



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS ARARANGUÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM TECNOLOGIA DA INFORMAÇÃO E
COMUNICAÇÃO

Reginaldo José da Rosa

Modelo de logística reversa baseado em contabilidade distribuída - *Blockchain*

Araranguá

2019

Reginaldo José da Rosa

Modelo de logística reversa baseado em contabilidade distribuída - *Blockchain*

Dissertação submetida ao Programa de Pós-Graduação em Tecnologias da Informação e Comunicação da Universidade Federal de Santa Catarina para a obtenção do título de Mestre em Tecnologias da Informação e Comunicação.
Orientador: Prof. Dr. Roderval Marcelino
Coorientador: Prof. Dr. Vilson Gruber

Araranguá

2019

Ficha de identificação da obra

Da Rosa, Reginaldo José

Modelo de logística reversa baseado em contabilidade distribuída - Blockchain / Reginaldo José Da Rosa ; orientador, Roderval Marcelino, coorientador, Vilson Gruber, 2019.

136 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Campus Araranguá, Programa de Pós-Graduação em Tecnologias da Informação e Comunicação, Araranguá, 2019.

Inclui referências.

1. Tecnologias da Informação e Comunicação. 2. Logística Reversa. 3. DLT. 4. Blockchain. 5. Contrato Inteligente. I. Marcelino, Roderval. II. Gruber, Vilson. III. Universidade Federal de Santa Catarina. Programa de Pós Graduação em Tecnologias da Informação e Comunicação. IV. Título.

Reginaldo José da Rosa

Modelo de logística reversa baseado em contabilidade distribuída - *Blockchain*

O presente trabalho em nível de mestrado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Prof. Martín Vigil, Dr.
Universidade Federal de Santa Catarina

Prof.(a) Eliane Pozzebon, Dr(a).
Universidade Federal de Santa Catarina

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de mestre em “Tecnologia da Informação e Comunicação”.

Prof(a). Dra. Andréa Cristina Trierweiller
Coordenador(a) do Programa

Prof. Dr. Roderval Marcelino
Orientador(a)

Araranguá, 31 de outubro de 2019.

AGRADECIMENTOS

Agradeço a Deus pela saúde e por ter colocado no meu caminho pessoas incríveis que foram essenciais para a conclusão desta pesquisa a quem extendo meus agradecimentos.

A minha esposa Michele Rodrigues da Rosa e meus filhos Marina Rodrigues da Rosa e Vinícius Rodrigues da Rosa pela paciência e incentivo nos momentos em que eu fraquejava. Amo vocês.

A empresa FARBEN, em nome de meu diretor Edilson Zanatta, que desde o início desta empreitada me incentivou e me dispensou do trabalho para que pudesse participar das disciplinas.

Aos colegas do mestrado que aturaram minhas constantes interrupções nas aulas e que ajudaram a ampliar meus conhecimentos, em especial Jefferson Pacheco dos Santos e Rogerio Hermínio da Silva.

Aos professores que dedicaram seu tempo para nos transmitir conhecimento, nos mantendo atualizados e instigando nossa curiosidade.

A Universidade Federal de Santa Catarina – Campus Araranguá e ao Programa de Pós-Graduação em Tecnologias da Informação e Comunicação por lutar para manter este programa de mestrado bem-conceituado.

Ao meu orientador prof. Dr. Roderval Marcelino que nunca mediu esforços para apoiar e orientar meus caminhos nesta jornada. Este período que trabalhamos juntos na pesquisa fez crescer ainda mais a amizade, admiração e respeito que tenho por este profissional.

Aos que não foram citados, mas que de alguma forma fizeram parte desta dissertação, fica meus sinceros agradecimentos.

“Talvez não tenha conseguido fazer o melhor, mas lutei para que o melhor fosse feito. Não sou o que deveria ser, mas Graças a Deus, não sou o que era antes”. (Marthin Luther King)

RESUMO

A tecnologia de contabilidade distribuída está num período de amadurecimento e tem recebido destaque nos mais variados setores da economia. O *Blockchain* é uma das tecnologias de contabilidade distribuída que se popularizou devido sua utilização na fundamentação do *Bitcoin*. Todavia sua aplicação tem se expandido além das criptomoedas para uso em negócios tradicionais. A logística reversa recebe destaque no meio corporativo, pois as ações para preservação do meio ambiente vêm pressionando empresas e governos. Desta forma esta pesquisa tem o objetivo de propor uma prova de conceito para avaliar se a tecnologia de contabilidade distribuída utilizando o *Blockchain* poderá suportar um novo modelo de sistema de logística reversa. Para tanto foi desenvolvido uma metodologia que contempla os atores previstos na Política Nacional de Resíduos Sólidos e os processos do ciclo de vida do produto das embalagens de aço. Neste contexto foi incluído um novo item com objetivo de incentivar o funcionamento do sistema, o ativo criptoLata. Para minimizar os trabalhos e ser produtivo na construção desta prova de conceito, somente os processos de Venda Indústria, Venda ao Consumidor e Devolução foram implementados. Para a definição e programação desta rede *Blockchain* foi utilizada a plataforma *Hyperledger Composer Playground*, com o propósito de reduzir o tempo de desenvolvimento e validação da rede de negócio. Para atender ao fluxo proposto, o contrato inteligente foi desenvolvido utilizando a linguagem *JavaScript*. Para sua validação o método de *Teste Funcional* foi aplicado, onde todas as configurações possíveis de transações foram simuladas. Após a realização dos testes, obteve-se um índice de assertividade de 100%. Com este resultado pode-se afirmar que a tecnologia *Blockchain* pode suportar um novo modelo de sistema de logística reversa.

Palavras-chave: Logística Reversa. DLTs. Blockchain. Contrato Inteligente.

ABSTRACT

Distributed ledger technology is in a period of maturity and has been highlighted in various sectors of the economy. Blockchain is one of the distributed ledger technologies that became popular due to its use in the Bitcoin foundation. However, its application has expanded beyond cryptocurrencies for use in traditional businesses. Reverse logistics is highlighted in the corporate environment, as actions to preserve the environment are pressuring companies and governments. Thus, this research aims to propose a proof of concept to evaluate if the distributed ledger technology using Blockchain can support a new reverse logistics system model. To this end, a methodology was developed that contemplates the actors foreseen in the National Solid Waste Policy and the life cycle processes of the steel packaging product. In this context, a new item was included to encourage the system to function, the asset cryptoLata. To minimize work and be productive in building this proof of concept, only the Industrial Sell, Consumer Sell and Return processes were implemented. For the definition and programming of this Blockchain network, the Hyperledger Composer Playground platform was used to reduce the time of development and validation of the business network. To meet the proposed flow, the smart contract was developed using the JavaScript language. For its validation the Functional Test method was applied, where all possible transaction configurations were simulated. After the tests, a 100% assertiveness index was obtained. With this result it can be stated that Blockchain technology can support a new reverse logistics system model.

Keywords: Reverse logistic. DLTs. Blockchain. Smart contract.

LISTA DE FIGURAS

Figura 1 – Etapas do desenvolvimento da dissertação.....	24
Figura 2 – Estrutura de uma <i>Blockchain</i>	27
Figura 3 – Exemplo de entrada e saída <i>hash</i>	29
Figura 4 – Pequena mudança na entrada / grande mudança na saída <i>hash</i>	29
Figura 5 – Esquema do funcionamento da SHA-256	30
Figura 6 – Variável e ponteiro	30
Figura 7 – Lista vinculada com ponteiros de <i>hash</i>	31
Figura 8 – Estrutura com dois <i>hashes</i> dos dois blocos de dados.....	32
Figura 9 – Representação de uma árvore <i>Merkel</i>	33
Figura 10 – Tipos de algoritmos de consenso.....	35
Figura 11 – Estrutura de uma PKI.....	38
Figura 12 – Projetos <i>Hyperledger</i>	40
Figura 13 – <i>Ledger</i> no <i>Hyperledger</i>	43
Figura 14 – Pares compartilhando o mesmo canal	43
Figura 15 – Rede composta por 2 organizações.....	45
Figura 16 – Representação de uma rede <i>Hyperledger</i>	45
Figura 17 – Estrutura do <i>Hyperledger Composer</i>	47
Figura 18 – Tela inicial do <i>Hyperledger Composer Playground</i>	50
Figura 19 – Tela de seleção de rede ou criação de nova rede.....	50
Figura 20 – Tela de criação de rede.....	51
Figura 21 – Tela de definição da rede.....	52
Figura 22 – Tela de Teste	52
Figura 23 – Exemplos da linguagem de modelagem	54
Figura 24 – Código da construção de uma transação.....	55
Figura 25 – Código da construção de um evento.....	56
Figura 26 – Ciclo da LR.....	58
Figura 27 – Fluxo entre os atores do SLR.....	59
Figura 28 – Aspectos políticos e legais da implantação do SLR.....	60
Figura 29 – Fluxo Logístico Proposto pela Coalizão	64
Figura 30 – Centro PROLATA de reciclagem.....	65
Figura 31 – Diagrama de atividades da metodologia proposta para implantar o SLR	85

Figura 32 – Recorte do diagrama de atividades.....	88
Figura 33 – Parte do Arquivo <i>Model Files</i> , definição dos <i>participants</i>	90
Figura 34 – Parte do Arquivo <i>Model Files</i> , definição dos <i>assets</i>	92
Figura 35 – Parte do Arquivo <i>Model Files</i> , definição das <i>transactions</i>	94
Figura 36 – Tela consulta participantes do Teste Funcional da rede	99
Figura 37 –Teste Funcional – Primeira linha do teste	102
Figura 38 –Teste Funcional – Segunda linha do teste	103
Figura 39 –Teste Funcional – Consulta <i>Asset AtivoVenda</i> primeiro registros da <i>Blockchain</i>	103
Figura 40 –Teste Funcional – Consulta <i>Asset CarteiraCriptoLata</i> primeiro registros	104
Figura 41 –Teste Funcional – Consulta lista das transações da <i>Blockchain</i>	104
Figura 42 –Teste Funcional – Consulta <i>Ledger</i> da <i>Blockchain</i>	105
Figura 43 – Gráfico Transações de sucesso no Teste Funcional – saldo Gestor ...	105
Figura 44 – Gráfico Transações de sucesso no Teste Funcional – saldo Participantes	106
Figura 45 – Gráfico comparativo dos resultados.....	106
Figura 46 – Gráfico Percentual comparação resultado esperado x obtidos	107
Figura 47 – Gráfico demonstração dos resultados da PoC.....	108
Figura 48 – Gráfico Percentual de execuções concluídas	108

LISTA DE QUADROS

Quadro 1 – Comparativo dos Algoritmos de Consenso	36
Quadro 2 – Situação da implantação dos sistemas de LR.....	62
Quadro 3 – Diretrizes para Pesquisa em DSR.....	80
Quadro 4 – Componentes do diagrama de atividades	82
Quadro 5 – Participantes.....	96
Quadro 6 – Lista das transações.....	97

LISTA DE TABELAS

Tabela 1 – Prioridades dos indicadores de LR.....	61
Tabela 2 – Número de citações dos artigos com “blockchain” nas bases WoS, Scopus e IEEE	67
Tabela 3 – Número de citações dos artigos com “logistic AND reverse” nas bases WoS e Scopus	67
Tabela 4 – Regras para os testes da PoC	97
Tabela 5 – Resultados esperados para o teste da PoC	99
Tabela 5 – Resultados esperados para o teste da PoC	100
Tabela 6 – Resultados obtidos com a execução das transações.....	100
Tabela 7 – Resumo das transações que obtiveram sucesso de execução	109

LISTA DE ABREVIATURAS E SIGLAS

ABAL – Associação Brasileira do Alumínio
ACV - Avaliação do ciclo de vida do produto
BFT - *Byzantine Fault Tolerance*
DAPPS - *Decentralized Applications*
DHT – *Distributed HashTable*
DLT - *Distributed Ledger Technology*
DSR - *Design Science Research*
EC2 - *Amazon Elastic Compute Cloud*
EDI - *Enterprise Data Integration*
e-PING - Padrões de Interoperabilidade de Governo Eletrônico.
ERP - *Enterprise Resource Planning*
GrSCM - *Green supply-chain management*
IEEE - Instituto de Engenheiros Eletricistas e Eletrônicos
IoT - Internet das Coisas
IPFS - *InterPlanetary File System*
JSON - *JavaScript Object Notation*
LR - Logística Reversa
MMA - Ministério do Meio Ambiente
MSP - *Membership Service Provider*
PEV - Pontos de Entrega Voluntária
PKI - *Public Key Infrastructure*
PNRS - Política Nacional de Resíduos Sólidos
PoC - *Proof of Concept*
PoW - *Proof of Work*
PPGTIC - Pós-Graduação em Tecnologias da Informação e Comunicação
P2P - *Peer-to-peer*
RSU - Resíduos Sólidos Urbanos
SCM – *Supply-Chain Management*
SHA-256 - Secure Hashing Algorithm 256
SINIR - Sistema Nacional de Informações sobre a Gestão dos Resíduos Sólidos
SLR - Sistemas de Logística Reversa
TIC - Tecnologias de Informação e Comunicação

UML - *Unified Modeling Language*

WoS – *Web of Science*

XML - *Extensible Markup Language*

SUMÁRIO

1	INTRODUÇÃO	17
1.1	APRESENTAÇÃO DO PROBLEMA	17
1.2	OBJETIVOS	19
1.2.1	Objetivo Geral	19
1.2.2	Objetivos Específicos	19
1.3	JUSTIFICATIVA.....	21
1.4	ESCOPO DO TRABALHO.....	22
1.5	ADERÊNCIA AO PROGRAMA DE PÓS-GRADUAÇÃO EM TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO	23
1.6	METODOLOGIA	23
1.7	ESTRUTURA DO TRABALHO	24
2	<i>BLOCKCHAIN</i>	26
2.1	ENTENDENDO A TECNOLOGIA <i>BLOCKCHAIN</i>	26
2.1.1	Funcionamento da função <i>Hash Criptográfico</i>	28
2.1.2	Ponteiros	30
2.1.3	Listas vinculadas	31
2.1.4	Árvore <i>Merkel</i>	31
2.1.5	Algoritmos de consenso	33
2.1.6	Chaves públicas e privadas	38
2.1.7	<i>Blockchain</i> Pública e Privada	38
2.2	<i>HYPERLEDGER</i>	39
2.2.1	Contratos Inteligentes	41
2.2.2	<i>Hyperledger Fabric</i>	41
2.3	<i>HYPERLEDGER COMPOSER</i>	46
2.3.1	Detalhes da estrutura <i>Hyperledger Composer</i>	47
2.3.1.1	<i>Model File</i>	47
2.3.1.2	<i>Script File</i>	48
2.3.1.3	<i>Access Control</i>	48
2.3.1.4	<i>Query File</i>	48
2.3.1.5	<i>Business Network Archive</i>	49
2.3.2	<i>Hyperledger Composer Playground</i>	49

2.3.3	Linguagem de Modelagem do <i>Hyperledger Composer</i>	53
2.3.4	Linguagem de <i>Script</i> do <i>Hyperledger Composer</i>	54
3	LOGISTICA REVERSA	57
3.1	PONTOS DE ENTREGA VOLUNTÁRIA – PEV	61
3.2	TERMOS E ACORDOS SETORIAIS.....	61
3.3	ACORDO SETORIAL PARA IMPLANTAÇÃO DO SISTEMA DE LOGÍSTICA REVERSA DE EMBALAGENS EM GERAL	63
3.4	TERMO DE COMPROMISSO PARA IMPLANTAÇÃO DO SISTEMA DE LOGÍSTICA REVERSA DE EMBALAGENS DE AÇO	64
4	ESTADO DA ARTE	66
5	PROCEDIMENTOS METODOLÓGICOS	79
5.1	Desenvolvimento da PoC	87
5.2	Contratos Inteligentes.....	90
6	RESULTADOS E DISCUSSÕES	96
7	CONSIDERAÇÕES FINAIS	110
	REFERÊNCIAS	112
	APÊNDICE B – <i>Script</i> Carga Inicial	119
	APÊNDICE C – <i>Script</i> Limpa Movimento	121
	APÊNDICE D – <i>Script</i> VendaIndustria	122
	APÊNDICE E – <i>Script</i> VendaConsumidor	128
	APÊNDICE F – <i>Script</i> Devolução	132
	APÊNDICE G – <i>Script</i> ConsultaVendas	135
	APÊNDICE H – Arquivo de <i>Query</i>	136

1 INTRODUÇÃO

1.1 APRESENTAÇÃO DO PROBLEMA

O *Blockchain* é uma *Distributed Ledger Technology* (DLT) e ficou conhecido mundialmente por ser a base para a implementação da criptomoeda *Bitcoin*, introduzido em 2008 por Satoshi Nakamoto (um pseudônimo utilizado pelo autor ou grupo de autores). Em 2017 o *Bitcoin* agitou o mercado financeiro quando sua cotação atingiu o valor em dólar de \$17,549.67, ou seja, uma alta expressiva de 1700% em um ano, principalmente quando analisado que em 2010 o preço mais alto de um *Bitcoin* atingiu foi \$0,39 (“Gráfico do Histórico de Preço do Bitcoin”, 2018) (FLOREA, 2018).

Podemos encontrar definições onde o *Blockchain* é confundido com DLT, isto se dá porque o *Blockchain* foi a primeira implementação funcional de uma DLT (BRANCO, 2018).

A tecnologia do *Blockchain* permite que as transações realizadas possuam características como imutabilidade e descentralização, além da dispensa de um órgão regulador devido ao algoritmo de validação por meio de votação e chegando ao consenso pelos nós participantes (SCHUEFFEL, 2017).

O *Blockchain* na prática é um conjunto de dados distribuídos em vários “nós”, usando redes *peer-to-peer* (P2P), onde qualquer alteração no conjunto de dados será refletida em todos os “nós” da rede. O *Blockchain* é uma lista encadeada que contém dados e um ponteiro de *hash* que aponta para seu bloco anterior, criando assim a cadeia. Um ponteiro de *hash* é semelhante a um ponteiro, mas em vez de apenas conter o endereço do bloco anterior, ele também contém o *hash* dos dados dentro do bloco anterior. Essa utilização dos ponteiros de *hash* que torna o *Blockchain* tão incrivelmente confiável e pioneiro, inviabilizando uma adulteração de conteúdo de um bloco, pois precisaria alterar todos os blocos anteriores para manter a rede íntegra (BLOCKGEEKS, 2018).

O protocolo *Blockchain* apresenta gargalos na escalabilidade da rede e algumas vulnerabilidades, além do consumo elevado de energia para geração de novos blocos. Isso motivou o desenvolvimento e evolução de novas DLTs,

consideradas variações do *Blockchain*, como a *Ethereum* e *Hyperledger*, e também novas tecnologias DLTs alternativas ao *Blockchain*, como o *Tangle* e o *Hashgraph* (SCHUEFFEL, 2017).

Devido as características citadas do *Blockchain*, além da utilização no desenvolvimento das criptomoedas, as empresas tem investido em pesquisa para utilizar o *Blockchain* em sua camada de negócios, com o intuito de otimizar os recursos, maior segurança e eliminar intermediários nas negociações (SCHUEFFEL, 2017) . O Contrato Inteligente é um contrato digital que define as regras para uma transação e não pode ser perdido ou adulterado. Surgiram com a rede *Ethereum*, são algoritmos auto executável, e com objetivo de solucionar diversos problemas empresariais, dentre os setores que tem realizado pesquisa e possui alguma aplicação desenvolvida ou estão planejando algo são: financeiro, saúde e logística.

A logística reversa passa a receber mais atenção no Brasil a partir da promulgação da lei 12.305 de 2 de agosto de 2010, que institui a Política Nacional de Resíduos Sólidos (PNRS) onde apresenta as definições e responsabilidades aos agentes envolvidos no ciclo de vida do produto (BRASIL, 2010a)(BRASIL, 2010b). A lei conceitua a logística reversa como:

“instrumento de desenvolvimento econômico e social caracterizado por um conjunto de ações, procedimentos e meios destinados a viabilizar a coleta e a restituição dos resíduos sólidos ao setor empresarial, para reaproveitamento, em seu ciclo ou em outros ciclos produtivos, ou outra destinação final ambientalmente adequada”.

E resíduos sólidos como:

“material, substância, objeto ou bem descartado resultante de atividades humanas em sociedade, a cuja destinação final se procede, se propõe proceder ou se está obrigado a proceder, nos estados sólido ou semissólido, bem como gases contidos em recipientes e líquidos cujas particularidades tornem inviável o seu lançamento na rede pública de esgotos ou em corpos d’água, ou exijam para isso soluções técnica ou economicamente inviáveis em face da melhor tecnologia disponível”.

A lei define também que a responsabilidade é compartilhada entre o poder público, fabricantes, importadores, distribuidores, comerciantes e os consumidores, a estruturar e implementar sistemas de logística reversa (SLR), mediante retorno dos produtos (embalagens) após o uso pelo consumidor, de forma independente do serviço público de limpeza urbana e de manejo dos resíduos sólidos, e lista os produtos: agrotóxicos, pilhas e baterias, pneus, óleos lubrificantes, lâmpadas

fluorescentes (de vapor de sódio e mercúrio e de luz mista), produtos eletroeletrônicos e seus componentes. O objetivo é de reutilização, reciclagem e tratamento dos resíduos sólidos, bem como disposição final ambientalmente adequada dos rejeitos.

O desafio para atender a lei 12.305 em um país de dimensões continentais com é o Brasil, talvez possa ser amenizado com a aplicação da tecnologia *Blockchain* possa o desenvolvimento de um SLR de forma a viabilizar tecnicamente e economicamente sua implantação.

Este trabalho se utilizará da plataforma *Hyperledger Composer Playground* para gerar a *Proof of Concept* (PoC) para modelar as ideias propostas na metodologia antes de sua implementação. A PoC é uma forma de testar uma ideia ou conceito que se deseja implementar e colocar no mercado. Desta forma se consegue com investimento mínimo de recursos e tempo, validar a ideia, identificar possíveis problemas no projeto e assim permitir a obtenção de *feedback* dos *stakeholders* envolvidos ou interessados no projeto (CONSULTING, 2015).

Questão da pesquisa:

A tecnologia de contabilidade distribuída, implementada utilizando *Blockchain*, pode suportar um novo modelo de logística reversa?

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Utilizar a tecnologia de contabilidade distribuída, *Blockchain*, para construção de uma prova de conceito para implantação de um sistema de logística reversa para embalagens de aço.

1.2.2 Objetivos Específicos

Para realizar a pesquisa e atingir o objetivo geral, será especificado com mais detalhes os objetivos:

- Estudar os fundamentos relacionados a tecnologia *Blockchain*;
- Pesquisar sobre a legislação referente a logística reversa;

- Propor um modelo de gestão de logística reversa baseada em contabilidade distribuída-*Blockchain*;
- Modelar e desenvolver um protótipo para realizar a prova de conceito do funcionamento da rede *Blockchain* aplicado à logística reversa;
- Testar e validar o contrato inteligente para o modelo proposto;

1.3 JUSTIFICATIVA

A logística reversa é assunto comum para uma boa parte das empresas. Grande parte dos insumos que as empresas siderúrgicas utilizam em seu processo de produção, são sucata gerada por seus clientes e, para isso, usam centros coletores de carga (PARDO et al., 2008). As indústrias de latas de alumínio também são destaque no grande aproveitamento de matéria-prima reciclada. Segundo a Associação Brasileira do Alumínio (ABAL), o Brasil em 2015 reciclou 97,9% das latinhas de alumínio para bebidas de um total de 292,5 mil toneladas, isso garante ao Brasil manter-se entre os países líderes na reciclagem deste material (ABAL, 2017).

Existem ainda outros setores da indústria nos quais o processo de gerenciamento da logística reversa é mais recente, como na indústria de eletrônicos, varejo e automobilística. Esses setores estão buscando o reaproveitamento de materiais para produção, por meio das devoluções dos clientes e das atividades de reciclagem e reaproveitamento da sucata de produtos e embalagens (PARDO et al., 2008). Estes têm percebido um aumento considerável nos últimos anos no escopo e escala da reciclagem. Dessa forma, empresas que possuem um processo de logística reversa bem gerido, tendem a se destacar em seu segmento de mercado, diferenciando-se de seus concorrentes.

Por consciência ambiental ou para atendimento da legislação, as empresas estão cada vez mais acompanhando o ciclo de vida de seus produtos. A responsabilidade estendida de fabricantes e importadores em relação aos produtos após sua vida útil e suas embalagens está tornando-se cada vez mais comum em todo o mundo, isso tem impulsionado as ações de concretização dos SLR (COUTO; LANGE, 2017).

Couto e Lange (2017) indicam uma cronologia muito recente de logística reversa, onde primeiros relatos datam de 1970, ganhando força a partir da década de 1980, mas apenas a partir dos anos 1990 passou a ser discutida e implantada. Porém somente em 1995 apareceram os primeiros trabalhos em logística reversa por questões socioambientais. No entanto a motivação para a realização da logística reversa por parte das empresas em geral possui três eixos fundamentais: ambiental, financeiro e legal. O Ambiental tem o apelo de marketing criando a imagem “verde”.

O financeiro com foco na reutilização e recuperação de produtos. A legislação tem motivado as empresas rever o ciclo de vida dos seus produtos assim impulsiona e regula a implantação da logística reversa no país.

Como nossa legislação define a gestão compartilhada e envolve muitos agentes entre eles fabricantes, importadores, distribuidores, comerciantes, setor público e consumidor, o grau de dificuldade de implementar um SLR eficaz é grande, principalmente a gestão do mesmo.

Desta forma este trabalho visa sugerir uma metodologia para a governança de SLR por meio da utilização da tecnologia *Blockchain*. A transparência é fundamental para que haja engajamento de todos os agentes, além de formas mais eficientes para a fiscalização dos acordos setoriais.

1.4 ESCOPO DO TRABALHO

Este trabalho se limita a estudar e propor uma metodologia para a implantação de um sistema de logística reversa para latas de aço de tintas em nível nacional. Avaliar as características e a viabilidade da tecnologia *Blockchain* para uma prova de conceito, mais especificamente o *Hyperledger Fabric* que atualmente é mantido pela fundação *The Linux Foundation*¹ juntamente com outros projetos.

Para o desenvolvimento e validação do contrato inteligente será utilizado o *Hyperledger Composer Playground*, ferramenta que possui recursos que permite focar no desenvolvimento e validação da rede. Também permite reaproveitar todo o trabalho realizado pra acelerar uma implantação real em ambiente de produção, sendo necessários somente a configuração da rede e o desenvolvimento das aplicações para consumirem a rede criada, os *Decentralized Applications* (DAPPS).

Para criar um ambiente propício a adesão dos participantes neste SLR, um elemento de incentivo será integrado na metodologia, aproveitando das características de descentralização e imutabilidade da tecnologia *Blockchain*.

¹ <https://www.hyperledger.org/projects/composer>

O sistema proposto não se aplica a outros ciclos de produtos ou resíduos sólidos uma vez que os participantes podem ser diferentes, para tanto necessitará de reformulação e adequação na metodologia.

1.5 ADERÊNCIA AO PROGRAMA DE PÓS-GRADUAÇÃO EM TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO

O programa de Pós-Graduação em Tecnologias da Informação e Comunicação (PPGTIC), busca resolver problemas de características interdisciplinares por intermédio das tecnologias computacionais (PPGTIC, 2018). O programa concentra as pesquisas nas áreas: Tecnologia Computacional, Tecnologia Gestão e Inovação e Tecnologia Educacional.

Esta pesquisa se integra ao programa na linha da Tecnologia Computacional. Esta linha tem como finalidade o desenvolvimento de novas soluções computacionais para a resolução de problemas de caráter interdisciplinar (PPGTIC, 2018).

Durante todo o trabalho de pesquisa podem ser observados indícios da proposta e sua aderência ao PPGTIC, sendo que uma das características desta pesquisa é ser interdisciplinar. No trabalho é explanado o desenvolvimento de uma metodologia para implantação de uma rede *Blockchain* com propósito de gerir e fiscalizar ações da logística reversa. Assim, a pesquisa faz uso de tecnologias computacionais inovadoras, o que vai ao encontro a proposta de interdisciplinaridade da linha de pesquisa e do programa com um todo.

1.6 METODOLOGIA

Esta seção da dissertação tem por objetivo descrever a metodologia empregada e classificar a pesquisa adotada.

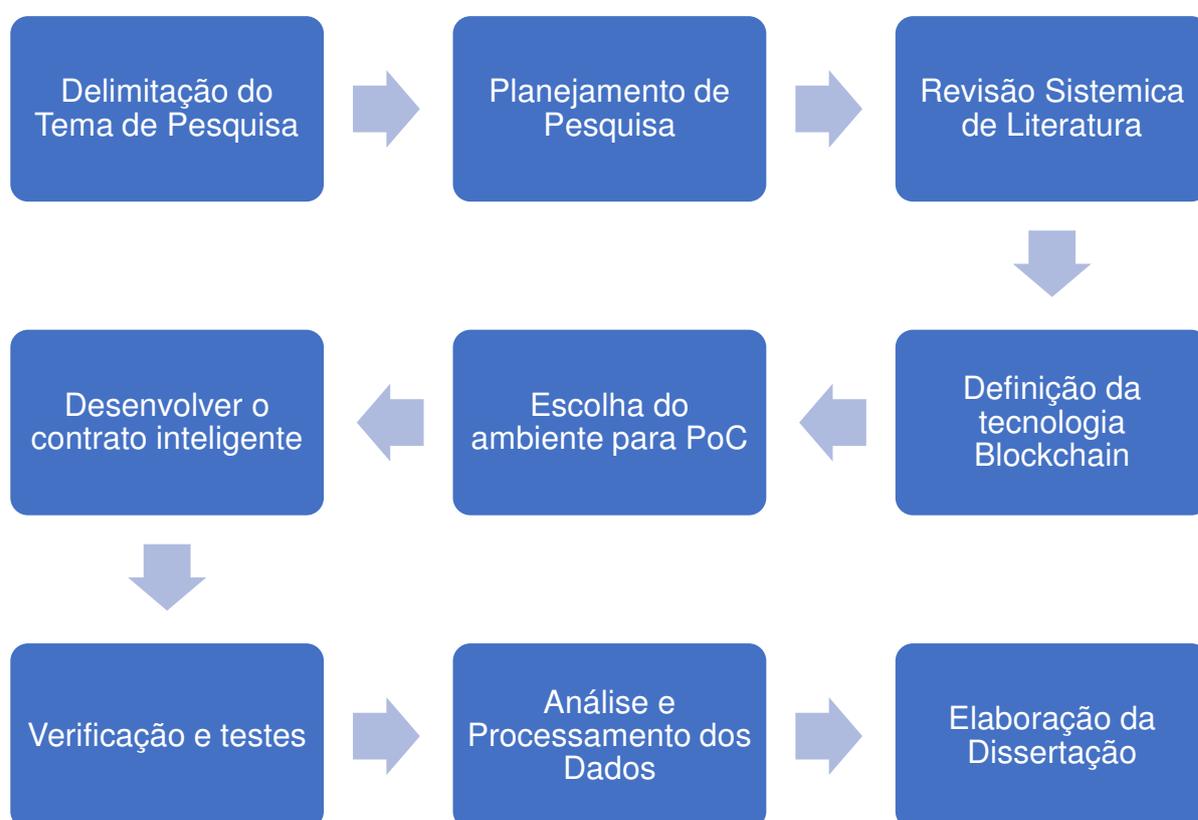
Este trabalho tem caráter interdisciplinar, pois tramita entre as áreas de ciência da computação, gestão, jurídica e logística.

Quanto aos procedimentos técnicos, sua fundamentação teórica está baseada na pesquisa bibliográfica por meio da compilação de registros em livros, artigos de bases de dados conceituadas, publicações de órgãos oficiais, etc. Devido ao tema ser

recente, a busca exploratória em sites diversos e oficiais serão realizadas para coleta de publicações atuais. O método *SystematicSearchFlow* (SSF) será aplicado para mapeamento do material bibliográfico (FERENHOF; FERNANDES, 2016).

Para o desenvolvimento da prova de conceito será utilizado o projeto *Hyperledger Fabric* em conjunto com o *Hyperledger Composer*, tanto para criação como validação do contrato inteligente. A figura 1 apresenta as etapas do desenvolvimento desta dissertação.

Figura 1 – Etapas do desenvolvimento da dissertação.



Fonte: Elaborada pelo autor.

1.7 ESTRUTURA DO TRABALHO

A estrutura a ser apresentada nesta seção é o que se planeja para o desenvolvimento desta dissertação sendo constituída de 7 capítulos.

No capítulo 1 será realizada a introdução do tema, os objetivos da pesquisa, além da justificativa, escopo e aderência ao programa.

O capítulo 2 aborda os conceitos computacionais aplicados para o desenvolvimento da pesquisa e o ambiente que será utilizado para a construção da PoC.

O capítulo 3 trará à tona a importância do assunto relativo a logística reversa, apresentará setores que tem o processo implementado e destacará alguns pontos da legislação brasileira que define a PNRS.

No capítulo 4 apresentará o estado da arte relativo a logística reversa, as tecnologias *Blockchain* e suas aplicações na logística e similares.

O capítulo 5 dedica-se a metodologia e recursos empregados durante o desenvolvimento e testes do contrato inteligente para implementar o fluxo da logística reversa aplicados no ambiente *Hyperledger Composer*.

O capítulo 6 trará os resultados e discussões com a aplicação das tecnologias computacionais na gestão do fluxo da logística reversa.

O capítulo 7 apresentará as considerações finais e os resultados obtidos a fim de prover subsídios para responder à pergunta da pesquisa e propor trabalhos futuros. Finalmente serão apresentadas as referências e apêndices.

2 BLOCKCHAIN

Para Upadyaya, Sharma e Arun (2017) a tecnologia *Blockchain* não está apenas mudando a maneira como usamos a Internet, mas também está revolucionando a economia global. Ao permitir a digitalização ou digitização² de ativos, está conduzindo uma mudança fundamental no significado da internet como a conhecemos.

Existe uma confusão com os termos *Bitcoin* e *Blockchain*, então para esclarecer logo no início. *Bitcoin* apareceu pela primeira vez em um *white paper* de 2008 escrito por Satoshi Nakamoto (um pseudônimo que representa uma pessoa ou grupo de pessoas) que detalhou um inovador sistema de numerário eletrônico que permitia que pagamentos *on-line* fossem realizados sem intermediários, dando origem as criptomoedas. O *Blockchain* é a tecnologia que permitiu o desenvolvimento do *Bitcoin*, esta sim foi a verdadeira revolução. A tecnologia *Blockchain* também eliminaria altas taxas intermediárias que se tornaram um fardo para os indivíduos e as empresas, e abrindo possibilidade para inúmeras outras aplicações (UPADHYAYA; SHARMA; ARUN, 2017).

2.1 ENTENDENDO A TECNOLOGIA BLOCKCHAIN

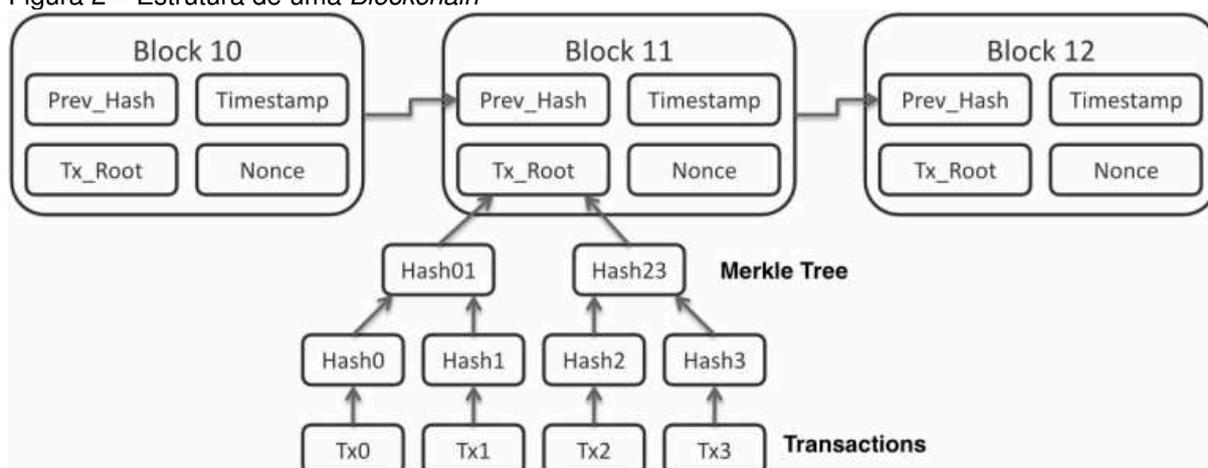
A tecnologia *Blockchain* fornece um livro-razão compartilhado, confiável, seguro e imutável para registrar transações, ativos digitais e "contratos inteligentes". Estes são gravados de forma distribuída em uma infinidade de computadores em todo o mundo em uma rede P2P³, onde cada membro mantém uma cópia do livro-razão. As transações são agrupadas em blocos protegidos por um *hash* (ver item 2.1.1) e validadas por um protocolo de consenso e interligadas com o bloco imediatamente anterior (HYPERLEDGER, 2019).

² Nos últimos anos, um novo termo, aparentemente escrito de forma errada, começou a ser usado pelas organizações: digitalização. Normalmente este termo está envolvido em atividades de revisão de processos e de inovação de negócios. O termo ainda não está disponível na maioria dos dicionários de língua portuguesa. <https://blog.smlbrasil.com.br/digitalizacao-digitizacao-ou-transformacao-digital/>

³ P2P (do inglês *peer-to-peer*, que significa par-a-par ou ponto-a-ponto) é um formato de rede de computadores em que a principal característica é a descentralização, onde o cada computador membro realiza funções de servidor e de cliente simultaneamente. <https://www.tecmundo.com.br/torrent/192-o-que-e-p2p-.htm>

Os mineradores, nome dado aos membros da rede com altos níveis de poder computacional, competem para validar as transações resolvendo problemas complexos codificados, em troca de recompensa financeira. O primeiro mineiro a resolver os problemas e validar o bloco é quem recebe a recompensa, e no caso do *Bitcoin* um minerador receberia uma quantia de *Bitcoins*, e na *Ethereum*⁴ em *Ether*⁵. O bloco validado de transações é então marcado com a data e hora e adicionado a uma cadeia em uma ordem cronológica linear. Novos blocos de transações validados são vinculados a blocos mais antigos, construindo uma cadeia de blocos, denominada então de *Blockchain*, ver figura 2. Todas as cadeias são atualizadas de modo que cada livro contábil da rede seja idêntico, dando a cada membro a capacidade de provar quem tem a propriedade de um ativo em determinado momento.

Figura 2 – Estrutura de uma *Blockchain*



Fonte: <https://conexaofintech.com.br/blockchain/blockchain-todos-serao-impactados/>

Schueffel (2017) e Upadhyaya, Sharma e Arun (2017) detalham as características de imutabilidade, descentralizada, aberta e criptográfica do *Blockchain*, permite que as pessoas confiem umas nas outras e realizem transações entre pares, tornando a necessidade de intermediários obsoleta. Isso também traz benefícios de segurança sem precedentes. Pois ataques que afetam grandes intermediários centralizados como bancos seria virtualmente impossível na rede *Blockchain*, graças

⁴*Ethereum* é uma plataforma descentralizada capaz de executar contratos inteligentes e aplicações descentralizadas usando a tecnologia *Blockchain*.

⁵ *Ether* é a criptomoeda da *Ethereum*.

ao vínculo da cadeia com os blocos anteriores. Caso um invasor tenha intenção de apropriar-se de um bloco em uma rede *Blockchain*, seria necessário invadir não somente o bloco específico, mas todos os blocos precedentes, voltando a história inteira desta *Blockchain* e eles precisariam fazer isso em todos os blocos dos livros de uma rede, que podem ser milhões, simultaneamente.

Os benefícios esperados com o uso da tecnologia *Blockchain* permitem aumento da eficiência e redução nos custos operacionais das transações, porém também pode contribuir para o fim de empresa ou instituição cujas atividades são consideradas intermediárias. Como exemplo, o setor financeiro ironicamente tem visão ambígua no uso da tecnologia. Por um lado, veem benefícios em melhorar uma variedade de processos e gerenciamento de transações e registros com a implantação de uma rede *Blockchain*. Por outro lado, enfrentam a ameaça pela eliminação de intermediários em alguns de seus processos, tornando irrelevante o modelo de negócios centralizado tradicional, vindo a extinguir serviços e até mesmo inviabilizando seu negócio (UPADHYAYA; SHARMA; ARUN, 2017).

No entanto, é crítico neste estágio da evolução do *Blockchain* reconhecer todas as mudanças sociais e de negócios que ela poderá provocar. Uma grande parte da discussão ainda se concentra na própria tecnologia, tornando a visão ainda míope e não reconhecer o *hype*⁶ da tecnologia.

Para compreender o *Blockchain* é necessário conhecer os conceitos de *hash*, ponteiros, listas vinculadas, árvore Merkel e algoritmo de consenso (“O que é hashing?”, 2019) (WALDMAN, 2018).

2.1.1 Funcionamento da função *Hash* Criptográfico

A função *hash* permite verificar a integridade da informação. Funciona da seguinte maneira, recebe uma *string* de qualquer tamanho como entrada e retorna uma saída de comprimento fixo. É fundamental que o algoritmo seja determinístico, ou seja, a mesma entrada resultará a mesma saída. No contexto do *Bitcoin*, as transações são tomadas como entrada, executa o algoritmo de *Secure Hashing*

⁶ *Hype* é o assunto que está "dando o que falar" ou algo sobre o qual todos falam e comentam.

Algorithm 256 (SHA-256) que fornece uma saída de um comprimento fixo de 256 bits (UPADHYAYA; SHARMA; ARUN, 2017). Independentemente do tamanho da entrada, a saída terá sempre o mesmo tamanho, figura 3, e qualquer pequena mudança na entrada gera uma saída totalmente diferente, como mostra a figura 4.

Existem outras funções de *hash* criptográficas, como o MD5 que produz uma saída de 128 bits, o SHA 1 de 160 bits e o Keccak-256 de 256 bits, este último é utilizado na rede *Ethereum*.

Para que uma função *hash* trabalhe com entradas de comprimentos arbitrário, necessita inicialmente ter uma construção de entrada de tamanho fixo e então utilizar o método de transformação *Merkle-Damgard* para permitir o uso de tamanho variável na entrada.

Figura 3 – Exemplo de entrada e saída *hash*

INPUT	HASH
Hi	639EFCDo8ABB273B1619E82E78C29A7DF02C1051B1820E99FC395DCAA3326B8
Welcome	53A53FC9E2A03F9B6E66D84BA701574CD9CF5F01FB498C41731881BCDC68A7C8

Fonte: (“O que é hashing?”, 2019)

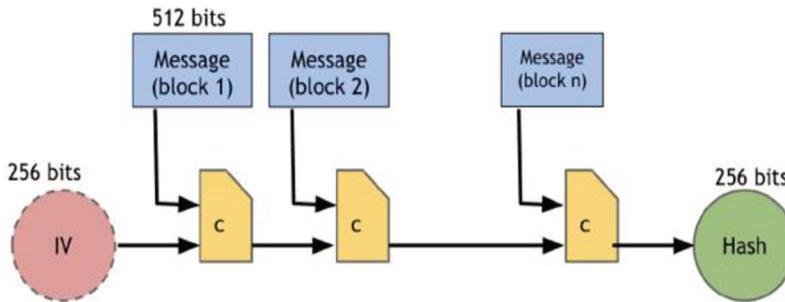
Figura 4 – Pequena mudança na entrada / grande mudança na saída *hash*

INPUT	HASH
This is a test	C7BE1ED902FB8DD4D48997C6452F5D7E309FBCDBE2808B16BCF4EDCE4Co7D14E
this is a test	2E99758548972A8E8822AD47FA1017FF72F06F3FF6A016851F45C398732BC50C

Fonte: (“O que é hashing?”, 2019)

Especificamente no SHA-256, a função de compactação recebe entradas de 768 bits e produz saídas de 256 bits, logo o tamanho do bloco é de 512 bits. Veja na Figura 5 a representação gráfica de como o SHA-256 funciona (NARAYANAN et al., 2016).

Figura 5 – Esquema do funcionamento da SHA-256



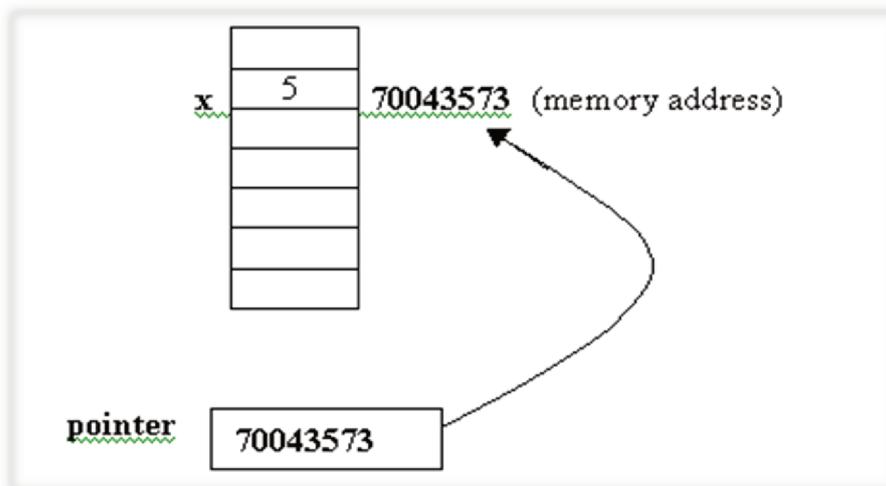
Fonte: (NARAYANAN et al., 2016)

2.1.2 Ponteiros

Na programação uma variável armazena uma informação, que pode ser texto, numérico ou lógico. Os ponteiros armazenam endereço de variáveis, ou seja, ele aponta para a localização de outra variável (“O que é hashing?”, 2019), representado na figura 6.

Podemos usar ponteiros de *hash* para construir todos os tipos de estruturas de dados. Intuitivamente, podemos usar uma estrutura de dados familiar que usa ponteiros como uma lista vinculada ou uma árvore de pesquisa binária e implementá-la com ponteiros de *hash*, em vez de ponteiros (NARAYANAN et al., 2016).

Figura 6 – Variável e ponteiro



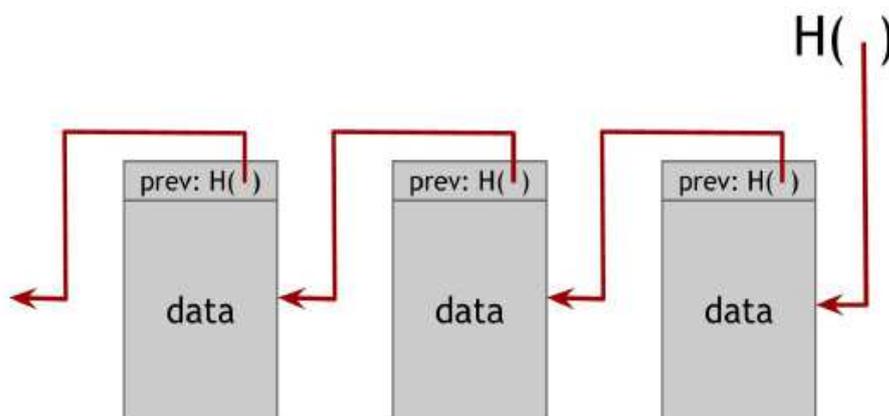
Fonte: (MASTERDAWEB, 2011)

2.1.3 Listas vinculadas

Lista vinculada ou encadeada, é uma sequência de blocos, que contém dados e são vinculados ao próximo bloco por meio de um ponteiro. Neste caso, o ponteiro contém o endereço do próximo bloco, assim o vínculo é realizado. O conteúdo do último bloco deve ser nulo, o que significa que não tem valor, pois está aguardando o surgimento de um novo bloco para ser preenchido com seu endereço. O primeiro bloco é chamado de “gênese” e seu ponteiro está no próprio sistema (“O que é hashing?”, 2019).

Narayanan et al. (2016) apresenta que podemos usar ponteiros de *hash* para construir todos os tipos de estruturas de dados, ver figura 7. As estruturas de dados que usam ponteiros como uma lista vinculada ou uma árvore de pesquisa binária podem ser alteradas para implementar ponteiros de *hash*, em vez de ponteiros normais. O *Blockchain* é uma lista encadeada de ponteiros de *hash*.

Figura 7 – Lista vinculada com ponteiros de *hash*



Fonte: (NARAYANAN et al., 2016)

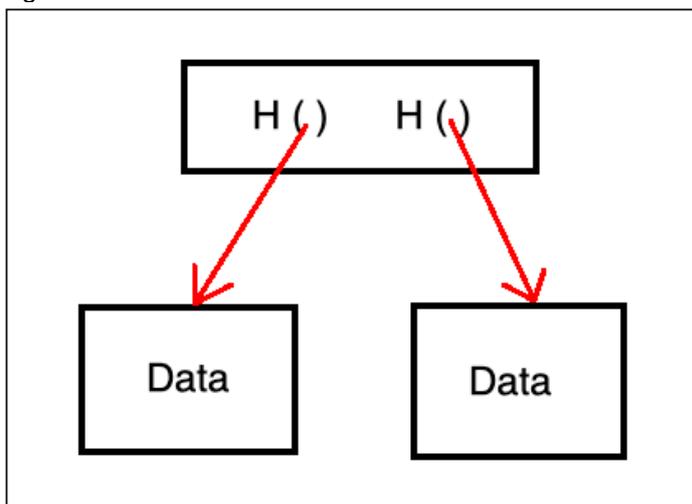
2.1.4 Árvore Merkle

Outra estrutura de dados conhecida, é a árvore binária, mas com uma pequena mudança, com a utilização de ponteiros de *hash* passou a ser chamada de árvore *Merkel*, denominação dada pelo seu inventor Ralph Merkle (UPADHYAYA; SHARMA; ARUN, 2017).

O princípio deste conceito é que os dados estão nas folhas, na base inferior da árvore, então para cada par de folha construir uma estrutura de dados com dois ponteiros de *hash*, um para cada folha do par, repetir para todas as folhas de dados da base, ver figura 8. Depois subir um nível e o bloco no nível conterá um ponteiro de *hash* de dois de seus filhos. Prosseguir por todo o caminho até o topo da árvore, chamada de a raiz da árvore. Será armazenado apenas do ponteiro de *hash* da raiz da árvore, ver figura 9.

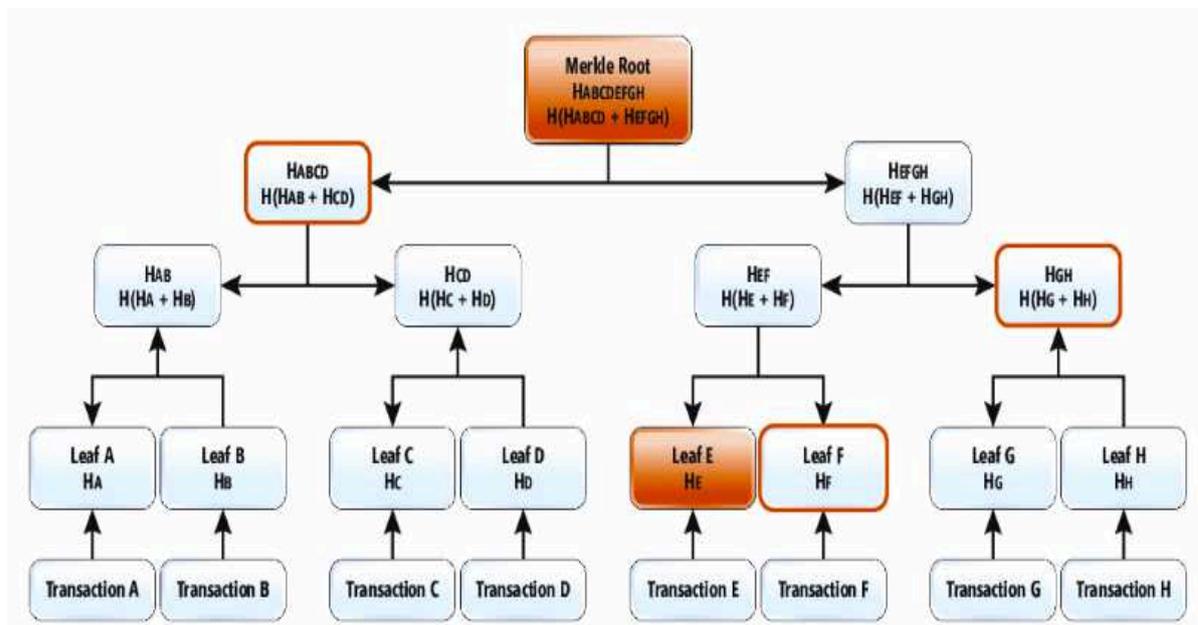
Assim caso um bloco de dados da base seja alterado, o *hash* mudará e não coincidirá com o conteúdo do nível acima. Mas se o nível acima também for calculado novamente então coincidindo como o próximo nível acima, e se isso for realizado sucessivamente até atingir a raiz da árvore, ainda iremos detectar a alteração, porque o *hash* a raiz da árvore que foi salvo não coincidirá com o recalculado. Estas são estruturas que o *Blockchain* usará para garantir a imutabilidade dos dados.

Figura 8 – Estrutura com dois *hashes* dos dois blocos de dados



Fonte: (UPADHYAYA; SHARMA; ARUN, 2017)

Os recursos apresentados nos itens anteriores credenciam para a criação de uma *Blockchain*. As listas vinculadas geram as ligações entre os blocos, os ponteiros de *hash* geram a segurança para não ocorrer adulteração nos blocos e a árvore Merkel garante a inviolabilidade das transações do bloco. Porém para que ocorra uma transação sem a necessidade de uma autoridade central que garanta a confiança, falta conhecer os algoritmos de consenso (WALDMAN, 2018).

Figura 9 – Representação de uma árvore *Merkel*

Fonte: (WALDMAN, 2018)

2.1.5 Algoritmos de consenso

Lamounier (2018) defini algoritmos de consenso como um processo de tomada de decisão por um grupo, onde cada indivíduo constrói e apoia a decisão. Foram projetados para criar igualdade e justiça no mundo *on-line*, ou seja, são responsáveis pela confiabilidade do processo de inclusão de blocos na rede.

Na rede *Blockchain* os algoritmos de consenso fazem com que os “nós” respeitem um protocolo que determina como os blocos são adicionados e mantidos. Existem diferentes tipos de algoritmos de consenso, o mais conhecido o *Proof-of-Work* (PoW), prova de trabalho (WALDMAN, 2018).

Nas redes públicas, o algoritmo de consenso é executado pelos mineradores, nós especiais com grande poder computacional que disputam entre si para resolver o problema matemático complexo definido pelo algoritmo de consenso. O minerador que primeiro resolver o problema recebe uma recompensa, geralmente em criptomoeda.

Upadhyaya, Sharma e Arun (2017) descreve com detalhes como funciona o PoW. O objetivo é encontrar o *hash* do bloco, porém o mesmo deve respeitar algumas

regras de dificuldade definida pela rede. Para resolver o quebra-cabeça matemático é necessário utilizar um número chamado “*nonce*”. Para formar o bloco, o “*nonce*” será concatenado ao *hash* do bloco anterior e as transações que estavam pendentes para serem adicionadas a rede *Blockchain*. Calcula-se o *hash* deste bloco e verifica-se a regra de dificuldade, números de zeros a esquerda do *hash* foi encontrado. Em caso negativo o “*nonce*” é alterado e o processo se reinicia. Em caso afirmativo o bloco é enviado para os demais “nós” validarem o “*nonce*” e se houver consenso entre os “nós” o bloco será adicionado ao *Blockchain* e o *Ledger* distribuído⁷ a todos os “nós”.

Como os mineradores estão competindo por recompensas financeiras, estes frequentemente investem em seu poder computacional para obterem maior êxito na criação de novos blocos.

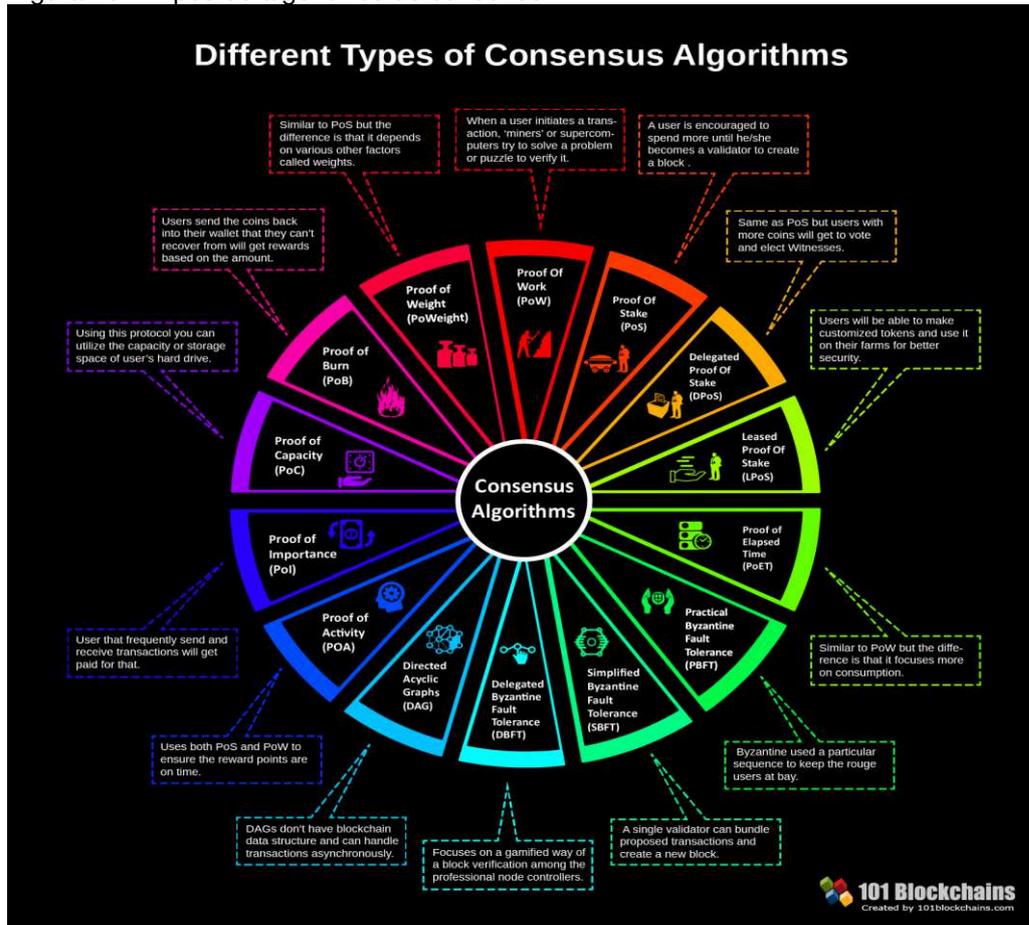
No *Bitcoin* uma das características a ser mantida pela rede é o tempo médio de 10 minutos para a criação de um novo bloco. Portanto uma propriedade de nível de dificuldade foi implementada e o tempo de criação dos blocos é monitorado. No *Bitcoin* mais especificamente a cada 2016 novos blocos esta propriedade é revisada para aumentar ou reduzir a complexidade do cálculo PoW, mantendo o tempo médio nos 10 minutos para cada novo bloco, ou seja, em média a cada 2 semanas ocorre esta revisão (UPADHYAYA; SHARMA; ARUN, 2017).

Outra propriedade importante é que após o *nonce* encontrado deve ser publicado com o bloco, junto com o *hash* do bloco anterior e as transações que se deseja salvar na rede. Isso é imprescindível para que qualquer “nó” consiga executar a função *hash* e verificar se o bloco é válido, isso será verificado por todos os “nós” que receberem a cópia do *Ledger* com o novo bloco incluído. Para que o bloco seja efetivado no *Blockchain* 50% mais 1 dos “nós” tem de confirmar a validação para que a rede entenda que o consenso foi alcançado.

Como abordado anteriormente existem muitos algoritmos de consenso, a figura 10 apresenta um infográfico e o quadro 1 lista algumas diferenças entre eles. Em síntese este quadro reforça a diversidade de algoritmos existentes e suas principais características, pontuando os prós e contras.

⁷ Um ledger distribuído é um banco de dados digital que registra a transação de ativos onde cada transação e seus detalhes são armazenados em vários lugares simultaneamente

Figura 10 – Tipos de algoritmos de consenso



Fonte: (LAMOUNIER, 2018)

Quadro 1 – Comparativo dos Algoritmos de Consenso

continua

Algoritmos de consenso	Plataforma Blockchain	Ano	Linguagem Progr.	Contr.Intel.	Prós	Contras
Prova de Trabalho (PoW)	Bitcoin	2009	C ++	Não	Menor oportunidades para 51% de ataque	Alto consumo de energia
					Melhor segurança	Centralização dos Mineradores
Proof of Stake (PoS)	NXT	2013	Java	Sim	Energia eficiente	Não atende necessidades específicas
					Mais descentralizado	
PoS Delegado	Lisk	2016	JavaScript	Não	Energia eficiente	Parcialmente centralizado
					Escalável	
					Maior segurança	<i>Double Spend</i> ataque
Prova de Participação Alugada	Ondas	2016	Scala	sim	Uso justo	Problema de descentralização
					Moedas de aluguel	
Prova do Tempo Decorrido	Sawtooth Hyperledger	2018	Python, JavaScript, Go, C++, Java e Rust	sim	Participação barata	Necessidade de <i>hardware</i> especializado
						Não é bom para o <i>Blockchain</i> público
Tolerância a Falha Bizantina Prática	Fabric Hyperledger	2015	JavaScript, Python, Java REST e Go	sim	Não há necessidade de confirmação	Lacuna de comunicação
					Redução de Energia	Ataque do tipo <i>Sybil</i>
Tolerância a Falha Bizantina Simplificada	Cadeia	2014	Java, Nó e Ruby	Não	Boa segurança	Indisponível no <i>blockchain</i> público
					Validação de Assinatura	

Quadro 1 – Comparativo dos Algoritmos de Consenso

conclusão

Algoritmos de consenso	Plataforma Blockchain	Ano	Linguagem Progr.	Contr.Intel.	Prós	Contras
Tolerância a Falha Bizantina Delegada	NEO	2016	Python, .NET, Java, C++, C, Ir, Kotlin, JavaScript	sim	Escalável	Conflitos na cadeia
					Rápido	
DAG	IOTA	2015	JavaScript, Rust, Java Go e C++	Em processo	Rede de baixo custo	Lacunas de implementação
					Escalabilidade	Não é adequado para contratos inteligentes
Prova de Atividade	Decred	2016	Ir	sim	Reduz a probabilidade do ataque de 51%	Maior consumo de energia
					Contribuição igual	Assinatura dupla
Prova de Importância	NEM	2015	Java, C++ XEM	sim	transação em colheita	Problema de descentralização
					Parceria de transação	
Prova de Capacidade	Burstcoin	2014	Java	sim	Barato	Favorece os peixes grandes
					Eficiente	Questão de descentralização
					Distribuído	
Prova de Queimadura	Slimcoin	2014	Python, C++, Shell, JavaScript	Não	Preservação da rede	Não é para investidores de curto prazo
					Customizável	Desperdiçando moedas
Prova de Trabalho	Filecoin	2017	SNARK / STARK	sim	Escalável	Problema com Incentivação

Fonte: (LAMOUNIER, 2018)

2.1.6 Chaves públicas e privadas

Nakamura e Geus (2007) definem que *Public Key Infrastructure* (PKI), “é o conjunto de *hardware*, *software*, pessoas, políticas e procedimentos necessários para criar, gerenciar, armazenar, distribuir e revogar certificados digitais com base em criptografia de chave pública”.

Uma PKI é composta de Autoridades de Certificação, Chaves públicas e privadas, Certificados Digitais e Lista de Revogação de Certificados. A chave privada e sua respectiva chave pública possui um relacionamento matemático exclusivo, onde uma mensagem encriptada pela chave pública somente pode ser decifrada pela chave privada, e vice-versa. Assim garantindo que a mensagem assinada digitalmente foi realmente enviada pelo emissor quando a chave pública conseguir decifrar a mensagem. A figura 11 apresenta a estrutura de uma PKI (HYPERLEDGER, 2019).

Em uma PKI a emissão do certificado pela Autoridade de Certificação é o processo que vincula as chaves públicas e privadas a uma pessoa ou organização (BRASIL, 2019).

Figura 11 – Estrutura de uma PKI



Fonte: (HYPERLEDGER, 2019)

2.1.7 Blockchain Pública e Privada

Existem duas categorias de *Blockchain*, as públicas, também chamadas de não permissionadas, e as privadas, chamadas de federadas ou permissionadas. As *Blockchains* públicas são caracterizadas por serem abertas, onde os membros não

se conhecem, não se tem controle de seus participantes, e os mesmos muitas vezes não confiam uns nos outros. Nestas redes o consenso é baseado em mineração como o PoW, e os mineradores competem entre si por recompensas financeiras, e isso exige alto poder computacional. Exemplos desta categoria são o *Bitcoin* e *Ethereum*.

As *Blockchain* privadas são caracterizadas por terem seus membros identificados, autenticados e autorizados. Geralmente fechados em grupos ou consórcios que possuem interesse corporativo comuns. Exemplos desta categoria são o Corda e o *Hyperledger*. O consenso nesta categoria baseia-se no protocolo *Byzantine Fault Tolerance* (BFT) e não necessita de incentivos, pois os participantes têm interesses comuns, e a principal vantagem é a baixa latência na tomada de decisão (GREVE et al., 2018).

2.2 HYPERLEDGER

O *Hyperledger*, que é um esforço colaborativo de código aberto criado para promover a tecnologia *Blockchain* para vários segmentos da indústria. É uma colaboração global, hospedada pela The Linux Foundation, com mais de 130 membros, incluindo líderes em finanças, bancos, internet das coisas, cadeia de suprimentos, manufatura e tecnologia. Possuem atualmente 8 projetos em andamento, incluindo *Hyperledger Fabric* e *Hyperledger* (UPADHYAYA; SHARMA; ARUN, 2017).

O *Hyperledger Fabric* implementa uma arquitetura modular para fornecer uma opção funcional aos designers de rede. Algoritmos específicos para identidade, ordenação (consenso) e criptografia, por exemplo, podem ser conectados a qualquer rede *Hyperledger Fabric* (HYPERLEDGER, 2019).

A IBM tem dedicado esforços com o *Hyperledger Fabric* para criar um *framework* e uma base de código distribuído abertos, padronizados e de nível corporativo, que implementa uma arquitetura modular, onde pode-se plugar componentes como algoritmos de identificação, ordenação e criptografia, conforme as necessidades do arquiteto da rede. Atualmente oferta através do *Bluemix* uma plataforma *Blockchain* na *IBM Cloud*. Alguns dos projetos tocados pela IBM são, uma

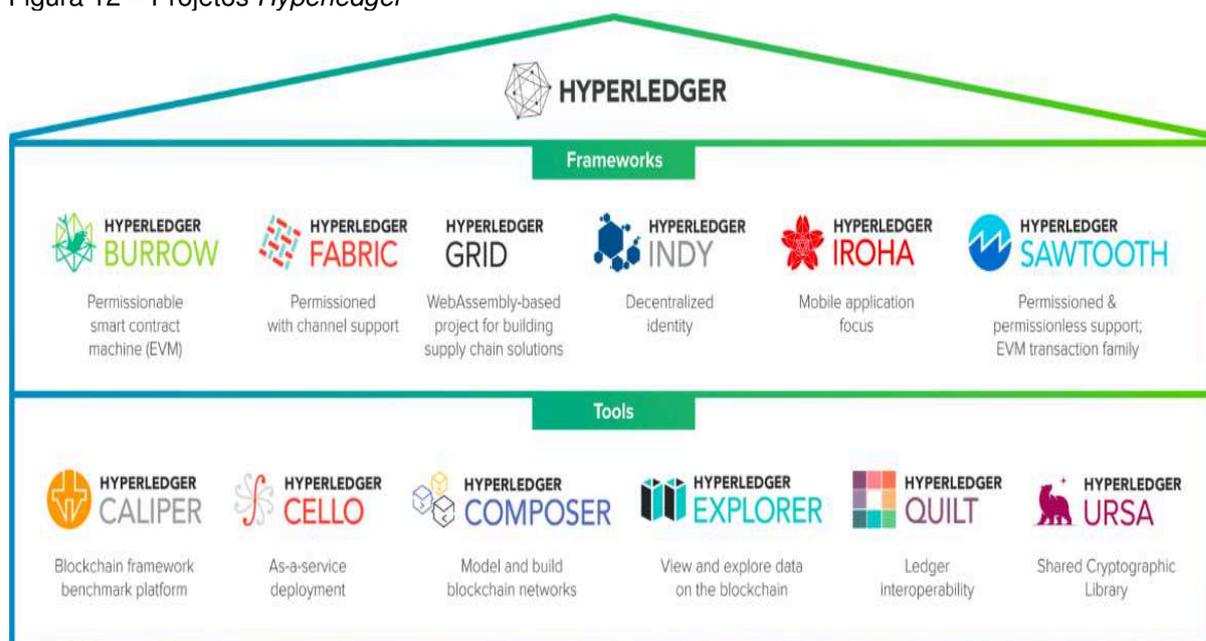
plataforma de *trade finance* para sete bancos europeus, bem como um piloto de bolsa de valores na América do Sul, (UPADHYAYA; SHARMA; ARUN, 2017).

A Microsoft também vem adicionando módulos BaaS (*Blockchain as a Service*) ao *Azure*, sua plataforma de computação em nuvem, desde 2015. Como uma plataforma aberta, flexível e escalável, o *Azure* oferece suporte a um número crescente de tecnologias de contabilidade distribuída que abordam negócios e requisitos técnicos para segurança, desempenho e processos operacionais. O *Azure* disponibiliza em sua plataforma a criação de rede *Blockchain Hyperledger* (MICROSOFT, 2019).

Ambas as plataformas possuem mecanismos para uma construção rápida de maneira a facilitar a implementação e deixar o arquiteto pensar em seu modelo e não na tecnologia em si.

O projeto *Hyperledger* mantido pela fundação Linux mantém em seu guarda-chuva 6 frameworks e 6 ferramentas para uso da tecnologia *Blockchain*, são eles: *Burrow*, *Fabric*, *Grid*, *Indy*, *Iroha*, *Sawtooth*, *Caliper*, *Cello*, *Composer*, *Explorer*, *Quilt* e *Ursa*. Cada um com funcionalidades e aplicações específicas, figura 12. Veremos mais sobre o *Hyperledger Fabric* e o *Composer*.

Figura 12 – Projetos *Hyperledger*



Fonte: <https://www.hyperledger.org/projects>

2.2.1 Contratos Inteligentes

O termo “contratos inteligentes” foi cunhado por Nick Szabo, em meados de 1994, quase no início da própria *World Wide Web* (WWW). Nick um criptógrafo conceituado, definiu as premissas para o contrato inteligente, porém na época não pôde ser implementado pois não existia um sistema financeiro nativo que suportasse transações programáveis. Desta forma se uma instituição financeira tivesse de autorizar ou liberar manualmente a transferência de dinheiro, isso iria contra os propósitos do contrato inteligente (UPADHYAYA; SHARMA; ARUN, 2017).

Um contrato inteligente é um código programável que define a lógica e as regras para uma transação entre diferentes organizações. Este precisa ser empacotado em um *chaincode*⁸ para ser implementado em uma *Blockchain* (HYPERLEDGER, 2019). Estes contratos são executados automaticamente quando uma condição pré-programada for acionada, assim uma cláusula do contrato é executada.

A tecnologia de contrato inteligente está sendo construída sobre o *Bitcoin* e outras moedas virtuais e como o *Bitcoin* além de ser uma criptomoeda, é um protocolo, um software, os contratos inteligentes podem falar com ele da mesma forma como fariam com qualquer outro código. As peças do quebra-cabeça estão se encaixando e agora um programa de computador pode acionar pagamentos (UPADHYAYA; SHARMA; ARUN, 2017).

2.2.2 Hyperledger Fabric

Diferentemente de uma *Blockchain* pública, os membros de uma rede *Hyperledger Fabric* se inscrevem através de um *Membership Service Provider* (MSP) confiável. Os principais componentes de uma rede *Blockchain Hyperledger Fabric*: Aplicativos, Pares (nós), Contratos inteligentes, *Ledgers*, Canais, Organizações, Ordenador e Autoridade Certificadora (HYPERLEDGER, 2019). Todos estes componentes são definidos abaixo:

⁸ Chaincode, também chamado de **contrato inteligente**, é essencialmente a lógica de negócios que governa como as diferentes entidades ou partes em uma rede blockchain interagem ou transacionam umas com as outras

O aplicativo é o meio que se terá acesso a *Blockchain*, se conectando aos pares e realizando transações ou consultas.

Os pares são elementos fundamentais ao *Hyperledger*, estes são os participantes autorizados da rede, ou seja, os “nós” da rede e hospedam os *Ledgers* e os Contratos inteligentes. Todos os Pares associados a um Canal compartilham e guardam cópias do mesmo *Ledger* e Contratos inteligentes, figura 14.

O ordenador é o responsável por organizar as transações em blocos e direcionar a inclusão no *Ledger*.

Os canais permitem que o *Hyperledger Fabric* crie um meio de comunicação entre grupos de participantes, definindo um ambiente de colaboração entre os grupos e ao mesmo tempo gerando privacidade, pois cada Canal terá seu próprio *Ledger*, ou seja, outros Pares que não tenha acesso ao Canal não poderão acessar estas transações. Um Par poderá participar de vários Canais simultaneamente, e deverá manter uma cópia do *Ledger* de cada Canal. O *Ledger* é composto por dois componentes: o Estado mundial⁹ e o *Log* de transação.

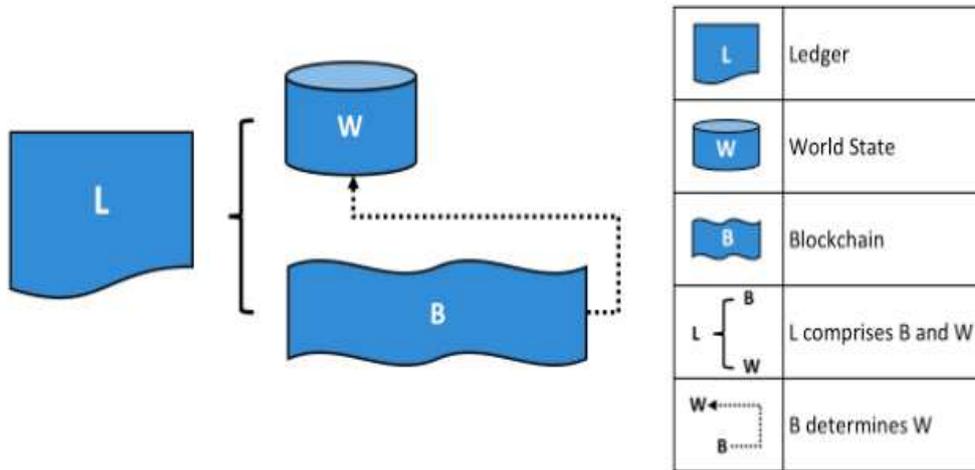
As redes *Hyperledger Fabric* são formadas por “Nós” de diferentes Organizações, que necessitam ser identificadas por fazerem parte de uma rede privada. Quando a rede for configurada, as políticas devem ser especificadas para o Canal e as Organizações devem concordar com estas políticas, fato determinante para fazer parte da rede.

Uma Autoridade Certificadora emite um Certificado Digital X.509 para cada Organização, que será a Identidade da Organização nas transações neste Canal ou rede.

O componente do Estado mundial descreve o estado do *Ledger* em um determinado momento no tempo, é um banco de dados. Este registra um par “chave-valor” geralmente em formato *JavaScript Object Notation* (JSON). O componente de *Log* de transações registra todas as transações que atualizam o valor atual do Estado mundial ou não, é o histórico da rede, a *Blockchain* propriamente dita. A figura 13 apresenta a composição de um *Ledger*.

⁹ Estado mundial, é um repositório de dados que mantém a visão do estado atual de um ativo

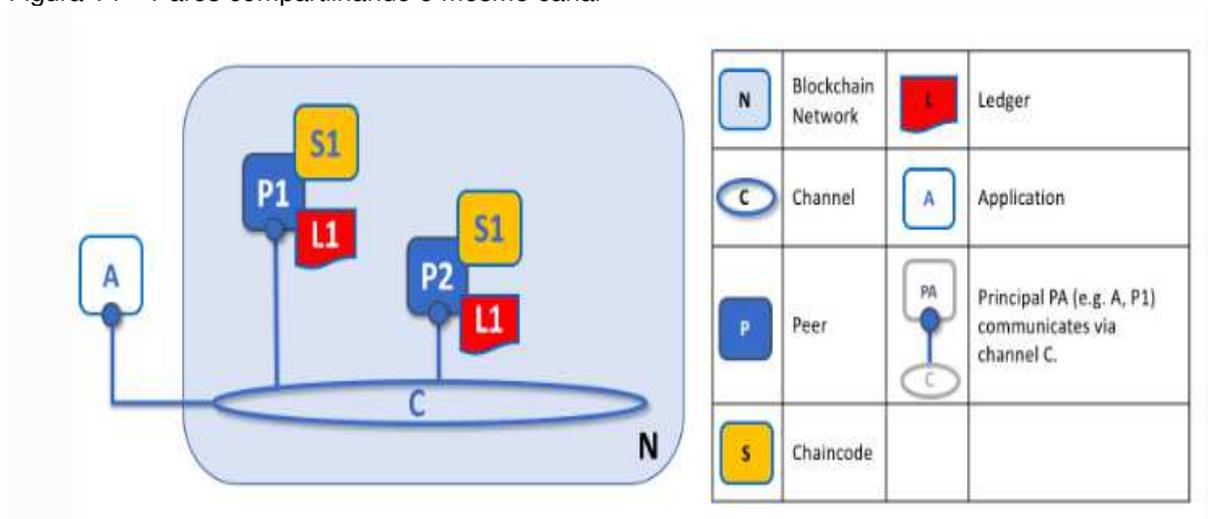
Figura 13 – Ledger no Hyperledger



Fonte: (HYPERLEDGER, 2019)

Quando um *chaincode* é instanciado em um canal, uma política de endosso é definida para ele; Todos os contratos inteligentes dentro do *chaincode* são disponibilizados para o canal (HYPERLEDGER, 2019).

Figura 14 – Pares compartilhando o mesmo canal



Fonte: (HYPERLEDGER, 2019)

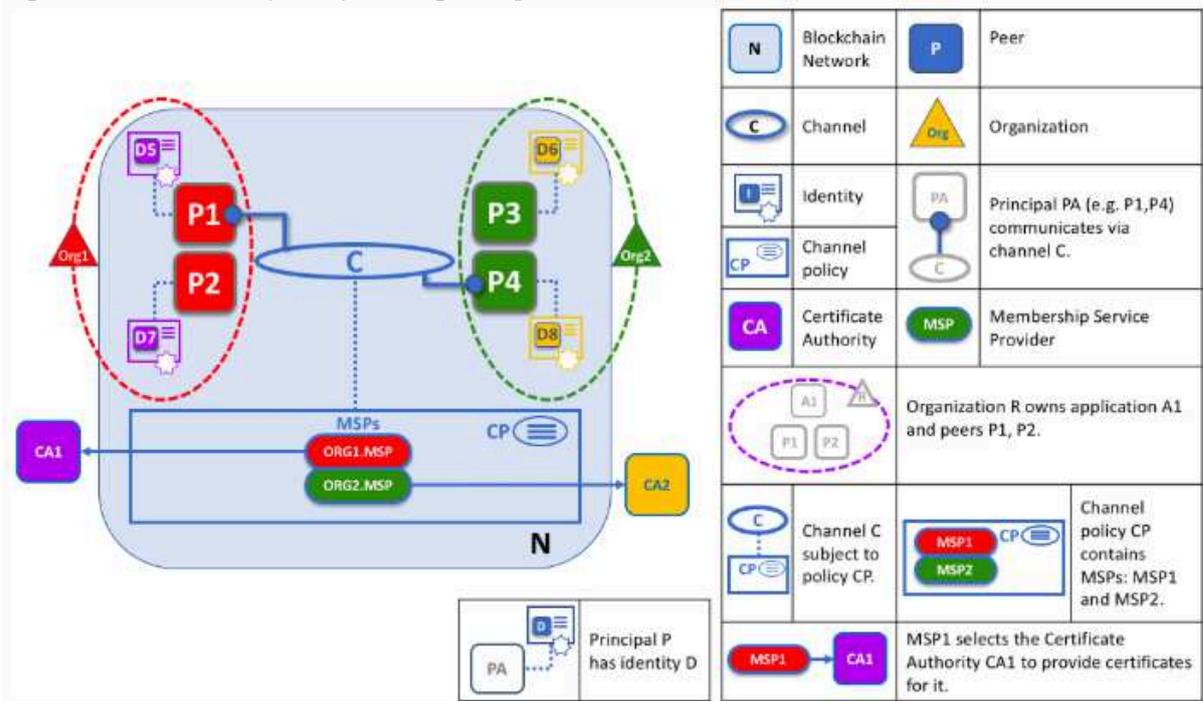
Um MSP é definido para a rede, ver figura 15, e sua principal função é abstrair o processo de emissão e validação de certificados, além de autenticação de usuários.

Outro “Nó” importantíssimo é o Ordenador, “Nó” eleito para executar as funções de consenso, este registra o *Ledger* e os Contratos inteligentes. Os Aplicativos precisam se conectar aos Pares para realizar consulta ou atualização do *Ledger*. As consultas são mais rápidas pois o Par resolve tudo com seu próprio *Ledger* sem necessidade de trocar mensagens com outros Pares.

Uma atualização depende de mais etapas. O processo se dá em três fases, e garante que todos os Pares em uma rede *Blockchain* mantenham seus *Ledgers* consistentes entre si. Na primeira fase, os Aplicativos trabalham com os Pares endossantes que assinam a transação, mas ainda não aplicam a atualização proposta em sua cópia do *Ledger*. Na segunda fase, esses endossos separados são coletados e juntados em uma transação enviada ao Ordenador, que recebe as transações até que o bloco atinja o tamanho específico ou o tempo de espera tenha expirado, então ordena e empacota em blocos. Na fase final, esses blocos são distribuídos aos Pares onde cada transação é validada conforme as políticas de endosso e verifica o Estado mundial dos Ativos¹⁰. Caso a transação seja validada o *Ledger* é atualizado e todos os Pares recebem notificação e toda a rede estará integrada e sincronizada, e os Aplicativos também são notificados sobre suas transações, se foram confirmadas ou invalidadas. Toda transação válida ou não é registrada no *Ledger* com uma marcação específica, porém somente as transações válidas alteram o Estado Mundial de um Ativo. A figura 16 apresenta uma rede *Hyperledger* completa.

¹⁰ Ativo é a representação no blockchain de um ativo tangível, como uma casa, ou intangível, como um serviço.

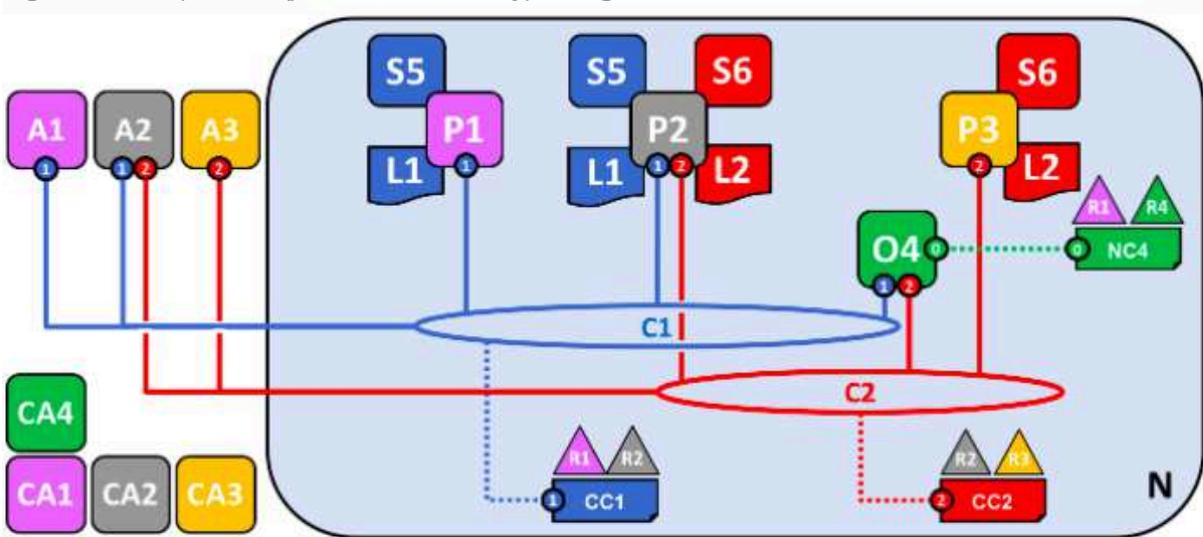
Figura 15 – Rede composta por 2 organizações



Fonte: (HYPERLEDGER, 2019)

Assim através destes passos se dá o Consenso no *Hyperledger*, em várias etapas, porém com velocidade muito superior as *Blockchains* públicas baseadas em PoW.

Figura 16 – Representação de uma rede *Hyperledger*



Fonte: (HYPERLEDGER, 2019)

2.3 HYPERLEDGER COMPOSER

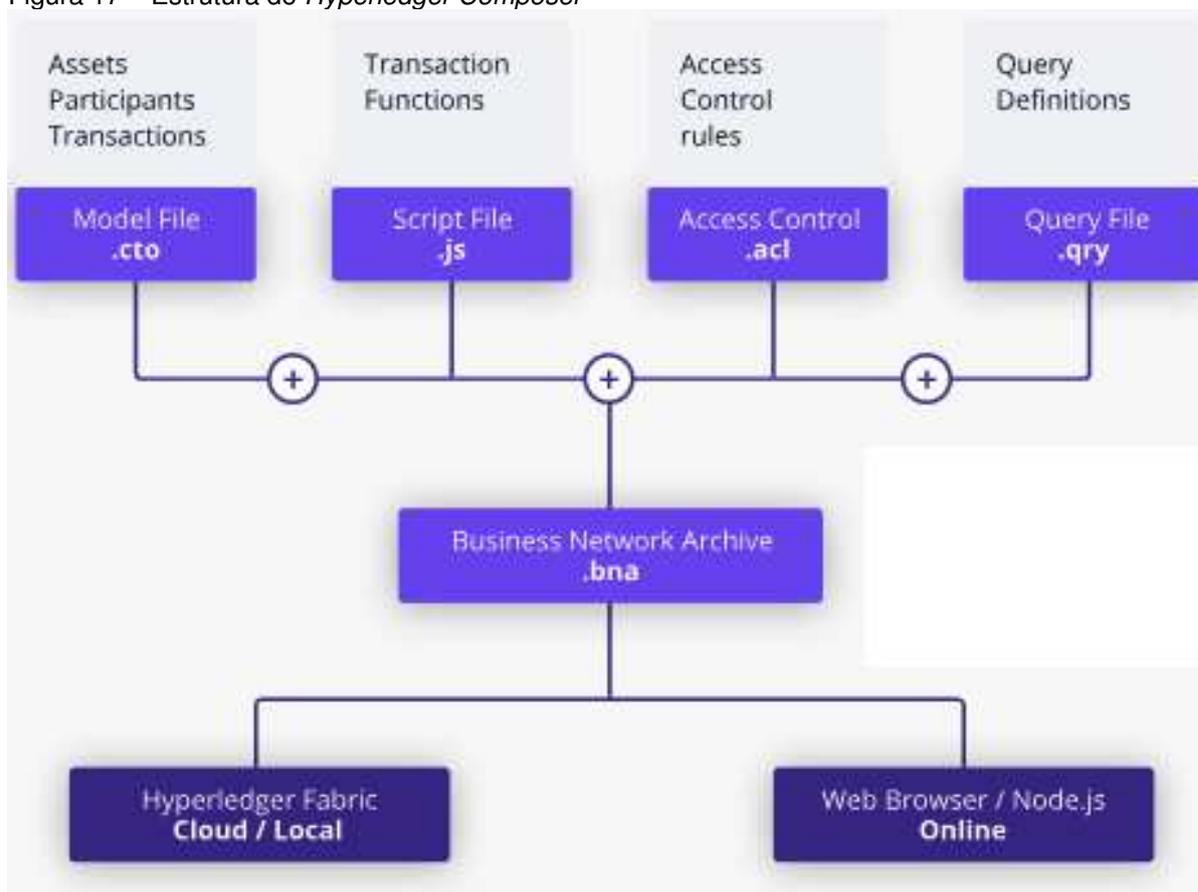
O *Hyperledger Composer* é uma ferramenta de código aberto incubada no projeto *Hyperledger* da *Linux Foundation*. Seu principal objetivo é facilitar e acelerar o desenvolvimento de aplicativos *Blockchain* para negócios. Todos os recursos são voltados para uma implementação rápida de um projeto *Blockchain* afim de modelar e testar uma aplicação de negócios, ou seja, ideal para gerar prova de conceito de um projeto (“Docs Hyperledger Composer”, 2019).

Há duas maneiras de usar o *Hyperledger Composer*. A primeira forma é através de uma versão *on-line* com a interface baseada em navegador chamada de *Hyperledger Composer Playground*, sem necessidade de instalações. A segunda opção é utilizar a versão *off-line* que depende da instalação de recursos como: CLI, *Composer Playground off-line* e *Hyperledger Fabric*. Além de ter instalado os pré-requisitos *cURL*, *Docker*, *NodeJs* e *NPM*. Os sistemas operacionais podem ser *MacOS*, *Linux* ou *Windows*.

Os componentes principais para a implementação são Ativos, Participantes e Transações, e possui um modelo de programação que contém uma linguagem de modelagem e um conjunto de APIs. Com estes recursos é possível criar aplicativos de negócios que permitem aos Participantes enviar Transações que trocam Ativos.

Para definir a rede de negócios no *Hyperledger Composer*, são usados a seguinte estrutura: *Model file*, *Script file*, *Access Control* e *Query File*. Uma rede após definida pode ser exportada ou importada, através da estrutura *Business Network Archive* (BNA), vide figura 17.

Figura 17 – Estrutura do *Hyperledger Composer*



Fonte: <https://hyperledger.github.io/composer/latest/introduction/introduction>

2.3.1 Detalhes da estrutura *Hyperledger Composer*

2.3.1.1 *Model File*

Ativo: Os ativos representam bens tangíveis e/ou intangíveis, ou serviços. Devem possuir um identificador único e podem conter propriedades para sua representação.

Participante: São os membros da rede, devem possuir um identificador único e também podem conter propriedades. Os participantes podem transacionar e possuir Ativos.

Transações: São os mecanismos pelos quais os Participantes podem alterar e consultar o estado atual de um Ativo.

2.3.1.2 Script File

A forma como o participante interage com o Ativo é por meio das transações. Estas são funções escritas na linguagem *JavaScript* e são acionadas em tempo de execução. Uma função pode manipular variáveis, realizar cálculos, acessar e alterar o *Ledger* e gerar eventos. Uma transação possui controle e em caso de falhar nenhuma alteração será realizada, somente em caso de sucesso que as alterações são aplicadas ao *Ledger*. O processador de transações permite acesso a API nativa do *Hyperledger Fabric* através da função `getNativeAPI`, sendo que as regras de controle de acesso não serão respeitadas nestas chamadas de função.

2.3.1.3 Access Control

Local para definição do controle de acesso de negócio e o controle de acesso à rede comercial. O acesso à rede é concedido usando o *namespace* do sistema. Para acessar o sistema usar o *namespace* `org.hyperledger.composer.system` e para acessar a rede o *namespace* `org.hyperledger.composer.system.Network`.

Existe uma sintaxe específica para definição das regras e estas são avaliadas de forma ordenada e definem se o acesso será permitido ou negado.

2.3.1.4 Query File

Possui uma linguagem específica para realizar consultas ao *Ledger*, aos históricos das transações. Estas consultas devem ser salvas todas em um único arquivo na definição da rede. Deve ser informado duas propriedades, *description* e *statement*, esta última aceita as seguintes declarações: *SELECT*, *FROM*, *WHERE*, *AND*, *OR*, *CONTAINS* e *ORDER BY*.

O *Hyperledger Composer* suporta dois tipos de consultas. As consultas nomeadas são especificadas na definição de rede comercial e são expostas como métodos GET pelo componente *composer-rest-server*. As consultas dinâmicas são construídas e executadas em tempo de execução em uma função do Processador de Transação ou no código do cliente. Estas podem ser utilizadas no *Hyperledger Fabric*,

não no Playground. As regras de acesso também serão aplicadas ao resultado da consulta, ou seja, se a consulta retornar um determinado conteúdo que o usuário não tenha permissão de acesso, este não será exibido.

2.3.1.5 *Business Network Archive*

Um arquivo com extensão “.bna” é um empacotamento das definições de uma rede *Blockchain* definida pelo *Hyperledger Composer*. Este arquivo pode ser utilizado para salvar as definições da rede e carregar em outro ambiente, inclusive podendo interoperar entre as versão *off-line* e *on-line* do *Hyperledger Composer Playground*.

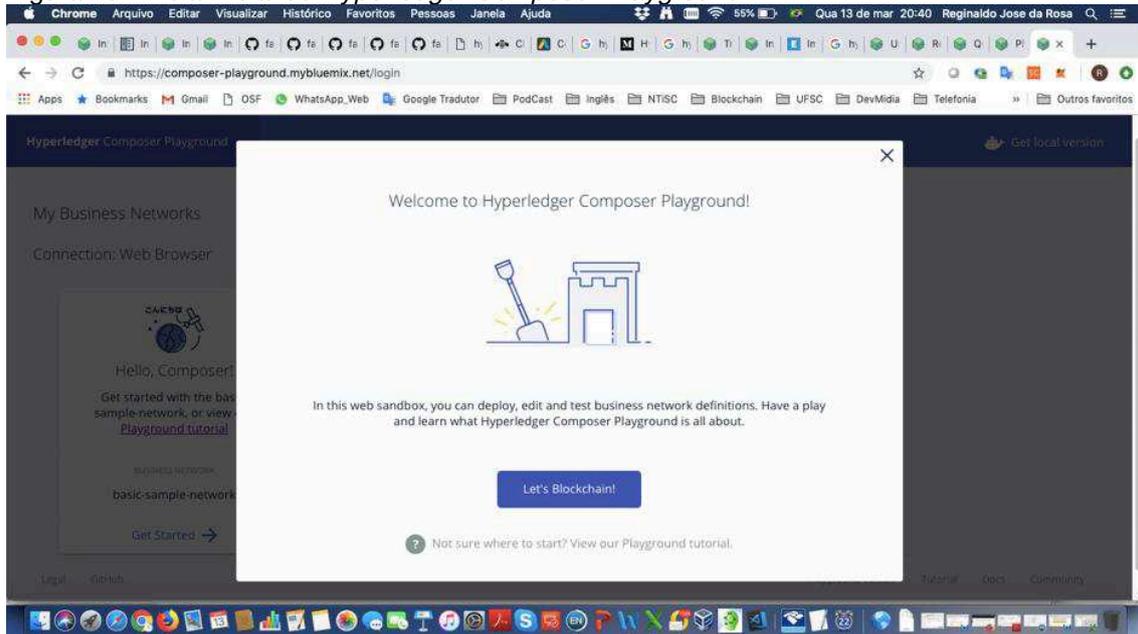
2.3.2 *Hyperledger Composer Playground*

Este é o ambiente baseado em navegador do *Hyperledger Composer*, pode ser acessado pelo link <https://composer-playground.mybluemix.net>, ver figura 18.

Inicialmente apresenta as redes definidas para serem conectadas, figura 19, ou permite a criação de uma nova rede com base em alguns modelos ou sem definição alguma para ser configurada do zero, figura 20.

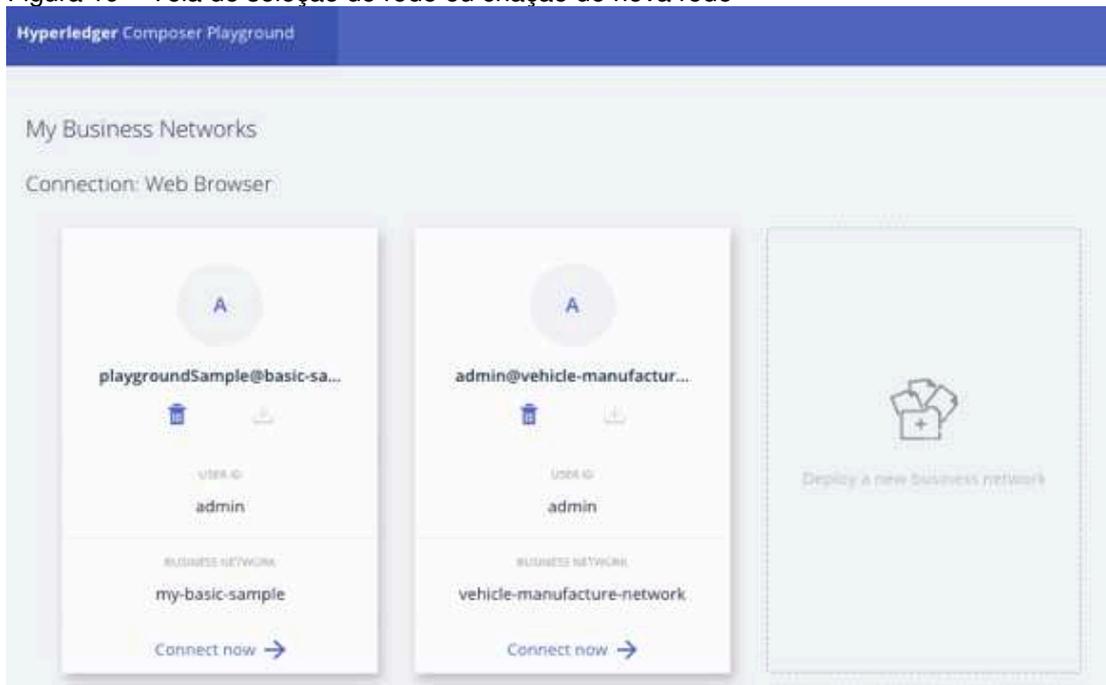
Após iniciar a rede de negócios, apresenta o painel de definição da rede, dos ativos, participantes, transações, eventos e regras de controle de acesso, figura 21. Ao clicar sobre da caixa da esquerda, *Model*, *Script* ou *Access Control* abrirá uma caixa para a programação correspondente, sendo que validação sintática é realizada automaticamente. Para que a rede possa ser utilizada, deve-se realizar o *deploy* para que a rede seja publicada. Ao clicar no botão “*Export*” será gerado o arquivo das definições da rede (.bna).

Figura 18 – Tela inicial do *Hyperledger Composer Playground*



Fonte: <https://composer-playground.mybluemix.net>

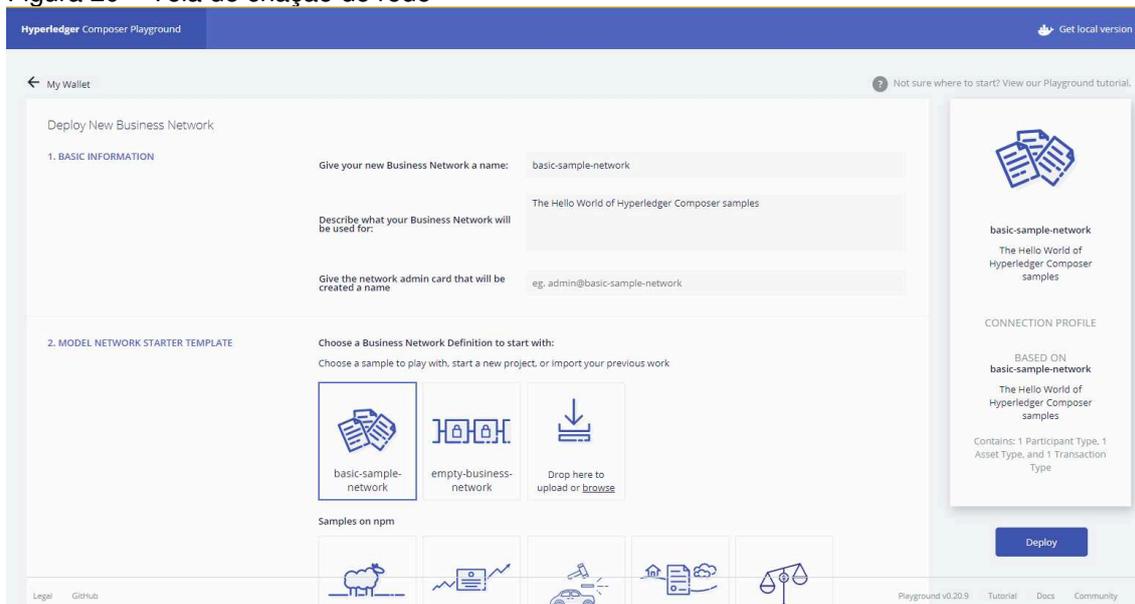
Figura 19 – Tela de seleção de rede ou criação de nova rede



Fonte: <https://composer-playground.mybluemix.net>

Após a publicação da rede, *deploy*, a mesma pode ser testada, e para isso basta clicar na aba “Test” no topo da página, ver a tela na figura 22.

Figura 20 – Tela de criação de rede

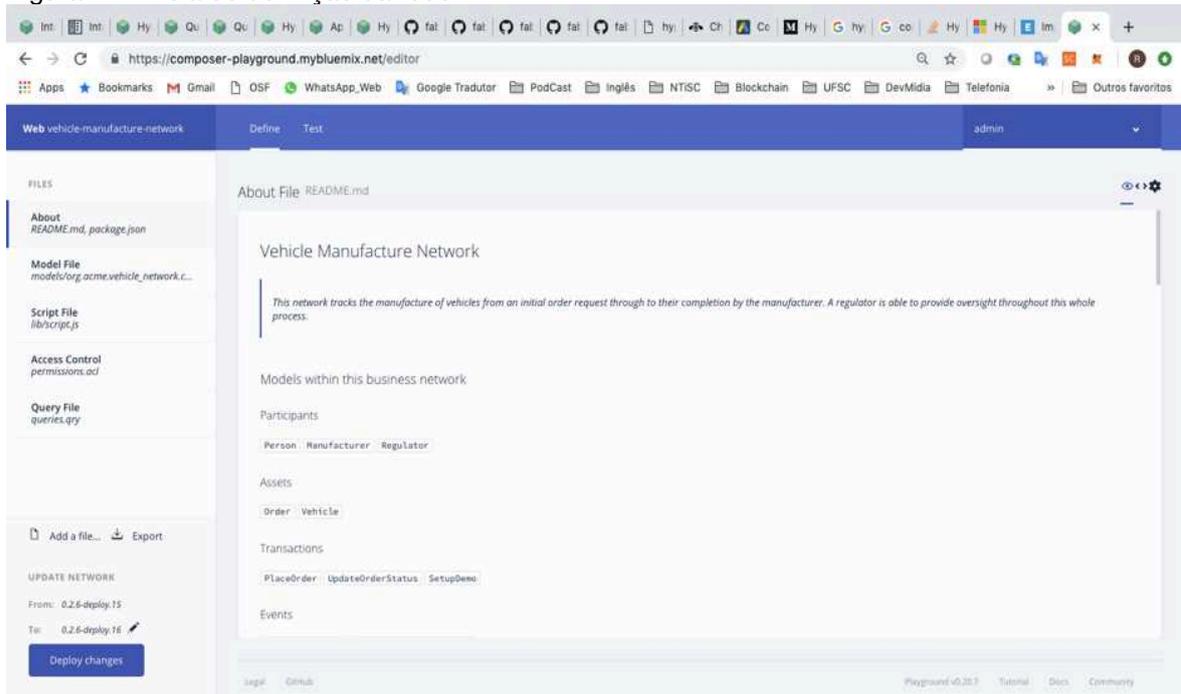


Fonte: <https://composer-playground.mybluemix.net>

A esquerda é listada os Participantes e Ativos programados na etapa da definição. Abaixo as Transações executadas, ou seja, o histórico da rede, pois as alterações nos Ativos são alteradas diretamente nos Ativos. Para ver o conteúdo dos Ativos e Participantes, basta clicar sobre um deles e o conteúdo será apresentado numa lista em formatação JSON. Para executar um *Script* ou Transação é necessário utilizar o botão “*Submit transaction*”, que apresenta uma tela onde deve ser selecionado a transação a ser executada. Em formato JSON apresenta as propriedades que necessitam ser preenchidas ou optar pelo preenchimento aleatório e automático. Ao clicar em “*Submit*” o processador de transações irá executar linha a linha de programação da Transação (*Script*) selecionado, onde *Ledger* e o Estado mundial da rede será atualizado conforme as regras definidas, ou seja, este seria o contrato inteligente da rede em execução.

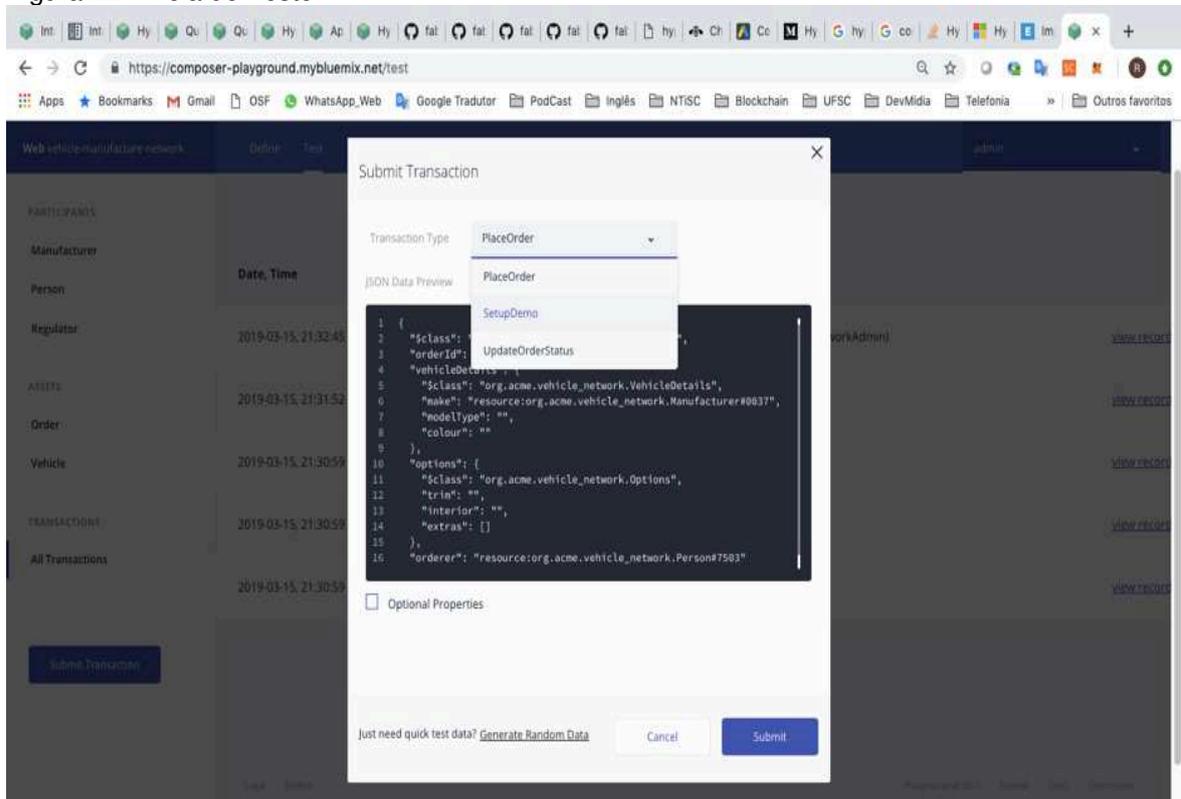
Existem algumas transações padrões que veem definidas na interface do *Hyperledger Composer Playground*, estas executam a Eliminação e Alteração nos Participantes e Ativos, e quando utilizadas registram o Histórico da Rede, garantindo a característica de imutabilidade do *Blockchain*. As transações realizadas, histórico, permite somente ser consultado, não pode ser alterado nem apagado.

Figura 21 – Tela de definição da rede



Fonte: <https://composer-playground.mybluemix.net>

Figura 22 – Tela de Teste



Fonte: <https://composer-playground.mybluemix.net>

2.3.3 Linguagem de Modelagem do *Hyperledger Composer*

O *Hyperledger Composer* possui uma poderosa linguagem orientada a objetos. A documentação oficial do *Hyperledger Composer* possui em detalhes esta linguagem, aqui será feito um breve resumo.

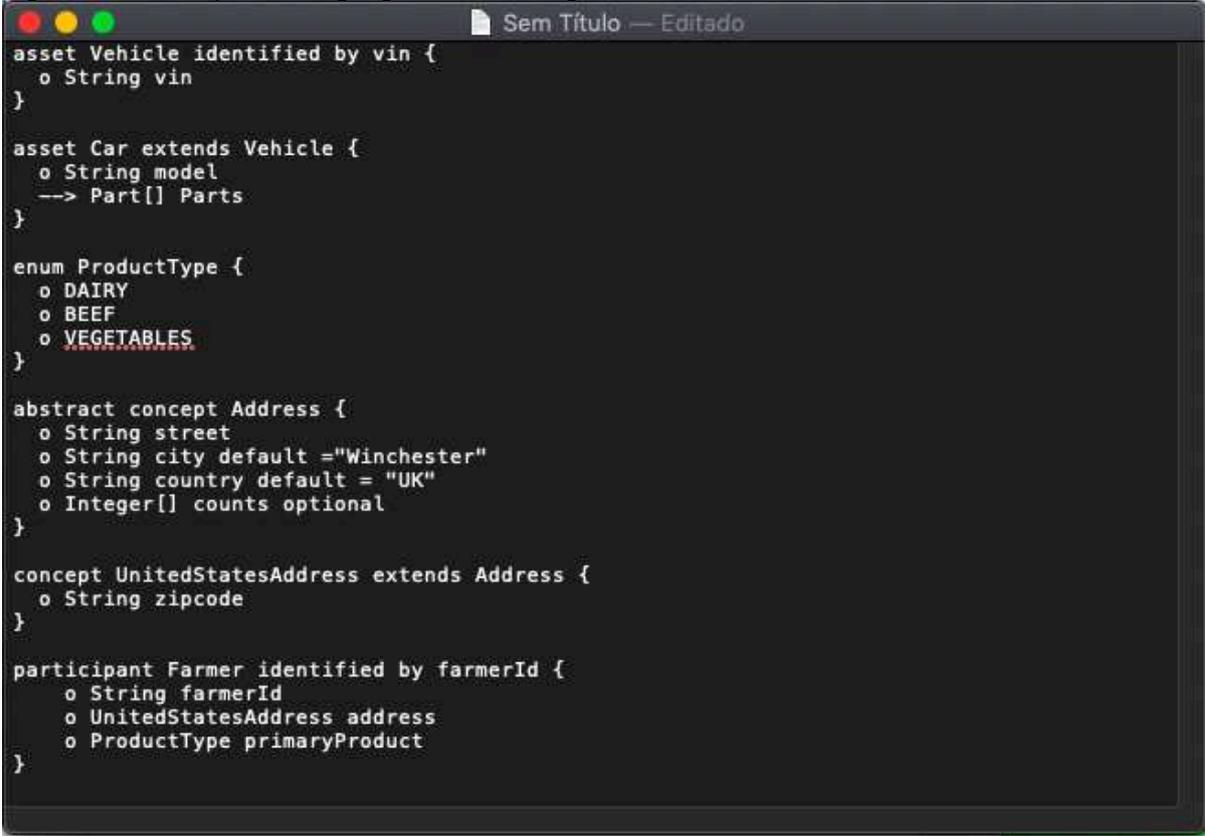
No arquivo *Model File*, deve conter um único “*namespace*” que conterà todas as definições da rede. Existe também um *namespace* de sistema que contém definições básicas como: Ativos, Participantes, Transações, Eventos, Definições de registros, Registros de históricos, Identificadores e Transações de sistema.

Para construção da definição da rede o *Hyperledger Composer* possui as classes: *Asset* e *Participant*, estas definem um identificador através da declaração (*identify by*). As classes *Transaction* e *Event* os identificadores são definidos automaticamente. Uma declaração “*abstract*” quando utiliza antes da classe indica que a mesma não poderá ser instanciada, somente utilizada para estender outra classe. A declaração “*concept*” cria uma espécie de estrutura de propriedades e pode ser utilizado para definir um tipo de uma propriedade, logo está armazenará em seu nome todas as propriedades relacionadas na definição. A declaração “*eNum*” define um tipo que contém de 1 a N elementos. A declaração “*-->*” define um relacionamento com outra classe, como uma chave estrangeira.

As classes possuem propriedades compostas por tipo e o nome da propriedade. Tipo podem ser os primitivos (*String, Integer, Double, Long, Datetime* e *Boolean*), um tipo criado pela declaração “*concept*”, um *eNum* ou uma Classe. Um *array* pode ser definido pela notação [] (colchetes) adicionado logo após o tipo da propriedade. Se a declaração “*optional*” for especificada no final da definição de uma propriedade, esta terá o preenchimento não obrigatório. Exemplos são visualizados na figura 23.

Uma classe *Transaction* e outra *Event* necessita ser definida para então permitir sua programação no modulo *Script File* como será visto no próximo tópico.

Figura 23 – Exemplos da linguagem de modelagem



```

asset Vehicle identified by vin {
  o String vin
}

asset Car extends Vehicle {
  o String model
  --> Part[] Parts
}

enum ProductType {
  o DAIRY
  o BEEF
  o VEGETABLES
}

abstract concept Address {
  o String street
  o String city default = "Winchester"
  o String country default = "UK"
  o Integer[] counts optional
}

concept UnitedStatesAddress extends Address {
  o String zipcode
}

participant Farmer identified by farmerId {
  o String farmerId
  o UnitedStatesAddress address
  o ProductType primaryProduct
}

```

Fonte: <https://composer-playground.mybluemix.net>

2.3.4 Linguagem de *Script* do *Hyperledger Composer*

Para construir a lógica da rede, utiliza-se a linguagem *JavaScript*, porém deve-se respeitar algumas regras na sua programação.

Primeiro as transações devem ser definidas no arquivo de modelagem e então criado a programação na seção *Script*.

Como regra cada função deve ser iniciada com comentários e necessariamente definir as linhas como: A primeira linha uma descrição legível do que faz a função. A segunda linha utiliza a *tag @param* e define o nome da função seu parâmetro e uma descrição a respeito do parâmetro. A terceira linha, a *tag @return* indica um retorno para a aplicação cliente, deve ser seguido pelo tipo do retorno, sendo um dos tipos primitivos ou tipos modelados na seção *Model File* da rede *Hyperledger Composer*. Este retorno não é obrigatório, porém se utilizado deve ser definido também na seção de modelagem da rede. A última linha obrigatoriamente

deve possuir a tag `@transaction` e indica ao processador de funções que o código abaixo é uma transação. Um exemplo na figura 24.

Figura 24 – Código da construção de uma transação

Arquivo de modelo:

```
namespace org.sample

@returns(Integer[])
transaction MyTransaction {

}
```

Função do processador de transação:

```
/**
 * Handle a transaction that returns an array of integers.
 * @param {org.sample.MyTransaction} transaction The transaction.
 * @returns {number[]} The array of integers.
 * @transaction
 */
async function myTransaction(transaction) {
  return [1, 2, 3];
}
```

Fonte: <https://composer-playground.mybluemix.net>

Eventos podem ser emitidos pelas transações, e podem ser assinados por aplicativos externos. Os eventos devem ser declarados na seção de *Model File* da rede, e emitidos de dentro de uma transação, porém não são emitidos quando da execução do seu código, mas sim após a transação ser confirmada. Para emitir um evento são necessários usar 3 funções: `getFactory`, `factory.newEvent()` e `Emit()`, como visto na figura 25.

Figura 25 – Código da construção de um evento

Arquivo de modelo:

```
namespace org.sample

event BasicEvent {
}
```

Função do processador de transação:

```
/**
 * @param {org.namespace.BasicEventTransaction} basicEventTransaction
 * @transaction
 */
async function basicEventTransaction(basicEventTransaction) {
  let factory = getFactory();

  let basicEvent = factory.newEvent('org.namespace', 'BasicEvent');
  emit(basicEvent);
}
```

Fonte: <https://composer-playground.mybluemix.net>

3 LOGISTICA REVERSA

Como muitos conceitos e processos tiveram origem no setor militar, a Logística também teve sua origem na guerra, como a área que cuidava do planejamento de vários itens importantes, armazenamento, distribuição e manutenção de vários tipos de materiais, como armas, roupas, além de alimentos, saúde, transportes e etc.

A logística reversa é “um ramo da logística que remete para a movimentação de um determinado produto, desde o ponto onde foi consumido até o ponto onde foi produzido”, (SIGNIFICADOS, 2014).

O termo Logística Reversa não é novo, têm-se relatos desde os anos 70 na literatura internacional. Nos anos 80 ganhou força o movimento, porém somente em 1990 passou a ser implantado e em 1995 que os primeiros trabalhos relacionando a LR com as questões socioambientais são publicados (COUTO; LANGE, 2017). A figura 26 apresenta o ciclo da LR.

A PNRS deixa claro que a LR é um instrumento utilizado para ampliar o correto reaproveitamento ou destinação de resíduos sólidos gerados após o uso de produtos introduzidos no mercado pelas indústrias, principalmente as embalagens.

Logística Reversa - conforme define o artigo 3º, inciso XII da PNRS, significa o "instrumento de desenvolvimento econômico e social caracterizado por um conjunto de ações, procedimentos e meios destinados a viabilizar a coleta e a restituição dos resíduos sólidos ao setor empresarial, para reaproveitamento, em seu ciclo ou em outros ciclos produtivos, ou outra destinação final ambientalmente adequada"; (BRASIL, 2010a).

As leis 11.107/2005 e 11.445/2007 tratavam sobre o manejo dos Resíduos Sólidos Urbanos (RSU), dos serviços de saneamento básico e a limpeza urbana, além do abastecimento de água, da coleta e tratamento de esgoto e da drenagem urbana. Mas a lei 12.305 de agosto de 2010 institui a PNRS que obriga a criação e implantação de sistemas de LR, desde que sua reutilização ou reciclagem não gere prejuízos maiores ao meio ambiente (BRASIL, 2010b)(COUTO; LANGE, 2017). A PNRS institui e detalha a responsabilidade compartilhada entre poder público, fabricantes, importadores, distribuidores, comerciantes e os consumidores, na implantação do

SLR. A figura 27 apresenta o fluxo de informação e relacionamento entre os atores do SLR definido na lei 12.305/2010.

Art. 30. É instituída a responsabilidade compartilhada pelo ciclo de vida dos produtos, a ser implementada de forma individualizada e encadeada, abrangendo os fabricantes, importadores, distribuidores e comerciantes, os consumidores e os titulares dos serviços públicos de limpeza urbana e de manejo de resíduos sólidos, (BRASIL, 2010a).

Couto e Lange (2017) compilaram em sua pesquisa, vide figura 28, os desafios do setor produtivo no Brasil relativo à implantação do SLR e os aspectos políticos e legais.

Figura 26 – Ciclo da LR

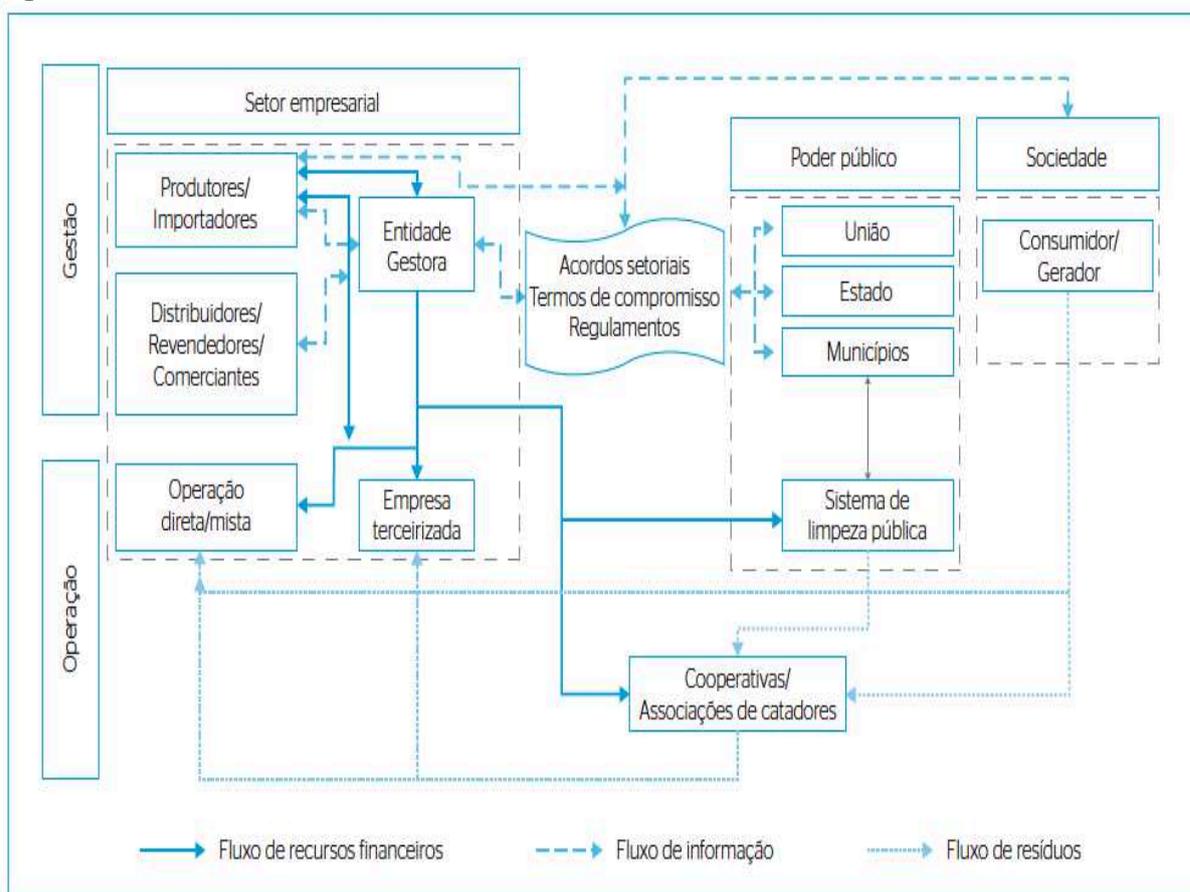


Fonte: <https://unicargo.com.br/2019/05/01/ja-ouviu-falar-em-logistica-reversa/>

Para contribuir e validar a importância do ciclo da LR, o trabalho de Minoves et al. (2015) realizado avaliando uma siderúrgica que em seu processo produtivo

utiliza minério de ferro e sucata como matéria-prima pontua a importância da utilização da sucata na conservação do meio ambiente. Também reporta sobre a extração do minério de ferro ter ação bastante agressiva ao meio ambiente e que a utilização da sucata originada do SLR, também tem benefícios financeiros. Um fator negativo levantado é a distância de onde as sucatas são transportadas, necessitando gestão de estoque e utilização de prepostos para se armazenamento. Por isso um SLR precisa ser bem estruturado para melhorar alguns indicadores. Outro ponto positivo para uso da sucata é a redução da emissão de gases e partículas prejudiciais ao ser humano eliminados no processo utilizando o minério de ferro. Fica evidente a importância da necessidade de um bom SLR e que este processo traz benefícios econômicos além dos ecológicos.

Figura 27 – Fluxo entre os atores do SLR



Fonte: (COUTO; LANGE, 2017)

Figura 28 – Aspectos políticos e legais da implantação do SLR

Categorias	Desafios
Legislação e normatização	<ul style="list-style-type: none"> • Necessidade de revisão da legislação em relação às questões de gerenciamento dos resíduos perigosos que estão no SLR. • Necessidade ou não de termo de doação no caso da transferência de resíduos eletroeletrônicos. • Custeio de produtos órfãos. • Complementação de marco regulatório com o objetivo de não favorecer o desenvolvimento de negócios sem o atendimento aos requisitos legais e técnicos relacionados aos SLR. • Estabelecimento de metas para a utilização de matéria-prima secundária.
Aspectos tributários	<ul style="list-style-type: none"> • Definição de critérios para a isenção de tributação da atividade de LR. • Promoção do mercado da reciclagem por meio de incentivos fiscais ao uso de material reciclado, à venda de produtos com conteúdo reciclável ou com <i>design</i> ecológico.
Instrumentos financeiros	<ul style="list-style-type: none"> • Linhas de crédito para investimentos na infraestrutura das recicladoras regionais. • Criar incentivos no mercado para vários atores envolvidos nos SLR.
Controle governamental	<ul style="list-style-type: none"> • Promover a articulação entre fabricantes, importadores, comércio, recicladores e poder público para o alinhamento dos objetivos do SLR, com a criação de instrumentos de controle que garantam a vinculação de todos aos seus respectivos SLR. • Regulamentação dos acordos setoriais, possibilitando a fiscalização dos procedimentos estabelecidos.

Fonte: (COUTO; LANGE, 2017)

A Logística Reversa num ambiente de alta competitividade pode ser um recurso de geração de valor para a empresa. No entanto as iniciativas são de forma isolada e não integradas à estratégia da empresa e precisam satisfazer inúmeros agentes: acionistas, funcionários, clientes, governo, comunidade local e outros *stakeholders*. Para atender estes diversos agentes é necessário definir indicadores de desempenho financeiros e não financeiros. Sendo assim o *Balanced Scorecard* (BSC) com as diferentes perspectivas (financeira, clientes, processos internos e aprendizado e crescimento) torna-se uma excelente escolha (HERNÁNDEZ; MARINS; CASTRO, 2012).

Os autores relacionam os indicadores de desempenho para medir os resultados da LR e os classificam por prioridade conforme sua pesquisa, ver tabela 1.

Tabela 1 – Prioridades dos indicadores de LR.

Indicadores	Prioridades
Custos de operação (CO)	30,134%
Recaptação de valor (RP)	20,681%
Inovação tecnológica (IT)	13,276%
Incentivo à reciclagem (IR)	8,988%
Criação de emprego (CE)	8,384%
Ações sociais e ambientais (AS)	7,806%
Cumprimento da legislação (CL)	4,188%
Serviço diferenciado (SD)	3,483%
Relações duradouras com clientes (RC)	3,061%

Fonte: (HERNÁNDEZ; MARINS; CASTRO, 2012)

3.1 PONTOS DE ENTREGA VOLUNTÁRIA – PEV

Os Pontos de Entrega Voluntária (PEV) são locais estrategicamente definidos de fácil acesso e com grande fluxo de pessoas, instalados em locais privados ou públicos para destinação de materiais pós-consumo. Cabe salientar que os PEVs são locais para recebimento de embalagens em geral, como resultado das atividades de consumo e não destinados ao recebimento de produtos.

A operacionalização pode ser considerada de responsabilidade dos fabricantes e importadores de produtos comercializados em embalagens. Os Distribuidores e Comerciantes devem ceder espaço para implantação de PEVs, celebrando contratos com os fabricantes/importadores. Os consumidores devem ser conscientizados a fazer a devolução das embalagens nos PEVs (MMA, 2015).

Os materiais recebidos nos PEV têm como destino principal as cooperativas que farão a triagem adequada desses resíduos. Estas cooperativas devem contabilizar as quantidades recolhidas para então alimentar um sistema de contabilização de quantidades recebidas e triadas, para geração dos dados a serem enviados ao Ministério do Meio Ambiente (MMA, 2015).

3.2 TERMOS E ACORDOS SETORIAIS

O Sistema Nacional de Informações sobre a Gestão dos Resíduos Sólidos (SINIR) é o órgão que atuará sob a coordenação e articulação do MMA e deverá

coletar e sistematizar dados relativos aos serviços públicos e privados de gestão e gerenciamento de resíduos sólidos, possibilitando:

- a) o monitoramento, a fiscalização e a avaliação da eficiência da gestão e gerenciamento dos resíduos sólidos inclusive dos sistemas de logística reversa;
- b) a avaliação dos resultados, impactos e acompanhamento das metas definidas nos planos, e a informação à sociedade sobre as atividades da Política Nacional;

A definição do Sistema de Informações a ser adotada será o resultado da análise das demandas de informações definidas na Lei nº 12.305/10 e seu Decreto Regulamentador nº 7.404/10, e de dados e informações coletados dos sistemas afins, a partir das necessidades de informação da União, dos Estados, dos Municípios e do Distrito Federal.

Quadro 2 – Situação da implantação dos sistemas de LR

Cadeias	Status atual
Embalagens Plásticas de Óleos Lubrificantes.	Acordo setorial assinado em 19/12/2012 e publicado em 07/02/2013
Lâmpadas Fluorescentes de Vapor de Sódio e Mercúrio e de Luz Mista.	Acordo setorial assinado em 25/11/2015. Publicado em 27/11/2015.
Embalagens em Geral.	Acordo setorial assinado em 25/11/2015. Publicado em 27/11/2015.
Embalagens de Aço.	Termo de compromisso assinado em 21/12/2018. Publicado em 27/12/2018.
Produtos Eletroeletrônicos e seus Componentes.	Dez propostas de Acordo setorial recebidas até junho de 2013, sendo 4 consideradas válidas para negociação. Proposta unificada recebida em janeiro de 2014. Em negociação.
Medicamentos.	Negociações encerradas. Proposta de Decreto elaborada. Consulta Pública realizada. Próxima etapa - Análise das contribuições recebidas na Consulta Pública e elaboração da minuta final do Decreto

Fonte: <http://www.sinir.gov.br/logistica-reversa>

Para que os distintos sistemas e serviços converseem, se integrem e se articulem, deverão adotar obrigatoriamente os Padrões de Interoperabilidade de Governo Eletrônico (e-PING) (SINIR, 2018).

Diversos setores possuem acordos relativos a LR, um posicionamento sobre a situação atual de cada um destes setores é listado no quadro 2.

Nos próximos itens serão abordados o acordo setorial das embalagens em geral e o termo das embalagens de aço.

3.3 ACORDO SETORIAL PARA IMPLANTAÇÃO DO SISTEMA DE LOGÍSTICA REVERSA DE EMBALAGENS EM GERAL

O Acordo Setorial para Implantação do Sistema de Logística Reversa de Embalagens em Geral foi assinado no dia 25/11/2015 e tem como objetivo garantir a destinação final ambientalmente adequada das embalagens. O objetivo deste acordo é a implantação do sistema de LR para as embalagens compostas de papel e papelão, plástico, alumínio, aço, vidro, ou ainda pela combinação destes materiais, como as embalagens cartonadas longa vida. A figura 29 mostra o fluxo logístico proposto para a implantação neste acordo. Foi assinado por 20 entidades representando mais de 1000 empresas e a UNIÃO por intermédio do MMA, denominado Coalizão. As entidades intervenientes anuentes deste acordo são: Compromisso Empresarial para Reciclagem – CEMPRE, Associação Brasileira de Embalagem – ABRE, Associação Nacional dos Aparistas de Papel – ANAP e Instituto Nacional das Empresas de Preparação de Sucata não Ferrosa e de Ferro e Aço – INESFA (MMA, 2015).

Para a implantação o acordo prevê no âmbito da responsabilidade compartilhada as seguintes etapas: Separação, Descarte, Transporte, Triagem, Classificação e Destinação.

A avaliação e monitoramento se dará por meio de relatórios anuais de desempenho entregues ao MMA e disponibilizados ao público em formato eletrônico. Este deve conter informações de todo sistema, porém para nosso estudo, vale citar os dados de pesos e tipo de embalagens, local de coleta e destinação, estimativa de embalagens colocadas no mercado, e dados da distribuição dos PVEs (MMA, 2015).

Figura 29 – Fluxo Logístico Proposto pela Coalizão



Fonte: Anexo III do Acordo Setorial LR Embalagens Geral

3.4 TERMO DE COMPROMISSO PARA IMPLANTAÇÃO DO SISTEMA DE LOGÍSTICA REVERSA DE EMBALAGENS DE AÇO

O Termo de Compromisso para Implantação do Sistema de LR de Embalagens de Aço foi assinado em 21/12/2018 e teve seu extrato publicado no D.O.U de 27/12/2018.

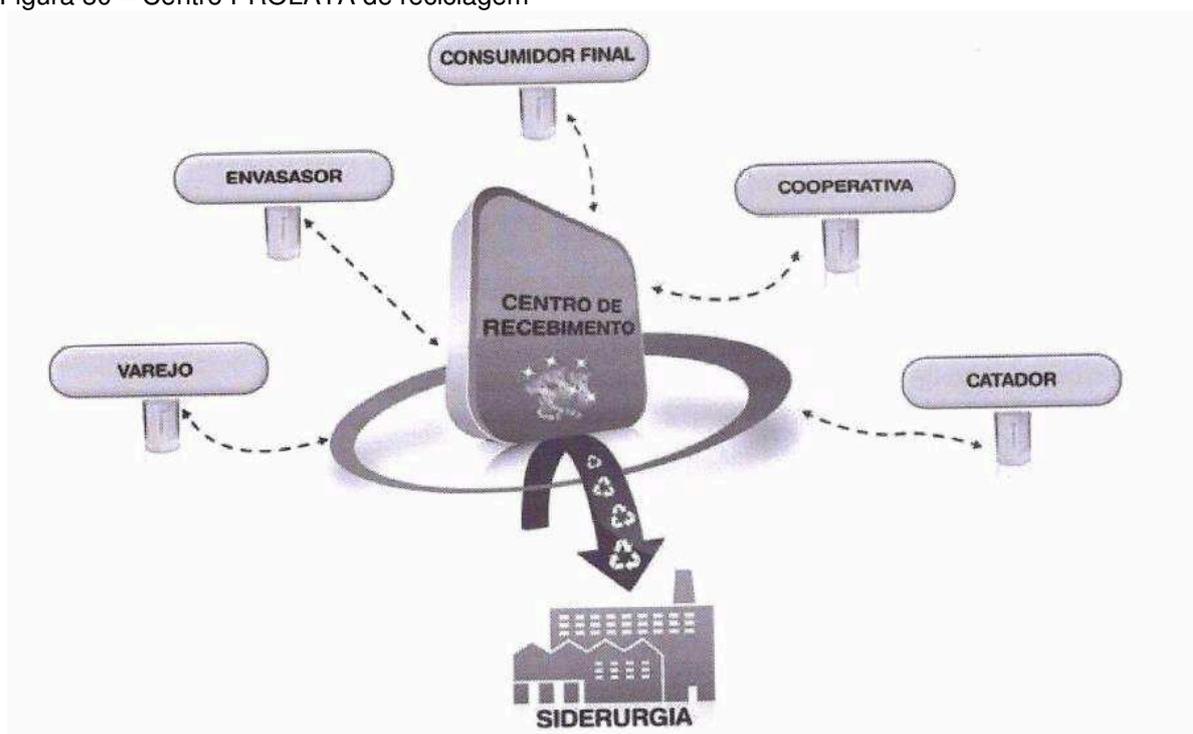
O acordo foi assinado por 4 entidades, PROLATA, ABRAFATI, ANAMACO e ABEAÇO, juntamente com a UNIÃO por intermédio do MMA. A representatividade de associados nas entidades participantes é respectivamente de 32, 34, 73 e 18 empresas associadas.

O objetivo deste acordo é de elevar os índices de reciclagem de latas de aço, reduzir os volumes nos aterros e maximizar o uso de recursos, estimulando o descarte seletivo e coleta das embalagens de aço, visto que possuem valor comercial. O programa PROLATA, investirá na capacitação das cooperativas com o objetivo de

fomentar a coleta e triagem das latas de aço, e fornecerá dados de coleta, descarte e reciclagem das latas de aço no Brasil (MMA, 2018).

O centro PROLATA de reciclagem, ver figura 30, tem como principais atores: Centro de recebimento, Siderurgia, Varejo, Envazador, Consumidor final, Cooperativa e Catador.

Figura 30 – Centro PROLATA de reciclagem



Fonte: (MMA, 2018)

4 ESTADO DA ARTE

A pesquisa foi realizada em bases de dados de produções acadêmicas do Instituto de Engenheiros Eletricistas e Eletrônicos (IEEE), *Scopus* da *Elsevier* e *Web of Science* da Thomson Reuters (WoS) foi empregado as palavras chaves para a consulta: (“*Blockchain*” AND (“*Logistic*” OR “*Reverse*”)). Como resultado foram obtidos 45 artigos, sendo 4 no IEEE, 31 na *Scopus* e 10 na WoS. Dentre os artigos, alguns estavam duplicados, após a limpeza resultaram 33 artigos, mas nem todos os textos completos foram obtidos. Após leitura de todos os resumos, identificou-se que em nenhum foi encontrado o tema principal da pesquisa, *Blockchain* com Logística Reversa (LR). Porém foram identificados 11 artigos com assunto relacionado a tecnologia *Blockchain* aplicada a Logística, assim estes foram separados para leitura integral e preparação deste capítulo.

Outra pesquisa foi realizada onde procurou-se identificar as publicações utilizando as palavras chaves separadamente “*blockchain*” e “*logistic reverse*” nas mesmas bases de dados IEEE, WoS e *Scopus*. A busca foi realizada em 2018 e como critério de seleção dos artigos foi considerado o número de citações. Sendo assim a busca com a *query* “*blockchain*” foram separados os 3 artigos mais citados e que se repetiram nas bases utilizadas, conforme resumo apresentado na tabela 2. A pesquisa com a *query* “*logistic AND reverse*” não obteve resultado significativo na base de dados IEEE, porém nas bases WoS e *Scopus* foram encontrados e selecionados conforme mostra a tabela 3.

Com base nestas pesquisas o capítulo estado da arte foi desenvolvido.

Lanko, Vatin e Kaklauskas (2018) apresentam em seu artigo as tecnologias de Identificação por rádio frequência (RFID), *Quick Response Code* (QRCode) e *Blockchain* trabalhando juntas para otimizar as rotinas do mercado de logística em geral e na prestação de serviços do setor de construção em particular, tendo como atores o produtor de cimento, pedreiras para extração de areia e cascalho fabricantes de diversos aditivos. Os sistemas centralizados não garantem o fornecimento de informações completas ou precisas para os participantes da cadeia de suprimentos, e muitos dos riscos no setor de logística vêm da falta de confiabilidade destas informações. O artigo trata da implementação de tecnologias de contrato inteligente e

Blockchain como forma de substituir os serviços centralizados por tecnologias descentralizados que garantirão um nível adequado de transparência e credibilidade.

Tabela 2 – Número de citações dos artigos com “blockchain” nas bases WoS, Scopus e IEEE

Título	WoS	Scopus	IEEE
<i>Blockchains and Smart Contracts for the Internet of Things</i>	105	200	180
<i>Decentralizing Privacy: Using Blockchain to Protect Personal Data</i>	74	162	90
<i>Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts</i>	59	126	92

Fonte: Autor

Vantagens destacadas são a redução de erros devido a menor interação humana, por falhas ou intencionais. A monitoração durante a entrega, registrando data, hora e localização por GPS em tempo real. Possibilidade de registro de todas as movimentações e de produtos utilizados permitindo rastreabilidade e controle de qualidade do produto final. A utilização das tecnologias RFID ou QRCode se justifica para a identificação, sua aplicação possui custos bem diferentes e limitações também por parte do QRCode, no caso a necessidade de proximidade e passividade. O *Blockchain* tem como função ser o agente gerador da credibilidade, devido suas características de ser um banco de dados distribuído, aberto e transparente para todos os usuários da rede, protegido de forma confiável, onde os registros são sequencialmente vinculados e armazenados em cópias idênticas por pares, e com criptografia.

Tabela 3 – Número de citações dos artigos com “logistic AND reverse” nas bases WoS e Scopus

Título	WoS	Scopus
<i>Green supply-chain management: A state-of-the-art literature review</i>	1090	1492
<i>Quantitative models for reverse logistics: A review</i>	893	1145
<i>Facility location and supply chain management - A review</i>	665	857

Fonte: Autor

A utilização da tecnologia *Blockchain* para integrar e manter atualizadas as informações entre os sistemas de *Enterprise Resource Planning* (ERP), sistema de gestão internos para otimizar seus processos e integrar todos os setores dentro da empresa. Algumas empresas possuem seus ERPs desenvolvidos internamente, e outras adquirem produtos de mercado, de fornecedores distintos. Para integrar

operações e processos entre empresas, utiliza-se geralmente troca de arquivos, *Extensible Markup Language* (XML) ou *Enterprise Data Integration* (EDI). A ideia central do artigo é gerar uma forma mais dinâmica utilizando *Blockchain* para manter uma rede integrada e apoiada nas características da tecnologia *Blockchain* pública ou privada, mantendo a padronização, transparência, objetividade e neutralidade nas transações (BANERJEE, 2018).

Para exemplificar, foi utilizado a cadeia de suprimentos, onde diversas empresas são envolvidas, todas com suas bases centralizadas e independentes. Assim muitas das informações serão alimentadas em seus ERPs, fazendo com que a mesma informação seja gerada na base de cada empresa participante. Então a empresa A irá comercializar seu produto à empresa B e C. Ambas necessitam registrar os dados do fornecedor e do produto, sendo que sem uma integração o trabalho será manual e passivo de inúmeros erros. No caso da utilização de transferências de arquivos, geralmente os dados são incluídos de forma automática, mas as manutenções futuras serão manuais e passíveis de não receberem atualizações. A proposta do artigo é de manter uma rede *Blockchain* onde cada participante é responsável por manter seus dados e de seus produtos, e todos participantes da cadeia se tornam um “nó” da rede, assim quando uma informação for adicionada ou atualizada no *Blockchain*, automaticamente todos os “nós” recebem esta cópia atualizada. A aplicação da tecnologia *Blockchain* traz segurança, transparência, eficiência e redução de custos a todos participantes, e isso está intrínseco na tecnologia. Além dos benefícios da reutilização dos dados, os contratos inteligentes podem ser outro recurso a trazer ganhos para as corporações, pois os contratos regem as regras da negociação, tornando todo processo transparente e auditável, evitando inúmeras disputas futuras.

Outro exemplo com muitos benefícios citado no artigo, é a cadeia de suprimentos de produção OEM. Esta possui potencial muito forte para obtenção de ganhos com a aplicação de uma rede *Blockchain*, como na gestão por contratos inteligentes, a auditoria de produtos e processos. No processo da logística, muitas soluções utilizando *Blockchain* foram desenvolvidas. Banerjee (2018) relata que os principais ganhos são a transparência e agilidade na tomada de decisão, através da obtenção de dados em tempo real de monitoramento da carga e notificação de

sinistros e desastres naturais. Embora estes fatores sejam importantes, a redução de papel também é um ponto forte deste, ajudando nas iniciativas verdes para proteger o planeta. Uma aplicação que tem explorado o uso do *Blockchain* é a identificação da procedência de produtos. Grandes marcas têm necessidade de comprovar de onde seus produtos estão sendo produzidos, pois os consumidores estão exigindo não somente produtos de qualidade mas com consciência verde e humanitária, assim as aplicações com *Blockchain* são utilizadas para trazer a transparência exigida pelos clientes, com todo o rastreamento dos materiais utilizados, como foram produzidos, armazenados e transportados até o ponto de venda.

No artigo de Gromovs e Lammi (2017), os autores focam como o programa de estudo das universidades está preparado para atender à necessidade futura da área de logística, pensando na aprendizagem dos futuros profissionais. As tecnologias disruptivas, Internet das Coisas (IoT), *Blockchain* e impressão 3D são consideradas relevantes e seu estudo necessita ser aprofundado para os professores e alunos. A IoT com máquinas e sensores conectados via internet são o conjunto de instrumentos. O *Blockchain* como suporte informativo, tornando-se a espinha dorsal para um novo tipo de internet, registrando as transações em um banco de dados distribuído e incorruptível em uma rede P2P que integrará todos os parceiros da cadeia de suprimentos, elevando-os a um grau mais alto de responsabilidade. A impressão 3D na logística pode ser considerada aliada no processo, podendo facilitar a descentralização da produção, reduzindo os prazos de fabricação, reduzindo o nível de armazenamento de peças sobressalentes e aumentando o papel do frete de última milha. Outra tecnologia discutida, são os veículos não tripuláveis, dos pequenos *drones* a caminhões, navios, locomotivas e guindastes também não tripulados e autônomos.

Diante deste fenômeno, em todo o mundo tem surgido consórcios de empresas, universidades e organizações para desenvolver aplicações com estas tecnologias para as áreas da logística, transporte e cadeia de suprimentos. Nos países da região do Mar Báltico Central (Finlândia, Suécia, Estônia e Letônia) foram comparados os programas de ensino de 4 universidades com cursos de bacharel em logística. O resultado do diagnóstico foi a necessidade de renovação de seus

programas em conjunto com outras ações para fomentar o aprendizado e ampliar as competências dos futuros alunos em conexão com as tecnologias IoT e *Blockchain*.

Em sintonia, Christidis e Devetsikiotis (2016) discutem sua aplicação integrada do *Blockchain* com IoT, apresentando alguns casos de uso e proposições para a junção das duas tecnologias. Os autores ponderam que a tecnologia *Blockchain* por si só se aplica aos casos, porém com a utilização de dispositivos IoT o processo torna-se totalmente ou parcialmente automatizado.

Um exemplo de aplicação bem diferente do que se tem encontrado costumeiramente ao uso da tecnologia *Blockchain* é apresentado neste artigo. Isso mostra que inicialmente deve-se compreender e identificar um problema, então verificar a viabilidade da aplicação da tecnologia para solucionar o problema, e não procurar um problema para a aplicação da tecnologia em si. Então foi criado um ambiente hipotético pensando nas atualizações de *firmware*¹¹ de dispositivos IoT, distribuídos aos milhares, e avaliado os custos que o fornecedor teria para manter este ambiente por muitos anos.

Todos os dispositivos IoT de um fabricante operando na mesma rede *Blockchain*, cujo fabricante tenha implementado um contrato inteligente que permite armazenar o *hash* da última atualização do *firmware* na rede, por exemplo via um sistema de arquivos *peer-to-peer* distribuído, como o *InterPlanetary File System* (IPFS) (RIBEIRO, 2018). Os dispositivos são enviados com o endereço do contrato inteligente incluído em seu cliente de *Blockchain* ou por meio de um serviço de descoberta onde encontrariam este endereço. Os primeiros pedidos serão atendidos pelo próprio nó do fabricante, mas depois que o binário se propaga-se para outros nós participantes, o nó do fabricante pode ser desativado. Assim quando um dispositivo que ingresse na rede muito depois do fabricante ter parado de participar da rede, este ainda pode recuperar a atualização de *firmware* desejada e ter certeza de que é o arquivo correto. Tudo isso acontece automaticamente, sem interação do usuário. Compare e contraste com o cenário centralizado onde o dispositivo busca pelo servidor do fabricante desativado para uma atualização e obtém um erro 404 (página não encontrada).

¹¹ *Firmware*, é um conjunto de instruções operacionais que são programadas diretamente no *hardware* de equipamentos eletrônicos

Indo além, este ambiente pode proporcionar a criação de um novo mercado para serviços entre dispositivos. No exemplo acima, os dispositivos que armazenam uma cópia do binário podem cobrar pelo serviço com objetivo de obtenção de lucro ou somente para manutenção de seus custos.

Os autores relatam algumas dificuldades para implantação segura de contratos inteligentes. Eles ressaltam que a combinação do *Blockchain* e IoT possuem promissoras aplicações onde a automatização de processos tende a otimizar os fluxos complexos e demorados, reduzindo tempo e custo, trazendo inúmeros benefícios aos participantes desta rede.

Com maior ênfase em privacidade de dados, Zyskind, Nathan e Pentland (2015) abordam soluções neste tema. De maneira muito clara apresenta cenário para compreensão e detalha os protocolos de maneira muito didática.

Ao passo que grandes corporações agregam milhares de usuários em suas plataformas, grandes quantidades de informações pessoais e sensíveis são mantidas em suas bases de dados e sua utilização não é de conhecimento dos usuários. Pois os indivíduos possuem pouco ou nenhum controle sobre seus dados armazenados por estas empresas. Então os autores propõem uma plataforma para que os reais proprietários dos dados possam deliberar sobre quais serviços podem ter acesso a suas informações, podendo ainda alterar ou mesmo excluir as permissões anteriormente atribuídas. A solução proposta inicialmente foi projetada para atender a fatia de aplicações para dispositivos móveis. As entidades envolvidas são: usuários, serviços e “nós”. Usuários são os proprietários das informações, serviços são as aplicações baixadas e instaladas nos celulares e “nós” são os responsáveis por manter o *Blockchain*.

A plataforma funciona de forma a armazenar dois tipos de transações, de dados (armazenamento e recuperação) e acessos (controle de permissões). Para cada nova combinação (usuário, serviço) acarretará numa nova entrada no *Blockchain*. O *Blockchain* armazenará somente este conjunto de chaves e um *hash* dos dados do indivíduo ou as permissões atribuídas para este serviço aos dados do indivíduo, sendo que as informações são armazenadas em um *off-blockchain*, ou seja um *Distributed HashTable* (DHT) com proposito de segurança, performance, escalabilidade e redução dos custos (CREIMER, 2019).

Como os dados armazenados no DHT, são criptografados com as chaves públicas do usuário e do serviço, qualquer um que obtenha acesso indevido as informações não conseguirão decifrar as informações. Assim somente estes poderão ter acesso às informações, sendo que o usuário terá acesso irrestrito e o serviço somente o que foi fornecido de permissão pelo usuário no momento dos registros do serviço. É importante observar que estas permissões podem ser alteradas ou mesmo excluída, isso porque o usuário passa a ter controle dos dados e do acesso a estes dados.

Seguindo a questão de privacidade, Kosbo et al. (2016) aborda a questão da privacidade nas redes *Blockchain* e no uso dos contratos inteligentes. O autor registra que os contratos inteligentes e as transações em uma rede *Blockchain* pública são publicamente visíveis. Embora as informações salvas no *Blockchain* serem identificadas pelas chaves públicas dos usuários e não pelos seus nomes, assim é possível utilizar mais de uma chave para tentar driblar as associações das transações ao indivíduo.

O trabalho de Ron e Shamir (2013) descarregou todos os registros da uma rede pública do *Bitcoin*. Realizando análises das transações, conseguiu-se identificar o comportamento típico dos usuários, como eles adquirem e como gastam seus *Bitcoins*, o saldo que mantêm em suas contas e como eles movimentam as criptomoedas entre seus clientes. Identificou-se também a utilização de diferentes contas para tentar proteger melhor sua privacidade.

Com base nisso foi proposto o *Hawk*, uma plataforma genérica onde programadores não especialistas possam desenvolver facilmente contratos inteligentes que preservam a privacidade. O compilador *Hawk* é responsável por compilar o programa para um protocolo criptográfico entre o *Blockchain* e os usuários. Como pode-se observar na modelagem do *Hawk*, a execução de contratos *Hawk* é facilitada por uma parte especial chamada “gerente”, um novo ator introduzido ao processo pelo *Hawk*. O gerente pode ver as entradas dos usuários e considerado confiável, se não divulgar os dados privados dos usuários.

Um programa *Hawk* é composto por duas partes: parte pública e parte privada. O público só terá acesso aos dados e ao contrato se os usuários assim desejarem, divulgando suas chaves públicas. Após compilação é gerado dois programa, um para

ser executado pelo usuário e um programa a ser executado pelo gerente (KOSBA et al., 2016).

As garantias de segurança do *Hawk* abrangem dois aspectos, privacidade na cadeia e segurança contratual. A privacidade na cadeia estipula que a privacidade transacional seja fornecida contra qualquer parte não envolvida no contrato, isso é conseguido enviando informações “criptografadas” para o *Blockchain*. A segurança contratual protege as partes envolvidas no mesmo contrato, engloba não apenas noções criptográficas de confidencialidade e autenticidade, mas também imparcialidade financeira na presença de comportamento de trapaça e abortamento. O gerente não deve ser equiparado a um terceiro confiável. Também não precisa ser confiável para manter a segurança ou a privacidade.

O trabalho foi completado com a implementação e avaliação de um protótipo de *Hawk*, onde obteve-se desempenho 10 vezes melhor que uma implementação direta. O autor lista alguns dos desenvolvimentos em protótipos: “leilão de lances fechados”, “rock, jogo de papel, tesoura”, “aplicativo de *crowdfunding*” e um “instrumento financeiro de *swap*”. Observou-se que o cálculo criptográfico do gerente (a parte mais cara do protocolo) está abaixo de 2,85 min usando 4 núcleos, traduzindo-se em menos de US\$ 0,14 do tempo da máquina virtual na *Amazon, Amazon Elastic Compute Cloud (EC2)* (AMAZON, 2018). Além disso, todo o cálculo *on-chain* (realizado por todos os mineiros) está abaixo de 20ms para todos os casos. Os autores têm previsão de abrir o *framework Hawk* no futuro.

Sobre *Green supply-chain management (GrSCM)*, Srivastava (2007) define que seu escopo varia de compra ecológica a cadeias de suprimento verdes integradas que vão de fornecedor a fabricante e cliente, e até Logística Reversa. O autor faz uma ampla revisão na literatura a partir de 1990, com cerca de 1500 livros, artigos de revistas e volumes editados, sendo que em seus filtros restaram 227 referências, que discorre sobre *Design Verde*, operações verdes e LR. Um dos resultados do trabalho foi gerar uma classificação, sendo três itens relacionados como principais, são eles: Importância do GrSCM, *Design Verde* e Operações Verdes.

Sobre importância do GrSCM, destaca-se três abordagens. Na abordagem “reativa”, as empresas comprometem-se com a gestão ambiental, começam a rotular os produtos que são recicláveis e usam iniciativas para reduzir o impacto ambiental

da produção. Na abordagem “proativa”, eles começam a antecipar-se as leis ambientais, realizando um compromisso de recurso modesto para iniciar a reciclagem de produtos e projetar produtos verdes. Na abordagem de “busca de valor”, as empresas integram atividades ambientais como compras ecológicas e implementação da ISO como iniciativas da sua estratégia de negócios.

O *Design Verde*, enfatiza a avaliação do ciclo de vida do produto (ACV), a criação de produtos ambientalmente conscientes, avaliação de todas as fases da vida de um produto, ou seja, extração e processamento das matérias-primas, produção, transporte e distribuição, uso, remanufatura e reciclagem. O escopo da ACV envolve o rastreamento de todos os fluxos de materiais e energia de um produto, desde a recuperação de suas matérias-primas do ambiente até o descarte do produto de volta ao ambiente.

Nas Operações Verdes, alguns dos principais desafios do GrSCM são: integração da remanufatura com operações internas, integração do design de produto, retorno de produto e incentivos da cadeia de suprimentos, integração da remanufatura e a LR com o design da cadeia de suprimentos.

O descarte sempre foi um problema convincente e levou à consciência verde. No caso do GrSCM, os esforços para minimizar o descarte têm sido o foco.

A reutilização de produtos e materiais não é um fenômeno novo. Fabricação e remanufatura verde, é uma área muito importante dentro das operações verdes. A desmontagem é um método sistemático de separar um produto em suas partes constituintes, componentes, subconjuntos ou outros agrupamentos. A qualidade dos produtos reparados é geralmente inferior à qualidade dos novos, o objetivo da renovação é levar produtos usados até uma qualidade específica.

A reciclagem impulsionada principalmente por fatores econômicos e regulatórios, é realizada para recuperar o conteúdo material de produtos usados e não funcionais. Porém a logística representa até 95% dos custos totais em reciclagem. Redesenhar redes logísticas para acomodar retornos de produtos e remanufatura e reutilização de tais peças e componentes pode muitas vezes ser lucrativo e está assumindo maior importância nos negócios. As empresas precisam perceber o valor oculto na LR e começar a se concentrar nessa área. É preciso entender o impacto financeiro das estratégias de LR.

Atualmente, as tecnologias de informação e comunicação (TICs) provavelmente desempenharão um papel fundamental na coordenação e integração das atividades do GrSCM.

Fleischmann et al. (1997) também afirma que a reutilização de produtos e materiais não é um fenômeno e também reforça que a recuperação dos produtos usados é economicamente mais atraente do que o descarte em muitas categorias de produtos. Apresenta que 1994 em toda Europa, principalmente na Alemanha e Holanda, houve um crescimento expressivo na reciclagem, reutilização e recuperação de vidros, metais e resíduos industriais, comparados com períodos anteriores.

A logística reversa engloba as atividades de logística desde os produtos usados que não são mais exigidos pelo usuário até os produtos serem novamente utilizáveis em novos produtos e postos no mercado.

Aspectos que motivaram a crescente onda de recuperação nos anos 90, além dos países aplicarem a legislação ambiental, teve também a cobranças dos clientes que aderiram a uma visão mais verde, assim fazendo pressão sobre os fornecedores por ações de cunho ambiental. O fator econômico também influenciou alguns fornecedores a aderirem a onda verde.

Este movimento de retorno de produtos e embalagens, gerou um problema de gestão de estoque, ou por excesso de produtos retornados ou por falta destes. Para combinar vantagens ecológicas e econômicas, como sugerido pelo conceito de economia "sustentável", diferentes formas de reutilização são aplicadas: reutilização direta, reparo, reciclagem e remanufatura. A reutilização direta faz uso de itens reutilizáveis como: garrafas, paletes ou contêineres, através de limpeza ou pequena manutenção. O reparo é a restauração de produtos com falha, embora possivelmente com uma perda de qualidade. A reciclagem é a recuperação de material sem conservar qualquer estrutura de produto original, como ocorre com metal, vidro e papel. A remanufatura procura manter a identidade do produto, busca trazer o produto de volta a uma condição "nova", através da desmontagem, revisão e substituição. Este processo envolve vários atores com funções distintas, incluindo: coleta, teste e reprocessamento. A reutilização pode ser realizada pelo produtor original ou por um terceiro (FLEISCHMANN et al., 1997).

Uma atenção especial deve ser dada ao projeto da rede de distribuição reversa. Esta possui uma estrutura de “muitos para poucos”, e possui uma incerteza considerável, tanto na oferta de produtos usados pelos clientes quanto nos mercados para recolocação de produtos recuperados. As possíveis funções no canal de distribuição reversa são: coleta, teste, classificação, transporte e processamento. Uma rede de distribuição deve ser projetada, determinando locais adequados para essas funções. Quanto ao sistema aplicado a reciclagem pode muitas vezes ser descrita como um sistema de malha aberta, ou seja, os produtos não retornam ao produtor original, mas serão usados em outras indústrias. Neste formato a integração de distribuição direta e reversa são escassas, pois os atores diferem em ambos os canais. A remanufatura e a reutilização costumam usar um sistema de circuito fechado, ou seja, o produto ou embalagem retornam e são reutilizados pelo produtor original. A distribuição inversa pode ocorrer diretamente através da rede original, intermediários ou através de provedores logísticos especializados.

Neste trabalho foi abordado sobre a gestão de estoque e inventário, porém muitas questões de incerteza tornam esta gestão muito particular de cada processo. Porém é consenso que os avanços da tecnologia da informação irão contribuir e muito nesta gestão e auxiliar no processo como um todo. Isso ocorreu também no estudo do planejamento da produção com reutilização de peças e materiais. Vários autores propõem métodos e modelos matemáticos, porém há incertezas na utilização destes sistemas. Assim as características econômicas para o processo são particularmente importantes e integram as áreas de design para a criação de produtos pensados para sua reutilização. Novamente a tecnologia da informação é citada no contexto de que o planejamento poderá fornecer meios para reduzir estas incertezas.

A LR por imposição da legislação ou pressão dos clientes ou pelo apelo Verde do marketing, cada vez mais se consolida. Existem inúmeros desafios ainda por serem transpostos, mas com o avanço dos estudos e da tecnologia da informação, irá suportar e tornar este processo cada vez mais presente nas indústrias de manufatura.

Melo, Nickel e Saldanha-da-Gama (2009), abordam o assunto da localização para uma melhor definição do *Supply-Chain Management* (SCM).

O SCM é o processo de planejar, implementar e controlar as operações da cadeia de suprimentos de maneira eficiente. O SCM abrange todos os movimentos e

armazenamento de matérias-primas, estoque em processo e produtos acabados desde o ponto de origem até o ponto de consumo.

No planejamento estratégico de um SCM, a localização das instalações tem lugar de destaque. Muitas variáveis e metodologias são utilizadas para definir a localização e número de instalações para atender melhor a demanda dos clientes, bem como minimizar as distâncias e reduzir custos fixos.

Um aspecto crucial de muitos problemas de localização diz respeito à existência de diferentes tipos de instalações, cada uma desempenhando um papel específico (por exemplo, produção, armazenamento) e um fluxo material natural (isto é, uma hierarquia) entre elas. Problemas de localização multi-período têm sido propostos para abordar situações em que os parâmetros mudam ao longo do tempo de forma pré-ditada e motivados pela incerteza que muitas vezes pode estar associada a alguns dos parâmetros, como as demandas e custos futuros dos clientes.

Normalmente a definição da localização não é uma decisão dos níveis táticos ou operacionais, porém muitos dos problemas estão fortemente relacionados a eles, como políticas de controle de estoque, escolha de modos e capacidades de transporte, layout e gerenciamento de armazém e roteamento de veículos, entre outros.

Este artigo também cita considerações sobre a logística reversa, referente as atividades dedicadas à coleta e recuperação de produtos dentro do SCM. Três aspectos podem ser mencionados para justificar atividades reversíveis: aspectos econômicos, diretivas governamentais e pressão do consumidor. As atividades de logística reversa geralmente são apoiadas por instalações específicas que podem ser de dois tipos principais diferentes: centros de coleta e instalações de recuperação de produtos devolvidos. Nesse contexto da estrutura do SCM, a rede precisa ser estendida com elos de transporte para fluxos de retorno, dos locais próximos aos clientes para locais onde ocorrem atividades de reparo, remanufatura e / ou reciclagem.

As tecnologias da informação avançaram muito e sua aplicação propiciou o empoderamento dos clientes, apresentando mudanças no seu comportamento e suas preferências, tornando os ciclos de vida de produtos mais curtos e tamanhos de lotes menores. Esses aspectos contribuíram para a crescente incerteza da demanda e,

como resultado, uma rede de cadeia de suprimentos robusta e bem projetada tornou-se ainda mais importante. Conseqüentemente, para a nova configuração do SCM modelos sofisticados de localização de instalações podem ser necessários.

5 PROCEDIMENTOS METODOLÓGICOS

A modelagem do SLR proposto baseia-se no ciclo de vida dos produtos das embalagens de aço, limitado ao mercado das indústrias de tintas. Este setor apresentou seu termo de acordo setorial para o governo somente em dezembro de 2018. Para incentivar as empresas e consumidores a aderir ao SLR este trabalho propõe a criação de um criptoAtivo através da tecnologia DLT. Além de integrar todas as empresas participantes a tecnologia *Blockchain* fornece segurança e transparência suficiente para que empresa concorrentes possam participar do mesmo SLR. A adesão trará benefícios as empresas de redução de custos obtida pelo compartilhamento dos recursos para a implantação do sistema e fiquem em *compliance* com a legislação vigente referente ao PNRS. O criptoAtivo deve circular livremente entre as empresas participantes indiferentemente da marca da embalagem devolvida, ou qual lojista recebeu a devolução da embalagem, ou se o crédito foi concedido para uma embalagem devolvida de uma indústria concorrente, o criptoAtivo pode ser utilizado em qualquer das empresas participantes.

A construção da PoC ou protótipo aplicará a metodologia *Design Science Research* (DSR). O método consiste em aplicar a ciência de design para a criação de artefatos com objetivo de resolver problemas do mundo real, apoiando a pesquisa e inovação. A ciência da computação incorporou a criação destes artefatos há muito tempo com o intuito de estudar e avaliar o comportamento de fenômenos organizacionais e humanos aplicados a sistemas de informação através da união da ciência do design e ciências naturais (HEVNER; CHATTERJEE, 2010).

Para o desenvolvimento da PoC com base no método DSR, a criação do artefato baseou-se na construção dos contratos inteligentes utilizando a plataforma *Hyperledger Composer* e a avaliação utilizou-se dos conceitos básicos para testes de softwares, com a técnica do Teste Funcional, que consiste em gerar entradas de dados e executar o software com estas entradas. O resultado obtido deve ser comparado com o resultado esperado, onde o teste será considerado bem sucedido se o resultado obtido foi igual ao esperado (NETO, 2007).

Para a aplicação do DSR e validação da PoC, serão seguidas as diretrizes para a pesquisa proposta por (HEVNER; CHATTERJEE, 2010) conforme o quadro 3.

Quadro 3 – Diretrizes para Pesquisa em DSR

Diretriz	Descrição
Design como um artefato	Desenvolver um diagrama de atividade para definir a metodologia e construir os contratos inteligentes no <i>hyperledger composer</i>
Relevância do problema	Implementar uma PoC baseada em <i>Blockchain</i> para viabilizar a implementação de um SLR
Avaliação de projeto	Aplicar técnicas de Teste Funcional para validar os contratos inteligentes
Contribuições da Pesquisa	Mostrar que a DLT implementada utilizando <i>Blockchain</i> , pode suportar um novo modelo de logística reversa.
Rigor na pesquisa	De modo a garantir o rigor científico, utilizar a Revisão Sistemática de Literatura com a metodologia DSR para construir e validar a PoC
Design como um processo de pesquisa	Utilizar a plataforma <i>Hyperledger Composer</i> para construir e testar os artefatos. Utilizar a linguagem <i>JavaScript</i> para codificar os contratos inteligentes
Comunicação de pesquisa	Com o intuito de fomentar a reprodução deste artefato por outros pesquisadores, procedimentar a construção da PoC e detalhar e disponibilizar os códigos fontes dos contratos inteligentes nos apêndices.

Fonte: Autor. Adaptado de (HEVNER; CHATTERJEE, 2010)

Durante a pesquisa e desenvolvimento da metodologia a fim de definir a operação do incentivo para a implantação e adesão ao SLR foi pensado no funcionamento do *Bitcoin*. Para garantir o bom funcionamento de sua rede, o *Bitcoin* evita que participantes maliciosos tentem fraudar a rede utilizando a premissa de recompensar os bons participantes, contrariando o pensamento comum que seria de punir os participantes maliciosos. Com este objetivo foi criado então a recompensa aos participantes denominados “Mineradores”, que recebem *Bitcoins* para garantir a segurança e funcionamento do sistema. Para um sistema de LR com base na PNRS que define a responsabilidade compartilhada aos participantes, não teremos mecanismos eficazes para punir quem não fizer sua parte no processo, sendo assim foi sugerido um sistema de recompensa ou incentivo, como o que ocorre com o *Bitcoin*, neste caso foi criado o criptoLata, um ativo que será utilizado como bonificação para o usuário que fizer a devolução correta da embalagem pós-consumo. Este ativo pode ser utilizado para obter descontos nas próximas compras, e isso

funciona para toda a cadeia envolvida no sistema. O criptoLata traz uma simplificação muito grande, eliminando a necessidade do controle e identificação das marcas das embalagens que estão sendo devolvidas e retornando para o ciclo de vida do produto proposto no fluxo da LR.

Para a representação gráfica da metodologia proposta para implantar este SLR foi utilizado o diagrama de atividades, que foi construído com uso da aplicação *on-line draw.io*¹², ver figura 31.

Segundo a *Unified Modeling Language* (UML), o diagrama de atividades é um diagrama de comportamento e representa de forma gráfica como será o funcionamento do software (ou processo), a execução de etapas e a integração de atividades e atores envolvidos. Mostra de forma simples como será a atuação do sistema na realidade de negócio na qual ele está inserido (VENTURA, 2016).

Para a preparação do diagrama foram usados os atores: Indústria, Lojista, Consumidor, Catadores, Siderúrgica, Aterro e Governo. E os seguintes processos: Venda Indústria, Venda ao consumidor, Devolução, Triagem e Reciclagem. O Gestor foi destacado no diagrama pois é um ator especial, que atua como controlador do processo e interage com todos os demais atores. Para maior clareza na compreensão da metodologia proposta, ao diagrama de atividades foi adicionado raias para destacar a interação dos atores com os processos, sendo as raias verticais usadas para identificar os atores e as horizontais os processos. O ativo denominado criptoLata, que será utilizado para o incentivo do SLR, foi adicionado ao diagrama exatamente nos processos no qual ele será criado ou utilizado para obtenção dos descontos.

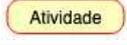
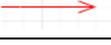
No quadro 4 para melhor leitura e compreensão do diagrama, são apresentados os componentes utilizados.

Para facilitar a compreensão e da aplicação da metodologia foi definido que cada lata colocada no mercado gera um criptoLata para a carteira do Gestor e uma unidade financeira será depositada para o Gestor fazer a administração e desenvolvimento do SLR. Abaixo são descritos todos os processos do diagrama

¹² <https://www.draw.io/>

proposto para desenvolver a metodologia de implementação da *Blockchain* e construção dos contratos inteligentes para gerir o SLR.

Quadro 4 – Componentes do diagrama de atividades

Componente	Descrição
	Início
	Fim
	Atividade
	Condição
	criptoAtivo
	Registro de dados
	Ligação de atividade
	Troca de titularidade de Ativo

Fonte: Autor.

a) Venda Indústria

Este é o primeiro processo e dá início ao SLR proposto, envolve dois agentes: a Indústria e o Lojista. Neste processo a Indústria realiza a venda para o Lojista. Nesta etapa será gerado a quantidade de criptoLata igual à quantidade de embalagens comercializadas, e uma quantidade de recursos financeiros serão repassados para o gestor. O valor investido nos criptoAtivos serão utilizados para implementação do programa, como consultorias, equipamentos, campanhas de divulgação, criação de PEVs entre outros. O Lojista interage nesta etapa pela compra do produto e recebimento do mesmo. Caso o Lojista tenha criptoLatas (veremos mais adiante como o Lojista os acumula) poderá utilizar para obter descontos na compra dos produtos, assim o criptoAtivo será transferido do Lojista para a Indústria. Caso a indústria tenha crédito de criptoLata ou receba nesta mesma transação, poderá utilizar para abater da nova quantidade de criptoLatas novos gerados e também da quantidade de recurso financeiro repassado para o gestor, assim a Indústria não será onerada

duas vezes, pelo desconto fornecido e pelo recurso aportado para o gestor.

Os dados referentes as vendas dos produtos e geração das criptoLatas serão registados e ficarão disponíveis para uso e controle do Governo ou de qualquer participante do SLR. Estes dados poderão ser utilizados para estatísticas de colocação de novas embalagens no mercado e ser cruzados com novos dados de reutilização e reciclagem, para acompanhamento das metas de cada programa e acordo setorial.

b) Venda Consumidor

Esta etapa do diagrama de atividades, envolve os agentes: Lojista e Consumidor. Caso consumidor possua créditos de criptoLatas, poderá utilizar para receber desconto em sua compra, neste caso os créditos utilizados são transferidos do Consumidor para o Lojista. Os créditos podem ter sido obtidos pela devolução das embalagens em qualquer outra empresa participante do SLR e em qualquer período, não foi abordado na metodologia a expiração da validade dos criptoLatas.

c) Devolução

Este processo possui interação de 3 agentes: Consumidor, Lojista e os Catadores. Conforme o programa se desenvolve, com divulgação em massa, criação de locais para coleta e cadastramento de associações de catadores, este processo terá fluidez e os volumes de coletas tende ao crescimento. Quando o consumidor efetuar a devolução da embalagem pós-consumo deverá ser registrado no sistema, então estará elegível para receber os créditos, recebendo uma quantidade de criptoLatas relativa a quantidade de embalagens devolvidas, independente da marca e estado da embalagem desde que sem restos de produtos em seu interior. A devolução poderá ocorrer junto com uma nova compra, porém a devolução deve ser processada antes para que o crédito possa ser registrado e consumido na transação de venda Consumidor subsequente. Os catadores que forem registrados no sistema terão acesso livre aos PEVs

para retirada das embalagens nas dependências dos Lojistas participantes, porém em trabalhos futuros pode-se incluir a movimentação de criptoLatas para este ator.

d) Triagem

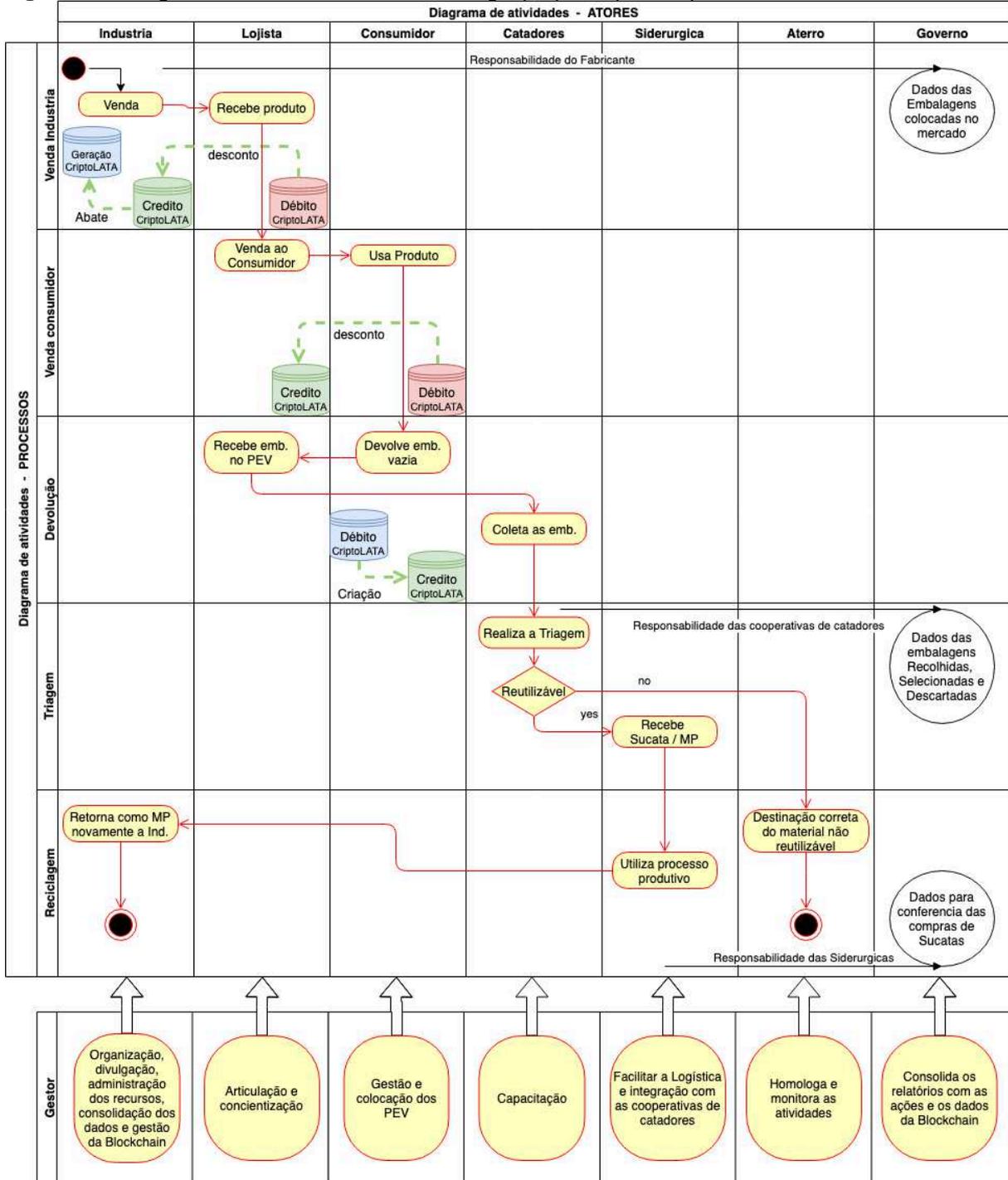
Após a coleta os catadores dão início ao processo de Triagem. Dois agentes estão envolvidos neste processo: Catadores e Siderúrgica. Os catadores realizam a triagem, separando as embalagens que poderão ser recicladas e descartadas, registrando estes dados que encaminhados para o programa, onde ficarão disponível para o Governo completar seu acompanhamento das metas dos acordos setoriais. Como as cooperativas têm receita pela comercialização das embalagens recolhidas, inicialmente seu benefício em participar do programa será em melhorar seu desempenho devido a serviços que podem ser ofertados pelo Gestor, como: consultorias, cursos de capacitação, subsídios na compra de equipamentos, etc. Benefícios este que tem como objetivo incentivar e manter a performance do programa na busca do atingimento ou superação das metas propostas.

e) Reciclagem

A reciclagem é o ponto alvo de todo o SLR, ou seja, converter o maior número de embalagens colocadas no mercado em embalagens recicladas. Muitas embalagens retornam para a indústria e são reutilizadas novamente no processo, outras são usadas em artesanatos e para outros fins. Mas a maioria acaba indo para aterros ou sendo descartadas de forma menos conscientes. Por isso a PNRS instituiu a responsabilidade compartilhada para a LR. Então este processo final do diagrama de atividades consolida as informações, tendo a Siderúrgica a responsabilidade de registrar as informações sobre as compras de sucatas, disponibilizando ao Governo os dados para fechamento das estatísticas, validação dos dados e verificação das metas estabelecidas nos acordos. Em trabalhos futuros pode ser avaliado a possibilidade de

geração de incentivos em criptoAtivos neste processo. Porém, neste estudo foi considerado que o recebimento de sucatas mais qualificadas e em maior quantidade é um incentivo a Siderúrgica participar do programa.

Figura 31 – Diagrama de atividades da metodologia proposta para implantar o SLR



Fonte: Autor

f) Gestor

A raia Gestor não é considerado um processo, mas está representado para apresentar as ações e interações do Gestor do programa com os demais atores. Como a Indústria que coloca as embalagens no mercado e a PNRS atribui a ela a responsabilidade de investir no SLR de suas embalagens, o Gestor se relaciona com a Indústria em promover, divulgar e operacionalizar o programa de LR. Para este financiamento a Indústria investe valores monetários no programa quando da geração dos criptoAtivos. Com o Lojista o Gestor atuará na conscientização e articulação para implantação de PEVs. O Consumidor será impactado pelo Gestor através das campanhas de divulgação e incentivo da devolução do programa. O Gestor não tem contato direto com consumidor, mas delegará a função de capacitar e incentivar a devolução das embalagens aos Lojistas. A relação do Gestor com os Catadores e associações de catadores, será mais próxima, por intermédio de consultoria, subsídios na compra de equipamentos e treinamentos com o intuito de ajudar a melhorar a qualidade e sustentabilidade do negócio e das pessoas que ali estão associados. Futuramente pode ser implementado o criptoLata para maior incentivo as cooperativas, permitindo sua utilização para consumo de um portfólio diferenciado desenvolvido pelo Gestor. Com as Siderúrgicas, será de aproximação com as cooperativas, de forma a facilitar a logística e reduzir os custos para ampliar o consumo da sucata, e reduzindo a exploração de minérios, assim também contribuindo para redução dos impactos ambientais desta extração e reduzindo as áreas de aterros. Com os Aterros trabalhará na homologação e monitoração das atividades. Com o Governo o Gestor tem responsabilidade atribuída pelos acordos de apresentar os relatórios consolidados das ações e com os dados armazenados na *Blockchain*, garantindo transparência e segurança das informações, necessária para que o programa possa ter sucesso devido a interação de tantos atores parceiros e concorrentes simultaneamente.

5.1 Desenvolvimento da PoC

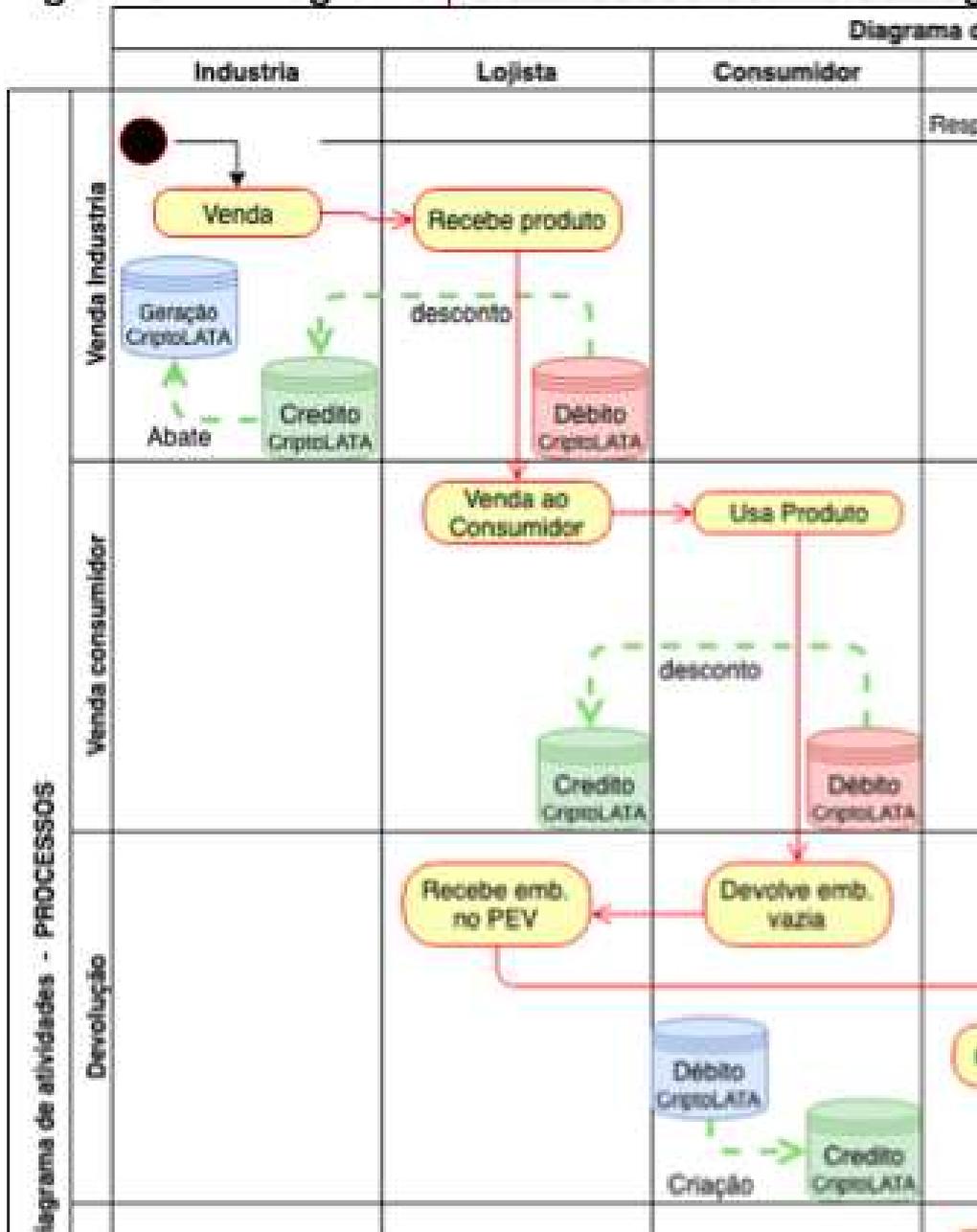
Para desenvolver a PoC foi optada pelo *Hyperledger*, pois tem como definição base a construção de DLT voltada para negócios, possuir contratos inteligentes, seus participantes serem conhecidos (*Blockchain* privada), além de possuir uma ferramenta de fácil programação e recursos para testes da rede de negócio desenvolvida, o *Hyperledger Composer Playground*.

Para simplificação da PoC foram implementadas somente as transações que possuem interação com o ativo criptoLata: VendaIndustria, VendaConsumidor e Devolução, conforme recorte do diagrama de atividades da metodologia na figura 32. As informações e cálculos relativo aos valores financeiros das transações foram desconsiderados neste trabalho, ficando como sugestão para trabalhos futuros, inclusive de avaliar se o criptoLata poderá sofrer variação conforme a lei da oferta e procura, ou mesmo especulação.

Para a construção do contrato inteligente foi utilizado a linguagem *JavaScript*. A ferramenta *Hyperledger Composer Playground* possui uma estrutura bem definida com linguagem de modelagem, linguagem de script e linguagem de query, que atendem aos requisitos das linguagens orientadas a objeto, como herança e abstração. No capítulo 2 item 2.3 temos todas as informações sobre a ferramenta.

Para modelagem da rede utilizando o *Hyperledger*, temos as classes *Participants*, *Assets* e *Transactions* que serão detalhadas a seguir, e são implementadas no *Hyperledger Composer Playground* no arquivo *Model Files*.

Figura 32 – Recorte do diagrama de atividades



Fonte: Autor

Para esta rede foi criado uma classe *Participants* abstrata denominada “Pessoa” e outra classe denominada “Participante” estendida da classe Pessoa com a adição da propriedade TipoParticipante. Desta maneira será possível registrar todos os participantes da rede de negócio. Esta propriedade poderá assumir um dos tipos de participantes predefinidos: Industria, Lojista, Consumidor, Catadores, Siderúrgica, Aterro, Governo e Gestor. A figura 33 apresenta o código para a definição dos

Participants. O item 2.3.3 apresenta as características da linguagem de modelagem utilizada pelo *Hyperledger Composer Playground*.

Para a classe de *Assets* utilizamos duas classes: “AtivoVenda” e “CarteiraCriptoLata”, ver figura 34. A classe AtivoVenda registra os dados da transação comercial, como número da nota na propriedade vendaID. A propriedade TipoOperacao, pode receber o conteúdo predefinido como venda_indústria, venda_consumidor ou devolução, através de seu tipo eNum TipoNota. As propriedades VendedorID e CompradorID são referências a classe Participante e dependendo do tipo da operação necessita que o Participante seja de um determinado TipoParticipante específico. A propriedade qtdEmbalagens indica a quantidade de embalagens colocadas no mercado, ou produtos comercializados nesta transação. A propriedade crédito registra se foi utilizado crédito do ativo criptoLata para concessão de descontos, esta propriedade é opcional pois nem toda venda ocorrerá a utilização de créditos.

A classe CarteiraCriptoLata foi criada para facilitar a consulta do saldo dos créditos de criptoLata de cada participante da rede. Este será movimentado pelo contrato inteligente, conforme realiza os registros das vendas irá creditar e/ou debitar estes créditos para os participantes da transação e podem ser consultados a qualquer momento. A propriedade criptoID receberá a mesma informação da propriedade codigoID do participante, a propriedade titular recebe a referência do participante e a propriedade saldo recebe o saldo de criptoLata que o participante titular possui. O código completo da definição da rede *Blockchain* encontra-se no apêndice A.

Para melhor compreensão do relacionamento entre as classes, faz-se uma analogia com uma conta bancária tradicional, a classe AtivoVenda registraria todas as movimentações de entrada e saída de dinheiro de sua conta. A classe CarteiraCriptoLata seria o saldo atual da sua conta. Então a cada novo movimento na conta o saldo é atualizado. Porém nesta rede, o saldo da CarteiraCriptoLata somente sofrerá alteração se o registro na classe AtivoVenda tiver conteúdo na propriedade crédito.

Figura 33 – Parte do Arquivo *Model Files*, definição dos *participants*

```
Model File models/model.cto
17
20 // DEFINE OS PARTICIPANTES
21
22 concept Endereco {
23   o String cidade
24   o String uf
25 }
26
27 abstract participant Pessoa identified by codigoId {
28   o String codigoId
29   o String name
30   o Endereco endereco optional
31 }
32
33 enum TipoParticipante {
34   o INDUSTRIA
35   o LOJISTA
36   o CONSUMIDOR
37   o CATADORES
38   o SIDERURGICA
39   o ATERRO
40   o GOVERNO
41   o GESTOR
42 }
43
44 participant Participante extends Pessoa {
45   o TipoParticipante tipo
46 }
```

Fonte: Autor

5.2 Contratos Inteligentes

No *Hyperledger Composer Playground* os contratos inteligentes são escritos na forma de transações, como se fossem funções nas linguagens tradicionais e deverão ser codificadas no arquivo *Script Model*, mas necessitam ser definidas previamente no arquivo *Model Files* como visto na figura 35. Para nossa rede definimos as 3 transações anteriormente definidas: VendaIndustria,

VendaConsumidor e Devolução. Também foi definido o *event* MensagemEvent para ser usado na comunicação do contrato inteligente e uma aplicação cliente, em caso de implementação futura. Como para a PoC será utilizado a ferramenta *Composer* para executar os testes, a comunicação com o usuário será pelo envio de mensagens através da console da ferramenta.

Na definição das *transactions*, é necessário especificar os parâmetros de entrada que serão necessários para sua execução. Na etapa de teste da rede, estes parâmetros devem ser preenchidos no formato JSON e serão validados antes mesmo de sua execução. O *Composer* auxilia o usuário carregando todos os parâmetros e sugerindo conteúdos aleatórios, porém parâmetros que são referência a um *Participant* ou *Asset* devem estar registrados na rede, caso contrário a execução é cancelada. O mesmo ocorre com parâmetros de tipos eNum, o conteúdo deve fazer parte da lista de opções previamente definida.

Os contratos inteligentes para a VendaIndustria e VendaConsumidor possuem os mesmos parâmetros: VendaID, vendedorID, compradorID, qtdEmbalagens e crédito. O contrato da Devolução possui os mesmos parâmetros com exceção da qtdEmbalagens.

Cada contrato deve ser programado para realizar as validações necessárias para atender as regras do negócio. Esta programação deve acionar as classes que registram as transações e alteram os status dos ativos, até mesmo a criação de ativos se for necessário.

Um *Participant* para o ator “Gestor” é uma premissa desta rede, logo este necessita estar criado antes da primeira transação para que os incentivos possam ser criados. O criptoLata proposto nesta metodologia não se utiliza do mesmo método de criação das criptomoedas, que em sua inicialização tem um volume muito grande de moedas gerados. A proposta para esta rede é gerar criptoLata toda vez que o contrato inteligente ou a transação “VendaIndustria” for executada. Não é prerrogativa deste projeto definir quanto em reais (R\$) um criptoLata representará para o SLR, podendo ficar como sugestão para trabalhos futuros. Então para efeito dos testes desta rede, foi considerado que uma embalagem representa um criptoLata, tanto para “VendaIndustria” como para “Devolução”. Esta rede não irá controlar a quantidade de

embalagem vendida ou comprada pelos participantes, mas sim os créditos dos ativos criptoLatas.

Figura 34 – Parte do Arquivo *Model Files*, definição dos *assets*.

```
Model File models/model.cto   
47  
48 // DEFINE OS ATIVOS  
49 enum TipoNota {  
50   o VENDA_INDUSTRIA  
51   o VENDA_CONSUMIDOR  
52   o DEVOLUCAO  
53 }  
54  
55 asset CarteiraCriptoLata identified by criptoId {  
56   o String criptoId  
57   --> Participante titular  
58   o Integer saldo  
59 }  
60  
61 asset AtivoVenda identified by vendaId {  
62   o String vendaId  
63   o TipoNota tipoOperacao  
64   --> Participante vendedorId  
65   --> Participante compradorId  
66   o Integer qtdEmbalagens  
67   o Integer credito optional  
68 }
```

Fonte: Autor

Um *Participant* para o ator “Gestor” é uma premissa desta rede, logo este necessita estar criado antes da primeira transação para que os incentivos possam ser criados. O criptoLata proposto nesta metodologia não se utiliza do mesmo método de criação das criptomoedas, que em sua inicialização tem um volume muito grande de moedas gerados. A proposta para esta rede é gerar criptoLata toda vez que o contrato inteligente ou a transação “VendaIndustria” for executada. Não é prerrogativa deste projeto definir quanto em reais (R\$) um criptoLata representará para o SLR, podendo ficar como sugestão para trabalhos futuros. Então para efeito dos testes desta rede, foi considerado que uma embalagem representa um criptoLata, tanto para

“VendaIndustria” como para “Devolução”. Esta rede não irá controlar a quantidade de embalagem vendida ou comprada pelos participantes, mas sim os créditos dos ativos criptoLatas.

Detalhando as regras para construção das transações no contrato inteligente:

- a) A transação “VendaIndustria”, ver apêndice D, é o fato gerador das criptoLatas, e vendedorID obrigatoriamente deve ser um Participante do tipo INDUSTRIA e o compradorID um LOJISTA, estes são pré-requisitos para sua execução e devem ser tratados pelo contrato inteligente. Cada embalagem vendida aumenta o saldo de criptoLatas do Gestor, que é responsável por implementar toda logística e treinamentos para que o SLR seja eficaz e eficientes. Numa implementação real deste projeto, espera-se que cada criptoLata gerado, um valor financeiro deva ser depositado na conta do Gestor. Quando a Industria realizar suas vendas e conceder descontos para os Lojistas mediante a transferência de saldo de criptoLata este também receberá um abatimento ao realizar o deposito ao gestor relativo as embalagens vendidas (colocadas no mercado).
- b) A transação “VendaConsumidor”, ver apêndice E, necessita que o vendedorID seja um Participante do tipo LOJISTA e o compradorID seja um CONSUMIDOR. Caso o consumidor tenha criptoLata poderá receber desconto do Lojista, porém o saldo de criptoLata utilizado na transação será do consumidor para o Lojista, que poderá se beneficiar com descontos em futuras compras com a Indústria que estiver participando do programa.
- c) A transação “Devolução”, ver apêndice F, é o processo mais esperado do SLR, é quando o consumidor conscientizado ou interessado no incentivo, entrega as embalagens pós-consumo ao Lojista ou em PEVs autorizados, assim acumulando criptoLata para obter descontos nas próximas compras. Nesta proposta, a entrega deve ser realizada e monitorada por um agente que irá fazer o registro e assim o criptoLata é creditado ao consumidor e debitado do saldo do Gestor. Esta transação necessita que o vendedorID seja um Participante do tipo LOJISTA e o compradorID do tipo CONSUMIDOR. Futuramente pode-se criar outros contratos inteligentes

que podem credenciar CATADORES a receber as embalagens diretamente e registrar a transação para que os saldos sejam transferidos do gestor para o consumidor, porém não consideramos este processo para a PoC.

Figura 35 – Parte do Arquivo *Model Files*, definição das *transactions*.

```
Model File models/model.cto   
70 //----- Contratos Inteligentes -----  
71 // DEFINE AS TRANSAÇÕES  
72  
73 transaction VendaIndustria {  
74     o String vendaId  
75     --> Participante vendedorId  
76     --> Participante compradorId  
77     o Integer qtdEmbalagens  
78     o Integer credito  
79 }  
80  
81 transaction VendaConsumidor {  
82     o String vendaId  
83     --> Participante vendedorId  
84     --> Participante compradorId  
85     o Integer qtdEmbalagens  
86     o Integer credito  
87 }  
88  
89 transaction Devolucao {  
90     o String vendaId  
91     --> Participante consumidorId  
92     --> Participante lojistaId  
93     o Integer credito  
94 }  
95  
96 event MensagemEvent {  
97     o String cMensagem  
98 }
```

Fonte: Autor

Para facilitar os testes durante o desenvolvimento, que exige muitas simulações criando e apagando registros, foi criada uma transação denominada *CargaInicial* (ver apêndice B) que cria os *Participants*, e uma transação chamada *LimpaMovimento* (ver apêndice C), que elimina todos os registros dos *Assets*, *VendaAtivo* e *CarteiraCriptoLatas*. O *Historian Record* que é o histórico da rede, ou melhor o *Ledger* da *Blockchain*, não tem como ser limpo e nem alterado, devido a característica de imutabilidade. Como o *Composer* possui interface de consultas isso facilita muito os testes da rede, porém para efeito de testarmos a linguagem de consulta, foi criada uma transação para permitir a consulta das vendas, utilizando assim a linguagem de *query*, onde o resultado foi apresentado no console. Também foi criado um evento para testar a capacidade de comunicação em caso de usar uma aplicação cliente para ter acesso a rede, ver apêndices G e H.

6 RESULTADOS E DISCUSSÕES

Os testes para validar a *Blockchain* e os contratos inteligentes desenvolvidos para a realização da PoC, validando o modelo sugerido para gestão de LR, foi implementado manualmente para que possa ser reproduzido facilmente pelos leitores.

Algumas premissas foram assumidas para a composição do cenário e definições das informações que são passadas como parâmetro nas transações, os contratos inteligentes da *Blockchain*.

Toda a descrição para construção e validação dos testes está aqui detalhado para que possa ser reproduzido por qualquer outra pessoa. O quadro 5 contém a lista dos participantes que devem ser gerados para que a reprodução dos testes siga conforme o esperado, a figura 36 apresenta os participantes registrados na *Blockchain* na ferramenta *Hyperledger Composer Playground*.

Quadro 5 – Participantes

codigoid	tipo
gestor	GESTOR
P1	INDUSTRIA
P2	LOJISTA
P3	CONSUMIDOR

Fonte: Autor

O quadro 6 apresenta as transações que foram implementadas, conforme mencionado, foram desenvolvidas somente as transações que tem iteração com o ativo criptoLata. O objetivo é apresentar os parâmetros de entrada que cada transação necessita e uma breve descrição de sua funcionalidade.

Com o intuito de simplificar e facilitar a validação dos testes, assume-se que os participantes registrados são aqueles que tem ação direta com o criptoLata, ou seja, Industria, Lojista e Consumidor e somente uma instância de cada um dos tipos.

A tabela 4 apresenta 47 combinações de transações e participantes variando os valores. A Tabela 5 possui os resultados esperados com a execução das transações, indicando os saldos das CarteirasCriptoLatas após a movimentação. A coluna “Mensagem” indica se é esperado que a transação seja realizada com sucesso (OK) ou se irá gerar uma inconsistência e conseqüentemente não ser completada, neste caso não registrando nenhuma movimentação nos ativos e indicando qual regra

foi acionada para a não execução do contrato, facilitando para o testador validar ou não o contrato inteligente. Todas as transações executadas com êxito são registradas no *Ledger* e alteram o estado mundial do Ativo da rede de negócios, *Blockchain*.

A tabela 6 por sua vez apresenta simplesmente o resultado obtido da execução dos contratos inteligentes utilizando os parâmetros de entrada definido na tabela 4. Observar que todas as tabelas 4, 5 e 6 possui a primeira coluna identificada com ID, usada para relacionar as 3 tabelas utilizadas neste Teste Funcional.

Quadro 6 – Lista das transações

Transações	Parâmetros	Definição
Venda Industria	vendald, vendedorId, compradorId, qtdEmbagens, crédito	Vendald deve ser única na rede. VendedorId deve ser um participante do tipo INDUSTRIA e compradorId do tipo LOJISTA. Por definição do modelo cada qtdEmbagens representa uma criptoLata gerada no sistema. Quando a indústria possuir créditos, poderá utilizar para abater do número de criptoLatas que serão geradas e consequentemente reduz o valor transferido ao gestor. O crédito quando informado irá prover desconto ao lojista e o saldo deve ser debitando o lojista e creditando na indústria.
Venda Consumidor	vendald, vendedorId, compradorId, qtdEmbagens, crédito	Vendald deve ser única na rede. VendedorId deve ser do tipo LOJISTA e o compradorId um CONSUMIDOR. A qtdEmbalagens ser positiva. O crédito quando informado irá prover desconto ao consumidor e o saldo deve mudar de carteira, debitando o consumidor e creditando o lojista.
Devolução	vendald, consumidorId, lojistaId, crédito	Vendald deve ser única na rede. ConsumidorId deve ser do tipo CONSUMIDOR e lojistaId um LOJISTA. O crédito identifica a quantidade de crédito gerada para o consumidor como recompensa pela colaboração com o processo da LR. Este crédito poderá ser utilizado na próxima compra do consumidor para obtenção de desconto. Neste processo um crédito de criptoLata debita o Gestor e credita o Consumidor.

Fonte: Autor

Tabela 4 – Regras para os testes da PoC

continua

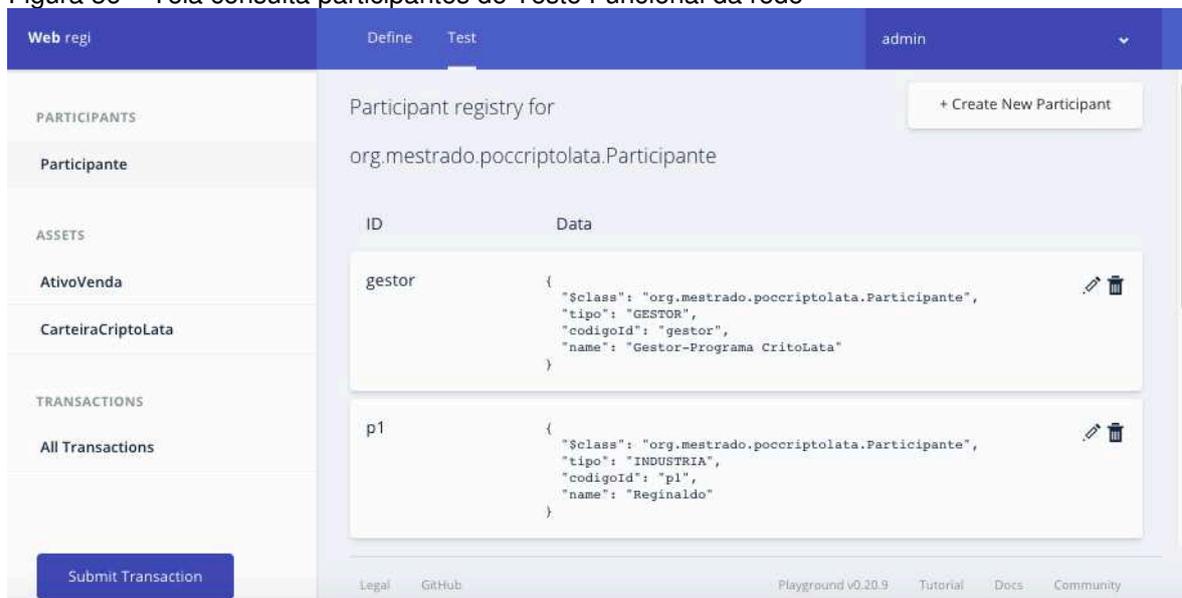
ID	Transação	Ativo Venda	Participantes		Valores	
	VendaIndustria		industrialID	LojistaID	qtdEmbalagens	Creditos
	VendaConsumidor		lojistaID	consumidorID		
	Devolução		consumidorID	LojistaID		
1	VendaIndustria	NF1	P1	P1	100	0
2	VendaIndustria	NF1	P1	P2	100	0
3	VendaIndustria	NF1	P1	P2	100	0
4	VendaIndustria	NF2	P1	P2	50	0

Tabela 4 – Regras para os testes da PoC

ID	Transação	Ativo Venda	Participantes		Valores		Conclusão
			industrialID	LojistaID	qtdEmbalagens	Creditos	
			lojistaID	consumidorID			
			consumidorID	LojistaID			
5	VendaIndustria	NF3	P1	P2	100	3	
6	VendaIndustria	NF4	P1	P2	0	0	
7	VendaIndustria	NF5	P2	P1	100	0	
8	VendaIndustria	NF5	P2	P2	100	0	
9	VendaIndustria	NF5	P2	P3	100	0	
10	VendaIndustria	NF5	P3	P1	100	0	
11	VendaIndustria	NF5	P3	P2	100	0	
12	VendaIndustria	NF5	P3	P3	100	0	
13	VendaConsumidor	NF6	P1	P1	10	0	
14	VendaConsumidor	NF6	P1	P2	10	0	
15	VendaConsumidor	NF6	P1	P3	10	0	
16	VendaConsumidor	NF6	P2	P1	10	0	
17	VendaConsumidor	NF6	P2	P2	10	0	
18	VendaConsumidor	NF6	P2	P3	10	0	
19	VendaConsumidor	NF6	P2	P3	10	0	
20	VendaConsumidor	NF7	P3	P1	10	0	
21	VendaConsumidor	NF7	P3	P2	10	0	
22	VendaConsumidor	NF7	P3	P3	10	0	
23	VendaConsumidor	NF8	P2	P3	10	3	
24	VendaConsumidor	NF8	P2	P3	0	0	
25	VendaConsumidor	NF8	P2	P3	-10	0	
26	VendaConsumidor	NF6	P2	P3	10	0	
27	VendaConsumidor	NF8	P2	P3	0	-3	
28	Devolução	NF9	P1	P1		1	
29	Devolução	NF9	P1	P2		1	
30	Devolução	NF9	P1	P3		1	
31	Devolução	NF9	P2	P1		1	
32	Devolução	NF9	P2	P2		1	
33	Devolução	NF9	P2	P3		1	
34	Devolução	NF9	P3	P1		1	
35	Devolução	NF9	P3	P2		1	
36	Devolução	NF9	P3	P2		1	
37	Devolução	NF10	P3	P2		-1	
38	Devolução	NF10	P3	P3		2	
39	VendaConsumidor	NF11	P2	P3	3	2	
40	VendaConsumidor	NF11	P2	P3	3	1	
41	VendaIndustria	NF11	P1	P2	10	2	
42	VendaIndustria	NF21	P1	P2	10	1	
43	VendaConsumidor	NF22	P2	P3	9	0	
44	Devolução	NF23	P3	P2		1	
45	Devolução	NF24	P3	P2		2	
46	VendaConsumidor	NF25	P2	P3	4	2	
47	VendaIndustria	NF26	P1	P2	2	1	

Fonte: Autor

Figura 36 – Tela consulta participantes do Teste Funcional da rede



Fonte: Rede de teste <https://composer-playground.mybluemix.net/test#>

Tabela 5 – Resultados esperados para o teste da PoC

continua

ID	Resultado esperado (CriptoLata)					Mensagem
	AtivoVenda	Gestor	P1	P2	P3	
1	NF1	0	0	0	0	Erro tipo participante
2	NF1	100	0	0	0	OK
3	NF1	100	0	0	0	Erro NF duplicada
4	NF2	150	0	0	0	OK
5	NF3	150	0	0	0	Erro Lojista sem credito
6	NF4	150	0	0	0	Erro valores zero
7	NF5	150	0	0	0	Erro tipo participante
8	NF5	150	0	0	0	Erro tipo participante
9	NF5	150	0	0	0	Erro tipo participante
10	NF5	150	0	0	0	Erro tipo participante
11	NF5	150	0	0	0	Erro tipo participante
12	NF5	150	0	0	0	Erro tipo participante
13	NF6	150	0	0	0	Erro tipo participante
14	NF6	150	0	0	0	Erro tipo participante
15	NF6	150	0	0	0	Erro tipo participante
16	NF6	150	0	0	0	Erro tipo participante
17	NF6	150	0	0	0	Erro tipo participante
18	NF6	150	0	0	0	OK
19	NF6	150	0	0	0	Erro tipo participante
20	NF7	150	0	0	0	Erro tipo participante
21	NF7	150	0	0	0	Erro tipo participante
22	NF7	150	0	0	0	Erro tipo participante
23	NF8	150	0	0	0	Erro consumidor sem credito
24	NF8	150	0	0	0	Erro valores zero

Tabela 5 – Resultados esperados para o teste da PoC

ID	Resultado esperado (CriptoLata)					Mensagem	conclusão
	AtivoVenda	Gestor	P1	P2	P3		
25	NF8	150	0	0	0	Erro valor negativo	
26	NF6	150	0	0	0	Erro NF duplicada	
27	NF8	150	0	0	0	Erro valor negativo	
28	NF9	150	0	0	0	Erro tipo participante	
29	NF9	150	0	0	0	Erro tipo participante	
30	NF9	150	0	0	0	Erro tipo participante	
31	NF9	150	0	0	0	Erro tipo participante	
32	NF9	150	0	0	0	Erro tipo participante	
33	NF9	150	0	0	0	Erro tipo participante	
34	NF9	150	0	0	0	Erro tipo participante	
35	NF9	149	0	0	1	OK	
36	NF9	149	0	0	1	Erro NF duplicada	
37	NF10	149	0	0	1	Erro valor negativo	
38	NF10	149	0	0	1	Erro tipo participante	
39	NF11	149	0	0	1	Erro consumidor credito insuficiente	
40	NF11	149	0	1	0	OK	
41	NF11	149	0	1	0	Erro NF duplicada	
42	NF21	159	0	0	0	OK	
43	NF22	159	0	0	0	OK	
44	NF23	158	0	0	1	OK	
45	NF24	156	0	0	3	OK	
46	NF25	156	0	2	1	OK	
47	NF26	158	0	1	1	OK	

Fonte: Autor

Tabela 6 – Resultados obtidos com a execução das transações

ID	Resultado Obtido na execução (CriptoLata)					Mensagem	continua
	CarteiraCriptoLata			P3			
Gestor	P1	P2					
1	0	0	0	0	0	Erro	
2	100	0	0	0	0	OK	
3	100	0	0	0	0	Erro	
4	150	0	0	0	0	OK	
5	150	0	0	0	0	Erro	
6	150	0	0	0	0	Erro	
7	150	0	0	0	0	Erro	
8	150	0	0	0	0	Erro	
9	150	0	0	0	0	Erro	
10	150	0	0	0	0	Erro	
11	150	0	0	0	0	Erro	
12	150	0	0	0	0	Erro	
13	150	0	0	0	0	Erro	
14	150	0	0	0	0	Erro	
15	150	0	0	0	0	Erro	
16	150	0	0	0	0	Erro	

Tabela 6 – Resultados obtidos com a execução das transações

conclusão

ID	Resultado Obtido na execução (CriptoLata)					Mensagem
	CarteiraCriptoLata					
	Gestor	P1	P2	P3		
17	150	0	0	0		Erro
18	150	0	0	0	0	OK
19	150	0	0	0		Erro
20	150	0	0	0		Erro
21	150	0	0	0		Erro
22	150	0	0	0		Erro
23	150	0	0	0		Erro
24	150	0	0	0		Erro
25	150	0	0	0		Erro
26	150	0	0	0		Erro
27	150	0	0	0		Erro
28	150	0	0	0		Erro
29	150	0	0	0		Erro
30	150	0	0	0		Erro
31	150	0	0	0		Erro
32	150	0	0	0		Erro
33	150	0	0	0		Erro
34	150	0	0	0		Erro
35	149	0	0	1	1	OK
36	149	0	0	1		Erro
37	149	0	0	1		Erro
38	149	0	0	1		Erro
39	149	0	0	1		Erro
40	149	0	1	0		OK
41	149	0	1	0		Erro
42	159	0	0	0		OK
43	159	0	0	0	0	OK
44	158	0	0	1	1	OK
45	156	0	0	3	3	OK
46	156	0	2	1	1	OK
47	158	0	1	1	1	OK

Fonte: Autor

Para execução dos testes, seguir os passos do item 2.3.2. para criar a instancia da rede de negócio e para a execução dos contratos inteligentes. Os dados de entrada para as transações, utilizar os registros relacionados na tabela 4.

A figura 37 apresenta a tela com a transação “VendaIndustria” e os dados preenchidos conforme consta na tabela 4, coluna ID número 1. A figura 37 também mostra que ao submeter a transação o contrato inteligente validou suas regras e identificou que não é permitido o vendedorID e compradorID serem o mesmo

participante, logo esta transação não foi executada com sucesso, pois esta linha do teste tentava induzir a transação a um erro, e uma mensagem no console do *Hyperledger Composer* indicando o qual o motivo da não execução da transação.

Na figura 38 tem-se a tela de execução da segunda linha do Teste Funcional (tabela 4), ou seja, a coluna ID número 2. Esta transação foi executada com sucesso, pois, todos as condições do contrato inteligente “VendaIndustria” foram atendidas. A figura 39 apresenta o registro do *Asset* AtivoVenda e a figura 40 o registro do *Asset* CarteiraCriptoLata, onde o Gestor recebeu um saldo de 100 criptoLatas. Como a transação foi realizada com sucesso, figura 41, a *Blockchain* realizou todo seu processo de criar o bloco, adicionar o *timestamp*, gerar o *hash* do bloco e submeter a rede para arquivamento do *Ledger*, figura 42.

Figura 37 –Teste Funcional – Primeira linha do teste

Submit Transaction

Transaction Type: VendaIndustria

JSON Data Preview

```
1 {
2   "$class": "org.mestrado.pocriptomata.VendaIndustria",
3   "vendaId": "NF1",
4   "vendedorId":
5     "resource:org.mestrado.pocriptomata.Participante#p1",
6   "compradorId":
7     "resource:org.mestrado.pocriptomata.Participante#p1",
8   "qtdEmbalagens": 100,
9   "credito": 0
10 }
```

Optional Properties

Error: 1-Venda nao realizada! Vendedor nao e INDUSTRIA ou Comprador nao e LOJISTA!!!
Vendedor = (p1) INDUSTRIA Comprador = (p1) INDUSTRIA

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

Fonte: Rede de teste <https://composer-playground.mybluemix.net/test#>

Figura 38 –Teste Funcional – Segunda linha do teste

Submit Transaction

Transaction Type: VendaIndustria

JSON Data Preview

```

1 {
2   "$class": "org.mestrado.pocriptomata.VendaIndustria",
3   "vendaId": "NF1",
4   "vendedorId":
5   "resource:org.mestrado.pocriptomata.Participante#p1",
6   "compradorId":
7   "resource:org.mestrado.pocriptomata.Participante#p2",
8   "qtdEmbalagens": 100,
9   "credito": 0
10 }

```

Optional Properties

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

Fonte: Rede de teste <https://composer-playground.mybluemix.net/test#>

Figura 39 –Teste Funcional – Consulta Asset AtivoVenda primeiro registros da Blockchain

Asset registry for org.mestrado.pocriptomata.AtivoVenda

+ Create New Asset

ID	Data
NF1	<pre> { "\$class": "org.mestrado.pocriptomata.AtivoVenda", "vendaId": "NF1", "tipoOperacao": "VENDA_INDUSTRIA", "vendedorId": "resource:org.mestrado.pocriptomata.Participante#p1", "compradorId": "resource:org.mestrado.pocriptomata.Participante#p2", "qtdEmbalagens": 100, "credito": 0 } </pre>

Collapse

Fonte: Rede de teste <https://composer-playground.mybluemix.net/test#>

Figura 40 –Teste Funcional – Consulta Asset CarteiraCriptoLata primeiro registros

ID	Data
gestor	{ "class": "org.mestrado.poccriptolata.CarteiraCriptoLata", "criptoid": "gestor", "titular": "resource:org.mestrado.poccriptolata.Participante#gestor", "saldo": 100 }
p1	{ "class": "org.mestrado.poccriptolata.CarteiraCriptoLata", "criptoid": "p1", "titular": "resource:org.mestrado.poccriptolata.Participante#p1", "saldo": 0 }

Fonte: Rede de teste <https://composer-playground.mybluemix.net/test#>

Figura 41 –Teste Funcional – Consulta lista das transações da *Blockchain*

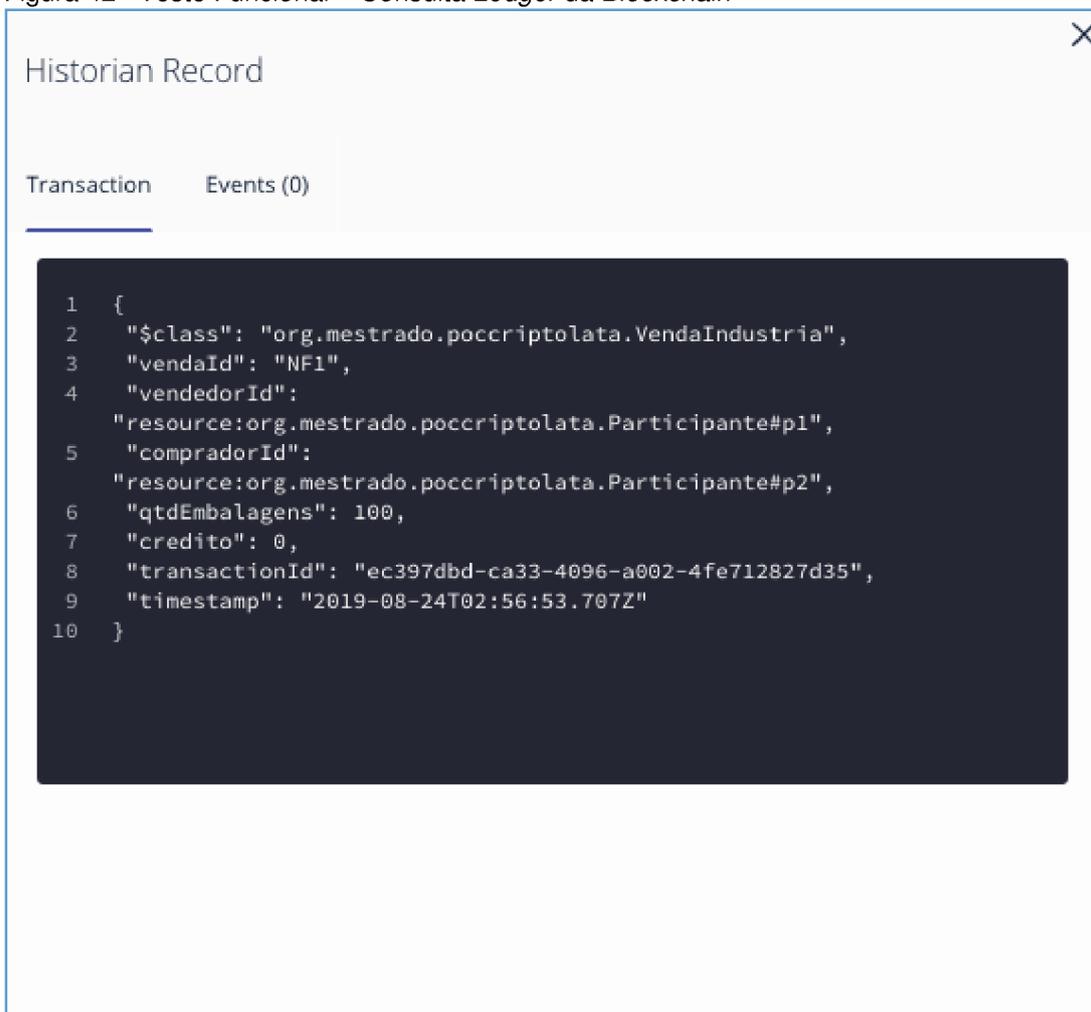
Participante	Date, Time	Entry Type	Participant
AtivoVenda	2019-08-23, 23:56:53	VendaIndustria	admin (NetworkAdmin) view record

Fonte: Rede de teste <https://composer-playground.mybluemix.net/test#>

Com base nos testes realizados, executando sequencialmente todas as 47 hipóteses da tabela 4, aplicadas a *Blockchain*, onde foi previsto todas combinações possíveis para colocar a prova os contratos inteligentes desta PoC. Após aplicar todas as configurações e fazer as devidas anotações, chegou-se a uma assertividade de 100% dos resultados esperados. Logo conclui-se que os contratos inteligentes desenvolvidos para esta PoC representam fielmente o modelo proposto e pode-se dizer que a PoC foi bem sucedida. Demonstrando assim que a tecnologia *Blockchain* pode suportar o modelo proposto para gestão de um processo de Logística Reversa, respondendo a nossa questão de pesquisa.

A figura 43 apresenta um gráfico com as transações que resultaram em sucesso no Teste Funciona e apresenta o saldo do Gestor movimentado em cada transação, e uma visão tabular destes dados é apresentado na tabela 7. Foram 11 transações cujo resultado movimentou o saldo da carteira do Gestor.

Um gráfico apresentando os saldos de cada participante utilizado na PoC, ou seja, P1=Industria, P2=Lojista e P3=Consumidor, pode ser visualizado na figura 44.

Figura 42 – Teste Funcional – Consulta *Ledger* da *Blockchain*

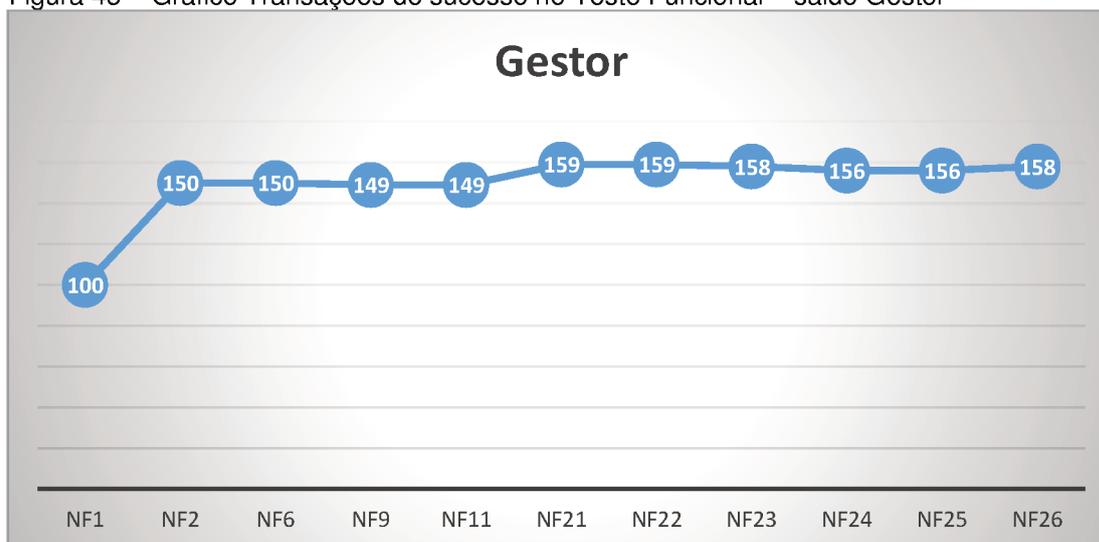
Historian Record

Transaction Events (0)

```
1  {
2    "$class": "org.mestrado.poccriptolata.VendaIndustria",
3    "vendaId": "NF1",
4    "vendedorId":
5    "resource:org.mestrado.poccriptolata.Participante#p1",
6    "compradorId":
7    "resource:org.mestrado.poccriptolata.Participante#p2",
8    "qtdEmbalagens": 100,
9    "credito": 0,
10   "transactionId": "ec397dbd-ca33-4096-a002-4fe712827d35",
11   "timestamp": "2019-08-24T02:56:53.707Z"
12 }
```

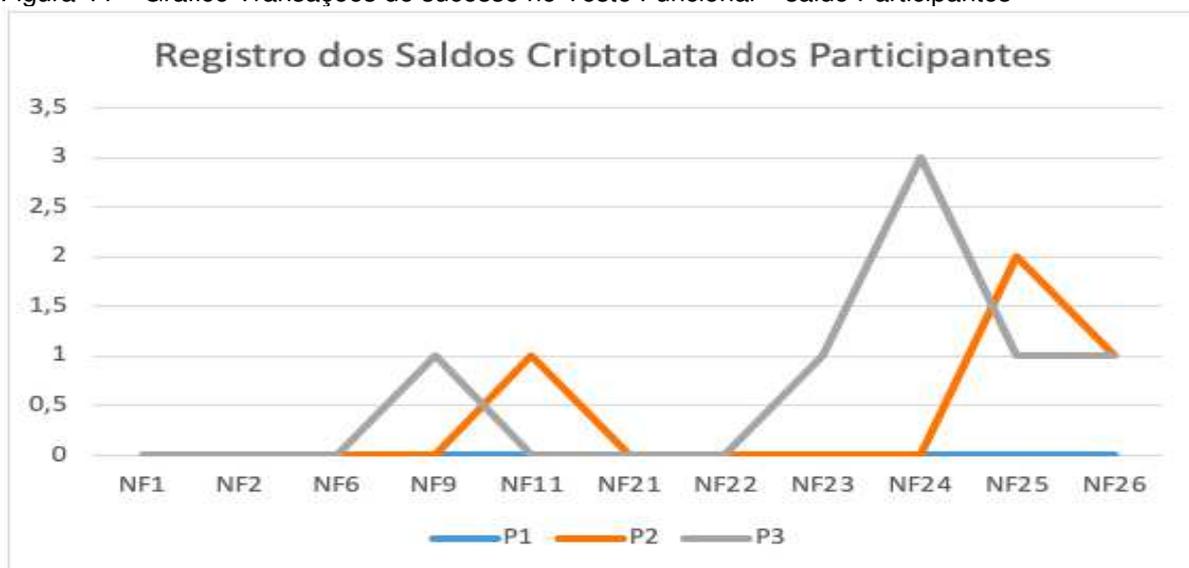
Fonte: Rede de teste <https://composer-playground.mybluemix.net/test#>

Figura 43 – Gráfico Transações de sucesso no Teste Funcional – saldo Gestor



Fonte: Autor

Figura 44 – Gráfico Transações de sucesso no Teste Funcional – saldo Participantes



Fonte: Autor

A figura 45 apresenta o comparativo entre o resultado esperado e o resultado obtido, demonstrando de forma gráfica a assertividade de 100% nos testes aplicados na PoC. De um total de 47 hipóteses testadas, 11 foram registradas na Blockchain, sendo as barras azuis os resultados esperados e as barras laranja os resultados obtidos.

Figura 45 – Gráfico comparativo dos resultados



Fonte: Autor

Das 47 hipóteses testadas, foram avaliados os erros e acertos esperados, e todas as 47 satisfizeram os testes planejados. Assim o gráfico da figura 46 apresenta uma pizza com 100% verdadeiro, ou seja, os contratos inteligentes atenderam todas as regras previstas para esta PoC.

Figura 46 – Gráfico Percentual comparação resultado esperado x obtidos

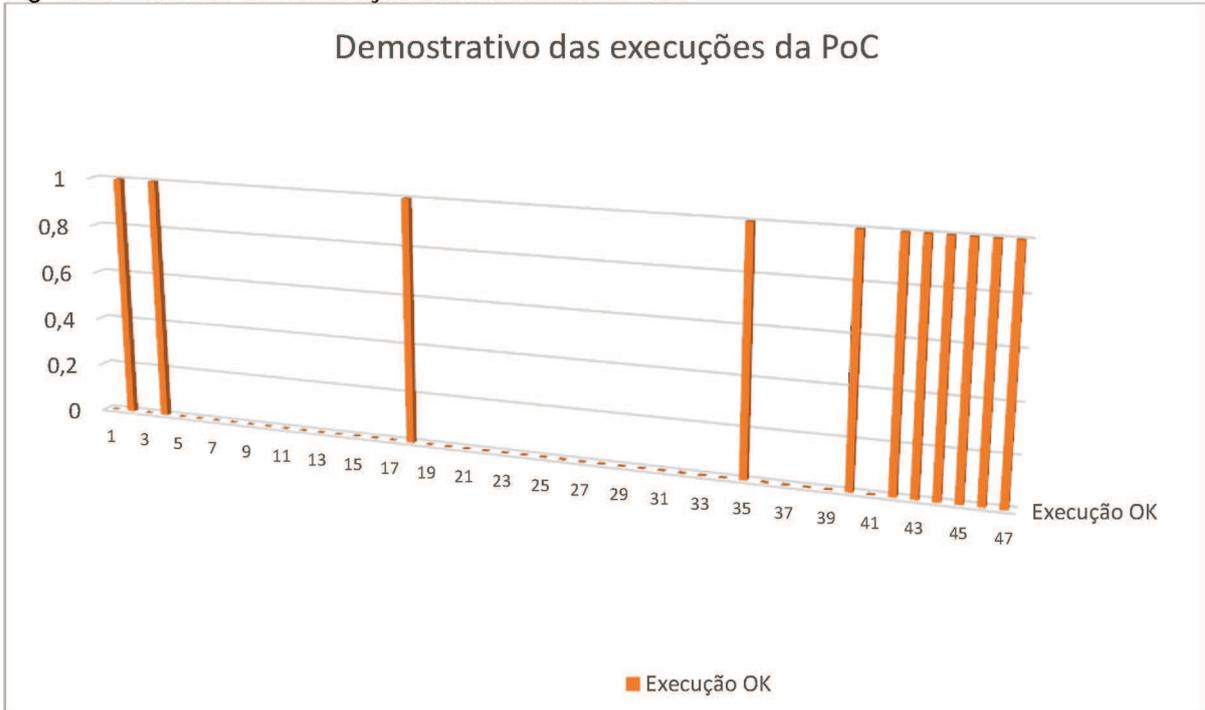


Fonte: Autor

Um demonstrativo da execução das hipóteses são apresentadas na figura 47, onde o eixo X têm-se os ID das hipóteses definidas na tabela 4, as hipóteses que foram executadas geraram registro na blockchain foram atribuídos valor 1 (um) para o gráfico e as que apresentaram mensagens no console e não atenderam as regras dos contratos inteligentes receberam valor 0 (zero).

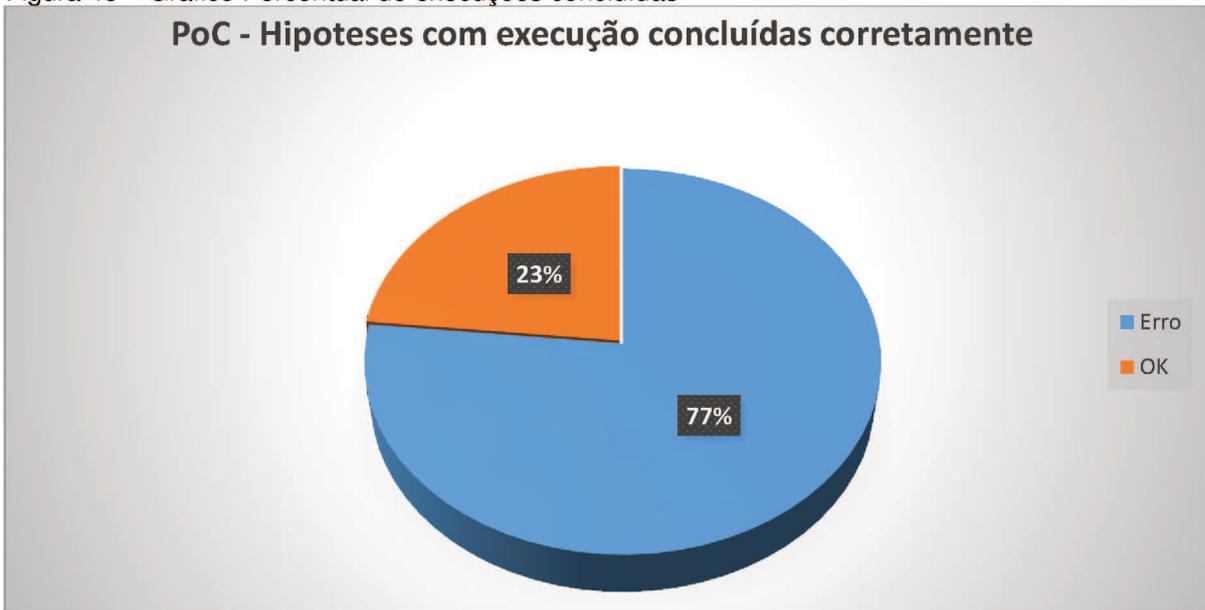
O gráfico de pizza da figura 48 apresenta um resumo das execuções que foram salvas na blockchain, ou seja, 11 transações representando 23% das hipóteses definidas para o teste tiveram êxito em gerar registros, e as outras 36 representando 77% não satisfizeram as regras dos contratos inteligentes.

Figura 47 – Gráfico demonstração dos resultados da PoC



Fonte: Autor

Figura 48 – Gráfico Percentual de execuções concluídas



Fonte: Autor

Tabela 7 – Resumo das transações que obtiveram sucesso de execução

Transação			Participantes		Valores		Saldos das Carteiras CriptoLata			
ID	Transação	AtivoVenda	vendedorID	compradorID	qtdEmbalagens	Creditos	Gestor	P1	P2	P3
2	VendaIndustria	NF1	P1	P2	100	-	100	-	-	-
4	VendaIndustria	NF2	P1	P2	50	-	150	-	-	-
18	VendaConsumidor	NF6	P2	P3	10	-	150	-	-	-
35	Devolução	NF9	P3	P2		1	149	-	-	1
40	VendaConsumidor	NF11	P2	P3	3	1	149	-	1	-
42	VendaIndustria	NF21	P1	P2	10	1	159	-	-	-
43	VendaConsumidor	NF22	P2	P3	9	-	159	-	-	-
44	Devolução	NF23	P3	P2		1	158	-	-	1
45	Devolução	NF24	P3	P2		2	156	-	-	3
46	VendaConsumidor	NF25	P2	P3	4	2	156	-	2	1
47	VendaIndustria	NF26	P1	P2	2	1	158	-	1	1

Fonte: Autor

7 CONSIDERAÇÕES FINAIS

O intuito desta pesquisa foi contextualizar o tema Logística Reversa, devido ao apelo ecológico que estamos vivenciando. Visto que o Brasil possui uma lei que rege este tema desde a publicação da lei 12.305 de agosto de 2010, onde foi criado a PNRS que instituiu responsabilidade compartilhada entre o poder público, fabricantes, importadores, distribuidores, comerciantes e os consumidores, na implantação do SLR.

Este trabalho buscou conceituar e aplicar uma das Tecnologias da Informação e Comunicação mais atuais, as DLTs, utilizando o *Blockchain* como base para a construção de uma PoC para a implantação de um SLR para embalagens de aço.

Durante os estudos foram apontadas as vantagens da aplicação da tecnologia *Blockchain* que a credenciam para o desenvolvimento do protótipo para construção de um SLR, como: a eliminação da intermediação de terceiros no processo, a imutabilidade dos dados e a distribuição das informações, garantindo acesso e segurança as informações transacionadas.

Para o desenvolvimento da pesquisa e no processo de especificação da PoC, utilizou-se o método DSR, apoiado na modelagem UML para conceituar formalmente a metodologia aplicada na proposição do SLR, através do diagrama de atividades. Para engajar todos os atores neste SLR foi proposto a criação de um Ativo denominado CriptoLata, que traz uma simplificação enorme nos controles das embalagens movimentadas no processo, além de servir de incentivo para que os envolvidos neste fluxo tenham interesse em seu perfeito funcionamento, pois agrega impacto financeiro nas transações.

Para o desenvolvimento do contrato inteligente, utilizou-se o *Hyperledger Composer Playground*, que permitiu de forma interativa a criação e teste da rede de negócio para atender as especificações predefinidas. Utilizando a linguagem *JavaScript* todos os participantes e ativos da rede foram criados, além da programação das regras necessárias para tornar o contrato inteligente consistente e seguro.

Para a validação do contrato, foi utilizado o método de Teste Funcional, onde na preparação dos dados para os testes, buscou-se contemplar todas as possibilidades de transações conforme o diagrama da figura 31, gerando transações

válidas e inválidas a fim de testar todas as regras do contrato. Os testes foram realizados manualmente no *Composer*, tendo seus resultados registrados em planilha que posteriormente à sua execução foram comparados aos resultados esperados. Foi obtido um índice de 100% de assertividade na validação, assegurando que a PoC atendeu ao seu propósito, que era de suportar um novo modelo de logística reversa, utilizando a tecnologia de contabilidade distribuída *Blockchain*.

Refletindo sobre a pesquisa, pode surgir alguns trabalhos futuros: 1) Ampliar a utilização das criptoLatas para os demais processos descritos no diagrama; 2) Implementar a valorização do criptoLata para ser tratado como uma criptomoeda real; 3) Estudar a possibilidade da criptoLata ter sua aplicação fora do fluxo do SLR, introduzindo-a no mercado de consumo na forma de um sistema de fidelização ou pontuação para troca de produtos; 4) Utilizar outras DLTs para comparar a facilidade da aplicação num SLR; 5) por último implementar este conceito numa aplicação real;

REFERÊNCIAS

- ABAL. **Sustentabilidade: Reciclagem**. 20 set. 2018. Disponível em: <<http://abal.org.br/sustentabilidade/reciclagem/fluxo-da-reciclagem/>>.
- AMAZON. **O que é o Amazon EC2?** 24 out. 2018. Disponível em: <https://docs.aws.amazon.com/pt_br/AWSEC2/latest/UserGuide/concepts.html>.
- BANERJEE, A. **Blockchain Technology: Supply Chain Insights from ERP** (P. Raj, G. C. Deka, Eds.) **Advances in Computers** Academic Press Inc., , 2018. . Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85046861214&doi=10.1016%2Fbs.adcom.2018.03.007&partnerID=40&md5=15c184f5607b2fce5ec5f46a3ce2178e>>.
- BLOCKGEEKS. **What Are Enterprise Blockchains?** 16 ago. 2018. Disponível em: <<https://blockgeeks.com/guides/enterprise-blockchains/>>.
- BRANCO, P. **Blockchain: a “Xerox” da DLT**. 25 set. 2018. Disponível em: <<https://www.smartpaymentsnews.com/seguranca/blockchain-a-xerox-da-dlt/>>.
- BRASIL. **LEI Nº 12.305, DE 2 DE AGOSTO DE 2010**. 15 set. 2018. Disponível em: <<http://www2.mma.gov.br/port/conama/legiabre.cfm?codlegi=636>>.
- BRASIL. **DECRETO Nº 7.404, DE 23 DE DEZEMBRO DE 2010**. 17 set. 2018. Disponível em: <<http://www2.mma.gov.br/port/conama/legiabre.cfm?codlegi=636>>.
- BRASIL. **RESOLUÇÃO Nº 151, DE 30 DE MAIO DE 2019**. Disponível em: <<http://www.in.gov.br/web/dou/-/resolucao-n-151-de-30-de-maio-de-2019-163675789>>.
- CHRISTIDIS, K.; DEVETSIKIOTIS, M. Blockchains and Smart Contracts for the Internet of Things. **IEEE Access**, v. 4, p. 2292–2303, 2016.
- CONSULTING, G. **Por que é importante fazer uma prova de conceito? (PoC)**. 17 set. 2018. Disponível em: <<https://gaea.com.br/por-que-e-importante-fazer-uma-prova-de-conceito/>>.
- COUTO, M. C. L.; LANGE, L. C. Análise dos sistemas de logística reversa no Brasil. **Engenharia Sanitaria e Ambiental**, v. 22, p. 889–898, 2017. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1413-41522017000500889&nrm=iso>.
- CREIMER, M. **BLOCKCHAIN, SIDECHAIN E OFF-CHAIN**. 21 out. 2018. Disponível em: <<https://www.blockmaster.com.br/artigos/blockchain-sidechain-e-off-chain/>>.
- Docs Hyperledger Composer**. 8 mar 2019. Disponível em: <<https://hyperledger.github.io/composer/latest/introduction/introduction.html>>.
- FERENHOF, H. A.; FERNANDES, R. F. Desmistificando a revisão de literatura como

base para redação científica: método SFFDEMYSTIFYING THE LITERATURE REVIEW AS BASIS FOR SCIENTIFIC WRITING: SSF METHOD. **Revista ABC**, v. v. 21, n., p. 550–563, 2016. Disponível em: <<https://revistaacb.emnuvens.com.br/racb/rt/captureCite/1194/0/AbntCitationPlugin>>.

FLEISCHMANN, M. et al. Quantitative models for reverse logistics: A review. **European Journal of Operational Research**, v. 103, n. 1, p. 1–17, 1997. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-0031277045&doi=10.1016%2FS0377-2217%2897%2900230-0&partnerID=40&md5=2b0e5cf294368c0e683288a366d0f05e>>.

FLOREA, B. C. Blockchain and Internet of Things data provider for smart applications. In: 2018 7th Mediterranean Conference on Embedded Computing (MECO), **Anais...**2018.

Gráfico do Histórico de Preço do Bitcoin. 8-3-2018. Disponível em: <<https://www.buybitcoinworldwide.com/pt-br/preco/>>.

GREVE, F. et al. **Blockchain e a Revolução do Consenso sob Demanda** **Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (XXXVI SBRC 2018)** Campos do Jordão - SP, 2018. . Disponível em: <<http://www.sbrc2018.ufscar.br/wp-content/uploads/2018/04/Capitulo5.pdf>>.

GROMOV, G.; LAMMI, M. Blockchain and Internet of Things Require Innovative Approach to Logistics Education. **Transport Problems**, v. 12, n. SpecialEdition, p. 23–34, 2017. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85041897643&doi=10.20858%2Ftp.2017.12.se.2&partnerID=40&md5=2806ca545753c72a98df498f8e638c4e>>.

HERNÁNDEZ, C. T.; MARINS, F. A. S.; CASTRO, R. C. Modelo de Gerenciamento da Logística Reversa / Reverse Logistics Management Model. **Gestão & Produção**, v. 19, n. 3, p. 445–456, 2012. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0104-530X2012000300001&lng=en&tlng=en>.

HEVNER, A.; CHATTERJEE, S. Design Research in Information Systems. v. 22, p. 9–22, 2010. Disponível em: <<http://link.springer.com/10.1007/978-1-4419-5653-8>>.

HYPERLEDGER. **Hyperledger-Fabricdocs Documentation**. 28 Fev. 2019. Disponível em: <<https://hyperledger.github.io/composer/latest/introduction/introduction.html>>.

KOSBA, A. et al. Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts. In: 2016 IEEE Symposium on Security and Privacy (SP), **Anais...**2016.

LAMOUNIER, L. **Algoritmos de Consenso: A Raiz Que Sustenta a Tecnologia Blockchain** . 7 Fev 2019. Disponível em: <<https://101blockchains.com/pt/algoritmos-de-consenso/#1>>.

LANKO, A.; VATIN, N.; KAKLAUSKAS, A. Application of RFID combined with blockchain technology in logistics of construction materials. (I. Ilin, O. Kalinina, Eds.) In: **Anais...EDP Sciences**, 2018. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85049454045&doi=10.1051%2Fmateconf%2F201817003032&partnerID=40&md5=1c4e6791318b6e718842171a71debf16>>.

MASTERDAWEB. **Ponteiros – Linguagem C/C++**. 04 fev 2019. Disponível em: <<https://blog.masterdaweb.com/programacao-1/linguagem-c/ponteiros-linguagem-cc/>>.

MELO, M. T.; NICKEL, S.; SALDANHA-DA-GAMA, F. Facility location and supply chain management - A review. **European Journal of Operational Research**, v. 196, n. 2, p. 401–412, 2009. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-58149242603&doi=10.1016%2Fj.ejor.2008.05.007&partnerID=40&md5=0dc835560937abbb4c2275cfd8ea7bb1>>.

MICROSOFT. **Hyperledger Fabric consortium network**. 29 jan 2019. Disponível em: <<https://docs.microsoft.com/pt-br/azure/blockchain/templates/hyperledger-fabric-consortium-blockchain>>.

MINOVES, J. S. et al. LOGÍSTICA DIRETA E LOGÍSTICA REVERSA NA PRODUÇÃO DO AÇO: ESTUDO DE CASO EM UMA EMPRESA SIDERÚRGICA. **Revista Inovação, Projetos e Tecnologias – IPTEC**, v. 3, p. 15, 2015.

MMA, M. do M. A. **ACORDO SETORIAL PARA IMPLANTAÇÃO DO SISTEMA DE LOGÍSTICA REVERSA DE EMBALAGENS EM GERAL** (M. do M. Ambiente, Ed.)COALIZÃO, , 2015. .

MMA, M. do M. A. **TERMO DE COMPROMISSO PARA IMPLANTAÇÃO DO SISTEMA DE LOGÍSTICA REVERSA DE EMBALAGENS DE AÇO** (M. do M. Ambiente, Ed.)PROLATA, , 2018. .

NAKAMURA, E. T.; GEUS, P. L. de. **Segurança de Redes em Ambientes Cooperativos**. São Paulo: Novatec, 2007.

NARAYANAN, A. et al. **Bitcoin and Cryptocurrency Technologies**. [s.l.] Princeton University, 2016.

NETO, A. Introdução a Teste de Software. **Engenharia de Software Magazine**, n. May, p. 54–59, 2007.

O que é hashing? 10-01–2019. Disponível em: <<https://curso.coinhousebrasil.com.br/2018/12/30/o-que-e-o-hashing/>>.

PARDO, J. de S. et al. **Logística Reversa**. 20 set. 2018. Disponível em: <<http://www.administradores.com.br/artigos/economia-e-financas/logistica-reversa-um-interesse-crescente/21853/>>.

PPGTIC. **Linhas de Pesquisa do Programa de Pós-Graduação em Tecnologias da Informação e Comunicação**. 15 abr. 2018. Disponível em: <<http://ppgtic.ufsc.br/linhas-de-pesquisa>>.

RIBEIRO, D. **Firmware**. 18 out. 2018. Disponível em: <<https://www.techtudo.com.br/dicas-e-tutoriais/noticia/2014/04/firmware-o-que-e-e-como-atualizar.html>>.

RON, D.; SHAMIR, A. **Quantitative Analysis of the Full Bitcoin Transaction Graph** *Financial Cryptography and Data Security - 17th International Conference, FC 2013* Okinawa, Japan, 2013. .

SCHUEFFEL, P. **Alternative Distributed Ledger Technologies Blockchain vs. Tangle vs. Hashgraph - A High-Level Overview and Comparison**. [s.l.: s.n.]

SIGNIFICADOS. **Significado de Logística**. 4 abr 2019. Disponível em: <<https://www.significados.com.br/logistica/>>.

SINIR. **Gestão de Resíduos Sólidos no Brasil**. 02 março 2019. Disponível em: <<http://sinir.gov.br/>>.

SRIVASTAVA, S. K. Green supply-chain management: A state-of-the-art literature review. **International Journal of Management Reviews**, v. 9, n. 1, p. 53–80, 2007. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-33847365367&doi=10.1111%2Fj.1468-2370.2007.00202.x&partnerID=40&md5=877d7c5c0e380c7b80222c0b58c3455e>>.

UPADHYAYA, V.; SHARMA, M.; ARUN, A. **Think Blockchain**. [s.l.: s.n.]

VANIN, F. **Blockchain: uma cronologia**. 29 jan 2019. Disponível em: <<https://blockchainacademy.com.br/blockchain-uma-cronologia/>>.

VENTURA, P. **Entendendo o Diagrama de Atividades da UML**. 10 abr 2019. Disponível em: <<https://www.ateomomento.com.br/uml-diagrama-de-atividades/>>.

WALDMAN, J. **Blockchain – Conceitos básicos do Blockchain**. 1 fev 2019. Disponível em: <<https://msdn.microsoft.com/pt-br/magazine/mt845650.aspx>>.

WIKIPEDIA. **Infraestrutura de chave pública**. 1 mar 2019. Disponível em: <https://en.wikipedia.org/wiki/Public_key_infrastructure>.

ZYSKIND, G.; NATHAN, O.; PENTLAND, A. ' . Decentralizing Privacy: Using Blockchain to Protect Personal Data. In: 2015 IEEE Security and Privacy Workshops, **Anais...**2015.

APÊNDICE A – Arquivos de definição da Blockchain

```

/*
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
 * implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

/**
 * Definição do namespace
 */
namespace org.mestrado.poccriptolata

// DEFINE OS PARTICIPANTES

concept Endereco {
  o String cidade
  o String uf
}

abstract participant Pessoa identified by codigold {
  o String codigold
  o String name
  o Endereco endereco optional
}

enum TipoParticipante {
  o INDUSTRIA
  o LOJISTA
  o CONSUMIDOR
  o CATADORES
  o SIDERURGICA
  o ATERRO
  o GOVERNO
  o GESTOR
}

```

```
participant Participante extends Pessoa {
  o TipoParticipante tipo
}

// DEFINE OS ATIVOS

asset CarteiraCriptoLata identified by criptold {
  o String criptold
  --> Participante titular
  o Integer saldo
}

asset AtivoVenda identified by vendald {
  o String vendald
  o TipoNota tipoOperacao
  --> Participante vendedorId
  --> Participante compradorId
  o Integer qtdEmbalagens
  o Integer credito optional
}

// Cria dados para Teste da Blockchain

// DEFINE AS TRANSAÇÕES
enum TipoNota {
  o VENDA_INDUSTRIA
  o VENDA_CONSUMIDOR
  o DEVOLUCAO
}

transaction VendaIndustria {
  o String vendald
  --> Participante vendedorId
  --> Participante compradorId
  o Integer qtdEmbalagens
  o Integer credito
}

transaction VendaConsumidor {
  o String vendald
  --> Participante vendedorId
  --> Participante compradorId
  o Integer qtdEmbalagens
  o Integer credito
}

transaction Devolucao {
  o String vendald
```

```
--> Participante consumidorId
--> Participante lojistaId
o Integer credito
}

transaction ConsultaVendas {
}

event ConsultaVendasEvent {
o String[] cLinha
}
event MensagemEvent {
o String cMensagem
}

transaction CargaInicial {
}

transaction LimpaMovimentos {
}
```

APÊNDICE B – Script Carga Inicial

```

/**
 * Setup the demo
 * @param {org.mestrado.poccriptolata.CargalInicial} CargalInicial - the SetupDemo
transaction
 * @transaction
 */

async function CargalInicial(setupDemo) {

// Carga nos participantes

console.log('Carga Inicial....');

const factory = getFactory();
const namespace = 'org.mestrado.poccriptolata';

const vPart = { '1': [{ 'codigold': 'gestor', 'name': 'Gestor-Programa CritoLata',
'tipo': 'GESTOR' }],
                '2': [{ 'codigold': 'p1', 'name': 'Reginaldo', 'tipo': 'INDUSTRIA' }],
                '3': [{ 'codigold': 'p2', 'name': 'Michele', 'tipo': 'LOJISTA' } ],
                '4': [{ 'codigold': 'p3', 'name': 'Marina', 'tipo': 'LOJISTA' } ],
                '5': [{ 'codigold': 'p4', 'name': 'Vinicius', 'tipo': 'CONSUMIDOR' } ] };

const vp = [];
let partCount = 0;
for (const IdPart in vPart) {
const vParticipantes = vPart[IdPart];
for (let i = 0; i < vParticipantes.length; i++) {

const partTemplate = vParticipantes[i];
const participante = factory.newResource(namespace, 'Participante',
partTemplate.codigold);

participanteExists = await getParticipantRegistry(namespace + '.Participante')
.then(function (participantRegistry) {
return participantRegistry.exists(partTemplate.codigold);
})
.then(function (exists) {
return exists;
});

if (participanteExists == true) {
console.log(vParticipantes[i].name + ' ja registrado ...')
} else {

```

```
console.log('Abre Verificacao - ' + partTemplate.codigold);

console.log('fecha teste = ' + vParticipantes[i].name);

participante.name = partTemplate.name;
participante.tipo = partTemplate.tipo;

vp.push(participante);
partCount++;

}

}

}

const participantesRegistry = await getParticipantRegistry(namespace +
'.Participante');
await participantesRegistry.addAll(vp);

//Cadastrando uma nova carteira para o GESTOR
const carteira = factory.newResource(namespace, 'CarteiraCriptoLata', 'gestor');
carteira.titular = factory.newRelationship(namespace, 'Participante', 'gestor');
carteira.saldo = 0;
const assetRegistry = await getAssetRegistry(carteira.getFullyQualifiedType());
await assetRegistry.add(carteira);

console.log('Termino....');
}
```

APÊNDICE C – *Script* Limpa Movimento

```
/**
 * Setup the demo
 * @param {org.mestrado.poccriptolata.LimpaMovimentos} Limpa os ativos Vendas
 * @transaction
 */

async function LimpaMovimentos() {

  const factory = getFactory();
  const namespace = 'org.mestrado.poccriptolata';

  var results = await query('selectVendas');
  console.log('Apagando '+results.length+' registros do asset AtivoVendas');
  for (let n = 0; n < results.length; n++) {
    let venda = results[n];

    await getAssetRegistry(namespace+'.AtivoVenda')
    .then(function (_AssetRegistry) {
      return _AssetRegistry.remove(venda.vendald);
    });
  }

  var results = await query('selectCarteiras');
  console.log('Apagando '+results.length+' registros do asset CarteiraCriptoLata');
  for (let n = 0; n < results.length; n++) {
    let carteira = results[n];

    await getAssetRegistry(namespace+'.CarteiraCriptoLata')
    .then(function (_AssetRegistry) {
      return _AssetRegistry.remove(carteira.criptold);
    });
  }
}
```

APÊNDICE D – Script VendaIndustria

```

/**
 * Cria os participantes
 * @param {org.mestrado.poccriptolata.VendaIndustria} Venda - Registra a venda e
 * cria o ativo criptoLata
 * @transaction
 */

async function VendaIndustria(p_venda) { // eslint-disable-line no-unused-vars

  console.log('=====  

  console.log('<<< Venda Industria para Lojista >>>');

  const factory = getFactory();
  const namespace = 'org.mestrado.poccriptolata';
  var incluiVenda = false;
  var saldoVendedor = 0;
  var saldoTransfer= 0;
  var saldoCriar= 0;

  if ((p_venda.vendedorId.tipo=='INDUSTRIA') &&
  (p_venda.compradorId.tipo=='LOJISTA')) {

  if (p_venda.qtdEmbalagens <= 0) {
    console.log('venda nao realizada! Qtd embalagem deve ser maior que zero!!!');
    throw new Error('Venda nao realizada! Embalagem ou credito devem ser
    maior que zero!!!');
  }

  if ((p_venda.qtdEmbalagens > 0) || (p_venda.credito >= 0)) {

  //Verifica se existe a carteira de CriptoLatas - Gestor
  var CarteiraGestorExists = await getAssetRegistry(namespace +
  '.CarteiraCriptoLata')
    .then(function (carteiraRegistryGestor) {
      return carteiraRegistryGestor.exists('gestor');
    })
    .then(function (exists) {
      return exists;
    });

  //Verifica se existe a carteira de CriptoLatas - Vendedor
  var CarteiraVendedorExists = await getAssetRegistry(namespace +
  '.CarteiraCriptoLata')
    .then(function (carteiraRegistryVendedor) {
      return carteiraRegistryVendedor.exists(p_venda.vendedorId.codigold);
    });

```

```

    })
    .then(function (exists) {
      return exists;
    });

    //Verifica se existe a carteira de CriptoLatas - Comprador
    var CarteiraCompradorExists = await getAssetRegistry(namespace +
'.CarteiraCriptoLata')
    .then(function (carteiraRegistryComprador) {
      return carteiraRegistryComprador.exists(p_venda.compradorId.codigold);
    })
    .then(function (exists) {
      return exists;
    });

    if (p_venda.credito>0) {
    if (CarteiraCompradorExists == true) {
      //Localiza Vendedor e carrega saldo
      saldoComprador = await getAssetRegistry(namespace +
'.CarteiraCriptoLata')
      .then(function (carteiraRegistryComprador) {
        return carteiraRegistryComprador.get(p_venda.compradorId.codigold);
      })
      .then(function (_type) {
        return _type.saldo;
      });

      console.log('...saldoComprador...' + saldoComprador)

      if (saldoComprador >= p_venda.credito) {

        console.log( '...Movimenta a Carteira Vendedor
('+p_venda.vendedorId.codigold+') em +' + p_venda.credito);

        //Verifica se existe a carteira de CriptoLatas
        var saldoTransfer = p_venda.credito;
        console.log( '...Movimente a Carteira Comprador
('+p_venda.compradorId.codigold+') em -' + p_venda.credito);

        //Verifica se existe a carteira de CriptoLatas - Comprador

        let _assetTypeComp;
        await getAssetRegistry(namespace + '.CarteiraCriptoLata')
        .then(function(_type) {
          _assetTypeComp = _type;
          return _type.get(p_venda.compradorId.codigold);
        })
        .then(function(_wallet){

```

```

        if (_wallet){
            _wallet.saldo -= p_venda.credito;
            _assetTypeComp.update(_wallet);
        }
    });

    //Define que transacao sera incluida
    incluiVenda = true;

    }else{
        console.log('venda nao realizada! Comprador com saldo
insuficiente!!!');
        console.log('Qtd Venda= '+p_venda.qtdEmbalagens);
        console.log('Saldo Comprador = '+saldoComprador);
        throw new Error('Venda nao realizada! Comprador com saldo
insuficiente!!! Qtd Venda= '+p_venda.qtdEmbalagens+' Saldo Comprador =
'+saldoComprador);
    }
    }else{
        incluiVenda = false;
        console.log('venda nao realizada! Carteira do Comprador nao
encontrada!!!');
        throw new Error('Venda nao realizada! Carteira do Comprador nao
encontrada!!!');
    }
    }else{

// Inclui um AtivoVenda ( Venda sem utilizar creditos )
        incluiVenda = true;
    }
    } else {
        console.log('...Transacao nao realizada... valores negativos...');
        throw new Error('...Transacao nao realizada... valores negativos...');
    }
    if (incluiVenda == true) {
        // Inclui um AtivoVenda ( Venda sem utilizar creditos )
        if (p_venda.qtdEmbalagens > 0) {
            console.log('...Criado a transacao VendaIndustria sem credito de
CriptoLata...'+p_venda.vendaId);
            const venda = factory.newResource(namespace, 'AtivoVenda',
p_venda.vendaId);
            venda.tipoOperacao = "VENDA_INDUSTRIA";
            venda.vendedorId = factory.newRelationship(namespace, 'Participante',
p_venda.vendedorId.getIdentifier());
            venda.compradorId = factory.newRelationship(namespace, 'Participante',
p_venda.compradorId.getIdentifier());
            venda.qtdEmbalagens = p_venda.qtdEmbalagens;
            venda.credito = p_venda.credito;

```

```

        const assetRegistry = await
getAssetRegistry(venda.getFullyQualifiedType());
        await assetRegistry.add(venda);
    }
    //Trata o credito que o vededer tem para abater da geracao de novas
criptoLatas

    if (CarteiraVendedorExists == false) {
        var saldoCriar = p_venda.qtdEmbalagens;
        if (saldoVendedor >= 0) {
            if (saldoVendedor >= saldoCriar) {
                saldoVendedor -= saldoCriar;
                saldoCriar = 0;
            }else{
                saldoCriar -= saldoVendedor;
                saldoVendedor = 0;
            }
        }
        //console.log('saldoVendedor='+saldoVendedor);
        //console.log('saldoCriar='+saldoCriar);

        //Cadastrando uma nova carteira para o Participante
        console.log( '...Carteira Vendedor - Criada
...' +p_venda.vendedorId.codigold);
        const carteira = factory.newResource(namespace, 'CarteiraCriptoLata',
p_venda.vendedorId.getIdentifier());
        carteira.titular = factory.newRelationship(namespace, 'Participante',
p_venda.vendedorId.getIdentifier());
        carteira.saldo = saldoVendedor;
        const assetRegistry = await
getAssetRegistry(carteira.getFullyQualifiedType());
        await assetRegistry.add(carteira);

    }else{

        var saldoCriar = p_venda.qtdEmbalagens;
        console.log('CarteiraVendedorExists <'+CarteiraVendedorExists+'>');
        console.log('p_venda.credito <'+p_venda.credito+'>');
        if (CarteiraVendedorExists == true) {
            saldoVendedor = await getAssetRegistry(namespace +
'.CarteiraCriptoLata')
                .then(function (carteiraRegistryVendedor) {
                    return carteiraRegistryVendedor.get(p_venda.vendedorId.codigold);
                })
                .then(function (_type) {
                    return _type.saldo;
                });
        }
    }
}

```

```

    }else{
        console.log('... Carteira Vendedor inexistente... e credito > 0 ...');
        console.log('saldoVendedor <'+saldoVendedor+'>');
        throw new Error('... Carteira Vendedor inexistente... e credito > 0 ...
SaldoVendedor =' +saldoVendedor);
    }
    saldoVendedor += saldoTransfer;
    //console.log('saldoVendedor=' +saldoVendedor);
    //console.log('saldoCriar=' +saldoCriar);
    if (saldoVendedor >= 0) {
        if (saldoVendedor >= saldoCriar) {
            saldoVendedor -= saldoCriar;
            saldoCriar = 0;
        }else{
            saldoCriar -= saldoVendedor;
            saldoVendedor = 0;
        }
    }
    //console.log('saldoVendedor=' +saldoVendedor);
    //console.log('saldoCriar=' +saldoCriar);
    //Verifica se existe a carteira de CriptoLatas
    console.log('p_venda.vendedorId.codigold
<' +p_venda.vendedorId.codigold+'>');
    let _assetTypeVend;
    await getAssetRegistry(namespace + '.CarteiraCriptoLata')
        .then(function(_type) {
            _assetTypeVend = _type;
            return _type.get(p_venda.vendedorId.codigold);
        })
        .then(function(_wallet){
            if (_wallet){
                _wallet.saldo = saldoVendedor;
                console.log('...Saldo vendedor Gravado... ' +saldoVendedor);
                _assetTypeVend.update(_wallet);
            }
        });
}

if (CarteiraGestorExists == false) {
    //Cadastrando uma nova carteira para o GESTOR
    console.log('...Criado a carteira do gestor...');
    const carteira = factory.newResource(namespace, 'CarteiraCriptoLata',
'gestor');
    carteira.titular = factory.newRelationship(namespace, 'Participante', 'gestor');
    carteira.saldo = saldoCriar;
    const _assetRegistry = await
getAssetRegistry(carteira.getFullyQualifiedType());
    await _assetRegistry.add(carteira);
}

```

```

    } else {

//      if (CarteiraGestorExists == true) {
        console.log('...Carrega Carteira do Gestor - VendaIndustria ');

        var _assetTypeGestor;
        await getAssetRegistry(namespace + '.CarteiraCriptoLata')
            .then(function(_type) {
                _assetTypeGestor = _type;
                return _type.get('gestor');
            })
            .then(function(_wallet){
                if (_wallet){
                    _wallet.saldo += saldoCriar;
                    _assetTypeGestor.update(_wallet);
                }
            });

        console.log( '...Movimente a Carteira Vendedor
(+p_venda.vendedorId.codigold+) em '+p_venda.credito);

    }

}
} else{
    console.log('venda nao realizada! Vendedor nao e INDUSTRIA ou Comprador
nao e LOJISTA!!!');
    console.log('tipo vendedor = (+p_venda.vendedorId.codigold+)
'+p_venda.vendedorId.tipo);
    console.log('tipo comprador = (+p_venda.compradorId.codigold+)
'+p_venda.compradorId.tipo);
    throw new Error('1-Venda nao realizada! Vendedor nao e INDUSTRIA ou
Comprador nao e LOJISTA!!! Vendedor = (+p_venda.vendedorId.codigold+)
'+p_venda.vendedorId.tipo+' Comprador = (+p_venda.compradorId.codigold+)
'+p_venda.compradorId.tipo);
}
}

```

APÊNDICE E – *Script VendaConsumidor*

```

/**
 * Cria os participantes
 * @param {org.mestrado.poccriptolata.VendaConsumidor} Venda - Registra a venda
 e cria o ativo criptoLata
 * @transaction
 */

    async function VendaConsumidor(p_venda) {

console.log('====');
console.log('<<< Venda Lojista para Consumidor >>>');

const factory = getFactory();
const namespace = 'org.mestrado.poccriptolata';
var incluiVenda = false;
var saldoVendedor = 0;
var saldoTransfer= 0;
var saldoCriar= 0;
var MsgTrans= "sem credito";

if ((p_venda.vendedorId.tipo=='LOJISTA') &&
(p_venda.compradorId.tipo=='CONSUMIDOR')) {

    if ((p_venda.qtdEmbalagens <= 0) && (p_venda.credito <= 0)) {
        console.log('venda nao realizada! Embalagem ou credito devem ser maior
que zero!!!');

        throw new Error('Venda nao realizada! Embalagem ou credito devem ser maior
que zero!!!');
    }

    if ((p_venda.qtdEmbalagens > 0) || (p_venda.credito >= 0)) {

        //Verifica se existe a carteira de CriptoLatas - Vendedor
        var CarteiraVendedorExists = await getAssetRegistry(namespace +
'.CarteiraCriptoLata')
        .then(function (carteiraRegistryVendedor) {
            return carteiraRegistryVendedor.exists(p_venda.vendedorId.codigold);
        })
        .then(function (exists) {
            return exists;
        }) ;

        //Verifica se existe a carteira de CriptoLatas - Comprador

```

```

var CarteiraCompradorExists = await getAssetRegistry(namespace +
'.CarteiraCriptoLata')
.then(function (carteiraRegistryComprador) {
  return carteiraRegistryComprador.exists(p_venda.compradorId.codigold);
})
.then(function (exists) {
  return exists;
});

if (p_venda.credito>0) {
  if (CarteiraCompradorExists == true) {
    //Localiza Vendedor e carrega saldo
    saldoComprador = await getAssetRegistry(namespace + '.CarteiraCriptoLata')
      .then(function (carteiraRegistryComprador) {
        return carteiraRegistryComprador.get(p_venda.compradorId.codigold);
      })
      .then(function (_type) {
        return _type.saldo;
      });

    console.log('...saldoCosumidor...' + saldoComprador)

    if (saldoComprador >= p_venda.credito) {

      //Verifica se existe a carteira de CriptoLatas
      var saldoTransfer = p_venda.credito;
      console.log( '...Movimente a Carteira Consumidor
(+p_venda.compradorId.codigold+) em -'+p_venda.credito);

      //Verifica se existe a carteira de CriptoLatas - Comprador

      let _assetTypeComp;
      await getAssetRegistry(namespace + '.CarteiraCriptoLata')
        .then(function(_type) {
          _assetTypeComp = _type;
          return _type.get(p_venda.compradorId.codigold);
        })
        .then(function(_wallet){
          if (_wallet){
            _wallet.saldo -= p_venda.credito;
            _assetTypeComp.update(_wallet);
          }
        });

      //Cadastrando uma nova carteira para o Participante
      if (CarteiraVendedorExists == false) {
        console.log( '...Carteira Lojista - Criada ...'+p_venda.vendedorId.codigold);

```

```

        const carteira = factory.newResource(namespace, 'CarteiraCriptoLata',
p_venda.vendedorId.getIdentifier());
        carteira.titular = factory.newRelationship(namespace, 'Participante',
p_venda.vendedorId.getIdentifier());
        carteira.saldo = saldoTransfer;
        const assetRegistry = await
getAssetRegistry(carteira.getFullyQualifiedType());
        await assetRegistry.add(carteira);

    }else{

        console.log( '...Movimenta a Carteira Lojista
('+p_venda.vendedorId.codigold+') em +' +p_venda.credito);

        let _assetTypeVend;
        await getAssetRegistry(namespace + '.CarteiraCriptoLata')
        .then(function(_type) {
            _assetTypeVend = _type;
            return _type.get(p_venda.vendedorId.codigold);
        })
        .then(function(_wallet){
            if (_wallet){
                _wallet.saldo += saldoTransfer;
                console.log('...Saldo Lojista Gravado... '+saldoVendedor);
                _assetTypeVend.update(_wallet);
            }
        });
    }

    //Define que transacao sera incluida
    incluiVenda = true;
    MsgTrans= "com Credito";

}

}else{
    console.log('venda nao realizada! Consumidor com saldo insuficiente!!!');
    console.log('Qtd Venda= '+p_venda.qtdEmbalagens);
    console.log('Saldo Consumidor = '+saldoComprador);
    throw new Error('Venda nao realizada! Consumidor com saldo insuficiente!!!
Qtd Venda='+p_venda.qtdEmbalagens+' Saldo Consumidor='+saldoComprador);
}
}

}else{
    console.log('venda nao realizada! Carteira do Consumidor nao encontrada!!!');
    throw new Error('Venda nao realizada! Carteira do Consumidor nao
encontrada!!!');
}
}

}

}

// Inclui um AtivoVenda ( Venda sem utilizar creditos )

```

```

    incluiVenda = true;
  }
} else {
  console.log('...Transacao nao realizada... valores negativos...');
  throw new Error('...Transacao nao realizada... valores negativos...');
}
if (incluiVenda == true) {
  // Inclui um AtivoVenda ( Venda sem utilizar creditos )
  if (p_venda.qtdEmbalagens > 0) {
    console.log('...Criado a transacao VendaConsumidor '+MsgTrans+' de
CriptoLata...'+p_venda.vendald);
    const venda = factory.newResource(namespace, 'AtivoVenda',
p_venda.vendald);
    venda.tipoOperacao = "VENDA_CONSUMIDOR";
    venda.vendedorId = factory.newRelationship(namespace, 'Participante',
p_venda.vendedorId.getIdentifier());
    venda.compradorId = factory.newRelationship(namespace, 'Participante',
p_venda.compradorId.getIdentifier());
    venda.qtdEmbalagens = p_venda.qtdEmbalagens;
    venda.credito = p_venda.credito;
    const assetRegistry = await getAssetRegistry(venda.getFullyQualifiedType());
    await assetRegistry.add(venda);

    //Trata o credito que o vededer tem para abater da geracao de novas
criptoLatas

  }else{
    console.log('...venda nao realizada! Qtd menor ou iggual a zero...');
    throw new Error('...venda nao realizada! Qtd menor ou iggual a zero...');
  }
  }else{
    console.log('venda nao realizada!!!!');
    throw new Error('venda nao realizada!!!!');
  }
}
}else{
  console.log('venda nao realizada! Vendedor nao e LOJISTA ou Comprador nao e
CONSUMIDOR!!!!');
  console.log('tipo lojista = ('+p_venda.vendedorId.codigold+')
'+p_venda.vendedorId.tipo);
  console.log('tipo consumidor = ('+p_venda.compradorId.codigold+')
'+p_venda.compradorId.tipo);
  throw new Error('venda nao realizada! Vendedor nao e LOJISTA ou Comprador
nao e CONSUMIDOR!!! Lojista = ('+p_venda.vendedorId.codigold+')
'+p_venda.vendedorId.tipo+' Consumidor = ('+p_venda.compradorId.codigold+')
'+p_venda.compradorId.tipo );
}
}
}

```

APÊNDICE F – Script Devolução

```

/**
 * Cria os participantes
 * @param {org.mestrado.poccriptolata.Devolucao} Devolucao - Registra a venda e
cria o ativo criptoLata
 * @transaction
 */

async function Devolucao(p_devolucao) {
  console.log('====');
  console.log('<<< Devolucao de embalagens - Consumidor para Lojista >>>');

  const factory = getFactory();
  const namespace = 'org.mestrado.poccriptolata';
  var saldoGestor = 0;

  if ((p_devolucao.consumidorId.tipo=='CONSUMIDOR') &&
(p_devolucao.lojistaId.tipo=='LOJISTA')) {

    if (p_devolucao.credito > 0) {

      //Verifica se existe a carteira de CriptoLatas - Gestor
      var CarteiraGestorExists = await getAssetRegistry(namespace +
'.CarteiraCriptoLata')
      .then(function (carteiraRegistryGestor) {
        return carteiraRegistryGestor.exists('gestor');
      })
      .then(function (exists) {
        return exists;
      });
      console.log('CarteiraGestorExists='+CarteiraGestorExists);
      var CarteiraConsumidorExists = await getAssetRegistry(namespace +
'.CarteiraCriptoLata')
      .then(function (carteiraRegistryGestor) {
        return carteiraRegistryGestor.exists(p_devolucao.consumidorId.codigold);
      })
      .then(function (exists) {
        return exists;
      });

      if (CarteiraGestorExists == true) {
        saldoGestor = await getAssetRegistry(namespace + '.CarteiraCriptoLata')
          .then(function (carteiraRegistryGestor) {
            return carteiraRegistryGestor.get('gestor');
          })
          .then(function (_type) {

```

```

    return _type.saldo;
  });

  if (saldoGestor >= p_devolucao.credito) {
    let _assetTypeGestor;
    await getAssetRegistry(namespace + '.CarteiraCriptoLata')
      .then(function(_type) {
        _assetTypeGestor = _type;
        return _type.get('gestor');
      })
      .then(function(_wallet){
        if (_wallet){
          _wallet.saldo -= p_devolucao.credito;
          _assetTypeGestor.update(_wallet);
        }
      });

    if (CarteiraConsumidorExists == false) {
      console.log( '...Carteira Consumidor - Criada
...'+p_devolucao.consumidorId.codigold);
      const carteira = factory.newResource(namespace, 'CarteiraCriptoLata',
p_devolucao.consumidorId.getIdentifier());
      carteira.titular = factory.newRelationship(namespace, 'Participante',
p_devolucao.consumidorId.getIdentifier());
      carteira.saldo = p_devolucao.credito;
      const assetRegistry = await
getAssetRegistry(carteira.getFullyQualifiedType());
      await assetRegistry.add(carteira);

    }else{

      console.log( '...Movimenta a Carteira Consumidor
('+p_devolucao.consumidorId.codigold+') em '+p_devolucao.credito);

      let _assetTypeCons;
      await getAssetRegistry(namespace + '.CarteiraCriptoLata')
        .then(function(_type) {
          _assetTypeCons = _type;
          return _type.get(p_devolucao.consumidorId.codigold);
        })
        .then(function(_wallet){
          if (_wallet){
            _wallet.saldo += p_devolucao.credito;
            console.log('...Saldo Consumidor Gravado... '+p_devolucao.credito);
            _assetTypeCons.update(_wallet);
          }
        });
    }
  }
}

```

```

        // Inclui um AtivoVenda ( Venda sem utilizar creditos )
        console.log('...Criado a transacao Devolucao de embalagens com credito de
CriptoLata...' + p_devolucao.vendald);
        const nota = factory.newResource(namespace, 'AtivoVenda',
p_devolucao.vendald);
        nota.tipoOperacao = "DEVOLUCAO";
        nota.vendedorId = factory.newRelationship(namespace, 'Participante',
p_devolucao.lojistaId.getIdentifier());
        nota.compradorId = factory.newRelationship(namespace, 'Participante',
p_devolucao.consumidorId.getIdentifier());
        nota.qtdEmbalagens = 0;
        nota.credito = p_devolucao.credito;
        const assetRegistry = await getAssetRegistry(nota.getFullyQualifiedType());
        await assetRegistry.add(nota);

    }else{
        console.log("Devolucao nao realizada! Gestor sem saldo de criptoLatas !!!!");
        throw new Error("Devolucao nao realizada! Gestor sem saldo de criptoLatas
!!!!");
    }

    }else{
        console.log("Devolucao nao realizada! Carteira do Gestor nao encontrada !!!!");
        throw new Error("Devolucao nao realizada! Carteira do Gestor nao encontrada
!!!!");
    }

    } else {
        console.log('...Transacao nao realizada... Credito deve ser maior que zero...');
        throw new Error('...Transacao nao realizada... Credito deve ser maior que
zero...');
    }
    }else{
        console.log('venda nao realizada! Vendedor nao e LOJISTA ou Comprador nao e
CONSUMIDOR!!!!');
        console.log('tipo lojista = (' + p_devolucao.lojistaId.codigold + ')
' + p_devolucao.lojistaId.tipo);
        console.log('tipo consumidor = (' + p_devolucao.consumidorId.codigold + ')
' + p_devolucao.consumidorId.tipo);
        throw new Error("Venda nao realizada! Vendedor nao e LOJISTA ou Comprador
nao e CONSUMIDOR!!! Lojista = (' + p_devolucao.lojistaId.codigold + ')
' + p_devolucao.lojistaId.tipo + ' Consumidor =
(' + p_devolucao.consumidorId.codigold + ') ' + p_devolucao.consumidorId.tipo);
    }
}
}

```

APÊNDICE G – Script ConsultaVendas

```
/**
 * Remove all high volume commodities
 * @param {org.mestrado.poccriptolata.ConsultaVendas} Consulta
 * @transaction
 */
async function ConsultaVendas() {

    const factory = getFactory();
    const namespace = 'org.mestrado.poccriptolata';

    // emit the event
    const consultaVendasEvent = factory.newEvent(namespace,
'ConsultaVendasEvent');

    let results = await query('selectVendas');
    const vet = ['vendald | vendedorld | compradorId | qtdEmbalagem'];
    for (let n = 0; n < results.length; n++) {
        let venda = results[n];
vet.push(venda.vendald + ' | '+ venda.vendedorld.getIdentifier() + ' | '+
venda.compradorId.getIdentifier() + ' | '+ venda.qtdEmbalagens );

        console.log(venda.vendald + ' '+ venda.vendedorld.getIdentifier() + ' '+
venda.compradorId.getIdentifier() + ' '+ venda.qtdEmbalagens);

    }
    consultaVendasEvent.cLinha = vet;
    emit(consultaVendasEvent);
}
}
```

APÊNDICE H – Arquivo de *Query*

```
query selectVendas {  
  description: "Lista todas as vendas"  
  statement:  
    SELECT org.mestrado.poccriptolata.AtivoVenda  
}
```