede	Exige hardware dedicado Aceitação de novos nós na rede é centralizada
Proof-of-Vote	Versão mais eficiente do POW Menor consumo de energia	Dificuldade de ser atualizado em caso de problema
Proof-of-Luck	Extremamente seguro Rápido	Exige processadores especializados da Intel para rodar o algoritmo

Fonte: O autor

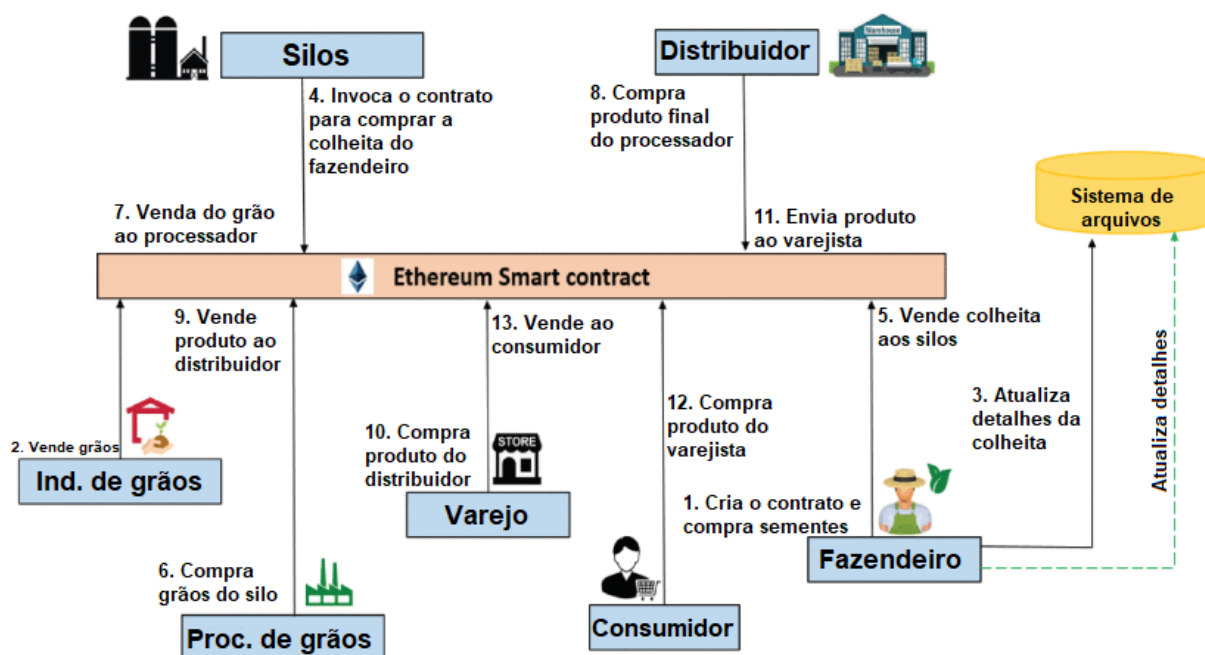
2.6 *Smart Contracts*

O conceito de *Smart Contracts* nasceu do trabalho de Szabo (1994, tradução nossa), em que ele descreve a ideia como sendo “um protocolo de transação computadorizado que executa os termos de um contrato”. Szabo (1997) propôs também a utilização de *Smart Contracts* embarcados em dispositivos de *hardware* e *software* para que fosse possível aplicar cláusulas contratuais de maneira segura em redes de computadores. Na prática, dentro de uma *Blockchain*, os *Smart Contracts* evoluíram a partir do seu conceito inicial e são hoje programas de computadores que executam determinadas tarefas quando uma ou mais condições do contrato são atingidas (CHRISTIDIS; DEVETSIKIOTIS, 2016). Antonopoulos e Wood (2019, tradução nossa) reforçam a definição de um *Smart Contract* como sendo “nem inteligente (*smart*) e nem um contrato (*contract*)”, mas sim um programa de computador que é executado na sua plataforma com as seguintes características: imutável, determinístico, limitado ao contexto da rede e descentralizado.

Conforme já mencionado anteriormente, Yano et al. (2018) utilizam uma solução de *Smart Contracts* para realizar rastreabilidade na cadeia da carne bovina aplicando um algoritmo de controle e precificação em uma aplicação privada do sistema de *Blockchain Ethereum*. A plataforma *Food Trust* também utiliza os *Smart Contracts* como ferramenta de criação dos produtos e dos atores envolvidos na cadeia de distribuição dos alimentos através de sua plataforma de *Blockchain Hyperledger* (IBM, 2019).

Salah et al. (2019) propõem um modelo de rastreabilidade de grãos de soja, que pode ser usado em diversos produtos na agricultura, utilizando *Smart Contracts* também na plataforma *Ethereum* (Figura 15). O modelo considera todo o ciclo do grão de soja chegando até o consumidor final, detalhando as entidades-relacionamento criadas através do *Smart Contract* para modelagem da cadeia de distribuição. Seu modelo sugere ainda o armazenamento de diversas informações relacionadas ao processo, incluindo imagens, documentos, assinaturas digitais e até mesmo arquivos de vídeo que possam servir como informação para tomadas de decisão caso seja necessário (SALAH et al., 2019).

Figura 15 – Rastreabilidade de grãos de soja utilizando *Smart Contracts*



Fonte: Salah et al. (2019, tradução nossa)

Já em um trabalho com um conceito generalista, Wang et al. (2019) sugerem uma proposta de rastreabilidade que seja aplicável a qualquer produto dentro de uma cadeia de suprimentos. Além de definir a base de qualquer cadeia de suprimentos, contendo os distribuidores, varejistas, consumidores e departamentos de regulação, o trabalho também define três diferentes *Smart Contracts*, sendo eles o *Product Registration Contract* (PRC) para registro dos produtos; o *Batch Addition Contract* (BAC) para amarrar produtos aos seus respectivos lotes de transporte; e, por fim, o *Transaction Update Contract* (TUC), que tem a função de atualizar o histórico dos lotes na *Blockchain* (WANG et al., 2019). Esse conceito de utilização dos *Smart Contracts* como aplicações e ferramentas de controle de fluxo de dados e criação dos bens e atores da cadeia servirá como base para a metodologia proposta neste trabalho.

Em um caso de aplicação prática muito similar ao trabalho de Wang et al. (2019), a empresa brasileira SBR PRIME utiliza *Blockchain* e *Smart Contracts* para rastreamento de grãos, conectando diversos aplicativos e dispositivos de IoT ao longo do processo logístico para coleta de dados e inserção automática na *Blockchain* (ALMEIDA, 2019).

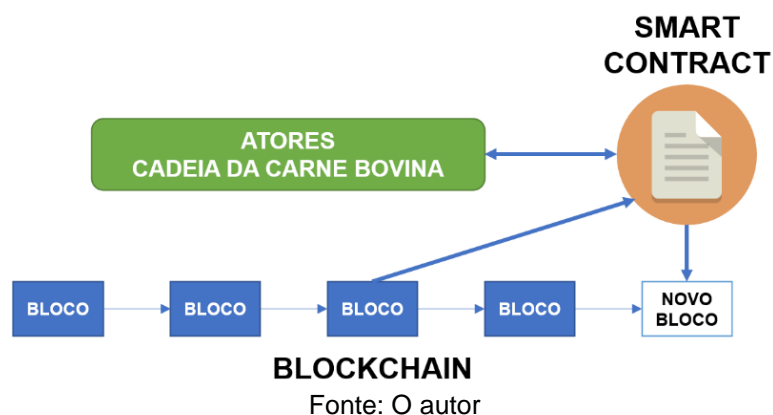
3. MODELO DE RASTREABILIDADE PARA A CADEIA DA CARNE BOVINA

Neste capítulo, será detalhada a proposta de solução para os problemas de rastreabilidade na cadeia da carne bovina apresentados ao longo do texto. Essa proposta é um modelo que utiliza *Smart Contracts* na *Blockchain* como uma ferramenta de rastreabilidade para fornecer um ambiente padronizado, completamente rastreável, confiável, viável e seguro para a cadeia da carne bovina. As características da tecnologia *Blockchain* permitem a criação de registros sequenciais e protegidos pela estrutura do dado em si, semelhante a uma lista ligada. Esses registros são de fato os blocos criados com as diversas transações ocorridas, sempre com um cabeçalho que carrega seu próprio *hash* formado a partir do *hash* do bloco anterior, bem como um *timestamp*. Esse desenho de sistema permite manter uma informação rastreável e segura para a cadeia da carne bovina.

Em comparação com os sistemas de rastreamento de carne bovina providos pelas empresas *SafeTrace* e *BeefChain* (BLOCKCHAIN..., 2019 e UW..., 2019), o modelo apresentado neste trabalho se assemelha ao utilizar as informações mais comuns de rastreabilidade animal como entradas na *Blockchain*, porém se difere ao detalhar a interação entre os atores da cadeia e a *Blockchain*. Também se destacam no modelo aqui sugerido a utilização explícita de *Smart Contracts* para o controle dos dados e o vínculo do DNA animal ao seu registro inicial na *Blockchain*. A implementação do modelo é também independente de um *token* financeiro, divergindo da implementação criada pela empresa *BeefLedger* (AUSTRALIAN..., 2019 e BEEFLEDGER..., 2020) que está atrelada a uma criptomoeda.

No modelo proposto por este trabalho, um *Smart Contract* com um algoritmo específico deve ser o responsável por criar os bens (ou *assets*), que são os animais e os produtos derivados, e os atores da cadeia da carne bovina (produtores, transportadoras, consumidores finais, etc.) dentro da *Blockchain* (Figura 16). Todos os atores farão parte do sistema como sendo a representação dos nós da rede, criando o ambiente descentralizado que permite as validações e tomadas de decisão dentro da *Blockchain*. O *Smart Contract* também deverá ter a função de validar dados de entrada, bem como atualizar os bens já existentes na *Blockchain* com novas informações.

Figura 16 – Smart Contract e a interação com os atores e a Blockchain



Será detalhado como os atores terão interação com a rede de *Blockchain* proposta, qual o papel e as funções do *Smart Contract* e as ferramentas que são utilizadas para a obtenção dos objetivos definidos.

3.1 Definição dos atores e da rede *Blockchain*

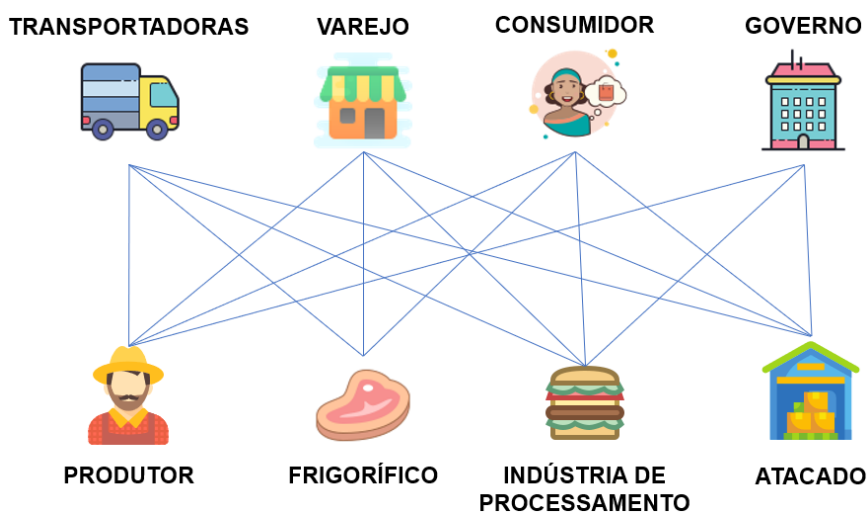
Com base no princípio das redes descentralizadas dos sistemas de *Blockchain* que foi apresentado, é possível observar um forte potencial de aplicação da tecnologia no contexto da cadeia da carne bovina. O modo como os atores da cadeia se relacionam pode ser diretamente aplicado às arquiteturas descentralizadas de rede, transferindo o poder de decisão e validação para todos os envolvidos, assim conferindo transparência e confiabilidade no processo. Nesse modelo, são definidos como atores da cadeia da carne bovina:

- Produtores primários: fazendeiros e produtores das etapas de cria, recria e engorda de gado.
- Frigoríficos: os frigoríficos e abatedouros estão dentro da etapa de processamento na cadeia da carne bovina.
- Indústrias: ainda na etapa de processamento, as indústrias de processamento de carne também fazem parte da cadeia, sendo responsáveis pela produção de embutidos e carne processada.
- Distribuidores: empresas responsáveis pela distribuição dos produtos ao atacado e varejo.

- Atacado: mercado de carne em grandes volumes, geralmente peças e produtos em maiores quantidades.
- Varejo: no varejo, foram inclusos supermercados, pequenas vendas e açougues.
- Transportadoras: as transportadoras são a espinha dorsal da cadeia de distribuição da carne bovina, estando envolvidas no processo desde o início até a entrega final ao atacado e varejo.
- Órgãos governamentais: qualquer órgão do governo ou agência reguladora que esteja diretamente ligada à cadeia da carne bovina.
- Consumidor final: o consumidor deve ter a capacidade de verificar toda a informação de rastreabilidade de um produto que teve seu registro mantido na *Blockchain*.

A Figura 17 ilustra como os atores da cadeia da carne bovina se relacionam de maneira descentralizada e distribuída na rede *Blockchain* proposta neste modelo.

Figura 17 – Atores da cadeia da carne bovina na rede *Blockchain*



Fonte: O autor.

A partir das estruturas convencionais das redes de *Blockchain*, podem existir nós leves e nós completos. No modelo proposto, com exceção do consumidor final, todos os outros atores da cadeia devem ser nós completos, contendo uma cópia completa do banco de registros da *Blockchain* e a capacidade de participar da rede como um nó minerador. Como o consumidor final só atua em modo leitura, um nó leve já é suficiente, contendo apenas um subconjunto do banco completo e atualizando

informações sob demanda a fim de assegurar performance. É possível que o consumidor final também tenha permissão de escrita na *Blockchain*, porém com a intenção de dar sua opinião sobre o produto consumido e tecer comentários, sugestões e reclamações. Isso pode gerar valor à plataforma, fechando o ciclo completo da cadeia de forma que até mesmo o consumidor interaja com o bem consumido de maneira ativa.

Os atores que fazem parte do sistema devem ser registrados individualmente na *Blockchain* através do *Smart Contract* seguindo a estrutura apresentada na Tabela 4.

Tabela 4 – Esquema de registro de identificação individual dos participantes (atores) da cadeia da carne bovina na *Blockchain*

Participante	Identificação
<i>Farmer</i>	<i>farmerId</i>
<i>Slaughterhouse</i>	<i>companyId</i>
<i>Transportation</i>	<i>companyId</i>
<i>ProcessingIndustry</i>	<i>companyId</i>
<i>Retail</i>	<i>companyId</i>
<i>Consumer</i>	<i>consumerId</i>

Fonte: O autor

O modelo proposto neste trabalho não tem o objetivo de especificar uma plataforma de *Blockchain* a ser utilizada, mas é importante ressaltar que, para uma implementação eficiente da proposta, deve ser utilizada uma plataforma que permita o balanceamento das cargas de dados de nós leves e completos através da construção de *DApps* (*Decentralized Apps* ou Aplicativos Descentralizados) específicos para cada caso. Os *DApps* são os aplicativos e *software* de utilização dos nós na *Blockchain* e recebem esse nome a partir da sua utilização descentralizada e interação com os *Smart Contracts* (WU et al., 2019). Os *DApps* não são parte do escopo deste trabalho, mas entende-se que as definições aqui apresentadas sejam detalhadas o suficiente para que *DApps* possam ser desenvolvidos para os casos de uso específicos na cadeia da carne bovina com relativa facilidade.

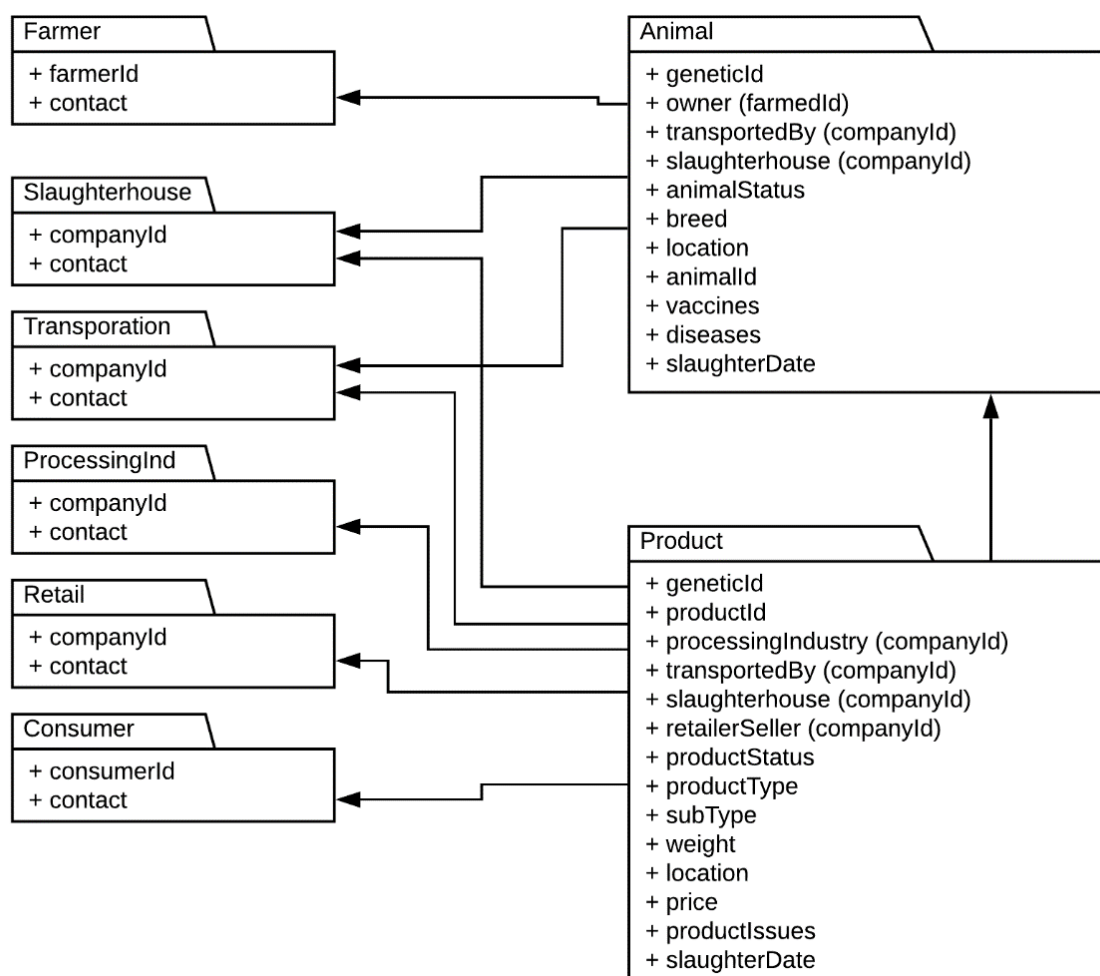
3.2 Definição dos bens rastreáveis

No modelo proposto, cada animal individual é definido como sendo um bem rastreável dentro da rede *Blockchain* e deve ser vinculado a um registro de identificação genética, ou seja, do seu DNA. Os detalhes da implementação de registro da identificação genética do animal serão apresentados mais adiante no trabalho. É mandatória a criação do animal na rede com alguma forma de identificação genética, pois essa é a característica que é herdada pelo bem definido a seguir como produto.

O produto é o outro bem rastreável definido no modelo. Os produtos na cadeia da carne bovina podem ser de diversos tipos, como embutidos, cortes específicos, peças inteiras, etc. No momento da sua criação na *Blockchain*, o produto herda a identificação genética do seu animal de origem, o que permite vinculá-lo de maneira imediata para prover total rastreabilidade.

Na sequência, a Figura 18 apresenta como os bens são modelados de acordo com suas características rastreáveis e suas relações com os atores da cadeia, as quais devem ser estabelecidas para prover um alto nível de rastreabilidade.

Figura 18 – Modelagem dos bens e seu relacionamento com os atores da cadeia da carne bovina



Fonte: O autor

As características denominadas *animalStatus* e *productStatus* são enumerações pré-definidas de acordo com a Tabela 5:

Tabela 5 – Definição das enumerações do status dos bens

<i>animalStatus</i>	<i>productStatus</i>
<i>Breeding</i>	<i>In Processing</i>
<i>Rearing</i>	<i>Packaging</i>
<i>In Transit</i>	<i>In Transit</i>
<i>Fattening</i>	<i>For sale</i>
<i>Slaughtering</i>	<i>Sold</i>
<i>Cutting</i>	<i>Expired</i>
<i>Deceased</i>	<i>Discarded</i>

Fonte: O autor

3.3 O DNA animal como identificação inicial obrigatória

Apesar de todas as garantias conferidas por sistemas de *Blockchain*, não existe uma maneira de assegurar a veracidade do dado que foi gravado. Isso cria um elo frágil na confiabilidade do processo e demanda a necessidade por um método eficaz de garantia da procedência das informações.

Para corrigir essa fragilidade que sistemas de rastreabilidade baseados em *Blockchain* possam ter, este trabalho sugere a utilização de uma identificação inicial obrigatória a partir do *fingerprint* genético, ou *DNA fingerprint*, como sendo o código único e inviolável que vincula de maneira não falsificável um animal que se pretende rastrear na *Blockchain* com seus dados registrados na rede. Essa abordagem se mostra factível uma vez que o sequenciamento genético dos animais é um procedimento comum entre os produtores na etapa da cria (VANKAN; BURNS, 1997) e o custo é acessível quando os métodos de reação em cadeia pela polimerase (PCR – *Polymerase Chain Reaction*) são utilizados para detecção do *fingerprint* genético, não apresentando um impacto considerável no custo final da manutenção do animal (BRUGNANO et al., 2014). Tanto Vázquez et al. (2004) como Davis et al. (2006) apresentam o *fingerprint* genético como sendo a melhor maneira de se obter um registro individual, rastreável e não falsificável do gado, podendo ser validado a qualquer momento ao longo da cadeia da carne bovina. Vankan e Burns (1997) descrevem que é possível obter uma identificação genética do gado a fim de detectar parentesco e paternidade de bezerros a partir da utilização dos já conhecidos marcadores genéticos, enquanto Georges et al. (1990) demonstram a possibilidade de relacionar parentesco entre bovinos de diferentes raças a partir de *fingerprints* genéticos.

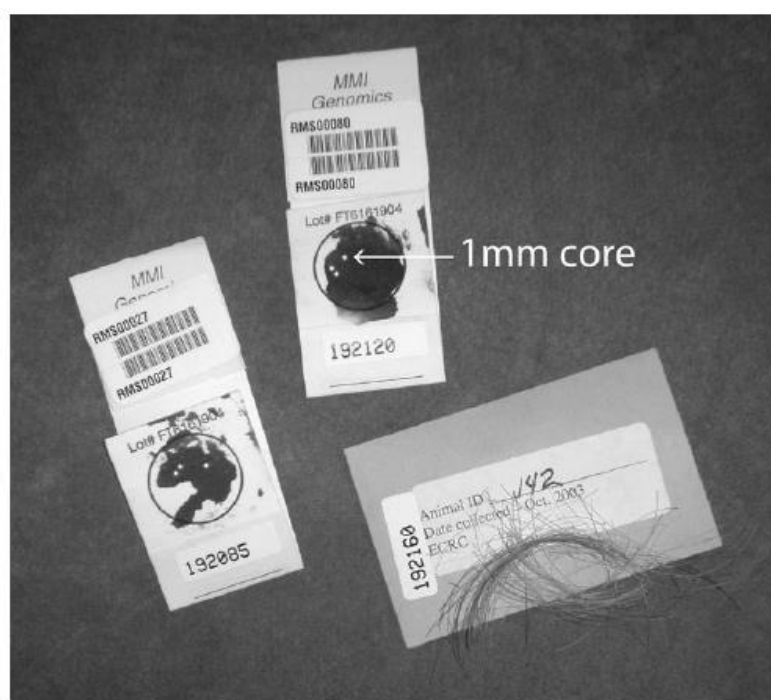
A *International Society for Animal Genetics* define 11 marcadores de microssatélites (BM2113, BM1824, ETH3, ETH10, ETH225, INRA23, SPS115, TGLA53, TGLA122, TGLA126 e TGLA227) presentes nos bovinos, que podem ser obtidos a partir de técnicas de PCR e que resultam em uma identificação genética única do animal a partir da observação do número de alelos presentes para cada marcador (VÁZQUEZ et al., 2004 e DAVIS et al., 2006). Davis et al. (2006) reforçam ainda que a probabilidade de se obter dois indivíduos com a mesma quantidade de alelos nos 11 marcadores é de 1 em 400.000.000. Com base nos trabalhos

apresentados, define-se que a identificação genética do animal deve ser uma entrada obrigatória em seu registro inicial na *Blockchain*.

De acordo com o estudo de caso apresentado por Davis et al. (2006), foi possível obter com 100% de precisão a identificação de um grupo de 34 animais. Foram extraídas duas amostras de cada um dos 34 animais para o teste de igualdade, mais 5 amostras individuais aleatórias para comparação com pelo e ainda mais 12 amostras extras com um grau de pureza baixo para verificar seu impacto na análise, totalizando 85 amostras. Foram utilizadas 31 amostras de pelo dos 34 animais do grupo, as quais já estavam armazenadas há um ano. Para testar a igualdade genética das amostras entre pelo e sangue, 26 amostras de pelo foram comparadas com 26 amostras duplas de sangue e mais 5 amostras de pelo com 5 amostras individuais de sangue (DAVIS et al., 2006). A partir desses resultados e corroborado pelos resultados do trabalho similar de Vázquez et al. (2004), é observável que duas amostras de sangue e uma de pelo apresentam um bom conjunto de material para a identificação genética e, portanto, foi definido que essa é a quantidade que deve ser armazenada ou analisada para o registro inicial do animal na *Blockchain*.

Para os casos de armazenamento do material genético pelo produtor, deve-se utilizar papéis com filtro especial para armazenamento de DNA com uma identificação do tipo código de barras (Figura 19). Esse código de barras deve ser o valor inserido como o ID genético do animal em seu primeiro registro na *Blockchain*, o que permitirá a rastreabilidade do próprio material genético no futuro. Felipe e Demanboro (2020, no prelo) sugerem que o tempo de vida máximo para uma entidade de rastreamento animal pode variar de 15 a 35 meses. Esse tempo é baseado na somatória de três períodos máximos distintos: o tempo de engorda até o abate do animal, o tempo de processamento da carne e o tempo máximo até o consumo, que são, respectivamente, 26, 1 e 8 meses (BORTOLUZZO; PEDRINOLA; MARTINS, 2011; MINERVA FOODS, 2015). Baseado nesses valores e para fins de análise futura, o tempo de armazenamento desses materiais foi definido como sendo de 36 meses.

Figura 19 – Amostras de sangue e pelo de um animal armazenadas em papéis com filtro especial para conservação de amostras genéticas



Fonte: Davis et al. (2006)

Já nos casos em que é possível obter o *fingerprint* genético logo no início do processo, o que é comum para produtores que trabalham com melhoramento genético na etapa da cria, é sugerida uma abordagem que gera uma identificação hexadecimal a partir do número de alelos encontrados para cada um dos 11 marcadores de microssatélites especificados pela *International Society for Animal Genetics* (VÁZQUEZ et al., 2004 e DAVIS et al., 2006). Vázquez et al. (2004) demonstraram que o número máximo de alelos por microssatélite variou entre 5 (BM1824 e TGLA126) e 16 (TGLA122) e, portanto, foi definido neste modelo que um valor hexadecimal máximo referente a cada microssatélite deve ser utilizado para cada um dos 11 marcadores a fim de formar uma identificação numérica para o animal (Tabela 6). Segundo Oliveira et al. (2006), não devem existir microssatélites com uma quantidade nula de alelos observados em um único indivíduo e, portanto, os índices hexadecimais propostos aqui devem sempre iniciar em 1. Essa condição permite que seja possível utilizar apenas um dígito hexadecimal por microssatélite se a contagem for iniciada em 0x0 representando 1 alelo e 0xF representando 16 alelos.

Na Tabela 6, observa-se o número máximo de alelos possíveis para cada um dos 11 microssatélites, obtendo uma variação numérica de identificações de

0x000000000000 a 0x94867E6EB4F, resultante da concatenação do valor máximo possível para cada um dos 11 microssatélites e que ocupa um tamanho de 44 bits. Em notação decimal, a identificação individual oferece uma variação numérica de 0 a 10.206.585.482.063 identificações numéricas individuais, o que se mostra extenso o suficiente para manter a unicidade da identificação animal na *Blockchain*.

Tabela 6 – Quantidade máxima de alelos observados por microssatélite

Microssatélites	Nº de alelos
BM2113	10
BM1824	5
ETH3	9
ETH10	7
ETH225	8
INRA23	15
SPS115	7
TGLA53	15
TGLA227	12
TGLA126	5
TGLA122	16

Fonte: Vázquez et al. (2004)

O trabalho de Vázquez et al. (2004) define que nove marcadores são suficientes para o estudo de caso proposto e, portanto, não indica a quantidade máxima de alelos para os marcadores TGLA53 e INRA23. Como Davis et al. (2006) utilizam todos os 11 marcadores e também apontam que somente o TGLA122 contém 16 alelos, define-se que os marcadores TGLA53 e INRA23 podem conter um valor máximo de 15 alelos, respeitando os resultados apresentados pelos autores (VÁZQUEZ et al., 2004 e DAVIS et al., 2006).

A Figura 20 mostra um exemplo de conversão de um *fingerprint* genético em valor hexadecimal a partir da quantidade detectada de alelos:

Figura 20 – Exemplo de conversão do resultado do número de alelos presentes para cada marcador de microssatélite em um *fingerprint* genético

Marcadores	BM2113	BM1824	ETH3	ETH10	ETH225	INRA23	SPS115	TGLA53	TGLA122	TGLA126	TGLA227
Qtd. Alelos	7	5	2	1	5	6	11	5	12	5	10
Conversão	6	4	1	0	4	5	A	4	B	4	9
ID = 0x641045A4B49											

Fonte: O autor

3.4 O *Smart Contract*

A síntese de todo o modelo proposto por este trabalho reside no *Smart Contract*. É na sua lógica que é proposto todo o controle do fluxo de dados de entrada e saída, bem como a criação e as atualizações das informações de rastreabilidade. Como premissas básicas para a construção do *Smart Contract*, ele deve ser capaz de:

- Criar os atores da cadeia.
- Criar animais (como um *asset*) vinculados a um *geneticId* inicial.
- Atualizar as informações do animal a partir de seu *geneticId*.
- Criar um produto (também um *asset*) de carne bovina que herda o *geneticId* de seu animal de origem e é identificado por um *productId*.
- Atualizar os produtos criados a partir de seu *productId*.
- Encerrar o ciclo do *asset*, impedindo novas inserções de dados.

Apesar de os *Smart Contracts* serem na prática programas de computador capazes de executar uma lógica de programação dentro da *Blockchain*, eles se apresentam como um paradigma não convencional de programação na maneira como interagem com a *Blockchain*, expedindo as transações para a rede (WOHRER; ZDUN, 2018). Os *Smart Contracts* podem ter funções clássicas de programas, mas a funcionalidade que os diferencia é exatamente essa capacidade de realizar as transações com a *Blockchain*. A essas funções especiais, aqui, dá-se o nome de funções de transação.

Para que o modelo proposto por este trabalho esteja de acordo com as especificações do *Smart Contract* apresentadas anteriormente, definimos as seguintes funções de transação:

- *createActor*: cria os atores da cadeia da carne bovina na *Blockchain* a partir de um ID único, como o nome do produtor ou empresa.

- *createAnimal*: a partir de um *geneticId* válido, cria um animal na *Blockchain* como um *asset* rastreável.
- *updateAnimal*: atualiza as informações de um animal já existente na *Blockchain* a partir de seu *geneticId*.
- *createProduct*: cria um produto a partir de um *geneticId* de um animal existente na *Blockchain* que já tenha sido abatido. O produto é identificado por um *productId*.
- *updateProduct*: atualiza as informações de um produto já existente na *Blockchain* a partir de seu *productId*.

A seguir, são apresentados os pseudocódigos de cada uma das funções de transação definidas acima.

O Pseudocódigo 1 apresenta o algoritmo da função de transação *createActor* presente no *Smart Contract*. Ela é responsável por criar os atores que irão interagir com a *Blockchain*. A título de validação, o algoritmo verifica se o ator já existe na *Blockchain* antes de efetivar a transação, que consiste sempre na inclusão de um registro na *Blockchain*.

Pseudocódigo 1 – Função de transação de criação de um ator da cadeia da carne bovina na *Blockchain*

Função de transação: createActor

Entrada:

id ← Id do ator da cadeia
c ← Contato do ator (opcional)

Função:

r ← estrutura de registro na Blockchain

Obter id da Blockchain

Se id = VERDADEIRO **então**

Retorna Erro

Senão

r.id = id

r.c = c

Adiciona transação com o registro r

Fim do Se

Fim

A função de transação *createAnimal*, apresentada no Pseudocódigo 2, recebe uma entrada composta pelos detalhes mínimos necessários à criação de um *asset* do tipo *Animal* na *Blockchain*. A função verifica, a partir do *geneticId*, se o animal já existe na *Blockchain* antes de enviar a transação, evitando animais duplicados. Essa função

também verifica se o *owner* (produtor/fazendeiro) do animal já existe na *Blockchain*, evitando que produtores sem o devido registro realizem entradas de animais no sistema.

Pseudocódigo 2 – Função de transação de criação de um animal na *Blockchain*

Função de transação: `createAnimal`

```

Entrada:
geneticId ← Fingerprint genético do animal
animalId  ← Identificação alternativa do animal (opcional)
breed     ← Raça do animal
location  ← Localização do animal
weight    ← Peso atual do animal
owner     ← Quem está em posse do animal no momento

Função:
r ← estrutura de registro na Blockchain
Obter geneticId da Blockchain
  Se geneticId = VERDADEIRO então
    Retorna Erro
  Fim do Se
Obter owner da Blockchain
  Se owner = FALSO então
    Retorna Erro
  Senão
    r.lifeStage = BREEDING
    r.geneticId = geneticId
    r.animalId  = animalId
    r.breed     = breed
    r.location  = location
    r.weight    = weight
    r.owner     = owner
    Adiciona transação com o registro r
  Fim do Se
Fim

```

A função de transação *updateAnimal*, apresentada no Pseudocódigo 3, tem o objetivo de atualizar as informações de um *asset* do tipo *Animal* já registrado na *Blockchain*. A partir de um *geneticId*, a função verifica se o animal existe e resgata os seus dados atuais, atualizando e concatenando os campos que foram enviados como entradas da função antes de submeter a transação de atualização à rede.

Pseudocódigo 3 – Função de transação de atualização de um animal existente na *Blockchain*

Função de transação: `updateAnimal`

```

Entrada:
geneticId      ← Fingerprint genético do animal
location       ← Localização do animal
weight        ← Peso atual do animal
lifeStage     ← Status do ciclo de vida do animal
owner         ← Quem está em posse do animal no momento
animalId      ← Identificação alternativa do animal (opcional)
vaccines      ← Vacinas aplicadas no animal (opcional)
diseases      ← Doenças pelas quais o animal foi acometido (opcional)
slaughterDate ← Data do abate (opcional)
transportedBy ← Transportadora no momento do registro (opcional)
slaughterhouse ← Frigorífico que abateu o animal (opcional)

Função:
r ← estrutura de registro na Blockchain
s ← estrutura do registro atual do animal na Blockchain
Obter geneticId da Blockchain
  Se geneticId = FALSO então
    Retorna Erro
  Fim do Se
Obter owner da Blockchain
  Se owner = FALSO então
    Retorna Erro
  Senão
    Obter s da Blockchain
    // Concatena os valores atuais com as entradas da função
    r.lifeStage = s.lifeStage
    Se r.lifeStage = DECEASED então
      Retorna Erro
    Fim do Se
    r.animalId = s.animalId + animalId
    r.location = s.location + location
    r.weight = s.weight + weight
    r.owner = s.owner + owner
    r.animalId = s.animalId + animalId
    r.vaccines = s.vaccines + vaccines
    r.diseases = s.diseases + diseases
    r.slaughterDate = slaughterDate
    r.transportedBy = s.transportedBy + transportedBy
    r.slaughterhouse = slaughterhouse
    Adiciona transação com o registro r
  Fim do Se
Fim

```

O Pseudocódigo 4 apresenta a função de transação *createProduct*, que é responsável pela criação de um *asset* do tipo Produto a partir de um outro *asset* do tipo Animal já registrado na *Blockchain*. A função verifica se o *geneticId* de entrada é um animal existente na *Blockchain*, bem como se o *productId* também já existe, evitando assim entradas duplicadas de produtos antes de submeter a transação à rede.

Pseudocódigo 4 – Função de transação de criação de um produto na *Blockchain*

Função de transação: `createProduct`

```

Entrada:
productId      ← Código do produto (código de barras ou QR code)
geneticId      ← Fingerprint genético do animal de origem
productType    ← Tipo do produto (corte ou carne processada)
productStatus  ← Status atual do produto
location       ← Localização atual do produto
weight        ← Peso do produto
processingIndustry ← Identificação da indústria de processamento (opcional)

Função:
r ← estrutura de registro na Blockchain
Obter geneticId da Blockchain
  Se geneticId = FALSO então
    Retorna Erro
  Fim do Se
Obter productId da Blockchain
  Se productId = VERDADEIRO então
    Retorna Erro
  Senão
    r.productId = productId
    r.productType = productType
    r.productStatus = productStatus
    r.location = location
    r.weight = weight
    r.processingIndustry = processingIndustry
    Adiciona transação com o registro r
  Fim do Se
Fim

```

A função de transação *updateProduct* é apresentada no Pseudocódigo 5. Ela executa uma ação similar à função *updateAnimal*, obtendo o registro mais atual do produto na *Blockchain* a partir de seu *productId*, atualizando e concatenando as novas informações presentes na entrada da função. Em seguida, as informações atualizadas são enviadas à rede em uma nova transação.

Pseudocódigo 5 – Função de transação de atualização de um produto existente na *Blockchain*

Função de transação: updateProduct

```

Entrada:
productId      ← Código do produto (código de barras ou QR code)
productStatus  ← Status atual do produto
location       ← Localização atual do produto
weight        ← Peso do produto
transportedBy  ← Transportadora no momento do registro (opcional)
retailerSeller ← Local da venda do produto no varejo
subtype       ← Subtipo do produto
price         ← Preço atual
productIssues  ← Problemas apresentados pelo produto

Função:
r ← estrutura de registro na Blockchain
s ← estrutura do registro atual do produto na Blockchain
Obter productId da Blockchain
  Se productId = FALSO então
    Retorna Erro
  Senão
    // Concatena os valores atuais com as entradas da função
    Obter s da Blockchain
    r.productStatus = productStatus
    Se r.productStatus = EXPIRED então
      Retorna Erro
    Fim do Se
    r.location = s.location + location
    r.weight = s.weight + weight
    r.transportedBy = s.transportedBy + transportedBy
    r.retailerSeller = s.retailerSeller + retailerSeller
    r.subtype = s.subtype + subtype
    r.price = s.price + price
    r.productIssues = s.productIssues + productIssues
    Adiciona transação com o registro r
  Fim do Se
Fim

```

O *Smart Contract* também deve conter três funções simples adicionais, responsáveis por obter todas as informações atuais dos *assets* registrados na *Blockchain* e verificar se o período de 36 meses definido no capítulo 3 como o tempo máximo de armazenamento desde a criação do *asset* já expirou, tomando as decisões adequadas para cada situação:

- *getAnimalInfo*: retorna todas as informações atuais de um animal registrado na *Blockchain*. A consulta deve ser feita a partir do *geneticId* ou do *animalId*, sendo que o segundo pode obter mais de um resultado.
- *getProductInfo*: retorna todas as informações atuais de um produto registrado na *Blockchain*. A consulta deve ser feita a partir do *productId*.

- *validateEndOfLife*: valida se o produto ou animal já existe há 36 meses ou mais na *Blockchain*. Se sim, ajusta o estado atual, impedindo novas escritas. Essa função deve ser executada dentro do *Smart Contract* sempre que ele for acessado por um *DApp*.

Todas as funções de transação e as funções simples do tipo *getter* descritas anteriormente provêm as funcionalidades necessárias para a implementação dos *DApps* que serão utilizados pelos atores da cadeia da carne bovina na interação com a *Blockchain*.

3.5 Protótipo funcional de um sistema de rastreabilidade baseado em *Blockchain*

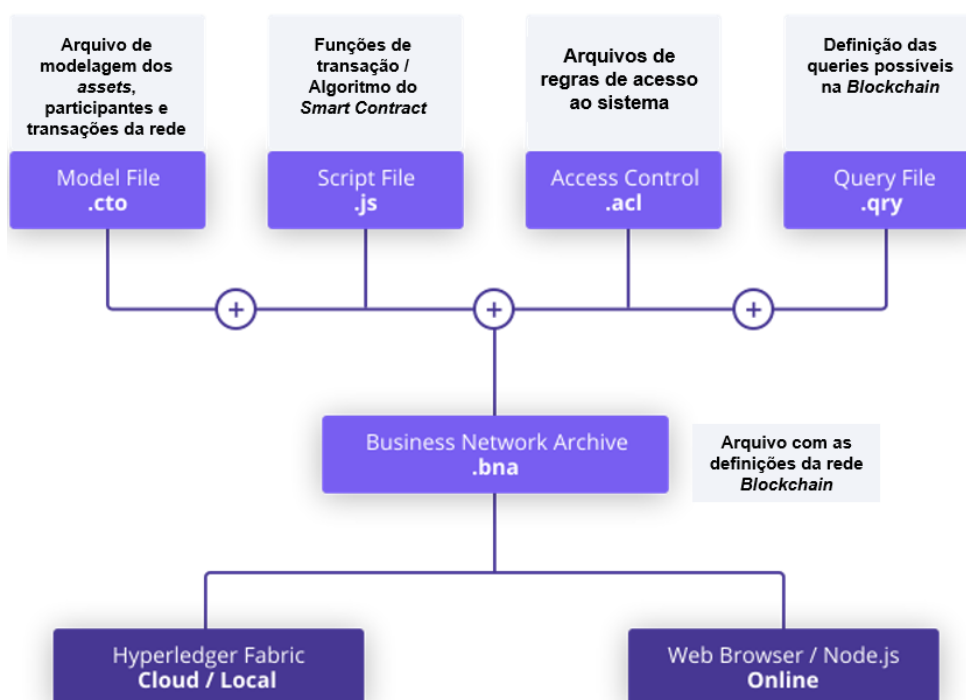
Para validar e demonstrar a factibilidade da proposta de rastreabilidade para a cadeia da carne bovina apresentada neste trabalho, foi desenvolvido um protótipo de um sistema baseado em *Blockchain*. Para esse protótipo, foi escolhida a plataforma permissionada *Hyperledger Fabric*. O *Hyperledger* é um projeto de código aberto criado pela IBM e mantido pela *Linux Foundation* que fornece toda a estrutura necessária para a construção de um sistema de *Blockchain* totalmente flexível. Apesar de permissionada, a *Blockchain* do *Hyperledger* conta com uma arquitetura modular, segura e independente de criptomoedas, além de possuir uma vasta quantidade de APIs para o desenvolvimento não somente do sistema, mas de *DApps* que venham a interagir com a *Blockchain* (BLUMMER et al., 2018).

Para a criação do ambiente de testes da *Blockchain*, foi utilizada a ferramenta *Hyperledger Composer*, oferecida como uma *sandbox* para protótipos de sistema baseados no *Hyperledger Fabric*. O *Hyperledger Fabric* possui uma arquitetura simples, que se baseia em alguns blocos e arquivos de configuração, conforme é descrito a seguir e apresentado na Figura 21:

- *.cto*: onde é feita a modelagem e a descrição dos *assets* dos participantes e das transações que estarão disponíveis na rede *Blockchain*.
- *.js*: código em *JavaScript* que é de fato o *Smart Contract*. Nele são programadas todas as funções simples e de transação que estarão disponíveis no contrato.
- *.acl*: arquivo com as regras de acesso ao sistema. É aqui que as permissões de leitura e escrita e o acesso a determinadas funções são configurados.

- .qry: chamadas para consultas na *Blockchain*. Aqui podem ser definidas *queries* padrão que serão acessadas pelos *DApps* ou outros componentes que interajam com a *Blockchain*.
- .bna: arquivo que contém as definições gerais da rede *Blockchain*. Pode ser usado para realizar o *deploy* da rede em outros servidores.
- Cloud/Local: computador no qual está instalada toda a estrutura da plataforma do *Hyperledger Fabric* e do *Composer*.
- Web Browser: *front-end* de testes e configuração do *Hyperledger Composer* para validação do protótipo.

Figura 21 – Estrutura organizacional do *Hyperledger Composer*



Fonte: Hyperledger Composer⁷ (2018, tradução nossa)

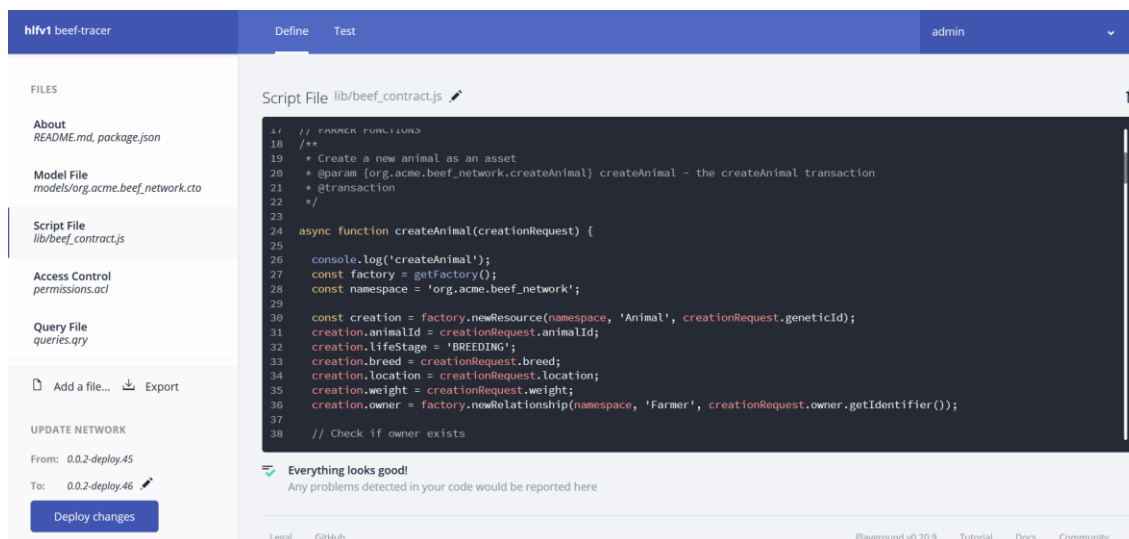
Para a instalação do sistema do *Hyperledger Fabric* e da ferramenta *Hyperledger Composer*, foi utilizada uma instância de um servidor na nuvem hospedada no serviço *Vultr*. As configurações da máquina escolhida foram:

⁷ Disponível em: <<https://hyperledger.github.io/composer/latest/introduction/introduction.html>>. Acesso em: 6 mar. 2020.

- CPU de um núcleo.
- Memória RAM de 2 GB.
- HD SSD de 55 GB.
- Sistema Operacional *Ubuntu Server 18.04 x64*.
- Franquia de dados de 2 TB.

Para a configuração do ambiente de desenvolvimento, foi utilizado o guia de instalação fornecido pela própria documentação do *Hyperledger Composer* (INSTALLING..., 2018). A seguir, a Figura 22 mostra o ambiente do *Hyperledger Composer* para definição dos arquivos já em execução e a Figura 23 apresenta o ambiente de testes do sistema, em que já é possível serem vistos os atores da cadeia criados, o menu de *assets* e o botão para submeter uma transação à *Blockchain*. Os códigos-fonte de todos os arquivos utilizados na construção do protótipo apresentado neste trabalho estão disponíveis no Apêndice.

Figura 22 – Ambiente de configuração e definição dos arquivos do *Hyperledger Composer*



Fonte: O autor

Figura 23 – Ambiente de testes da *Blockchain* do *Hyperledger Fabric* com os atores da cadeia da carne, assets e botão de submissão de transações

The screenshot displays the Hyperledger Fabric Playground interface. On the left, a sidebar menu is visible with sections for PARTICIPANTS, ASSETS, and TRANSACTIONS. Under PARTICIPANTS, there are sub-sections for Consumer, Farmer, ProcessingIndustry, Retail, Slaughterhouse, and Transportation. Under ASSETS, there are sub-sections for Animal and Product. Under TRANSACTIONS, there is a sub-section for All Transactions. A blue button labeled 'Submit Transaction' is located at the bottom of the sidebar.

The main content area is titled 'Participant registry for org.acme.beef_network.Consumer' and includes a '+ Create New Participant' button. It contains a table with two columns: 'ID' and 'Data'. The table lists two participants: 'Consumidor_A' and 'Consumidor_B'. Each entry shows a JSON object with 'Sclass' and 'consumerId' fields. To the right of each entry are edit and delete icons.

ID	Data
Consumidor_A	<pre>{ "Sclass": "org.acme.beef_network.Consumer", "consumerId": "Consumidor_A" }</pre>
Consumidor_B	<pre>{ "Sclass": "org.acme.beef_network.Consumer", "consumerId": "Consumidor_B" }</pre>

At the bottom of the interface, there are links for 'Legal', 'GitHub', 'Playground v0.20.9', 'Tutorial', 'Docs', and 'Community'.

Fonte: O autor

4. VALIDAÇÃO DO MODELO PROPOSTO

Foi proposto um estudo de caso do modelo de rastreabilidade da carne no ambiente do *Hyperledger Composer* detalhado no capítulo 3 com o objetivo de validar e verificar a factibilidade da implementação. Uma vez que o tamanho do banco de dados gerado para armazenar os registros da *Blockchain* tem um impacto direto na demanda por infraestrutura e, conseqüentemente, nos custos, um experimento para avaliar o peso de cada registro animal no tamanho do banco também foi realizado.

4.1 Estudo de caso do modelo

O ambiente de testes do *Hyperledger Composer* permite o envio de transações à *Blockchain* a partir das funções de transação criadas no *Smart Contract*, que foi desenvolvido utilizando a linguagem *JavaScript* a partir dos pseudocódigos apresentados no subitem 3.4. Na Figura 24, é apresentada a transação de criação do animal com dados básicos, como seu proprietário (Fazendeiro_1), raça (Nelore), localização (Altair/SP) e peso (28,3 kg). É possível verificar que a identificação genética na transação (*geneticId*) obedece ao padrão sugerido por este trabalho.

Figura 24 – Ferramenta de submissão das transações à *Blockchain* no ambiente de testes do *Hyperledger Composer*

Submit Transaction

Captura Retangular

Transaction Type createAnimal

JSON Data Preview

```
1 {
2   "$class": "org.acme.beef_network.createAnimal",
3   "geneticId": "0x752156b5ca",
4   "owner": "resource:org.acme.beef_network.Farmer#Fazendeiro_1",
5   "breed": "Nelore",
6   "location": "Altair/SP",
7   "weight": "28.3 Kg"
8 }
```

Optional Properties



Just need quick test data? [Generate Random Data](#) Cancel Submit

Fonte: O autor

É possível observar na Figura 25 que, a partir do menu de Animais visto na Figura 24, um *asset* do tipo Animal foi criado de fato na *Blockchain* por meio de uma identificação genética e seus dados iniciais de rastreio. Nota-se que o animal foi registrado na etapa de cria (*breeding*):

Figura 25 – Registro do animal na *Blockchain* e sua visualização com as informações atuais do menu “Animal” na ferramenta

Date, Time	Entry Type	Participant	
2020-05-23, 19:00:53	createAnimal	admin (NetworkAdmin)	view record

ID	Data	
0x752156b5ca	<pre>{ "\$class": "org.acme.beef_network.Animal", "geneticId": "0x752156b5ca", "owner": "resource:org.acme.beef_network.Farmer#Fazendeiro_1", "lifeStage": "BREEDING", "breed": "Nelore", "location": "Altair/SP", "weight": "28.3 Kg" }</pre>	 

Collapse

Fonte: O autor

A seguir é mostrado o resultado do estudo de caso do processo completo de rastreio de um animal na cadeia da carne bovina desde seu nascimento até a venda no varejo utilizando *Blockchain* e *Smart Contracts*. Para tanto, faz-se uso de uma sequência de imagens retiradas da ferramenta *Hyperledger Composer* hospedada na nuvem e desenvolvida a partir do modelo proposto neste trabalho:

- a) Figura 26 e Figura 27: transação de atualização do animal é enviada à *Blockchain*. O animal procede da etapa de cria (*breeding*) para a recria (*rearing*). São atualizadas as informações de localização (de Altair/SP para Sumaré/SP), proprietário (Fazendeiro_2), peso (agora 102,43 kg), ID do animal (identificação por brinco RFID 0003658125896547), transportadora responsável (Transportadora_AA) e vacinas aplicadas (febre aftosa).

Figura 26 – Informações da transação de atualização do animal

Submit Transaction

Transaction Type updateAnimal ▼

JSON Data Preview

```

1  {
2    "$class": "org.acme.beef_network.updateAnimal",
3    "geneticId":
4    "resource:org.acme.beef_network.Animal#0x752156b5ca",
5    "owner": "resource:org.acme.beef_network.Farmer#Fazendeiro_2",
6    "transportedBy":
7    "resource:org.acme.beef_network.Transportation#Transportadora_AA",
8    "lifeStage": "REARING",
9    "weight": "102.43 Kg",
10   "location": "Sumaré/SP",
11   "animalId": "0003658125896547",
12   "vaccines": "Febre aftosa"
13 }
```

Fonte: O autor

Figura 27 – Animal já atualizado na *Blockchain* com as informações enviadas pela transação de atualização

ID	Data
0x752156b5ca	<pre style="margin: 0;"> { "\$class": "org.acme.beef_network.Animal", "geneticId": "0x752156b5ca", "owner": "resource:org.acme.beef_network.Farmer#Fazendeiro_2", "transportedBy": "resource:org.acme.beef_network.Transportation#Transportadora_AA", "lifeStage": "REARING", "breed": "Nelore", "location": "Altair/SP Sumaré/SP", "weight": "28.3 Kg 102.43 Kg", "animalId": "0003658125896547", "vaccines": "Febre aftosa" }</pre>

Collapse

Fonte: O autor

- b) Figura 28: animal já atualizado com informações da próxima etapa, a de engorda (*fattening*). Aqui, novamente, o proprietário passa a ser outro (Fazendeiro_3), o peso do animal é atualizado para 350,77 kg, a localização é agora Santa Helena/GO e vacinas contra botulismo e leptospirose foram aplicadas. A transportadora ainda é a mesma.

Figura 28 – Animal já atualizado com as informações na etapa de engorda

0x752156b5ca	<pre> { "\$class": "org.acme.beef_network.Animal", "geneticId": "0x752156b5ca", "owner": "resource:org.acme.beef_network.Farmer#Fazendeiro_3", "transportedBy": "resource:org.acme.beef_network.Transportation#Transportadora_AA", "lifeStage": "FATTENING", "breed": "Nelore", "location": "Altair/SP Sumaré/SP Santa Helena/GO", "weight": "28.3 Kg 102.43 Kg 350.77 Kg", "animalId": "0003658125896547 0003658125896547", "vaccines": "Febre aftosa Botulismo Leptospirose" } </pre>
--------------	---

Collapse

Fonte: O autor

- c) Figura 29: aqui, as informações do animal já abatido podem ser observadas, além de novos dados, como a identificação do frigorífico em que ocorreu o abate (Frigorífico_1), o peso final do animal (402,33 kg), uma identificação adicional do número do GTA, a transportadora responsável pela última movimentação (Transportadora_BB) e a data e a hora do abate.

Figura 29 – Dados do animal já registrados na *Blockchain* após o abate

ID	Data
0x752156b5ca	<pre> { "\$class": "org.acme.beef_network.Animal", "geneticId": "0x752156b5ca", "owner": "resource:org.acme.beef_network.Farmer#Fazendeiro_3", "transportedBy": "resource:org.acme.beef_network.Transportation#Transportadora_BB", "slaughterhouse": "resource:org.acme.beef_network.Slaughterhouse#Frigorifico_1", "lifeStage": "SLAUGHTERED", "breed": "Nelore", "location": "Altair/SP Sumaré/SP Santa Helena/GO São Paulo/SP", "weight": "28.3 Kg 102.43 Kg 350.77 Kg 402.33 Kg", "animalId": "0003658125896547 0003658125896547 SIL: 5647898541", "vaccines": "Febre aftosa Botulismo Leptospirose", "slaughterDate": "2020-05-24T01:14:06.331Z" } </pre>

Fonte: O autor

- d) Figura 30 e Figura 31: mostram o resultado de um passo importante, que é a criação de um *asset* do tipo Produto na *Blockchain*. Nesse caso, o produto é uma peça de carne e foi criado com uma identificação do tipo presente em códigos de barra. Uma observação importante é que o produto carrega o parâmetro *geneticId* 0x752156b5ca como uma informação herdada do *asset* do tipo Animal criado anteriormente.

Figura 30 – Transação de criação do produto vinculado à identificação genética de um animal já existente na *Blockchain*

Transaction Type createProduct ▼

JSON Data Preview

```
1  {
2    "$class": "org.acme.beef_network.createProduct",
3    "productId": "000388372623939 00 33",
4    "geneticId":
5    "resource:org.acme.beef_network.Animal#0x752156b5ca",
6    "slaughterhouse":
7    "resource:org.acme.beef_network.Slaughterhouse#Frigorifico_1",
8    "productStatus": "PROCESSING",
9    "productType": "BEEF",
10   "weight": "10.20 Kg",
11   "location": "São Paulo/SP"
12 }
```

Optional Properties

Just need quick test data? [Generate Random Data](#) Cancel Submit

Fonte: O autor

Figura 31 – Dados do produto já registrados na *Blockchain*

ID	Data
000388372623939 00 33	<pre>{ "\$class": "org.acme.beef_network.Product", "productId": "000388372623939 00 33", "geneticId": "resource:org.acme.beef_network.Animal#0x752156b5ca", "slaughterhouse": "resource:org.acme.beef_network.Slaughterhouse#Frigorifico_1", "productStatus": "PROCESSING", "productType": "BEEF", "weight": "10.20 Kg", "location": "São Paulo/SP" }</pre>

Fonte: O autor

- e) Figura 32: o produto originado do animal criado no estudo de caso já apresentado na *Blockchain* como estando à venda no varejo. Essa é a última etapa antes da venda ou expiração, quando o *asset* deixa de ser atualizado e passa a ser somente leitura. É possível observar que a peça inteira de carne criada inicialmente foi transformada em uma peça menor com peso de 1,5 kg. O local de venda (Supermercado_A, São Paulo/SP) e o preço (R\$ 54,90) também são informações contidas nessa atualização do produto.

Figura 32 – Produto à venda no varejo derivado da peça maior com as informações atualizadas

ID	Data
000388372623939 00 33	<pre>{ "\$class": "org.acme.beef_network.Product", "productId": "000388372623939 00 33", "geneticId": "resource:org.acme.beef_network.Animal#0x752156b5ca", "slaughterhouse": "resource:org.acme.beef_network.Slaughterhouse#Frigorifico_1", "retailSeller": "resource:org.acme.beef_network.Retail#Supermercado_A", "transportedBy": "resource:org.acme.beef_network.Transportation#Transportadora_BB", "processingCompany": "resource:org.acme.beef_network.ProcessingIndustry#ProcIndustry_1", "productStatus": "ON_SALE", "productType": "BEEF", "subType": "Picanha Pacote", "weight": "1.5 Kg", "location": "São Paulo/SP", "price": "R\$ 54.90" }</pre>

Fonte: O autor

A ferramenta também permite visualizar em ordem cronológica todas as transações que foram submetidas e aprovadas pelo consenso da *Blockchain*. A partir do menu *All Transactions* visto na Figura 23, são exibidos todos os registros históricos da *Blockchain*. Na Figura 33, é possível observar os registros na *Blockchain* de criação do animal, atualização do animal e criação do produto:

Figura 33 – Registros históricos das transações na *Blockchain*

Date, Time	Entry Type	Participant	
2020-05-23, 19:14:39	updateAnimal	admin (NetworkAdmin)	view record
2020-05-23, 19:09:51	updateAnimal	admin (NetworkAdmin)	view record
2020-05-23, 19:03:20	RemoveAsset	admin (NetworkAdmin)	view record
2020-05-23, 19:00:53	createAnimal	admin (NetworkAdmin)	view record
2020-05-23, 22:23:28	UpdateAsset	admin (NetworkAdmin)	view record
2020-05-23, 22:19:21	createProduct	admin (NetworkAdmin)	view record
2020-05-23, 22:15:52	updateAnimal	admin (NetworkAdmin)	view record
2020-05-23, 19:14:39	updateAnimal	admin (NetworkAdmin)	view record
2020-05-23, 19:09:51	updateAnimal	admin (NetworkAdmin)	view record

Fonte: O autor.

Cada um dos registros pode ser verificado em detalhes com todas as informações que foram registradas na transação. Na Figura 34, é possível observar esses detalhes no menu *All Transactions* ao abrir a opção *view record*:

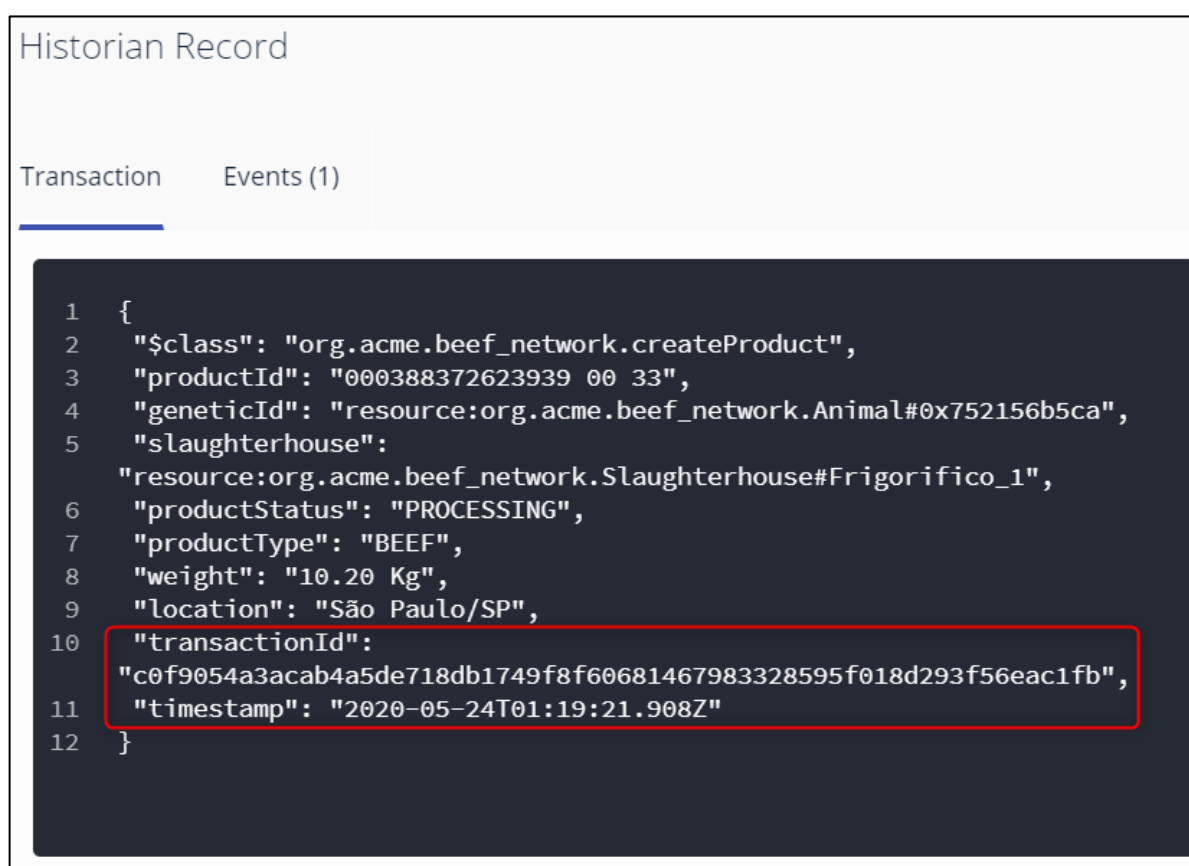
Figura 34 – Opção de visualização dos detalhes das transações na *Blockchain*

Date, Time	Entry Type	Participant	
2020-05-23, 22:23:28	UpdateAsset	admin (NetworkAdmin)	view record
2020-05-23, 22:19:21	createProduct	admin (NetworkAdmin)	view record
2020-05-23, 22:15:52	updateAnimal	admin (NetworkAdmin)	view record
2020-05-23, 19:14:39	updateAnimal	admin (NetworkAdmin)	view record
2020-05-23, 19:09:51	updateAnimal	admin (NetworkAdmin)	view record

Fonte: O autor

Esse é o registro histórico da transação no momento em que ela foi aceita pelo mecanismo de consenso da *Blockchain* e, além das informações inseridas pela aplicação que originou a transação, também observamos um dado de *transactionId*, que é o *hash* do bloco gerado por essa transação, também contendo o *hash* do bloco da transação anterior. A transação contém também o *timestamp*, que corresponde às informações de data e hora do momento em que a transação foi registrada na *Blockchain* (Figura 35).

Figura 35 – Registro histórico de uma transação



```
1  {
2    "$class": "org.acme.beef_network.createProduct",
3    "productId": "000388372623939 00 33",
4    "geneticId": "resource:org.acme.beef_network.Animal#0x752156b5ca",
5    "slaughterhouse":
6      "resource:org.acme.beef_network.Slaughterhouse#Frigorifico_1",
7    "productStatus": "PROCESSING",
8    "productType": "BEEF",
9    "weight": "10.20 Kg",
10   "location": "São Paulo/SP",
11   "transactionId":
12     "c0f9054a3acab4a5de718db1749f8f60681467983328595f018d293f56eac1fb",
13   "timestamp": "2020-05-24T01:19:21.908Z"
14 }
```

Fonte: O autor

4.2 Impactos do tamanho do banco de dados

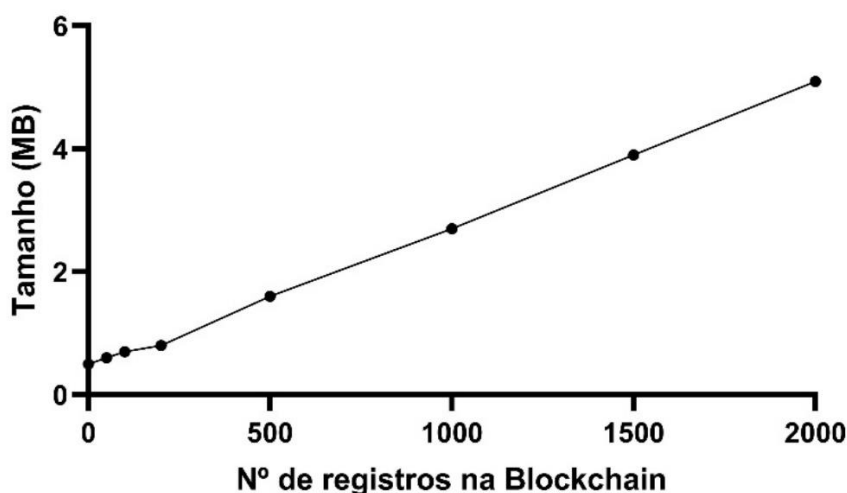
Para o experimento, foi observado qual era o tamanho do banco de dados utilizado pelo *Hyperledger*, o *CouchDB*, em *megabytes* (MB) para uma determinada quantidade de registros.

Utilizando um *script* em *Python*, foram inseridos automaticamente 1, 50, 100, 200, 500, 1.000, 1.500 e 2.000 registros na *Blockchain*. Todos os registros continham o mesmo conjunto de dados de animal, sendo eles:

- ID genético: Número variando de 1 a 2.000.
- Proprietário: Fazendeiro_1.
- Raça: Nelore.
- Localização: Santa Helena de Goiás/GO.
- ID animal: 00155987BED47FEC | 121551BDED47FEC.
- Vacinas: Febre Aftosa | Botulismo | Clostridioses.
- Doenças: Leptospirose.
- Data do abate: 2020-08-23T18:40:07.555Z.
- Peso: 106.51 Kg.

A Figura 36 mostra o resultado do experimento acima:

Figura 36 – Tamanho do banco de dados em relação ao número de registros inseridos na *Blockchain*



Fonte: O autor

Observa-se que o crescimento do tamanho do banco de dados é linear em relação à quantidade de registros na *Blockchain*. Para uma *Blockchain* com 2.000 registros de animais, o banco ficou com um tamanho de 5,1 MB, resultando que o tamanho de cada registro foi de 2,67 KB. Os registros foram inseridos com uma quantidade limitada de dados, mas na prática esse número pode variar dependendo

do tipo de registro. Dessa forma, partindo das informações sobre animais e produtos apresentadas no capítulo 3, considerando a variação que pode haver no tamanho dos registros mais um *overhead* adicional imposto pela *Blockchain*, é possível estimar que um registro individual teria um tamanho máximo de 3 vezes o observado no experimento, ou seja, 7,83 KB. Com esse resultado, é possível estimar o tamanho final do banco de dados apenas multiplicando o peso individual de cada registro (7,83 KB) pela quantidade estimada de registros na *Blockchain*. A partir disso e considerando uma extrapolação do tamanho do banco de dados a fim de armazenar dez registros de cada um dos 1,65 bilhões de animais de todo o rebanho mundial (ABIEC, 2019), os dados poderiam ocupar mais de 120 *terabytes* (TB). Se o rebanho brasileiro de 213,5 milhões de cabeças (IBGE, 2018) fosse considerado no mesmo cenário de dez registros por animal, o tamanho do banco cairia para pouco mais de 15,56 TB. Esse valor pode ser gerenciável a partir de uma infraestrutura em nuvem ou ainda diluindo o banco em estruturas menores espalhadas através dos diversos nós da rede, assim como é feito em outras redes de registros distribuídos, como o *Tangle* (POPOV, 2018).

Os experimentos realizados permitiram estimar um tamanho individual de registro na *Blockchain* de 7,83 KB com tamanhos de banco de dados podendo variar de 15,56 a 120 TB, que podem ser considerados pequenos para os moldes atuais de consumo e armazenamento de dados, uma vez que produzimos diariamente mais de 2,5 quintilhões de *bytes* (MAUGEY; TONI, 2020). Entretanto, na prática, o tamanho de um rebanho rastreado ficaria limitado à localidade, mercado, exportação ou importação, etc., e, portanto, o banco de dados deve ser bem menor do que isso.

4.3 Impactos financeiros

Em sistemas descentralizados, como é o caso da *Blockchain*, os custos estão associados à manutenção e à comunicação entre os nós, algo que já está estabelecido e é conhecido quando a rede é formada. Dessa forma, não existe a necessidade de gastos adicionais para a construção de uma rede *Blockchain* exceto os já mencionados. Os custos poderiam ser considerados se houver um provedor central do serviço de rastreabilidade, que poderia, por sua vez, desenvolver os aplicativos descentralizados que farão a interface com a *Blockchain*. Dessa maneira, esses integradores poderiam gerar lucro ao fornecer serviços dessa espécie e,

consequentemente, haveria um custo para os atores da cadeia que contratarem o serviço. A implementação prática do modelo descrito neste trabalho também demandaria custos de desenvolvimento, que poderiam ser absorvidos pelos atores envolvidos e interessados na criação de tal sistema, ou ainda diluídos no serviço oferecido por um integrador.

Sistemas de *Blockchain* como o *Hyperledger* ou o *Ethereum* podem ser implementados com baixo custo de desenvolvimento (ANTONOPOULOS; WOOD, 2019), se mostrando alternativas viáveis no modelo aqui proposto para a cadeia da carne bovina.

5. ANÁLISE E CONCLUSÃO

A utilização de um sistema de *Blockchain* aliado aos *Smart Contracts* como ferramentas de rastreamento na cadeia da carne bovina se apresenta como uma alternativa viável aos métodos atuais. O sistema implementado no estudo de caso, apresentado no capítulo 4, pôde ser construído com relativa facilidade a partir do modelo descrito no capítulo 3, pois as definições dos atores envolvidos e a interação dos *assets* com a cadeia da carne bovina ficam evidentes e factíveis nos mais variados sistemas de *Blockchain* disponíveis, uma vez que seu detalhamento é independente de um sistema específico. Dessa maneira, o trabalho entrega um modelo que pode ser implementado nas mais diversas redes de *Blockchain* do mercado que possuam a capacidade de executar *Smart Contracts*, como o próprio *Hyperledger Fabric* utilizado no estudo de caso, *Ethereum*, *Cardano*, *Stellar*, entre outros. A imutabilidade dos dados e a transparência nas informações conferidas pela *Blockchain* são características fundamentais em sistemas de rastreabilidade de cadeias logísticas, atendendo às demandas do setor da carne bovina, principalmente com foco em exportação e segurança alimentar, em que essa exigência é ainda maior. Um dos aspectos mais positivos do modelo apresentado é a vinculação do DNA animal com o seu registro inicial na *Blockchain*, reforçando o elo frágil de sistemas similares de rastreabilidade que enfrentam a dificuldade de validar a consistência e a veracidade das informações que são enviadas à rede. A utilização do DNA é mencionada por Sander, Semejin e Mahr (2018) como uma solução promissora para a rastreabilidade da carne bovina, porém sem apresentar detalhes de implementação. Temas como a clonagem de DNA animal podem impactar negativamente essa técnica no futuro, mas são discussões complexas que envolvem questões ética e científicas de longo prazo.

Sistemas de rastreabilidade para bens específicos utilizando *Blockchain* e *Smart Contracts*, como o apresentado neste trabalho, ou ainda em trabalhos similares como de Salah et al. (2019) e Mathisen (2018), podem ser estendidos a aplicações mais genéricas, buscando entregar modelos comuns que possam atender a uma variação maior de mercados e produtos. Perboli, Musso e Rosano (2018) apresentam uma aplicação desse tipo para qualquer cadeia de suprimentos, assim como o trabalho de Wang et al. (2019), que utiliza conceitos genéricos de cadeias logísticas para modelar diversos *Smart Contracts* que poderiam atender outros mercados.

Conforme mostrado nos resultados do capítulo 4, o tamanho do banco de dados pode variar consideravelmente dependendo do tamanho do rebanho a ser controlado e da utilização da rede. Como o modelo apresentado no trabalho não detalha a distribuição dos dados aos atores envolvidos na cadeia, mais análises são necessárias para compreender melhor como se daria a implementação prática de um sistema dessa natureza em diversos cenários da cadeia da carne, além da demanda de memória não volátil em cada caso de aplicação. Isso poderia exigir uma distribuição mais detalhada dos nós da rede, similar aos modos de nós completos e nós leves utilizados em diversos sistemas de *Blockchain*. Casos de uso mais comuns e confinados a mercados específicos não precisariam se preocupar com o tamanho do banco de dados por conta do pequeno tamanho dos registros individuais, mas esse tema deve ser levado em conta para expansões do sistema de rastreabilidade em larga escala. Um banco de dados de apenas 2 GB poderia armazenar mais de 275 milhões de registros com um baixo impacto na infraestrutura necessária, permitindo que o sistema possa ser utilizado até mesmo em celulares com baixa capacidade de memória não volátil.

Com base nos resultados obtidos pelo estudo de caso e a análise do modelo e tecnologia empregados, a utilização de *Blockchain* e *Smart Contracts* como meio de controle dos registros de animais na cadeia da carne bovina se mostra uma opção viável aos métodos de rastreabilidade atuais, pois apresenta características importantes e de alto valor frente às necessidades do mercado, como a transparência total do processo, acessibilidade, segurança e confiabilidade. O vínculo do registro inicial do animal com o seu DNA se mostra uma técnica valiosa para garantir a rastreabilidade completa do consumidor final até a origem do produto. O baixo impacto financeiro e de infraestrutura em diversos casos de aplicação na cadeia da carne também suporta a factibilidade da utilização desse tipo de modelo como uma solução que pode ser aplicada no mercado sem reflexos diretos à operação atual.

A partir do modelo aqui descrito, faz-se necessário um detalhamento melhor dos custos de desenvolvimento e fornecimento de serviços de rastreabilidade para os mais diversos cenários da pecuária, bem como um estudo de como o tempo e o custo das transações da *Blockchain* podem influenciar as mais variadas possibilidades de implementação nos segmentos da cadeia da carne bovina.

O banco de dados associado à implementação do modelo proposto neste trabalho cresce de forma linear em termos do número de atores e transações

envolvidas. As dimensões que pode alcançar em um determinado contexto deve ser objeto de uma análise mais detalhada em trabalhos futuros, considerando que, a partir do momento em que o animal se torna um produto, cria-se uma árvore de possibilidades dependente dos cenários considerados.

REFERÊNCIAS BIBLIOGRÁFICAS

ABIEC - Associação Brasileira das Indústrias Exportadoras de Carne. BeefREPORT - Perfil da Pecuária no Brasil. **ABIEC**, São Paulo, 2019. Disponível em: <<http://www.abiec.org.br/Sumario2019.aspx>>. Acesso em: 20 abr. 2019.

AKITA, F. How does Bitcoin force consensus among Byzantine generals? **AkitaOnRails**, São Paulo, 01 nov. 2017. Disponível em: <<http://www.akitaonrails.com/2017/11/01/how-does-bitcoin-force-consensus-among-byzantine-generals>>. Acesso em: 01 mai. 2019.

ALMEIDA, L. SBR aposta no blockchain para conectar o agronegócio. **Organic NEWS Brasil**, 08 jun. 2019. Disponível em: <<https://organicsnewsbrasil.com.br/negocio/feiras-eventos/bio-brazil-fair-biofach-america-latina-2018/thinking-green/sbr-aposta-no-blockchain-para-conectar-o-agronegocio/>>. Acesso em: 05 set. 2020.

ANTONOPOULOS, A. M. **Mastering Bitcoin**. 2ª ed. Sebastopol: O'Reilly Media Inc., 2017.

ANTONOPOULOS, A. M.; WOOD, G. **Mastering Ethereum**. 1ª ed. Sebastopol: O'Reilly Media Inc., 2019.

AUSTRALIAN insurer NTI in blockchain beef traceability pilot. **Ledger Insights**, Austrália, 2019. Disponível em: <<https://www.ledgerinsights.com/nti-blockchain-food-traceability-beef/>>. Acesso em: 01 mai. 2020.

BAKKER, E.; DAGEVOS, H. Reducing Meat Consumption in Today's Consumer Society: Questioning the Citizen-Consumer Gap. **Journal of Agricultural and Environmental Ethics**, Basiléia, v. 25, p. 877-894, set. 2011. Disponível em: <<https://doi.org/10.1007/s10806-011-9345-z>>. Acesso em: 07 abr. 2019.

BARUSELLI, M. S. As diferenças do confinamento no Brasil e no mundo. **DSM**, 08 fev. 2019. Disponível em: <https://www.dsm.com/products/tortuga/pt_BR/homeblog/Desmistificando_a_realidade_brasileira.html>. Acesso em: 08 mar. 2020.

BASHAR, G.; HILL, G.; SINGHA, S.; MARELLA, P.; DAGHER, G.G.; XIAO, J. Contextualizing Consensus Protocols in Blockchain: A Short Survey. In: IEEE INTERNATIONAL CONFERENCE ON TRUST, PRIVACY AND SECURITY IN INTELLIGENT SYSTEMS AND APPLICATIONS, 1., 2019, Los Angeles. **Proceedings...** Piscataway: IEEE, 2019. p. 190-195. Disponível em: <<https://doi.org/10.1109/TPS-ISA48467.2019.00031>>. Acesso em: 06 fev. 2020.

BASTIANI, A. O que é e como funciona o Proof of Stake? **Criptofacil**, 11 dez. 2019. Disponível em: <<https://www.criptofacil.com/o-que-e-e-como-funciona-o-proof-of-stake/>>. Acesso em: 08 mar. 2020.

BEEFLIEDGER: Blockchain tracking from paddock to plate. **QUT Blockchain**, 2020. Disponível em: <<https://research.qut.edu.au/blockchain/projects/beefledger/>>. Acesso em: 01 mai. 2020.

BLOCKCHAIN é usado em projeto de rastreamento de produção de carne bovina. **CIO**, 07 nov. 2019. Disponível em: <<https://cio.com.br/blockchain-e-usado-em-projeto-de-rastreamento-de-producao-de-carne-bovina/>>. Acesso em: 01 mai. 2020.

BLUMMER, T.; BOHAN, S.; BOWMAN, M.; CACHIN, C.; GASKI, N.; GEORGE, N.; GRAHAM, G.; HARDMAN, D.; JAGADEESAN, R.; KEITH, T.; KHASANSHYN, R.; KRISHNA, M.; KUHRT, T.; HORS, A.L.; LEVI, J.; LIBERMAN, S.; MENDEZ, E.; MIDDLETON, D.; MONTGOMERY, H.; O'PREY, D.; REED, D.; TEIS, S.; VOELL, D.; WALLACE, G.; YAN B. An Introduction to Hyperledger. **HYPERLEDGER**, jul. 2018. Disponível em: <<https://www.hyperledger.org/learn/white-papers>>. Acesso em: 26 nov. 2019.

BORTOLUZZO, A. B., PEDRINOLA, P. D., MARTINS, S. R. Tempo Ideal Para Abate de Gado de Corte Via Maximização do Lucro. **Insper Working Paper**, 2011. Disponível em: <https://www.insper.edu.br/en/wp-content/uploads/2013/12/2011_wpe239.pdf>. Acesso em: 04 jun. 2019.

BRUGNANO, F. de M. L.; BRANDÃO, P. E.; RICHTZENHAIN, L. J.; ARAÚJO, E. S.; BOANOVA, A. B.; TELLES, E. O.; SILVA, S. O. S.; SANCHES, S. A.; BALIAN, S. de C. Levantamento de custo da reação em cadeia pela polimerase (PCR) em amostras de carne. **Revista Analytica**, São Paulo, v. 13, n. 72, p. 62-68, set. 2014. Disponível em: <<https://repositorio.usp.br/item/002686274>>. Acesso em: 30 abr. 2020.

CASTRO, R. J. S.; OHARA, A.; AGUILAR, J. G. S.; DOMINGUES, M. A. F. Nutritional, functional and biological properties of insect proteins: Processes for obtaining, consumption and future challenges. **Trends in Food Science & Technology**, [s.l.], v. 76, p. 82-89, jun. 2018. Disponível em: <<https://doi.org/10.1016/j.tifs.2018.04.006>>. Acesso em: 07 abr. 2019.

CHRISTIDIS, K.; DEVETSIKIOTIS, M. Blockchains and Smart Contracts for the Internet of Things. **IEEE Access**, Piscataway, v. 4, p. 2292-2303, mai. 2016. Disponível em: <<https://doi.org/10.1109/ACCESS.2016.2566339>>. Acesso em: 28 abr. 2019.

MINERVA FOODS. Como conservar a carne bovina. In: **Minerva Blog**, 25 ago. 2015. Disponível em: <<https://portal.minervafoods.com/blog/post/como-conservar-carne-bovina>>. Acesso em: 06 jun. 2019.

CROSBY, M.; NACHIAPPAN; PATTANAYAK, P.; VERMA. S.; KALYANARAMAN, V. BlockChain Technology: Beyond Bitcoin. **Applied Innovation Review**, Berkeley, n. 2, p. 6-19, jun. 2016. Disponível em: <<http://scet.berkeley.edu/wp-content/uploads/AIR-2016-Blockchain.pdf>>. Acesso em: 27 mar. 2019.

DAVIS, J.; TATUM, J.; HELLER, J.; JOHNSTON, E.; FANTIN, D.; CUNNINGHAM, W. Use of DNA Fingerprinting for Verifying Identity of Individual Cattle Within a Forty-Eight Hour Response Period. **The Professional Animal Scientist**, Champagne, v. 22, n. 2, p. 139-143, abr. 2006. Disponível em: <[https://doi.org/10.15232/S1080-7446\(15\)31078-0](https://doi.org/10.15232/S1080-7446(15)31078-0)>. Acesso em: 10 ago. 2019.

DOUCEUR, J. R. The Sybil Attack. In: INTERNATIONAL WORKSHOP ON PEER-TO-PEER SYSTEMS, 1., 2002, Cambridge. **Proceedings...** Berlim: Springer, 2002. p. 251-260. Disponível em: <https://doi.org/10.1007/3-540-45748-8_24>. Acesso em: 01 mai. 2019.

DUNN, J. E. Blockchain Bandit stole \$54 million of Ethereum by guessing weak keys. **naked security by SOPHOS**, 25 abr. 2019. Disponível em: <<https://nakedsecurity.sophos.com/2019/04/25/blockchainbandit-stole-54-million-of-ethereum-by-guessing-weak-keys/>>. Acesso em: 01 mai. 2019.

DWORKIN, M. J. SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. **FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION**, Gaithersburg, v. 202, p. 1-28, ago. 2015. Disponível em: <<https://doi.org/10.6028/NIST.FIPS.202>>. Acesso em: 28 abr. 2019.

FEI, P.; SHUI-SHENG, Q.; MIN, L. A Secure Digital Signature Algorithm Based on Elliptic Curve and Chaotic Mappings. **Circuits, Systems and Signal Processing**, v. 24, p. 585-597, out. 2005. Disponível em: <<https://doi.org/10.1007/s00034-005-2409-4>>. Acesso em: 01 mai. 2019.

FELIPPE, A. D.; DEMANBORO, A. C. Smart contracts and blockchain: an application model for traceability in the beef supply chain. In: BRAZILIAN TECHNOLOGY SYMPOSIUM, 5., 2019, Campinas. **Proceedings...** No prelo 2020.

FERRAREZE, R. R.; JUNIOR, S. S. B.; BAPTISTA, R. D. Modelo de gestão de resíduos: desafios e perspectivas do setor de frigoríficos. **DRd - Desenvolvimento Regional em Debate**, v. 8, n. 2, p. 68-88, jun. 2018. Disponível em: <<http://www.periodicos.unc.br/index.php/drd/article/view/1721>>. Acesso em: 27 abr. 2019.

GALVEZ, J. F.; MEJUTO, J. C.; SIMAL-GANDARA, J. Future challenges on the use of blockchain for food traceability analysis. **Trends in Analytical Chemistry**, v. 107, p. 222-232, out. 2018. Disponível em: <<https://doi.org/10.1016/j.trac.2018.08.011>>. Acesso em: 05 mar. 2019.

GEORGES, M.; LATHROP, M.; HILBERT, P.; MARCOTTE, A.; SCHWERS, A.; SWILLENS, S.; VASSART, G.; HANSET, R. On the use of DNA fingerprints for linkage studies in cattle. **Genomics**, Salt Lake, v. 6, n. 3, p. 461-474, mar. 1990. Disponível em: <[https://doi.org/10.1016/0888-7543\(90\)90476-B](https://doi.org/10.1016/0888-7543(90)90476-B)>. Acesso em: 19 set. 2019.

GRANDE, E. T. G.; VIEIRA, S. L. Beef traceability by radio frequency identification system in the production process of a slaughterhouse. **Journal of Information Systems and Technology Management**, São Paulo, v. 10, n. 1, p. 99-115, abr. 2013. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1807-17752013000100007>. Acesso em: 27 abr. 2019.

IBGE. Efetivo dos rebanhos, por tipo de rebanho. **Sistema IBGE de Recuperação Automática – SIDRA**, 2018. Disponível em: <<https://sidra.ibge.gov.br/tabela/3939#/n1/all/v/all/p/last%201/c79/2670/l/v,p+c79,t/resultado>>. Acesso em: 05 set. 2020.

IBM. About IBM Food Trust. **IBM Food Trust**, 2019. Disponível em: <<https://www.ibm.com/downloads/cas/8QABQBDR>>. Acesso em: 01 mai. 2020.

IBOPE – Instituto Brasileiro de Opinião Pública e Estatística. Pesquisa de opinião pública sobre vegetarianismo. **IBOPE inteligência**, abr. 2018. Disponível em: <http://www.svb.org.br/images/Documentos/JOB_0416_VEGETARIANISMO.pdf>. Acesso em: 07 abr. 2019.

INSTALLING Hyperledger Composer. **Hyperledger Composer**, 2018. Disponível em: <<https://hyperledger.github.io/composer/latest/installing/installing-index>>. Acesso em: 09 jul. 2019.

ISO – International Organization for Standardization. ISO 9000:2015: Quality management systems – Fundamentals and vocabulary. **Online Browsing Platform (OBP)**, Geneva, 2015. Disponível em: <<https://www.iso.org/obp/ui/#iso:std:iso:9000:ed-4:v1:en>>. Acesso em: 05 mai. 2019.

KAHN, M. A.; SALAH, K. IoT Security: Review, blockchain solutions, and open challenges. **Future Generation Computer Systems**, v. 82, p. 395-411, mai. 2018. Disponível em: <<https://doi.org/10.1016/j.future.2017.11.022>>. Acesso em: 05 mai. 2019.

KAMATH, R. Food Traceability on Blockchain: Walmart's Pork and Mango Pilots with IBM. **The JBBA**, Londres, v. 1, n. 1, p. 47-53, 12 jun. 2018. Disponível em: <[https://doi.org/10.31585/jbba-1-1-\(10\)2018](https://doi.org/10.31585/jbba-1-1-(10)2018)>. Acesso em: 01 mai. 2020.

KING, R. Top 10 Cryptocurrencies 2019: What's The Most Popular Cryptocurrency Today? **BitDegree**, 29 abr. 2019. Disponível em: <<https://www.bitdegree.org/tutorials/top-10-cryptocurrencies/>>. Acesso em: 01 mai. 2019.

LENG, K.; BI, Y.; JING, L.; FU, H.; NIEUWENHUYSE, I. V. Research on agricultural supply chain system with double chain architecture based on blockchain technology. **Future Generation Computer Systems**, v. 86, p. 641-649, set. 2018. Disponível em: <<https://doi.org/10.1016/j.future.2018.04.061>>. Acesso em: 15 abr. 2020.

MAESA, D. D. F.; MARINO, A.; RICCI, L. Uncovering the Bitcoin Blockchain: An Analysis of the Full Users Graph. In: INTERNATIONAL CONFERENCE ON DATA SCIENCE AND ADVANCED ANALYTIC, 3., 2016, Montreal. **Proceedings...** Piscataway: IEEE, 2016. p. 538-546. Disponível em: <<https://doi.org/10.1109/DSA.A.2016.52>>. Acesso em: 01 mai. 2019.

MAPA - Ministério da Agricultura, Pecuária e Abastecimento. Rastreabilidade Animal. **SISBOV**, 03 jan. 2017. Disponível em: <<http://www.agricultura.gov.br/assuntos/saude-animal-e-vegetal/saude-animal/rastreabilidade-animal>>. Acesso em: 27 abr. 2019.

MATHISEN, M. **The Application of Blockchain Technology in Norwegian Fish Supply Chains**. 2018. 75 f. Dissertação (Mestrado em Engenharia Mecânica) – Departamento de Engenharia Mecânica e Industrial, Norwegian University of Science and Technology, Trondheim, 2018.

MAUGEY, T.; TONI, L. Large Database Compression Based on Perceived Information. **IEEE Signal Processing Letters**, v. 27, p. 1735-1739, 2020. Disponível em: <<https://ieeexplore.ieee.org/document/9204845/>>. Acesso em: 31 dez. 2020.

NAKAMOTO, S. Bitcoin: A Peer-to-Peer Electronic Cash System. **Bitcoin**, 31 out. 2008. Disponível em: <<https://bitcoin.org/bitcoin.pdf>>. Acesso em: 05 mar. 2019.

OLIVEIRA, E. J.; PÁDUA, J. G.; ZUCCHI, M. I.; VENCOVSKY, R.; VIEIRA, M. L. C. Origin, evolution and genome distribution of microsatellites. **Genetics and Molecular Biology**, São Paulo, v. 29, n. 2, p. 294-307, 2006. Disponível em: <<http://dx.doi.org/10.1590/S1415-47572006000200018>>. Acesso em: 17 mai. 2020.

OLIVER, C. G.; RICOTTONE, A.; PHILIPPOPOULOS, P. Proposal for a fully decentralized blockchain and proof-of-work algorithm for solving NP-complete problems. **Cornell University Computer Science Department - Distributed, Parallel, and Cluster Computing**, Ithaca, 02 set. 2017. Disponível em: <<https://arxiv.org/abs/1708.09419v2>>. Acesso em: 01 mai. 2019.

OLSEN, P.; BORIT, M. The components of a food traceability system. **Trends in Food Science & Technology**, Cambridge, v. 77, p. 143-149, jul. 2018. Disponível em: <<https://doi.org/10.1016/j.tifs.2018.05.004>>. Acesso em: 27 abr. 2019.

ONU – Organização das Nações Unidas, Department of Economic and Social Affairs, Population Division. **World Population Prospects 2019: Highlights**, 2019. Disponível em: <https://population.un.org/wpp/Publications/Files/WPP2019_Highlights.pdf>. Acesso em: 07 abr. 2019.

PAWCZUK, L.; MASSEY, R.; SCHATSKY, D. Breaking blockchain open – Deloitte's 2018 global blockchain survey. **Deloitte**, 2018. Disponível em: <<https://www2.deloitte.com/content/dam/Deloitte/us/Documents/financial-services/us-fsi-2018-global-blockchain-survey-report.pdf>>. Acesso em 27 mar. 2019.

PERBOLI, G.; MUSSO, S.; ROSANO, M. Blockchain in Logistics and Supply Chain: A Lean Approach for Designing Real-World Use Cases. **IEEE Access**, Piscataway, v. 6, p. 62018-62028, 16 out. 2018. Disponível em: <<https://doi.org/10.1109/ACCESS.2018.2875782>>. Acesso em: 26 abr. 2020.

PIZZUTI, T.; MIRABELLI, G.; GRASSO, G.; PALDINO, G. MESCO (MEat Supply Chain Ontology): An ontology for supporting traceability in the meat supply chain. **Food Control**, v. 72, p. 123-133, fev. 2017. Disponível em: <<https://doi.org/10.1016/j.foodcont.2016.07.038>>. Acesso em: 27 abr. 2019.

POPOV, S. The Tangle. **IOTA Research Papers**, 30 abr. 2018. Disponível em: <https://assets.ctfassets.net/r1dr6vzfxhev/2t4uxvslqk0EUau6g2sw0g/45eae33637ca92f85dd9f4a3a218e1ec/iota1_4_3.pdf>. Acesso em: 12 set. 2020.

SAATH, K. C. O.; FACHINELLO, A. L. Crescimento da demanda mundial de alimentos e restrições do fator terra no Brasil. **Revista de Economia e Sociologia Rural**, Brasília, v. 56, n. 2, p. 196-210, jun. 2018. Disponível em: <<http://dx.doi.org/10.1590/1234-56781806-94790560201>>. Acesso em: 16 abr. 2019.

SALAH, K.; NIZAMUDDIN, N.; JAYARAMAN, R.; OMAR, M. Blockchain-Based Soybean Traceability in Agricultural Supply Chain. **IEEE Access**, Piscataway, v. 7, p. 73295-73305, 20 mai. 2019. Disponível em: <<https://doi.org/10.1109/ACCESS.2019.2918000>>. Acesso em 17 abr. 2020.

SALIMITARI, M.; CHATTERJEE, M. A Survey on Consensus Protocols in Blockchain for IoT Networks. **IEEE Internet of Things Journal**. Disponível em: <<https://arxiv.org/abs/1809.05613>>. Acesso em: 08 mar. 2020. No prelo 2020.

SANDER, F.; SEMEJIN, J.; MAHR, D. The acceptance of blockchain technology in meat traceability and transparency. **British Food Journal**, Bingley, v. 120, n. 9, p. 2066-2079, 03 set. 2018. Disponível em: <<https://doi.org/10.1108/BFJ-07-2017-0365>>. Acesso em: 23 fev. 2019.

SILVA, A. B. **A efetividade da utilização do controle realizado pela companhia integrada de desenvolvimento agrícola de Santa Catarina através da colocação de brincos em bovinos como meio de prova nos crimes de abigeato**. 2018. 64 f. Monografia (Graduação em Direito) - Faculdade de Ciências Sociais, Direito, Negócios e Serviços, Fundação UNISUL, Tubarão, 2018.

SILVA, A. L. da; MAFEI, L.; BORDIN, R. de A.; CUNHA, G. J. da. A rastreabilidade na cadeia da bovinocultura de corte brasileira. **Tekhne e Logos**, Botucatu, v. 9, n. 2, p. 20-33, set. 2018. Disponível em: <<http://revista.fatecbt.edu.br/index.php/tl/article/view/556>>. Acesso em: 27 abr. 2019.

SCARVADA, A. J.; BATALHA, M. O.; RIBEIRO, P. C. C. Aplicação de RFID na cadeia de carne bovina brasileira: estudos de caso. In: HESSEL, F. et al (org.). **Implementando RFID na cadeia de negócios: tecnologia a serviço da excelência**. 3ª ed. Porto Alegre: EDIPUCRS, 2014.

SHARMA, K.; JAIN, D. Consensus Algorithms in Blockchain Technology: A Survey. In: INTERNATIONAL CONFERENCE ON COMPUTING, COMMUNICATION AND NETWORKING TECHNOLOGIES, 10., 2019, Kanpur. **Proceedings...** Piscataway: IEEE, 2019. Disponível em: <<https://doi.org/10.1109/ICCCNT45670.2019.8944509>>. Acesso em 08 mar. 2020.

SHARMA, T. K. Permissioned and permissionless blockchains: a comprehensive guide. **Blockchain Council**, 13 nov. 2019. Disponível em: <<https://www.blockchain-council.org/blockchain/permissioned-and-permissionless-blockchains-a-comprehensive-guide/>>. Acesso em 01 mai. 2020.

SZABO, N. Smart Contracts. **Nick Szabo's E-Commerce and Security White Papers**, 1994. Disponível em: <<https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html>>. Acesso em: 26 mai. 2019.

_____. Formalizing and Securing Relationship on Public Networks. **First Monday - Peer-Reviewed Journal on the Internet**, 01 set. 1997. Disponível em: <<https://ojphi.org/ojs/index.php/fm/article/view/548/469#Contracts>>. Acesso em: 26 mai. 2019.

TIAN, F. An agri-food supply chain traceability system for China based on RFID & blockchain technology. In: INTERNATIONAL CONFERENCE ON SERVICE SYSTEMS AND SERVICE MANAGEMENT, 13., 2016, Kunming. **Proceedings...** Piscataway: IEEE, 2016. Disponível em: <<https://doi.org/10.1109/ICSSSM.2016.7538424>>. Acesso em: 25 de fev. de 2019.

TSE, D.; ZHANG, B.; YANG, Y.; CHENG, C. MU, H. Blockchain application in food supply information security. In: IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL ENGINEERING AND ENGINEERING MANAGEMENT, 2017, Cingapura. **Proceedings...** Piscataway: IEEE, 2017, p. 1357-1361. Disponível em: <<https://doi.org/10.1109/IEEM.2017.8290114>>. Acesso em: 27 abr. 2020.

UW Technology Aids in First-Ever Blockchain Beef Shipment. **University of Wyoming**, 07 fev. 2019. Disponível em: <<https://www.uwyo.edu/uw/news/2019/02/uw-technology-aids-in-first-ever-blockchain-beef-shipment.html>>. Acesso em: 01 mai. 2020.

VALLE, C. Pesquisa do IBOPE aponta crescimento histórico no número de vegetarianos no Brasil. **Sociedade Vegetariana Brasileira**, 20 mai. 2018. Disponível em <<https://www.svb.org.br/2469-pesquisa-do-ibope-aponta-crescimento-historico-no-numero-de-vegetarianos-no-brasil>>. Acesso em: 07 abr. 2019.

VANKAN, D. M.; BURNS, B. M. DNA fingerprinting – how it works and applications for the beef industry. **Proc. Assoc. Advmt. Anim. Breed. Genet.**, v. 12, 1997. Disponível em: <<http://www.livestocklibrary.com.au/handle/1234/5654>>. Acesso em: 10 ago. 2019.

VÁZQUEZ, J. F.; PÉREZ, T.; UREÑA, F.; GUDÍN, E.; ALBORNOZ, J.; DOMÍNGUEZ, A. Practical Application of DNA Fingerprinting To Trace Beef. **Journal of Food Protection**, v. 67, n. 5, p. 972-979, 2004. Disponível em: <http://meridian.allenpress.com/jfp/article-pdf/67/5/972/1671910/0362-028x-67_5_972.pdf>. Acesso em: 16 mai. 2020.

VERONESE, G. S.; CORREIA, M.; BESSANI, A. N.; LUNG, L. C.; VERISSIMO, P. Efficient Byzantine Fault-Tolerance. **IEEE Transactions on Computers**, Los Alamitos, v. 62, n. 1, p. 16-30, 15 nov. 2011. Disponível em: <<https://doi.org/10.1109/TC.2011.221>>. Acesso em: 01 mai. 2019.

WANG, S.; LI, D.; ZHANG, Y.; CHEN, J. Smart Contract-Based Product Traceability System in the Supply Chain Scenario. **IEEE Access**, Piscataway, v. 7, p. 115122-115133, 16 ago. 2019. Disponível em: <<https://doi.org/10.1109/ACCESS.2019.2935873>>. Acesso em: 28 mar. 2020.

WOHRER, M.; ZDUN, U. Smart contracts: security patterns in the ethereum ecosystem and solidity. In: INTERNATIONAL WORKSHOP ON BLOCKCHAIN ORIENTED SOFTWARE ENGINEERING, 1., 2018, Campobasso. **Proceedings...** Piscataway: IEEE, 2018, p. 2-7. Disponível em: <<https://doi.org/10.1109/IWB OSE.2018.8327565>>. Acesso em: 30 abr. 2020.

WU, K.; MA, Y.; HUANG, G.; LIU, X. A First Look at Blockchain-based Decentralized Applications. **Cornell University**, 03 set. 2019. Disponível em: <<https://arxiv.org/abs/1909.00939>>. Acesso em: 05 mai. 2020.

XIAO, Y.; ZHANG, N.; LOU, W.; HOU, Y. T. A Survey of Distributed Consensus Protocols for Blockchain Networks. **IEEE Communications Surveys & Tutorials**, p. 1-31, 28 jan. 2020. Disponível em: <<https://doi.org/10.1109/COMST.2020.2969706>>. Acesso em: 08 mar. 2020.

YANO, I. H.; SANTOS, E. H. dos; CASTRO, A. de; BERGIER, I.; SANTOS, P. M.; OLIVEIRA, S. R. de M.; ABREU, U. G. P. de. Modelo de rastreamento bovino via Smart Contracts com tecnologia Blockchain. **Embrapa**, dez. 2018. Disponível em: <<https://ainfo.cnptia.embrapa.br/digital/bitstream/item/188315/1/Modelo-rastreamento-CT-130.pdf>>. Acesso em: 18 jun. 2019.

ZHENG, Z.; XIE, S.; DAI, H.; CHEN, X.; WANG, H. An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. In: IEEE INTERNATIONAL CONGRESS ON BIG DATA, 6., 2017, Honolulu. **Proceedings...** Piscataway: IEEE, 2017, p. 558-563. Disponível em: <<https://doi.org/10.1109/BigDataCongress.2017.85>>. Acesso em: 01 mai. 2019.

APÊNDICES

Apêndice A – Código-fonte do *Smart Contract* em *JavaScript* utilizado no estudo de caso feito com o *Hyperledger Fabric*

```

1. // FARMER FUNCTIONS
2. /**
3.  * Create a new animal as an asset
4.  * @param {org.acme.beef_network.createAnimal} createAnimal - the createAnimal transaction
5.  * @transaction
6.  */
7.
8. async function createAnimal(request) {
9.
10.   console.log('createAnimal');
11.   const factory = getFactory();
12.   const namespace = 'org.acme.beef_network';
13.
14.   const creation = factory.newResource(namespace, 'Animal', request.geneticId);
15.   creation.animalId = request.animalId;
16.   creation.lifeStage = 'BREEDING';
17.   creation.breed = request.breed;
18.   creation.location = request.location;
19.   creation.weight = request.weight;
20.   creation.owner = factory.newRelationship(namespace, 'Farmer', request.owner.getIdentifier());
21.
22.   // Check if owner exists
23.   const participantRegistry = await getParticipantRegistry(namespace + '.Farmer');
24.   const ownerCheck = await participantRegistry.exists(request.owner.getIdentifier());
25.   if (!ownerCheck) {
26.     throw new Error('This Farmer does not exist!')
27.   } else {
28.     // save the order
29.     const assetRegistry = await getAssetRegistry(creation.getFullyQualifiedType());
30.     await assetRegistry.add(creation);
31.     // emit the event
32.     const createAnimalEvent = factory.newEvent(namespace, 'createAnimalEvent');
33.     createAnimalEvent.geneticId = creation.geneticId;
34.     createAnimalEvent.owner = creation.owner;
35.     createAnimalEvent.location = creation.location;
36.     createAnimalEvent.breed = creation.breed;
37.     createAnimalEvent.weight = creation.weight;
38.     emit(createAnimalEvent);
39.   }
40. }
41.
42. /**
43.  * Update a Animal in the ledger
44.  * @param {org.acme.beef_network.updateAnimal} updateAnimal
45.  * @brief A transaction to update data in the ledger of a given product
46.  * @transaction
47.  */
48. async function updateAnimal(request) {
49.
50.   console.log('updateAnimal');
51.   const factory = getFactory();
52.   const namespace = 'org.acme.beef_network';
53.
54.   // This is a mandatory field
55.   const animalRegistry = await getAssetRegistry(namespace + '.Animal');
56.   const idCheck = await animalRegistry.exists(request.geneticId.getIdentifier());

```

```

57.
58.   if (!lidCheck) {
59.       throw new Error("This Animal does not exist!");
60.       return;
61.   }
62.
63.   let animal = await animalRegistry.get(request.geneticId.getIdentifier());
64.
65.   // This is a mandatory field
66.   const participantRegistry = await getParticipantRegistry(namespace + '.Farmer');
67.   const ownerCheck = await participantRegistry.exists(request.owner.getIdentifier());
68.
69.   if (ownerCheck) {
70.       animal.owner = await participantRegistry.get(request.owner.getIdentifier());
71.   } else {
72.       throw new Error("This Farmer does not exist!");
73.       return;
74.   }
75.
76.   if (request.transportedBy) {
77.       transportRegistry = await getParticipantRegistry(namespace + '.Transportation');
78.       ;
79.       transportCheck = await transportRegistry.exists(request.transportedBy.getIdentifier());
80.       // Throw error and leave if transportation is to be updated but company doesn't exist
81.       if (!transportCheck) {
82.           throw new Error("This Transportation company does not exist!");
83.           return;
84.       } else {
85.           animal.transportedBy = await transportRegistry.get(request.transportedBy.getIdentifier());
86.       }
87.   }
88.
89.   // Get Slaughterhouse registry and perform proper checks
90.   if (request.slaughterhouse) {
91.       slaughterhouseRegistry = await getParticipantRegistry(namespace + '.Slaughterhouse');
92.       slaughterhouseCheck = await slaughterhouseRegistry.exists(request.slaughterhouse.getIdentifier());
93.       // Throw error and leave if transportation is to be updated but company doesn't exist
94.       if (!slaughterhouseCheck) {
95.           throw new Error("This Slaughterhouse company does not exist!");
96.           return;
97.       } else {
98.           animal.slaughterhouse = await slaughterhouseRegistry.get(request.slaughterhouse.getIdentifier());
99.       }
100.   }
101.
102.   // Always mandatory
103.   animal.lifeStage = request.lifeStage;
104.
105.   // Mandatory at creation only
106.   if (request.weight)
107.       animal.weight = animal.weight + ' | ' + request.weight;
108.
109.   if (request.location)
110.       animal.location = animal.location + ' | ' + request.location;
111.
112.   // Always optional
113.   if (request.animalId && animal.animalId)
114.       animal.animalId = animal.animalId + ' | ' + request.animalId;
115.   else if (request.animalId)

```

```

114.     animal.animalId = request.animalId;
115.
116.     if (request.vaccines && animal.vaccines)
117.         animal.vaccines = animal.vaccines + ' | ' + request.vaccines;
118.     else if (request.vaccines)
119.         animal.vaccines = request.vaccines;
120.
121.     if (request.diseases && animal.diseases)
122.         animal.diseases = animal.diseases + ' | ' + request.diseases;
123.     else if (request.diseases)
124.         animal.diseases = request.diseases;
125.
126.     if (request.slaughterDate)
127.         animal.slaughterDate = request.slaughterDate;
128.
129.     await animalRegistry.update(animal);
130. }
131.
132. /**
133.  * Create a new product as an asset
134.  * This new product inherits information from its originary animal
135.  * @param {org.acme.beef_network.createProduct} createProduct - the createProduct tr
    ansaction
136.  * @transaction
137.  */
138.
139. async function createProduct(request) {
140.
141.     console.log('createAnimal');
142.     const factory = getFactory();
143.     const namespace = 'org.acme.beef_network';
144.
145.     const creation = factory.newResource(namespace, 'Product', request.productId);
146.     // Mandatory fields inherited from relationships
147.     creation.geneticId = factory.newRelationship(namespace, 'Animal', request.geneti
    cId.getIdentifier());
148.     creation.slaughterhouse = factory.newRelationship(namespace, 'Slaughterhouse', r
    equest.slaughterhouse.getIdentifier());
149.
150.     // Check if animal exists based on their ID
151.     // (Products can only be created if their source animal exists in the ledger)
152.     const animalRegistry = await getAssetRegistry(namespace + '.Animal');
153.     const animalCheck = await animalRegistry.exists(request.geneticId.getIdentifier(
    ));
154.
155.     if (!animalCheck) {
156.         throw new Error('This Animal does not exist!');
157.         return;
158.     }
159.
160.     // Create only if set - it's optional
161.     if (request.processingIndustry) {
162.         const procIndRegistry = await getParticipantRegistry(namespace + '.Processin
    gIndustry');
163.         const procIndCheck = await procIndRegistry.exists(request.processingIndustry
    .getIdentifier());
164.         if (procIndCheck) {
165.             creation.processingCompany = await procIndRegistry.get(request.processin
    gIndustry.getIdentifier());
166.         } else {
167.             throw new Error('This Processing Industry does not exist!');
168.             return;
169.         }
170.     }
171.
172.     // Create only if set - it's optional

```

```

173.     if (request.retailSeller) {
174.         const retailRegistry = await getParticipantRegistry(namespace + '.Retail');
175.         const retailCheck = await retailRegistry.exists(request.retailSeller.getIden-
tifier());
176.         if (retailCheck) {
177.             creation.retailSeller = await retailRegistry.get(request.retailSeller.ge-
tIdentifier());
178.         } else {
179.             throw new Error('This Retail company does not exist!');
180.             return;
181.         }
182.     }
183.
184.     // Mandatory fields
185.     creation.productId = request.productId;
186.     creation.productType = request.productType;
187.     creation.productStatus = request.productStatus;
188.     creation.weight = request.weight;
189.     creation.location = request.location;
190.
191.     // Optional fields
192.     if (request.subType)
193.         creation.subType = request.subType
194.
195.     if (request.price)
196.         creation.price = request.price
197.
198.     if (request.productIssues)
199.         creation.productIssues = request.productIssues
200.
201.     // Write transaction to the ledger
202.     const assetRegistry = await getAssetRegistry(creation.getFullyQualifiedType());
203.     await assetRegistry.add(creation);
204. }
205.
206. /**
207.  * Update a product in the ledger
208.  * @param {org.acme.beef_network.updateProduct} updateProduct
209.  * @brief A transaction to update data in the ledger of a given product
210.  * @transaction
211.  */
212. async function updateProduct(request) {
213.
214.     console.log('updateProduct');
215.     const factory = getFactory();
216.     const namespace = 'org.acme.beef_network';
217.
218.     const productRegistry = await getAssetRegistry(namespace + '.Product');
219.     const idCheck = await productRegistry.exists(request.product.getIdentifier());
220.
221.     if (!idCheck) {
222.         throw new Error('This product does not exist in the ledger!');
223.         return;
224.     }
225.
226.     let product = await productRegistry.get(request.product.getIdentifier());
227.
228.     if (request.transportedBy) {
229.         const transportRegistry = await getParticipantRegistry(namespace + '.Transpo-
rtation');
230.         const transportCheck = await transportRegistry.exists(request.transportedBy.
getIdentifier());
231.         if (transportCheck) {

```



```

232.         product.transportedBy = await transportRegistry.get(request.transportedB
y.getIdentifier());
233.     } else {
234.         throw new Error('This Transportation company does not exist!');
235.         return;
236.     }
237. }
238.
239.     if (request.retailSeller) {
240.         const retailRegistry = await getParticipantRegistry(namespace + '.Retail');
241.         const retailCheck = await retailRegistry.exists(request.retailSeller.getIden
tifier());
242.         if (retailCheck) {
243.             product.retailSeller = await retailRegistry.get(request.retailSeller.get
Identifier());
244.         } else {
245.             throw new Error('This Retail company does not exist!');
246.             return;
247.         }
248.     }
249.
250.     if (request.processingIndustry) {
251.         const procIndRegistry = await getParticipantRegistry(namespace + '.Processin
gIndustry');
252.         const procIndCheck = await procIndRegistry.exists(request.processingIndustry
.getIdentifier());
253.         if (procIndCheck) {
254.             product.processingCompany = await procIndRegistry.get(request.processing
Industry.getIdentifier());
255.         } else {
256.             throw new Error('This Processing Industry does not exist!');
257.             return;
258.         }
259.     }
260.
261.     if (request.productStatus)
262.         product.productStatus = request.productStatus;
263.
264.     if (request.productType)
265.         product.productType = request.productType;
266.
267.     if (request.subType)
268.         product.subType = request.subType;
269.
270.     if (request.weight && product.weight)
271.         product.weight = product.weight + ' | ' + request.weight;
272.     else if (request.weight)
273.         product.weight = request.weight;
274.
275.     if (request.location && product.location)
276.         product.location = product.location + ' | ' + request.location;
277.     else if (request.location)
278.         product.location = request.location;
279.
280.     if (request.price && product.price)
281.         product.price = product.price + ' | ' + request.price;
282.     else if (request.price)
283.         product.price = request.price;
284.
285.     if (request.productIssues && product.productIssues)
286.         product.productIssues = product.productIssues + ' | ' + request.productIssue
s;
287.     else if (request.productIssues)
288.         product.productIssues = request.productIssues;
289.

```

```
290.     if (request.saleDate)
291.         product.saleDate = request.saleDate;
292.
293.     await productRegistry.update(product);
294. }
```

Apêndice B – Código-fonte do modelo dos participantes e assets da rede *Blockchain no Hyperledger Fabric*

```

1. namespace org.acme.beef_network
2.
3. // BASE DEFINITIONS
4.
5. participant Farmer identified by farmerId {
6.     o String farmerId
7.     o String contact optional
8. }
9.
10. participant Slaughterhouse identified by companyId {
11.     o String companyId
12.     o String contact optional
13. }
14.
15. participant Transportation identified by companyId {
16.     o String companyId
17.     o String contact optional
18. }
19.
20. participant ProcessingIndustry identified by companyId {
21.     o String companyId
22.     o String contact optional
23. }
24.
25. participant Retail identified by companyId {
26.     o String companyId
27.     o String contact optional
28. }
29.
30. participant Consumer identified by consumerId {
31.     o String consumerId
32.     o String contact optional
33. }
34.
35. enum AnimalStatus {
36.     o BREEDING
37.     o REARING
38.     o FATTENING
39.     o IN_TRANSIT
40.     o SLAUGHTERING
41.     o CUTTING
42.     o SLAUGHTERED
43.     o DECEASED
44. }
45.
46. enum ProductStatus {
47.     o PROCESSING
48.     o PACKAGING
49.     o IN_TRANSIT
50.     o ON_SALE
51.     o SOLD
52.     o EXPIRED
53.     o DISCARDED
54. }
55.
56. enum ProductType {
57.     o BEEF
58.     o PROCESSED_MEAT
59. }
60.
61. asset Product identified by productId {
62.     o String productId

```

```

63. --> Animal geneticId
64. --> Slaughterhouse slaughterhouse
65. --> Retail retailSeller optional
66. --> Transportation transportedBy optional
67. --> ProcessingIndustry processingCompany optional
68. o ProductStatus productStatus
69. o ProductType productType
70. o String subType optional
71. o String weight
72. o String location
73. o String price optional
74. o String productIssues optional
75. o DateTime saleDate optional
76. }
77.
78. asset Animal identified by geneticId {
79.   o String geneticId
80.   --> Farmer owner
81.   --> Transportation transportedBy optional
82.   --> Slaughterhouse slaughterhouse optional
83.   o AnimalStatus lifeStage
84.   o String breed
85.   o String location
86.   o String weight
87.   o String animalId optional
88.   o String vaccines optional
89.   o String diseases optional
90.   o DateTime slaughterDate optional
91. }
92.
93. transaction updateAnimal {
94.   --> Animal geneticId
95.   --> Farmer owner
96.   --> Transportation transportedBy optional
97.   --> Slaughterhouse slaughterhouse optional
98.   o AnimalStatus lifeStage
99.   o String weight optional
100.     o String location optional
101.     o String animalId optional
102.     o String vaccines optional
103.     o String diseases optional
104.     o DateTime slaughterDate optional
105.   }
106.
107.   transaction createAnimal {
108.     o String geneticId
109.     --> Farmer owner
110.     o String breed
111.     o String location
112.     o String weight
113.     o String animalId optional
114.     o String vaccines optional
115.     o String diseases optional
116.     o DateTime slaughterDate optional
117.   }
118.
119.   transaction createProduct {
120.     o String productId
121.     --> Animal geneticId
122.     --> Slaughterhouse slaughterhouse
123.     --> Retail retailSeller optional
124.     --> ProcessingIndustry processingCompany optional
125.     o ProductStatus productStatus
126.     o ProductType productType
127.     o String subType optional
128.     o String weight

```

```
129.         o String location
130.         o String price optional
131.         o String productIssues optional
132.     }
133.
134.     transaction updateProduct {
135.         --> Product product
136.         --> Retail retailSeller optional
137.         --> Transportation transportedBy optional
138.         --> ProcessingIndustry processingIndustry optional
139.         o ProductStatus productStatus optional
140.         o ProductType productType optional
141.         o String subType optional
142.         o String weight optional
143.         o String location optional
144.         o String price optional
145.         o String productIssues optional
146.         o DateTime saleDate optional
147.     }
148.
149.     event createAnimalEvent {
150.         o String geneticId
151.         --> Farmer owner
152.         o String breed
153.         o String location
154.         o String weight
155.     }
156.
157.     event createProductEvent {
158.         o String productId
159.         --> Animal geneticId
160.         o ProductStatus productStatus
161.         o ProductType productType
162.         o String weight
163.         o String location
164.     }
```

Apêndice C – Código-fonte do *script* em *Python* para inserção automática de animais na *Blockchain*

```

1. #!/usr/bin/python3.6
2.
3. # Usage: createAnimals.py [NumberOfAnimalsToCreate] [StartGeneticId]
4. # Example: createAnimals.py 10 1
5.
6. import sys
7. import os
8.
9. def runSubmitTransactionCommand(command):
10.     stream = os.popen(command)
11.     output = stream.read()
12.     print(output)
13.
14. if (len(sys.argv) != 3):
15.     print('You must enter two arguments')
16.     print('Usage: createAnimals.py [NumberOfAnimalsToCreate] [StartGeneticId]')
17.     print('Example: createAnimals.py 10 1')
18.     exit(-1)
19.
20.
21. NumberOfAnimalsToCreate = int(sys.argv[1])
22.
23. submit_command = 'composer transaction submit --card admin@beef-tracer --data '
24. function = '{"class":"org.acme.beef_network.createAnimal', '
25. geneticId_field = '"geneticId":'
26. geneticId_value = sys.argv[2]
27. geneticId = geneticId_field + '"' + geneticId_value + '", '
28. owner = '"owner":"resource:org.acme.beef_network.Farmer#Fazendeiro_1"', '
29. breed = '"breed":"Nelore"', '
30. location = '"location":"Santa Helena de Goias/GO"', '
31. animalId = '"animalId":"00155987BED47FEC | 121551BDED47FEC"', '
32. vaccines = '"vaccines":"Febre Aftosa | Botulismo | Clostridioses"', '
33. diseases = '"diseases":"Leptospirose"', '
34. slaughterDate = '"slaughterDate":"2020-08-23T18:40:07.555Z"', '
35. weight = '"weight":"106.51 Kg"}\''
36.
37. for i in range(NumberOfAnimalsToCreate):
38.     geneticId = geneticId_field + '"' + geneticId_value + '", '
39.     command = submit_command + function + geneticId + owner + breed + location + ani
malId + vaccines + diseases + slaughterDate + weight
40.     print('Running transaction {}'.format(i + 1))
41.     runSubmitTransactionCommand(command)
42.     geneticId_Integer = int(geneticId_value)
43.     geneticId_Integer += 1
44.     geneticId_value = str(geneticId_Integer)

```

Apêndice D – Link para o repositório no *GitHub* com o projeto completo da *Blockchain* com o *Hyperledger Fabric* e instruções de instalação

<<https://github.com/adfelippe/blockchain-beef-tracer.git>>