

Thiago Basso

Plataforma +Precoce: Simulador de Sistemas de Produção de Novilho Precoce

Campo Grande - MS

2018

Thiago Basso

Plataforma +Precoce: Simulador de Sistemas de Produção de Novilho Precoce

Dissertação apresentada ao Programa de Mestrado Stricto Sensu em Ciência da Computação Aplicada, mantido pela Universidade Federal do Mato Grosso do Sul, como parte dos requisitos para a obtenção do título de Mestre em Computação Aplicada (Área de Concentração: Engenharia de Software).

Universidade Federal do Mato Grosso do Sul - UFMS

Faculdade de Computação - FACOM

Programa de Pós-Graduação em Ciência da Computação Aplicada

Orientadora: Prof.^a Dr.^a Débora Maria Barroso Paiva (Facom/UFMS)

Coorientadores: Dr. Rodrigo da Costa Gomes (Embrapa Gado de Corte)

Msc. Fernando Rodrigues Teixeira Dias (Embrapa Pantanal)

Campo Grande - MS

2018

*Este trabalho é dedicado a minha filha,
com quem tanto aprendo a cada dia.*

Agradecimentos

Obrigado Deus pelo dom da vida e cuidado de tudo que não está ao meu alcance.

Obrigado meus pais, Ancila e Claudino Basso, pela educação e valores ensinados. Vocês são minha fortaleza.

Obrigado minha esposa, Lisiany Ferreira Azambuja Basso, pela parceria na vida e futuro que sonhamos.

Obrigado minha Maria, filhota querida que é raio de sol em nossas vidas.

Obrigado minha irmã, Karla Cristina Basso Azuma, você é meu exemplo a ser seguido. Ao meu cunhado, Felipe Cazuo Azuma, continue cuidando bem dela.

Obrigado meus sobrinhos, Gigi e Caio, por todos sorrisos e brincadeiras.

Obrigado minha orientadora, Débora Maria Barroso Paiva, pela confiança depositada e dedicação durante nossa jornada.

Obrigado meu coorientador, Fernando Rodrigues Teixeira Dias, pela paciência com minhas limitações e por todos ensinamentos compartilhados.

Obrigado meu coorientador, Rodrigo da Costa Gomes, pelo profissionalismo e ajuda demandada.

Obrigado professora Maria Istela Cagnin pelas colaborações e apontamentos que ajudaram a engrandecer este trabalho.

Obrigado colega Camilo Carromeu por todas explicações, ajuda técnica e sempre me acolher nas visitas à Embrapa Gado de Corte.

Obrigado colegas Cleison Marin, Fábio Noboru Abe e Lucas de Souza Rodrigues por compartilharem morada em Campo Grande.

Obrigado colegas e amigos Tairo Forbat Araujo, Susana Guimarães de Paula, Cristiano Brandão Blans, Gregório Takashi Higashikawa, Leandro Feuser e Emerson Peres por todas contribuições e momentos de descontração.

Obrigado amigos Gabriel Del Vigna Alencar, Tiago Flores Zeferino, Renato Machado Martins e Herbert Lee Barbosa Veríssimo de Barros por sempre estarem presentes.

“Olhem de novo esse ponto. É aqui. É nosso lar. Somos nós. Nele, todos a quem ama, todos a quem conhece, todos de quem já ouviu falar, cada ser humano que já existiu, viveram as suas vidas. O conjunto da nossa alegria e nosso sofrimento, milhares de religiões, ideologias e doutrinas econômicas, cada caçador e coletor, cada herói e covarde, cada criador e destruidor da civilização, cada rei e camponês, cada casal apaixonado, cada mãe e pai, cada criança cheia de esperança, inventores e exploradores, cada professor, cada político corrupto, cada "superestrela", cada "líder supremo", cada santo e pecador na história da nossa espécie viveu ali - em um grão de poeira suspenso em um raio de sol (...)”

(Carl Sagan, Pálido Ponto Azul, 1994)

Resumo

O setor pecuário brasileiro passa por uma evolução em seu modelo de produção, a partir de tecnologias de diversas naturezas. Neste cenário, a adoção de inovações computacionais tem figurado como diferencial para a obtenção de competitividade de mercado, produção de melhor qualidade e melhoria da rentabilidade. No âmbito das Tecnologias da Informação e Comunicação (TICs) os simuladores têm se destacado como ferramentas que permitem ao produtor rural aprimorar seus processos, auxiliando na tomada de decisões gerenciais, desenvolvimento de projetos, avaliação do uso de recursos e do impacto ambiental, dentre outros. Analisando o contexto da utilização de softwares simuladores na área de concentração de sistemas de produção de bovinos de corte, foi possível observar que há muitos desafios e oportunidades e que, apesar de existirem vários softwares para este propósito, ainda existe espaço para novos trabalhos que objetivem principalmente levar esta possibilidade de uso aos produtores e técnicos rurais. Neste sentido, o objetivo deste trabalho foi desenvolver um software capaz de simular sistemas melhorados de produção de gado de corte e gerar indicadores econômicos, físicos e ambientais a partir de parâmetros de produção, reprodução, consumo, compras, rebanho, área, equipe, tempo e emissões. Ao final do desenvolvimento, foi criado um modelo para verificar os cálculos realizados pelo simulador, cujos resultados foram verificados por pesquisadores da Embrapa Gado de Corte e Embrapa Pantanal. Espera-se que a construção desta ferramenta proporcione a produtores rurais e técnicos a possibilidade de simularem sistemas de produção melhorados que possam ser replicados nas propriedades rurais, elevando a produtividade, a qualidade e a sustentabilidade da pecuária de corte.

Palavras-chaves: indicadores, gado de corte, pecuária de precisão, simulação computacional, modelagem computacional.

Abstract

The Brazilian livestock sector undergoes an evolution in its production model, from technologies of diverse natures. In this scenario, the adoption of computational innovations has figured as a differential to obtain market competitiveness, production of better quality and profitability improvement. In the field of Information and Communication Technologies (ICTs) the simulators have stood out as tools that allow farmers to improve their processes, assisting in managerial decision making, project development, resource use and environmental impact assessments, among others. Analyzing the context of the use of simulator software in the beef cattle production systems concentration area, it was possible to observe that there are many challenges and opportunities, and, although there are several software products for this purpose, there is still room for new software developments that aim mainly to take this possibility of use to farmers and technicians. In this sense, the purpose of this study was to develop a piece of software capable of simulating improved systems of beef cattle production and generate economic, physical and environmental indicators from production, reproduction, consumption, purchase, herd, area, team, time and emission parameters. At the end of the development, a model was created to verify the calculations performed by the simulator, whose results were verified by *Embrapa Gado de Corte* and *Embrapa Pantanal* researchers. It is expected that the construction of this tool will enable rural and technical producers to simulate improved production systems that can be replicated in rural properties, increasing the productivity, quality and sustainability of beef cattle.

Keywords: indicators, beef cattle, precision livestock farming, computational simulation, computational modeling.

Lista de Ilustrações

Figura 1 – Resumo das atividades da pecuária de corte e seus produtos (adaptado de CEZAR et al. 2005).	27
Figura 2 – Arquitetura do Framework Titan (CARROMEU, 2007).	31
Figura 3 – Processo para realização de mapeamento sistemático (adaptado de Petersen et al. 2008).	37
Figura 4 – Representação do processo de triagem utilizado para definir os trabalhos relevantes.	45
Figura 5 – Mapa Sistemático - Cenário de softwares para simulação de sistemas de bovinos de corte.	48
Figura 6 – Editoras que possuem publicações relacionadas a simuladores de sistemas de produção de bovino de corte QP2.	49
Figura 7 – Arquitetura da Plataforma +Precoce (GOMES, 2014).	53
Figura 8 – Discussão sobre conceitos relativos ao problema [parte 1 de 2].	55
Figura 9 – Discussão sobre conceitos relativos ao problema [parte 2 de 2].	55
Figura 10 – Primeiro esboço do modelo computacional [parte 1 de 2].	56
Figura 11 – Primeiro esboço do modelo computacional [parte 2 de 2].	57
Figura 12 – Diagrama de Entidades e Relacionamentos do modelo computacional de simulação.	59
Figura 13 – Organização das tarefas a serem executadas.	60
Figura 14 – Menus dos Módulos Administrador e Desenhista.	65
Figura 15 – Módulo Desenhista - Tela de Listagem de Modelos.	66
Figura 16 – Módulo Desenhista - Tela de Cadastro ou Edição de Fluxo.	66
Figura 17 – Módulo Desenhista - Tela de Visualização de Modelo.	67
Figura 18 – Organização do projeto da biblioteca de simulação.	69
Figura 19 – Diagrama de Classes da biblioteca de simulação.	70
Figura 20 – Representação do primeiro nó do grafo.	71
Figura 21 – Representação das estações de produção.	72
Figura 22 – Representação das estações de tratamento.	73
Figura 23 – Representação dos nós de reúso.	74
Figura 24 – Representação das estações de soma de produção.	76
Figura 25 – Representação das estações de soma de tratamento.	76
Figura 26 – Representação dos nós terminais.	77
Figura 27 – Exemplo de caso de teste elaborado pelo analista de negócios com a descrição de cada um dos componentes.	80
Figura 28 – Legenda utiliza para os grafos das próximas figuras.	81
Figura 29 – Caso de Teste 1 - Só estações de Produção.	82

Figura 30 – Caso de Teste 2 - Estações de Produção e Tratamento.	83
Figura 31 – Caso de Teste 3 - Estações de Produção, Tratamento e Reúso na fronteira (sem sucesso).	84
Figura 32 – Caso de Teste 4 - Estações de Produção, Tratamento e Reúso fora da fronteira (com sucesso).	85
Figura 33 – Caso de Teste 5 - Dois Reúsos e chegando ao Node11 sem todos fluxos de referência calculados.	86
Figura 34 – Trecho do relatório de estoque.	87
Figura 35 – Conferência do estoque para o caso de teste 4.	88
Figura 36 – Modelo de avaliação utilizado no teste dos indicadores.	91
Figura 37 – Tela do Módulo Portal exibindo o resultado de uma simulação realizada.	92
Figura 38 – O grafo do Modelo.	153
Figura 39 – Cálculo em Estação de Transformação.	163
Figura 40 – Cálculo em Estação de Tratamento.	163

Lista de Tabelas

Tabela 1 – Maiores exportadores de carne em 2016 (ABIEC, 2017).	18
Tabela 2 – Informações dos estados maiores produtores pecuários do Brasil (ABIEC, 2017).	19
Tabela 3 – Bases de buscas utilizadas.	40
Tabela 4 – Termos relevantes para a construção da <i>string</i> de busca padrão.	40
Tabela 5 – <i>String</i> de busca padrão.	40
Tabela 6 – Total de trabalhos encontrados em cada base e suas respectivas <i>strings</i> de busca.	44
Tabela 7 – Resumo do processo de triagem	46
Tabela 8 – Visão geral do mapa sistemático.	47
Tabela 9 – Categorias de recursos e hierarquia utilizadas para testar todas as funcionalidades implementadas.	88
Tabela 10 – Parâmetros e valores utilizados para testar todas as funcionalidades implementadas.	90
Tabela 11 – Filtro 1 - Critérios de exclusão	102
Tabela 12 – Dicionário de Dados - Entidade Sistema	126
Tabela 13 – Dicionário de Dados - Entidade Sistema-Parâmetro	127
Tabela 14 – Dicionário de Dados - Entidade Parâmetro	128
Tabela 15 – Dicionário de Dados - Entidade Categoria	128
Tabela 16 – Dicionário de Dados - Entidade Simulação	128
Tabela 17 – Dicionário de Dados - Entidade Simulação-Parâmetro	129
Tabela 18 – Dicionário de Dados - Entidade Modelo	129
Tabela 19 – Dicionário de Dados - Entidade Infraestrutura	130
Tabela 20 – Dicionário de Dados - Entidade Infraestrutura-Categoria	130
Tabela 21 – Dicionário de Dados - Entidade Nó	131
Tabela 22 – Dicionário de Dados - Entidade Etapa	131
Tabela 23 – Dicionário de Dados - Entidade Nó-Etapa	131
Tabela 24 – Dicionário de Dados - Entidade Fluxo	132
Tabela 25 – Dicionário de Dados - Entidade Fluxo-Categoria	133
Tabela 26 – Dicionário de Dados - Entidade Recurso	133
Tabela 27 – Dicionário de Dados - Entidade Recurso-Categoria	133
Tabela 28 – Dicionário de Dados - Entidade Fórmula	134
Tabela 29 – Dicionário de Dados - Entidade Fórmula-Categoria	134
Tabela 30 – Dicionário de Dados - Entidade Fórmula-Parâmetro	134

Lista de Algoritmos

Algoritmo 1 – Algoritmo para cálculo de simulação	68
Algoritmo 2 – Algoritmo que calcula nós e fluxos do modelo computacional utilizando a busca em largura (adaptado de GOLDBARG; GOLDBARG 2012).	71
Algoritmo 3 – Rotina de cálculo para o primeiro nó (após o raiz) do grafo.	72
Algoritmo 4 – Rotina de cálculo para estações de produção.	73
Algoritmo 5 – Rotina de cálculo para estações de tratamento.	74
Algoritmo 6 – Rotina de cálculo para nós de reúso.	75
Algoritmo 7 – Rotina de cálculo para estações de soma de produção.	76
Algoritmo 8 – Rotina de cálculo para estações de soma de tratamento.	77
Algoritmo 9 – Rotina de cálculo para nós terminais.	77

Glossário

- Acoplamento** Medida do grau de dependência, um dos outros, dos componentes de um software. Fraco ou baixo acoplamento facilita, dentre outras coisas, a manutenção.
- Analista de Negócios** Profissional que conhece objetivos e operações de determinada área de uma organização e atua para prover soluções que permitam o alcance das metas estabelecidas.
- Arquitetura de Software** Relaciona às estruturas em alto nível de um software que definem seus componentes, suas propriedades e seus relacionamentos.
- Array** Estrutura de dados que armazena uma coleção de elementos identificados por um índice.
- Coesão** Medida relacionada ao grau de responsabilidade dos componentes de um software. Busca-se a alta ou forte coesão, onde os componentes tenham uma única responsabilidade e a realizem de maneira satisfatória.
- CMS** *Content Management System*, em português, Sistema de Gerenciamento de Conteúdo. É um sistema que permite gerenciar a criação e modificação de conteúdo digital, geralmente provendo acesso a múltiplos usuários em um ambiente colaborativo.
- Diagrama de Classes** Representação da estrutura e relações das classes que compõem um software.
- Diagrama Entidade Relacionamento** Representação da organização do banco de dados. Utiliza um conjunto pré estabelecido de símbolos para representar suas entidades e seus relacionamentos.
- Estória de Usuário** Descrição informal, breve e simples de uma necessidade do usuário, contada a partir de sua perspectiva.
- Form** A *tag form* do HTML (*Hypertext Markup Language*, em português Linguagem de Marcação de Hipertexto) define um formulário (agrupamento de *tags input* do HTML) utilizado para coletar a entrada de dados do usuário.
- Framework** Conjunto de conceitos utilizado para resolver um problema em um domínio específico. Relacionado ao desenvolvimento de software, consiste

em uma abstração de códigos computacionais que cooperam entre si provendo alguma funcionalidade genérica, à qual o desenvolvedor irá customizar para criar sua aplicação.

- Grafo** Estudado em teoria dos grafos, inicialmente em matemática e, posteriormente, em computação onde são estudados algoritmos para sua manipulação. É uma estrutura composta por dois conjuntos: um de vértices e outro de arestas (pares de vértices).
- Indentação** É um termo aplicado ao código fonte de um software e se refere ao recuo apresentado, bastante útil para ressaltar e definir sua estrutura, facilitando a leitura.
- Máquina Virtual** É uma emulação de um sistema computacional que provê as funcionalidades de um computador físico, a nível de hardware e software.
- Módulo de Software** Conjunto ou agrupamento de elementos do software para atender a um assunto bem definido.
- Open Source* Um software é dito *open source* quando seu código é disponibilizado com uma licença de direitos autorais que permite estudar, modificar ou distribuir gratuitamente o software.
- Padrão de Projeto** É uma solução de projeto geral, que trata um problema recorrente de projeto em sistemas Orientados a Objeto, descrevendo o problema, a solução e quando deve ser aplicado. O modelo de solução é um arranjo genérico de classes e objetos com suas interações e relacionamentos. Para ser aplicado esse modelo deve ser customizado e implementado para o contexto particular do problema a ser resolvido.
- Plataforma** Relacionado ao ambiente no qual o software é executado, em nosso contexto refere-se a *Web*, à qual permite que o usuário possa acessar a aplicação de qualquer lugar, bastando para isso possuir dispositivo com acesso a internet e que tenha um *web browser*.
- Projeto de Software** Envolve o planejamento da solução em software, refletindo as decisões sobre a organização lógica do software e que especifica o software que será construído. Atividade realizada após especificação dos requisitos e antes da codificação.
- Provisionamento** Consiste em deixar um servidor pronto para ser utilizado, seja ele físico ou virtual, com softwares, configurações e dados iniciais necessários.

Regra de Negócio	Especifica a forma como a instituição realiza seus negócios. É ela quem especifica as particularidades das funcionalidades a serem desenvolvidas.
Requisitos de Software	Descrição das funcionalidades fornecidas pelo software e restrições operacionais. Refletem as necessidades do cliente para resolver determinado problema.
<i>REST</i>	<i>Representational State Transfer</i> , em português, Transferência de Estado Representacional. Modelo de arquitetura que fornece padrões para comunicação entre sistemas distribuídos (baseados em comunicação via rede).
<i>REST-like</i>	Uma implementação é dita <i>REST-like</i> quando não segue todos princípios do modelo <i>REST</i> , enquanto as que seguem estritamente todos os princípios é dita <i>RESTful</i> .
<i>Script</i>	Sequência de comandos computacionais que automatizam a execução de tarefas para um operador humano.
<i>String</i>	Sequência de caracteres.
TIC	Tecnologias da Informação e Comunicação, converge elementos tecnológicos tais como hardwares, redes de comunicação, dispositivos móveis, softwares, etc., que correspondem a todas tecnologias que realizam mediação ou interferem nos processos de comunicação entre seres humanos.

Sumário

1	INTRODUÇÃO	18
1.1	Motivação e Justificativa	19
1.2	Objetivo	21
2	EMBASAMENTO TEÓRICO	22
2.1	Conceitos Sobre Pecuária	22
2.1.1	Pecuária e Bovinocultura de Corte	22
2.1.2	Pecuária de Precisão	22
2.1.3	Carcaça Bovina	24
2.1.4	Novilho Precoce	25
2.1.5	Sistemas de Produção de Gado de Corte	26
2.2	Simulação Computacional	27
2.2.1	Simuladores na Agropecuária	29
2.3	Titan Framework	30
2.4	Boas Práticas para Codificação	32
2.5	Boas Práticas de Desenvolvimento de Software	32
2.5.1	Envolvimento do Cliente e Compromisso	33
2.5.2	<i>Product Backlog</i>	34
2.5.3	<i>Sprint</i>	34
2.5.4	Ambiente de Desenvolvimento Informativo	34
2.6	Considerações Finais	34
3	MAPEAMENTO SISTEMÁTICO	36
3.1	Planejamento do Mapeamento Sistemático	37
3.1.1	Questões de Pesquisa	38
3.1.2	Bases de Busca	39
3.1.3	<i>String</i> de Busca Padrão	40
3.1.4	Triagem - Critérios de Inclusão e Exclusão	40
3.1.5	Esquema de Classificação	41
3.2	Execução do Mapeamento Sistemático	42
3.2.1	<i>Strings</i> de Busca Específicas	43
3.2.2	Triagem dos Trabalhos	43
3.2.2.1	Filtro 1	43
3.2.2.2	Filtro 2	44
3.2.2.3	Filtro 3	45
3.2.3	<i>Keywording</i>	46

3.3	Resultados do Mapeamento Sistemático	46
3.4	Considerações Finais	50
4	ESPECIFICAÇÃO DE REQUISITOS E PROJETO DO SIMULADOR DA PLATAFORMA +PRECOCE	51
4.1	Comunicação	51
4.2	Visão Geral do Sistema	52
4.3	Identificação de Requisitos e Projeto Inicial do Simulador da Plataforma +Precoce	54
4.4	Projeto do Simulador da Plataforma +Precoce	56
4.5	Organização Inicial do <i>Product Backlog</i>	58
4.6	Considerações Finais	61
5	IMPLEMENTAÇÃO DOS MÓDULOS DESENHISTA E SIMULADOR DA PLATAFORMA +PRECOCE	62
5.1	Ambiente Utilizado para Construção do Módulo Desenhista	62
5.2	Trabalho Realizado para Construção do Módulo Desenhista	62
5.3	Ambiente Utilizado para Construção do Módulo Simulador	67
5.4	Trabalho Realizado para Construção do Módulo Simulador	68
5.5	Avaliando o Módulo Simulador	79
5.6	Considerações Finais	89
6	CONCLUSÕES	93
6.1	Limitações e Sugestões para Trabalhos Futuros	94
	REFERÊNCIAS	96
	APÊNDICES	101
	APÊNDICE A – MAPEAMENTO SISTEMÁTICO - FILTRO 1	102
	APÊNDICE B – MAPEAMENTO SISTEMÁTICO - FILTRO 3	120
	APÊNDICE C – REQUISITOS FUNCIONAIS	124
	APÊNDICE D – DICIONÁRIO DE DADOS DO MODELO COMPUTACIONAL	126
	APÊNDICE E – INTEGRAÇÃO DO MÓDULO SIMULADOR À PLATAFORMA +PRECOCE	135

ANEXOS

137

ANEXO A – CONCEITOS E REQUISITOS FUNCIONAIS - MÓDULOS 0 E 1

A.1	Objetivo	138
A.2	Introdução	138
A.3	Principais conceitos	139
A.3.1	Sistema	139
A.3.2	Fases	139
A.3.3	Tecnologias	140
A.3.4	Programas	140
A.3.5	Tamanho	140
A.3.6	Filtros	140
A.3.7	Protocolo	141
A.3.8	Modelo	141
A.3.9	Parâmetro	142
A.3.9.1	Área de produção	144
A.3.9.2	Parâmetros de protocolo	144
A.3.9.3	Parâmetros de desempenho	144
A.3.9.4	Parâmetros de mercado	144
A.3.9.5	Parâmetros de qualidade (da carcaça)	145
A.3.10	Indicador	145
A.3.10.1	Físicos	146
A.3.10.2	Econômico-financeiros	146
A.3.10.3	Ambientais	146
A.3.11	Simulação	146

ANEXO B – CONCEITOS E REQUISITOS FUNCIONAIS - MÓDULO 2

B.1	Objetivo	149
B.2	Modelo Conceitual e Requisitos Funcionais	149
B.3	Safras e Ciclos de produção	150
B.4	Sistemas, Modelos, Estações e Recursos	151
B.5	Estações e Durações	152
B.6	Fluxos, Entradas e Saídas	152
B.7	Estações de Transformação e Tratamento	154
B.7.1	Estação de Transformação	154
B.7.2	Estação de Tratamento	155
B.8	Recursos	156
B.8.1	Área	156

B.8.2	Categoria (animal)	157
B.8.3	Insumo	157
B.8.4	Serviço	158
B.9	Unidade de medida	158
B.10	Preços	158
B.11	Infraestrutura	158
B.12	Fórmulas	159
B.12.1	Fórmulas de cálculo de Coeficientes	159
B.12.2	Fórmulas de cálculo de Duração e Fração	159
B.12.3	Fórmulas de cálculo de Preços	159
B.12.4	Fórmulas de cálculo de Indicadores	160
B.12.5	Características comuns a todas as Fórmulas	160
	ANEXO C – MÓDULO 2 – CÁLCULO DA SIMULAÇÃO	161
C.1	Objetivo	161
C.2	Introdução	161
C.3	Cálculo de Coeficientes, Durações e Preços	161
C.3.1	Calcula Coeficientes	161
C.3.2	Calcula Durações	162
C.3.3	Calcula Preços	162
C.4	Cálculo de Fluxos	162
C.4.1	Calcula um Ciclo de produção	162
C.4.2	Calcula a Safra	166
C.4.3	Totaliza as Entradas e Saídas do Modelo	166
C.4.4	Reúsa os Coprodutos do Modelo	167
C.4.5	Calcula a Área de produção default	167
C.4.6	Ajusta as Quantidades à Área de produção informada	168
C.5	Cálculo de Indicadores	168
C.5.1	Calcula Receitas e Custos	169
C.5.2	Calcula Indicadores de Infraestrutura	169
C.5.3	Calcula Indicadores de Mão de obra própria	169
C.5.4	Calcula Indicadores de Área de produção	169
C.5.5	Calcula Indicadores de rebanho	169
C.5.6	Calcula Indicadores ambientais	170
C.5.7	Calcula os demais Indicadores	170
	ANEXO D – ALGORITMO PARA CÁLCULO DA SIMULAÇÃO	171

1 Introdução

O agronegócio destaca-se como elemento fundamental na economia e na sociedade brasileira. O notável crescimento do setor agropecuário brasileiro tem se sustentado no aumento da produtividade, muito mais do que na ocupação de novas terras, fato este sustentado pela aplicação de tecnologias e conhecimento acumulado tanto pelos produtores quanto por instituições que integram as cadeias produtivas do agronegócio. A inovação é a base para o desenvolvimento sustentável e atributo potencializador para criação de novas oportunidades e enfrentamento de desafios impostos por mercados globalizados (SANTOS et al., 2011).

O agronegócio é um dos principais pilares econômicos do país, contribuindo em 2016 com R\$1,48 trilhão ou 23,6% do PIB total brasileiro. O setor pecuário é responsável por 31% desse total (R\$460 bilhões), sendo que a exportação de carne bovina representou quase 3% de todas as exportações brasileiras, o que contribuiu para a manutenção do saldo comercial positivo brasileiro, demonstrando a importância econômica da atividade (ABIEC, 2017).

O Brasil é um dos maiores produtores e exportadores de carne bovina do mundo. No cenário global, o Brasil rotineiramente figura como detentor do maior rebanho, apresentando 219,1 milhões de cabeças de gado em 2016, ficando à frente de outros países de destaque no setor de produção de carne como EUA (92,9 milhões), China (112,7 milhões) e Índia (188,6 milhões) (ABIEC, 2017). O Brasil ocupou por vários anos o posto de maior exportador de carne bovina do mundo (Tabela 1).

Tabela 1 – Maiores exportadores de carne em 2016 (ABIEC, 2017).

País	Exportações Toneladas Equivalente Carcaça (mil TEC)	Exportação / Produção (%)	Produção Toneladas Equivalente Carcaça (mil TEC)
Brasil	1.832,2	20,04%	9.142,0
Índia	1.558,5	54,73%	2.847,8
Austrália	1.428,6	68,43%	2.087,9
EUA	1.227,0	10,70%	11.471,1

Embora ainda não tenha figurado como maior produtor de carne mundial, a ABIEC (Associação Brasileira das Indústrias Exportadoras de Carne) acredita que o país está a poucos passos de se tornar e aposta na utilização da tecnologia pela bovinocultura como principal ferramenta que levará o Brasil a esse posto (Sociedade Nacional de Agricultura, 2016; ABIEC, 2017).

Acompanhado pela queda do PIB brasileiro, o ano de 2015 não teve resultados tão expressivos para o setor da pecuária de corte, apresentando retração do abate e queda nas exportações, passando a ocupar o terceiro lugar no ranking mundial dos países exportadores de carne bovina. Apesar da queda, o efetivo de bovinos apresentou crescimento de 1,3% em relação ao ano de 2014, totalizando 215,20 milhões de cabeças, sendo a região centro-oeste a que ainda apresenta o maior rebanho (IBGE, 2015). Crescimento que foi mantido nos anos subsequentes (ABIEC, 2016; ABIEC, 2017).

A bovinocultura de corte é uma atividade de bastante importância econômica para a região centro-oeste, possuindo grandes propriedades e produtores especializados no seu desenvolvimento. Clima, relevo e solo favorecem a atividade que é possibilitada pelas grandes plantas frigoríficas presentes na região (IBGE, 2015). No cenário nacional, o Mato Grosso do Sul assume papel de destaque na criação de bovinos, apresentando o quarto maior efetivo de cabeças de gado (Tabela 2).

Tabela 2 – Informações dos estados maiores produtores pecuários do Brasil (ABIEC, 2017).

Estados	Rebanho estimado em 2016	Porcentagem do rebanho no total do Brasil	Porcentagem de animais exclusivamente destinados a corte no rebanho do Estado
Mato Grosso	30.258.870	13,81%	94,60%
Minas Gerais	24.368.597	11,12%	41,36%
Goiás	22.754.548	10,39%	68,51%
Mato Grosso do Sul	22.175.082	10,12%	94,65%
Pará	19.477.676	8,89%	89,56%

A concorrência global impõe forte ritmo na produção de gado de corte. Um mercado com concorrência acirrada exige cada vez mais produtos com melhor qualidade e cuja produção considere o bem-estar animal, baixo impacto ambiental e justiça social em sua cadeia produtiva (EUCLIDES FILHO, 2013). O presente projeto vem ao encontro desta ideia visando a auxiliar o produtor rural a melhorar seus sistemas de produção.

O uso das Tecnologias da Informação e Comunicação (TICs) no setor agrícola brasileiro está bem abaixo do que se apresenta em outros setores da indústria e de serviços em geral. Aos poucos, o uso de TICs na pecuária e agricultura se tornará essencial como já o é no comércio, na área financeira, etc. A utilização das TICs e, especificamente simuladores, se tornará ferramenta fundamental para o auxílio à tomada de decisão, desenvolvimento de projetos e uso de recursos naturais (SANTOS et al., 2011).

1.1 Motivação e Justificativa

O produtor de carne bovina no Brasil encontra cada vez mais oportunidades para participar de programas de parceria que estimulam a produção de carne de qualidade

(conforme apresentado na Seção 2.1.4), por exemplo, incentivos governamentais (ex.: Novo Novilho Precoce do Estado de Mato Grosso do Sul), incentivos da indústria (ex.: Boi no Ponto, Fomento Angus Marfrig), alianças mercadológicas (ex.: Associação Sulmato-grossense de Produtores de Novilho Precoce) e de cooperativas (ex.: Cooperativa de Carnes Maria Macia). Estas iniciativas estimulam o produtor por meio de pagamentos diferenciados em relação ao preço balcão (mercado comum) para aqueles produtores que atingem certos critérios de qualidade na produção da carcaça bovina. Sendo assim, tais iniciativas proporcionam o aumento das receitas das propriedades que se adequem a esses critérios de produção, ao passo que visam melhorar a qualidade da carne bovina produzida no Brasil.

Frente a estas oportunidades, os produtores rurais enfrentam desafios para se adequar aos critérios de qualidade definidos pelos programas que buscam carcaças de animais cada vez mais jovens, pesados e com acabamento de gordura adequado. Essas características só poderão ser obtidas com o emprego de novas tecnologias relacionadas ao manejo de bovinos, que demandam maior investimento econômico por parte do produtor e maior grau de conhecimento para empregar as múltiplas possibilidades de interação dessas novas tecnologias.

Neste sentido, a Embrapa Gado de Corte vem conduzindo estudos há várias décadas visando a construção de sistemas melhorados de produção que sirvam de referência ao produtor rural (CORRÊA; ARRUDA, 1988; CORRÊA et al., 2000; CORRÊA et al., 2003; CORRÊA et al., 2006; EUCLIDES et al., 2001; MELO FILHO et al., 2006). Estes sistemas trazem consigo a descrição das técnicas empregadas na produção, coeficientes técnicos estimados e indicadores de desempenho econômico, facilitando sua análise, interpretação e adoção.

No entanto, esses sistemas propostos estão disponibilizados na forma de publicações técnico-científicas e os indicadores são elaborados utilizando planilhas automatizadas, que são destinadas a pesquisadores e não a produtores e técnicos da iniciativa privada, o que dificulta o alcance e utilização por parte destes atores, além de dificultar o compartilhamento de atualizações e criação de novos modelos. A criação da Plataforma +Precoce (GOMES, 2014) visa preencher esta lacuna, criando um local centralizado para produtores e técnicos terem acesso a sistemas de produção melhorados propostos por meio de pesquisas, seus indicadores e descrição, garantindo que sejam informações passíveis de atualização e enquadramento em diferentes realidades de produção. A Plataforma permitirá que o usuário faça mudanças em parâmetros, de modo a reproduzir as características de determinada propriedade rural e simule os resultados dos sistemas de produção através de um software de simulação.

Facilitando o acesso a sistemas melhorados, a Plataforma pretende fomentar a adoção de tecnologias pela pecuária de corte e a participação em programas que valorizem

a produção de carne de qualidade, levando o conhecimento destes sistemas a um maior número de produtores rurais. Por consequência, estimular pesquisadores e técnicos a estudar e criar sistemas evoluídos e passíveis de serem divulgados pela Plataforma e, com isso, contribuir para a melhora da produtividade, da rentabilidade e da qualidade da carne bovina produzida no país.

1.2 Objetivo

A Plataforma +Precoce tem como objetivo prover o acesso aos produtores de gado de corte a sistemas melhorados de produção, com vistas ao aumento da produtividade e sustentabilidade desta atividade e melhoria da qualidade da carne brasileira. O objetivo deste trabalho é servir de aporte à Plataforma com a construção do software de simulação.

Os objetivos específicos são:

- Especificar o modelo computacional que atenda a simulação dos sistemas de produção de gado de corte definidos pela Embrapa.
- Desenvolver software que permita a administração (cadastrar, visualizar, editar e excluir) desses modelos.
- Desenvolver software simulador que atenda esse modelo.
- Desenvolver módulos adicionais que gerem indicadores econômicos e ambientais de forma integrada às simulações.
- Integrar o simulador à Plataforma web +Precoce.
- Avaliar o software simulador na Plataforma web +Precoce.

Nas seções que seguem são apresentados os detalhes deste trabalho. No Capítulo 2 é feita descrição de conceitos que permeiam a área do projeto desenvolvido. O Capítulo 3 apresenta o mapeamento sistemático, conduzido para prover o conhecimento inicial desta área de pesquisa. Nos Capítulos 4 e 5 são apresentados os resultados obtidos. Por fim, no Capítulo 6 é feita a conclusão do trabalho, destacando o que se obteve para cada um dos objetivos propostos e as contribuições e limitações deste trabalho.

2 Embasamento Teórico

2.1 Conceitos Sobre Pecuária

Este trabalho é parte do escopo maior de um projeto desenvolvido no Centro Nacional de Pesquisa de Gado de Corte (CNPGC) da Embrapa que visa a construção de um simulador para sistemas de produção de novilho precoce. Neste capítulo são apresentados os principais conceitos e definições importantes para o desenvolvimento deste trabalho.

2.1.1 Pecuária e Bovinocultura de Corte

O termo pecuária é relativo às atividades que tratam de aspectos da criação de gado, ao passo que gado é a representação de um conjunto de animais (bois, búfalos, carneiros, cavalos, etc.) utilizados para algum fim (HOUAISS; VILLAR; FRANCO, 2009).

A bovinocultura é estudada pela Zootecnia e trata do manejo e técnicas para criação de bovinos, envolvendo, dentre outros, o estudo de tecnologias de reprodução, raças, alimentação, suplementação e manejo da cadeia produtiva dessa espécie animal (MARQUES, 1969; PIRES, 2010). Os tipos zootécnicos tratados pela bovinocultura são dois: bovinocultura de corte e leiteira, sendo o primeiro tipo o foco deste trabalho, ou seja, a produção de bovinos visando à produção de carne (MARQUES, 1969).

Corrêa et al. (2000) relatam que a bovinocultura de corte brasileira era caracterizada pela criação extensiva, com baixo uso de insumos, em que o crescimento da produção dependia da incorporação de novas áreas, mas em meados de 1990 esse cenário começou a mudar, impulsionado, dentre outros fatores, pela crescente preocupação com o meio ambiente, escassez de investimentos públicos e uso intensivo de capital. Este cenário foi decisivo para o CNPGC, que possuía significativo acervo de tecnologias, serviços e produtos, porém, com baixa taxa de utilização, migrar seus esforços visando a adoção pelos produtores de sistemas de produção mais intensivos, que apresenta melhor uso dos recursos, especialmente da área disponível.

2.1.2 Pecuária de Precisão

A pecuária de precisão possui origem que remete à agricultura de precisão. E, esta por sua vez, difere da tradicional por levar em conta a variação espacial e temporal, em detrimento da visão da lavoura como espaço homogêneo. A pecuária de precisão tem potencial para ajudar pecuaristas da mesma forma que a agricultura de precisão já auxilia

o produtor agrícola há vários anos e, apesar de algumas aplicações serem comuns às duas técnicas, a produção de gado infere manipulação de animais vivos, o que acrescenta novos desafios, pois trata-se de seres sencientes, cujas necessidades devem ser levadas em consideração (WATHES et al., 2008).

Antes de definir o que é pecuária de precisão, é importante contextualizar as origens desta tecnologia. PIRES et al. (2004) traçam o contexto histórico da agricultura de precisão. Os fundamentos surgiram em publicações datadas de 1929, em que os autores constataram a existência de grandes variações quanto a necessidade da aplicação de calcário de acordo com a localização da área de aplicação. No entanto, essa visão heterogênea da propriedade não foi levada adiante em virtude do surgimento de equipamentos de tração mecânica que facilitavam a aplicação de insumos a taxas uniformes (visão homogênea da propriedade). A visão heterogênea da propriedade só volta ao foco com os avanços tecnológicos relacionados ao desenvolvimento de microcomputadores, sensores e sistemas de rastreamento e satélites que ocorreram na década de 80.

Os primeiros avanços concentravam estudos relacionados ao desenvolvimento de sensores, porém, com o surgimento do GPS (*Global Positioning System*)¹ houve considerável incremento nas pesquisas desenvolvidas na área e novas tecnologias foram desenvolvidas, como exemplo, máquinas agrícolas providas de receptores GPS, com armazenagem de dados de produção associados à coordenadas geográficas, que possibilita a correta aplicação do insumo de acordo com a localização no espaço geográfico (INAMASU; BERNARDI, 2014; PIRES et al., 2004).

Uma visão atualizada da agricultura de precisão estabelece seu relacionamento com sistemas de gestão, que empregam conjuntos de tecnologias e procedimentos para que lavouras e sistemas de produção sejam otimizados (MOLIN, 2002 apud PIRES et al., 2004), propiciando maior produtividade e mitigação de impactos ambientais (BERNARDI et al., 2017).

A pecuária de precisão é definida por WATHES et al. (2008) como sendo a gestão da produção pecuária usando princípios e tecnologias da engenharia de processos, principalmente com o uso de sensores inteligentes. A adoção de métodos gerenciais baseados em computador é premissa para que se consiga obter produção pecuária sustentável, cenário em que as especificações do produto sejam cumpridas com lucro, enquanto impactos ambientais adversos são minimizados e a saúde e o bem-estar animal são promovidos.

Pela mesma linha e de forma complementar, autores nacionais da área da computação como Cáceres et al. (2011) definem pecuária de precisão como a área que estuda o uso das tecnologias de informação e comunicação na produção de bovinos. Na pecuária de precisão, a tecnologia da informação é utilizada para garantir o uso de boas práticas no

¹ Em português: Sistema Global de Posicionamento.

processo de produção pecuária, almejando assim, impulsionar a produtividade. A utilização de tecnologias pode ser feita em vários níveis e em diversas áreas da indústria bovina, lançando novos desafios para as disciplinas de Ciência da Computação no que tange o desenvolvimento de novas ferramentas e dispositivos (software e hardware) que aumentem a competitividade de estabelecimentos pertencentes a indústria pecuária.

Os principais problemas da indústria pecuária que demandam o uso de soluções computacionais estão relacionados à gestão de rebanho com uso de dispositivos eletrônicos, simulação de crescimento de pastagens, previsão de preço de *commodities*, diagnóstico de doenças e linhagem genéticas em animais, classificação de carcaças, simulação de práticas de manejo e logística, desenvolvimento de ferramentas de apoio a tomada de decisões, aspectos do comportamento animal, emissão de gases poluentes, etc. Estas soluções, além de possuírem seus próprios desafios de desenvolvimento na área da Computação, também devem estar adequadas às características de uso na indústria pecuária, por exemplo, a necessidade de produção de equipamentos robustos adaptados ao uso em condições adversas e uso de tecnologias móveis que permitam a comunicação de dados capturados a serem transmitidos por longas distâncias, dentre outras (CÁCERES et al., 2011; SANTOS et al., 2011).

A adoção de práticas da pecuária de precisão pode ser usada para melhorar a gestão da propriedade rural em diversas perspectivas (PIRES; DOS SANTOS NETO; COSTA JUNIOR, 2014):

- Agronômica: ajuste de práticas culturais para as reais necessidades de cultura (tipos de pastagem, fertilização, etc.);
- Técnica: gestão de calendários (planejamento de atividades, vacinação, etc.);
- Ambiental: redução dos impactos agrícolas; e
- Econômica: aumento da produção e/ou redução de custos.

2.1.3 Carcaça Bovina

A carcaça bovina é o produto resultante do abate de um animal. Destinado ao mercado consumidor, a carcaça deve seguir diretrizes de padronização, sendo que o Brasil possui legislação específica que define os critérios que esse produto deve atender para poder ser comercializado. De acordo com o decreto de lei Nº 9013 de 29 de março de 2017, que dispõem sobre a inspeção industrial e sanitária de produtos de origem animal, a carcaça é definida como (BRASIL, 2017):

“Art. 277. Para os fins deste Decreto, carcaças são as massas musculares e os ossos do animal abatido, tecnicamente preparado, desprovido de cabeça,

órgãos e vísceras torácicas e abdominais, respeitadas as particularidades de cada espécie, observado ainda:

I - nos bovinos, nos búfalos e nos equídeos a carcaça não inclui pele, patas, rabo, glândula mamária, testículos e vergalho, exceto suas raízes;”

2.1.4 Novilho Precoce

Em um mercado competitivo e globalizado, com necessidades elevadas de giro de capital, qualidade de produto e alta demanda (EUCLIDES FILHO, 2000) o tempo do nascimento até o abate é fator diferencial e competitivo para o produtor rural, de modo que a produção do novilho precoce pode ser vista como consequência do progresso no setor pecuário, em que grupos de produtores conseguem, com a utilização de tecnologias de produção, abater novilhos em torno de doze meses de idade (“novilho superprecoce”), diferença significativa, frente ao sistema de produção tradicional, onde o abate do bovino ocorre em torno dos quarenta e oito meses (ALMEIDA et al., 1999).

O abate em idade mais jovem sugere um manejo intensivo visando o aumento nos índices de produtividade e contribuição real no âmbito da sustentabilidade ambiental, ao passo que a valorização da carne de melhor qualidade contribui para dimensão social e econômica da sustentabilidade (ABREU et al., 2015). Existe correlação direta entre carne de melhor qualidade e novilho precoce, pois, como a carne provém de animais jovens (em torno de dois anos de idade) ela apresenta textura com maior maciez e melhor acabamento de gordura. A classificação da carne proveniente de um novilho precoce deve atender alguns padrões pré-estabelecidos para sua classificação (ALMEIDA et al., 1999):

- Dentição:
 - Dente de leite
 - Dois dentes definitivos
 - Quatro dentes definitivos
- Peso mínimo:
 - Machos com carcaças de 255 kg
 - Fêmeas com carcaças de 180 kg
- Carcaça:
 - Convexa
 - Subconvexa
 - Retilínea
- Espessura da camada de gordura que recobre a carcaça:
 - Mínima de 3 mm.

2.1.5 Sistemas de Produção de Gado de Corte

Segundo Euclides Filho (2000), um sistema de produção de gado de corte é definido da seguinte forma:

“Entende-se por sistema de produção de gado de corte o conjunto de tecnologias e práticas de manejo, bem como o tipo de animal, o propósito da criação, a raça ou grupamento genético e a ecorregião onde a atividade é desenvolvida. Devem-se considerar, ainda, ao se definir um sistema de produção, os aspectos sociais, econômicos e culturais, uma vez que esses têm influência decisiva, principalmente, nas modificações que poderão ser impostas por forças externas e, especialmente, na forma como tais mudanças deverão ocorrer para que o processo seja eficaz e as transformações alcancem os benefícios esperados. Permeando todas essas considerações, devem estar a definição do mercado e a demanda a ser atendida, ou seja, quais são e como devem ser atendidos os clientes ou consumidores”.

Em 1977 o CNPGC iniciou seus trabalhos em modelagem de sistemas desenvolvendo seu primeiro modelo matemático. Tal modelo visava inferir o impacto econômico causado pelo uso de novas tecnologias no manejo de bovinos (TORRES et al., 1977 apud CORRÊA et al., 2000). Outros estudos foram conduzidos e, em 1983, o centro chegou a um sistema baseado em tecnologias simples, de fácil aplicação prática e baixo custo, cujos resultados zootécnicos se mostraram acima da média no Brasil Central, sendo o modelo adotado nos onze anos seguintes. Por se tratar de um sistema de produção extensivo sua evolução foi descontinuada, em detrimento da formulação de um novo, semi-intensivo (CORRÊA et al., 2000).

Os sistemas de produção servem ao produtor pecuário como modelo a ser seguido para obtenção de um resultado próximo ao estudado. A escolha de qual sistema atende melhor ao produtor não depende unicamente do seu desejo. Fatores relacionados à capacidade de investimento, área disponível, bioma em que está inserido, diferenças regionais, particularidades de microrregiões e até mesmo fatores sociais e culturais influenciam nessa decisão. Outro ponto de primordial importância está relacionado à demanda do mercado consumidor e indicadores de retorno econômico e ambientais. Dessa forma, adotar um sistema de produção está diretamente ligado a melhoria da eficiência produtiva (EUCLIDES FILHO, 2000; CORRÊA et al., 2000).

Para se obter o animal pronto para ser abatido, a pecuária de corte define três fases distintas: cria, recria e engorda. São estas fases que irão gerar retorno econômico aos sistemas de produção, podendo ser desenvolvidas como atividades isoladas ou combinadas conforme observado na Figura 1 e descrito a seguir (CEZAR et al., 2005):

- Cria: rebanho de fêmeas em reprodução, podendo estar incluída a recria de fêmeas para reposição, para crescimento do rebanho e para venda. Venda de todos os machos

desmamados, com idade entre 7 a 9 meses. Comercialização de bezerras desmamadas e novilhas, vacas e touros. Em geral, as bezerras desmamadas, novilhas jovens (1 a 2 anos) são vendidas para reprodução.

- Cria e recria: cria com retenção de machos até 15 a 18 meses de idade (garrotes), quando então são comercializados.
- Cria, recria e engorda: também conhecida como atividade de ciclo completo, os machos são vendidos como bois gordos para abate, com idade entre 15 e 42 meses.
- Recria e engorda: inicia com o bezerro desmamado, oriundo de sistemas de cria e termina com o boi gordo. Na existência de oferta de garrote de melhor qualidade pode iniciar com este tipo animal, reduzindo o período de recria e engorda quando associado a uma boa alimentação. O mesmo ocorre com bezerros desmamados de alta qualidade. Embora utilizem predominantemente machos, existe também a utilização de fêmeas.
- Engorda (terminação): pouco realizada como atividade isolada atualmente devido a redução de oferta de bois magros e vacas magras. Bastante exercida em décadas passadas onde existia grande oferta de boi magro (24 a 36 meses de idade). É mais característica da atividade de confinamento.

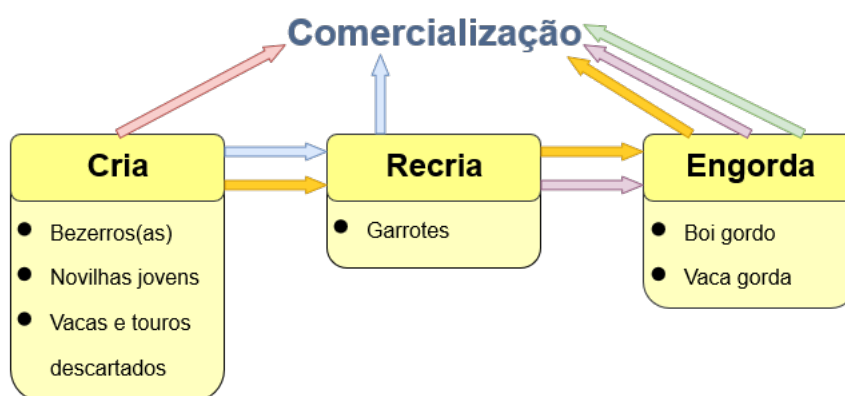


Figura 1 – Resumo das atividades da pecuária de corte e seus produtos (adaptado de CEZAR et al. 2005).

2.2 Simulação Computacional

A palavra simulação possui diversos significados. Por exemplo, dentre os encontrados no dicionário Houaiss (HOUAISS; VILLAR; FRANCO, 2009), estão:

- Ação ou efeito de simular;

- Imitação do funcionamento de um processo por meio do funcionamento de outro;
- **Simulação analógica:** teste ou experiência em que os modelos empregados têm comportamento análogo ao da realidade;
- **Simulação digital:** experiência ou ensaio constituído por uma série de cálculos numéricos e decisões de escolha limitada, executados de acordo com um conjunto de normas preestabelecidas e apropriadas à utilização de computadores.

Freitas Filho (2008) faz uma breve abordagem da simulação computacional, citando que em 1974 Thomas J. Schriber definia simulação, no clássico livro “*Simulation Using GPSS*”, como: “simulação implica na modelagem de um processo ou sistema, de tal forma que o modelo imite as respostas do sistema real numa sucessão de eventos que ocorrem ao longo do tempo” (SCHRIBER, 1974 apud FREITAS FILHO, 2008). GPSS (*General Purpose Simulation System*)² foi a primeira linguagem comercial de programação voltada à simulação digital de sistemas. Nessa época ainda era comum o uso de modelos analógicos e físicos para o estudo do comportamento de sistemas.

Em 1975, Robert Shannon definiu a simulação com o uso de um modelo computacional, isto é, “um programa de computador cujas variáveis apresentam o mesmo comportamento dinâmico e estocástico do sistema real que representa” (SHANNON, 1975 apud FREITAS FILHO, 2008). Uma definição que abrange todo o processo de simulação com o auxílio de modelos computacionais é definida por Dennis Pegden em 1991, da seguinte forma: “simulação é o processo de projetar um modelo computacional de um sistema real e conduzir experimentos com este modelo com o propósito de entender seu comportamento e/ou avaliar estratégias para sua operação”(PEGDEN, 1991 apud FREITAS FILHO, 2008).

Freitas Filho (2008) destaca que na visão apresentada por Pegden (1991), a simulação se apresenta como um processo amplo, que não compreende simplesmente a construção de um modelo, mas também a construção de todo o método experimental, que busca:

- Descrever o comportamento do sistema;
- Realizar observações para construir teorias e hipóteses; e
- Garantir que efeitos produzidos por alterações no sistema ou nos métodos utilizados sejam suportados pelo modelo, de modo a prever o comportamento futuro do sistema.

Na Engenharia, a simulação de sistemas físicos reais é amplamente utilizada para observar o comportamento de modelos em ensaios sob condições diversas, permitindo a avaliação sem incorrer em riscos financeiros, acidentes, atrasos, etc. A simulação, como recurso

² Em português: Sistema de Simulação de Propósito Geral

utilizado em Engenharia, pode ser classificada da seguinte forma (BAZZO; PEREIRA, 2013):

- **Icônica:** utiliza a construção de modelos físicos, geralmente em escala reduzida, dos sistemas físicos reais. Como exemplo, podem ser citados os túneis de vento, utilizados nos projetos de carros e aviões, onde um modelo geométrico semelhante é utilizado para avaliar a influência da forma no arrasto aerodinâmico;
- **Analógica:** quando se utiliza a comparação de sistemas complexos ou de difícil manuseio, com sistemas mais simples e tangíveis, fazendo com que o sistema se comporte de forma análoga ao outro. Como exemplo, pode-se citar a utilização de água para representar o ar passando pelas pás de uma turbina; e
- **Matemática:** utiliza a descrição matemática para as características essenciais dos elementos idealizados, fornecendo um modelo de previsão, em que os valores de variáveis de entrada do modelo matemático definem valores de saída do modelo matemático. Como exemplo, pode-se citar o dimensionamento da potência de um motor para movimentar uma esteira ou a potência de uma bomba hidráulica para sucção de água de um poço artesiano.

Neste trabalho objetiva-se a construção de um simulador digital (ou computacional), que dará ao produtor rural e a técnicos a facilidade de testar modificações no sistema escolhido, podendo alterar variáveis de entrada e verificar como o sistema se comporta a essas mudanças.

2.2.1 Simuladores na Agropecuária

A simulação de sistemas de produção de bovinos de corte assume papel estratégico para o produtor como ferramenta de apoio à tomada de decisão quando se objetiva produzir uma carne de melhor qualidade, mais atrativa ao mercado consumidor e obter maior lucratividade (EUCLIDES FILHO, 2000).

Existem vários simuladores disponíveis no setor agropecuário, sendo mencionados três deles a seguir:

- **APSIM:** simulador de sistemas agrícolas, contém módulos que permitem a simulação de sistemas que cobrem simulações envolvendo solo, plantas, animais e clima (APSIM Initiative, 2016);
- **PaSim:** o principal objetivo da ferramenta é prover simulações dos impactos das alterações climáticas nas pastagens e obter *feedback* relacionado à emissão de gases do efeito estufa na atmosfera. Atualmente a ferramenta é desenvolvida pela

Unidade de Pesquisas de Ecossistemas de Pastagens do Instituto Nacional para Pesquisas Agrícolas Francês, utilizando a linguagem Fortran 90. A ferramenta é composta por submodelos para plantas, animais, microclima, biologia e física do solo e gerenciamento (PASIM, 2016);

- Embrapa Invernada: ferramenta de apoio para o planejamento da produção de bovinos de corte. Permite realizar simulações do crescimento de pastagens, pastejo e crescimento de animais, com possibilidade de formular dietas otimizadas e comparar e analisar cenários de produção (INVERNADA, 2016).

2.3 Titan Framework

Titan é um *framework* Web para instanciação de gerenciadores de conteúdo (*CMS* – *Content Management Systems*), cuja proposta é ser uma ferramenta simples e completa que permita criar aplicações robustas de maneira rápida e fácil. Seu desenvolvimento foi realizado pelo grupo de pesquisas do LEDES-DCT/UFMS (CARROMEU, 2007), e os detentores de sua licença (versão 3) são a Universidade Federal de Mato Grosso do Sul (UFMS), por intermédio do Laboratório de Engenharia de Software (LEDES) da Faculdade de Computação (FACOM) e a Empresa Brasileira de Pesquisa Agropecuária (EMBRAPA), pelo Laboratório de Precisão Pecuária, Meio Ambiente e Engenharia de Software (PLEASE Lab) (Titan Framework, 2016).

O *framework* Titan foi implementado com o uso de tecnologias *Open Source*, utilizando a linguagem *PHP* (originalmente foi nomeado *Personal Home Page* por seu criador Rasmus Lerdorf e, renomeado como o acrônimo recursivo para **PHP: Hypertext Preprocessor** quando a implementação de referência passou para o *The PHP Group*)³, banco de dados *PostgreSQL*⁴ e a linguagem de marcação XML para configuração das instâncias de aplicação. Como pode ser visto na Figura 2, sua arquitetura é composta por um núcleo, que oferece os recursos necessários à execução das instâncias. Esse núcleo possui uma API (*Application Programming Interface*)⁵ que oferece diversas funcionalidades, tais como (CARROMEU, 2007):

- Controle de acesso e autenticação;
- *Parser* e *cache* de arquivos de configuração;
- Busca, paginação e ordenação;
- Gráficos e relatórios quantitativos;

³ Acessível pelo endereço: <http://php.net/>. *Personal Home Page*, em português: Página Pessoal. *PHP: Hypertext Preprocessor*, em português: *PHP: Pré-processador de hipertexto*

⁴ Acessível pelo endereço: <http://www.postgresql.org/>

⁵ Em português: Interface de Programação de Aplicativos

- Monitoramento de modificações – links RSS (*Really Simple Syndication*)⁶;
- *Asynchronous Javascript And XML* (*Extensible Markup Language*)⁷ ou Ajax;
- Controle de versão e auditoria;
- Sistema de *upload* de arquivos;
- Geração automática para impressão e PDF;
- Sistema de alertas;
- Sistema de *debug* e *log*;
- Salas de chat;
- XML *transformation*.

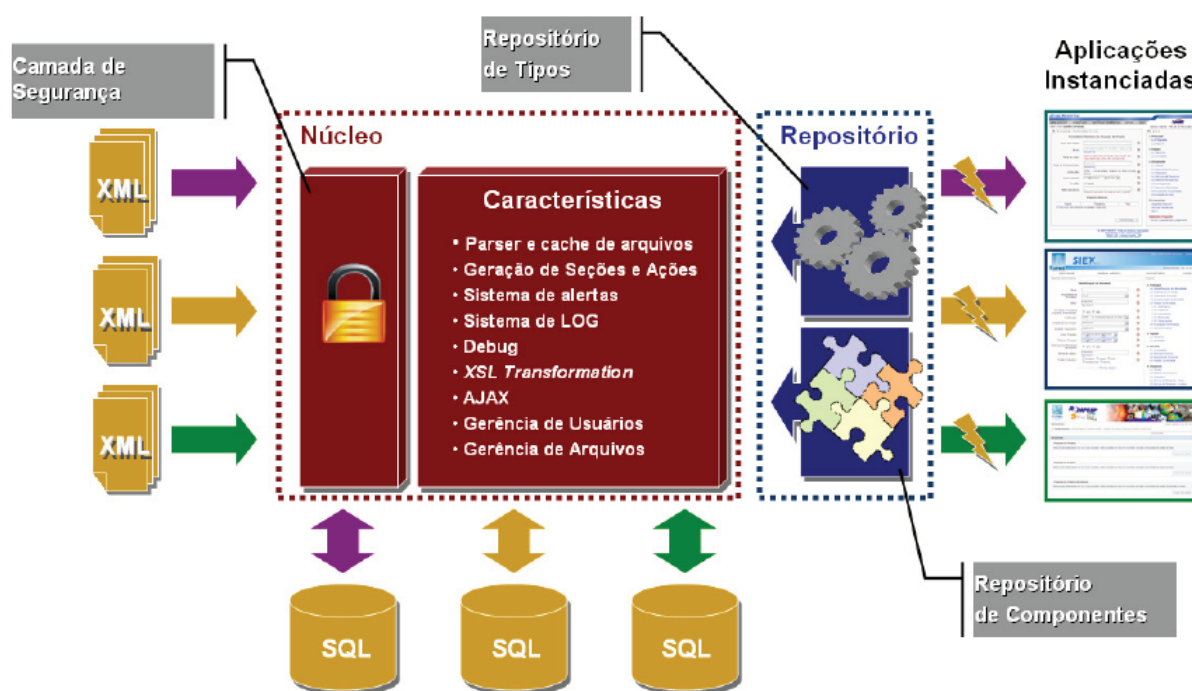


Figura 2 – Arquitetura do Framework Titan (CARROMEU, 2007).

Além do núcleo, existem também os repositórios de tipos e de componentes. O primeiro é composto por classes que tratam os tipos de dados do *framework*, realizando a conversão para suas classes específicas de modo que seja possível manipulá-los na aplicação. Os componentes são artefatos de códigos com funcionalidades específicas, sendo possível utilizar os componentes existentes no repositório, estendê-los ou criar novos (CARROMEU, 2007).

⁶ Em português: Sindicacão Realmente Simples

⁷ Em português: Linguagem de Marcação Extensível

A arquitetura do Titan utiliza conceitos de encapsulamento caixa-branca, recebendo como entrada um arquivo de marcação estendida (XML) e transformando-o em tempo de execução em um gerenciador de conteúdo, de modo que, se determinada funcionalidade não puder ser instanciada pelo *framework*, é possível codificar um novo componente para suportá-la.

2.4 Boas Práticas para Codificação

Depois de concluída a construção da Plataforma +Precoce, esta precisa ser mantida, assim, uma das preocupações do projeto é que o código gerado apresente boa legibilidade de modo a facilitar sua manutenção pelas equipes futuras. Para isso, foram considerados os conceitos apresentados por [Martin \(2011\)](#) e [Boswell e Foucher \(2012\)](#) para a codificação do simulador.

A ideia central para a escrita de bom código é que o mesmo seja fácil de entender, mas esse é um conceito difícil de ser definido. [Boswell e Foucher \(2012\)](#) chegaram ao “Teorema Fundamental da Legibilidade” depois de observarem e estudarem muitos códigos: “Códigos devem ser escritos de modo a minimizar o tempo necessário para sua compreensão”. Entender um código significa levar o leitor a ser capaz de fazer alterações, encontrar *bugs* e compreender a interação com o restante do sistema.

Sendo assim, boas práticas como: usar nomes significativos e informativos, priorizar estética de indentação e formatação, usar bons comentários, simplificar *loops* e lógica, priorizar alta coesão e baixo acoplamento, evitar repetição de código e tratamento de erros completo, dentre outras, são técnicas recomendadas para a produção de bom código e foram empregados no desenvolvimento da Plataforma +Precoce.

2.5 Boas Práticas de Desenvolvimento de Software

Ainda que o desenvolvimento deste projeto seja individual é interessante que boas práticas para o gerenciamento de projetos sejam seguidas de modo a organizar o trabalho e dinamizar a produção. Alguns preceitos relacionados a métodos ágeis foram utilizados neste trabalho, embora metodologias ágeis como Programação Extrema ou Scrum não tenham sido usadas.

[Shore e Warden \(2008\)](#) destacam que o desenvolvimento ágil não é um processo específico a ser seguido. Os autores se referem ao Manifesto Ágil como sendo a maneira canônica desta forma de pensar. Este manifesto é assinado por uma dezena de profissionais da área do desenvolvimento de software e enfatiza ([Manifesto Ágil, 2016](#)):

- “Indivíduos e interação mais que processos e ferramentas”;

- “Software em funcionamento mais que documentação abrangente”;
- “Colaboração com o cliente mais que negociação de contratos”; e
- “Resposta a mudanças mais que planejamento”.

Buscando estes valores, algumas práticas ágeis foram seguidas no desenvolvimento da Plataforma +Precoce. No entanto, combinar essas boas práticas e criar um método próprio para o desenvolvimento deste projeto não é um objetivo deste trabalho e nem algo facilmente alcançável (SHORE; WARDEN, 2008; COHN, 2011; KNINBERG, 2007). Sendo assim, a ideia seguida foi refinar o modo de trabalho de maneira interativa de acordo com as necessidades que surgiram, seguindo o ciclo PDCA (*Plan Do Check Act*)⁸ (SUTHERLAND, 2016). As principais práticas utilizadas serão destacadas no restante do capítulo.

2.5.1 Envolvimento do Cliente e Compromisso

Ainda que um projeto atenda a critérios como tempo, orçamento e escopo, pode ser que o mesmo não obtenha sucesso, pois não consegue atrair os usuários e por isso não agregam valor ao negócio (SHORE; WARDEN, 2008). Esta é uma preocupação que deve ser levada em consideração ao se desenvolver um produto e destacada em alguns dos princípios do manifesto ágil, tais como: “Pessoas relacionadas aos negócios e desenvolvedores devem trabalhar em conjunto e diariamente, durante todo o curso do projeto” (Manifesto Ágil, 2016).

No Scrum existe o papel de “Dono do Produto” que é o responsável por decidir qual trabalho deve ser feito e manter a ordem da lista de pendências (*backlog*), garantindo assim o foco no objetivo correto. Esse papel deve ser exercido por alguém que tenha profunda conexão com o cliente e seja capaz de dar a equipe de desenvolvimento o retorno dos clientes, suas opiniões e pensamentos. Esta pessoa deve ser capaz de se colocar no lugar de quem quer que seja e conseguir obter valor daquilo que está sendo construído (SUTHERLAND, 2016; KNINBERG, 2007; COHN, 2011).

A ideia do envolvimento é propiciar um ambiente em que *stakeholders*, gerentes, gestores, desenvolvedores, enfim, todos que estejam relacionados ao projeto assumam um papel de compromisso e não sejam apenas meros participantes (SUTHERLAND, 2016).

⁸ *Plan* (planeje), *Do* (faça), *Check* (verifique), *Act* (aja). Inicie com o planejamento do que será feito e então faça-o, em seguida verifique o que foi feito e avalie o que deu certo, o que deu errado, o que pode ser melhorado e então mude seu modo de trabalho (SUTHERLAND, 2016).

2.5.2 *Product Backlog*

O Scrum organiza o trabalho a ser feito através do *product backlog*, que é uma lista priorizada de atividades (requisitos, estórias, coisas que o cliente deseja) escritas utilizando a linguagem do cliente. Resumidamente, é uma lista de tudo que precisa ser feito para transformar a visão do produto em realidade. O *backlog* do produto é mantido pelo dono do produto e deve ser dinâmico de modo a permitir que itens possam ser adicionados, removidos, modificados, repriorizados a cada “*sprint*”, e com isso entregar valor mais rapidamente (COHN, 2011; KNINBERG, 2007; SUTHERLAND, 2016).

2.5.3 *Sprint*

O Scrum segue uma abordagem de desenvolvimento iterativa e incremental (COHN, 2011), com ciclos pequenos de desenvolvimento, que permite a identificação de problemas rapidamente e a correção do problema o quanto antes, eliminando, deste modo, tudo que constitui desperdício de esforço (SUTHERLAND, 2016).

É importante que cada *sprint* tenha uma duração rigorosa e que não varie durante o desenvolvimento do projeto (de uma até no máximo seis semanas), seja planejada visando um objetivo a ser atingido e de acordo com a capacidade e velocidade da equipe e, ao final, se obtenha como resultado software pronto e funcionando que agregue valor ao projeto (COHN, 2011; KNINBERG, 2007; SUTHERLAND, 2016).

2.5.4 Ambiente de Desenvolvimento Informativo

Um dos objetivos do Scrum é a transparência. Saber em que cada membro da equipe está trabalhando e como o projeto avança para sua conclusão é importante. Esses objetivos foram transferidos para o quadro Scrum, que basicamente define três colunas para a organização da *sprint*, a saber: “Pendente”, “Fazendo” e “Feito”. Com esta abordagem, sempre que um desenvolvedor assume uma tarefa todos sabem quem está trabalhando nela e quando foi concluída. Como o quadro possui tudo que precisa ser feito na *sprint* fica simples e visual para qualquer pessoa saber a evolução do projeto e melhorar a confiança dos envolvidos (SHORE; WARDEN, 2008; SUTHERLAND, 2016).

2.6 Considerações Finais

Os termos e conceitos apresentados neste capítulo ajudaram a contextualizar a área de aplicação da ferramenta de simulação da Plataforma +Precoce. Neste sentido, vale ressaltar que a utilização de sistemas melhorados para produção de bovinos de corte impactará de maneira positiva na sustentabilidade econômica e ambiental das propriedades

rurais. A ferramenta de simulação apoiará os produtores na tomada de decisão sobre qual o melhor sistema utilizar em sua propriedade.

3 Mapeamento Sistemático

Antes de iniciar o projeto de pesquisa, é importante que se conheça o contexto atual do assunto objeto de pesquisa (o “estado da arte”), sendo assim, uma revisão de literatura não é suficiente, o ideal é que ela seja completa para que apresente maior valor científico. Para tanto, pesquisas já realizadas devem ser analisadas objetivando conhecer seu contexto e estudar seus resultados. É dado o nome de “revisão sistemática da literatura”, o ato de identificar, avaliar e interpretar pesquisas relevantes, suas questões e fenômenos de interesse de maneira imparcial e de modo a ser repetível, utilizando uma metodologia confiável, rigorosa e auditável (KEELE, 2007).

Realizando a revisão sistemática é possível, por exemplo, encontrar lacunas em uma pesquisa para sugerir trabalhos futuros; apresentar os resultados atuais da área de pesquisa; resumir evidências empíricas de métodos ou tecnologias da área de Engenharia de Software (ES) apresentando seus benefícios e limitações; fornecer o embasamento necessário para novas atividades de pesquisa, ou contrapor hipóteses teóricas e gerar novas hipóteses de trabalho. Inicialmente a prática da revisão sistemática de literatura foi empregada na medicina baseada em evidências, ao passo que muitas diretrizes relacionadas à sua aplicação em ES refletem esta origem (KEELE, 2007).

Por ser mais extensa que uma simples revisão de literatura, a revisão sistemática torna menos provável que os resultados alcançados sejam tendenciosos. É possível ainda fornecer informações sobre efeitos de algum fenômeno através da ampla gama de configurações e métodos empíricos; no caso dos estudos apresentarem resultados consistentes a revisão provê evidências de que o fenômeno é robusto e transferível ao passo que se os resultados forem inconsistentes as fontes de variação podem ser estudadas. Para o caso de estudos quantitativos, a revisão sistemática possibilita que dados sejam combinados utilizando técnicas de meta-análise, aumentando assim a probabilidade de detectar efeitos reais, que estudos individuais não conseguiriam detectar. Todavia, para sua realização é necessário muito mais esforço do que as revisões de literatura tradicionais (KEELE, 2007; PETERSEN et al., 2008).

Os estudos individuais que contribuem para uma determinada área de pesquisa são chamados de estudos primários, ou seja, seu objetivo é investigar uma determinada questão. Os estudos secundários revisam estudos primários relacionados a uma questão de pesquisa, de modo a contribuir na síntese de evidências sobre esta questão. A revisão sistemática é uma forma de estudo secundário, que pode ser complementada com estudos terciários, que é uma revisão sistemática de revisões sistemáticas. O mapeamento sistemático, objeto deste capítulo, possui escopo mais abrangente e sua execução é recomendada quando

existem poucas evidências acerca do tema ou se o mesmo for muito amplo (KEELE, 2007; PETERSEN et al., 2008; BAILEY et al., 2007).

As orientações para a realização de uma revisão sistemática de literatura abordadas por Keele (2007), definem três fases: planejamento, execução e resultados. Estes passos foram executados neste trabalho conforme apresentado nas próximas seções.

3.1 Planejamento do Mapeamento Sistemático

O objetivo neste tópico é a realização de um mapeamento sistemático, sendo assim, as orientações definidas em Petersen et al. (2008) e em Bailey et al. (2007) serviram de base para a organização do planejamento. Um esboço geral e passos podem ser vistos na Figura 3.



Figura 3 – Processo para realização de mapeamento sistemático (adaptado de Petersen et al. 2008).

Bailey et al. (2007) destacam que a fase de planejamento do mapeamento sistemático é semelhante à da revisão sistemática. A etapa de execução do mapeamento sistemático tem seu principal foco nas três atividades a seguir:

- Identificação de pesquisas;
- Seleção de estudos primários; e
- Avaliação da qualidade do estudo.

A etapa de extração de resultados é muito mais ampla e voltada principalmente para a classificação e categorização dos dados extraídos.

O protocolo utilizado neste trabalho segue Petersen et al. (2008) e consiste em um processo adaptado que possui quatro etapas, cada qual produzindo um resultado (Figura 3).

A primeira etapa do processo consiste em definir as questões de pesquisa, ou seja, é necessário definir o escopo de avaliação sobre o que se objetiva responder com a execução do mapeamento.

A segunda etapa consiste em conduzir pesquisas em bases de publicações técnico-científicas de modo a identificar estudos primários. Para a construção das *strings* de busca é comum utilizar palavras-chave relacionadas às questões de pesquisa que se deseja responder. A escolha das bases de pesquisa também deve ser criteriosa, por exemplo, se o objeto desta pesquisa está relacionado à pecuária de corte e Engenharia de Software, não é necessário buscar publicações em bases que cubram publicações relacionadas a áreas como ciência da saúde. É necessário adicionar bases cujo foco sejam publicações relacionadas a agropecuária e tecnologia. Ao final da segunda etapa é esperado que se mapeie todos os trabalhos encontrados relacionados às *strings* de busca.

A terceira etapa consiste em realizar a triagem dos trabalhos encontrados levando em consideração critérios para inclusão e exclusão daqueles que são relevantes para responder as questões de pesquisa.

Na quarta etapa os trabalhos relevantes são organizados dentro de um esquema de classificação.

Na quinta etapa é relatado o resultado do mapeamento sistemático com vista a criar o mapa sistemático para responder as questões propostas na etapa 1 e então tecer conclusões acerca do mapeamento sistemático.

No decorrer desta seção o protocolo para o mapeamento sistemático foi definido e serão apresentados os resultados obtidos em cada etapa.

3.1.1 Questões de Pesquisa

Considerando o processo apresentado na Figura 3, a primeira etapa do mapeamento é definir as questões de pesquisa. Petersen et al. (2008) e Bailey et al. (2007) expõem que o principal objetivo do mapeamento é fornecer uma visão geral da área, identificando a quantidade, tipos de pesquisa e seus resultados, ao passo que, identificar os fóruns em que as pesquisas desta área têm sido publicadas se torna relevante. Esses objetivos interferem diretamente na formulação das questões de pesquisa. Seguindo esta perspectiva, as questões de pesquisa (QP) levantadas no presente estudo são:

QP1: Existe software que realiza a simulação de sistemas melhorados de produção de bovinos de corte?

QP2: Em quais editoras estão disponíveis publicações relacionadas a simuladores de sistemas de produção de bovino de corte?

3.1.2 Bases de Busca

Chen, Babar e Zhang (2010) mencionam que a pesquisa de publicações em bases de busca é o método dominante para a seleção de estudos utilizados em um mapeamento ou revisão sistemática de literatura, sendo a escolha dos mecanismos de busca usualmente feitas com base em preferências ou experiências pessoais. No entanto os autores destacam em seu estudo que a escolha baseada em evidências do mecanismo de busca é essencial para que se encontrem estudos primários mais relevantes e sugerem que as buscas sejam executadas em mais de um mecanismo, de modo a conseguir maior cobertura possível de pesquisas relevantes.

As bases de busca podem ser divididas em duas categorias segundo Chen, Babar e Zhang (2010): motores de indexação e sites de editores. O primeiro realiza a indexação de trabalhos publicados em bases de várias editoras, ao passo que os sites de editores fazem a indexação apenas de sua base de dados. Neste trabalho foram incluídos dois mecanismos de busca mencionados no referido trabalho, a saber o *SCOPUS* (se enquadra na categoria de motores de indexação) e o *ACM Digital Library* (se enquadra nas duas categorias). A relação de todas bases selecionadas pode ser vista na Tabela 3.

Na conclusão de seu trabalho, Chen, Babar e Zhang (2010) mencionam que na prática outros fatores devem ser levados em conta para a seleção das bases de dados eletrônicas. Outros motores de busca foram adicionados visando suprir peculiaridades do presente trabalho. Foram definidas as seguintes bases:

- Periódicos da CAPES, por se tratar de um motor de indexação de referência nacional e que mantém parceria com várias instituições públicas, incluindo a UFMS; e
- Biblioteca Digital Brasileira de Teses e Dissertações – BDTD, que reúne em um único portal teses e dissertações defendidas em todo o país e da qual a UFMS também faz parte.

Uma das áreas de concentração da pesquisa está relacionada a tecnologias computacionais para a pecuária. Por este motivo, houve a preocupação em levantar bases de buscas relacionadas a esta área de concentração. A Embrapa disponibiliza vários serviços de acesso às suas bases de publicações. Foram selecionadas quatro:

- BDP@ – Bases de Dados de Pesquisa Agropecuária, que apresenta trabalhos impressos e digitais;
- Infoteca-e – Informação Tecnológica em Agricultura, que apresenta informações de tecnologias produzidas pela Embrapa (de cunho técnico);

- Alice – Acesso Livre à Informação Científica da Embrapa, que possibilita o acesso a informação digital científica produzida pela Embrapa; e
- Publicações Embrapa Gado de Corte, que reúne trabalhos produzidos por esta unidade.

Tabela 3 – Bases de buscas utilizadas.

Base de Dados Eletrônicos	Endereço
SCOPUS	< https://www.scopus.com/ >
ACM Digital Library	< http://dl.acm.org/ >
Periódicos CAPES	< http://www.periodicos.capes.gov.br/ >
BDTD – Biblioteca Digital Brasileira de Teses e Dissertações	< http://bdtd.ibict.br/ >
BDP@ – Bases de Dados de Pesquisa Agropecuária	< https://www.bdpa.cnptia.embrapa.br >
Infoteca-e – Informação Tecnológica em Agricultura	< http://www.infoteca.cnptia.embrapa.br/ >
Alice – Acesso Livre à Informação Científica da Embrapa	< http://www.alice.cnptia.embrapa.br/ >
Publicações Embrapa Gado de Corte	< https://www.embrapa.br/gado-de-corte/publicacoes >

3.1.3 *String* de Busca Padrão

Na Tabela 4 são exibidas palavras-chaves, sinônimos e termos relacionados às questões de pesquisa e ao tema do trabalho. Esses termos foram identificados e utilizados para a construção da *string* de busca padrão (Tabela 5).

Tabela 4 – Termos relevantes para a construção da *string* de busca padrão.

Termos de interesse
Simulador, “pecuária de corte”, “pecuária de precisão”, “bovinocultura de corte”, “bovinocultura de precisão”, bovino, bovinos, software, <i>program</i> , “ <i>computer system</i> ”, <i>simulator</i> , “ <i>computer simulation</i> ”, “ <i>beef cattle</i> ”, <i>livestock</i> , “ <i>cattle raising</i> ”, <i>bovine</i> , <i>cattle</i>

Tabela 5 – *String* de busca padrão.

<i>String</i>
(software OR program OR “computer system”) AND simulator AND (“beef cattle” OR livestock OR “cattle raising”)

3.1.4 Triagem - Critérios de Inclusão e Exclusão

Para realizar a triagem dos trabalhos o primeiro passo é definir critérios de inclusão (CI), ou seja, condições que o trabalho deve satisfazer e, critérios de exclusão (CE) que presentes excluem o trabalho dos resultados.

Os critérios de inclusão definidos são:

- CI1: O trabalho apresenta um software simulador (indicando: construção, uso, descrição, etc.) para sistemas de produção de bovinos de corte.

Os critérios de exclusão foram:

- CE1: O trabalho não está em inglês, espanhol ou português;
- CE2: O trabalho está relacionado a outro estudo primário do(s) mesmo(s) autor(es) – neste caso somente o mais recente será incluso;
- CE3: O trabalho já foi selecionado em outra base de busca eletrônica;
- CE4: O trabalho não está disponível *on-line* ou precisa de permissão específica para realizar *download*;
- CE5: O trabalho completo não está disponível gratuitamente;
- CE6: O trabalho está disponível somente na forma de resumo, apresentação de *slides* ou apenas uma pequena parte do trabalho é disponibilizada.

Com o intuito de organizar a triagem e deixá-la eficiente, foi definido um processo que consiste na aplicação de três filtros a serem executados nos trabalhos encontrados. A primeira filtragem baseou-se unicamente nos critérios de exclusão definidos. No segundo filtro, de posse dos trabalhos restantes, investigaram-se as palavras-chave, título e resumo para verificar se, de fato, o trabalho está relacionado ao tema desejado. Por fim, no terceiro filtro, os trabalhos que restaram foram completamente lidos para garantir sua relevância para o mapeamento sistemático.

3.1.5 Esquema de Classificação

Para o esquema de classificação foram definidas duas facetas comumente utilizadas em mapeamentos sistemáticos de ES. A primeira faceta trata do tipo de pesquisa encontrada. Utilizou-se a abordagem adaptada de [Wieringa et al. \(2006\)](#) que define as seguintes categorias:

- Pesquisa de Avaliação: implementa e avalia uma técnica na prática. Isto é, demonstra como a técnica foi utilizada e quais vantagens e desvantagens foram identificadas com sua aplicação;
- Proposta de Solução: uma proposição de solução para um problema é proposta, podendo ser nova ou extensão de uma técnica existente. Não é necessária validação

completa de seus benefícios e aplicabilidade, apenas uma prova do conceito, por um pequeno exemplo, ou boa linha de argumentação;

- Pesquisa de Validação: as técnicas investigadas não foram implementadas na prática, são trabalhos estudados em laboratório;
- Artigo Filosófico: esboçam um novo olhar sobre algo que já existe, estruturando um novo quadro conceitual;
- Artigo de Opinião: expressam a opinião pessoal do autor sobre algo que julga errado ou certo, bom ou ruim, ou como julga que algo deve ser feito;
- Artigo de Experiência: deve ser a experiência pessoal do autor sobre como algo foi feito e não do porquê ter sido feito, contendo uma lista de lições aprendidas a partir de sua experiência.

Não necessariamente os trabalhos serão classificados em uma única categoria, mas essa organização sugere que os trabalhos possam ser corretamente classificados em apenas uma. Por exemplo, a pesquisa de avaliação e de validação devem utilizar um método de pesquisa confiável ao passo que as outras categorias permitem classificar pesquisas não empíricas (WIERINGA et al., 2006; PETERSEN et al., 2008). Petersen et al. (2008) sugerem está abordagem por ser fácil de utilizar, não sendo necessário avaliar cada um dos artigos em detalhes para classificá-lo, como é feito em um mapeamento sistemático.

A segunda faceta aborda o tipo de contribuição fornecido pela publicação. Foi utilizada uma abordagem adaptada de Mujtaba et al. (2008), tendo em vista o critério de inclusão estabelecido:

- Método ou Processo: descreve regras de como cada coisa foi feita ou ainda descreve o fluxo de trabalho, as atividades e ações realizadas em cada etapa da construção do software;
- Modelo: descreve o modelo utilizado para a construção do software;
- Ferramenta: descreve somente o software construído.

O trabalho pode prestar contribuição em mais de uma categoria, neste caso a classificação ficará a cargo da mais geral, na seguinte ordem: Método ou Processo, Modelo e Ferramenta.

3.2 Execução do Mapeamento Sistemático

Nesta etapa o objetivo é encontrar estudos e publicações que respondam as questões de pesquisa de modo satisfatório.

Este processo pode ser executado de forma manual ou de forma automática (KEELE, 2007). Neste trabalho optou-se por realizar buscas automatizadas, isto é, executando buscas utilizando bases de busca eletrônicas (CHEN; BABAR; ZHANG, 2010).

No restante desta seção é detalhada a condução da pesquisa. Inicialmente a *string* padrão foi adaptada para cada mecanismo de busca, a seguir, os trabalhos encontrados passaram pelo processo de triagem e, por fim, os trabalhos que restaram são àqueles relevantes para responder à QP1 e QP2.

3.2.1 *Strings* de Busca Específicas

Cada mecanismo de busca teve sua *string* relacionada montada com base na *string* padrão (Tabela 5) mesclada, conforme necessidade, aos termos destacados na Tabela 4. As *strings* apresentam pequenas diferenças com vistas a atender as particularidades de cada base de busca eletrônica e são exibidas na Tabela 6. Foi encontrado um total de setenta e um trabalhos como resultado destas buscas.

3.2.2 Triagem dos Trabalhos.

Um resumo do processo de triagem pode ser visto na Figura 4. A execução de cada um dos filtros é discutida a seguir.

3.2.2.1 Filtro 1

O resultado apresentado na execução do Filtro 1 está descrito no Apêndice A.

Após a realização da pesquisa, os trabalhos foram baixados e organizados em um inventário de conteúdo¹ para facilitar a manipulação. Para verificação dos critérios de exclusão, cada trabalho foi aberto e verificado se atendia algum dos critérios estabelecidos. Os trabalhos e critérios atendidos estão listados no Apêndice A. Deve-se notar que muitos dos trabalhos das bases Infoteca-e e Alice estavam disponíveis como publicações em anais, resumos e relatórios, ou seja, composições de vários trabalhos. Deste modo, o critério CE6 foi levado em consideração para triagem. As publicações que foram mantidas, continham trabalhos que apresentavam no mínimo a seguinte estrutura: resumo, materiais e métodos e conclusão.

Ao todo, vinte e quatro trabalhos foram excluídos (de um total de setenta e um), resultando em quarenta e sete trabalhos a serem analisados na etapa seguinte.

¹ Optou-se por organizar os arquivos com a utilização de um inventário de conteúdo (ferramenta definida na arquitetura de informação) feito na ferramenta Microsoft Excel 2013, em detrimento a outras ferramentas a esse propósito como por exemplo o Mendeley (<https://www.mendeley.com/>). Este inventário consiste em um planilha eletrônica com a relação de todos os trabalhos encontrados e algumas colunas de controle, ex.: número de identificação único, nome do arquivo baixado, link de origem, base de busca, etc.

Tabela 6 – Total de trabalhos encontrados em cada base e suas respectivas *strings* de busca.

Base de Dados Eletrônicos	String de Busca	Número de Trabalhos Encontrados
SCOPUS	(software OR program OR “computer system”) AND simulator AND (“beef cattle” OR livestock OR “cattle raising”)	12
ACM Digital Library	(software OR program OR “computer system”) AND (simulator OR “computer simulation”) AND (“beef cattle” OR livestock OR “cattle raising”)	2
Periódicos CAPES	(software OR program OR “computer system”) AND simulator AND (“beef cattle” OR livestock OR “cattle raising”)	13
BDTD – Biblioteca Digital Brasileira de Teses e Dissertações	(software OR program OR “computer system”) AND simulator AND (“beef cattle” OR livestock OR “cattle raising”) AND (bovine OR cattle)	20
BDP@ – Bases de Dados de Pesquisa Agropecuária	simulador AND ((pecuária OR bovinocultura) AND (“de corte” OR “de precisão”))	5
Infoteca-e – Informação Tecnológica em Agricultura	(software OR “sistema computacional”) AND simulador AND ((pecuária OR bovinocultura) AND (“de corte” OR “de precisão”)) AND (bovino OR bovinos)	7
Alice – Acesso Livre à Informação Científica da Embrapa	(software OR “sistema computacional”) AND simulador AND ((pecuária OR bovinocultura) AND (“de corte” OR “de precisão”)) AND (bovino OR bovinos)	10
Publicações Embrapa Gado de Corte	(software OR “sistema computacional”) AND simulador AND ((pecuária OR bovinocultura) AND (“de corte” OR “de precisão”)) AND bovino	2
		Total - 71

3.2.2.2 Filtro 2

A segunda etapa do processo de triagem consistiu em ler título, resumo e palavras-chaves de cada trabalho resultante do filtro 1, título, sumário, apresentação e introdução, quando disponíveis, no caso de livro ou capítulo de livro, para verificar se o trabalho atendia ao critério de inclusão. Em caso de dúvidas, o trabalho foi mantido para a próxima etapa do processo de triagem.

Os trabalhos T63, T68 e T69² tratavam de publicações agrupadas de diversos trabalhos que não foram excluídos no filtro 1. Deste modo o filtro 2 foi aplicado para todos artigos da publicação. T63 continha cento e oitenta e dois artigos, T68 continha cento e

² Referência ao ID do referido trabalho. A consulta pode ser feita no Apêndice A.

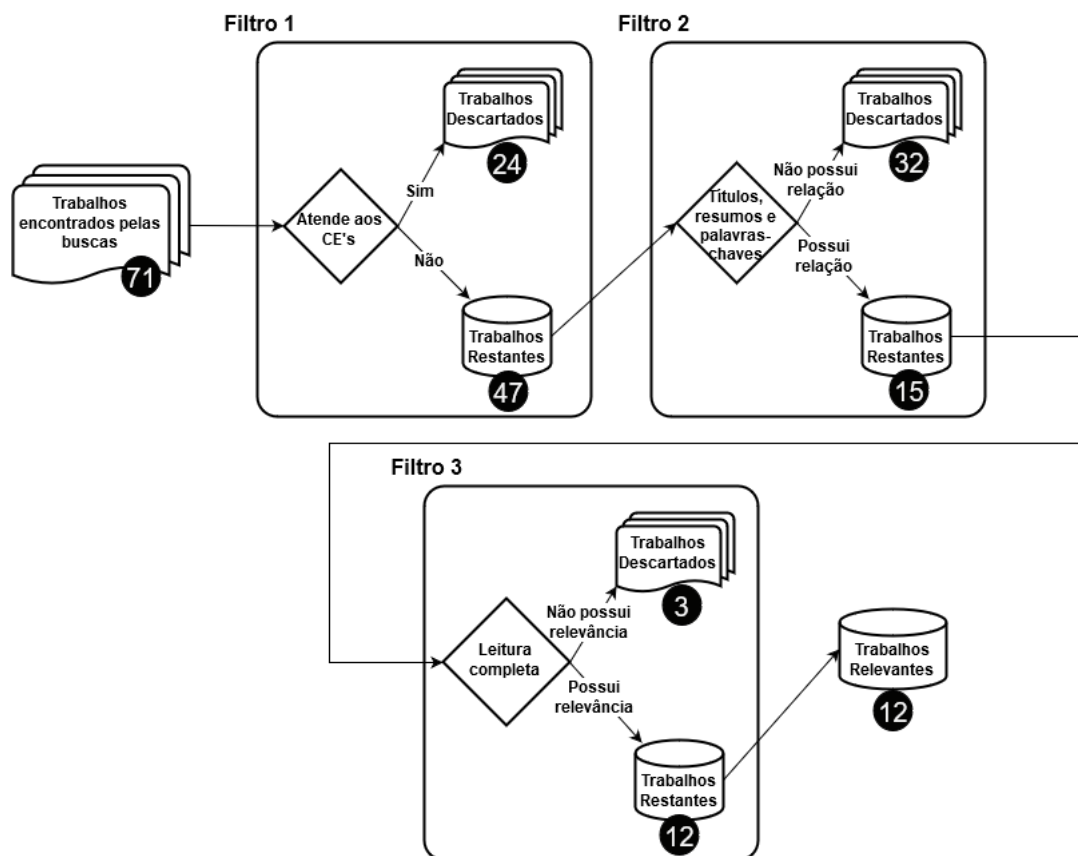


Figura 4 – Representação do processo de triagem utilizado para definir os trabalhos relevantes.

sessenta e um e T69 trinta e oito artigos. Apenas o T68 apresentou um artigo que passou pelo filtro.

Ao todo foram removidos trinta e dois trabalhos, restando assim, nesta etapa, quinze trabalhos. Alguns trabalhos se enquadraram em mais de um critério de exclusão, como pode ser visto no Apêndice A, porém na tabela é contabilizado apenas o primeiro critério de exclusão apresentado.

3.2.2.3 Filtro 3

A terceira etapa do processo consistiu em ler todos os trabalhos e realizar um resumo breve sobre cada um deles para verificar se os mesmos atendiam ao critério de inclusão CI1. O resultado apresentado está descrito no Apêndice B.

Nesta etapa do processo de triagem foram removidos três trabalhos. Um resumo geral de todos os totalizadores apresentados em cada etapa do processo de triagem pode ser visto na Tabela 7³. O processo de triagem foi finalizado com doze trabalhos que atendem aos critérios de exclusão e ao critério de inclusão, sendo então, estes doze trabalhos o

³ Alguns trabalhos se enquadraram em mais de um critério de exclusão, porém nesta tabela é contabilizado apenas o primeiro critério de exclusão apresentado.

resultado da seleção final do processo de triagem.

Tabela 7 – Resumo do processo de triagem.

Base de Busca	Resultados	Aplicação dos Critérios de Exclusão						Inclusão	Inclusão	Trabalhos Resultantes
		1º etapa						2º etapa	3º etapa	
		CE1	CE2	CE3	CE4	CE5	CE6	CI1	CI1	
SCOPUS	12	-	-1	-	-1	-1	-3	-5	-	1
ACM Digital Library	2	-	-	-	-	-	-	-2	-	0
Periódicos CA-PES	13	-	-	-6	-	-	-	-5	-	2
BDTD	20	-	-	-	-1	-	-	-11	-3	5
BDP@	5	-	-	-	-4	-	-1	-	-	0
Infoteca-e	7	-	-	-	-	-	-4	-3	-	0
Alice	10	-	-	-	-	-	-2	-5	-	3
Publicações Embapa Gado de Corte	2	-	-	-	-	-	-	-1	-	1
Total	71	-0	-1	-6	-6	-1	-10	-32	-3	12

3.2.3 Keywording

Petersen et al. (2008) define um processo que chama de *keywording* para fazer a organização dos trabalhos de acordo com o esquema de classificação definido. O processo consiste em ler os resumos dos trabalhos resultantes do processo de triagem e extrair palavras-chaves significativas ao contexto, quando necessário pode-se ler as seções de introdução e conclusão. Através dessas palavras-chaves é feita sua definição no esquema de classificação.

Esta abordagem foi utilizada na classificação do mapeamento realizado neste trabalho, com a diferença que na etapa de triagem os trabalhos foram lidos integralmente e, o resumo feito para cada trabalho auxiliou em sua correta classificação. As categorias elencadas para o esquema de classificação já estavam previamente definidas com base nos trabalhos de Wieringa et al. (2006) e Mujtaba et al. (2008) e não foram atualizadas após análise dos trabalhos. O resultado desse processo pode ser visto na Tabela 8.

3.3 Resultados do Mapeamento Sistemático

Concluído o passo de execução do mapeamento sistemático foi possível desenhar o mapa sistemático, responder as questões elencadas na etapa de planejamento e tecer comentários acerca dos resultados.

Na Figura 5 é exibido um gráfico de bolhas em que é possível observar o mapa sistemático obtido após a execução do *keywording* em relação ao esquema de classificação definido. Pela análise da faceta tipo de pesquisa é possível observar que 50% das publicações utilizam métodos empíricos para realizar avaliação ou validação do modelo ou ferramenta

Tabela 8 – Visão geral do mapa sistemático.

Trabalho	Tipo de Pesquisa	Tipo de Contribuição
T3 (MARCOS et al., 2011)	Artigo de Experiência	Método ou Processo
T19 (HOLZWORTH et al., 2014)	Proposta de Solução	Modelo
T20 (GREINER et al., 2014)	Pesquisa de Avaliação	Modelo
T28 (PEREIRA, 2015)	Pesquisa de Validação	Ferramenta
T31 (FONTOURA JÚNIOR, 2008)	Pesquisa de Validação	Modelo
T32 (MEIRELLES, 2005)	Pesquisa de Avaliação	Modelo
T37 (BONATTI, 2009)	Pesquisa de Validação	Modelo
T40 (SANTANA, 2012)	Pesquisa de Validação	Ferramenta
T60 (COSTA; BARBOSA, 1997)	Artigo de Opinião	Ferramenta
T61 (MASSRUHÁ et al., 2014)	Artigo de Filosófico	Ferramenta
T68 (VARELA; BUSSONI; DIEGUEZ, 2015)	Proposta de Solução	Ferramenta
T70 (COSTA et al., 2004)	Artigo de Experiência	Ferramenta

apresentado. Apenas um trabalho contribui com métodos ou processos para construção de software de simulação para sistemas de produção de gado de corte.

Em relação as questões de pesquisa, a resposta para QP1 foi positiva. Dos trabalhos primários selecionados, 12 atendem os critérios definidos e que mencionam softwares com propósito de simulação de sistema de produção para bovinos de corte⁴:

- O APSIM (T19) que apresenta o GRAZPLAN para simulação de ruminantes bovinos, de alcance global, utilizado em países como Austrália, EUA, Índia e China;
- PPFS (T20);
- IFSM (T28);
- RURAL FAZPEC (T31);
- Embraptec (T40);
- Gerenpec (T70);
- Softwares desenvolvidos com ferramentas de apoio à simulação tais como Stella e Arena (T37 e T60);
- Os trabalhos T3, T32, T61 e T68 não apresentam o nome do software que realiza a simulação.

Em resposta a QP2, as editoras que mais contribuíram com trabalhos no processo de mapeamento sistemático foram Embrapa com três trabalhos e Elsevier com dois trabalhos. Os demais contribuíram com um trabalho cada. O resultado pode ser visto na Figura 6.

⁴ Para mais detalhes sobre cada um dos trabalhos, consulte o Apêndice B.

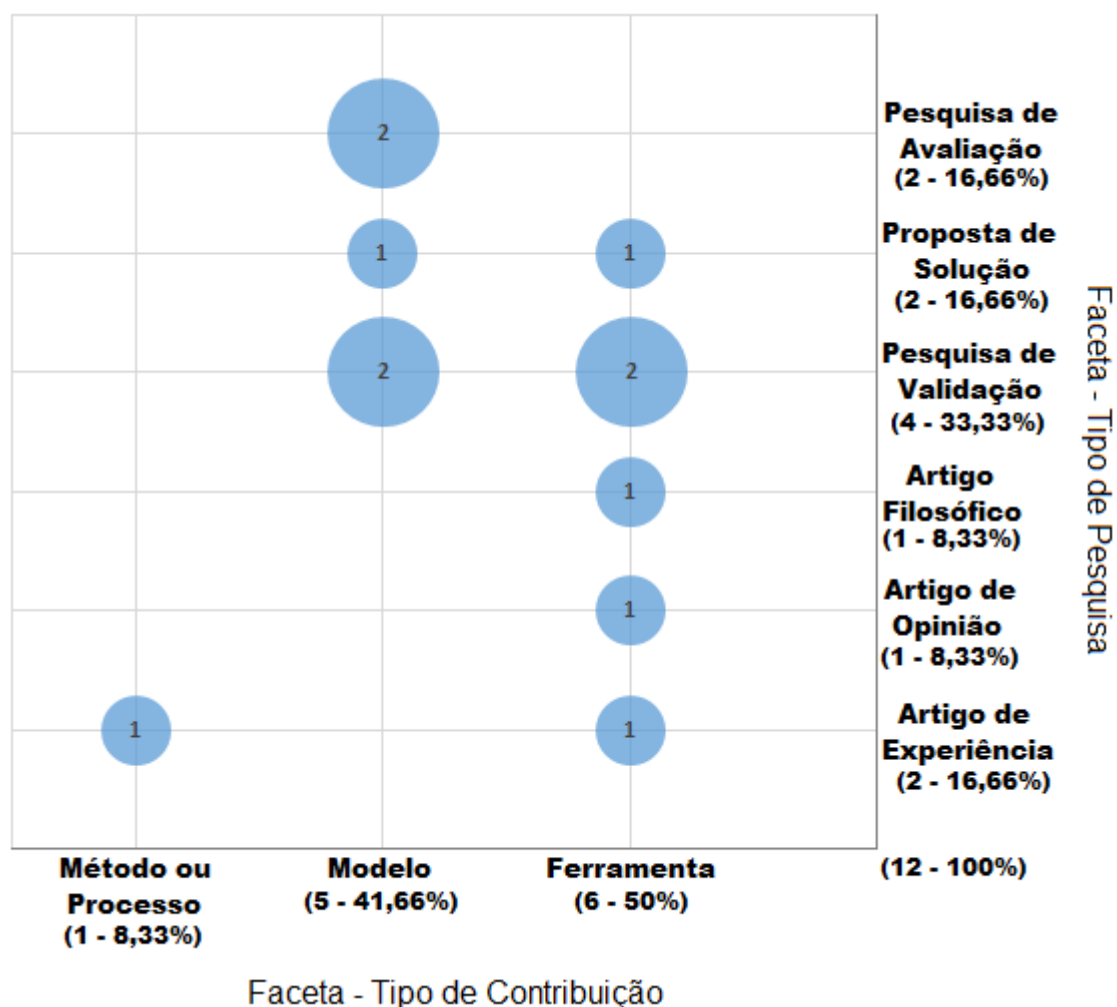


Figura 5 – Mapa Sistemático - Cenário de softwares para simulação de sistemas de bovinos de corte.

No embasamento teórico foram destacados três softwares utilizados para simulação na área agropecuária. Este mapeamento contemplou todos, porém dois deles de maneira indireta. O artigo de [Holzworth et al. \(2014\)](#) é inteiramente dedicado ao APSIM, neste artigo é descrito seus módulos e traçado um panorama global de uso e publicações científicas relacionadas. Os softwares PaSim e Invernada são apenas mencionados no Capítulo 13 da publicação de [Massruhá et al. \(2014\)](#), porém, não são dados detalhes a respeito de como, ou qual tipo de simulações realizam.

A base de busca BDP@ possui publicações a respeito do software Invernada, porém são poucas e em sua maioria disponibilizadas via CD-ROM. Para chegar a essas publicações a *string* de busca padrão teria que ser bastante alterada. No caso de tentar aumentar o detalhamento da *string* alguns trabalhos com potencial relevância não seriam contemplados, ao passo que, diminuindo o detalhamento da *string*, o número de trabalhos retornados seria muito grande, o que iria aumentar sobremaneira a complexidade da etapa de triagem. Outra solução possível seria adicionar esses trabalhos utilizando buscas manuais, ou outras



Figura 6 – Editoras que possuem publicações relacionadas a simuladores de sistemas de produção de bovino de corte QP2.

estratégias como *snowballing*, porém conflitaria com a forma planejada para execução das buscas (PETERSEN et al., 2008; CHEN; BABAR; ZHANG, 2010; PETERSEN; VAKKALANKA; KUZNIARZ, 2015).

No Apêndice B é feito um breve resumo dos trabalhos encontrados como resultado do mapeamento sistemático. O artigo relacionado ao APSIM é melhor descrito e também os motivos que levaram à inclusão da publicação que menciona o PaSim e o Invernada.

Dentre os softwares encontrados, um deles chamou atenção: o Embrapec. Desenvolvido pela Embrapa em meados de 2007, teve o desenvolvimento descontinuado. Desta descoberta iniciou-se uma discussão no grupo de trabalho do projeto +Precoce que identificou diferenças entre as propostas das plataformas:

- O Embrapec permite modelar e simular um sistema de produção a partir do início, considerando dados informados pelo usuário, ao passo que na proposta do +Precoce, o usuário escolhe dentre modelos prontos de sistemas recomendados;
- O Embrapec não apresenta sistemas de produção de referência, ao passo que um dos focos principais da Plataforma +Precoce é servir de repositório de sistemas de produção oriundos de pesquisas científicas e prospecção; e
- O Embrapec não descreve o sistema de produção em nível de fases e tecnologias por meio de protocolos, enquanto a Plataforma +Precoce irá disponibilizar protocolos detalhados de cada sistema de produção modelado, facilitando ao usuário produtor realizar a replicação do sistema para sua propriedade, com apoio técnico.

3.4 Considerações Finais

No decorrer desse capítulo todos os passos realizados para a execução do mapeamento sistemático foram detalhados. Com a confecção do mapa sistemático foi possível observar a cobertura da área de pesquisa e identificar a quantidade e tipos de pesquisa, no que tange o desenvolvimento de simuladores para produção de gado de corte. Este panorama não seria possível com a realização de uma revisão de literatura comum. No entanto vale destacar que o mapeamento sistemático exige esforço considerável, em maior parte para sua execução, mas também para a confecção de um planejamento adequado. [Petersen et al. \(2008\)](#) recomenda utilizar revisão sistemática de modo a complementar o mapeamento, ou seja, primeiro a área temática é estruturada com o mapeamento e, posteriormente, uma área específica é investigada com uma revisão sistemática. Com essa visão, nas sugestões de trabalhos futuros apresentada no Capítulo 6, são recomendadas algumas possíveis evoluções para revisões sistemáticas.

4 Especificação de Requisitos e Projeto do Simulador da Plataforma +Precoce

Neste capítulo são apresentados os requisitos funcionais, arquitetura da Plataforma, modelo computacional e metodologia de trabalho utilizada para o desenvolvimento do projeto.

4.1 Comunicação

Grande parte da comunicação inicial foi feita em reuniões presenciais ou via videoconferência com o pesquisador da Embrapa, Fernando Rodrigues Teixeira Dias, autor da documentação (Anexos A, B e C) e analista de negócios responsável, que atuou como intermediário na comunicação entre cliente e desenvolvedor, além de ser o autor do algoritmo de simulação. Outros membros do projeto assumiram os papéis de coorientador (Rodrigo da Costa Gomes), analista responsável (Camilo Carromeu) e desenvolvedores do Módulo Portal (Eduardo Buzinaro e Rodrigo A. de O. Siqueira). A orientadora Débora Maria Barroso Paiva e a pesquisadora Maria Istela Cagnin também foram envolvidas nas discussões.

Um ponto a se destacar é a localização dos integrantes da equipe (Campo Grande, Dourados e Corumbá - MS), sendo assim, uma das primeiras práticas adotadas foi a realização de reunião mensal entre todos membros do projeto, onde todos os envolvidos pudessem obter uma visão geral do progresso e discutir dificuldades e próximos passos, objetivos próximos à revisão de *sprint*, retrospectiva de *sprint* e planejamento de *sprint*, práticas da metodologia Scrum (COHN, 2011). Essas reuniões inicialmente ocorreram de forma presencial, mas, para viabilizar a participação de todos, o grupo optou por reuniões remotas utilizando a ferramenta *Google Meet*¹ ou *appear.in*².

Ao desenvolver módulos que dependiam de outros, houve bastante comunicação entre os desenvolvedores. Assim, para facilitar a comunicação entre os desenvolvedores utilizou-se a ferramenta *Slack*³ que permite comunicação síncrona e assíncrona.

Para garantir um ambiente informativo quanto ao andamento do projeto e execução das tarefas, todas as tarefas de desenvolvimento e progresso eram informadas no *GitLab*⁴.

Durante a etapa de construção do simulador houve comunicação constante com o

¹ Acessível pelo endereço <https://meet.google.com/> ou via aplicativo *mobile Hangouts*

² Acessível pelo endereço <https://appear.in/>

³ Ferramenta *cloud-based*, acessível pelo endereço: <https://slack.com/>.

⁴ Acessível pelo endereço: <https://about.gitlab.com/>.

analista de negócios da Embrapa, com envolvimento eventual de outros membros da equipe. O analista de negócios acompanhava o progresso e execução das tarefas e rotineiramente, esclarecia dúvidas acerca das funcionalidades desenvolvidas. Além das ferramentas de comunicação citadas anteriormente, outras TICs foram amplamente utilizadas: e-mail, conversas telefônicas, *Skype*⁵ e *WhatsApp*⁶.

4.2 Visão Geral do Sistema

A Plataforma +Precoce é uma aplicação web que permite a usuários clientes a simulação e o acesso às informações de sistemas melhorados de produção de gado de corte, construídos a partir do conhecimento acumulado de experimentos da Embrapa Gado de Corte e parceiros. A aplicação deve ser flexível o suficiente para que seus usuários possam simular tais sistemas com dados o mais próximo possível de suas propriedades, de modo a colaborar com a produção de bovinos dentro do contexto de programas e parcerias que bonificam carcaças por qualidade e, por conseguinte, contribuir para melhorar a qualidade da carne brasileira. Além disso, busca fornecer uma ferramenta robusta para a avaliação de sistemas de produção e obter um banco de dados de sistemas de produção que possa beneficiar diversos produtores e diferentes setores.

O projeto de construção da Plataforma +Precoce encontra-se em desenvolvimento pela Embrapa Gado de Corte desde agosto de 2014, sendo assim, muitas atividades já haviam sido desenvolvidas antes do início deste projeto, por exemplo, a prospecção e coleta de dados de sistemas melhorados de produção que se almejam modelar na Plataforma. As atividades para o desenvolvimento da Plataforma +Precoce foram iniciadas em 2016, naquele ano já existia uma proposta geral para a Plataforma (Figura 7 e Anexos A, B e C)⁷. Este trabalho foca a construção do **Módulo Desenhista** e do **Módulo Simulador**.

A Plataforma +Precoce deverá suportar diversos modelos de sistemas de produção de gado de corte e simplificar sua conversão para modelos passíveis de serem simulados computacionalmente. A gerência para estes modelos será propiciada pelo **Módulo Desenhista**, em que o ator denominado “Desenhista”, ficará responsável por modelar e gerenciar os sistemas de produção considerando o modelo computacional. Este papel será exercido por pesquisadores da Embrapa instruídos para esta função.

O **Módulo Simulador** é responsável pela execução das simulações requisitadas e, de forma geral, este módulo faz o carregamento da base de dados do modelo computacional escolhido, a execução das rotinas de cálculo de simulação e a entrega dos resultados na

⁵ Acessível pelo endereço: <https://www.skype.com/>.

⁶ Acessível pelo endereço: <https://www.whatsapp.com/>.

⁷ Os anexos serviram de guia inicial para o desenvolvimento do projeto, porém, durante seu curso o entendimento da Plataforma foi evoluindo, de modo que eles não refletem a especificação do trabalho entregue.

forma de indicadores de desempenho.

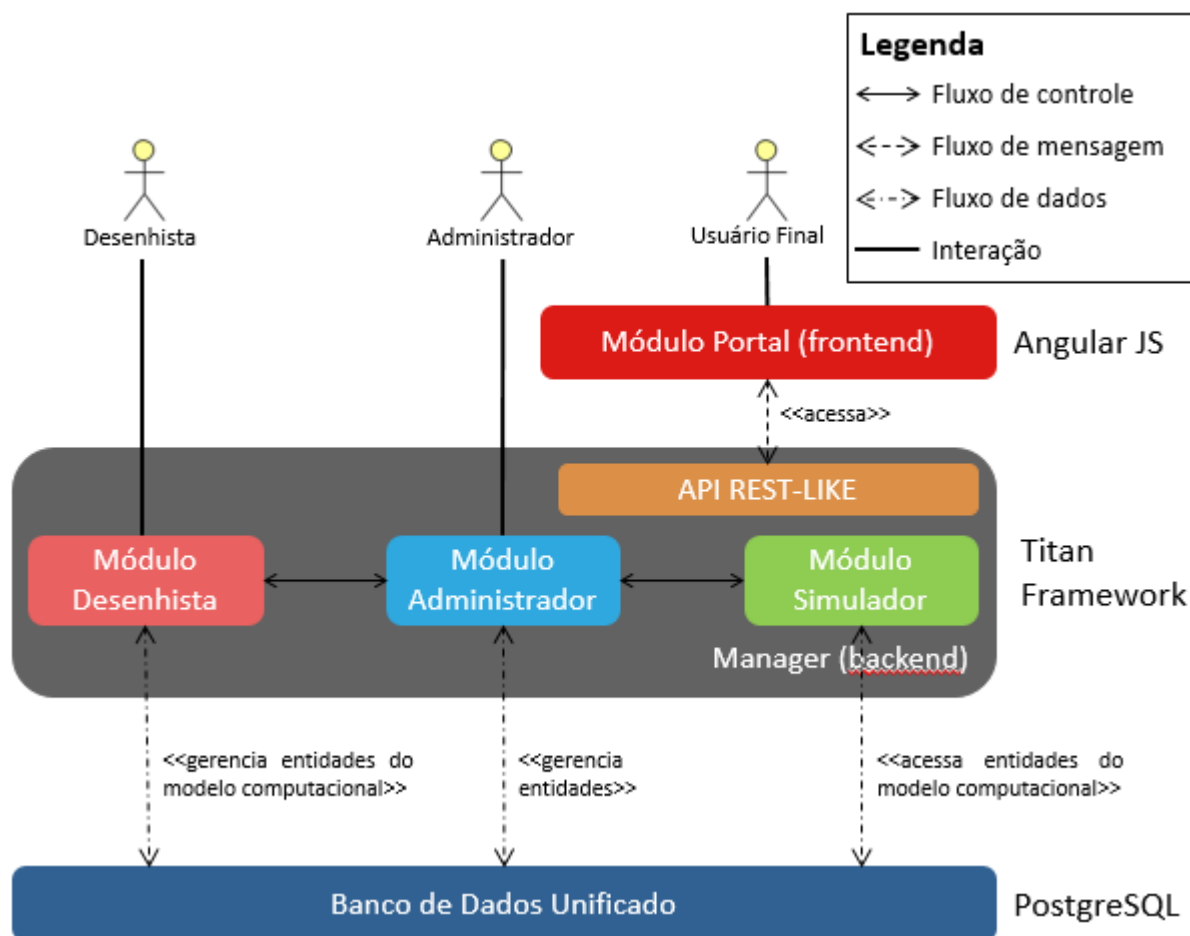


Figura 7 – Arquitetura da Plataforma +Precoce (GOMES, 2014).

O **Módulo Portal** é a interface com o usuário final (produtores rurais, técnicos, etc.). Neste módulo acontece:

- A seleção de sistema para simulação,
- A consulta ao protocolo do sistema e simulações já realizadas,
- A exclusão, edição ou cópia de simulações,
- O ajuste de valores de parâmetros para a simulação,
- A requisição de nova simulação e
- A exibição de indicadores e seus valores para a simulação (resultados da simulação) e a comparação com outras simulações do mesmo sistema ou não.

O **Módulo Administrador** gerencia as demais funcionalidades da Plataforma.

Manager corresponde à camada que abriga os **Módulos Desenhista, Administrador e Simulador** e corresponde a instância do Titan *Framework*.

O **Módulo Portal** foi desenvolvido utilizando a tecnologia *AngularJS*⁸. Os **Módulos Desenhista e Administrador** utilizam o Titan *Framework*, com base de dados em *PostgreSQL*. O **Módulo Simulador** foi escrito em *PHP* puro, sendo carregado no **Manager** via gerenciador de dependências *Composer*⁹. Vários serviços rodam no **Manager** (o Titan *Framework* oferece comunicação com outras aplicações via *API REST-like* (CARROMEU, 2016)) para responder às requisições do **Módulo Portal**.

4.3 Identificação de Requisitos e Projeto Inicial do Simulador da Plataforma +Precoce

O objetivo desta etapa foi elaborar o Diagrama de Entidade Relacionamentos (DER) que representasse o modelo computacional para o simulador e que contribuísse para o desenvolvimento do Módulo Desenhista.

Após a leitura e análise dos documentos disponíveis (Anexos A, B e C) iniciou-se a etapa de modelagem. As Figuras 8 e 9 apresentam esboços originados da discussão e apresentação de conceitos do problema pelo analista de negócios.

É possível observar à esquerda na Figura 8 uma abstração de um sistema de produção segundo uma representação adotada para o desenvolvimento da Plataforma. Em resumo, um sistema é uma caixa preta em que recursos entram por suas fronteiras (fluxos inferiores), são processados e geram como saídas coprodutos, resíduos ou emissões (fluxos a direita) e um produto (fluxo superior). À direita estão alguns exemplos de fórmulas para cálculo de coeficientes e durações. Estas fórmulas são em geral simples e envolvem as quatro operações matemáticas básicas (soma, subtração, multiplicação ou divisão), contendo apenas valores de parâmetros do modelo ou constantes numéricas. No canto direito superior é exibida a representação da estação de produção (àquela época denominada “transformação”), representada como um retângulo onde fluxos entram (lado inferior), fluxos de tratamento saem (lado direito) e apenas um único fluxo de produção sai (lado superior). Esse único fluxo de saída é denominado referência pois é ele que será utilizado para calcular os coeficientes dos demais fluxos (c_1 , c_2 , c_3 , c_4 e c_5). O pontilhado traçado em diagonal simboliza a divisão da ocupação dos recursos na estação, no caso, metade do tempo da estação será ocupado pelos recursos dos fluxos que entram e a outra metade pelos recursos dos fluxos que saem (essa é a divisão da ocupação convencional para cálculo do estoque).

⁸ Foi utilizada a versão 1.5. Acessível pelo endereço: <https://angularjs.org/>

⁹ Acessível pelo endereço: <https://getcomposer.org/>.

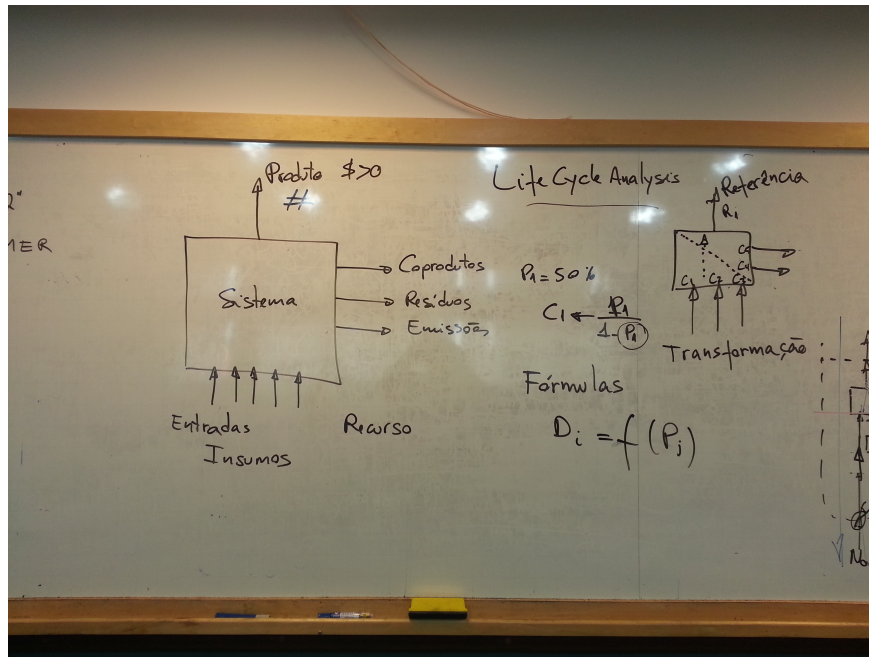


Figura 8 – Discussão sobre conceitos relativos ao problema [parte 1 de 2].

A Figura 9 é continuação da anterior, nela aparece a representação da estação de tratamento, semelhante à de produção com a diferença que o fluxo de referência é representado pelo fluxo de tratamento que entra na estação (esquerda). A caixa pontilhada exibida mais abaixo é a representação interna de um sistema de produção, composto por várias estações interligadas por fluxos. Os fluxos que ultrapassam a fronteira do sistema (pontilhado) são os mesmos descritos no início do parágrafo anterior.

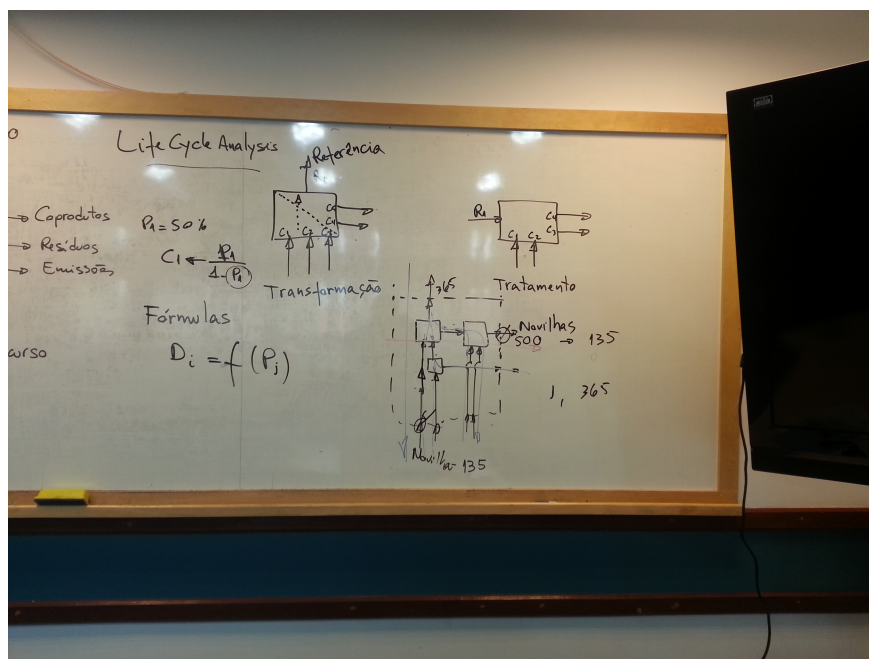


Figura 9 – Discussão sobre conceitos relativos ao problema [parte 2 de 2].

O primeiro desenho para o modelo computacional pode ser visto nas Figuras 10 e 11. É a representação da tradução dos conceitos exibidos nas Figuras 8 e 9 em termos de entidades e relacionamentos em banco de dados. Um sistema é representado por um modelo. Um modelo possui estações e fluxos de entrada e saída. Os fluxos carregam recursos, organizados em hierarquias de tipos. Estações utilizam fórmulas de duração e fluxos utilizam fórmulas de coeficientes que envolvem parâmetros. Os valores desses parâmetros são informados pelo usuário na hora de solicitar uma simulação. A execução da simulação calcula as quantidades e datas dos fluxos e, a partir destes, os valores dos indicadores de desempenho. Durante as iterações seguintes foram realizados refinamentos no modelo até chegar a um desenho passível de ser codificado explicado a seguir e exibido no DER da Figura 12.

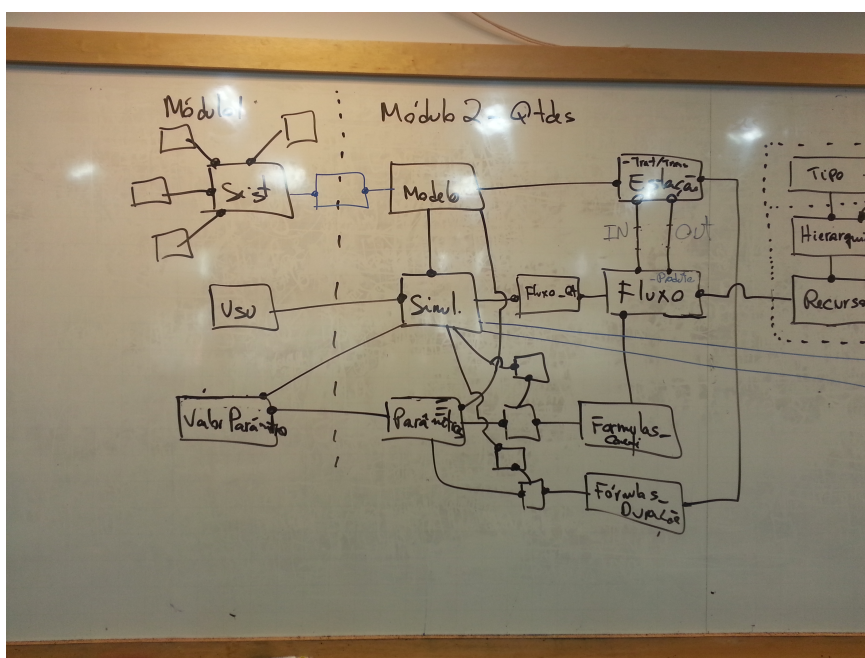


Figura 10 – Primeiro esboço do modelo computacional [parte 1 de 2].

Visando a facilitar quais são as funcionalidades implementadas para os módulos objetos deste trabalho foi elaborado um documento de requisitos exibido no Apêndice C. Essa lista exhibe apenas requisitos funcionais, ao passo que, os Anexos A, B e C exibem conceitos e requisitos.

4.4 Projeto do Simulador da Plataforma +Precoce

O DER apresentado na Figura 12 corresponde ao modelo computacional proposto para possibilitar a simulação (Módulo Simulador) e a gerência (Módulo Desenhista) de sistemas melhorados de produção de gado de corte. As entidades com nomes prefixados por “*simulator.*” são do escopo do modelo computacional. Aquelas prefixadas por “*precoce.*”

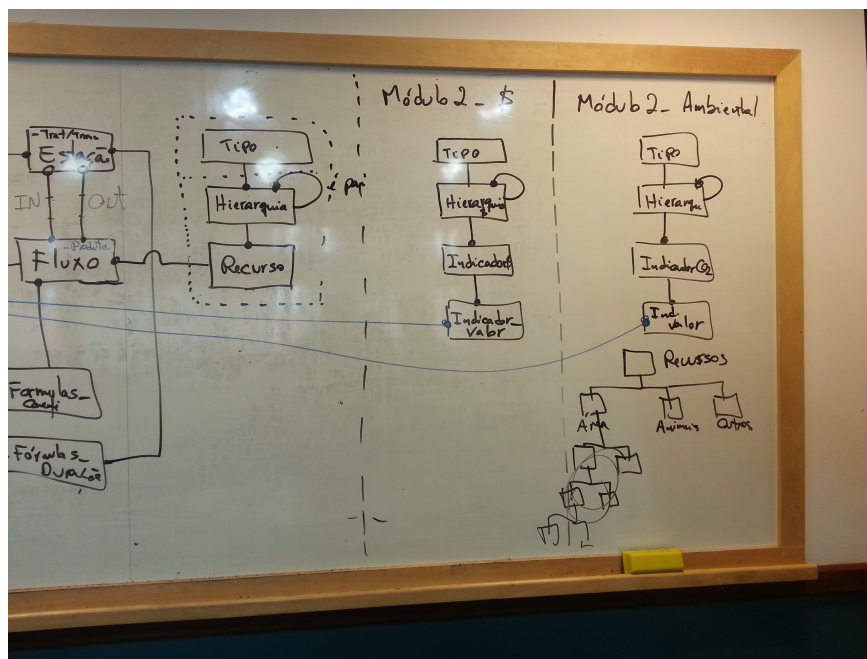


Figura 11 – Primeiro esboço do modelo computacional [parte 2 de 2].

fazem parte do escopo do Módulo Administrativo e Portal (fora do escopo deste trabalho) e são entidades fronteiriças aos Módulos Administrador, Portal e Simulador. Outros detalhes são apresentados no Dicionário de Dados (Apêndice D).

A base da solução computacional para a simulação corresponde a um modelo em grafo, representado pelas entidades “*simulator.node*” e “*simulator.flow*”, onde *node* representa os nós do grafo e *flow* as arestas que conectam esses nós. O cálculo das simulações é feito a partir do percurso deste grafo e do cálculo de fórmulas matemáticas relacionadas a essas entidades (relacionamentos com a entidade “*simulator.formula*”).

As arestas representam fluxos do sistema de produção, i.e., a movimentação de recursos (relacionamento com a entidade “*simulator.resource*”) que ocorre entre os nós (estações do sistema de produção). Para a simulação é importante conhecer quando as estações iniciam e terminam seu funcionamento e também a quantidade de recurso que vai de uma estação para outra.

Cada grafo está atrelado a um modelo de simulação, representado pela entidade “*simulator.model*”, que por sua vez é o responsável pela simulação dos sistemas de produção (entidade “*precoce.system*”). Quando o usuário aciona uma simulação no Módulo Portal, os dados preenchidos para cada um dos parâmetros são salvos na entidade “*precoce.simulation_parameter*” que são os valores utilizados nos parâmetros das fórmulas (“*simulator.formula_parameter*” utilizadas em “*simulator.node*” e “*simulator.flow*”). A entidade “*precoce.simulation*” guarda as informações da simulação executada pelo usuário para que seja passível de nova execução. Uma decisão tomada durante o desenvolvimento do projeto foi que os indicadores (resultado das simulações) não seriam persistidos em

base de dados, de modo que a cada recuperação de simulação, todo o processamento seja executado com base no sistema escolhido, seu modelo e valores de parâmetros da simulação, estes sim, persistidos na base de dados.

Uma adição que se tornou necessária para calcular os indicadores foi a criação da entidade “*simulator.stage*”, que define agrupamentos de nós a serem utilizados no cálculo de um indicador. Por exemplo, para se calcular indicadores relacionados a venda de produtos ou coprodutos é preciso saber quais nós representam vendas. Uma decisão para a estrutura em banco dos indicadores foi que os valores calculados não seriam persistidos, assim, quando o cliente solicitar uma simulação antiga, todos indicadores são novamente calculados. Esta é a razão de não existir estrutura para armazenar de indicadores.

A entidade “*simulator.infrastructure*”¹⁰ armazena informações sobre os itens de infraestrutura utilizados por determinado modelo. Três fórmulas são utilizadas para cálculo de seus atributos: valor inicial, valor residual e vida útil. Cada item de infraestrutura deve estar relacionado a uma categoria (“*simulator.infrastructurecategory*”).

4.5 Organização Inicial do *Product Backlog*

Foi definido juntamente com o pesquisador responsável pela elaboração do modelo matemático a organização e priorização das tarefas. Inicialmente notou-se que era necessário basicamente transformar as entidades do modelo computacional em *CRUD’s* (*Create, Read, Update and Delete*)¹¹. Uma decisão de projeto nesta etapa foi que os *CRUD’s* deveriam ser simples, utilizando sempre que possível os tipos e componentes disponibilizados nativamente pelo *Framework Titan*([CARROMEU, 2016](#)).

Foi possível obter um conjunto de estórias de usuário que segue os seis atributos para escrita de boas estórias ([COHN, 2004](#)):

- Independente;
- Negociável;
- De valor para o usuário;
- Estimável;
- Pequena;

¹⁰ Apesar de ter sido criado controle para gestão das entidades relacionadas a itens de infraestrutura no Módulo Desenhista, no Módulo Simulador essas entidades não foram utilizadas. Em primeiro momento essas entidades podem perder seu uso, uma vez que essa funcionalidade pode ser postergada ao cálculo de indicadores, entretanto, como não foram pensados indicadores relacionados à infraestrutura, optou-se por manter as entidades e deixar a cargo de implementação futura fazer uso ou remover as entidades do projeto.

¹¹ Em português: criar, ler, atualizar e deletar. São as quatro operações básicas para manipulação de dados em banco.

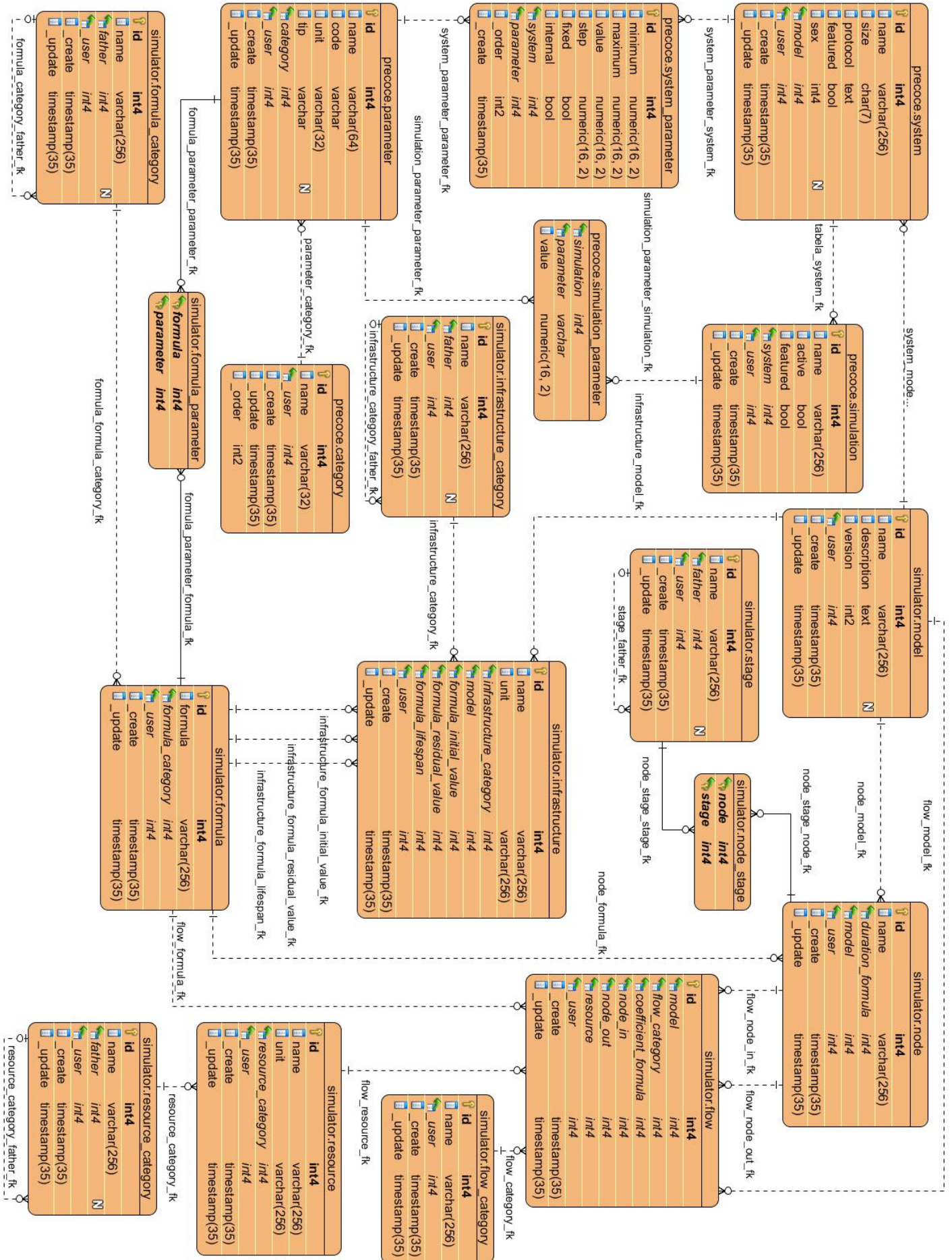


Figura 12 – Diagrama de Entidades e Relacionamentos do modelo computacional de simulação.

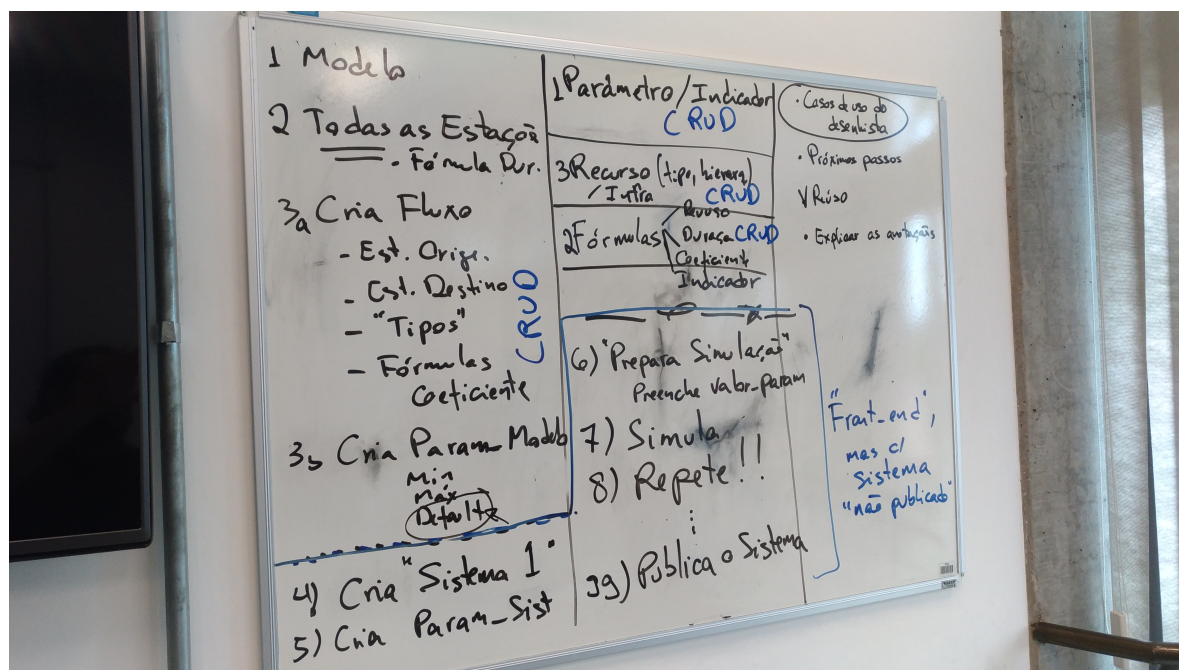


Figura 13 – Organização das tarefas a serem executadas.

- Testável.

O *Product Backlog* foi composto pelas seguintes histórias, elencadas por ordem de prioridade:

- Um usuário desenhista pode cadastrar, editar e excluir modelos;
- Um usuário desenhista pode cadastrar, editar e excluir parâmetros e suas categorias e hierarquias;
- Um usuário desenhista pode cadastrar, editar e excluir fórmulas de coeficiente, duração, valor inicial, valor residual ou vida útil;
- Um usuário desenhista pode cadastrar, editar e excluir nós;
- Um usuário desenhista pode cadastrar, editar e excluir recursos e suas categorias e hierarquias;
- Um usuário desenhista pode cadastrar, editar e excluir itens de infraestrutura e suas categorias e hierarquias;
- Um usuário desenhista pode cadastrar, editar e excluir fluxos;
- Um usuário desenhista pode cadastrar, editar e excluir parâmetros associados ao sistema, definir os valores mínimo e máximo aceitáveis e o valor padrão.

4.6 Considerações Finais

No decorrer deste capítulo a arquitetura da Plataforma +Precoce foi apresentada, bem como as funcionalidades de cada um de seus módulos. Os principais conceitos que compõem o modelo computacional foram explicados e também os passos que se seguiram para a obtenção do modelo final. Os requisitos funcionais que guiaram a elaboração do projeto estão descritos no Apêndice C. Ao final foi apresentado como foi o início do desenvolvimento, com a inserção das primeiras estórias ao *Product Backlog*. No próximo capítulo será abordada a construção dos módulos da Plataforma que são resultados deste trabalho.

5 Implementação dos Módulos Desenhista e Simulador da Plataforma +Precoce

Neste capítulo são apresentadas as implementações do Módulo Desenhista e Módulo Simulador. O primeiro tem como objetivo realizar a gerência das entidades relativas ao modelo computacional, enquanto o segundo é responsável por executar as simulações com base nesses modelos.

5.1 Ambiente Utilizado para Construção do Módulo Desenhista

O Módulo Desenhista é responsável por gerenciar o modelo computacional permitindo que o usuário desenhista consiga inserir, ler, editar e excluir dados das entidades do modelo computacional. Este módulo foi construído utilizando o Titan *Framework*.

Nesta fase, este projeto deveria se integrar ao que já havia ou estava sendo desenvolvido pela equipe da Embrapa Gado de Corte e, portanto, foi necessário adequar-se ao *know-how* da equipe. Dentre as tecnologias utilizadas pelo grupo de pesquisa destacam-se:

- Utilização do *Git*¹ como repositório e ferramenta de controle de versão do código;
- Utilização do *GitLab* como gerenciador Web para repositórios *Git* e manutenção de lista de tarefas;
- Utilização do *Vagrant*² como gerenciador de máquinas virtuais.

O *Vagrant* permite a configuração de máquinas virtuais (*VirtualBox*, no caso) com a utilização de *scripts* de configuração e ainda oferece aporte ao provisionamento do software que será executado nessa máquina via *scripts* (*shell script*, no caso), permitindo que com poucos comandos o desenvolvedor tenha um ambiente de desenvolvimento completamente configurado com as tecnologias necessárias ao projeto.

5.2 Trabalho Realizado para Construção do Módulo Desenhista

Entendido o contexto da forma de trabalho utilizada pela equipe da Embrapa Gado de Corte, iniciou-se o desenvolvimento do projeto. Foi realizado um ciclo (denominado *sprint* no Scrum) com um mês de duração, onde todas as estórias do *Product Backlog* foram desenvolvidas. Embora nenhuma metodologia ágil tenha sido seguida estritamente,

¹ Acessível pelo endereço: <https://git-scm.com/>.

² Acessível pelo endereço: <https://www.vagrantup.com/>.

práticas de desenvolvimento ágil foram importantes fontes de inspiração para a organização do desenvolvimento de forma interativa e incremental e com equipes distribuídas. Foram seguidas as convenções de código disponibilizadas na documentação, relativa ao item de boas práticas para o uso do Titan *Framework* (Titan Framework, 2016). Ao final do ciclo a aplicação foi disponibilizada via *auto-deploy*³ para que os usuários pudessem testar as funcionalidades em ambiente de homologação.

Após o uso inicial do sistema pelos usuários (pesquisadores da Embrapa), foi observado que somente a implementação dos *CRUD*'s, sem regras, poderiam causar inconsistências nos modelos cadastrados e algumas regras de negócios adicionais foram definidas e então implementadas para prevenir estas inconsistências:

- Quando um desenhista vincular modelo a um sistema, os atributos do modelo (nós, fluxos e itens de infraestrutura) não poderão mais ser alterados, adicionados ou excluídos;
- O desenhista só pode cadastrar fórmulas matematicamente válidas e com uso de parâmetros que já tenham sido cadastrados ou números (ex.: $2 * \text{CÓDIGO_PARÂMETRO}$);
- O usuário desenhista não poderá adicionar ou remover parâmetros relacionados ao sistema;
- Ao vincular modelo ao sistema, os parâmetros utilizados nas fórmulas do modelo deverão ser automaticamente relacionados ao sistema;
- Quando um usuário desenhista cadastrar ou editar o código de um parâmetro, o campo “code” deve permitir o uso do caractere # (cerquilha)⁴ e _ (*underscore*), letras em maiúscula e números. Para outros campos de código, o caractere # (cerquilha) não é permitido.

E algumas novas funcionalidades foram propostas e implementadas:

- O usuário desenhista deverá ser informado, via tela de listagem de modelo, quais sistemas utilizam cada modelo;
- Um usuário desenhista pode cadastrar, editar ou excluir etapas e suas hierarquias;
- O usuário desenhista pode copiar um modelo (e seus relacionamentos e entidades dependentes) para facilitar a rotina de cadastro de modelos;

³ *Script Cron* que roda nas instâncias do Titan e verifica se a *tag* da cópia local da aplicação é a mesma do repositório *Git*. Sendo diferente é feita a atualização do código e migração do banco de dados (caso exista). Em caso de sucesso, passa-se a rodar a versão mais recente da aplicação (definida pela *tag*); em caso de falha é feito *rollback* do banco e da aplicação para a *tag* original.

⁴ A utilização do caractere cerquilha foi o artifício encontrado, como solução simples e rápida, para um problema que surgiu na etapa de codificação dos indicadores. Ele serve para associar o valor de um parâmetro (prefixo antes do cerquilha) à uma categoria de recurso (sufixo após cerquilha).

- O usuário desenhista pode copiar um sistema (e seus relacionamentos e entidades dependentes) para facilitar a rotina de cadastro de sistemas;
- O usuário desenhista deverá ser informado, via tela de listagem de fórmulas, em quais nós e fluxos a fórmula está sendo utilizada.

Para implementação de algumas regras de negócios, outros componentes e tipos, além dos nativos do *Framework*, precisaram ser desenvolvidos. Deve-se destacar o componente *formula* que valida, na criação e edição de dados, se os parâmetros utilizados pela fórmula existem na base e o componente *system* que, ao vincular um modelo ao sistema, identifica todas as fórmulas utilizadas pelo mesmo, extrai os parâmetros e preenche a entidade associativa “*precoce.system_parameter*”. Relacionado aos tipos, destacam-se o *CollectionEditNoNewDelete*, de comportamento semelhante ao tipo nativo *Collection*, porém, não permite que no momento da edição de um sistema o usuário desenhista insira ou exclua parâmetros e também o tipo *ReservedCharacter*, que define o padrão do campo “*name*” em categorias de recursos, categorias de infraestrutura e etapas.

Após a correção da aplicação para a adequação às novas regras e após o desenvolvimento das novas funcionalidades, a aplicação foi disponibilizada no servidor de homologação para verificação e aceite final do usuário desenhista. Algumas telas deste módulo são exibidas nas Figuras 14, 15, 16 e 17.

A Figura 14 apresenta o menu dos Módulos Administrador e Desenhista. À esquerda é visto o menu principal, que contempla todas as telas acessíveis para estes módulos. Ao lado o submenu para o agrupamento “+Precoce”, exibe o link para a tela de “Simulações Realizadas”, que exibe histórico de todas as simulações realizadas na Plataforma. Logo abaixo é exibido o submenu para o agrupamento “Configurações”. Este submenu exibe acessos as telas de gerenciamento de “Sistemas”, e seus atributos caracterizadores: “Biomassas”, “Fases”, “Processos”, “Programas” e “Tecnologias”. O último submenu apresenta o agrupamento “Simulador”, com todas as funcionalidades de gerenciamento dos *CRUD*'s desenvolvidos para o Módulo Desenhista. Os demais submenus compõem telas do Módulo Administrador.

Na tela apresentada na Figura 15 é possível observar uma tela típica de listagem do Titan, a que exibe os modelos cadastrados. Em colunas ao final de cada linha da tabela são exibidas as ações de visualizar, editar, excluir e copiar modelo. As demais telas de listagem são semelhantes quanto à apresentação e comportamento, com algumas regras de negócio específicas para cada caso.

A Figura 16 exibe um exemplo de tela de cadastro de dados. Essa tela apresenta os campos necessários para o cadastro de um fluxo. O canto superior direito apresenta o menu de ações disponíveis. É possível salvar os dados preenchidos (ação que irá processar o *form*, validar dados e persistir em base, no caso de sucesso na validação ou retornar erro,

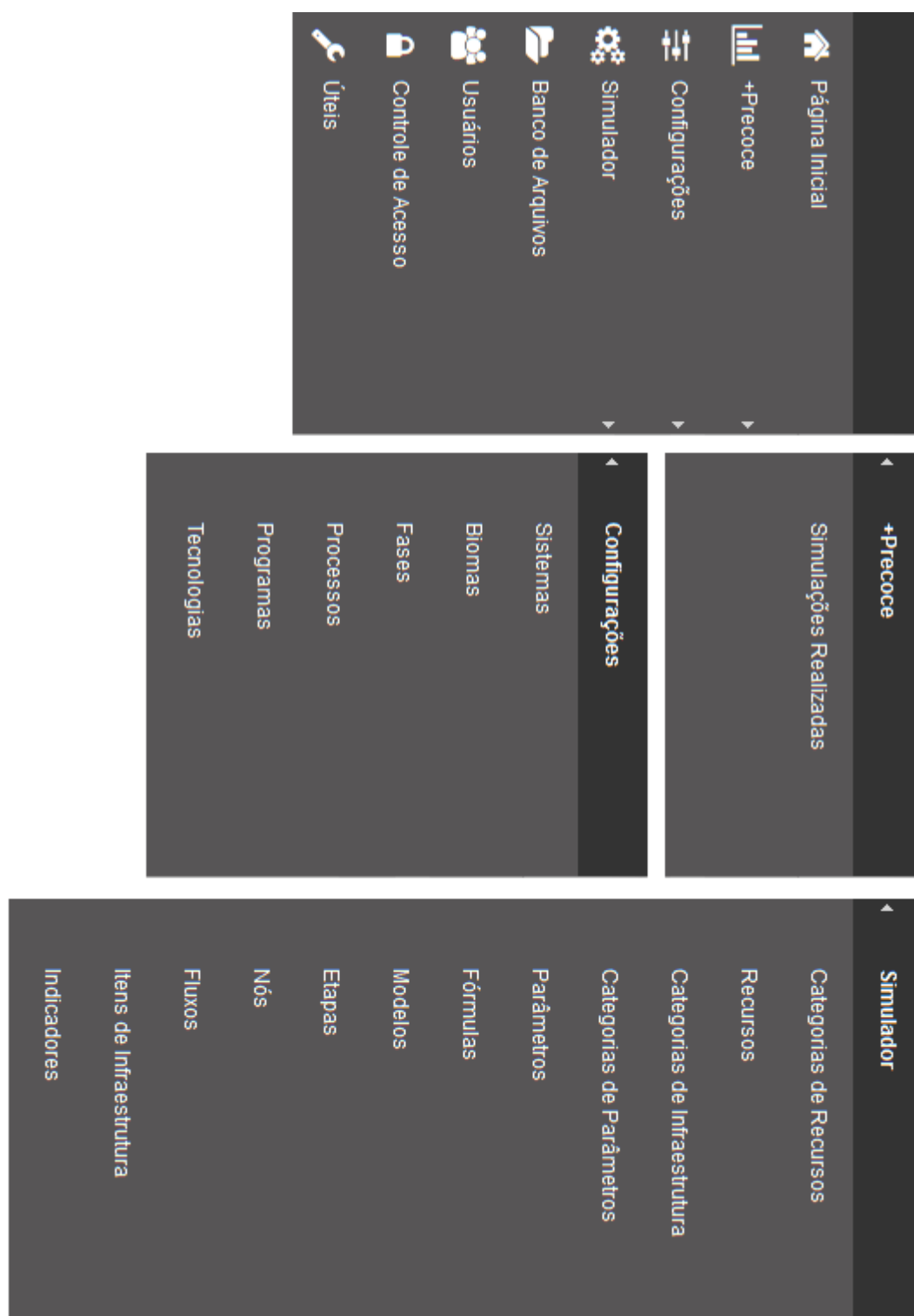


Figura 14 – Menus dos Módulos Administrador e Desenhista.

no caso de falha na validação). A tela para edição de dados é idêntica, com a diferença que os dados já cadastrados virão preenchidos. As demais telas de cadastro e edição são semelhantes quanto à apresentação de dados e comportamento, com algumas regras de negócio específicas para cada caso.

A Figura 17 exibe um exemplo de tela de visualização de dados. Essa tela mostra

os dados cadastrados para um modelo. O canto superior direito apresenta o menu de ações disponíveis. É possível voltar para a tela de listagem, editar os dados visualizados ou imprimir. As demais telas de visualização são semelhantes quanto a apresentação de dados e comportamento, com algumas regras de negócio específicas para cada caso.

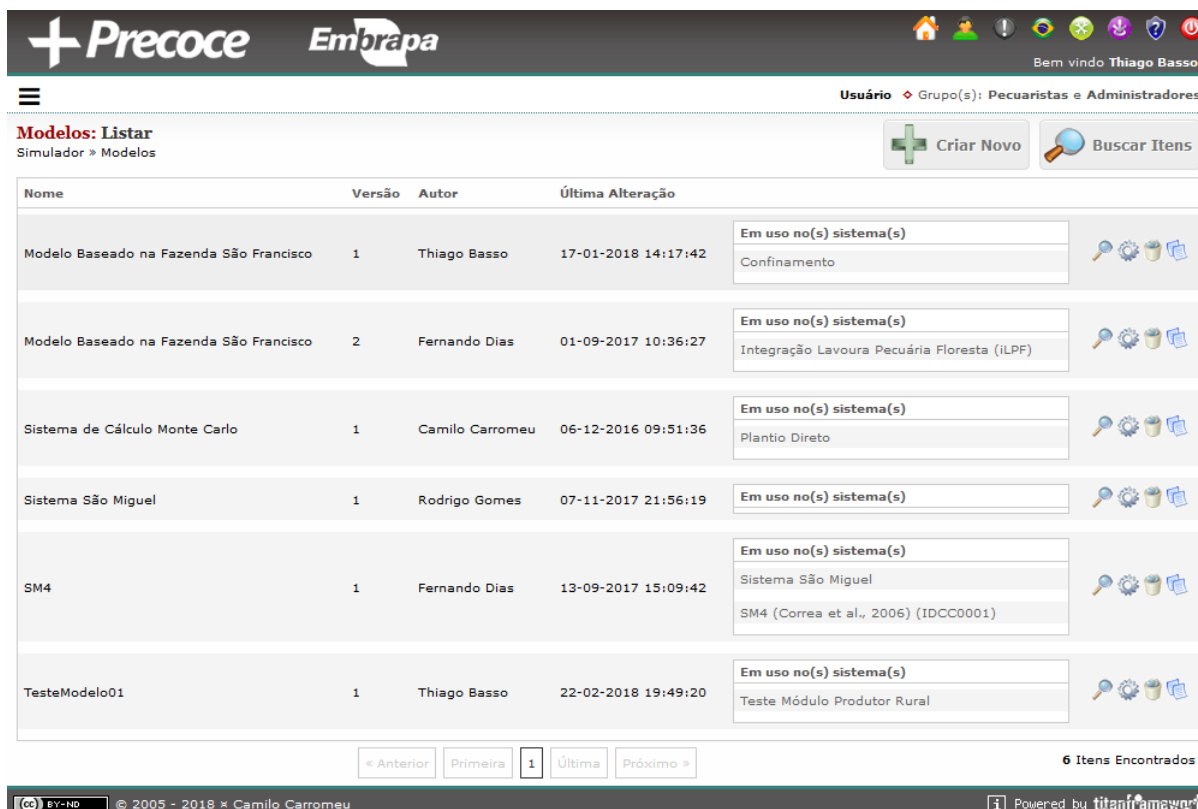


Figura 15 – Módulo Desenhista - Tela de Listagem de Modelos.

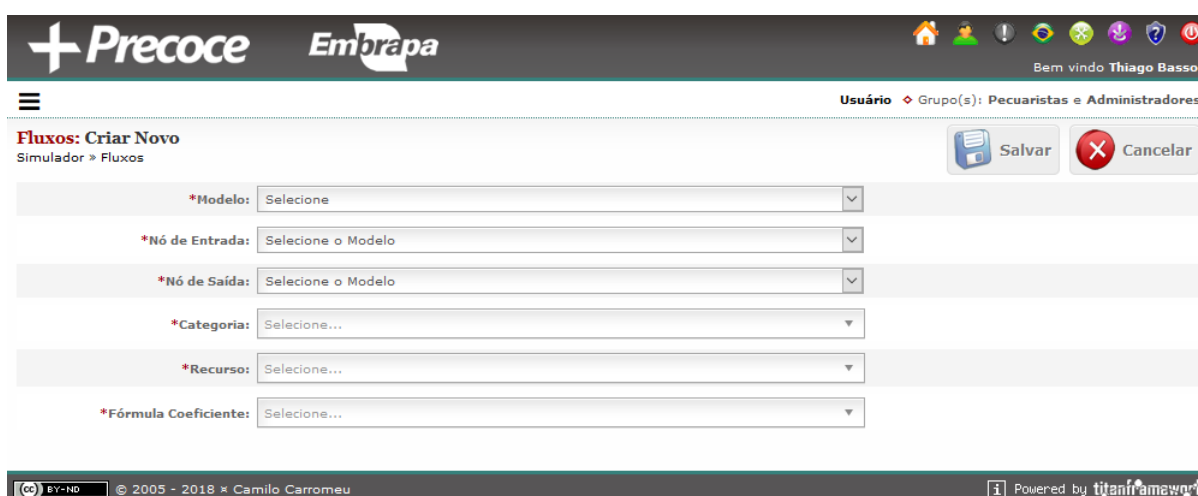


Figura 16 – Módulo Desenhista - Tela de Cadastro ou Edição de Fluxo.

The screenshot shows the 'Modelos: Visualizar' page in the +Precoce Embrapa system. The page header includes the logo, user name 'Thiago Basso', and group 'Pecuaristas e Administradores'. The main content area displays a table of nodes with the following data:

Nome	Fórmula de Duração	Etapa
Fim_Produto	1	
Tratamento1	1	
Tratamento2	1	
Tratamento3	1	
Node1	1	CONFINAMENTO, DESMAMA, ENGORDA, PUERPERIO, PUERPERIO_IATF, RECRIA, RECRIA_ENGORDA
Node2	2	CONFINAMENTO, RECRIA_ENGORDA, ACASALAMENTO, CRIA, EMISSOES
Node3	4	CONFINAMENTO, ACASALAMENTO, EMISSOES
Node4	8	ACASALAMENTO
Node5	16	RECRIA_ENGORDA
Node6	128	CONFINAMENTO, AMAMENTACAO
Node8	32	CONFINAMENTO, DESMAMA, ENGORDA, RECRIA, AMAMENTACAO
Node9	64	RECRIA_ENGORDA, ACASALAMENTO, CRIA, MONTA, PRENHEZ
INI_PRIMIPARA_PARIDA	2	CUSTO_FIXO
INI_PRIMIPARAS	1	CUSTO_FIXO
INI_NULIPARAS	1	CUSTO_FIXO
INI_MULTIPARAS	1	CUSTO_FIXO

Figura 17 – Módulo Desenhista - Tela de Visualização de Modelo.

5.3 Ambiente Utilizado para Construção do Módulo Simulador

O objetivo desta etapa foi a construção do Módulo Simulador. Este módulo é o responsável por realizar o processo de simulação. Seu funcionamento consiste em receber um ID de “*precoce.simulation*”, referente a entrada feita pelo usuário via Módulo Portal, carregar o modelo computacional relacionado, realizar os cálculos do modelo e devolver como saída um *JSON*⁵ contendo uma lista de indicadores calculados e seus resultados, para ser exibido ao usuário no Módulo Portal.

Para facilitar o desenvolvimento e testes do Módulo Simulador, seu desenvolvimento foi realizado de forma independente do *Framework* Titan, utilizando servidor *Apache* local integrado com o *PHP* 7.1.7, provido pela ferramenta *XAMPP*⁶. Após reunião com o analista responsável pelo projeto na Embrapa, decidiu-se manter esse módulo como uma

⁵ Acrônimo para *JavaScript Object Notation* é um padrão aberto utilizado para transferência de dados de objetos entre aplicações.

⁶ Acessível pelo endereço: <https://www.apachefriends.org/>.

biblioteca externa ao *Framework* Titan e depois integrá-lo à aplicação via gerenciador de dependências *Composer*.

O código do simulador foi documentado utilizando padrão de documentação compatível com o *phpDocumentor*⁷. Assim, além da documentação que pode ser lida diretamente no código, caso necessário é possível gerar documentação externa ao código, passível de ser lida em navegador web com recursos em *hiperlink* que fazem a ligação entre a explicação e o trecho de código a que se refere, organizada pelos *namespaces* e classes.

5.4 Trabalho Realizado para Construção do Módulo Simulador

O pesquisador da Embrapa fez um esboço do que seria o algoritmo de simulação, documento que é apresentado no Anexo D. A biblioteca de simulação foi implementada utilizando Orientação a Objetos e estrutura em Grafos. Basicamente o algoritmo de simulação executa os quatro passos mostrados no Algoritmo 1.

Algoritmo 1: Algoritmo para cálculo de simulação

input :ID para precoce.simulation
output :JSON com lista de indicadores calculados

- 1 Algoritmo Simulação
- 2 **begin**
- 3 Carregar dados e modelo computacional do sistema de produção;
- 4 Percorrer grafo calculando durações de estações, coeficientes, datas e quantidades de cada fluxo;
- 5 Calcular estoque de cada recurso em cada estação para cada dia da safra;
- 6 Calcular indicadores aplicáveis ao modelo;
- 7 **end**

O projeto da biblioteca de simulação está organizado da seguinte forma (Figura 18). A pasta raiz é a “*simulator*”. A pasta “*vendor*” (padrão do gerenciador de dependências *Composer*) armazena as bibliotecas externas utilizadas pelo projeto; a biblioteca “*rodriados/mathr v1.0*”⁸ foi utilizada para realizar o *parsing* das fórmulas matemáticas, substituir as variáveis por seus valores armazenados em banco e executar as operações. A pasta “*tests*” contém os arquivos auxiliares, que foram utilizados durante a codificação para testar a implementação, gerar relatórios e controlar tempos de execução. A pasta “*src*” contém os arquivos da biblioteca propriamente ditos. Cada uma de suas subpastas contemplam um *Namespace*, que representam os pacotes de códigos organizados pela sua função. O *namespace* “*Simulator\ComputationalModel*” armazena todas as classes envolvidas na montagem do modelo computacional, carga de dados e execução da simulação. “*Simulator\Graph*” armazena as classes que implementam a estrutura de grafos utilizada

⁷ Acessível por: <https://www.phpdoc.org/>.

⁸ Acessível pelo endereço <https://github.com/rodriados/mathr>

para percorrer o modelo computacional. “*Simulator\IndicatorCalculator*” aglutina todas classes que implementam rotinas de cálculo de indicadores, que são a saída de uma simulação. “*Simulator\Utils*” contém implementações auxiliares da biblioteca, em sua maioria, funções estáticas que são utilizadas em várias partes do código.

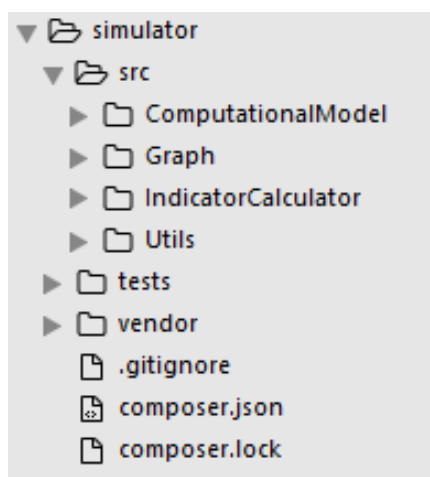


Figura 18 – Organização do projeto da biblioteca de simulação.

O diagrama de classes da estrutura lógica da biblioteca de simulação é apresentada na Figura 19. A classe *ComputationalModel* é responsável por gerenciar o processo de simulação. É ela que inicia a execução do Algoritmo 1, recebendo como entrada uma conexão com a base de dados e o ID que identifica a simulação a ser executada. A classe *ResultQueryFactory* devolve à *ComputationalModel* os dados solicitados da base. Os dados do modelo computacional e os dados relativos aos valores dos parâmetros para a simulação (preenchidos pelo usuário final, e armazenado em banco) são carregados. O grafo do modelo computacional é montado e percorrido utilizando o algoritmo de busca em largura, com algumas modificações, conforme exibido no Algoritmo 2.

Durante o percurso de busca em largura, ao visitar cada nó é definida sua categoria, a saber: terminal, produção (referida no Anexo B como transformação⁹), tratamento, soma de produção, soma de tratamento e reúso. A informação de quem é o nó raiz e primeiro nó é obtida via consulta em banco. O nó raiz é quem inicia o percurso pelo grafo (Algoritmo 2). A categoria do nó define como as quantidades e datas dos fluxos relacionados devem ser calculadas, cada qual com suas especificidades. Esse conceito assemelha-se ao usado no padrão de projeto *Strategy* (ERICH et al., 2000), que foi utilizado para esse propósito. A interface *CalculateStrategy* anota o método de cálculo que deve ser tratado em cada uma das implementações das classes de cálculo:

⁹ Os fluxos de transformação também sofreram ajustes na nomenclatura e passaram a ser nomeados como fluxos de produção.

Algoritmo 2: Algoritmo que calcula nós e fluxos do modelo computacional utilizando a busca em largura (adaptado de GOLDBARG; GOLDBARG 2012).

```

1 Algoritmo BuscaEmLarguraModificado
2 begin
3   Define uma fila Q vazia;
4   Insere vértice raiz em Q;
5   while Q não for vazio do
6     v := Retira primeiro elemento da fila Q;
7     if v não tiver sido visitado then
8       Define a categoria do Nó associado ao vértice;
9       Calcula o Nó de acordo com a sua estratégia (CalculateStrategy, fig. 19);
10      /* O Nó associado ao vértice v pode não ser calculado caso seus
11      fluxos de referência não tenham sido calculados. */
12      if v foi calculado then
13        Marca v como visitado;
14        foreach vizinho em Vértices Vizinhos de v do
15          if vizinho não foi visitado then
16            | Insere vizinho no fim da fila Q;
17          end
18        end
19      end
20 end

```

- *CalculateNodeOneStrategy*: representado na Figura 20, o primeiro nó do grafo será uma estação de produção que está ligado ao nó raiz, isto é o único nó do modelo que apresenta apenas um fluxo de entrada do tipo Produção. O primeiro nó do grafo terá comportamento de cálculo específico, diferente daquele especificado para a sua categoria (estação de produção). No Algoritmo 3 é apresentada a estratégia de cálculo para esta categoria de nó.

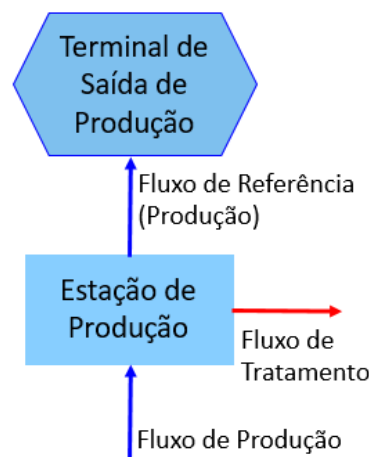


Figura 20 – Representação do primeiro nó do grafo.

Algoritmo 3: Rotina de cálculo para o primeiro nó (após o raiz) do grafo.

```

1 Algoritmo CalculateNodeOneStrategy
2 begin
3   duracao := Resultado da fórmula de duração relacionada ao nó;
4   diaFim := Último dia da safra (isto é, 365);
5   diaInicio := diaFim - duracao;
6   fluxoReferencia := Fluxo de saída do tipo produção;
7   fluxoReferencia.quantidade := 1.0 * Resultado da fórmula de coeficiente
   relacionada ao fluxo;
8   fluxoReferencia.dia := diaFim;
9   fluxoReferencia.calculado := verdadeiro;
10  foreach fluxo em Fluxos relacionados ao Nó do
11    if fluxo != fluxoReferencia then
12      if fluxo de entrada then
13        | fluxo.dia := diaInicio;
14      end
15      if fluxo de saída then
16        | fluxo.dia := diaFim;
17      end
18    end
19    fluxo.quantidade := fluxoReferencia.quantidade * Resultado da fórmula de
   coeficiente relacionada ao fluxo;
20    fluxo.calculado := verdadeiro;
21  end
22  calculado := verdadeiro;
23 end

```

- *CalculateProductionStrategy*: representados na Figura 21, são os nós que apresentam um só fluxo de produção de saída, nenhum fluxo de tratamento de entrada, um ou mais fluxos de produção de entrada e zero ou mais fluxos de tratamento de saída. O fluxo de referência deve ter sido calculado para que ocorra a execução da rotina de cálculo do nó. No Algoritmo 4 é apresentada a estratégia de cálculo para esta categoria de nó.

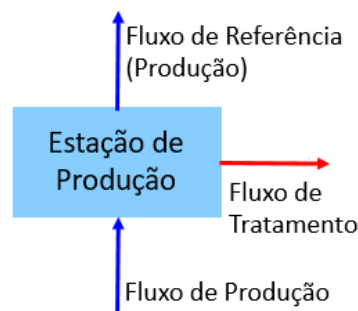


Figura 21 – Representação das estações de produção.

Algoritmo 4: Rotina de cálculo para estações de produção.

```

1 Algoritmo CalculateProductionStrategy
2 begin
3   fluxoReferencia := Fluxo de saída do tipo produção;
4   if fluxoReferencia.calculado == verdadeiro then
5     duracao := Resultado da fórmula de duração relacionada ao nó;
6     diaFim := fluxoReferencia.dia;
7     diaInicio := diaFim - duracao;
8     foreach fluxo em Fluxos relacionados ao Nó do
9       if fluxo != fluxoReferencia then
10        if fluxo de entrada then
11          | fluxo.dia := diaInicio;
12        end
13        if fluxo de saída then
14          | fluxo.dia := diaFim;
15        end
16      end
17      fluxo.quantidade := fluxoReferencia.quantidade * Resultado da fórmula
18      de coeficiente relacionada ao fluxo;
19      fluxo.calculado := verdadeiro;
20    end
21    calculado := verdadeiro;
22 end

```

- *CalculateTreatmentStrategy*: representados na Figura 22, são os nós que não apresentam fluxo de produção de saída, um só fluxo de tratamento de entrada, zero ou mais fluxos de produção de entrada e um ou mais fluxos de tratamento de saída. O fluxo de referência deve ter sido calculado para que ocorra a execução da rotina de cálculo do nó. No Algoritmo 5 é apresentada a estratégia de cálculo para esta categoria de nó.

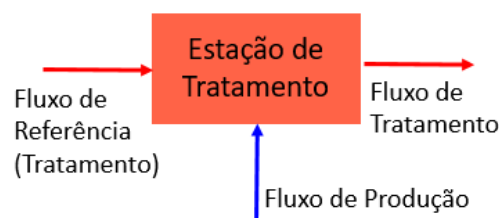


Figura 22 – Representação das estações de tratamento.

Algoritmo 5: Rotina de cálculo para estações de tratamento.

```

1 Algoritmo CalculateTreatmentStrategy
2 begin
3   fluxoReferencia := Fluxo de entrada do tipo tratamento;
4   if fluxoReferencia.calculado == verdadeiro then
5     duracao := Resultado da fórmula de duração relacionada ao nó;
6     diaInicio := fluxoReferencia.dia;
7     diaFim := diaInicio + duracao;
8     foreach fluxo em Fluxos relacionados ao Nó do
9       if fluxo != fluxoReferencia then
10        if fluxo de entrada then
11          | fluxo.dia := diaInicio;
12        end
13        if fluxo de saída then
14          | fluxo.dia := diaFim;
15        end
16      end
17      fluxo.quantidade := fluxoReferencia.quantidade * Resultado da fórmula
18      de coeficiente relacionada ao fluxo;
19      fluxo.calculado := verdadeiro;
20    end
21    calculado := verdadeiro;
22 end

```

- *CalculateReuseStrategy*: representados na Figura 23 são os nós que apresentam sempre quatro fluxos: um de saída de produção, um de entrada de tratamento, um de entrada de produção e um de saída de tratamento, todos do mesmo recurso. Os fluxos de referência devem ter sido calculados para que ocorra a execução da rotina de cálculo do nó. No Algoritmo 6 é apresentada a estratégia de cálculo para esta categoria de nó.

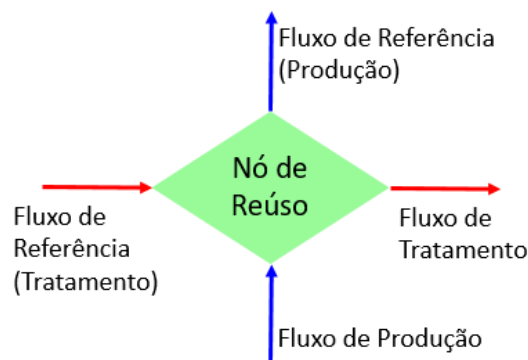


Figura 23 – Representação dos nós de reuso.

Algoritmo 6: Rotina de cálculo para nós de reuso.

```

1 Algoritmo CalculateReuseStrategy
2 begin
3   fluxoReferenciaProducao := Fluxo de saída do tipo produção;
4   fluxoReferenciaTratamento := Fluxo de entrada do tipo tratamento;
5   if fluxoReferenciaProducao e fluxoReferenciaTratamento calculados then
6     duracao := diaInicio := diaFim := nulo;
7     if dia de safra dos fluxos de referência são iguais then
8       balanço = fluxoReferenciaTratamento.quantidade -
9         fluxoReferenciaProducao.quantidade;
10      if balanço negativo then
11        fluxoProducaoEntrada.quantidade := -balanço;
12        fluxoTratamentoSaida.quantidade := 0;
13      end
14      if balanço positivo then
15        fluxoProducaoEntrada.quantidade := 0;
16        fluxoTratamentoSaida.quantidade := balanço;
17      end
18      if balanço == zero then
19        fluxoProducaoEntrada.quantidade := 0;
20        fluxoTratamentoSaida.quantidade := 0;
21      end
22    end
23  else
24    fluxoProducaoEntrada.quantidade :=
25      fluxoReferenciaProducao.quantidade;
26    fluxoTratamentoSaida.quantidade :=
27      fluxoReferenciaTratamento.quantidade;
28  end
29  fluxoProducaoEntrada.dia := fluxoReferenciaProducao.dia;
30  fluxoTratamentoSaida.dia := fluxoReferenciaTratamento.dia;
31  fluxoProducaoEntrada.calculado := verdadeiro;
32  fluxoTratamentoSaida.calculado := verdadeiro;
33  calculado := verdadeiro;
34 end

```

- *CalculateSumProductionStrategy*: representados na Figura 24, são os nós que apresentam dois ou mais fluxos de saída de produção e apenas um fluxo de entrada de produção, todos do mesmo recurso. Os fluxos de referência devem ter sido calculados para que ocorra a execução da rotina de cálculo do nó. No Algoritmo 7 é apresentada a estratégia de cálculo para esta classe de nó.

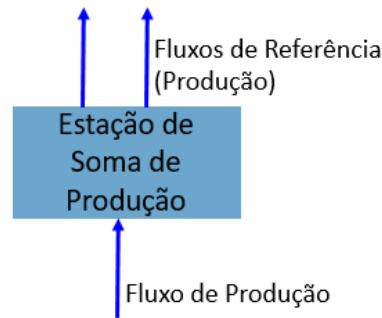


Figura 24 – Representação das estações de soma de produção.

Algoritmo 7: Rotina de cálculo para estações de soma de produção.

```

1 Algoritmo CalculateSumProductionStrategy
2 begin
3   fluxosReferencia := Fluxos de saída do tipo produção;
4   if todos fluxosReferencia calculados then
5     diaInicio := dia dos fluxosReferencia com menor dia;
6     diaFim := dia dos fluxosReferencia com maior dia;
7     duracao := diaFim - diaInicio;
8     fluxoProducaoEntrada.dia := diaInicio;
9     fluxoProducaoEntrada.quantidade := Soma das quantidades dos
      fluxosReferencia;
10    fluxoProducaoEntrada.calculado := verdadeiro;
11    calculado := verdadeiro;
12  end
13 end

```

- *CalculateSumTreatmentStrategy*: representados na Figura 25, são os nós que apresentam dois ou mais fluxos de entrada de tratamento e apenas um fluxo de saída de tratamento, todos do mesmo recurso. Os fluxos de referência devem ter sido calculados para que ocorra a execução da rotina de cálculo do nó. No Algoritmo 8 é apresentada a estratégia de cálculo para esta categoria de nó.

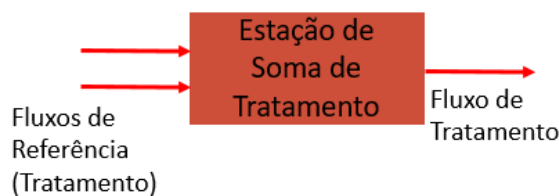


Figura 25 – Representação das estações de soma de tratamento.

Algoritmo 8: Rotina de cálculo para estações de soma de tratamento.

```

1 Algoritmo CalculateSumTreatmentStrategy
2 begin
3   fluxosReferencia := Fluxos de entrada do tipo tratamento;
4   if todos fluxosReferencia calculados then
5     diaInicio := dia dos fluxosReferencia com menor dia;
6     diaFim := dia dos fluxosReferencia com maior dia;
7     duracao := diaFim - diaInicio;
8     fluxoTratamentoSaida.dia := diaFim;
9     fluxoTratamentoSaida.quantidade := Soma das quantidades dos
      fluxosReferencia;
10    fluxoTratamentoSaida.calculado := verdadeiro;
11    calculado := verdadeiro;
12  end
13 end

```

- *CalculateTerminalStrategy*: representados na Figura 26, são os nós que apresentam apenas um fluxo de entrada ou de saída de produção ou de tratamento. Indicam a saída ou entrada de algum produto de ou para o sistema. No Algoritmo 9 é apresentada a estratégia de cálculo para este tipo de nó.

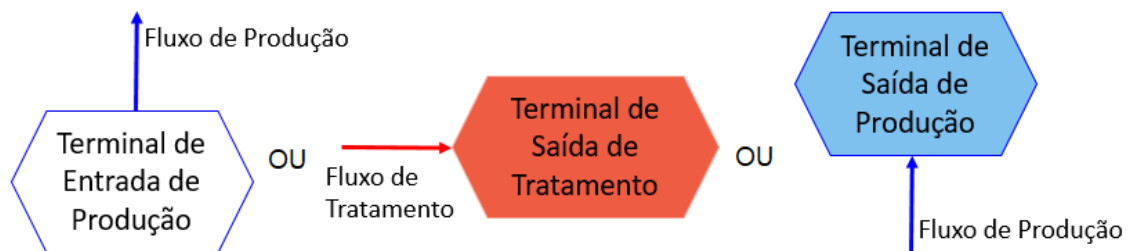


Figura 26 – Representação dos nós terminais.

Algoritmo 9: Rotina de cálculo para nós terminais.

```

1 Algoritmo CalculateTerminalStrategy
2 begin
3   diaInicio := diaFim := duracao := nulo;
4   calculado := verdadeiro;
5 end

```

O objeto de instância da classe *ComputationalModel* segue para o passo seguinte do Algoritmo 1, realizando o cálculo do estoque. Esse método foi adaptado da função *calculateStock* apresentada no Anexo D. Nesta etapa ocorre a consolidação das quantidades de recursos nas estações em cada dia da safra. A estrutura que controla essa informação é

um *Array* de três dimensões indexado por estação, recurso e dia de safra (isto é, o módulo do dia de uso da estação por 365, supondo a safra definida como período de um ano). Foi definida uma taxa de permanência de 50% para as estações de produção e de tratamento, que se refere a permanência dos recursos na estação, ou seja, do primeiro dia até a metade da duração da estação, ela é ocupada pelos recursos relativos aos fluxos que entram na estação e a partir da metade + 1 da duração até o penúltimo dia da estação ela é ocupada pelos recursos que saem. Foi convencionado que os recursos permanecem na estação até o penúltimo dia pois, tanto o primeiro, quanto o último dia condizem com a execução dos fluxos e estes sempre serão instantâneos, ou seja, sempre estarão na estação de destino no dia que ocorrem.

Para as estações de soma de tratamento o estoque é calculado acumulando as quantidades dos fluxos de entrada nos dias da safra de ocorrência de cada fluxo com todo o estoque saindo no dia de ocorrência do fluxo de maior dia de safra. Nas estações soma de produção o estoque é calculado abatendo as quantidades dos fluxos de saída nos dias de safra de cada fluxo, com todo o estoque entrando no dia de ocorrência do fluxo de menor dia de safra. Tanto as informações dos fluxos calculados no percurso do grafo, quanto o estoque são mantidas na instância do objeto da classe *SimulationData*.

No último passo do Algoritmo 1 é executada rotina para cálculo dos indicadores¹⁰. A execução segue com o objeto instanciado da classe *ComputationalModel* organizando a execução do cálculo dos indicadores. Cada indicador definido para uso na Plataforma tem especificidades para a realização de seu cálculo, razão pela qual o padrão de projeto *Strategy* foi novamente utilizado. As classes para cálculo dos indicadores são aquelas em amarelo no Diagrama de Classes da Figura 19. Em comum, todas essas classes implementam o método de cálculo utilizando informações previamente calculadas para fluxos e estoque, de modo que a anotação do método *calculate* na interface *IndicatorStrategy* recebe como parâmetro o objeto *SimulationData*.

Outro aspecto comum às classes que implementam indicadores é que todas derivam de um *Framework* de Cálculo (especialização da classe *CalculationFramework*), definido pelo analista de negócios da Embrapa como sendo: $\left(\frac{(\text{Numerador1} - \text{Numerador2}) * \text{Numerador3}}{(\text{Denominador1} - \text{Denominador2}) * \text{Denominador3}} \right)$. Essa classe devolve ao método chamador, que armazena no objeto *SimulationData*, uma lista de indicadores calculados. Uma especificidade é que cada indicador pode ter várias formas de ser representado, variando em conjunto de estações envolvidas (etapas), unidade de medida que o valor é exibido ou categoria de recurso utilizado, sendo cada uma dessas variantes também um indicador. Por exemplo: o indicador Duração Média tem seu valor calculado para estações que pertencem as etapas de recria, engorda e confinamento (3 variantes); o indicador Rebanho é calculado para as unidades de medida Kg de peso vivo e número de cabeças (2 variantes); o indicador Total de Matrizes é calculado para os tipos

¹⁰ A definição de como calcular cada um dos indicadores foi repassada pelo analista de negócios.

de recurso Nuliparas, Primiparas, Multiparas ou Matrizes (4 variantes); dentre outros. Esse é o motivo de retornar uma lista, pois cada implementação é responsável por calcular todas as variantes, isto é, todos os indicadores calculados por aquela classe.

A rotina de cálculo dos indicadores foi construída utilizando os recursos de reflexão do *PHP*. O uso de reflexão permite que operações sejam carregadas em tempo de execução e testadas dinamicamente quanto a sua estrutura (assinatura de interfaces, assinatura de métodos e especialização de determinadas classes). Inicialmente a ideia era que cada classe de implementação de indicador pudesse ser armazenada em um repositório e, com o uso de reflexão, ser carregada dinamicamente e testada quanto a algumas regras básicas, a saber: se é uma subclasse de *CalculationFramework* e se implementa a interface *IndicatorStrategy* e seu método *calculate*. Desta forma, poderia existir uma tela dentro dos Módulos Administrador ou Desenhista em que o usuário responsável adicionaria novos indicadores para serem calculados e fazer *upload* do arquivo contendo a implementação da classe em *PHP*. No entanto, essa ideia foi descontinuada, dando-se prioridade à codificação dos indicadores que já haviam sido definidos. Ainda assim, a solução foi implementada com o uso de reflexão, com a diferença que a carga de quais indicadores estão disponíveis na Plataforma vem de uma lista fixa em código e não da base de dados. No total foram implementadas 42 classes que geram como saída um total de 122 variantes dos indicadores representados como classes.

5.5 Avaliando o Módulo Simulador

Cada uma das etapas do Algoritmo 1 foram avaliadas para garantir sua correta execução. Para este propósito o analista de negócios da Embrapa elaborou 5 modelos que não representam nenhum sistema real, mas que permitiram mais facilmente conferir os resultados da simulação em várias situações que podem aparecer em modelos de sistemas reais. Na Figura 27 é exibido um dos casos elaborados, a caixa cinza representa a fronteira do sistema de produção. Não foram representados os nós terminais, assim, os fluxos que cruzam a fronteira do sistema devem ser cadastrados contendo um nó terminal.

As avaliações executadas foram realizadas em ambiente de desenvolvimento. Os *scripts* com chamadas às rotinas necessárias foram implementadas na pasta *tests* (Figura 18).

De posse dos casos preparados para a avaliação, os mesmos foram cadastrados na base utilizando as funcionalidades do Módulo Desenhista. Primeiramente foi avaliado se os grafos repassados estavam sendo corretamente carregados da base. Com este propósito a rotina *generateDotLanguage()* foi desenvolvida para que fosse possível comparar de forma visual o grafo cadastrado com o que fora repassado. O objetivo desta rotina é exibir, em

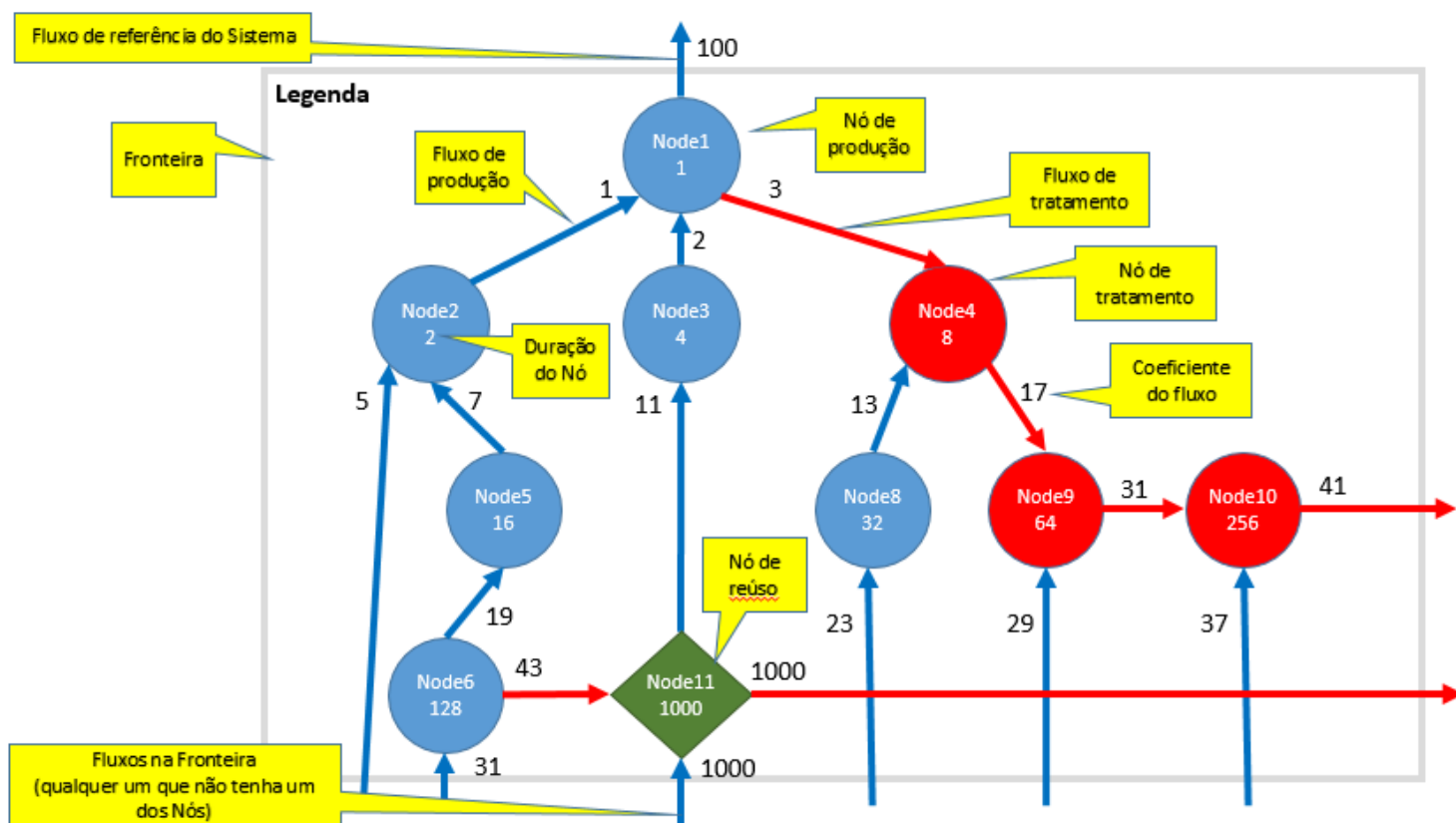


Figura 27 – Exemplo de caso de teste elaborado pelo analista de negócios com a descrição de cada um dos componentes.

linguagem *DOT*¹¹ (GANSNER; KOUTSOFIOS; NORTH, 2015), o grafo cadastrado. Com o uso da *DOT language* é possível utilizar ferramentas *web* gratuitas para renderizar a saída em imagem, neste caso a ferramenta escolhida foi a *Viz.js*¹². Cada um dos grafos cadastrados são exibidos nas Figuras 29, 30, 31, 32 e 33. Nas imagens, junto aos nós são exibidos o nome do nó e a fórmula de duração, junto aos fluxos é exposto o recurso carregado, sua categoria e a fórmula de coeficiente. A Figura 28 apresenta a legenda para entendimento das imagens, sendo válido destacar que os recursos e seus tipos foram atribuídos aleatoriamente. Os grafos foram comparados e obteve-se sucesso em todos os casos.

A segunda parte da avaliação verificou se o grafo era corretamente calculado. Cada um dos casos de teste foi previsto visando a testar os limites do percurso no grafo, pois estes influenciam a ordem do cálculo. Para verificar a correção dos resultados desta etapa, as variáveis dos fluxos (coeficiente, quantidade e data) foram inseridas nas saídas da *DOT language* (vide marcação X, Y, Z exibida na última linha do *label* ao lado do fluxo). Os resultados apresentados nas Figuras 29, 30, 31, 32 e 33 foram comparados com modelos resolvidos manualmente.

¹¹ Linguagem para descrição de grafos.

¹² Acessível pelo endereço: <http://viz-js.com/>.

A Figura 29 exibe o caso mais simples em que só existem estações de produção. Ele foi elaborado para verificar se a rotina de cálculo *CalculateProductionStrategy* estava correta. Para o segundo caso (Figura 30) foram inseridas estações de tratamento. Ele foi elaborado para verificar se a rotina de cálculo *CalculateTreatmentStrategy* estava correta. No terceiro caso (Figura 31) foram inseridos nós de reúso. Ele foi elaborado para verificar se a rotina de cálculo *CalculateReuseStrategy* estava correta, na condição em que o reúso não ocorre. O quarto caso (Figura 32) avalia a condição em que o reúso ocorre e está localizado fora da fronteira. No quinto caso (Figura 33) foi avaliado a corretude do cálculo quando ocorrem dois reúsos e se o percurso do grafo é feito de forma correta. Nesta situação o algoritmo de busca em largura (ver Algoritmo 2) ao visitar o “Node2” irá inserir o “Node11” na fila de percurso. Porém, o “Node11” só poderá executar sua rotina de cálculo após o “Node13” ter sido calculado, pois ainda dependerá de um dos fluxos de referência. Obteve-se sucesso em todos os casos.



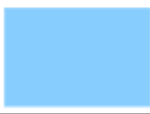


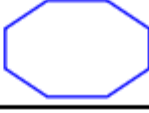




Legenda			
	Nó Raiz Terminal de Saída de Produção		Nó Terminal de Saída de Tratamento
	Estação de Produção		Estação de Tratamento
	Nó de Reúso		Nó Terminal de Entrada de Produção
	Estação de Soma de Produção		Estação de Soma de Tratamento
	Fluxo de Produção		Fluxo de Tratamento
(X, Y, Z)	X: Coeficiente do Fluxo Y: Quantidade do Fluxo Z: Dia do Fluxo		

Figura 28 – Legenda utilizada para os grafos das próximas figuras.

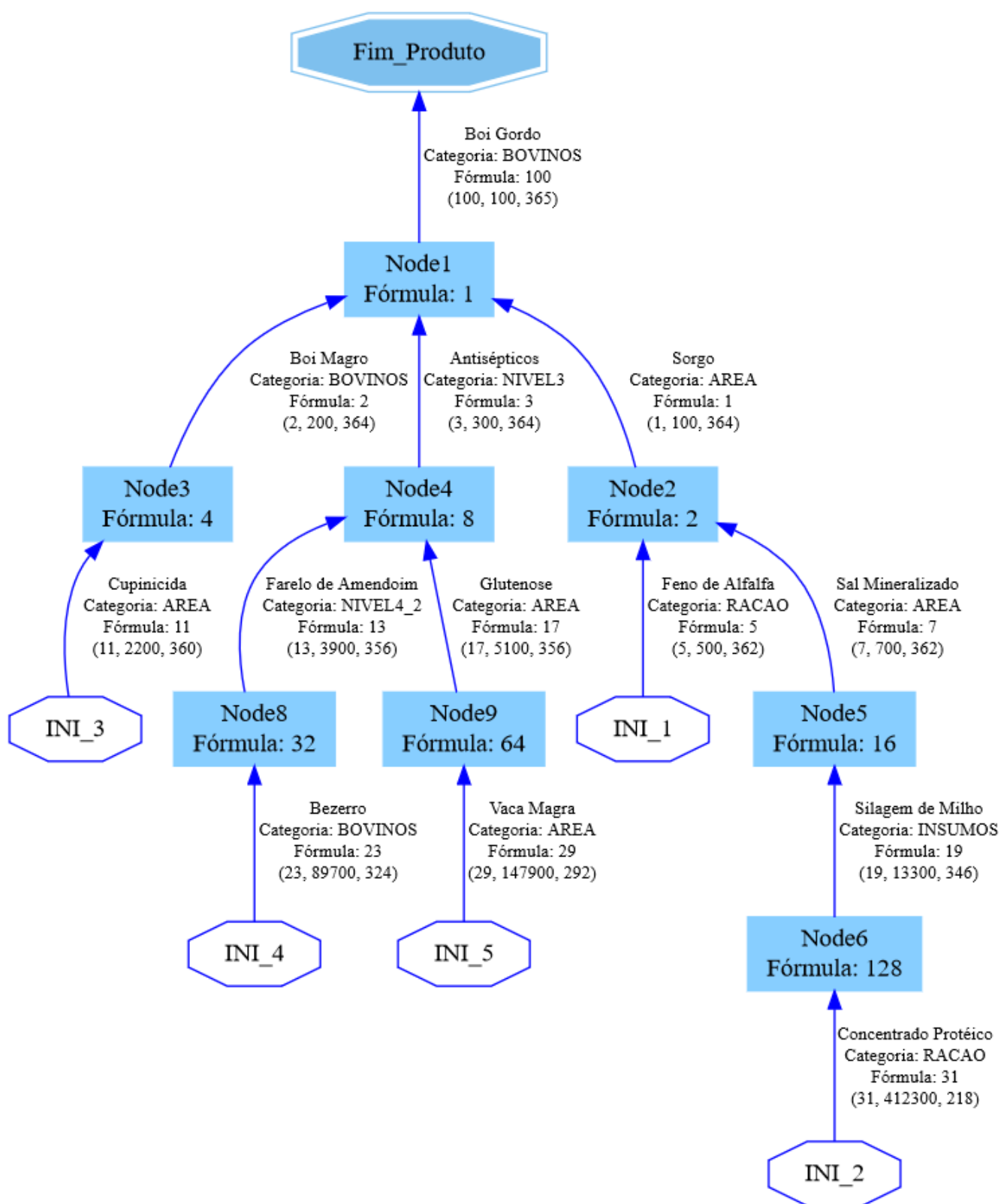


Figura 29 – Caso de Teste 1 - Só estações de Produção.

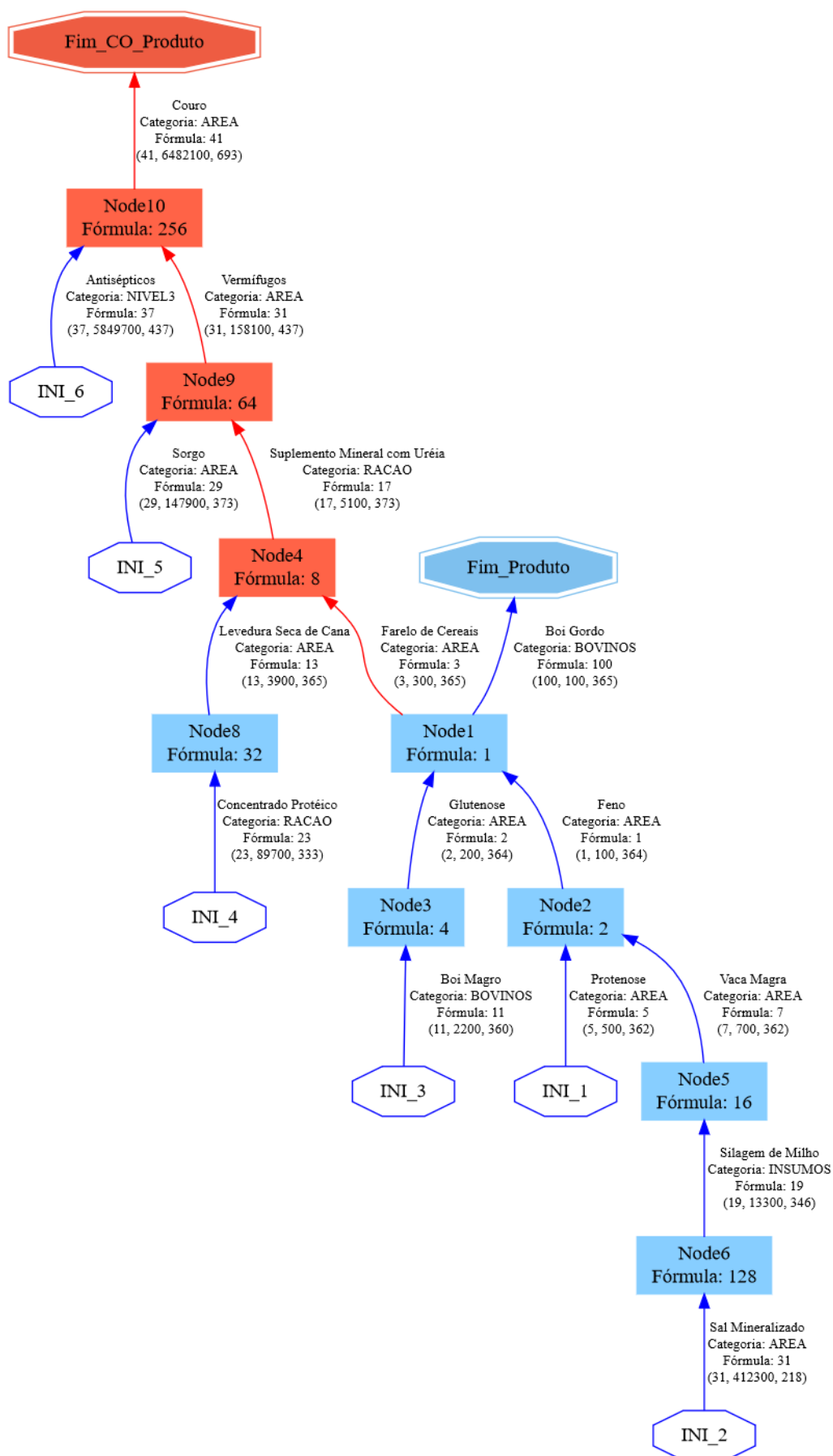


Figura 30 – Caso de Teste 2 - Estações de Produção e Tratamento.

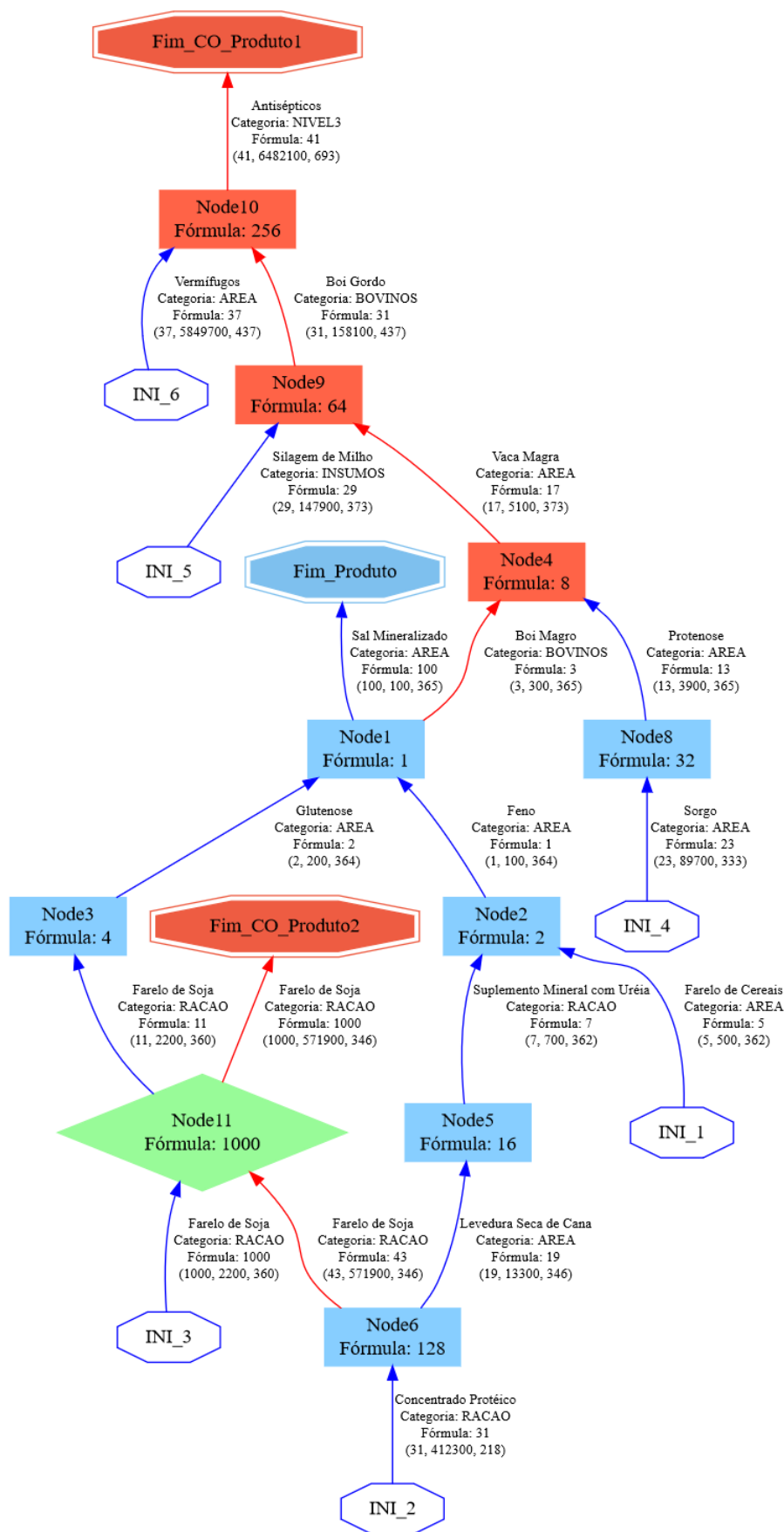


Figura 31 – Caso de Teste 3 - Estações de Produção, Tratamento e Reúso na fronteira (sem sucesso).

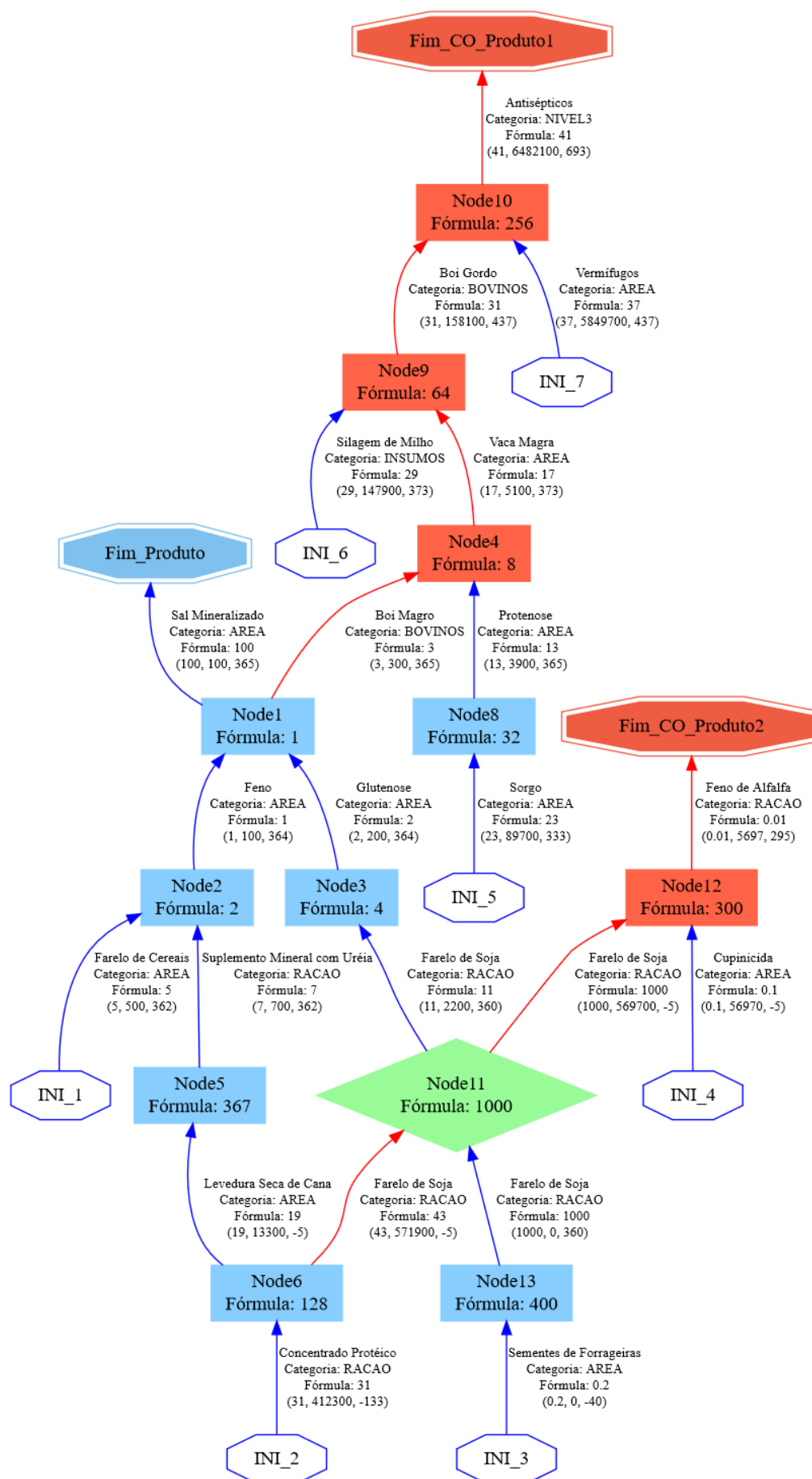


Figura 32 – Caso de Teste 4 - Estações de Produção, Tratamento e Reúso fora da fronteira (com sucesso).

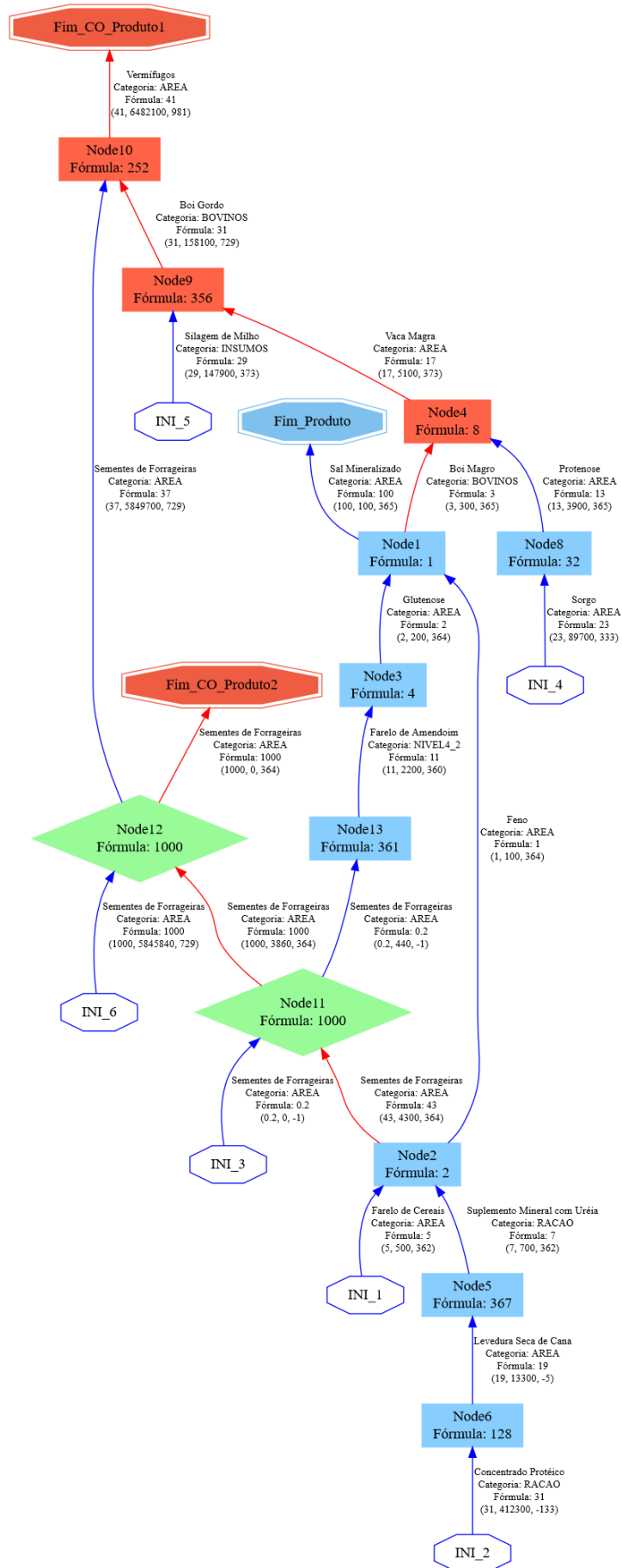


Figura 33 – Caso de Teste 5 - Dois Reúso e chegando ao Node11 sem todos fluxos de referência calculados.

A próxima etapa de avaliação fez a verificação da rotina para cálculo do estoque (linha 5 do Algoritmo 1). Para esta análise foi desenvolvido relatório contendo a saída do estoque dia a dia, com cada informação separada por *pipes* |, conforme trecho exibido na Figura 34.

```
ESTOQUE CALCULADO:
recurso: Antisépticos | nó: Node10 | dia: 200 | qtdy: 6482100
recurso: Antisépticos | nó: Node10 | dia: 201 | qtdy: 6482100
recurso: Antisépticos | nó: Node10 | dia: 202 | qtdy: 6482100
recurso: Antisépticos | nó: Node10 | dia: 203 | qtdy: 6482100
recurso: Antisépticos | nó: Node10 | dia: 204 | qtdy: 6482100
.           .           .           .
.           .           .           .
.           .           .           .
```

Figura 34 – Trecho do relatório de estoque.

De posse dessas informações o analista de negócios desenvolveu uma planilha eletrônica para comparar as saídas com o que era esperado. O resultado para a análise do caso de teste 4 (Figura 32) é exibido na Figura 35. Nas linhas correspondentes ao Node13, em que a coluna “Status” exibe a informação “?”, há um caso em que a quantidade calculada do recurso em estoque é zero. Para casos como esse ficou definido que não é necessário registrar estoque zero.

Por fim, para a última etapa de avaliação criou-se um modelo completo para verificar os indicadores. Esse modelo atinge todas as restrições para calcular cada um dos indicadores (na prática, os modelos não necessariamente irão calcular todos os indicadores implementados, pois nem todos indicadores se aplicam a todos os modelos). Categorias de recursos necessárias ao teste foram cadastradas (Tabela 9). Etapas foram vinculadas aos nós necessários. Os recursos utilizados foram vinculados às categorias de recursos e vinculados aos fluxos de acordo com as necessidades de cálculo. Todos os parâmetros necessários foram adicionados e seus valores para teste foram atribuídos (Tabela 10). O grafo do modelo de teste é apresentado na Figura 36.

Para verificar as respostas dos indicadores calculados, o analista de negócios desenvolveu nova planilha. Os valores calculados para cada fluxo e estoque foram inseridos na planilha, e também os valores de parâmetros (Tabela 10) e categorias de recursos (Tabela 9). De posse desses dados o referido analista calculou cada uma das 122 variações de indicadores utilizando planilha eletrônica. E, por fim, os valores calculados pela biblioteca, com aqueles calculados na planilha foram confrontados. Algumas incoerências surgiram e foram corrigidas, ora correções no código, ora correções no cálculo da planilha. Houve também alterações na especificação do cálculo de alguns indicadores. Por fim todos

Nó	Recurso	Valores			Orientação	Status
		Qtd	Mín de Dia	Máx de Dia		
Node1	Glutenose	200	364	364	1 Entra	OK
	Feno	100	364	364	1 Entra	OK
Node10	Antissépticos	6.482.100	200	327	128 Sai	OK
	Vermífugos	5.849.700	72	199	128 Entra	OK
	Boi Gordo	158.100	72	199	128 Entra	OK
Node12	Cupinicida	56.970	0	364	150 Entra	OK
	Feno de Alfalfa	5.697	145	294	150 Sai	OK
	Farelo de Soja	569.700	0	364	150 Entra	OK
Node13	Farelo de Soja	0	160	359	200 Sai	?
	Sementes de Forrageiras	0	0	364	200 Entra	?
Node2	Suplemento Mineral com Uréia	700	362	362	1 Entra	OK
	Farelo de Cereais	500	362	362	1 Entra	OK
	Feno	100	363	363	1 Sai	OK
Node3	Glutenose	200	362	363	2 Sai	OK
	Farelo de Soja	2.200	360	361	2 Entra	OK
Node4	Boi Magro	300	0	3	4 Entra	OK
	Vaca Magra	5.100	4	7	4 Sai	OK
	Protenose	3.900	0	3	4 Entra	OK
Node5	Suplemento Mineral com Uréia	700	179	361	183 Sai	OK
	Levedura Seca de Cana	13.300	0	364	184 Entra	OK
Node6	Concentrado Protéico	412.300	232	295	64 Entra	OK
	Levedura Seca de Cana	13.300	296	359	64 Sai	OK
	Farelo de Soja	571.900	296	359	64 Sai	OK
Node8	Sorgo	89.700	333	348	16 Entra	OK
	Protenose	3.900	349	364	16 Sai	OK
Node9	Silagem de Milho	147.900	8	39	32 Entra	OK
	Vaca Magra	5.100	8	39	32 Entra	OK
	Boi Gordo	158.100	40	71	32 Sai	OK
(vazio)	(vazio)	(vazio)				
Total Geral		0	364	1942		

Figura 35 – Conferência do estoque para o caso de teste 4.

Tabela 9 – Categorias de recursos e hierarquia utilizadas para testar todas as funcionalidades implementadas.

Categoria de Recurso	Categoria Pai
AREA	-
BEZERRAS	-
BEZERROS	-
BOVINOS	-
MATRIZES	-
MULTIPARAS	-
MULTIPARAS_PARIDAS	-
NIVEL3	-
NULIPARAS	-
PRIMIPARAS	-
PRIMIPARAS_PARIDAS	PRIMIPARAS
RACAO	-
TOUROS	-

indicadores foram aferidos com sucesso.

Finalizado os testes de unidades, o Módulo Simulador foi integrado ao *Manager* como uma biblioteca carregada pelo gerenciador de dependências *Composer*, a descrição dos passos necessários para realizar a configuração pode ser vista no Apêndice E. Por fim, foi avaliada a integração da aplicação para testar se todos os módulos estavam se comunicando corretamente e se a Plataforma estava funcional. Na Figura 37 é possível ver a tela do Módulo Portal que apresenta o resultado de uma simulação realizada no ambiente de homologação, verificando assim a integração da biblioteca ao restante da aplicação.

5.6 Considerações Finais

No decorrer deste capítulo foram destacados as ferramentas utilizadas na execução da codificação e também detalhes acerca da implementação dos Módulos Desenhista e Simulador. Algumas telas do Módulo Desenhista foram apresentadas para exemplificar o comportamento das funcionalidades construídas. Com relação ao Módulo Simulador foram apresentados alguns dos algoritmos utilizados em rotinas de cálculos e o processo para avaliação do módulo foi detalhado. A descrição de como integrar o Módulo Simulador à Plataforma é descrita no Apêndice E. No próximo capítulo são feitas as considerações finais acerca do trabalho desenvolvido, identificando as limitações enfrentadas e sugestões para trabalhos futuros.

Tabela 10 – Parâmetros e valores utilizados para testar todas as funcionalidades implementadas.

Parâmetro: Valor	Parâmetro: Valor
TAMANHO: 2.50	PLANTIO_DIRETO: 75.00
SISTEMAS_INTEGRADOS: 10.00	OCUPACAO: 0.00
ABATE_ANO: 100.00	EQUIVALENTE_CARACA: 500.00
NUMERO_DE_CABECAS: 10000.00	VENDA_ANO: 1000.00
PROTEINADO: 50.00	INDICE_DE_CONFIANCA: 0.00
SAL_MINERAL: 8000.00	IDADE#BOVINOS: 720.00
PRECO#MATRIZES: 2.67	PRECO_PESO#MATRIZES: 29.67
PESO#MATRIZES: 39.67	PRECO_ARROBA#MATRIZES: 70.00
AREA_PRODUCAO: 182.50	PESO#RACAO: 10.70
MASSA_SECA#RACAO: 640.70	DIA_REBANHO: 364.00
CUSTO_FIXO#BOVINOS: 139.00	PESO#BOVINOS: 28.00
CARCACA#BOVINOS: 28.00	PRECO#BOVINOS: 19.00
PRECO_ARROBA#BOVINOS: 86.54	PRECO_PESO#BOVINOS: 756.13
PESO#BEZERROS: 129.00	PESO#BEZERRAS: 99.00
PRECO#AREA: 43.90	PRECO_PESO#AREA: 23.70
PESO#AREA: 23.70	PRECO_ARROBA#AREA: 29.86
CARCACA#AREA: 23.12	PRECO#RACAO: 29.67
PRECO_PESO#RACAO: 291.67	PRECO_ARROBA#RACAO: 329.67
CARCACA#RACAO: 9.67	CARCACA#MATRIZES: 20.00
PRECO_PESO#MULTIPARA_PARIDA: 29.67	PRECO_ARROBA#MULTIPARA_PARIDA: 70.00
PESO#MULTIPARA_PARIDA: 39.67	PRECO#MULTIPARA_PARIDA: 2.67
CARCACA#MULTIPARA_PARIDA: 20.00	PRECO#MULTIPARAS: 2.67
PRECO_PESO#MULTIPARAS: 29.67	PESO#MULTIPARAS: 39.67
PRECO_ARROBA#MULTIPARAS: 70.00	CARCACA#MULTIPARAS: 20.00
PRECO#NULIPARAS: 2.67	PRECO_PESO#NULIPARAS: 29.67
PESO#NULIPARAS: 39.67	PRECO_ARROBA#NULIPARAS: 70.00
CARCACA#NULIPARAS: 20.00	PRECO#PRIMIPARA_PARIDA: 2.67
PRECO_PESO#PRIMIPARA_PARIDA: 29.67	PRECO_ARROBA#PRIMIPARA_PARIDA: 70.00
PESO#PRIMIPARA_PARIDA: 39.67	CARCACA#PRIMIPARA_PARIDA: 20.00
PRECO#PRIMIPARAS: 2.67	PRECO_PESO#PRIMIPARAS: 29.67
PESO#PRIMIPARAS: 39.67	PRECO_ARROBA#PRIMIPARAS: 70.00
CARCACA#PRIMIPARAS: 20.00	PRECO#PRIMIPARAS_PARIDAS: 2.67
PRECO_PESO#PRIMIPARAS_PARIDAS: 29.67	PRECO_ARROBA#PRIMIPARAS_PARIDAS: 70.00
PESO#PRIMIPARAS_PARIDAS: 39.67	CARCACA#PRIMIPARAS_PARIDAS: 20.00
PRECO_ARROBA#MULTIPARAS_PARIDAS: 70.00	PRECO_PESO#MULTIPARAS_PARIDAS: 29.67
PESO#MULTIPARAS_PARIDAS: 39.67	PRECO#MULTIPARAS_PARIDAS: 2.67
CARCACA#MULTIPARAS_PARIDAS: 20.00	IDADE#BEZERROS: 220.00
IDADE#BEZERRAS: 260.00	IDADE#PRIMIPARAS: 350.00
CO2EQ#PRIMIPARAS: 2.67	CO2EQ_PESO#PRIMIPARAS: 29.67
CO2EQ#BOVINOS: 242.70	CO2EQ_PESO#BOVINOS: 239.67
CO2EQ#AREA: 20.70	CO2EQ_PESO#AREA: 59.84
CO2EQ#NULIPARAS: 17.20	CO2EQ_PESO#NULIPARAS: 85.12
CO2EQ#RACAO: 78.60	CO2EQ_PESO#RACAO: 72.92

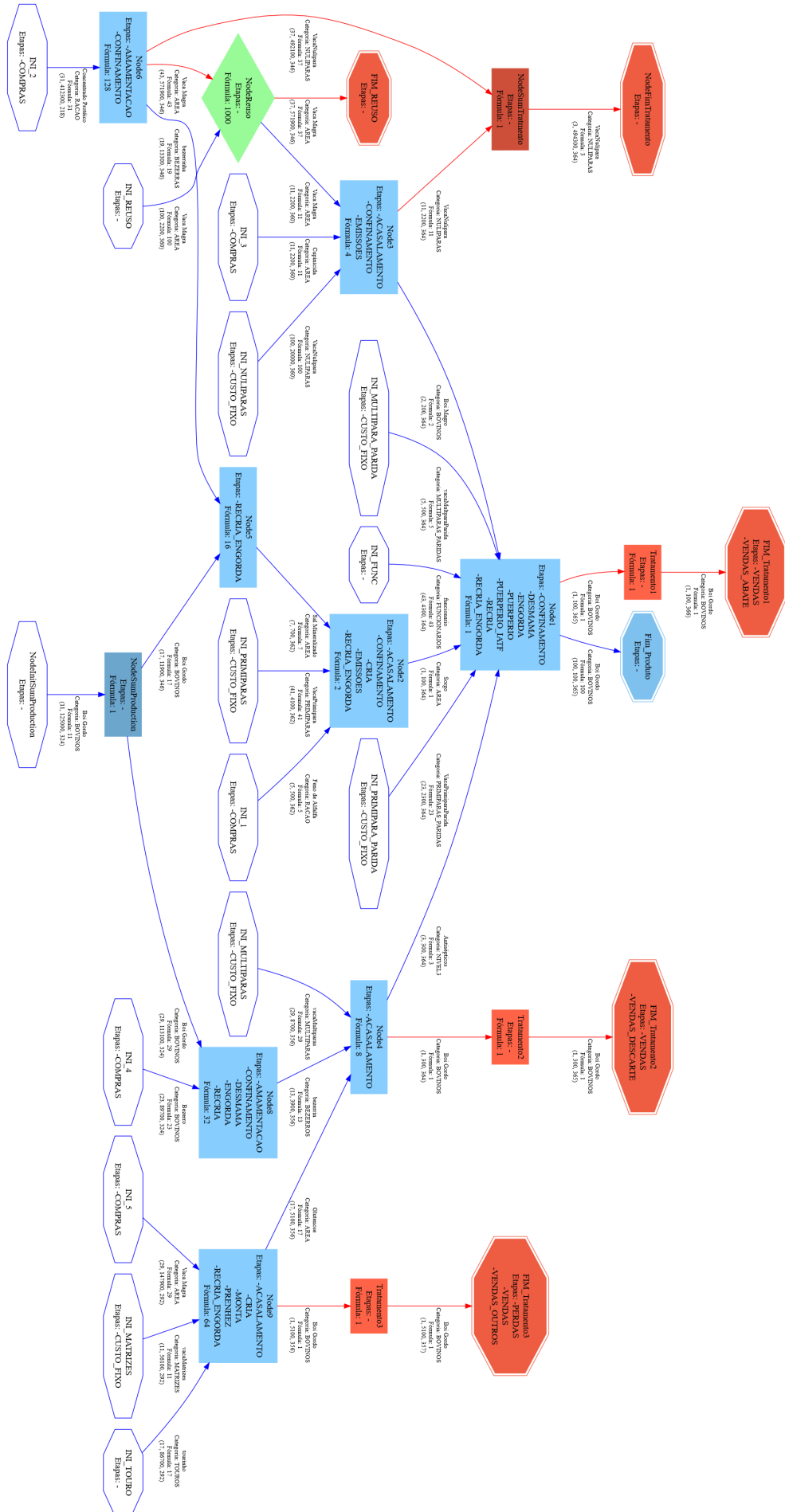


Figura 36 – Modelo de avaliação utilizado no teste dos indicadores.

The screenshot displays a web interface for comparing simulations. At the top right, there are logos for '+Precoce Embrapa' and '+PRECOCE test' with the text 'Sistema de Simulação de Aderência à Protocolos de Raças'. Below the main title 'Comparar Simulações', the user is identified as 'Testando M. Produtor Rurali'. There are two buttons: 'Relatório' (Report) and 'Fechar' (Close). The main content area is a table with two columns: a list of simulation metrics and their corresponding values. The metrics include 'Área', 'Área de produção relativa', 'Área de produção total', 'Produção', 'Vendas', 'Rebanho', 'Consumo', 'Reprodução', 'Econômicos', 'Tempo', 'Compras', 'Emissões', and 'CO2eq'. The values for these metrics are: 147900, 172.38, (blank), (blank), (blank), (blank), (blank), (blank), (blank), (blank), (blank), (blank), and 173248.82039848. At the bottom, there are two 'Select an Option' dropdown menus, each with a 'Comparar' button. A 'Voltar' (Back) button is located at the bottom right.

Metric	Value
Área	147900
Área de produção relativa	172.38
Área de produção total	
Produção	
Vendas	
Rebanho	
Consumo	
Reprodução	
Econômicos	
Tempo	
Compras	
Emissões	
CO2eq	173248.82039848

Figura 37 – Tela do Módulo Portal exibindo o resultado de uma simulação realizada.

6 Conclusões

Inicialmente foi desenvolvido um mapeamento sistemático para a revisão de literatura na busca de publicações que indicassem softwares simuladores para sistemas de produção de bovinos. Obteve-se sucesso nas buscas e doze trabalhos foram identificados. Um dos softwares encontrados, o Embraptec, foi desenvolvido pela Embrapa, tendo sido posteriormente descontinuado.

Para o desenvolvimento dos Módulos Desenhista e Simulador da Plataforma +Precoce, o primeiro passo foi definir um modelo computacional que desse aporte à realização das Simulações. O desenvolvimento deste modelo foi fruto de várias discussões com o analista de negócios da Embrapa, sendo refinado até atingir alto grau de maturidade que permitisse a codificação dos módulos.

Um dos pontos levados em consideração durante a codificação dos módulos é que seu desenvolvimento seria continuado, seja por desenvolvedores da Embrapa, terceirizados ou acadêmicos da UFMS. Sendo assim, as boas práticas mencionadas no Capítulo 2.4 foram seguidas pensando na manutenibilidade do código, o que infere um desenvolvimento com constante refatoração para atingir esses objetivos¹. A utilização de padrões de projeto também corrobora para este fim.

A medida que o desenvolvimento dos módulos era concluído, sua disponibilização em ambiente de homologação era feito via funcionalidades de *Auto-Deploy* propiciadas pelo Titan *Framework*. Um ponto positivo a se destacar, a entrega contínua possibilitada pelos *scripts* de *Auto-Deploy*, agilizaram o processo de entrega. Outras práticas de automatização utilizadas pela Embrapa para prover o ambiente de desenvolvimento com a utilização de *scripts* também agilizaram o início da codificação, pois possibilitam que com poucos comandos seja possível obter um ambiente de desenvolvimento pronto para execução do projeto, sem necessidade de instalar serviços, linguagens e realizar configurações. Todos esses passos são automatizados pelos *scripts*.

Para verificação das funcionalidades do Módulo Desenhista foi realizada avaliação de aceitação com os clientes (pesquisadores da Embrapa). Para o Módulo Simulador avaliações unitárias foram executadas para verificar cada um dos passos da rotina de simulação. Cinco casos de testes foram construídos com a finalidade de avaliar a carga dos modelos computacionais e a execução do cálculo do grafo (execução correta do percurso, cálculo correto das variáveis dos nós e cálculo correto das variáveis dos fluxos). Todos casos foram executados com sucesso. Em seguida, testou-se o cálculo de estoque (para os

¹ O uso de desenvolvimento guiado por testes ou testes automatizados não foi utilizado. Os testes unitários eram feitos verificando as saídas de cada rotina, de acordo com a entrada dada.

mesmos sete casos). Neste ponto o código teve de ser corrigido para não armazenar estoque de quantidades iguais a zero, pois, estas seriam indiferentes nos cálculos de indicadores. Feita a correção, obteve-se sucesso neste teste.

Para verificar o cálculo dos indicadores um modelo completo foi construído de modo a ser possível atingir todas as rotinas de cálculo de indicadores. Esta configuração é obtida informando todas as estações, parâmetros e categorias de recursos utilizada por cada um dos indicadores. Os resultados foram confrontados com os calculados via planilha pelo analista de negócios e foram corrigidas as rotinas nas quais os resultados divergiam até a obtenção de sucesso em todas comparações.

Por fim, a biblioteca de simulação foi integrada ao *Manager* e disponibilizada em ambiente de homologação para avaliação de integração e uso por parte dos demais pesquisadores da Embrapa envolvidos no projeto.

O desenvolvimento deste projeto contribuiu com o avanço da Plataforma +Precoce, pois a abordagem conceitual que se tinha acerca do simulador foi evoluída para um produto final implementado em código, que está apto a progredir de acordo com as necessidades futuras da Plataforma.

A boa comunicação entre os envolvidos proveu visão compartilhada dos objetivos a se atingir com a construção dos módulos, dando coerência às ações desenvolvidas e planejamento de cada uma das etapas. Constante comunicação, entregas rápidas, com envolvimento de usuários na utilização e avaliação da Plataforma durante o desenvolvimento, favoreceram a construção de confiança no produto, garantindo geração de valor a cada etapa.

6.1 Limitações e Sugestões para Trabalhos Futuros

Uma das limitações do projeto foi a ausência de um modelo de sistema real para experimentação da Plataforma, em tempo hábil para se detectar e implementar oportunidades de melhorias. Durante a elaboração de um modelo de sistema real por parte do analista de negócios, percebeu-se que seria conveniente a possibilidade de se utilizar submodelos dentro de um modelo, de modo que um nó do grafo pudesse representar um subgrafo. Essa modificação é necessária para simplificar o trabalho do usuário desenhista ao cadastrar modelos extensos, uma vez que um subgrafo pode se repetir dentro de um ou mais modelos. Além disso, em alguns sistemas o grafo poderia implicar em cálculo circular, na forma como o algoritmo foi definido, quando um nó com dois ou mais fluxos de referência (nós de reuso ou de soma) têm o cálculo de um de seus fluxos de referência dependendo do cálculo de um de seus fluxos de resultado. Uma solução simples para estas situações depende também do uso de submodelos. Essa limitação foi detectada próximo ao fim do prazo do mestrado. A implementação de tratamento de submodelos e a solução

para cálculo circulares ficará a cargo do desenvolvimento em trabalhos futuros.

Uma das etapas do desenvolvimento desse trabalho foi a realização de mapeamento sistemático de literatura visando encontrar publicações que falassem sobre softwares de simulação de sistemas de produção de bovinos. Como trabalho futuro seria interessante realizar revisão de literatura com questões de pesquisa com enfoque em processos da Engenharia de Software. Outra possibilidade seria comparar as funcionalidades dos softwares encontrados no mapeamento com as funcionalidades da Plataforma +Precoce e realizar pesquisa com usuários finais acerca de quais novas funcionalidades seriam interessantes a serem inseridas no projeto.

Com a conclusão dos Módulos Desenhista e Simulador a Plataforma +Precoce está disponível para uso. No entanto, é um projeto com potencial para melhorias, como por exemplo, o aprimoramento da experiência de uso. Questões como usabilidade, acessibilidade e responsividade são temas a serem aprimorados na Plataforma. Ao que tange o Módulo Desenhista os cadastros dos modelos computacionais são baseados em *CRUD's* o que torna a atividade para cadastro dos grafos um processo dispendioso e repetitivo. O desenvolvimento de ferramenta visual para esta etapa facilitaria o uso para o desenhista, evitaria erros de cadastros e agilizaria o processo, tornando-o transparente e intuitivo para o usuário.

No tocante ao Módulo Portal a arquitetura de informação apresentada tem espaço para refatoração, relatórios mais detalhados com os resultados das simulações e gráficos que apresentem os dados ainda faltam ser desenvolvidos.

No que tange ao Módulo Simulador esta é a primeira versão da biblioteca. Muito provavelmente, à medida que novos modelos de sistemas forem sendo traduzidos para o modelo computacional, novas necessidades surgirão, como, a possibilidade de uso de submodelos, mencionada mais acima. É recomendável que se mantenham esforços para constante evolução do módulo.

Referências

- ABIEC. *Perfil da Pecuária no Brasil: Relatório Anual 2016*. São Paulo, 2016.
- ABIEC. *Perfil da Pecuária no Brasil: Relatório Anual 2017*. São Paulo, 2017.
- ABREU, U. de et al. Sistemas intensivos de produção na região tropical brasileira. In: *Anais / X Jornada NESPRO e II Simpósio Internacional sobre Sistemas de Produção de Bovinos de Corte*. Porto Alegre: UFRGS, 2015.
- ALMEIDA, A. J. d. et al. *Novilho precoce: caminho para o sucesso na pecuária*. Campo Grande: Ed. UFMS, 1999.
- APSIM Initiative. *The Agricultural Production Systems sIMulator*. 2016. Disponível em: <<https://www.apsim.info/>>. Acesso em: 01.09.2016.
- BAILEY, J. et al. Evidence relating to object-oriented software design: A survey. In: IEEE. *Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on*. [S.l.], 2007. p. 482–484.
- BATISTA, P. B. *Caracterização bioeconômica de diferentes idades de acasalamento de gado de corte no sudoeste da Bahia e no noroeste do Paraná*. Universidade Estadual do Oeste do Parana, 2009. Disponível em: <http://tede.unioeste.br/tede//tde_busca/arquivo.php?codArquivo=530>.
- BAZZO, W. A.; PEREIRA, L. T. d. V. *Introdução à engenharia: conceitos, ferramentas e comportamentos*. 4ª ed. Florianópolis: Ed. da UFSC, 2013.
- BERNARDI, A. d. C. et al. Potencial de uso das tecnologias de agricultura e pecuária de precisão e automação. *Embrapa Pecuária Sudeste-Documents (INFOTECA-E)*, São Carlos, SP: Embrapa Pecuária Sudeste, 2017., 2017.
- BONATTI, F. K. Q. *Desenvolvimento de modelo dinâmico de simulação para sistemas de recria e terminação de bovinos em pasto*. Universidade Federal de Viçosa, 2009. Disponível em: <<http://locus.ufv.br/handle/123456789/5629>>.
- BOSWELL, D.; FOUCHER, T. *A Arte de Escrever Programas Legíveis: Técnicas Simples e Práticas para a Elaboração de Programas Fáceis de Serem Lidos e Entendidos*. São Paulo: Novatec Editora, 2012.
- BRASIL. *Decreto nº 9.013, de 29 de Março de 2017. Regulamenta a Lei nº 1.283, de 18 de dezembro de 1950, e a Lei nº 7.889, de 23 de novembro de 1989, que dispõem sobre a inspeção industrial e sanitária de produtos de origem animal*. [S.l.], 2017. Disponível em: <http://www.planalto.gov.br/ccivil_03/_Ato2015-2018/2017/Decreto/D9013.htm>.
- CÁCERES, E. N. et al. Computational precision livestock-position paper. In: *II Workshop of the Brazilian Institute for Web Science Research*. Rio de Janeiro: [s.n.], 2011. p. 9.
- CARROMEU, C. *Linha de produtos de software no processo de geração de sistemas Web de apoio a gestão de fomento de projetos*. Dissertação (Mestrado em Ciência da Computação) — Departamento de Computação e Estatística - Universidade Federal de Mato Grosso do Sul, 2007.

- CARROMEU, C. *Titan Framework Development CookBook*. 2016. Disponível em: <<https://svn.cnpgc.embrapa.br/titan/document/Cookbook.pdf>>. Acesso em: 01.11.2016.
- CEZAR, I. M. et al. *Sistemas de produção de gado de corte no Brasil: uma descrição com ênfase no regime alimentar e no abate*. [S.l.]: Embrapa Gado de Corte, 2005.
- CHEN, L.; BABAR, M. A.; ZHANG, H. Towards an evidence-based understanding of electronic data sources. 2010.
- COHN, M. *User Stories Applied: For Agile Software Development*. [S.l.]: Pearson Education, Inc., 2004.
- COHN, M. *Desenvolvimento de Software com Scrum: Aplicando Métodos Ágeis com Sucesso*. Porto Alegre: Bookman, 2011.
- CORRÊA, E. S.; ARRUDA, Z. J. d. *Avaliação preliminar do sistema de produção de gado de corte implantado no CNPGC período: 1983/84 a 1986/87*. Campo Grande: Empresa Brasileira de Pesquisa Agropecuária, Centro Nacional de Pesquisa de Gado de Corte, 1988.
- CORRÊA, E. S. et al. *Produção de carne de qualidade em pastagem: alternativas para o sistema físico da Embrapa Gado de Corte*. Campo Grande: Embrapa Gado de Corte, 2003.
- CORRÊA, E. S. et al. *Sistemas de produção melhorados para gado de corte em Mato Grosso do Sul*. Campo Grande: Embrapa Gado de Corte, 2006.
- CORRÊA, E. S. et al. *Sistema semi-intensivo de produção de carne de bovinos Nelore no Centro-Oeste do Brasil*. [S.l.]: Embrapa Gado de Corte, 2000.
- COSTA, F. P. et al. *Gerenpec: aplicativo para planejamento da fazenda de gado de corte*. [S.l.]: Embrapa Gado de Corte, 2004.
- COSTA, M. A. B.; BARBOSA, P. F. Ferramentas computacionais de apoio à tomada de decisão em sistemas agropecuários e agroindustriais. Embrapa, 1997.
- ERICH, G. et al. *Padrões de Projeto: Soluções Reutilizáveis de Software Orientado a Objetos*. Trad. Luiz AM Salgado. Porto Alegre: Bookman, 2000.
- EUCLIDES FILHO, K. *Produção de bovinos de corte e o trinômio genótipo-ambiente-mercado*. [S.l.]: Embrapa Gado de Corte, 2000.
- EUCLIDES FILHO, K. Cenários para a cadeia produtiva da carne bovina no Brasil. In: ROSA, A. do N.; MARTINS, E. N.; MENEZES, G. R. de O.; SILVA, L. O. C. da (Ed.). *Melhoramento genético aplicado em gado de corte: Programa Geneplus-Embrapa*. Brasília, DF: Embrapa; Campo Grande, MS: Embrapa Gado de Corte. [S.l.]: Embrapa Gado de Corte (CNPGC), 2013. cap. 1.
- EUCLIDES, V. P. B. et al. Desempenho de novilhos f1s angus-nelore em pastagens de brachiaria decumbens submetidos a diferentes regimes alimentares. *Revista brasileira de zootecnia*, SciELO Brasil, v30, n. 2, p. 470–481, 2001.
- FONTOURA JÚNIOR, J. A. S. d. *Modelo de simulação do desempenho reprodutivo de vacas de corte baseado no escore de condição corporal*. Universidade Federal de Pelotas, 2008. Disponível em: <<http://repositorio.ufpel.edu.br/handle/123456789/2628>>.

- FORMIGONI, I. B. *Estimação de valores econômicos para características componentes de índices de seleção em bovinos de corte*. Universidade de São Paulo, 2002. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/74/74131/tde-03092003-133102/>>.
- FREITAS FILHO, P. J. d. *Introdução a modelagem e simulação de sistemas com aplicações em Arena*. 2ª ed. ed. Florianópolis: Visual Books, 2008.
- GANSNER, E.; KOUTSOFIOS, E.; NORTH, S. *Drawing graphs with dot*. 2015. <https://graphviz.gitlab.io/_pages/pdf/dotguide.pdf>. Acessado em 03/06/2018.
- GOLDBARG, M.; GOLDBARG, E. *Grafos: Conceitos, algoritmos e aplicações*. Rio de Janeiro: Elsevier, 2012.
- GOMES, R. d. C. *Plataforma +precoce: sistemas melhorados para a produção do novillo precoce*. 2014.
- GREINER, R. et al. Scenario modelling to support industry strategic planning and decision making. *Environmental Modelling E Software*, v. 55, p. 120 – 131, 2014. ISSN 1364-8152. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1364815214000218>>.
- HOLZWORTH, D. P. et al. APSIM – evolution towards a new generation of agricultural systems simulation. *Environmental Modelling E Software*, v. 62, p. 327 – 350, 2014. ISSN 1364-8152. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1364815214002102>>.
- HOUAISS, A.; VILLAR, M. d. S.; FRANCO, F. M. d. M. *Dicionário Houaiss da Língua Portuguesa*. Rio de Janeiro: Objetiva, 2009.
- IBGE. *Produção da Pecuária Municipal 2015*. Rio de Janeiro: Instituto Brasileiro de Geografia, 2015. v43.
- INAMASU, R. Y.; BERNARDI, A. d. C. Agricultura de precisão. *Embrapa Instrumentação-Capítulo em livro científico (ALICE)*, In: BERNARDI, AC de C.; NAIME, J. de M.; RESENDE, AV de; BASSOI, LH; INAMASU, RY (Ed.). *Agricultura de precisão: resultados de um novo olhar*. Brasília, DF: Embrapa, 2014. p. 21-33., 2014.
- INVERNADA. *Embrapa Invernada*. 2016. Disponível em: <<https://www.invernada.cnptia.embrapa.br/>>. Acesso em: 01.09.2016.
- KEELE, S. Guidelines for performing systematic literature reviews in software engineering. In: *Technical report, Ver. 2.3 EBSE Technical Report. EBSE*. [S.l.]: sn, 2007.
- KNINBERG, H. Scrum e xp direto das trincheiras: Como nós fazemos scrum. *Estados Unidos: C4media*, 2007.
- Manifesto Ágil. *Manifesto para o desenvolvimento ágil de software*. 2016. Disponível em: <<http://www.manifestoagil.com.br/>>. Acesso em: 01.11.2016.
- MARCOS, C. et al. Refactoring of a beef-cattle farm simulator. *IEEE Latin America Transactions*, v. 9, n. 7, p. 1099–1104, 2011. Cited By 1. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-84856135035&doi=10.1109%2fTLA.2011.6129709&partnerID=40&md5=962aa903803795be591e22665d8174e4>>.
- MARQUES, D. d. C. *Criação de bovinos*. São Paulo: Nobel, 1969.

- MARTIN, R. C. *Código Limpo: Habilidades Práticas do Agile Software*. Rio de Janeiro: Alta Books, 2011.
- MASSRUHÁ, S. M. F. S. et al. *Tecnologias da informação e comunicação e suas relações com a agricultura*. Brasília: Embrapa, 2014.
- MEIRELLES, F. D. P. *Modelo computacional de um rebanho bovino de corte virtual utilizando simulação Monte Carlo e redes neurais artificiais*. Universidade de São Paulo, 2005. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/74/74131/tde-03052005-094734/>>.
- MELO FILHO, G. et al. *Sistemas de produção praticados e sistemas melhorados propostos-Bacia do Rio Formoso, Bonito, MS*. Campo Grande: Embrapa Gado de Corte, 2006.
- MOLIN, J. P. Desafios da agricultura brasileira a partir da agricultura de precisão. *Simpósio sobre rotação soja/milho no plantio direto*, v. 3, p. 49–59, 2002.
- MOREIRA, H. L. *Objetivos de seleção e valores econômicos para bovinos Nelore em sistema de ciclo completo*. Universidade de São Paulo, 2015. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/17/17135/tde-06012016-164024/>>.
- MUJTABA, S. et al. Software product line variability: A systematic mapping study. *School of Engineering, Blekinge Inst. of Technology*, 2008.
- PASIM. *The Pasture Simulation Model (PaSim)*. 2016. Disponível em: <<https://www1.clermont.inra.fr/urep/modeles/pasim.htm>>. Acesso em: 01.09.2016.
- PEREIRA, C. H. *Simulação da sustentabilidade econômica e ambiental em três sistemas de produção de bovinos de corte do RS*. [S.l.]: Universidade Federal do Rio Grande do Sul, 2015.
- PETERSEN, K. et al. Systematic mapping studies in software engineering. In: *EASE*. [S.l.: s.n.], 2008. v. 8, p. 68–77.
- PETERSEN, K.; VAKKALANKA, S.; KUZNIARZ, L. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, Elsevier, v. 64, p. 1–18, 2015.
- PIRES, A. V. *Bovinocultura de corte*. Piracicaba: FEALQ, 2010. v1.
- PIRES, J. L. F. et al. *Discutindo agricultura de precisão-aspectos gerais*. Passo Fundo: Embrapa Trigo, 2004.
- PIRES, P. P.; DOS SANTOS NETO, Q. I.; COSTA JUNIOR, J. B. G. Histórico e desafios futuros da pesquisa em pecuária de precisão na embrapa gado de corte. In: *Anais 1º Simpósio Brasileiro de Pecuária de Precisão Aplicada à Bovinocultura de Corte*. Campo Grande: Embrapa Gado de Corte, 2014.
- SANTANA, R. A. V. *Desempenho bioeconômico de sistemas intensivos de cria e de ciclo completo por meio de simulação*. [s.n.], 2012. Disponível em: <<http://repositorio.unb.br/handle/10482/10841>>.

- SANTOS, A. R. d. et al. Sw agro: Estudo do mercado brasileiro de software para o agronegócio. *Estudo do mercado brasileiro de software para o agronegócio*. Campinas: Embrapa Informática Agropecuária, 2011.
- SHORE, J.; WARDEN, S. *A Arte do Desenvolvimento Ágil*. Rio de Janeiro: Alta Books, 2008.
- Sociedade Nacional de Agricultura. *Brasil será o maior produtor mundial de carne bovina em cinco anos, prevê Abiec*. 2016. Disponível em: <<http://sna.agr.br/brasil-sera-o-maior-produtor-mundial-de-carne-bovina-em-5-anos-preve-abiec/>>. Acesso em: 01.11.2016.
- SUTHERLAND, J. *Scrum: a arte de fazer o dobro do trabalho na metade do tempo*. São Paulo: Leya, 2016.
- Titan Framework. *Titan Framework*. 2016. Disponível em: <<http://www.titanframework.com/>>. Acesso em: 01.11.2016.
- VARELA, F.; BUSSONI, A.; DIEGUEZ, F. Desarrollo de un simulador para manejo de producciones silvopastoriles sobre campo natural. In: *3 Congreso Nacional de Sistemas Silvopastoriles - VIII Congreso Internacional de Sistemas Agroforestales*. Iguazú, Misiones - Argentina: EEA INTA, 2015.
- WATHES, C. et al. Is precision livestock farming an engineer's daydream or nightmare, an animal's friend or foe, and a farmer's panacea or pitfall? *Computers and electronics in agriculture*, Elsevier, v. 64, n. 1, p. 2–10, 2008.
- WIERINGA, R. et al. Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. *Requirements Engineering*, Springer, v. 11, n. 1, p. 102–107, 2006.

Apêndices

APÊNDICE A – Mapeamento Sistemático - Filtro 1

Este apêndice apresenta o resultado obtido na execução do filtro 1, que visa à obtenção dos trabalhos relevantes à pesquisa. Na Tabela 11 é possível ver a relação de todos os trabalhos encontrados após a execução das buscas e os critérios de exclusão adotados. Os trabalhos que possuem alguma marcação na coluna “Critério(s) de Exclusão” foram descartados, e os que não apresentam nenhuma indicação foram mantidos para a próxima etapa da triagem.

Tabela 11 – Filtro 1 - Critérios de exclusão

ID	Base de Dados Eletrônicos	Trabalho	Critério(s) de Exclusão	Link	Referência Bibliográfica
T1	SCOPUS	Analyzing ranch profitability from varying cow sales and heifer replacement rates for beef cow-calf production using system dynamics		https://goo.gl/j008w3	TURNER, B. et al. Analyzing ranch profitability from varying cow sales and heifer replacement rates for beef cow-calf production using system dynamics. Agricultural Systems, 2013.
T2	SCOPUS	Implications of deep drainage through saline clay for groundwater recharge and sustainable cropping in a semi-arid catchment, Australia		https://goo.gl/DTEQW1	TIMMS, W.; YOUNG, R.; HUTH, N. Implications of deep drainage through saline clay for groundwater recharge and sustainable cropping in a semi-arid catchment, Australia. Hydrology and Earth System Sciences, 2012.

Tabela 11 – continuação da página anterior

ID	Base de Dados Eletrônicos	Trabalho	Critério(s) de Exclução	Link	Referência Bibliográfica
T3	SCOPUS	Refactoring of a beef-cattle farm simulator		https://goo.gl/B9d1MH	MARCOS, C. et al. Refactoring of a beef-cattle farm simulator. IEEE Latin American Transactions, 2011.
T4	SCOPUS	A component-based framework for simulating agricultural production and externalities		https://goo.gl/DFdkpY	DONATELLI, M. et al. A component-based framework for simulating agricultural production and externalities. [s.n.], 2010.
T5	SCOPUS	Combined use of simulation modeling and linear programming for logistic design of the slurry distribution systems	CE6	https://goo.gl/ru5FfS	BUSATO, P.; BERRUTO, R. Combined use of simulation modeling and linear programming for logistic design of the slurry distribution systems. 2009.
T6	SCOPUS	Agile development for a beef-cattle farm simulator	CE2	https://goo.gl/h2Z4yy	ARROQUI, M. et al. Agile development for a beef-cattle farm simulator. IEEE Latin America Transactions, 2009.
T7	SCOPUS	Opportunities and trade-offs in dual-purpose cereals across the southern Australian mixed-farming zone: A modelling study	CE5, CE6	https://goo.gl/1SpTRO	MOORE, A. Opportunities and trade-offs in dual-purpose cereals across the southern Australian mixed-farming zone: A modelling study. Animal Production Science, 2009.
Continua na próxima página					

Tabela 11 – continuação da página anterior

ID	Base de Dados Eletrônicos	Trabalho	Critério(s) de Exclução	Link	Referência Bibliográfica
T8	SCOPUS	Simulation of odour dispersion downwind from natural windbreaks using the computational fluid dynamics standard k-E model	CE4, CE6	< https://goo.gl/g1PGXk >	LIN, X. et al. Simulation of odour dispersion downwind from natural windbreaks using the computational fluid dynamics standard k-E model. Canadian Journal of Civil Engineering, 2009.
T9	SCOPUS	QMSim: A large-scale genome simulator for livestock		< https://goo.gl/1odtQS >	SARGOLZAEI, M.; SCHENKEL, F. QMSim: A large-scale genome simulator for livestock. Bioinformatics, 2009.
T10	SCOPUS	Assessing alternative fecal coliform direct deposit modeling approaches	CE6	< https://goo.gl/SU47Qy >	HALL, K. et al. Assessing alternative fecal coliform direct deposit modeling approaches. 2008.
T11	SCOPUS	Design of a multimodel of a dairy cows herd attacked by mastitis		< https://goo.gl/4q18gh >	FORCE, C.; PÉROCHON, L.; HILL, D. Design of a multimodel of a dairy cows herd attacked by mastitis. Simulation Modelling Practice and Theory, 2002.
T12	SCOPUS	The food and agricultural policy simulator (USA).	CE6	< https://goo.gl/rsVQOu >	SALATHE, L.; PRICE, J.; GADSON, K. The food and agricultural policy simulator (USA). Agricultural Economics Research, 1982.
Continua na próxima página					

Tabela 11 – continuação da página anterior

ID	Base de Dados Eletrônicos	Trabalho	Critério(s) de Exclução	Link	Referência Bibliográfica
T13	ACM Digital Library	Autonomous gathering of livestock using a multi-functional sensor network platform		< https://goo.gl/IHWXiK >	DONIEC, M. et al. Autonomous gathering of livestock using a multi-functional sensor network platform. In: Proceedings of the 6th Workshop on Hot Topics in Embedded Networked Sensors. New York, NY, USA: ACM, 2010.
T14	ACM Digital Library	GASP iv simulation of flush water recycling systems		< https://goo.gl/H2wOLV >	FEHR, R. L. et al. Gasp iv simulation of flush water recycling systems. In: Proceedings of the 9th Conference on Winter Simulation - Volume 2. Winter Simulation Conference, 1977.
T15	Periódicos CAPES	Refactoring of a Beef-Cattle Farm Simulator	CE3	< https://goo.gl/RKLe76 >	MARCOS, C. et al. Refactoring of a beef-cattle farm simulator. IEEE Latin America Transactions, 2011.
T16	Periódicos CAPES	Agile Development for a Beef-Cattle Farm Simulator	CE3	< https://goo.gl/LVzTUs >	ARROQUI, M. et al. Agile development for a beef-cattle farm simulator. IEEE Latin America Transactions, 2009.
T17	Periódicos CAPES	QMSim: a large-scale genome simulator for livestock	CE3	< https://goo.gl/ZV2tm5 >	SARGOLZAEI, M.; SCHENKEL, F. S. Qmsim: a large-scale genome simulator for livestock. Bioinformatics, 2009.
Continua na próxima página					

Tabela 11 – continuação da página anterior

ID	Base de Dados Eletrônicos	Trabalho	Critério(s) de Exclução	Link	Referência Bibliográfica
T18	Periódicos CAPES	Development and evaluation of an integrated simulation model for assessing smallholder crop–livestock production in Yucatán, Mexico		https://goo.gl/L17Nn2	PARSONS, D. et al. Development and evaluation of an integrated simulation model for assessing smallholder crop–livestock production in yucatán, mexico. Agricultural Systems, 2011.
T19	Periódicos CAPES	APSIM - Evolution towards a new generation of agricultural systems simulation		https://goo.gl/Kl3ey	HOLZWORTH, D. P. et al. APSIM – evolution towards a new generation of agricultural systems simulation. Environmental Modelling & Software, 2014.
T20	Periódicos CAPES	Scenario modelling to support industry strategic planning and decision making		https://goo.gl/xEx2A8	GREINER, R. et al. Scenario modelling to support industry strategic planning and decision making. Environmental Modelling & Software, 2014.
T21	Periódicos CAPES	Implications of deep drainage through saline clay for groundwater recharge and sustainable cropping in a semi-arid catchment, Australia	CE3	https://goo.gl/OUPzr0	TIMMS, W. A.; YOUNG, R. R.; HUTH, N. Implications of deep drainage through saline clay for groundwater recharge and sustainable cropping in a semi-arid catchment, australia. Hydrology and Earth System Sciences, 2012.

Continua na próxima página

Tabela 11 – continuação da página anterior

ID	Base de Dados Eletrônicos	Trabalho	Critério(s) de Exclução	Link	Referência Bibliográfica
T22	Periódicos CAPES	Implications of deep drainage through saline clay for groundwater recharge and sustainable cropping in a semi-arid catchment, Australia	CE3	< https://goo.gl/vbhnu4 >	TIMMS, W. A.; YOUNG, R. R.; HUTH, N. Implications of deep drainage through saline clay for groundwater recharge and sustainable cropping in a semi-arid catchment, australia. Hydrology and Earth System Sciences, 2012.
T23	Periódicos CAPES	Design of a multimodel of a dairy cows herd attacked by mastitis	CE3	< https://goo.gl/kFxFt04 >	FORCE, C.; PÉROCHON, L.; HILL, D. Design of a multimodel of a dairy cows herd attacked by mastitis. Simulation Modelling Practice and Theory, 2002.
T24	Periódicos CAPES	An Overview of NEMPIS: National Economic Milk Policy Impact Simulator		< https://goo.gl/UDV96v >	KAISER, H. An Overview of NEMPIS: National Economic Milk Policy Impact Simulator. [S.l.], 1992.
T25	Periódicos CAPES	Effects os U.S. dairy policies on markets for milk and dairy products		< https://goo.gl/N43gVy >	PRICE, J. M. Effects of U.S. Dairy Policies on Markets for Milk and Dairy Products. [S.l.], 2004.
T26	Periódicos CAPES	The effects of porcine somatotropin on the consumption of soybean meal		< https://goo.gl/WHjhKp >	SHUKER, I. G.; APLAND, J.; CORNELIUS, S. The Effects of Porcine Somatotropin on the Consumption of Soybean Meal. [S.l.], 1990.
Continua na próxima página					

Tabela 11 – continuação da página anterior

ID	Base de Dados Eletrônicos	Trabalho	Critério(s) de Exclusão	Link	Referência Bibliográfica
T27	Periódicos CAPES	An economic analysis of contracting arrangements used by the Minnesota Turkey Industry		< https://goo.gl/7L9uwx >	DORNBUSH, C. W.; BOEHLJE, M. An Economic Analysis of Contracting Arrangements used by the Minnesota Turkey Industry. [S.l.], 1988.
T28	BDTD	Simulação da sustentabilidade econômica e ambiental em três sistemas de produção de bovinos de corte do RS		< https://goo.gl/ScXaD1 >	PEREIRA, C. H. Simulação da sustentabilidade econômica e ambiental em três sistemas de produção de bovinos de corte do RS. [S.l.]: Universidade Federal do Rio Grande do Sul, 2015.
T29	BDTD	Objetivos de seleção e valores econômicos para bovinos Nelore em sistema de ciclo completo		< https://goo.gl/vj7KEs >	MOREIRA, H. L. Objetivos de seleção e valores econômicos para bovinos Nelore em sistema de ciclo completo. Universidade de São Paulo, 2015.
T30	BDTD	Estimação de valores econômicos para características componentes de índices de seleção em bovinos de corte.		< https://goo.gl/ZyWVjR >	FORMIGONI, I. B. Estimação de valores econômicos para características componentes de índices de seleção em bovinos de corte. Universidade de São Paulo, 2002.
Continua na próxima página					

Tabela 11 – continuação da página anterior

ID	Base de Dados Eletrônicos	Trabalho	Critério(s) de Exclução	Link	Referência Bibliográfica
T31	BDTD	Modelo de simulação do desempenho reprodutivo de vacas de corte baseado no escore de condição corporal		< https://goo.gl/oYGdM9 >	FONTOURA JÚNIOR, J. A. S. da. Modelo de simulação do desempenho reprodutivo de vacas de corte baseado no escore de condição corporal. Universidade Federal de Pelotas, 2008.
T32	BDTD	Modelo computacional de um rebanho bovino de corte virtual utilizando simulação Monte Carlo e redes neurais artificiais		< https://goo.gl/uR5imf >	MEIRELLES, F. D. P. Modelo computacional de um rebanho bovino de corte virtual utilizando simulação Monte Carlo e redes neurais artificiais. Universidade de São Paulo, 2005.
T33	BDTD	Estudo de diferentes estruturas de grupos genéticos aditivos visando ao aumento da eficiência de seleção em bovinos de corte		< https://goo.gl/RIq10i >	JÚNIOR, G. A. de O. Estudo de diferentes estruturas de grupos genéticos aditivos visando ao aumento da eficiência de seleção em bovinos de corte. Universidade de São Paulo, 2013.
Continua na próxima página					

Tabela 11 – continuação da página anterior

ID	Base de Dados Eletrônicos	Trabalho	Critério(s) de Exclução	Link	Referência Bibliográfica
T34	BDTD	EAGGLE: a linear programming model for optimizing mitigation strategies of greenhouse gases emissions in beef cattle production systems.		< https://goo.gl/tGTLfx >	SILVA, R. de O. EAGGLE: a linear programming model for optimizing mitigation strategies of greenhouse gases emissions in beef cattle production systems. [s.n.], 2013.
T35	BDTD	Desenvolvimento de um modelo bioeconômico para determinação de ponderadores econômicos utilizados em índices de seleção em gado de corte.	CE4, CE6	< https://goo.gl/G9ZKau >	BRUMATTI, R. C. Desenvolvimento de um modelo bioeconômico para determinação de ponderadores econômicos utilizados em índices de seleção em gado de corte. Universidade de São Paulo, 2002.
T36	BDTD	Caracterização bioeconômica de diferentes idades de acasalamento de gado de corte no sudoeste da Bahia e no noroeste do Paraná		< https://goo.gl/QdI3R1 >	BATISTA, P. B. Caracterização bioeconômica de diferentes idades de acasalamento de gado de corte no sudoeste da Bahia e no noroeste do Paraná. Universidade Estadual do Oeste do Paraná, 2009.

Continua na próxima página

Tabela 11 – continuação da página anterior

ID	Base de Dados Eletrônicos	Trabalho	Critério(s) de Exclução	Link	Referência Bibliográfica
T37	BDTD	Desenvolvimento de modelo dinâmico de simulação para sistemas de recria e terminação de bovinos em pasto		< https://goo.gl/880HYh >	BONATTI, F. K. Q. Desenvolvimento de modelo dinâmico de simulação para sistemas de recria e terminação de bovinos em pasto. Universidade Federal de Viçosa, 2009.
T38	BDTD	Seleção genômica em bovinos de corte		< https://goo.gl/4zMY8G >	PICCOLI, M. L. Seleção genômica em bovinos de corte. Universidade Federal do Rio Grande do Sul, 2015.
T39	BDTD	Desenvolvimento de quadro magnético reprodutivo e software para bovinos		< https://goo.gl/1vDPLm >	ZAMPIÉR, D. J. Desenvolvimento de quadro magnético reprodutivo e software para bovinos. Universidade Federal de Viçosa, 2015.
T40	BDTD	Desempenho bioeconômico de sistemas intensivos de cria e de ciclo completo por meio de simulação		< https://goo.gl/Q7EZvZ >	SANTANA, R. A. V. Desempenho bioeconômico de sistemas intensivos de cria e de ciclo completo por meio de simulação. [s.n.], 2012.
T41	BDTD	Avaliação genética de bovinos de corte compostos		< https://goo.gl/zTG7U8 >	BUENO, R. S. Avaliação genética de bovinos de corte compostos. Universidade Federal de Viçosa, 2007.
Continua na próxima página					

Tabela 11 – continuação da página anterior

ID	Base de Dados Eletrônicos	Trabalho	Critério(s) de Exclução	Link	Referência Bibliográfica
T42	BDTD	Efeito da escala e do modelo de análise na avaliação genética de escores visuais: estudo de simulação		< https://goo.gl/TUhJUu >	BARICHELLO, F. Efeito da escala e do modelo de análise na avaliação genética de escores visuais: estudo de simulação. Universidade Estadual Paulista (UNESP), 2011.
T43	BDTD	Análise econômica de sistemas agroflorestais como alternativa para ovinocultores no sudoeste paulista		< https://goo.gl/CnFP7G >	GONZÁLEZ, L. R. Análise econômica de sistemas agroflorestais como alternativa para ovinocultores no sudoeste paulista. Universidade de São Paulo, 2014.
T44	BDTD	Avaliação econômica e simulação em sistemas agroflorestais		< https://goo.gl/W0OcFR >	CORDEIRO, S. A. Avaliação econômica e simulação em sistemas agroflorestais. Universidade Federal de Viçosa, 2010.
T45	BDTD	Avaliação da dinâmica do efetivo bovino no Estado do Mato Grosso e seu impacto no controle da brucelose bovina		< https://goo.gl/Q9zB6V >	KURODA, R. B. dos S. Avaliação da dinâmica do efetivo bovino no Estado do Mato Grosso e seu impacto no controle da brucelose bovina. Universidade de São Paulo, 2012.
Continua na próxima página					

Tabela 11 – continuação da página anterior

ID	Base de Dados Eletrônicos	Trabalho	Critério(s) de Exclução	Link	Referência Bibliográfica
T46	BDTD	Modelo locacional dinâmico para a cadeia agroindustrial da carne bovina brasileira		< https://goo.gl/PZMnHj >	ZUCCHI, J. D. Modelo locacional dinâmico para a cadeia agroindustrial da carne bovina brasileira. Universidade de São Paulo, 2010.
T47	BDTD	Strategies to improve the efficiency of genomic selection in animal breeding programs		< https://goo.gl/OvhQn5 >	NEVES, H. H. d. R. Strategies to improve the efficiency of genomic selection in animal breeding programs. Universidade Estadual Paulista (UNESP), 2013.
T48	BDP@	Agronegócio, tecnologia e inovação: anais.	CE4	< https://goo.gl/RjYCEB >	CONGRESSO BRASILEIRO DE AGROINFORMÁTICA; SIMPÓSIO BRASILEIRO DE TECNOLOGIA DA INFORMÁTICA NO AGRONEGÓCIO COOPERATIVO. Agronegócio, tecnologia e inovação: anais. [S.l.]: Embrapa, 2006.
T49	BDP@	Anais do I Simpósio Nacional de Instrumentação Agropecuária	CE4	< https://goo.gl/KZpz1c >	SIMPÓSIO NACIONAL DE INSTRUMENTAÇÃO AGROPECUÁRIA. Anais do I Simpósio Nacional de Instrumentação Agropecuária. [S.l.]: Embrapa, 1997.

Continua na próxima página

Tabela 11 – continuação da página anterior

ID	Base de Dados Eletrônicos	Trabalho	Critério(s) de Exclução	Link	Referência Bibliográfica
T50	BDP@	Modelos para a tomada de decisões na produção de bovinos e ovinos.	CE4	< https://goo.gl/nsUgL4 >	EVERLING, D. M. et al. Modelos para a tomada de decisões na produção de bovinos e ovinos. [S.l.]: Embrapa, 2002.
T51	BDP@	Integração de um modelo de simulação dinâmica e um banco de dados relacional para tomada de decisões na bovinocultura de corte.	CE4	< https://goo.gl/a6LsZJ >	TORRES, G. D. et al. Integração de um modelo de simulação dinâmica e um banco de dados relacional para tomada de decisões na bovinocultura de corte. In: Congresso Brasileiro de Agroinformática. [S.l.: s.n.], 2007.
T52	BDP@	Desenvolvimento de um jogo educativo baseado em um simulador de sistema de produção de bovinos de corte	CE6	< https://goo.gl/susofz >	BORTOLUCCI JÚNIOR, J.; MANCINI, A. L. Desenvolvimento de um jogo educativo baseado em um simulador de sistema de produção de bovinos de corte. Embrapa.
T53	Infoteca-e	Anais do I Seminário Sul-Brasileiro de Informática na Agricultura	CE6	< https://goo.gl/1fn7C7 >	SEMINÁRIO SUL-BRASILEIRO DE INFORMÁTICA NA AGRICULTURA. Anais do I Seminário Sul-Brasileiro de Informática na Agricultura. [S.l.]: Embrapa, 1996.
Continua na próxima página					

Tabela 11 – continuação da página anterior

ID	Base de Dados Eletrônicos	Trabalho	Critério(s) de Exclução	Link	Referência Bibliográfica
T54	Infoteca-e	Relatório de atividades 2009-2011: Embrapa Informática Agropecuária.	CE6	< https://goo.gl/F6fDw2 >	EMBRAPA INFORMÁTICA AGROPECUÁRIA. Relatório de atividades 2009-2011: Embrapa Informática Agropecuária. [S.l.]: Embrapa, 2012.
T55	Infoteca-e	Simulação da evolução do rebanho considerando potencial de crescimento e habilidade materna como critérios de seleção em bovinos de corte.		< https://goo.gl/kqcu94 >	PAULA, E. J. H. de et al. Simulação da evolução do rebanho considerando potencial de crescimento e habilidade materna como critérios de seleção em bovinos de corte. Archives of Veterinary Science, 2009.
T56	Infoteca-e	Memória do Workshop O Cerrado no Contexto das Mudanças Climáticas Globais.	CE6	< https://goo.gl/iT6q82 >	COSTA, C. J. et al. Memória do Workshop O Cerrado no Contexto das Mudanças Climáticas Globais. [S.l.]: Embrapa, 2010.
T57	Infoteca-e	Relatório anual de atividades da Empresa Brasileira de Pesquisa Agropecuária: gestão 1997.	CE6	< https://goo.gl/iu5X6N >	EMBRAPA. Relatório anual de atividades Empresa Brasileira de Pesquisa Agropecuária: gestão 1997. [S.l.]: Embrapa, 1998.
Continua na próxima página					

Tabela 11 – continuação da página anterior

ID	Base de Dados Eletrônicos	Trabalho	Critério(s) de Exclução	Link	Referência Bibliográfica
T58	Infoteca-e	Biocombustíveis sólidos: fonte energética alternativa visando à recuperação de áreas degradadas e à conservação do Bioma Caatinga.		< https://goo.gl/p4KrJs >	TAVARES, S. R. d. L. Biocombustíveis sólidos: fonte energética alternativa visando à recuperação de áreas degradadas e à conservação do Bioma Caatinga. [S.l.]: Embrapa, 2014.
T59	Infoteca-e	A pesquisa florestal na Embrapa 1978-1993: versão preliminar.		< https://goo.gl/P1IuUH >	PINTO JÚNIOR, J. E.; FERREIRA, C. A. A pesquisa florestal na Embrapa 1978-1993: versão preliminar. [S.l.]: Embrapa, 2008.
T60	Alice	Ferramentas computacionais de apoio à tomada de decisão em sistemas agropecuários e agroindustriais.		< https://goo.gl/eStCtm >	COSTA, M. A. B.; BARBOSA, P. F. Ferramentas computacionais de apoio à tomada de decisão em sistemas agropecuários e agroindustriais. Embrapa.
T61	Alice	Tecnologias da informação e comunicação e suas relações com a agricultura.		< https://goo.gl/J2XYan >	MASSRUHÁ, S. M. F. S. et al. Tecnologias da informação e comunicação e suas relações com a agricultura. Brasília: Embrapa, 2014.
Continua na próxima página					

Tabela 11 – continuação da página anterior

ID	Base de Dados Eletrônicos	Trabalho	Critério(s) de Exclução	Link	Referência Bibliográfica
T62	Alice	Resumos da X Mostra de Estagiários e Bolsistas da Embrapa Informática Agropecuária	CE6	https://goo.gl/MDJNVt	MASSRUHÁ, S. M. F. S.; OLIVEIRA, S. R. d. M.; CROCE, M. G. Resumos da X Mostra de Estagiários e Bolsistas da Embrapa Informática Agropecuária. [S.l.]: Embrapa, 2014.
T63	Alice	Anais do SIAGRO: ciência, inovação e mercado 2014.		https://goo.gl/3ISLJr	VAZ, C. M. P.; MILORI, D. M. B. P.; CRESTANA, S. Anais do SIAGRO: ciência, inovação e mercado 2014. [S.l.]: Embrapa, 2014.
T64	Alice	Resumos da IX Mostra de Estagiários e Bolsistas da Embrapa Informática Agropecuária	CE6	https://goo.gl/OujLjx	MASSRUHÁ, S. M. F. S. et al. Resumos da IX Mostra de Estagiários e Bolsistas da Embrapa Informática Agropecuária. [S.l.]: Embrapa, 2013.
T65	Alice	Análise da produção de embriões na fertilização in vitro e transferência de embriões para doadoras Nelore.		https://goo.gl/hH2fEx	BELTRAME, R. T. et al. Análise da produção de embriões na fertilização in vitro e transferência de embriões para doadoras nelore. Ciência Animal Brasileira, 2010.
T66	Alice	Outros equipamentos e métodos.		https://goo.gl/jNOZnJ	BISCEGLI, C. I. et al. Outros equipamentos e métodos. [S.l.]: Embrapa, 1996.
Continua na próxima página					

Tabela 11 – continuação da página anterior

ID	Base de Dados Eletrônicos	Trabalho	Critério(s) de Exclução	Link	Referência Bibliográfica
T67	Alice	Sistema plantio direto: evolução e implementação.		< https://goo.gl/5qFYOK >	DENARDIN, J. E. et al. Sistema plantio direto: evolução e implementação. [S.l.]: Embrapa, 2011.
T68	Alice	Desarrollo de un simulador para manejo de producciones silvopastoriles sobre campo natural.		< https://goo.gl/j0qGfu >	VARELA, F.; BUSSONI, A.; DIEGUEZ, F. Desarrollo de un simulador para manejo de producciones silvopastoriles sobre campo natural. In: 3 Congreso Nacional de Sistemas Silvopastoriles - VIII Congreso Internacional de Sistemas Agroforestales. Iguazú, Misiones - Argentina: EEA INTA, 2015.
T69	Alice	Anais do Congresso Brasileiro de Soja		< https://goo.gl/tZgfJg >	CONGRESSO BRASILEIRO DE SOJA. Anais do Congresso Brasileiro de Soja. [S.l.]: Embrapa, 1999.
T70	Publicações Embrapa Gado de Corte	Gerenpec: aplicativo para planejamento da fazenda de gado de corte.		< https://goo.gl/Yvu3r9 >	COSTA, F. P. et al. Gerenpec: aplicativo para planejamento da fazenda de gado de corte. [S.l.]: Embrapa Gado de Corte, 2004.
Continua na próxima página					

Tabela 11 – continuação da página anterior

ID	Base de Dados Eletrônicos	Trabalho	Critério(s) de Exclução	Link	Referência Bibliográfica
T71	Publicações Embrapa Gado de Corte	Controlpec 1.0: controle financeiro simplificado para a fazenda de pecuária de corte.		https://goo.gl/GDpRkP	COSTA, F. P.; CORRÊA, E. S. Controlpec 1.0: Controle financeiro simplificado para a fazenda de pecuária de corte. Campo Grande: Embrapa Gado de Corte, 2006.

APÊNDICE B – Mapeamento Sistemático - Filtro 3

Este apêndice apresenta comentários resumidos de cada um dos trabalhos lidos integralmente na realização do último filtro do processo de triagem, indicando se o trabalho em questão atendeu ou não o critério de inclusão CI1.

T3 (MARCOS et al., 2011): O trabalho explicita o processo de refatoração de um simulador de fazenda de gado de corte, justificando a necessidade em termos da qualidade do software e demonstra o método utilizado, o porquê de se ter tomado a decisão de migrar a programação para uma abordagem de orientação a aspectos, dificuldades encontradas e pormenores de outras decisões tomadas. O objetivo do software é obter a projeção de uma empresa agropecuária que tenha como base a produção de gado em sistemas de pastagem. O trabalho se enquadra no critério de inclusão CI1.

T19 (HOLZWORTH et al., 2014): Este trabalho trata do simulador APSIM, software com mais de vinte anos de existência, que evoluiu para se tornar um *framework* de aporte para muitos aspectos da produção agrícola, desde a simulação gênica até simulação de fazendas com vários campos de atuação, dentre outros. O objetivo do trabalho foi de atualizar uma publicação anterior, de 2003, em que eram descritos os principais atributos do APSIM, sendo assim os autores relatam as transformações e a evolução para o que eles chamam de “próxima geração” de *frameworks* de simulação. O APSIM foi construído de modo que vários modelos de simulação possam ser acoplados e que permita a troca de informação entre esses modelos. Outro aspecto é que existe uma grande comunidade envolvida e atuante no trabalho com o simulador, seja compartilhando ideias, dados ou código. Existe o suporte a modelos animais que oferecem uma diversidade de opções para a simulação de gado bovino e ovino em sistemas agrícolas. Os autores destacam o modelo GRAZPLAN para biologia de ruminantes que oferece aporte a grande variedade de genótipos bovinos, estruturas de rebanho, gestão comercial, reprodutiva, de pastagens, suplementação alimentar, etc. O trabalho se enquadra no critério de inclusão CI1.

T20 (GREINER et al., 2014): Apresenta o software simulador *Pastoral Properties Futures Simulator* (PPFS), cujo foco é servir de ferramenta de apoio estratégico em face às incertezas ambientais, de mercado e institucionais às quais a indústria australiana enfrenta. O software foi implementado utilizando a ferramenta Stella e quatro modelos de fazendas foram desenvolvidos para capturar o que se considerou características típicas empresariais e dos sistemas de produção de gado na região. O simulador conta com o módulo de “gado bovino” responsável por simular a dinâmica do rebanho. Os autores listam os módulos

do sistema e dão detalhes da arquitetura e parâmetros utilizados nas dinâmicas de cada modelo. Apesar do software descrito não ter o foco na simulação de sistemas de produção, o módulo “gado bovino”, tem características neste sentido, fornecendo dados da dinâmica do rebanho para outros módulos que impactam em manejo e produção de pastagem, mão de obra, tamanho de rebanho, insumos agrícolas, alimento, condição ambiental da terra e emissão de gases do efeito estufa. Sendo assim, o trabalho se enquadra no critério de inclusão CI1.

T28 (PEREIRA, 2015): O trabalho visa avaliar produtividade animal, custos de produção, retorno financeiro, pegada de carbono, nitrogênio, água e energia de três sistemas de produção de bovinos de corte do Rio Grande do Sul, com o uso de simulações no software *Integrated Farm System Model* (IFSM). O trabalho se enquadra no critério de inclusão CI1.

T29 (MOREIRA, 2015): O objetivo do estudo foi o de estimar parâmetros genéticos para características produtivas, reprodutivas e de qualidade de carcaça avaliadas no programa de seleção Nelore Brasil e também estimar valores econômicos para características de importância econômica (tendem a maximizar o lucro e reduzir custos) em sistemas de bovinos de corte de ciclo completo. As simulações feitas para melhoramento genético utilizaram o modelo animal do software WOMBAT e os valores produtivo e econômico foram obtidos através de planilhas do *Microsoft Excel*, onde era simulado o aumento no valor das características de seleção objetivadas e assim recalculado os índices selecionados. O trabalho não se enquadra no critério de inclusão CI1.

T30 (FORMIGONI, 2002): O foco do estudo é determinar valores econômicos para propriedades especializadas na fase de cria, para tanto foi proposto um modelo de análise econômica das propriedades produtoras com vista a estimar valores econômicos; e então valores econômicos foram estimados com o uso do modelo e seus resultados apresentados. A ideia é que a escolha dos reprodutores tenha foco na avaliação genética para melhoramento animal, levando em conta as DEPs (Diferença Esperada na Progenie) para esta seleção. O trabalho não se enquadra no critério de inclusão CI1.

T31 (FONTOURA JÚNIOR, 2008): Este trabalho realiza uma revisão de literatura para avaliar o uso de modelos matemáticos na pecuária de corte, desenvolve um modelo matemático dinâmico para descrição do processo reprodutivo e simula seu desempenho a partir do escore de condição corporal do parto à maturidade sexual em sistemas de produção de bovinos; por fim é feita a avaliação deste modelo. O modelo foi implementado como um módulo no software RURAL FAZPEC. O modelo pode servir para fins didáticos pois permite a visualização da dinâmica dos sistemas de cria e impacto das mudanças provocados pelas estratégias de manejo. Na revisão de literatura os autores citam vários trabalhos realizados em cima de modelos de simulação de sistemas pecuários. O trabalho atende ao critério CI1.

T32 (MEIRELLES, 2005): Utilizando a técnica de simulação Monte Carlo, o estudo simula o nascimento, crescimento, engorda, maturação sexual e reprodução de bovinos em sistemas extensivos; com isso os autores construíram dois softwares, o primeiro para análise de características de produção (ganho de peso) e o segundo para manejo (dinâmica de um sistema de produção de bovinos). A segunda etapa do trabalho aborda a utilização de Redes Neurais Artificiais na avaliação e predição de valores genéticos para ganho de peso. A primeira etapa do trabalho atende ao critério CI1.

T36 (BATISTA, 2009): Objetiva caracterizar a viabilidade econômica de sistemas de cria em diferentes idades de acasalamento para gado de corte com o auxílio do software NTIA/CM da EMBRAPA, que realiza cálculo matricial, e planilhas do *Microsoft Excel* que simulam a evolução do rebanho. O trabalho não atende ao critério CI1.

T37 (BONATTI, 2009): Apresenta um modelo de simulação computacional para sistemas de recria e terminação de bovinos, em pasto, utilizando a ferramenta Stella. O modelo desenvolvido passou por diversos testes para avaliar a sensibilidade das respostas biológicas às modificações de variáveis como: taxa de lotação, fornecimento de suplementação e crescimento potencial de forragem. A validação foi feita com a comparação de valores obtidos na literatura. O trabalho atende ao critério CI1.

T40 (SANTANA, 2012): Apresenta quatro sistemas de produção de bovinos, dois de ciclo completo e dois de cria, que foram simulados no software Embraprec para simulações de sistemas bioeconômicos. O Embraprec é apresentado como instrumento analítico de tomada de decisões gerenciais e tecnológicas para sistemas de produção de pecuária de corte que visa simular de forma complexa os processos biológicos e suas interações. Os autores destacam que o software é um potente instrumento de experimentação, composto por mais de trezentos e setenta variáveis; como saída é informado: custo de operação, investimentos necessários, impostos e encargos, fluxo de caixas, etc. É permitida também a análise do desempenho do sistema criatório e resultados técnicos correspondentes. O trabalho atende ao critério CI1.

T60 (COSTA; BARBOSA, 1997): Relata o uso de ferramentas computacionais de apoio à tomada de decisão no setor agropecuário. O trabalho comenta que as simulações de sistemas apresentaram grande crescimento a partir de meados dos anos 80 devido ao surgimento de softwares que facilitavam a modelagem e flexibilidade de aplicação. O trabalho exemplifica o uso destes softwares em um projeto desenvolvido pela EMBRAPA – CPPSE (Centro de Pesquisa de Pecuária do Sudeste) que visa a integrar o software de simulação ARENA com o *shell* de análise NEXPERT, criando assim um simulador que representa um sistema de produção de gado de corte para auxiliar na tomada de decisão da política de compra e descarte do pecuarista. Os autores relatam ainda que esse software é uma adaptação do modelo TAMU que também tem como objetivo simular um sistema de produção de bovinos de corte. Sendo assim, o trabalho atende ao critério CI1.

T61 (MASSRUHÁ et al., 2014): Trata-se de uma publicação organizada como livro que no capítulo 13 destaca o tema “Desenvolvimento de simuladores na agropecuária”. Este capítulo discute a definição e usos da simulação e modelagem matemática. É destacado ainda, a crescente aplicação de simuladores nas mais diversas áreas da agropecuária: desenvolvimento e produção de plantas e de animais, apoio a tomada de decisões gerenciais, dinâmica de pragas, doenças e contaminantes, avaliações de impacto ambiental, dinâmica da terra, manejo da água e avaliação de tecnologias em geral.

A publicação cita que grandes empresas da área têm se dedicado à criação de arcabouços e infraestrutura de simulação, a saber: OMS, APSIM, VLE e Record; e cita três softwares simuladores, a saber: DSSAT, Grazplan e Invernada, porém não especifica o conceito de uso dos mesmos; cita a integração de simuladores com softwares de informações geográficas, estatísticos (PaSim), etc.; cita também a construção de um jogo digital baseado na simulação de sistemas de produção de bovinos, executado pela Embrapa informática Agropecuária, mas não lança mais detalhes sobre este trabalho. Os autores destacam ainda que o desenvolvimento de simuladores para agropecuária é raro no Brasil, sendo o foco principal trabalhos que visam o aperfeiçoamento de simuladores desenvolvidos no exterior. É citado ainda outros arcabouços de simulação: MaCSim, JDEVS, VLE e SimPy.

O trabalho atende ao critério CI1 porquê especificou o propósito do jogo citado, mesmo sem descrever mais detalhes do projeto. Para os demais softwares citados não é especificado o foco da simulação.

T68 (VARELA; BUSSONI; DIEGUEZ, 2015): Este trabalho relata a crescente adoção de sistemas silvipastoris (coexistência entre componentes florestais, pastagens e animais) para a produção de gado de corte no Uruguai. O trabalho visa construir um simulador que rode na plataforma web, de apoio à tomada de decisão para este tipo de sistema, baseado na evolução animal, fornecimento de matéria prima e evolução florestal, que permita aos seus usuários simular os impactos nos resultados econômicos e produtivos. O trabalho mostra algumas telas e considerações feitas acerca das decisões de desenvolvimento, porém não presta maiores detalhes do software, seu nome, se já foi lançado ou onde será disponibilizado e até mesmo se foi desenvolvido completamente. O trabalho atende ao critério CI1.

T70 (COSTA et al., 2004): O trabalho apresenta o software Gerenpec®, cujo propósito é projetar a evolução de uma fazenda pecuária no tempo de dez anos, consolidando números anuais referentes aos bovinos (estoque, variação de inventário, compras e vendas, perdas, taxa de abate, etc.), às pastagens (capacidade, taxa de lotação, etc.) e ao sistema de produção como um todo (receita total, gastos, margem bruta e outros indicadores econômicos). O software atende ao critério CI1.

APÊNDICE C – Requisitos Funcionais

Compilação dos requisitos funcionais atendidos pelos Módulos Desenhista e Simulador. Os requisitos foram extraídos dos Anexos [A](#), [B](#), [C](#) e [D](#).

1. O sistema deve permitir cadastro, alteração e exclusão de *modelos*, contendo os seguintes atributos: nome, descrição e versão. O sistema deve prover forma de visualizar quais sistemas utilizam cada modelo. O sistema deve prover forma de visualizar nós, fluxos e itens de infraestrutura vinculados a um modelo;
2. O sistema deve permitir cadastro, alteração e exclusão de *categorias de parâmetros*, contendo o seguinte atributo: nome;
3. O sistema deve permitir cadastro, alteração e exclusão de *parâmetros*, contendo os seguintes atributos: nome, código, categoria de parâmetro, unidade de medida e dica. O atributo código não deve ser editável, seu valor deve ser gerado automaticamente a partir do atributo nome, seguindo o padrão: converter caracteres em minúsculo para maiúsculo, substituir espaços por “_” e remover caracteres especiais diferentes de “_” e “#”;
4. O sistema deve permitir cadastro, alteração e exclusão de *fórmulas*, contendo os seguintes atributos: fórmula e tipo. Os tipos possíveis são: Fluxo (coeficiente), Nó (duração), Infraestrutura (Valor Inicial), Infraestrutura (Valor Residual) ou Infraestrutura (Vida Útil). A fórmula permite apenas códigos de parâmetros, operadores matemáticos e números. O sistema deve prover forma de visualizar onde cada fórmula está sendo utilizada;
5. O sistema deve permitir cadastro, alteração e exclusão de *etapas*, contendo os seguintes atributos: nome e hierarquia. Para nome é permitido apenas o uso de caracteres alfanuméricos em maiúsculo e do caractere especial “_”. Hierarquia indica abaixo de qual etapa ela ficará;
6. O sistema deve permitir cadastro, alteração e exclusão de *nós*, contendo os seguintes atributos: modelo, nome, etapas e fórmula de duração;
7. O sistema deve permitir cadastro, alteração e exclusão de *categorias de recursos*, contendo os seguintes atributos: nome e hierarquia. Para nome é permitido apenas o uso de caracteres alfanuméricos em maiúsculo e do caractere especial “_”. Hierarquia indica abaixo de qual categoria de recurso ela ficará;

8. O sistema deve permitir cadastro, alteração e exclusão de *recursos*, contendo os seguintes atributos: nome, unidade de medida e categoria de recurso;
9. O sistema deve permitir cadastro, alteração e exclusão de *categorias de infraestrutura*, contendo os seguintes atributos: nome e hierarquia. Para nome é permitido apenas o uso de caracteres alfanuméricos em maiúsculo e do caractere especial “_”. Hierarquia indica abaixo de qual categoria de recurso ela ficará;
10. O sistema deve permitir cadastro, alteração e exclusão de *itens de infraestrutura*, contendo os seguintes atributos: modelo, nome, unidade de medida, categoria de infraestrutura, fórmula de valor inicial, fórmula de valor residual e fórmula de vida útil;
11. O sistema deve permitir cadastro, alteração e exclusão de *fluxos*, contendo os seguintes atributos: modelo, nó de entrada, nó de saída, categoria de fluxo, recurso e fórmula de coeficiente. As categorias de fluxos possíveis são: tratamento e produção;
12. O sistema deve permitir cópia de *modelo* e seus itens relacionados: nós, fluxos e infraestrutura;
13. O sistema deve permitir que ao vincular um *modelo* a um *sistema de produção* todos os parâmetros (utilizados em fórmulas de nós, fluxo e infraestrutura) sejam associados ao sistema.
14. O sistema deve permitir edição de *parâmetros vinculados a sistemas de produção*, contendo os seguintes atributos: valor mínimo, valor máximo, valor padrão, precisão, *flag* de fixo, *flag* de interno e ordem de exibição;
15. O sistema deve permitir cópia de *sistema de produção* e seus itens relacionados: bioma, fase, processo, programa, tecnologias, parâmetros vinculados e modelo;
16. O sistema deve permitir *simulação computacional* de um sistema de produção com base no modelo a ele vinculado, recebendo como entrada valores dos parâmetros para simulação e exibindo como saída lista de indicadores calculados;

APÊNDICE D – Dicionário de Dados do Modelo Computacional

Este apêndice apresenta o dicionário de dados para o modelo computacional. Todas as entidades exibidas no DER da Figura 12 serão aqui explicitadas. Os atributos "*_user*", "*_create*", "*_update*" são necessários ao uso do Titan *Framework* e servem para armazenar: usuário que realizou ação, data de criação e data de atualização dos dados respectivamente. Como estes atributos são comuns a quase todas as entidades (com exceção daquelas em relacionamento N-N), estes atributos não serão explicitados na descrição de cada entidade.

Tabela 12 – Dicionário de Dados - Entidade Sistema

Entidade: precoce.system			
Representa os sistemas de produção de gado de corte			
Atributo	Tipo	Restrições	Descrição
id	serial	Chave Primária	Identificador da tupla
name	character varying(256)	-	Nome do sistema de produção
size	character(7)	-	Tamanho da propriedade em que o sistema se baseia
protocol	text	-	Protocolo para a implementação do sistema de produção. Irá orientar o usuário final da plataforma sobre como empregar o modelo em sua propriedade
featured	bit(1)	-	
sex	sexenum	O tipo sexenum é um enumerador que permite os valores: female, cast-male e whole-male	Indica a sexagem predominante no rebanho do sistema de produção
model	integer	Chave estrangeira para a entidade simulator.model	Especifica qual será o modelo computacional responsável por implementar o sistema de produção

Tabela 13 – Dicionário de Dados - Entidade Sistema-Parâmetro

Entidade: precoce.system_parameter			
Resolução da associação entre as entidades precoce.system e precoce.parameter. Apesar do uso dos parâmetros serem em entidades pertencentes ao Modelo, seu relacionamento é com Sistema pois para cada um dos sistemas de produção os valores de mínimo, máximo e padrão podem sofrer alterações			
Atributo	Tipo	Restrições	Descrição
id	serial	Chave Primária	Identificador da tupla
minimum	numeric(16,2)	-	Indica o menor valor permitido para o parâmetro
maximum	numeric(16,2)	-	Indica o maior valor permitido para o parâmetro
value	numeric(16,2)	-	Valor padrão para o parâmetro (informado pelo desenhista)
step	numeric(16,2)	-	Incremento permitido para o valor do parâmetro
fixed	bit(1)	-	Sendo fixo o valor não poderá ser alterado pelo usuário final daquele definido como valor padrão
internal	bit(1)	-	Define que o parâmetro não deve ser exibido para o usuário final
system	integer	Chave estrangeira para a entidade precoce.system. Os valores de system e parameter combinados devem ser únicos.	Sistema relacionado
parameter	integer	Chave estrangeira para a entidade precoce.parameter. Os valores de system e parameter combinados devem ser únicos.	Parâmetro relacionado

Tabela 14 – Dicionário de Dados - Entidade Parâmetro

Entidade: precoce.parameter			
São variáveis que irão carregar valores dentro dos modelos computacionais para cada sistema. Esses valores, dependendo do caso, poderão ser informados pelos desenhistas ou usuário final			
Atributo	Tipo	Restrições	Descrição
id	serial	Chave Primária	Identificador da tupla
name	character varying(64)	-	Nome do parâmetro
code	character varying	Valor deve ser único	Código alfanumérico que irá identificar os parâmetros dentro das fórmulas
unit	character varying(32)	-	Unidade de medida do parâmetro
tip	character varying	-	Dica de tela para o parâmetro
category	integer	Chave estrangeira para a entidade precoce.category	Categoria relacionada

Tabela 15 – Dicionário de Dados - Entidade Categoria

Entidade: precoce.category			
Especifica a categoria à qual um parâmetro pertence			
Atributo	Tipo	Restrições	Descrição
id	serial	Chave Primária	Identificador da tupla
name	character varying(32)	Valor deve ser único	Nome da categoria de parâmetro
_order	smallint	-	Ordem de exibição

Tabela 16 – Dicionário de Dados - Entidade Simulação

Entidade: precoce.simulation			
Simulações realizadas dentro da plataforma +Precoce			
Atributo	Tipo	Restrições	Descrição
id	serial	Chave Primária	Identificador da tupla
name	character varying(256)	-	Nome da simulação
active	bit(1)	-	Indica se a simulação está ativa ou não, i.e., se é possível executá-la
featured	bit(1)	-	Indica se deve ser exibida como destaque
system	integer	Chave estrangeira para a entidade precoce.system	Sistema relacionado

Tabela 17 – Dicionário de Dados - Entidade Simulação-Parâmetro

Entidade: precoce.simulation_parameter			
Indica os valores de parâmetros a serem utilizados para simular um sistema de produção			
Atributo	Tipo	Restrições	Descrição
simulation	integer	Chave estrangeira para a entidade precoce.simulation	Sistema relacionado
parameter	character varying	Chave estrangeira para a entidade precoce.parameter (campo code)	Parâmetro relacionado
value	numeric(16,2)	-	Valor do parâmetro para determinada simulação

Tabela 18 – Dicionário de Dados - Entidade Modelo

Entidade: simulator.model			
Entidade base para representação de um modelo computacional que será utilizado pelos sistemas para execução da simulação			
Atributo	Tipo	Restrições	Descrição
id	serial	Chave Primária	Identificador da tupla
name	character varying(256)	Os valores de name e version combinados devem ser únicos	Nome do modelo
description	text	-	Descrição do modelo
version	smallint	Os valores de name e version combinados devem ser únicos. O valor deve ser maior que zero	Versão do modelo

Tabela 19 – Dicionário de Dados - Entidade Infraestrutura

Entidade: simulator.infraestructure			
Define os itens de infraestrutura que são utilizados dentro de um sistema de produção			
Atributo	Tipo	Restrições	Descrição
id	serial	Chave Primária	Identificador da tupla
name	character varying(256)	Os valores name e model combinados devem ser únicos	Nome do item de infraestrutura
unit	character varying(256)	-	Unidade de medida do item de infraestrutura
infraestructure_category	integer	Chave estrangeira para a entidade simulator.infraestructure_category	Categoria de infraestrutura relacionada
model	integer	Os valores de name e model combinados devem ser únicos. Chave estrangeira para a entidade simulator.model	Modelo relacionado
formula_initial_value	integer	Chave estrangeira para a entidade simulator.formula	Fórmula relacionada para cálculo do valor inicial do item de infraestrutura
formula_residual_value	integer	Chave estrangeira para a entidade simulator.formula	Fórmula relacionada para cálculo do valor residual do item de infraestrutura
formula_lifespan	integer	Chave estrangeira para a entidade simulator.formula	Fórmula relacionada para cálculo da vida útil do item de infraestrutura

Tabela 20 – Dicionário de Dados - Entidade Infraestrutura-Categoria

Entidade: simulator.infraestructure_category			
Especifica a categoria à qual um item de infraestrutura pertence			
Atributo	Tipo	Restrições	Descrição
id	serial	Chave Primária	Identificador da tupla
name	character varying(256)	Valor deve ser único	Nome da categoria de infraestrutura
father	integer	Chave estrangeira para a entidade simulator.infraestructure_category	Categoria de infraestrutura pai. Serve para montar a hierarquia de exibição

Tabela 21 – Dicionário de Dados - Entidade Nó

Entidade: simulator.node			
Representa os nós/estações de um modelo computacional. Estações são os locais onde os recursos que transitam pelo sistema de produção são utilizados e transformados em uma saída (para outro nó/estação ou como saída do sistema)			
Atributo	Tipo	Restrições	Descrição
id	serial	Chave Primária	Identificador da tupla
name	character varying(256)	Os valores de name e model combinados devem ser únicos	Nome do nó
duration_formula	integer	Chave estrangeira para a entidade simulator.formula	Fórmula relacionada para cálculo em dias da duração do nó
model	integer	Chave estrangeira para a entidade simulator.model	Modelo relacionado

Tabela 22 – Dicionário de Dados - Entidade Etapa

Entidade: simulator.stage			
Alguns indicadores dependem dos resultados calculados em agrupamentos de nós/estações, a esse conjunto é dado o nome de Etapa, que pode ser composta de um ou mais nós/estações. Por exemplo: recria, confinamento, engorda, vendas, compras, etc.			
Atributo	Tipo	Restrições	Descrição
id	serial	Chave Primária	Identificador da tupla
name	character varying(256)	Valor deve ser único	Nome da Etapa
father	integer	Chave estrangeira para a entidade simulator.node	Etapa pai. Serve para montar a hierarquia de exibição/cálculo

Tabela 23 – Dicionário de Dados - Entidade Nó-Etapa

Entidade: simulator.node_stage			
Resolução da associação entre etapa e nó. Um nó pode pertencer a várias etapas e uma etapa pode ter vários nós			
Atributo	Tipo	Restrições	Descrição
node	integer	Chave primária	Nó relacionado
stage	integer	Chave primária	Etapa relacionada

Tabela 24 – Dicionário de Dados - Entidade Fluxo

Entidade: simulator.flow			
Representa a movimentação de recursos entre os nós de um modelo computacional			
Atributo	Tipo	Restrições	Descrição
id	serial	Chave Primária	Identificador da tupla
model	integer	Chave estrangeira para a entidade simulator.model	Modelo relacionado
flow_category	integer	Chave estrangeira para a entidade simulator.flow_category. Os valores de coefficient_formula, node_in, node_out e resource combinados devem ser únicos	Categoria de Fluxo relacionada
coefficient_formula	integer	Chave estrangeira para a entidade simulator.formula	Fórmula relacionada para cálculo do coeficiente, isto é, valor utilizado para cálculo da quantidade do recurso referido pelo fluxo
node_in	integer	Chave estrangeira para a entidade simulator.node. Os valores de coefficient_formula, node_in, node_out e resource combinados devem ser únicos. Os valores de node_in e node_out devem ser diferentes	Nó relacionado. Representa o nó de destino do fluxo ou de onde o fluxo "entra"
node_out	integer	Chave estrangeira para a entidade simulator.node. Os valores de coefficient_formula, node_in, node_out e resource combinados devem ser únicos. Os valores de node_in e node_out devem ser diferentes	Nó relacionado. Representa o nó de origem do fluxo ou de onde o fluxo "sai"
resource	integer	Chave estrangeira para a entidade simulator.resource. Os valores de coefficient_formula, node_in, node_out e resource combinados devem ser únicos	Recurso relacionado. É o recurso que está sendo repassado de um nó a outro

Tabela 25 – Dicionário de Dados - Entidade Fluxo-Categoria

Entidade: simulator.flow_category			
Especifica a categoria à qual um fluxo pertence			
Atributo	Tipo	Restrições	Descrição
id	serial	Chave Primária	Identificador da tupla
name	character varying(256)	O valor deve ser único	Nome da Categoria de Fluxo

Tabela 26 – Dicionário de Dados - Entidade Recurso

Entidade: simulator.resource			
Tudo que pode fluir de um nó a outro no modelo computacional			
Atributo	Tipo	Restrições	Descrição
id	serial	Chave Primária	Identificador da tupla
name	character varying(256)	O valor deve ser único	Nome do recurso
unit	character varying(256)	-	Unidade de medida do recurso
resource_category	integer	Chave estrangeira para a entidade simulator.resource_category	Categoria de recurso relacionada

Tabela 27 – Dicionário de Dados - Entidade Recurso-Categoria

Entidade: simulator.resource_category			
Especifica a categoria à qual um recurso pertence			
Atributo	Tipo	Restrições	Descrição
id	serial	Chave Primária	Identificador da tupla
name	character varying(256)	O valor deve ser único	Nome da categoria de recurso
father	integer	Chave estrangeira para a entidade simulator.resource_category	Categoria de recurso pai. Serve para montar a hierarquia de exibição/cálculo

Tabela 28 – Dicionário de Dados - Entidade Fórmula

Entidade: precoce.formula			
Fórmulas matemáticas simples, utilizadas para realizar algum cálculo durante a simulação			
Atributo	Tipo	Restrições	Descrição
id	serial	Chave Primária	Identificador da tupla
formula	character varying(256)	Os valores de formula e formula_category combinados devem ser únicos	Expressão algébrica, isto é, a equação é definida com o uso de variáveis, números e operadores
formula_category	integer	Chave estrangeira para a entidade simulator.formula_category. Os valores de formula e formula_category combinados devem ser único	Categoria de fórmula relacionada

Tabela 29 – Dicionário de Dados - Entidade Fórmula-Categoria

Entidade: simulator.formula_category			
Especifica a categoria à qual uma fórmula pertence			
Atributo	Tipo	Restrições	Descrição
id	serial	Chave Primária	Identificador da tupla
name	character varying(256)	O valor deve ser único	Nome da categoria de fórmula
father	integer	Chave estrangeira para a entidade simulator.formula_category	Categoria de fórmula pai. Serve para montar a hierarquia de exibição

Tabela 30 – Dicionário de Dados - Entidade Fórmula-Parâmetro

Entidade: simulator.formula_parameter			
Resolução da associação entre fórmula e parâmetros. Parâmetros são as variáveis da expressão algébrica que terá seus valores atribuídos pelo usuário (ou pelo desenhista, quando interno) ao executar uma simulação (ver entidade Simulação Parâmetro, Tabela 17)			
Atributo	Tipo	Restrições	Descrição
formula	integer	Chave estrangeira para a entidade simulator.formula. Os valores de formula e parameter combinados devem ser únicos	Fórmula relacionada
parameter	integer	Chave estrangeira para a entidade precoce.parameter. Os valores de formula e parameter combinados devem ser únicos	Parâmetro relacionado

APÊNDICE E – Integração do Módulo Simulador à Plataforma +Precoce

A integração seguiu algumas etapas. Primeiramente foi necessário adicionar um *endpoint* ao serviço *RESTful* disponibilizado pelo *Framework* Titan. A esse propósito, foi adicionado ao Módulo Administrador, junto ao componente *simulation* o arquivo `run.php` para responder por esse serviço.

Para adicionar a biblioteca de simulação ao Módulo Administrador o arquivo que faz a gestão de dependências do *Composer* foi alterado. Uma preocupação é que a biblioteca não terá uma licença pública, sendo assim, o repositório onde se encontra o código da biblioteca tem acesso restrito a membros da Embrapa. Para adicionar um repositório particular às buscas do *Composer*, basta adicionar a diretriz ao arquivo `composer.json`:

```
"repositories": [{
    "type": "vcs",
    "url": "https://git.cnpge.embrapa.br/mais-precoce/simulator"
}]
```

Configurada a localização do repositório, basta adicionar chamada a biblioteca de simulação (`"maisprecoce/simulator": "dev-master"`), junto à lista de pacotes requeridos.

Ao *endpoint* `run.php`, o seguinte trecho de código foi adicionado:

```
1 try {
2     $simulation = new \Simulator\ComputationalModel
3                 \ComputationalModel(Database::singleton(), $id);
4     $simulation->calculateGraph();
5     $simulation->calculateStock();
6     $indicators = $simulation->calculateIndicators();
7     $pattern = \Simulator\Utils\UsefulHelp::getReturnPattern($indicators);
8 } catch (\Throwable $t) {
9     Log::singleton()->add("Erro ao executar simulacao - ". $t);
10    http_response_code(500);
11    echo "Erro ao realizar simulacao!";
12 }
13
14 http_response_code(200);
15 echo json_encode($pattern, JSON_UNESCAPED_UNICODE);
```


Na linha 2 é instanciado um objeto da classe *ComputationalModel*, recebendo como parâmetros a conexão com a base utilizada e o identificador da simulação (ID de entrada para *precoce.simulation*). As linhas 4 e 5 realizam o cálculo do grafo e estoque e na linha 6 os indicadores são calculados. Na linha 7 a função estática *UsefulHelp::getReturnPattern()* transforma a lista de indicadores calculados em uma lista para conversão em JSON. Caso ocorra algum erro as linhas de 8 a 12 fazem seu tratamento, adicionando *log* que poderá ser consultado pelo Módulo Administrador e retornando ao chamador pacote HTTP com o status de Erro Interno no Servidor (500). Caso não ocorra erro as linhas 14 e 15 são executadas e retorna status de sucesso juntamente com a lista JSON a ser exibida no Módulo Portal.

Anexos

ANEXO A – Conceitos e Requisitos Funcionais - Módulos 0 e 1

A.1 Objetivo

Este documento descreve os principais conceitos e requisitos funcionais da Plataforma +Precoce com o objetivo de validar o entendimento com a equipe do projeto e dar subsídios para levantamento de requisitos e desenho da Plataforma pela equipe de engenharia de software do projeto.

Este documento limita-se aos conceitos e requisitos funcionais necessários para o desenvolvimento dos módulos 0 e 1.

Recomenda-se a leitura de outros documentos, especialmente a especificação da arquitetura da Plataforma +Precoce.

A.2 Introdução

A Plataforma +Precoce será uma aplicação na web que permitirá a um Usuário Cliente simular os resultados de Sistemas de produção de gado de corte cujos Modelos matemáticos estiverem disponíveis. Estes Modelos terão sido previamente construídos a partir de conhecimento acumulado na Embrapa ou em publicações, bem como experimentos previamente realizados pela Embrapa e parceiros.

O Usuário Cliente terá a possibilidade de ajustar Parâmetros que modificam o desempenho de Sistemas cadastrados na Plataforma. Se estes ajustes estiverem dentro de limite de valores recomendados previamente estabelecidos, a Plataforma indicará ao Usuário Cliente que os resultados da Simulação são considerados prováveis pela Embrapa, desde que o Sistema modelado seja implementado como descrito em um Protocolo que define o Sistema.

A Plataforma poderá evoluir no futuro de modo a atender também outros tipos de sistemas de produção (de leite, pecuária de outros animais, ou mesmo lavoura), mas o escopo do projeto +Precoce limita-se à produção de gado bovino de corte, e decisões de arquitetura da Plataforma foram tomadas considerando este escopo.

A.3 Principais conceitos

Este tópico descreve os principais conceitos que precisam ser representados na Plataforma +Precoce para que os requisitos funcionais sejam atendidos. Os conceitos que iniciam com letra maiúscula (Sistema, Usuário, Cliente, Parâmetro, etc.) correspondem aproximadamente ao que em engenharia de software costuma ser denominado como entidade, objeto, ou classe.

A.3.1 Sistema

Sistema de produção de gado de corte. Cada Sistema possui um nome único e uma descrição única, isto é, que não se repetem em outro Sistema. Cada Sistema está associado a Filtros:

- uma ou mais Fases,
- uma ou mais Tecnologias,
- zero, um ou mais Programas de bonificação,
- um e somente um Tamanho.

A.3.2 Fases

Representam as etapas do ciclo de vida do animal em um Sistema de produção de gado bovino de corte. São elas:

- Cria (“C”)
- Recria (“R”)
- Engorda (“E”)

Cada Sistema pode estar associado a qualquer combinação de Fases, mas, ao menos inicialmente, os Sistemas que estarão disponíveis na Plataforma estarão associados apenas às seguintes combinações:

- Cria (“C”)
- Recria e Engorda (“RE”)
- Cria, Recria e Engorda (“CRE”), também chamado “Ciclo completo”.

A.3.3 Tecnologias

Tecnologias empregadas nos Sistema (p.ex., Confinamento, Desmama precoce, IATF).

A.3.4 Programas

Programas de bonificação que podem gerar receita adicional, a depender dos Processos associados ao Sistema (p.ex., “Carrefour® Novilho Leve”). Sistemas associados a um Programa incluem, em seu Modelo matemático, o efeito destes Programas nas Fórmulas usadas para calcular determinados Indicadores econômico-financeiros.

A.3.5 Tamanho

Define os valores mínimo, máximo e default para o Parâmetro “Área de produção” do Sistema. Cada Sistema está associado a um e somente um Tamanho. São eles:

- Pequeno (“P”): de 100 a 1000 ha de Área de produção (default: 500 ha)
- Médio (“M”): de 1001 ha a 5000 ha de Área de produção (default: 3000 ha)
- Grande (“G”): de 5001 ha ou mais de Área de produção (default: 20.000 ha)

A Área de produção inclui toda Área de qualquer tipo em uso pelo Sistema e calculada para o Modelo matemático do Sistema. Em princípio, os únicos tipos de Área considerados na Plataforma serão as áreas de pastagem (nativa ou formada) e de confinamento. Áreas de lavoura, produção de feno e silagem não serão modeladas na Plataforma, e o produto destas áreas deve ser modelado como Recursos que entram no Sistema, vindo de fora deste.

Os valores mínimo, máximo e *default* para o Parâmetro “Área de produção” são os mesmos para todos os Sistemas de um mesmo Tamanho, independentemente de Tecnologias, e Programas. É responsabilidade do Usuário Administrador da Plataforma manter esta restrição para todos os Sistemas ¹.

A.3.6 Filtros

O Usuário Cliente localiza o Sistema que deseja simular através da seleção de uma ou mais opções em Filtros para as Fases, Tecnologias, Programas e Tamanho associados ao Sistema.

¹ Permitir valores mínimos, default e máximos de Área de Pastagem diferentes a depender de Fases ou Tecnologias não permitiria ao Usuário Cliente saber o que esperar de cada uma das opções do Filtro “Tamanho”. O modo como a Plataforma foi definida funcionalmente permite relaxar esta restrição, e cabe ao Usuário Administrador a disciplina de mantê-la.

O modo de funcionamento dos Filtros vai depender dos recursos disponíveis no software a ser utilizado no desenvolvimento da Plataforma, mas deve levar em conta a quantidade de Sistemas cadastrados, de opções em cada Filtro e das prováveis quantidades de opções associadas a cada Sistema, que se estima sejam:

- Fases: 3 (de uma a três por Sistema)
- Tecnologias: de 20 a 50 (de um a 10 por Sistema)
- Programas: 30 a 50 (de um a 50 por Sistema)
- Tamanhos: 3 (apenas um por Sistema)
- Sistemas: 50 a 100

Na navegação do Usuário Cliente, os Filtros serão todos “multivalorados”, i.e., permitirão a seleção de mais de uma opção. Os filtros “Fase” e “Tamanho” possuem uma quantidade pequena e fixa de opções (três). Os demais Filtros podem possuir dezenas de valores, e por isso precisam usar componentes que levem isto em consideração (busca textual da opção automática no preenchimento, p.ex.). Na seleção, os Filtros atuam como “E” da opções selecionadas em cada Filtro, e “E” de todos os Filtros.

A.3.7 Protocolo

Documento ² que descreve o Sistema e orienta a aplicação das Tecnologias associadas ao Sistema, considerando o seu Tamanho, Fases e Programas de bonificação. Cada Sistema tem um e somente um Protocolo.

Além das recomendações sobre como implantar e gerenciar o Sistema, o Protocolo pode conter informações adicionais, por exemplo, de interesse acadêmico, como a descrição dos experimentos que avalizam o desempenho estimado para o Sistema, e seus resultados.

Ao cadastrar um Sistema, o Usuário Administrador deve se certificar de que as opções de Filtro associadas ao Sistema, o seu Modelo, os valores mínimo, máximo e *default* de seus Parâmetros e os seus Indicadores estejam coerentes com o que foi declarado no Protocolo.

A.3.8 Modelo

Modelo matemático que usa o valor atribuído a seus Parâmetros para calcular os Recursos necessários para a máxima produção que o Sistema consegue produzir para a Área de produção informada pelo Usuário Cliente para o Sistema, em Safras anuais

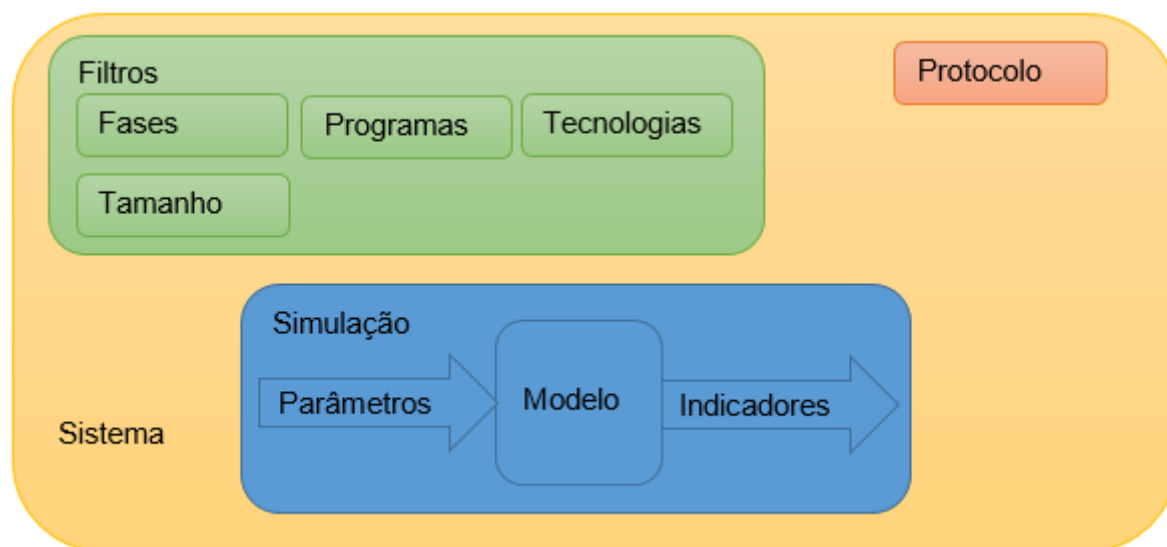
² No momento, a decisão técnica é implementar o protocolo como um objeto “texto rico”, isto é, texto HTML com formatação, mas é possível se usar outros formatos, p.ex., PDF.

iguais. Este processo de cálculo é denominado Simulação. Como resultado do cálculo da Simulação, o valor dos Indicadores é estabelecido.

Cada Sistema disponível para Simulação está associado a um e somente um Modelo disponível para Simulação. Opcionalmente, mais Modelos podem ser associados ao mesmo Sistema, para efeito de histórico de versões, mas apenas um destes Modelos estará disponível e será considerado para Simulação.

Dois ou mais Sistemas podem compartilhar o mesmo Modelo, mesmo que estes Sistemas estejam associados a Protocolos, Fases, Programas, Tamanhos e Tecnologias diferentes ³. Isto pode ocorrer quando a diferença entre os Sistemas estiver apenas nos valores mínimo, máximo e *default* de Parâmetros. Nestes casos, o Usuário Administrador da Plataforma poderá cadastrar um novo Sistema a partir de cópia de outro Sistema, mantendo o Modelo adotado neste.

Cabe ao Usuário Administrador da Plataforma cuidar para que a associação entre o Sistema, seus valores mínimo, máximo e *default* de Parâmetros e seu Modelo, seja coerente com os respectivos Protocolos, inclusive quando houver necessidade de ajustar ou substituir um Modelo compartilhado por vários Sistemas.



A.3.9 Parâmetro

Valor que será usado no cálculo da Simulação do Modelo associado ao Sistema.

Cada Parâmetro possui um nome único e uma descrição única, isto é, que não se repetem em outro Parâmetro.

Cada Parâmetro possui um valor mínimo, máximo e *default* recomendados definidos pelo Usuário Administrador para cada Sistema em que o Parâmetro se aplica.

³ Ou seja, todos os Filtros, mas é bem pouco provável que um mesmo Modelo sirva para Sistemas com Fases e Tamanhos distintos, especialmente.

Os valores mínimo e máximo representam o intervalo de valores recomendados para o Parâmetro em Simulações do Sistema. O valor *default* do Parâmetro é o valor mais recomendado, isto é, aquele que maximiza a confiança nos valores calculados pela Simulação para os Indicadores do Sistema.

Alguns Parâmetros podem ser definidos com a restrição adicional de permitirem apenas números inteiros, para todos os Modelos e Sistemas. Dentre estes, alguns podem ser representados por uma lista de opções em formato texto, cada opção associada a um número inteiro. Também nestes casos os Parâmetros possuem valores mínimo, máximo e *default*, seguindo a ordem dos números inteiros associados às respectivas opções.

Cada Parâmetro está associado a uma Unidade de medida comum a todos os Modelos e Sistemas que define o formato de apresentação de seu conteúdo (p.ex.,

Alguns Parâmetros podem ser assinalados como Livres para o Sistema, isto é, permitem ajustes para fora dos limites máximo e mínimo recomendados.

Além dos limites mínimo e máximo recomendados, os Parâmetros possuem limites mínimo e máximo permitidos, também mantidos pelo Administrador para cada Sistema⁴, para evitar problemas ou inconsistências nos cálculos dos Modelos matemáticos (por exemplo, divisão por zero).

Pode haver casos em que o valor do Parâmetro é definido pelo Administrador como Fixo para o Sistema. Neste caso, o Usuário não conseguirá alterar o seu valor para fora do valor *default*.

Alguns Parâmetros podem ser assinalados como Internos para o Sistema, isto é, não são exibidos para o Usuário Cliente, naquele Sistema, mas apenas para o Usuário Administrador. Este é provavelmente o caso da maioria dos Parâmetros de valor Fixo, mas um Parâmetro Fixo não é necessariamente Interno, nem vice-versa.

A maioria dos Modelos usa a maioria dos Parâmetros, mas não necessariamente todos os Parâmetros são usados em todos os Modelos.

A quantidade de Parâmetros será muito grande, e por isso é preciso uma forma amigável do Usuário Cliente navegar pelos Parâmetros. Além dos Parâmetros poderem ser declarados como Internos para um Sistema (e portanto invisíveis ao Usuário Cliente), dois outros atributos dos Parâmetros para um Sistema podem ser usados para melhorar a usabilidade destes.

Os Parâmetros pertencem a um Tipo no Sistema, Tipo que serve exclusivamente para agrupá-los nas telas que o Usuário Cliente usará para ajustar os valores destes Parâmetros para cálculo de Simulação. Os Tipos de Parâmetros são compartilhados por toda a Plataforma, mas o Tipo de um Parâmetro pode mudar de um Sistema para outro.

⁴ Para a maioria dos Parâmetros, os valores mínimo e máximo permitidos serão iguais para todos os Sistemas, mas ainda assim a Plataforma permitirá ajustá-los separadamente para cada Sistema.

Outro atributo de cada Parâmetro em cada Sistema define se este está entre os Principais, isto é, aqueles que a maioria dos Usuários Clientes gostaria de poder ajustar, ou Outros, isto é, aqueles que o Usuário Cliente verá apenas se clicar em “ver mais...”.

Os Parâmetros podem estar associados a uma Hierarquia, que os organiza em árvore, para a montagem de relatórios.

A seguir apresentamos uma lista possível de Tipo de Parâmetros. Exceto pelo primeiro Tipo, todos os demais são apenas sugestões e podem ser ajustados pelo Usuário Administrador, após a implantação da Plataforma:

A.3.9.1 Área de produção

Um único Parâmetro, Principal (isto, é, sempre exibido) e essencial para o cálculo de todas as Simulações, e por isso é exceção: É comum a todos os Sistemas. O Usuário Administrador deve garantir que este Parâmetro esteja disponível para todos os Sistemas, com valores mínimo, máximo e *default* acompanhando a descrição dada pelo Tamanho do Sistema.

A.3.9.2 Parâmetros de protocolo

Definidos pelas recomendações técnicas disponíveis no Protocolo do Sistema (p.ex.: “Idade à desmama”, “Quantidade de suplemento por UA”).

Parâmetros de protocolo são usados para definir Quantidades de Recursos (itens de custo variável) e Infraestrutura (itens de custo fixo) necessários para a operação do Sistema como definido pelo seu Protocolo.

A.3.9.3 Parâmetros de desempenho

Representam determinadas expectativas de desempenho zootécnico do Sistema, se o Protocolo que define o Sistema for aplicado corretamente (p.ex.: “Taxa de natalidade”, “Ganho de peso médio diário”, “Taxa de mortalidade”).

No caso dos Parâmetros de desempenho, os valores mínimo e máximo recomendados representam o intervalo de valores que a Embrapa recomenda para que os Indicadores sejam estimados pelo Modelo dentro de limite de confiança estabelecido no Protocolo.

A.3.9.4 Parâmetros de mercado

Variável em geral externa ao Sistema que precisa ser considerada pelo Modelo no cálculo de seus Indicadores econômico-financeiros. A maioria dos Parâmetros de mercado é necessária para calcular o Preço dos Recursos e itens de Infraestrutura definidos por Parâmetros de protocolo (p.ex.: “Preço da arroba”, “Preço do bezerro”, “Salário” de determinado “Perfil profissional”). Outros casos são alíquotas de impostos.

A Plataforma não considerará ajustes nos Parâmetros de mercado para efeito de emissão de mensagem de alerta de Simulação com Parâmetro Livre.

A.3.9.5 Parâmetros de qualidade (da carcaça)

Atributos relacionados à qualidade da carcaça produzida. Estes Parâmetros são considerados no cálculo de Indicadores econômico-financeiros, de modo a remunerar de forma diferenciada a carcaça que atenda a critérios de Programas de bonificação. Alguns exemplos:

- Maturidade: Escala baseada na presença de dentes incisivos inferiores.
- Acabamento: Medida visual de cobertura de gordura subcutânea da carcaça que varia do grau 1 ao grau 5.
- Rendimento de carcaça: Percentual do peso do produto por animal depois da retirada de couro, vísceras, cabeça, patas, rabo e aparelho reprodutor do animal abatido. Medido em kg ou @.
- Classificação: Classificação do bovino macho obtida a partir da composição do conceito de maturidade e do status sexual (inteiro e castrado), podendo ser J0, J2, J4, T0, T2, T4, intermediário (I, 6 dentes) e adulto (A, acima de 6 dentes).

A.3.10 Indicador

Nome dado pela Plataforma para o valor de uma dada variável calculada pela Simulação. Os Indicadores são calculados na Simulação a partir dos valores atribuídos aos Parâmetros quando a Simulação foi solicitada.

Cada Indicador possui um nome único e uma descrição única, isto é, que não se repetem em outro Indicador.

Os Indicadores podem estar associados a uma Hierarquia, que os organiza em árvore, para a montagem de relatórios.

O Módulo de Relatórios exibir os valores de Indicadores e Parâmetros de uma Simulação.

Nem todos os Indicadores disponíveis para a Plataforma precisam obrigatoriamente ser usados em todos os Modelos.

A maioria dos Indicadores têm valores numéricos, mas alguns podem ser alfanuméricos, representando uma lista ordenada de opções, de forma semelhante ao descrito para Parâmetros.

Assim como os Parâmetros, os Indicadores também estão associados a uma Unidade de medida que define um formato de apresentação de seu conteúdo (p.ex, %, R\$).

A posição do Indicador na Hierarquia pode ser usada para definir a ordem e posição do Indicador nas telas de Relatório da Plataforma em que Indicadores são apresentados. Uma sugestão de Hierarquia de um só nível para os Indicadores é apresentada a seguir:

A.3.10.1 Físicos

Estimam o desempenho físico do Sistema simulado. Diferentemente dos Parâmetros de desempenho⁵, os Indicadores físicos são calculados pela Simulação (p.ex.: “Toneladas de carne produzida por ano”).

A.3.10.2 Econômico-financeiros

Estimam o desempenho econômico ou financeiro do Sistema simulado (p.ex.: “Receita bruta anual” e “Lucro líquido anual”).

A.3.10.3 Ambientais

Estimam o desempenho ambiental do Sistema simulado (p.ex.: “Emissão de GEE de CO₂eq em ton”).

A.3.11 Simulação

Uma Simulação é a execução do cálculo do Modelo do Sistema, produzindo os valores dos Indicadores a partir dos valores escolhidos para os Parâmetros do Sistema. Uma Simulação armazena as seguintes informações sobre si:

- O Sistema selecionado pelo Usuário Cliente para a Simulação;
- O Modelo usado na Simulação;
- Os valores dos Parâmetros do Sistema usados no cálculo;
- Os valores dos Indicadores obtidos do cálculo;
- O momento do último salvamento da Simulação;
- O Usuário que executou o último salvamento da Simulação;
- O momento do último (re)cálculo da Simulação;
- O Usuário que executou o último (re)cálculo da Simulação;

⁵ Embora sejam resultados de experimentos, os Parâmetros de desempenho são variáveis de entrada dos Modelos.

- Um nome único e uma descrição única, isto é, que não se repetem em outra Simulação.

A janela inicial do Usuário Cliente exibe as Simulações criadas por ele. O Usuário Cliente pode abrir uma destas Simulações, para consulta ou ajustes, ou pesquisar os Sistemas disponíveis, usando os Filtros já descritos.

O Usuário Administrador pode assinalar alguns Sistemas como "Preferidos". Estes Sistemas serão exibidos na tela inicial, independentemente da configuração dos Filtros.

Ao selecionar um Sistema para consulta e criação de Simulação, são exibidos os dados do Sistema e de sua Simulação *default*. A Simulação *default* é aquela (re)calculada a partir dos valores *default* dos seus Parâmetros, e que é sempre recalculada pelo Usuário Administrador do Sistema quando este disponibiliza ou atualiza o Sistema para consulta e Simulação na Plataforma.

Um Usuário Cliente pode criar novas Simulações do Sistema a partir de cópia da Simulação *default* do Sistema ou de outras Simulações criadas pelo mesmo Usuário Cliente, com ou sem possíveis ajustes de valores de Parâmetros.

Ao criar uma nova Simulação, o Usuário é perguntado se permite que os dados de sua Simulação sejam utilizados pela Embrapa em publicações (*default*: Sim), e se esta permissão inclui a identificação do Usuário que criou a Simulação (*default*: Não).

Ao criar uma nova Simulação, o Protocolo do Sistema é copiado para a Simulação em objeto que permite alterar o conteúdo desta cópia, mas excluir, alterar ou acrescentar imagens.

O número máximo de Simulações que pode ser criada por cada Usuário Cliente é definido pelo Usuário Administrador da Plataforma, como um atributo do Usuário Cliente.

O Usuário Cliente pode ajustar o valor de um ou mais Parâmetros das Simulações que ele criou, e pedir o seu recálculo. Os Parâmetros definidos como Livres podem ser ajustados para valores fora dos valores mínimo e máximo recomendados, mas dentro dos valores mínimo e máximo permitidos para o Sistema. Os valores dos demais Parâmetros não podem ser ajustados fora do intervalo dado por seus valores mínimo e máximo recomendados.

O Usuário Cliente pode tornar Livres um ou mais Parâmetros além daqueles definidos previamente pelo Usuário Administrador como Livres para o Sistema. Neste caso, o Usuário Cliente é avisado pela Plataforma de que os resultados da Simulação criada pelo Usuário Cliente não são mais garantidos nas condições especificadas no Protocolo do Sistema.

Depois de salva, uma Simulação pode ser recalculada a qualquer momento, mas apenas pelo Usuário Cliente que a criou ou o Usuário Administrador. O momento em que

a Simulação é recalculada é armazenado na Simulação⁶.

Uma Simulação pode ser salva sem ser recalculada, p.ex., para permitir que o Usuário não perca os ajuste intermediários que tiver feito em Parâmetros antes de recalculer a Simulação. A Plataforma pergunta ao Usuário Cliente se deseja recalculer a Simulação cada vez que o Usuário pedir para salvá-la.

Ao abrir a Simulação, a Plataforma alerta se o momento do último salvamento da Simulação é posterior ao momento do último recálculo, e pergunta ao Usuário Cliente se deseja recalculer a Simulação.

Também ao abrir uma Simulação, a Plataforma verifica se o Sistema sofreu qualquer alteração pelo Usuário Administrador depois do último salvamento da Simulação, e avisa ao Usuário Cliente que infelizmente será preciso refazer a Simulação a partir dos valores *default* do Sistema. Neste caso, o Usuário Cliente tem as opções de excluir a Simulação ou abri-la totalmente bloqueada. Este alerta será repetido em todas as situações que esta Simulação for usada, inclusive no Módulo de Relatórios⁷.

Os dados das Simulações *default* e criadas pelo Usuário Cliente ficam disponíveis para seu uso do Módulo de Relatórios, especialmente os valores dos Indicadores e Parâmetros.

Inicialmente, apenas um Relatório estará disponível para os Usuário Clientes. Este Relatório permitirá a comparação dos Indicadores e Parâmetros de duas ou mais Simulações salvas pelo Usuário Cliente ou *default*. O número máximo de Simulações que pode ser exibido neste Relatório é definido pelo Usuário Administrador do Sistema como um atributo de configuração geral da Plataforma.

As Simulações criadas pelo Usuário Cliente são automaticamente excluídas quando o Usuário Cliente que as criou é excluído da Plataforma pelo Usuário Administrador.

⁶ A decisão de persistir ou não os valores calculados para uma Simulação na base de dados é uma decisão técnica (não funcional), que levará em conta desempenho, necessidade de controles, etc.

⁷ Em princípio, apenas a substituição do Modelo e a alteração de algum Parâmetro exigiria isso, e mesmo em muitos dos casos bastaria recalculer a Simulação, com grande chance de se aproveitar boa parte do que o Usuário Cliente ajustou, mas o tratamento de cada caso complicaria a Plataforma e implicaria em assumir muitos pressupostos que podem confundir o Usuário Cliente.

ANEXO B – Conceitos e Requisitos Funcionais - Módulo 2

B.1 Objetivo

Este documento descreve os principais conceitos e requisitos funcionais do “Módulo 2” da Plataforma +Precoce, também chamado de “Simulador”, com o objetivo de validar o entendimento com a equipe do projeto e dar subsídios para desenho da Plataforma pela equipe de engenharia de software do projeto.

Recomenda-se leitura prévia do documento de Conceitos e Requisitos dos Módulos 0 e 1, especialmente quanto à definição de Conceitos compartilhados entre os Módulos 0, 1 e 2 (Sistema, Modelo, Recurso, Simulação, Parâmetro, Indicador).

B.2 Modelo Conceitual e Requisitos Funcionais

Os requisitos funcionais necessários que devem ser atendidos pela implementação do Módulo 2: Simulador são apresentados a seguir, ao mesmo tempo em que alguns Conceitos são definidos.

Os Conceitos são apresentados aqui iniciados por maiúscula. Cada Conceito equivale, aproximadamente, ao que em tecnologia da informação é denominado “classe”, “objeto”, “entidade” ou mesmo “atributo”.

Os Conceitos dependem uns dos outros para serem bem definidos, e algumas vezes a definição de um Conceito aparece no texto depois de ter sido mencionado pela primeira vez.

Os requisitos funcionais aparecem ao longo do texto, na forma de afirmações sobre os Conceitos e o relacionamento entre eles.

Exceto quando explicitamente declarado, todos os Conceitos e requisitos funcionais descritos neste documento precisam ser considerados na implementação da Plataforma em software.

Alguns dos requisitos que devem ser atendidos por um Modelo matemático bem elaborado não serão implementados como regras de construção do modelo impostas pela Plataforma, mas devem ser garantidos pelo Usuário Desenhista do Modelo via “boas práticas” de modelagem usando a Plataforma. Estas boas práticas são descritas em outro documento, o “Manual do Desenhista”.

Este documento descreve os Conceitos e requisitos funcionais necessários para o cálculo da Simulação, mas o detalhamento do processo de cálculo em si é descrito em outro documento.

Esta versão do documento ainda não detalha os Conceitos e requisitos funcionais necessários para o cálculo de emissões em CO₂eq. No entanto, pode-se adiantar já que este cálculo aproveitará em grande medida os Conceitos aqui descritos, especialmente Fluxos, Recursos e Infraestrutura, e que este cálculo deverá ser compatível com o calculado pelo aplicativo “E-Missions”, já disponível.

B.3 Safras e Ciclos de produção

O Sistema transforma Recursos ao longo de um período de tempo denominado Ciclo de produção e medido em dias.

A Safra é um período de tempo de 365 dias em que vários Ciclos de produção se sobrepõem. A Plataforma supõe que vários Ciclos de produção intervalados em exatamente 365 dias se sobrepõem na Área de produção do Sistema, sendo que um destes Ciclos de produção produz uma certa Quantidade do Produto no dia 365.

Por convenção, o primeiro dia da Safra é o dia 1, e o último dia da Safra é o dia 365. A data no ano que corresponde a estes dias é definida por Parâmetro do Modelo.

Embora a duração da Safra seja de 365 dias, o Ciclo de produção pode levar mais ¹ ou menos ² do que 365 dias. Mesmo nestes casos, a Safra terá 365 dias, e o Produto “sai” do Sistema no último dia da Safra, o dia 365.

Qualquer outra Saída do Sistema em qualquer outro dia da Safra ou de qualquer um dos Ciclos de produção é considerado Coproduto. O último dia da Safra não necessariamente é o último dia de um Ciclo de produção completo, pois alguns Coprodutos podem ser produzidos após o Produto³.

As Quantidades de Recursos que entram e saem do Sistema durante a Safra é dada pela soma das Quantidades dos Recursos que entram e saem de todos os Ciclos de produção entre 1 e 365, já descontado o reuso de Recursos que saem em um Ciclo de produção e podem ser aproveitados pelo Ciclo de produção seguinte, no mesmo dia da Safra⁴.

¹ P.ex., a compra de bezerros (Recurso) ocorre no mesmo ano em que ocorre a venda de bois gordos (Produto) que foram comprados como bezerros há mais de um ano.

² P.ex., sistemas de Recria ou Recria e Engorda.

³ Ou seja, o Ciclo de produção pode ter dias anteriores ao dia 1, e dias posteriores ao dia 365.

⁴ P.ex., novilhas nascidas e recriadas em um Ciclo de produção são inseminadas no Ciclo de produção seguinte.

B.4 Sistemas, Modelos, Estações e Recursos

As transformações pelas quais os Recursos passam ao longo de um Ciclo de produção são representadas por um Modelo matemático definido para o Sistema.

O Modelo pode ser decomposto em Estações conectadas por Fluxos.

Cada Estação do Modelo pertence a uma e somente uma Fase do Sistema⁵.

Uma Estação pode usar Área de um ou mais tipos⁶ durante o Ciclo de produção para transformar os Recursos disponíveis nas Entradas da Estação em Recursos disponíveis nas Saídas das Estação.

A maioria⁷ das Estações do Modelo usa animais do rebanho de produção, que permanecem na Estação entre o Início e o Término da Estação.

As Áreas utilizadas pela Estação são devolvidas ao ambiente após o seu Término, podendo ser reutilizadas por outra Estação ou pela mesma Estação, em outro Ciclo de produção ou no mesmo Ciclo de produção⁸.

As Áreas e os animais do rebanho de produção são exemplos de tipos de Recursos que uma Estação recebe em suas Entradas, transforma em outros tipos de Recursos, e devolve em suas Saídas.

Outros tipos de Recursos são Insumos (fertilizantes, p.ex.) e Serviços (consultorias, p.ex.)⁹.

Uma Estação pode possuir mais de uma Entrada ou Saída para um mesmo tipo de Recurso, e o Modelo pode conter mais de uma Estação que produza ou consuma um mesmo tipo de Recurso.

As quantidades de Recursos nas Entradas e Saídas das Estações a cada dia do Ciclo de produção e da Safra são calculadas pela Simulação do Modelo, usando os valores dos Parâmetros do Modelo definidos para a Simulação.

⁵ “Cria”, “Recria, ou “Engorda”, ver especificação do Módulo 1.

⁶ Hoje não tem sentido pensar uma Estação que não use Área, ou que use Áreas de mais de um Tipo, mas mesmo esta regra não será exigida pela Plataforma, e será garantida pelo Usuário Desenhista, no cuidado no desenho do Modelo.

⁷ Estações de manutenção ou reforma de pasto, em geral, não incluem animais.

⁸ É responsabilidade do Desenhista garantir que a cada dia da Safra a Área total usada pelo Modelo seja sempre a mesma, para quaisquer valores de Parâmetros ajustados pelo Usuário Cliente para a Simulação. Boas práticas de modelagem, descritas em outro documento, garantem isto.

⁹ Insumos e Serviços geralmente são apenas consumidos, isto é, aparecem apenas em Entradas, e não nas Saídas, mas também aqui cabe ao Usuário Desenhista especificar, e nenhuma regra da Plataforma imporá esta restrição.

B.5 Estações e Durações

A Duração de uma Estação é o intervalo de tempo em dias entre o dia de Início e o dia de Término da Estação. Para qualquer Estação: Início + Duração = Término + 1¹⁰.

A Duração de uma Estação é calculada no início da Simulação do Modelo, usando os valores dos Parâmetros definidos para a Simulação. As datas de Início e Término são calculadas pela Simulação a partir da Duração da Estação e do Início ou Término de uma outra Estação conectada a esta¹¹.

Os Recursos na Entrada da Estação entram nesta no dia de Início da Estação, e lá permanecem por uma Fração da Duração que é calculada a partir dos valores dos Parâmetros definidos para a Simulação. Todos os Recursos de Saída da Estação saem da Estação no dia de Término, tendo nela permanecido pela fração restante da Duração da Estação¹².

B.6 Fluxos, Entradas e Saídas

O Modelo de um Sistema pode ser representado por um grafo em que as Estações são conectadas por Fluxos.

Considerando o relacionamento entre Fluxos e a fronteira do Modelo (e do Sistema), há três tipos de Fluxo:

- Fluxos internos ao Modelo, que conectam uma Saída de uma Estação a uma e somente uma Entrada de uma outra Estação, de mesmo tipo de Recurso;
- Fluxos de entrada no Modelo, que conectam uma ou mais Entradas de uma ou mais Estações a uma e somente uma Entrada do Modelo, de mesmo tipo de Recurso;
- Fluxos de saída do Modelo, que conectam uma ou mais Saídas de uma ou mais Estações a uma e somente uma Saída do Modelo, de mesmo tipo de Recurso.

O grafo que representa o Modelo é orientado segundo dois sentidos independentes:

- O sentido em que Recursos fluem entre Estações;
- O sentido em que o grafo deve ser percorrido para cálculo da Simulação.

O grafo do Modelo é acíclico nestes dois sentidos.

¹⁰ Portanto, Início e Término iguais representam uma Estação com um dia de Duração, e não é possível representar Estações com Duração zero.

¹¹ Para detalhes, ler documento de especificação do cálculo da Simulação.

¹² Ou seja, para modelar situações que exigem dias diferentes de Entrada ou de Saída de algum Recurso, é preciso usar duas ou mais Estações.

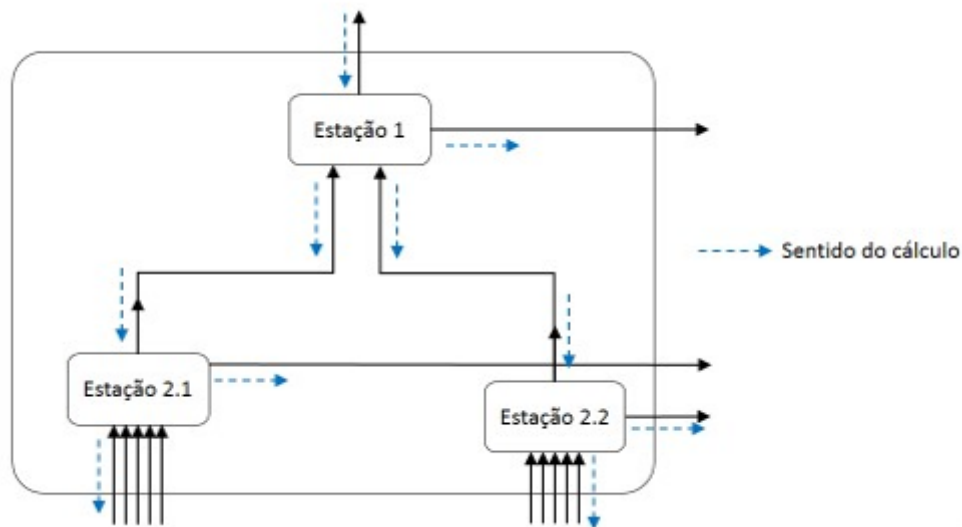


Figura 38 – O grafo do Modelo.

Considerando o relacionamento entre o sentido dos Fluxos e do cálculo da Simulação, há dois tipos de Fluxos:

- Fluxos de Transformação;
- Fluxos de Tratamento.

Na representação adotada nas figuras deste documento, os Fluxos horizontais são de Tratamento, e os verticais de Transformação.

Nos Fluxos de Transformação (verticais), o sentido no grafo a ser percorrido para o cálculo da Simulação (de cima para baixo) é oposto ao sentido com que os Recursos fluem (de baixo para cima).

Nos Fluxos de Tratamento (horizontais), o sentido do cálculo da Simulação (da esquerda para a direita) acompanha o sentido com que os Recursos fluem no grafo (da esquerda para a direita).

A Entrada e Saída conectadas por um Fluxo acompanham o tipo do Fluxo: Tratamento ou Transformação. Sendo assim, nas figuras presentes deste documento:

- Entradas de Estações (que vêm de Fluxo) de Transformação são representadas no lado inferior das Estações;
- Saídas de Estações (que vão para Fluxo) de Transformação são representadas no lado superior das Estações;
- Entradas de Estações (que vêm de Fluxo) de Tratamento são representadas no lado esquerdo das Estações;

- Saídas de Estações (que vão para Fluxo) de Tratamento são representadas no lado direito das Estações.

As Entradas (que atravessam a fronteira) do Modelo são sempre de Transformação. As Saídas (que atravessam a fronteira) do Modelo são sempre de Tratamento, exceto por uma única Saída de Transformação, que representa o Produto.

As Saídas de Tratamento podem representar Coprodutos, Perdas, Resíduos ou Emissões. Esta diferenciação é usada no cálculo da Simulação. P.ex., Coprodutos possuem valor de mercado, isto é, Preço maior do que zero, e por isso geram Receita.

B.7 Estações de Transformação e Tratamento

As Estações também são classificadas em Transformação e Tratamento, e isto define as Entradas e Saídas destas Estações e como considerá-las no cálculo da Simulação.

B.7.1 Estação de Transformação

As Estações de Transformação visam à garantir o suprimento total da Quantidade consumida por um Fluxo de Transformação que entra em outra Estação. A Estação de Transformação pode também consumir outros Recursos em suas Entradas de Transformação e suprir outros Recursos em suas Saídas de Tratamento.

A Estação de Transformação possui uma e somente uma Saída para Fluxo de Transformação. Este Fluxo é Referência para o cálculo dos demais Fluxos que entram e saem da Estação. Além disso, a Estação de Transformação possui:

- Nenhuma Entrada (de Fluxos) de Tratamento;
- Uma ou mais Entradas (de Fluxos) de Transformação;
- Zero¹³, uma ou mais Saídas (para Fluxos) de Tratamento;
- Uma Saída (de Fluxo) de Transformação.

São exemplos de Estações de Transformação:

- Desmama de bezerros e bezerras;
 - Saída (para Fluxo) de Transformação: Bezerro desmamado;
 - Entradas (de Fluxos) de Transformação: Pastagem nativa, Medicamentos, etc.;

¹³ Estações geralmente possuem apenas uma Entrada e uma Saída de Área, e uma Entrada e uma Saída de animais, mas esta regra deve ser cuidada pelo Usuário Desenhista, e não será exigida pela Plataforma.

- Saída (para Fluxos) de Tratamento: Bezerras desmamadas, vacas, etc.;
- Recria de bezerro
 - Saída (para Fluxo) de Transformação: Boi magro no peso certo para confinamento;
 - Entradas (de Fluxo) de Transformação: Pastagem plantada, Medicamentos, etc.;
 - Saída (para Fluxo) para Tratamento: Boi magro para estação adicional de recria a pasto, porque não atingiu peso.

B.7.2 Estação de Tratamento

A Estação de Tratamento visam à garantir o consumo total da Quantidade suprida por um Fluxo de Tratamento que sai de outra Estação. A Estação de Tratamento pode também consumir outros Recursos em suas Entradas de Transformação e suprir outros Recursos em suas Saídas de Tratamento.

A Estação de Tratamento possui uma e somente uma Entrada de Fluxo de Tratamento. Este Fluxo é Referência para o cálculo dos demais Fluxos que entram e saem da Estação. Além disso, a Estação de Tratamento possui:

- Uma ou mais Saídas (para Fluxos) de Tratamento;
- Zero, uma ou mais Entradas de (Fluxos de) Transformação;
- Nenhuma Saída (para Fluxos) de Transformação.

São exemplos de Estações de Tratamento:

- Reforma de pasto ¹⁴;
 - Entrada (de Fluxo) de Tratamento: Pasto precisando de reforma;
 - Entradas (de Fluxos) de Transformação: Sementes, Fertilizantes, etc.;
 - Saída (para Fluxo) de Tratamento: Pasto reformado, Resíduos, etc.;
- Recria de bezerra;
 - Entrada (de Fluxo) de Tratamento: Bezerra desmamada;
 - Entradas (de Fluxo) de Transformação: Pastagem nativa, Medicamentos, etc.;
 - Saída (para Fluxo) de Tratamento: Novilha para reprodução, Novilhas para descarte, etc.

¹⁴ Este é um exemplo de Estação que, de fato, não se repete anualmente, mas a repetição anual é premissa básica da Plataforma. Para modelar estes casos, o Desenhista deve incluir Estações de repetição anual com frações proporcionais dos Recursos (Áreas, Insumos, Serviços).

B.8 Recursos

Neste texto, usamos a expressão “Recurso” ou “tipo de Recurso” para representar um item de uma hierarquia que classifica a natureza de tudo aquilo que pode fluir entre Entradas e Saídas do Modelo e suas Estações.

Apenas os itens nas “folhas” desta hierarquia podem ser atribuídos a uma Entrada ou Saída de uma Estação.

Os níveis mais altos da hierarquia que classifica os Recursos são comuns a todos os Modelos, de modo a permitir a comparação de Indicadores e Parâmetros entre Simulações de Modelos diferentes.

Apenas um tipo de Recurso pode ser atribuído a cada Fluxo.

Alguns tipos de Recursos podem ser sinalizados como variáveis de Quantidade inteiras (p.ex., Categorias animais). Para estes, a Plataforma fará todos os cálculos de Quantidades supondo que são quantidades decimais com dupla precisão, como todas as Quantidades dos demais Recursos¹⁵. Os valores de Quantidades sinalizadas como inteiras serão arredondados para números inteiros apenas no momento da apresentação pelo Módulo 3 da Plataforma.

São exemplos de tipos de Recursos:

- Uma Categoria animal (p.ex., “Vaca com bezerro ao pé”);
- Um tipo de Área (p.ex., “Pastagem nativa”);
- Um tipo de Insumo (p.ex., “NPK 4-14-8”);
- Um tipo de Serviço (p.ex., “Consulta de veterinário”).

A classificação dos tipos de Recursos no nível mais alto da hierarquia define tratamentos diferenciados das Quantidades do Recurso em cada Fluxo e Estação, no cálculo da Simulação:

B.8.1 Área

Área(s) necessária(s) no Início da Estação, e devolvida(s) no Término, na maioria das vezes como Coprodutos. P.ex., “Pastagem nativa” ou “Pastagem formada”, ou “Pasto a recuperar”.

¹⁵ Bem como quase todas as demais variáveis numéricas da Plataforma, a única exceção são os Parâmetros inteiros de lista de opções, ver documentação dos Módulos 0 e 1.

B.8.2 Categoria (animal)

Agrupam ¹⁶ os animais do rebanho bovino ¹⁷ por características como o gênero (macho ou fêmea), raça e função no rebanho, de modo que, a cada dia, cada animal do rebanho bovino do Modelo pertence a uma e somente uma Categoria animal¹⁸.

A Quantidade contada em Categoria animal é cabeças, isto é, indivíduos, não peso, para simplificar a modelagem matemática dos Fluxos.

Apenas uma Categoria animal é Produto do Sistema¹⁹. Outras Categorias animais são sempre Coprodutos.

O peso ou idade inicial ou final dos animais não definem nem podem estar limitados pela descrição de uma Categoria, pois os pesos e idade mínimos e máximos das Categorias podem variar a cada Simulação, a depender de Parâmetros de desempenho e Durações que podem ser ajustadas pelo Usuário Cliente, embora obviamente os pesos e idades não saem de determinadas faixas, para Simulações com Parâmetros dentro dos valores limites definidos pelo Usuário Desenhista.

Mais de uma Categoria animal pode entrar e sair de uma mesma Estação, nos casos em que animais de mais de uma Categoria compartilham uma mesma área física no mesmo período de tempo. Por exemplo, uma certa Quantidade de “Vacas com bezerro ao pé” e “Bezerros recém-nascidos” entram numa Estação de Amamentação, e uma certa Quantidade de “Vacas” e “Bezerros desmamados” saem (além de Perdas). Outro exemplo: Uma certa Quantidade de Touros e Vacas entram e saem de uma Estação de Monta.

Os Fluxos das Categorias Animais que entram e saem da Estação, bem como de Área(s), Insumos, e Serviços necessários para a Transformação (ou Tratamento) das várias Categorias animais envolvidas na Estação, são calculados a partir de um único Fluxo de Saída de Transformação (ou de Entrada de Tratamento) de uma das Categorias envolvidas, conforme descrito no tópico “Estações de Transformação e Tratamento”.

B.8.3 Insumo

Representa material consumido pela Estação (p.ex., Fertilizantes, Suplementos).

¹⁶ Ou seja, os animais individuais não são modelados na Plataforma, mas representados como Quantidades de suas Categorias.

¹⁷ Os animais de trabalho (p.ex., equinos) são modelados como Infraestrutura, como descrito mais adiante.

¹⁸ O termo “Categoria animal” foi escolhido pois aparenta ser de uso comum entre os Usuários produtores e Pesquisadores.

¹⁹ P.ex., “Boi gordo” em Modelo de Sistema que inclui a Fase de Engorda, ou “Bezerro desmamado”, em Modelo de Sistema com apenas a Fase de Cria.

B.8.4 Serviço

Prestação de serviço que dependa do uso de ativos e mão-de-obra temporária não incluídos na Infraestrutura do Modelo²⁰.

B.9 Unidade de medida

Cada tipo de Recurso possui uma Unidade de medida. A Unidade de medida é definida pelo Usuário Desenhista, uma por tipo de Recurso no nível “folha” (mais baixo) da hierarquia de tipos de Recursos.

B.10 Preços

Cada tipo de Recurso está associado a dois Preços, não necessariamente iguais: um Preço de Compra para quando o Recurso for Entrada do Modelo, e um Preço de Venda para quando o Recurso for Saída do Modelo. Os Preços de Compra e de Venda valem para a Safra inteira.

O Preço do Recurso já inclui qualquer custo de transação por necessidade de compras em lote, custos de estocagem, etc.

Os Preços de Compra e de Venda de cada tipo de Recurso são Parâmetros do Modelo com valores que podem ser ajustados pelo Usuário Cliente para uma dada Simulação.

Os Preços de Compra e de Venda são definidos por tipo de Recurso e sempre maiores do que zero, independentemente se este atravessa a fronteira do Modelo como Entrada ou Saída de Produto, Coproduto, Perda, Resíduos e Emissões, etc.

O cálculo da Simulação não considera, no cálculo de Indicadores econômicos, os Preços de Venda de Recursos que estiverem sinalizados nos Fluxos de Saída do Modelo como Perdas, Resíduos e Emissões. Custos com o Tratamento de Perdas, Resíduos e Emissões devem ser internalizados no Modelo, através da inclusão de Estações de Tratamento adequadas.

B.11 Infraestrutura

Além dos Recursos, o Sistema depende também de itens de Infraestrutura.

Os itens de Infraestrutura são atributos do Modelo do Sistema que não estão associados a nenhuma Estação específica. A quantidade de itens de Infraestrutura do Modelo do Sistema, por tipo, é estável ao longo da Safra e é definida por Parâmetros do Modelo mantidos pelo Usuário Desenhista do Modelo mas que podem ser ajustados pelo

²⁰ Infraestrutura do Modelo definida mais adiante.

Usuário Cliente para cada Simulação, se este ajuste tiver sido habilitado no Modelo pelo Usuário Desenhista e dentro de limites definidos por ele definidos.

Os itens de Infraestrutura são classificados em uma hierarquia que, em seu nível mais alto, é dividida em: Instalações, Equipamentos, Animais de trabalho ²¹ e Mão-de-obra própria. Esta classificação no nível mais alto da hierarquia é considerada pela Simulação para o cálculo de Indicadores específicos.

A hierarquia que classifica os itens da Infraestrutura é comum a todos os Modelos, de modo a permitir a comparação de Indicadores e Parâmetros entre Simulações de Modelos diferentes.

De modo semelhante aos Recursos, os itens da Infraestrutura também possuem Unidades de medida. Além da Unidade de medida, os itens da Infraestrutura possuem outros Parâmetros associados que são utilizados nas Fórmulas de cálculo de Indicadores.

B.12 Fórmulas

Alguns cálculos precisam ser feitos no início da Simulação, antes do cálculo dos Fluxos²². Outros cálculos precisam ser feitos após o cálculo dos Fluxos. A especificação destes cálculos se dá através de fórmulas matemáticas simples.

B.12.1 Fórmulas de cálculo de Coeficientes

As Quantidades diárias de Entradas e Saídas das Estações são calculadas na Simulação a partir de Coeficientes. Antes do cálculo dos Fluxos, cada Coeficiente de cada Fluxo é calculado por meio de uma Fórmula associada ao Coeficiente. Cada Fórmula de cálculo de Coeficientes usa um ou mais Parâmetros do Modelo.

B.12.2 Fórmulas de cálculo de Duração e Fração

A Duração de cada Estação e a Fração da Duração em que os Recursos permanecem na Entrada da Estação são também calculadas por meio de Fórmulas, antes do cálculo dos Fluxos.

B.12.3 Fórmulas de cálculo de Preços

Há muitas situações em que o Preço de um Recurso precisa ser calculado a partir de Parâmetros. Por exemplo, Recursos de Categoria animal são contados em quantidade, não em peso, e portanto os Preços precisam ser definidos por cabeça, para cada Categoria

²¹ A Plataforma não modelará produção própria de animais de trabalho.

²² Mais detalhes no documento que descreve o cálculo da Simulação.

animal, embora para o Usuário Cliente possa fazer mais sentido definir um preço por arroba. Para poder tratar estes casos, Parâmetros de Mercado e de Desempenho são usados em Fórmulas de cálculo de Preços. No exemplo dado, Fórmulas para Preços podem calcular Preços de Categorias a partir de Parâmetros de Desempenho em @/cabeça para cada Categoria, e um Parâmetro de Mercado que define R\$/@.

Os Preços são calculados após ²³ o cálculo dos Fluxos e antes do cálculo do Indicadores.

B.12.4 Fórmulas de cálculo de Indicadores

Cada Indicador é calculado usando-se uma Fórmula associada ao Indicador. Além de Parâmetros, as Fórmulas usadas no cálculo de Indicadores podem utilizar as Quantidades das Entradas e Saídas de cada Estação e os Preços calculados para cada Recurso.

Os Indicadores são calculados após o cálculo dos Fluxos.

B.12.5 Características comuns a todas as Fórmulas

Em geral, as Fórmulas envolvem apenas as quatro operações e condições do tipo “Se ... então ... senão...”, para tratar os casos que envolvam Parâmetros que são números inteiros ou lista de opções, mas é recomendável ²⁴ que seja permitido que as Fórmulas sejam descritas na forma de qualquer texto que possa ser executado pelas rotinas de cálculo da Simulação.

Uma Fórmula pode ser simplesmente uma equivalência a um Parâmetro ou a uma constante numérica.

As Fórmulas são cadastradas pelo Usuário Desenhista do Modelo.

Nos casos em que o Usuário Desenhista quiser utilizar resultados de uma Fórmula em outra, ele deve repetir as Fórmulas precedente(s), para evitar cálculo circular.

²³ Poderiam ser calculados antes ou mesmo em paralelo, pois não dependem nem afetam o cálculo dos Fluxos.

²⁴ Este requisito pode ser adiado para futuras versões e depende da solução técnica que for escolhida para a implementação de Fórmulas.

ANEXO C – Módulo 2 – Cálculo da Simulação

C.1 Objetivo

Este documento descreve o processo de cálculo da Simulação, escopo do “Módulo 2” da Plataforma +Precoce, também chamado de “Simulador”, com o objetivo de validar o entendimento com a equipe do projeto e dar subsídios para desenho da Plataforma pela equipe de engenharia de software do projeto.

A descrição aqui não pretende ser uma descrição do algoritmo, mas sim deixar claro o resultado que se espera.

É recomendável a leitura prévia de documentos especificando conceitos e requisitos funcionais dos Módulos 0, 1 e 2 da Plataforma.

C.2 Introdução

As etapas de Cálculo podem ser divididas em três grandes etapas sequenciais:

- Cálculo de Coeficientes. Durações e Preços;
- Cálculo de Fluxos;
- Cálculo de Indicadores.

C.3 Cálculo de Coeficientes, Durações e Preços

Pode ser subdividida nas etapas descritas a seguir, que podem ser calculadas em paralelo.

C.3.1 Calcula Coeficientes

Calcula o Coeficiente de cada Fluxo do Modelo, a partir dos valores definidos pelo Usuário Cliente para os Parâmetros para a Simulação do Modelo, e da Fórmula definida pelo Usuário Desenhista para cada Coeficiente. Estes Coeficientes serão usados no cálculo das Quantidades diárias de cada Fluxo do Modelo.

C.3.2 Calcula Durações

Calcula a Duração de cada Estação do Modelo, a partir dos valores definidos pelo Usuário Cliente para os Parâmetros para a Simulação do Modelo e da Fórmula definida pelo Usuário Desenhista para cada Duração. Estas Durações serão usadas no cálculo do Início e Término de cada Estação do Modelo.

Calcula também a Fração da Duração que os Recursos de Entrada permanecem na Entrada de cada Estação, a partir dos valores definidos pelo Usuário Cliente para os Parâmetros para a Simulação do Modelo e da Fórmula definida pelo Usuário Desenhista para cada Fração.

C.3.3 Calcula Preços

Calcula o Preço de cada Recurso presente nas Entradas e Saídas de Coprodutos do Modelo, a partir dos valores definidos pelo Usuário Cliente para os Parâmetros para a Simulação do Modelo, e da Fórmula definida pelo Usuário Desenhista para cada Preço. Estes Preços serão usados no cálculo das Receitas e Custos das Entradas e Saídas de Coproduto do Modelo.

C.4 Cálculo de Fluxos

Calcula as Quantidades diárias nas Entradas e Saídas de cada Estação do Modelo, a partir dos Coeficientes dos Fluxos e Durações das Estações calculados na etapa anterior e do grafo¹ do Modelo. Pode ser subdividida nas seguintes etapas sequenciais descritas a seguir.

C.4.1 Calcula um Ciclo de produção

Percorre o grafo do Modelo calculando a Quantidade diária de Recursos em cada Entrada e cada Saída de cada Estação, em cada dia de um Ciclo de produção.

A Quantidade de Produto na Saída do Modelo é fixa e dada por valor default de Parâmetro interno² do Sistema. O dia em que esta Quantidade ocorre é o último dia da Safra, isto é, 365.

A Quantidade de Produto na Saída do Modelo é usada para o cálculo das Quantidades em todas as demais Entradas e Saídas, percorrendo-se as Estações seguindo o sentido do cálculo no grafo do Modelo.

¹ Isto é, o conjunto de Estações e Fluxos do Modelo, e seus relacionamentos.

² Isto é, oculto do Usuário Cliente, acessível ao Usuário Administrador.

Em cada Estação de Transformação, a Quantidade de Recursos em cada Entrada de Transformação e em cada Saída de Tratamento é calculada a partir da Quantidade de Recursos na **Saída de Transformação** multiplicada pelos Coeficientes dos demais Fluxos.

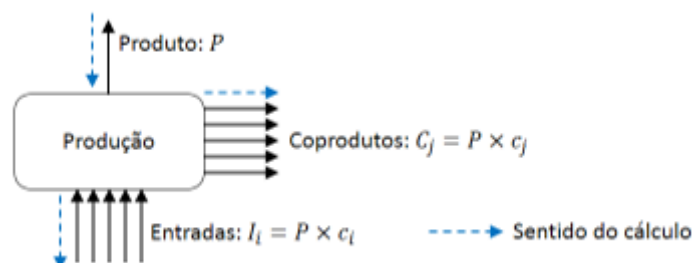


Figura 39 – Cálculo em Estação de Transformação.

Em cada Estação de Tratamento, a Quantidade de Recursos em cada Entrada de Transformação e em cada Saída de Tratamento é calculada a partir da Quantidade de Recursos na **Entrada de Tratamento** multiplicada pelos Coeficientes dos demais Fluxos.

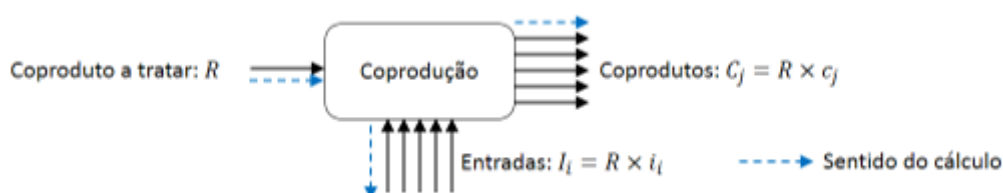


Figura 40 – Cálculo em Estação de Tratamento.

Quanto à distribuição das Quantidades calculados no tempo, a Plataforma supõe que:

- Os Recursos entram nas Estações no dia de Início da Estação (inclusive) e permanecem na Entrada da Estação por tempo correspondente a uma Fração da Duração³;
- Os Recursos de Saída surgem dentro da Estação no dia seguinte (inclusive) e permanecem na Saída da Estação até o dia de Término da Estação (inclusive);
- A Estação que tem por Entrada um dos Fluxos que sai de uma dada Estação, tem como Início o dia seguinte ao Término da Estação de onde sai o Fluxo;

Descrição resumida do cálculo é apresentada a seguir:

³ Default = 50%, mínimo = 1%, máximo = 100%, para provavelmente todas as Estações. A Fórmula de cálculo das Durações deve garantir tempo mínimo de permanência das Entradas de 1 dia, e tempo mínimo de permanência das Saídas em zero.

- Para a última Estação de Transformação (isto é, a que produz o Produto do Sistema):
 - Assume a Quantidade diária na Saída de Transformação igual ao valor default de Parâmetro interno do Sistema;
 - Assume o dia de Término da Estação como igual ao dia de Término da Safra⁴;
 - Assume o dia de Início da Estação como igual a Término – Duração + 1;
 - Calcula a Quantidade diária em cada Saída de Tratamento usando o Coeficiente do Fluxo correspondente multiplicado pela Quantidade na Saída de Transformação;
 - Calcula a Quantidade diária em cada Entrada de Transformação usando o Coeficiente do Fluxo correspondente multiplicado pela Saída de Transformação;
 - Assume que a Quantidade diária na Saída de Transformação se repete desde Início + Duração * Fração até o dia de Término da Estação⁵;
 - Assume que a Quantidade diária em cada Saída de Tratamento se repete desde Início + Duração * Fração até o dia de Término da Estação⁶;
 - Assume que a Quantidade diária em cada Entrada de Transformação se repete desde Início até Término - Duração * Fração⁷;
- Para cada uma das demais Estações de Transformação⁸:
 - Assume a Quantidade diária de Saída de Transformação como igual à Quantidade diária da Entrada de Estação conectada à Saída desta Estação por Fluxo de Transformação;
 - Assume o dia de Término da Estação como o dia anterior ao **Início da Estação** conectada a Saída desta Estação por Fluxo de Transformação;
 - Assume o dia de Início da Estação como igual a Término – Duração + 1;
 - Calcula a Quantidade diária em cada Saída de Tratamento usando o Coeficiente do Fluxo correspondente multiplicado pela Saída de Transformação;
 - Calcula a Quantidade diária em cada Entrada de Transformação usando o Coeficiente do Fluxo correspondente multiplicado pela Saída de Transformação;
 - Assume que a Quantidade diária na Saída de Transformação se repete desde Início + Duração * Fração até Término da Estação⁹;

⁴ Isto é, 365.

⁵ Parte inteira apenas, se duração ímpar.

⁶ Parte inteira apenas, se duração ímpar.

⁷ Arredonda para cima, para compensar o valor truncado assumido para Saídas.

⁸ Em negrito o que muda com relação ao algoritmo anterior.

⁹ Parte inteira apenas, se duração ímpar.

- Assume que a Quantidade diária em cada Saída de Tratamento se repete desde Início + Duração * Fração até Término da Estação¹⁰;
- Assume que a Quantidade diária em cada Entrada de Transformação se repete desde Início até Término - Duração * Fração¹¹.
- Para cada uma das Estações de Tratamento¹²:
 - Assume a Quantidade diária de **Entrada de Tratamento** como igual à Quantidade diária da **Saída de Estação** conectada a esta **Entrada** por Fluxo de **Tratamento**;
 - Assume o dia de **Início** da Estação como o dia **seguinte** ao **Término** da Estação conectada à **Entrada** desta Estação por Fluxo de **Tratamento**;
 - Assume o dia de Término da Estação como igual a **Início + Duração - 1**;
 - Calcula a Quantidade diária em cada Saída de Tratamento usando o Coeficiente do Fluxo correspondente multiplicado pela **Entrada de Tratamento**;
 - Calcula a Quantidade diária em cada Entrada de Transformação usando o Coeficiente do Fluxo correspondente multiplicado pela **Entrada de Tratamento**;
 - Assume que a Quantidade diária na **Entrada de Tratamento** se repete desde Início da Estação até Término - Duração * Fração¹³;
 - Assume que a Quantidade diária em cada Saída de Tratamento se repete desde Início + Duração * Fração até Término da Estação¹⁴;
 - Assume que a Quantidade diária em cada Entrada de Transformação se repete desde Início até Término - Duração * Fração¹⁵.
- Calcula e armazena¹⁶:
 - O Início do Ciclo de produção como o menor dia de Início dentre todas as Estações do Modelo;
 - O Término do Ciclo de produção como o maior dia de Término dentre todas as Estações do Modelo;
 - A Duração do Ciclo de produção como sendo igual a Término – Início + 1¹⁷.

¹⁰ Parte inteira apenas, se duração ímpar.

¹¹ Arredonda para cima, para compensar o valor truncado assumido para Saídas.

¹² Em negrito o que muda com relação ao algoritmo anterior.

¹³ Arredonda para cima, para compensar o valor truncado assumido para Saídas.

¹⁴ Parte inteira apenas, se duração ímpar.

¹⁵ Arredonda para cima, para compensar o valor truncado assumido para Saídas.

¹⁶ O Início, Término e Duração do Ciclo de produção serão depois usados em cálculo de Indicadores.

¹⁷ Repare que o Ciclo de produção pode terminar após a Saída do Produto do Sistema, ou seja, após o dia 365, pois Estações de Coprodução podem terminar depois. Além disso, o dia Início do Ciclo de produção pode ser zero ou negativo, isto é, antes do início da Safra (dia 1).

C.4.2 Calcula a Safra

As Quantidades de Entradas e Saídas calculadas na etapa anterior visam a atender um Ciclo de produção apenas, mais especificamente, aquele cujo último dia de Saída do Produto coincide com o dia de término da Safra (isto é, dia 365). Falta ainda somar as Quantidades diárias das Entradas e Saídas dos Ciclos de produção que atenderão os anos seguintes e cujas datas se sobrepõem no tempo ao período da Safra (dias de 1 a 365). Como Ciclos de produção se repetem anualmente, com a data da Saída do Produto do Sistema se repetindo sempre no dia 365 de cada ano, não é necessário recalculá-lo para vários Ciclos de produção, bastando usar o Ciclo de produção já calculado para repetir os próximos:

- Para cada Entrada e Saída de cada Estação, e para cada dia do Ciclo de produção calculado:
 - Se o dia em que a Quantidade ocorre for anterior ao início da Safra¹⁸, desloca o dia em que a Quantidade ocorre somando o menor múltiplo de 365 dias suficiente para que o dia se torne maior ou igual ao início da Safra (isto é, dia 1);
 - Se o dia em que a Quantidade ocorre for posterior ao término da Safra¹⁹, desloca o dia em que a Quantidade ocorre subtraindo o menor múltiplo de 365 dias suficiente para que o dia se torne menor ou igual ao término da Safra (isto é, dia 365).

C.4.3 Totaliza as Entradas e Saídas do Modelo

O Produto do Sistema torna-se disponível para o ambiente no dia de Término da última Estação, que é o último dia da Safra²⁰.

As demais Saídas do Modelo consolidam a Quantidade de Recursos que saem do Sistema em Fluxos de Tratamento por tipo de Recurso e por dia da Safra.

As Entradas do Modelo consolidam a Quantidade de Recursos que entram no Sistema, por tipo de Recurso e por dia da Safra.

- Supõe inicialmente zeradas todas as Quantidades diárias de todas as Entradas e Saídas do Modelo, em todos os dias da Safra;
- Para a única Saída de Produto do Modelo:

¹⁸ Isto é, menor do que 1.

¹⁹ Isto é, maior do que 365.

²⁰ Por definição, o dia 365.

- Na data de Término da Safra, assume a Saída de Produto como a Saída de Transformação da última Estação.
- Para cada Saída de Coproduto, Perda, Resíduo, ou Emissão do Modelo, para cada Fluxo de Tratamento conectado a esta Saída e Estação de onde parte este Fluxo:
 - Na data de Término da Estação, soma a Quantidade da Saída da Estação à Saída correspondente do Modelo.
- Para cada Entrada do Modelo, para cada Fluxo de Transformação conectado a esta Entrada e Estação para onde vai este Fluxo:
 - Na data de Início da Estação, soma a Quantidade da Entrada da Estação à Entrada correspondente do Modelo.

C.4.4 Reúsa os Coprodutos do Modelo

As Entradas do Modelo podem reusar os Coprodutos disponíveis nas Saídas do Modelo que forem do mesmo tipo de Recurso e estiverem disponíveis no dia da Safra em que forem necessários²¹. Perdas, Resíduos e Emissões não são reusados nas Entradas do Modelo.

O ajuste das Quantidades nas Entradas e Saídas do Modelo de modo a considerar o reúso é descrito a seguir:

- Para cada dia da Safra²² e cada Entrada do Modelo, procura Saída de Coproduto do Modelo que seja do mesmo tipo de Recurso com Quantidade diferente de zero no mesmo dia, e:
- Se a Quantidade no dia para a Entrada for maior ou igual do que a Quantidade no mesmo dia para a Saída de Coproduto, subtrai da Entrada a Quantidade da Saída e zera a Quantidade da Saída de Coproduto no dia;
- Senão, subtrai da Saída de Coproduto a Quantidade da Entrada e zera a Quantidade da Entrada.

C.4.5 Calcula a Área de produção default

Calcula a Área de produção default necessária para o Modelo produzir a Quantidade default de Produto do Sistema.

A Área de produção default necessária para o Sistema pode ser calculada assim:

²¹ Esta é a única situação em que um Coproduto vira Produto, isto é uma Saída “horizontal” é usada para alimentar uma Entrada “vertical”. É também a única situação em que há um ciclo no grafo, mas que é tratado de forma segura do modo como as etapas de cálculo foram definidas.

²² Isto é, de 1 a 365.

- Para cada dia da Safra, soma as Quantidades de todos os tipos de Áreas nas Entradas e Saídas de todas as Estações do Sistema, naquele dia;
- Assume a Área de produção necessária como o máximo²³ das somas encontradas nos dias da Safra.

C.4.6 Ajusta as Quantidades à Área de produção informada

As Quantidades de Recursos e Área de produção default necessária foram calculadas até aqui para uma dada Quantidade default de Produto do Sistema e guardam entre si relações de proporcionalidade que são as mesmas para qualquer Quantidade do Produto do Sistema que for usada na Simulação no mesmo Modelo e com os mesmos valores para os demais Parâmetros. Portanto, é possível ajustar todas as Quantidades de Recursos do Sistema e a Quantidade do Produto do Sistema em proporção a uma nova Área de produção informada pelo Usuário Cliente, simplesmente multiplicando se todas as Quantidades (inclusive a de Produto do Sistema) por um Fator que represente o ajuste na Área de produção.

Quando o Usuário altera o Parâmetro Área de produção dentro dos limites definidos pelo Tamanho do Sistema, um Fator é calculado para este ajuste. P.ex., se a Área de produção informada pelo Usuário é 15% maior do que a Área de produção default calculada na etapa anterior, todas as Quantidades de todas as Entradas e Saídas devem ser multiplicadas por 1,15, inclusive a Quantidade do Produto do Sistema.

C.5 Cálculo de Indicadores

A maioria dos Indicadores são econômico-financeiros. Destes, a Plataforma irá calcular apenas os necessários para a elaboração de uma Demonstração de Resultado anual da Safra.

Os Indicadores podem ser calculados de forma consolidada (somatório, etc.) em agrupamentos definidos pelas hierarquias de Recursos, Parâmetros e dos próprios Indicadores.

Se um Sistema estiver associado a um Programa, alguns dos Indicadores econômico-financeiros considerarão, em suas Fórmulas, o efeito da bonificação do Programa, levando em conta os Parâmetros de qualidade.

A etapa de Cálculo de Indicadores pode ser subdividida nas etapas descritas a seguir, sendo que algumas delas podem ser calculadas em paralelo.

²³ Repare que este critério não exige que a Área usada pelo Modelo seja constante no tempo. Este requisito não é exigido pela Plataforma, mas é recomendado ao Usuário Desenhista.

C.5.1 Calcula Receitas e Custos

Calcula a Receita ou Custo gerado por cada Saída e Entrada do Modelo.

- Multiplica a Quantidade de cada Entrada e Saída do Modelo em todos os dias da Safra pelo Preço (de Entrada ou de Saída, conforme o caso) para o Recurso, e armazena o valor total de Receita e Custo da Safra em separado, para cada tipo de Recurso.

C.5.2 Calcula Indicadores de Infraestrutura

Esta etapa calcula o investimento, depreciação e custo de oportunidade implicados pelos itens de Infraestrutura, exceto Mão-de-obra própria.

Cada item de Infraestrutura possui Parâmetros de mercado usados no cálculo de investimento e depreciação: Investimento inicial, Vida útil, Valor residual e Juros sobre o capital e Parâmetro(s) de Desempenho usados para definir a quantidade deste item de Infraestrutura para o Sistema.

Para efeito de cálculo de investimento e depreciação, a Plataforma supõe que os Itens de Infraestrutura são adquiridos no início do Ciclo de produção que produziu a Safra corrente.

C.5.3 Calcula Indicadores de Mão de obra própria

Cada Perfil profissional é um tipo de Recurso de Mão-de-obra própria, com Custo trabalhista unitário definido por Parâmetro de mercado e Parâmetro(s) de Desempenho para definir a quantidade necessária de um dado Perfil profissional no Sistema.

C.5.4 Calcula Indicadores de Área de produção

Esta etapa calcula o investimento, depreciação e custo de oportunidade da Área de produção do Sistema, de modo semelhante ao que é feito para Infraestrutura, mas considerando a Quantidade total de Área por tipo calculada para o Modelo na etapa Cálculo de Fluxos.

C.5.5 Calcula Indicadores de rebanho

Esta etapa calcula o investimento, depreciação, valor residual e custo de oportunidade em animais do rebanho bovino de modo semelhante ao que é feito para Infraestrutura, mas considerando a Quantidade total de animais por Categoria no Modelo calculada para o Modelo na etapa Cálculo de Fluxos.

C.5.6 Calcula Indicadores ambientais

Esta etapa calcula Indicadores ambientais a partir de Fórmulas que consideram as Quantidades de Entradas e Saídas do Modelo para cada tipo de Recurso calculada para o Modelo na etapa Cálculo de Fluxos.

C.5.7 Calcula os demais Indicadores

Esta etapa calcula os Indicadores não calculados nas etapas anteriores, sempre usando Fórmulas que usam Parâmetros e Quantidades de Entradas e Saídas do Modelo para cada tipo de Recurso calculada para o Modelo na etapa Cálculo de Fluxos. P.ex. Indicadores de eficiência podem ser acrescentados, por exemplo, avaliando a utilização da Área de produção e os percentuais de Perdas. Alguns destes Indicadores podem ser especialmente úteis para o Usuário Desenhista, enquanto estiver refinando o Desenho do Modelo.

ANEXO D – Algoritmo para Cálculo da Simulação

Este documento define em inglês estruturado uma proposta de algoritmos para as principais funções (ou rotinas) necessárias ao cálculo da Simulação na Plataforma +Precoce.

A sintaxe utilizada nesta versão é um “mix” de linguagens de programação procedurais de sintaxe mais simples e popular comuns (Python, C, Javascript, Matlab / Octave, Lua, Julia), mas a sintaxe será ajustada em versões próximas para seguir a de uma linguagem de programação específica, provavelmente, Python.

A linguagem de programação que efetivamente será usada para codificar os algoritmos aqui descritos será a já usada pela equipe do CNPGC e definida para a Plataforma, isto é, PHP.

O trecho a seguir define as estruturas de dados que manterão na memória durante o processamento todo dado lidos da base de dados, resultados de cálculos intermediários, e resultados finais.

Algoritmo para cálculo da simulação

```

1 /* Armazena todos os dados da simulacao necessarios ao seu
2 calculo. */
3 structure Simulation {
4     /* Representa a qtde de dias entre fluxos de saida scuessivos
5     do produto do Sistema, ou seja, a qtde de dias entre ciclos de
6     producao concomitantes na fazenda e deslocados no tempo. Este
7     periodo foi chamado na especificacao de safra. Em principio,
8     simulation.cycle sera sempre 365 (ou seja, safras anuais), mas
9     sera definido por parametro Interno da plataforma para permitir
10    evolucoes (por exemplo, para modelar sistemas ILPF. */
11    .Cycle as Integer
12
13    /* Valores dos Parametros usados na Simulacao. */
14    .Parameter[:] as Float
15
16    /* Formulas que usam os Parametros para o calculo de
17    Coeficientes. */
18    .Formula[:] as Text
19

```

```
20  /* Nos do Grafo do Modelo. */
21  .Node[:] as Node
22
23  /* Fluxos que interligam os Nos do Grafo do Modelo. */
24  .Flow[:] as Flow
25
26  /* Qtde de cada Recurso em cada Estacao em cada dia de
27  simulation.Cycle, ja consolidando os Fluxos de varios Ciclos de
28  producao concomitantes. */
29  .Stock[node as Node, resource as Resource, day as Integer] as Float
30
31  /* Valores dos Indicadores calculados para a Simulacao. */
32  .Index[:] as Float
33 }
34
35 /* A estrutura a seguir define um No do Grafo do Modelo usado na
36 Simulacao e seus Fluxos. Esta estrutura aparece nas estruturas
37 "Simulation" e "Flow" (e vice-versa). Todas as referencias a um
38 objeto apontam para o mesmo objeto, ou seja, alteracoes em
39 atributos de um No afetam este no em todas as estruturas em que
40 este No aparece. */
41 structure Node {
42  /* Duracao da Estacao em Dias. Igual a zero, para nos que nao
43  forem Estacoes. */
44  .Duration as Integer
45
46  /* Numero de 0 a 1 que indica quantos dias da Duracao da Estacao
47  os Recursos dos Fluxos de Entrada permanecem na Estacao antes de
48  darem lugar aos Recursos dos Fluxos de Saida. */
49  .Permanence as Float
50
51  /* Inicio da Estacao. */
52  .Start as Integer
53
54  /* Termina da Estacao. Igual ao Inicio, se No nao for Estacao. */
55  .Finish as Integer
56
57  /* Verdadeiro, se No de Producao. */
58  .IsProduction as Boolean
59
60  /* Verdadeiro, se No de Tratamento. */
61  .IsTreatment as Boolean
```

```
62
63  /* Verdadeiro , se No de Reuso.*/
64  .IsReuse as Boolean
65
66  /* Verdadeiro , se No de Manutencao. */
67  .IsMaintenance as Boolean
68
69  /* Verdadeiro , se No Terminal (isto e, associado a um so Fluxo). */
70  .IsTerminal as Boolean
71
72  /* Todos os Fluxos que entram ou saem do No. A estrutura abaixo
73  aponta para os mesmos Fluxos, e existe apenas para facilitar a
74  referencia direta aos Fluxos de Tratamento ou Producao,
75  Entrada ou Saida. */
76  .Flow[:] as Flow
77  .Treatment
78      .Input
79          /* Fluxos de Tratamento que entram no No. */
80          .Flow[:] as Flow
81      .Output
82          /* Fluxos de Tratamento que saem do No. */
83          .Flow[:] as Flow
84  .Production
85      .Input
86          /* Fluxos de Producao que entram no No. */
87          .Flow[:] as Flow
88      .Output
89          /* Fluxos de Producao que saem do No. */
90          .Flow[:] as Flow
91  }
92
93  /* Um Fluxo do Grafo do Modelo */
94  structure Flow{
95      /* Coeficiente para calculo do Fluxo no No Predecessor. */
96      .Coeficient as Float
97
98      /* Qtde do Fluxo. */
99      .Qty as Float
100
101      /* Dia em que o Fluxo ocorre. */
102      .Day as Integer
103
```

```
104  /* Recurso do Fluxo. */
105  .Resource as Resource
106
107  /* No onde o Fluxo entra. */
108  .To as Node
109
110  /* No de onde o Fluxo sai. */
111  .From as Node
112
113  /* No de calculo anterior. Igual a '.To', se '.IsTreatment' ou
114  '.From', se .IsProduction. */
115  .Next as Node
116
117  /* Proximo No de calculo. Igual a '.To', se '.IsTreatment' ou
118  '.From', se .IsProduction. */
119  .Previous as Node
120
121  /* Verdadeiro, se Fluxo for de Producao. */
122  .IsProduction as Boolean
123
124  /* Verdadeiro, se Fluxo for de Tratamento. */
125  .IsTreatment as Boolean
126
127  /* Verdadeiro, se Fluxo for Terminal, isto e, for o unico
128  Fluxo de um No. */
129  .IsTerminal as Boolean
130
131  /* Verdadeiro, se Fluxo for Coproduto, isto e, Terminal de Saida
132  de Tratamento para a Tecnosfera com Preco de Venda > 0. */
133  .IsCoproduct as Boolean
134
135  /* Verdadeiro, se Fluxo for Residuo, isto e, Terminal de Saida
136  de Tratamento para a Tecnosfera com Preco de Venda <=0. */
137  .IsResidue as Boolean
138
139  /* Verdadeiro, se Fluxo for Emissao, isto e, Terminal de Saida
140  de Tratamento para o Ambiente. */
141  .IsEmission as Boolean
142
143  /* Verdadeiro, se Fluxo for Extracao, isto e, Terminal de Entrada
144  de Producao vinda do Ambiente. */
145  .IsExtraction as Boolean
```

```
146 }
147
148 /* A estrutura a seguir define um Recurso atribuível a um Fluxo. */
149 structure Resource {
150     ...
151 }
152
153 /* A funcao aqui definida calcula e atualiza na base de dados uma
154 Simulacao definida por 'simulationId', que e a chave da Simulacao
155 na base de dados. Esta funcao e executada quando o botao "Simular"
156 e pressionado, chama outras funcoes para as etapas do calculo,
157 salva a Simulacao na base de dados (e objetos dependentes) e
158 retorna a chave do registro da Simulacao criado. Todas as funcoes
159 aqui definidas atualizam o objeto que recebem como parametro.
160 Exceto onde explicitamente indicado, todo o processo e
161 sequencial. */
162 function calculateSimulation (simulationId) {
163     /* A funcao chamada a seguir cria uma estrutura global 'simulation'
164     que contera os dados da Simulacao durante as etapas de calculo. */
165     global simulation = createSimulation()
166     simulation.Id = simulationId
167
168     /* A funcao chamada a seguir carrega a estrutura 'simulation' com
169     os dados da Simulacao que estao na base de dados. */
170     loadSimulation(simulation)
171
172     /* A funcao chamada a seguir calcula os Coeficientes para os Fluxos
173     e as Duracoes das Estacoes da Simulacao a partir dos valores dos
174     Parametros armazenados na 'simulation', e atualiza os valores
175     destes Coeficientes e Duracoes nos Fluxos e Nos respectivos,
176     inclusive nas referencias redundantes presentes na estrutura. */
177     calculateCoefficients(simulation)
178
179     /* Percorre o Grafo da Simulacao calculando as Qtdes, Dias dos
180     Fluxos e datas de Inicio e Terminio dos Nos. */
181     calculateGraph(simulation)
182
183     /* Calcula o Estoque de cada Recurso em cada EstaCao em cada dia
184     de 1 a "simulation.Cycle", ja consolidando os Ciclos de producao
185     concomitantes. */
186     calculateStock(simulation)
187
```



```
188 /* Calcula os Indicadores financeiros , economicos , fisicos e
189 ambientais. */
190 calculateIndexes(simulation)
191
192 /* Atualiza a base de dados com os resultados dos calculos
193 armazenados na estrutura global 'simulation'. */
194 writeSimulation(simulation)
195 return
196 }
197
198 /* A funcao aqui definida calcula Fluxos e Nos, atualizando a
199 estrutura 'simulation' que, para isto, ja precisa ter carregados
200 o Grafo do Modelo do Sistema, os Coeficientes de todos os Fluxos
201 e as Duracoes de todas as Estacoes. */
202 function calculateGraph(simulation) {
203 /* Calcula o primeiro No, que e o Terminal de Saida de Producao
204 do Sistema, e seu unico Fluxo. */
205 node = simulation.Node[1] {
206 /* O No Terminal de Saida do Produto do Sistema ocorre no dia
207 em que o Ciclo termina. */
208 node.Start = node.Finish = simulation.Cycle
209
210 /* Desnecessario, pois e o valor default de inicializacao,
211 mas mantido aqui para clareza. */
212 node.Duration = 0
213 node.IsCalculated = true
214
215 /* O No Terminal de Saida de Producao de Sistema tem um so
216 Fluxo com Qtde = 1. */
217 node.Flow[1].Qty = 1.0
218 node.Flow[1].Day = node.Start
219 node.Flow[1].IsCalculated = true
220
221 /* Calcula recursivamente o Subgrafo que se inicia no No
222 seguinte. */
223 calculateSubgraph(node.Flow[1].Next)
224 }
225 return
226 }
227
228 /* A funcao aqui definida calcula recursivamente o Subgrafo que
229 começa no No 'node'. Este algoritmo funciona apenas para
```

```

230 Subgrafos orientados aciclicos com apenas um No sem predecessor. */
231 function calculateSubgraph(node) {
232   for flow in node.Flow[:] {
233     /* Se algum dos Fluxos predecessores do No nao tiver sido
234     calculado, sai da funcao e espera uma nova passagem pelo
235     No em uma nova interacao. */
236     if flow.Next = node and not flow.IsCalculated {
237       return
238     }
239   }
240
241   /* Calcula o No e seus Fluxos sucessores. */
242   calculateNode(node)
243
244   /* Calcula os Subgrafos que se iniciam nos Nos seguintes. */
245   for flow in node.Flow[:] {
246     if flow.Previous = node {
247       calculateSubgraph(flow.Next)
248     }
249   }
250   return
251 }
252
253 /* Esta funcao define a funcao que sera usada para calcular o No
254 e seus Fluxos sucessores a partir dos Fluxos predecessores
255 ("de referencia") e de outros atributos do No e seus Fluxos. */
256 function calculateNode(node) {
257   if node.IsProduction calculateProduction(node) {...}
258   if node.IsTreatment calculateTreatment(node) {...}
259   if node.IsReuse calculateReuse(node) {...}
260   if node.IsTerminal calculateTerminal(node) {...}
261   if node.IsMaintenance calculateMaintenance(node) {...}
262   return
263 }
264
265 /* Esta funcao calcula o No de Producao e seus Fluxos sucessores. */
266 function calculateProduction(node) {
267   /* O Fluxo de referencia e o unico Fluxo de Saida de Producao,
268   e e o primeiro Fluxo, por construcao. */
269   referenceFlow = node.Flow[1]
270
271   /* O Termino da Estacao ocorre um dia antes do Fluxo de Saida

```

```
272  ocorrer. */
273  node.Finish = referenceFlow.Day - 1
274
275  /* A Duracao inclui os dias de Inicio e Terminio da Estacao. */
276  node.Start = node.Finish - node.Duration + 1
277  node.IsCalculated = true
278  for flow in node.Flow[:] {
279    if flow.IsInput {
280      flow.Day = node.Start
281    }
282    elseif flow.IsOutput {
283      flow.Day = node.Finish + 1
284    }
285    flow.Qty = referenceFlow.Qty * flow.Coefficient
286    flow.IsCalculated = true
287  }
288  return
289 }
290
291 /* Define funcao que calcula No de Tratamento e seus Fluxos
292 sucessores. */
293 function calculateTreatment(node) {
294   /* O Fluxo de referencia e o unico Fluxo de Entrada de Tratamento,
295   e e o primeiro Fluxo, por construcao. */
296   referenceFlow = node.Flow[1]
297   /* O Inicio da Estacao ocorre no mesmo dia em que ocorre o Fluxo
298   de Entrada. */
299   node.Start = referenceFlow.Day
300   /* A Duracao inclui os dias de Inicio e Terminio da Estacao. */
301   node.Finish = node.Start + node.Duration - 1
302   node.IsCalculated = true
303   for flow in node.Flow[:] {
304     if flow.IsInput {
305       flow.Day = node.Start
306     }
307     elseif flow.IsOutput {
308       flow.Day = node.Finish + 1
309     }
310     flow.Qty = referenceFlow.Qty * flow.Coefficient
311     flow.IsCalculated = true
312   }
313   return
```

```
314 }
315
316 /* Esta funcao calcula No de Reuso e seus Fluxos sucessores. */
317 function calculateReuse(node) {
318     node.Start = node.Treatment.Input.Flow[1].Day
319     node.Finish = node.Production.Output.Flow[1].Day - 1
320     node.IsCalculated = true
321
322     /* Se Dias dos Fluxos Predecessores coincidirem , faz o balanço
323     para reuso dos Fluxos. */
324     if node.Finish % simulation.Cycle = node.Start % simulation.Cycle {
325         balance = node.Treatment.Input.Flow[1].Qty
326                 - node.Production.Output.Flow[1].Qty
327         if balance < 0 {
328             node.Production.Input.Flow[1].Qty = - balance
329             node.Treatment.Output.Flow[1].Qty = 0
330         }
331         elseif balance > 0 {
332             node.Production.Input.Flow[1].Qty = 0
333             node.Treatment.Output.Flow[1].Qty = balance
334         }
335         else {
336             node.Production.Input.Flow[1].Qty = 0
337             node.Treatment.Output.Flow[1].Qty = 0
338         }
339     }
340
341     /* Se os Fluxos Predecessores não coincidirem , não há o Reuso ,
342     e cada Fluxo segue sem ser afetado. */
343     else {
344         node.Production.Input.Flow[1].Qty
345             = node.Production.Output.Flow[1].Qty
346         node.Treatment.Output.Flow[1].Qty
347             = node.Treatment.Input.Flow[1].Qty
348     }
349
350     node.Production.Input.Flow[1].Day
351         = node.Production.Output.Flow[1].Day
352     node.Treatment.Output.Flow[1].Day
353         = node.Treatment.Input.Flow[1].Day
354     node.Production.Input.Flow[1].IsCalculated = true
355     node.Treatment.Output.Flow[1].IsCalculated = true
```

```

356     return
357 }
358
359 /* Define funcao que calcula No Terminal. */
360 function calculateTerminal(node) {
361     node.Start = node.Finish = node.Flow[1].Day
362     nodeDuration = 0
363     node.IsCalculated = true
364     return
365 }
366
367 /* Define funcao que calcula No de Manutencao. */
368 /* Estacoes de Manutencao tem a mesma Duracao que simulation.Cycle,
369 mas recebem os Fluxos de todos os Ciclos de producao concomitantes. */
370 function calculateMaintenance(node) {
371     node.Start = 1
372     node.Finish = simulation.Cycle
373     node.Duration = simulation.Cycle
374     node.IsCalculated = true
375     return
376 }
377
378 /* Define a funcao que calcula o Estoque dia a dia de cada Recurso
379 em cada Estacao, a partir das Qtde e Dias de cada Fluxo da Estacao. */
380 function calculateStock(simulation) {
381     /* Percorre cada Fluxo da Simulacao uma vez, alocando as Qtde
382     nas Estacoes de Origem ("From") e Destino ("To") de cada Fluxo.
383     Este calculo tambem seria ser feito dentro de "calculationNode()",
384     e provavelmente sera, em uma proxima versao deste algoritmo. */
385     for all flow in simulation.Flow[:] {
386         /* Acumula nas Estacoes as qtde dos Fluxos que entram. */
387         node = flow.To
388         if node.IsProduction or node.IsTreatment {
389             for day in node.Start :
390                 (node.Finish - int(node.Duration * node.Permanence) + 1) {
391                 /* Repete flow.Qty em cada dia da Permanencia dos Recursos
392                 de cada Fluxo de Entrada, mas ja trasportando o Dia do Fluxo
393                 para um dia do periodo de simulation.Cycle. O "+ 1" indica
394                 que o arredondamento de "node.Duration * node.Permanence"
395                 favorece a contagem dos Recursos que entram. */
396                 simulation.Stock(flow.Resource, node, day % simulation.Cycle)
397                 += flow.Qty

```

```
398     }
399   }
400
401   /* Nos de Reuso e Terminais nao sao considerados na contagem. */
402   elseif node.IsReuse or node.IsTerminal {
403     next loop
404   }
405   elseif node.IsMaintenance {
406     ...
407   }
408
409   /* Acumula nas Estacoes as qtdes dos Fluxos que saem. */
410   node = flow.From
411   if node.IsProduction or node.IsTreatment {
412     for day in (node.Start + int(node.Duration * node.Permanence))
413       : node.Finish {
414       /* Repete flow.Qty em cada dia da Permanencia dos Recursos de
415       cada Fluxo de Entrada, mas ja trasportando o Dia do Fluxo para
416       um dia do periodo de simulation.Cicle. */
417       simulation.Stock(flow.Resource, node, day % simulation.Cycle)
418         += flow.Qty
419     }
420   }
421   /* Nos de Reuso e Terminais nao sao considerados na contagem. */
422   elseif node.IsReuse or node.IsTerminal {
423     next loop
424   }
425   elseif node.IsMaintenance {
426     ...
427   }
428 }
```