

SimulaMM: aplicação *desktop* para análise bromatológica na produção de ração

Clodoaldo Aparecido Peternella¹, Michele Cristiani Barion²

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas – Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP) – *Campus Hortolândia* – São Paulo – SP – Brasil

kroduzz@gmail.com¹, michele_barion@hotmail.com²

Abstract. *The Inline NIR spectrophotometry equipment is composed of sensors that stay on a conveyor belt in the feed production line and are responsible for bromatological analysis, in real time, of the ingredients that pass through there. These analyses have a very high variability - on the order of 2 to 3 samples per minute - if considered individually. For this reason a statistical tool has to be used to better represent these points. This statistic has to be adjusted for each particular case and various parameters are modified in the application to best fit each case. As the parameters are changed, new simulations are performed by the algorithm and presented by means of scatter plots. With the result, the program enables technical analysis in the optimization of feed formulas, from storage to data interpretation.*

Resumo. *O equipamento de espectrofotometria NIR Inline é composto por sensores que ficam em uma esteira na linha de produção de ração e são responsáveis por analisar bromatologicamente, em tempo real, os ingredientes que ali passam. Estas análises têm uma variabilidade muito grande – da ordem de 2 a 3 amostras por minuto - se consideradas individualmente. Por este motivo há que se lançar mão de uma ferramenta estatística para representar melhor estes pontos. Esta estatística tem que ser ajustada para cada caso em particular e vários parâmetros são modificados na aplicação para ajustar da melhor forma possível caso a caso. Com a mudança dos parâmetros, novas simulações são realizadas pelo algoritmo e apresentadas por meio de gráficos de dispersão. Com o resultado, o programa possibilita análises técnicas na otimização das fórmulas de ração, desde o armazenamento, até a interpretação dos dados.*

1. Introdução

Em um processo laboratorial são colhidas amostras para análise bromatológica do alimento. A palavra bromatologia deriva do grego *broma* + *bromatos* = “dos alimentos”, além do *logos* = “ciência”. Segundo [Salman *et. al* 2010], a análise bromatológica é o processo pelo qual se verifica a composição de cada ingrediente/componente a ser usado para o desenvolvimento da ração, considerando o seu valor alimentício e calórico, propriedades físicas, químicas e toxicológicas. Cada ingrediente composto na fórmula, possui suas qualidades nutricionais, tendo na sua classificação informações sobre os componentes do alimento. A Figura 1 exemplifica a composição do grão de soja.

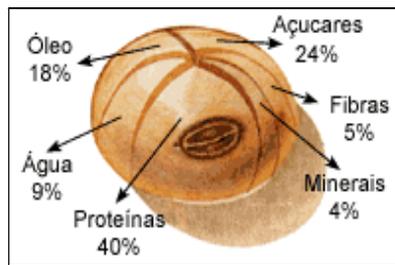


Figura 1. Composição de um grão de soja

Utilizando o método de Weende, proposto por Henneberg em 1894, com base nos resultados de investigações realizadas na Estação Experimental de Weende na Alemanha [MACIEL 2015 E SALMAN *et. al* 2010], os componentes do alimento são separados em frações compostas por: umidade, matéria seca, cinzas ou matéria mineral, proteína bruta, gordura bruta, fibra bruta e os não nitrogenados, também chamado de extrativos não azotados, conforme representado no modelo da Figura 2.

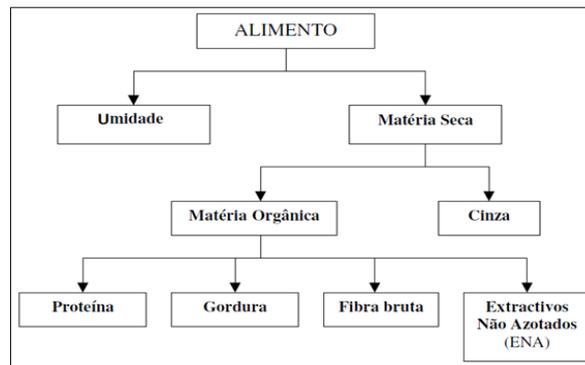


Figura 2. Método Weende proposto por Henneberg
Figura adaptada de [Teixeira 2003]

A partir dessas análises e dados obtidos, são definidos parâmetros para uma leitura padrão, assim criando uma receita base a ser seguida em escala de produção.

Estando a fórmula disponível na linha de produção, o ingrediente é analisado com equipamento de espectrofotometria de luz na faixa do infravermelho próximo ou *Near-Infrared Reflectance (Near Infra Red inline)*, que se encontra na esteira de produção. A Figura 3 traz a representação do *NIR*¹.

¹ Para apresentação do *NIR* em funcionamento, acesse ao vídeo através do link <https://drive.google.com/file/d/14bRJD-NroIbt7CfTcbrkua8Pe2aoFiS/view?usp=sharing>.

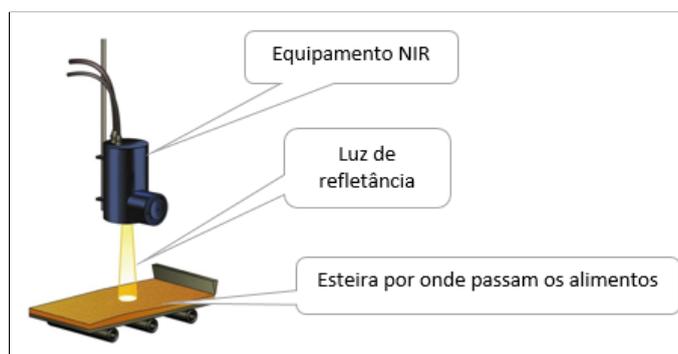


Figura 3. *Near-Infrared Reflectance (NIR)*²

Conforme apresenta [Castanheira 2020], os resultados destas análises geram uma nuvem de pontos para cada dupla ingrediente/nutriente, que são armazenados numa base de dados relacional. Com a nuvem de pontos, um gráfico de dispersão é gerado. Na estatística, a medida de dispersão serve para verificar quanto os valores encontrados estão dispersos ou afastados em relação à média.

Diante do exposto e através de uma base que armazena dados sobre os nutrientes de cada ingrediente que compõe a produção de ração, a proposta é criar uma aplicação *desktop* com os seguintes objetivos:

a. Plotagem gráfica dos dados resultantes da análise bromatológica, apresentando uma a nuvem de pontos para cada dupla ingrediente/nutriente. Ressalta-se que esse armazenamento de dados se fez através da leitura de cada nutriente por meio do equipamento *NIR inline*.

b. Geração de uma nova média móvel, por meio de um algoritmo iterativo, para uma melhor representação da nuvem de pontos das análises.

Para abordagem da proposta apresentada, além dessa introdução, o presente trabalho está organizado em seis seções ordenadas em: fundamentação teórica, metodologia, desenvolvimento, conclusões, trabalhos futuros e relação das referências utilizadas tanto na pesquisa sobre as metodologias adotadas, quanto das ferramentas que foram fundamentais para o desenvolvimento.

2. Referencial Teórico

Esta seção apresentará os principais conceitos e definições que fundamentam os aspectos de negócios e técnicos utilizados no desenvolvimento deste trabalho.

2.1. Espectrofotometria

A espectrofotometria é um método que estuda a interação da luz com a matéria e a partir desse princípio permite a realização de diversas análises. Cada composto químico absorve, transmite ou reflete luz ao longo de um determinado intervalo de comprimento de onda. A espectrofotometria pode ser utilizada para identificar e quantificar substâncias químicas a

² Imagem extraída do link https://finnasensors.com/wp-content/uploads/2020/04/SMARTIII_Illustration.jpg.

partir da medição da absorção e transmissão de luz que passa através da amostra [KASVI 2016].

2.2. Funcionamento NIR

O equipamento de espectrofotometria usa a radiação eletromagnética do infravermelho próximo - NIR (do inglês, *Near Infrared*) trabalhando com o comprimento e a frequência de ondas efetuando as análises das amostras. Atualmente a técnica tem sido aplicada para a determinação de proteína, umidade, amido, óleos, lipídeos e celulose em produtos provenientes da agricultura, tais como, grãos e óleos vegetais [FIDÊNCIO 2001].

2.3. Desenvolvimento Incremental

Segundo [Sommerville 2011], o desenvolvimento incremental se baseia na ideia de desenvolver uma implementação inicial, obter *feedback* do usuário ou terceiro e fazer a evolução do *software* através de várias versões até o sistema desejado. A Figura 4 ilustra as etapas do desenvolvimento incremental.

A iteração do sistema, permite que partes sejam testadas e refinadas mesmo que incompletas, e com isso, o cliente possa avaliar o sistema em um estágio inicial do desenvolvimento.

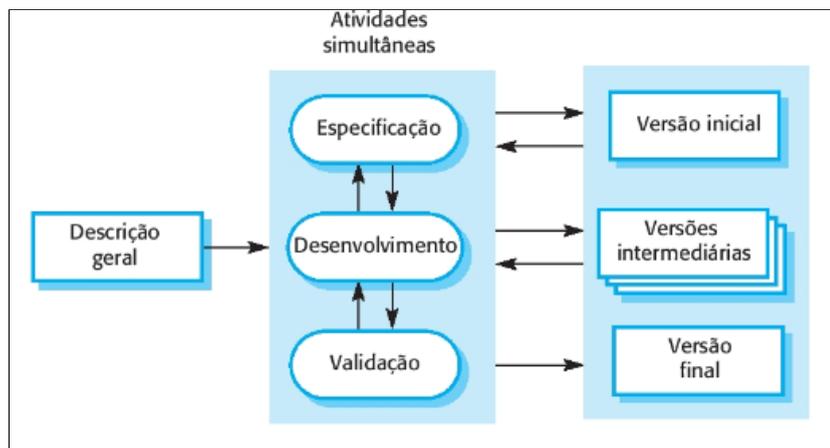


Figura 4. Desenvolvimento Incremental [SOMMERVILLE 2011]

2.4. Requisitos funcionais e não funcionais

Como apresenta [Sommerville 2011], os requisitos de um sistema são as descrições dos serviços que o sistema deve prestar e as restrições de sua operação. Esses requisitos refletem as necessidades dos clientes de um sistema que atende um determinado propósito, como controlar um dispositivo, fazer um pedido ou encontrar informações.

Os requisitos são classificados como funcionais e não funcionais. Os requisitos funcionais devem descrever o que o *software* deve fazer em detalhes para as funções do sistema. A apresentação dos requisitos funcionais, pode ser documentada e modelada utilizando diagramas em *UML* (*Unified Modeling Language*, ou seja, Linguagem de Modelagem Unificada). Já os requisitos não funcionais estão relacionados ao obter uma

resposta rápida do usuário, tempo de atualização de tela, dentre outras métricas quantitativas e restrições do sistema [SOMMERVILLE 2011].

Neste trabalho serão apresentados os diagramas de casos de uso e de atividade. Conforme destaca [Guedes 2009]:

- Diagrama de casos de uso é utilizado nas fases de levantamento e análise de requisitos do sistema, apresentando os atores e as suas funcionalidades. “Apresenta uma linguagem simples e de fácil compreensão para que os usuários possam ter uma ideia geral de como o sistema irá se comportar”.
- Diagrama de atividade tem como finalidade a descrição dos passos a serem percorridos para conclusão de uma atividade específica. “Pode representar um método com certo grau de complexidade, um algoritmo ou mesmo um processo completo”.

2.5. *StarUML*

A plataforma [*StarUML* 2021] é um projeto que foi renomeado e lançado em 2005 se tornando código aberto. A ferramenta dá suporte na modelagem *UML*, oferecendo vários tipos de diagramas, além da possibilidade de gerar o código fonte por meio dos diagramas criados para diversas linguagens de programação, tais como o *Delphi*.

2.6. Programação Orientada a Objetos

Como apresenta [Félix 2016] em sua obra, a programação orientada a objetos, também conhecida por sua sigla POO, facilita a relação da máquina com o mundo real. Isso porque qualquer coisa do mundo físico pode ser ter um representante na esfera digital, como por exemplo, uma pessoa, um emprego ou um objeto. “Essas ‘coisas’ são o que chamamos de objetos na POO”.

Aplicam-se os conceitos de classe, objeto, atributo, encapsulamento, herança e polimorfismo, como são descritos abaixo [FÉLIX 2016].

- Classe, objeto e atributos: são três conceitos que estão relacionados, sendo: objetos correspondem a elementos da vida real onde possuem seus atributos, ou seja, suas características. A classe é um agrupamento dos objetos, considerando suas características e operações específicas. Exemplo: A classe aluno tendo como atributos prontuário, nome, telefone e email, além das operações como ‘fazer matrículas’, ‘enviar mensagens’. Os alunos que fazem parte da classe, se tornam os objetos.
- Encapsulamento: é um mecanismo interno do objeto ‘escondido’ do usuário. É “restringir a visibilidade de informações e detalhes de implementações [...]”. É definir quais objetos devem ficar visíveis e quais necessitam ficar ocultos”.
- Herança: um objeto novo nem sempre é criado do zero. Ele pode ‘herdar’ atributos e ações de outros já existentes. Por exemplo, o professor ou o aluno ‘herdam’ características e ações da classe pessoa. “A herança está ligada à hierarquização de classes, simplificando e evitando a criação de código desnecessário”.
- Polimorfismo: é um recurso das linguagens orientadas a objetos, que literalmente indica a capacidade de um objeto assumir várias formas. Em outras palavras, o polimorfismo

permite que referencie propriedades e métodos de classes diferentes por meio de um mesmo objeto, por exemplo, um comando ‘abre’, faz um objeto entrar em ação, seja ele uma janela, uma porta ou uma tampa de garrafa. Várias classes podem implementar a mesma operação de maneira diferente. A mesma operação (com o mesmo nome) tem várias ("poli") formas ("morfismo") em cada classe. "Permite que objetos de classes inferiores na hierarquia atuem como elementos das classes mais altas".

Como um exemplo de linguagem de programação que fundamenta o paradigma da POO é a linguagem *Delphi*.

2.7. *Delphi*

O *Delphi* é uma ferramenta *RAD* (*Rapid Application Development* – Desenvolvimento Rápido de Aplicações), sendo criada pela *Borland*. É extensível e sua *IDE* (*Integrated Development Environment* – Ambiente de Desenvolvimento Integrado) pode ser ampliada e personalizada com a adição de componentes e ferramentas criadas utilizando-se o *Object Pascal* [CANTÚ 2003]. Desde 2006 a empresa Embarcadero se tornou proprietária da licença de utilização da linguagem *Delphi*. Para o desenvolvimento desta proposta utiliza-se a versão *Delphi XE2*, lançada em 2011.

O *Object Pascal* está associado ao programa POO, que além de possuir as características tradicionais das mesmas como classes e objetos, também possui interfaces (semelhantes às encontradas em *COM* e *Java*), tratamento de exceção, programação *multithreaded* e algumas características não encontradas nem mesmo em C++, como *RTTI* (*Run Time Type Information*). Assim como o C++, o *Object Pascal* é uma linguagem híbrida, pois além da orientação a objeto possui também uma parte da antiga linguagem estruturada [CANTÚ 2003].

2.8. Linguagem *SQL*

A linguagem padrão para banco de dados relacional é a *SQL* (*Structured Query Language* - Linguagem Estruturada para Consulta), a qual segue a padronização *ANSI* (*American National Standards Institute*).

Existem diversos tipos de Sistemas Gerenciadores de Banco de Dados Relacionais que possibilitam o uso da linguagem *SQL* e que também oferecem recursos para administração e manutenção do banco de dados. Algumas características do modelo relacional é: evitar redundâncias nas inserções, compartilhamento dos campos entre tabelas, relacionamento entre tabelas através de campos do tipo chave, além da facilidade de executar buscas e pesquisas através das tuplas inseridas nas respectivas tabelas que compõem um *database* [PUGA *et. al* 2013].

Nesta aplicação será utilizado o *SQL Server 2019 Express*, que é uma versão gratuita do *SQL Server*. O mercado oferece também opções de bancos de dados *open source*³, tais como: MySQL, PostgreSQL e MariaDB.

³ *Open Source* é um termo que significa “código aberto” e se refere ao código-fonte de um programa/site/aplicativo que pode ser adaptado aos interesses específicos do programador.

2.9. Média móvel simples

Como apresenta o autor [Bonafini 2015], a média móvel simples (MMS) é um conceito aplicado na área da estatística, resultante da soma dos valores numéricos, dividida pela quantidade de elementos somados. A Figura 5 ilustra a fórmula utilizada na composição da média móvel simples.

$$MMS_n = \frac{(y_1 + y_2 + y_3 + \dots + y_n)}{n}$$

Figura 5. Fórmula do cálculo da média móvel [BONAFINI 2015]

2.9.1. Gráfico de dispersão

Conforme os autores [Farber e Larson 2009], os dados de uma correlação entre pares ordenados, (x,y), colocados em um plano coordenado como pontos, é chamado de gráfico de dispersão. Este tipo de gráfico é utilizado para pontuar em um eixo vertical e horizontal quanto uma variável é afetada por outra. A Figura 6 ilustra um gráfico de dispersão.

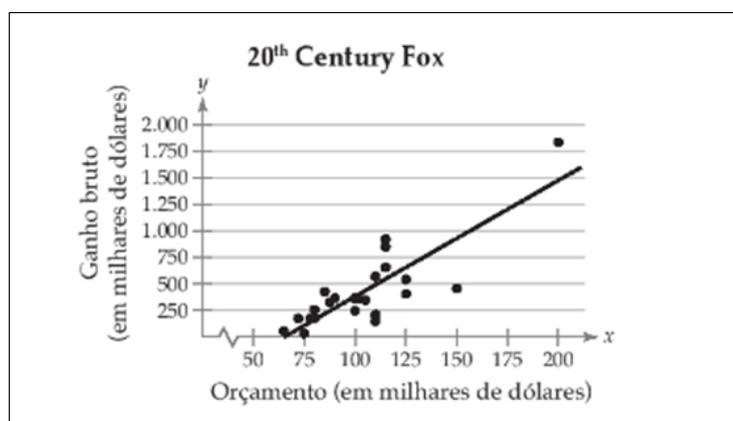


Figura 6. Gráfico de dispersão [FARBER E LARSON 2009]

2.9.2. Método de Identificação de Outliers via Distância de Mahalanobis

O fator *MDI* (*Mahalanobis Distance Index*), usa a distância de Mahalanobis (MD) como um método para detectar *outliers* em dados multivariados. Observando as médias amostrais, os pontos que têm uma distância maior do que o resto da nuvem de pontos, são considerados *outliers*, conforme é observado na Figura 7. Como apresentam os autores [Filzmoser *et al.* 2020], esses pontos mais distantes precisam ser descartados para que não interfiram no cálculo das médias móveis.

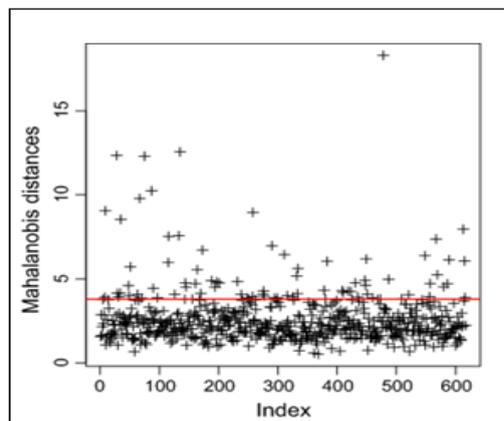


Figura 7. Amostras de leitura com a presença de *outliers* [Filzmoser et al. 2020]⁴

3. Metodologia e Materiais

A aplicação proposta utiliza a captação dos dados obtidos através do equipamento *NIR* e alocados no banco de dados relacional. A coleta desses dados traz uma análise quantitativa das médias móveis e qualitativa para os pontos das duplas ingredientes/nutrientes.

Além de ser um trabalho aplicado, a pesquisa bibliográfica tendo como fontes, livros, apostilas e manuais, foi fundamental para gerar o embasamento teórico quanto às ferramentas para desenvolvimento, além do entendimento do *NIR* para leitura, armazenamento e uso de cada dado gerado para composição da saída de resultados, sendo esta leitura o requisito não-funcional que fundamenta o início do levantamento de dados para compor a proposta.

O modelo incremental utilizado dividiu cada iteração das partes do sistema com suas funcionalidades, respeitando os padrões *UML*.

O projeto é voltado para o ambiente *desktop* e faz uso da linguagem *Delphi*, que utiliza a orientação a objetos.

O gerenciamento do projeto utiliza um controle de versão administrado pelo *TortoiseSVN*, que é um cliente *open source* para *Windows* e armazena em um repositório na nuvem, criado através do OneDrive em uma pasta na conta do desenvolvedor.

4. Desenvolvimento

Esta seção retrata as fases do desenvolvimento do projeto como um todo, sendo elas: estrutura do sistema, levantamento e documentação dos requisitos, modelo de dados do sistema e implementação da proposta.

4.1. Diagrama Caso de Uso

As funcionalidades apresentadas na Figura 8, representam as ações que o usuário pode executar no programa SimulaMM.

A Tabela 1 apresenta uma explicação das principais funcionalidades da proposta, considerando a abordagem na Figura 8.

⁴ Imagem adaptada do *link* <https://link.springer.com/article/10.1007/s11004-020-09861-6/figures/2>.

Tabela 1. Abordagem das funcionalidades da proposta

FUNCIONALIDADE	AÇÃO
Selecionar período	Escolha de períodos específicos com detalhamento para ano, dia, mês e horas a serem mostrados.
Selecionar ingrediente/nutriente	Seleção para o ingrediente e nutriente desejado, atualizando a visualização automaticamente dos pontos no gráfico após a escolha.
Alterar média móvel	Apresenta uma tela na qual o usuário visualiza o valor padrão mínimo e máximo de cada nutriente e poderá editar novos valores, se assim o desejar, gerando a nova média móvel para os nutrientes relacionados ao ingrediente selecionado. Os dados alterados serão visualizados junto à média móvel anterior e só serão modificados oficialmente na base, caso o usuário aceite as alterações.
Oficializar parâmetros	Através de um comando <i>SQL</i> , insere os valores das tabelas “temp” nas tabelas oficiais.

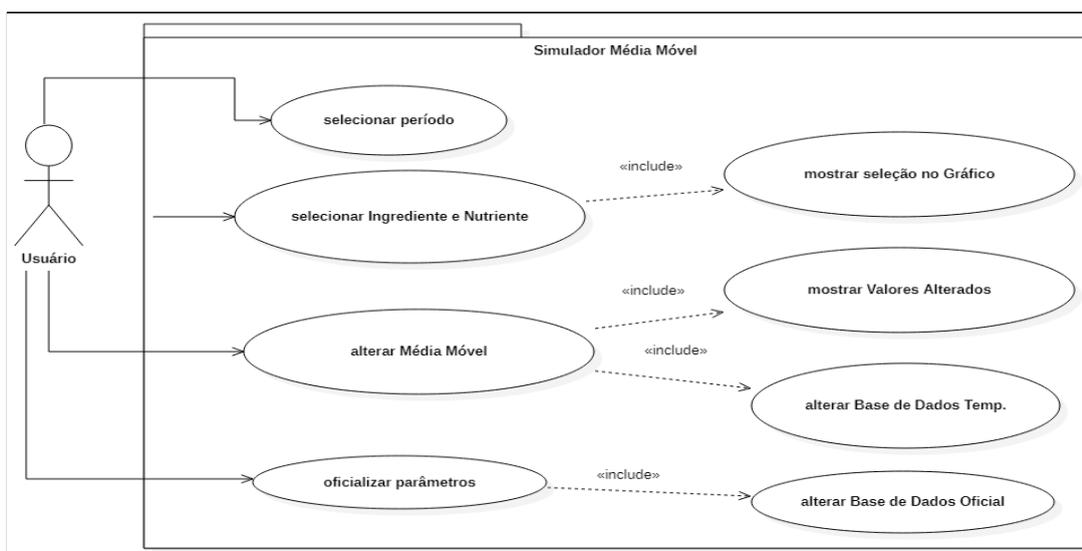


Figura 8. Diagrama de caso de uso

4.2. Diagrama de Atividade

Um dos modelos de um fluxograma encontrados na *UML* é o diagrama de atividades, que conforme a Figura 9, descreve as etapas do sistema e esclarece as funcionalidades do caso de uso descrito anteriormente. O processo se inicia com a coleta dos dados, posteriormente esses dados são armazenados na base e visualizados através do programa SimulaMM.

Com as informações disponíveis, o usuário contempla em tempo real as análises obtidas da leitura *NIR* e caso queira, poderá ajustar os parâmetros das médias móveis observadas.

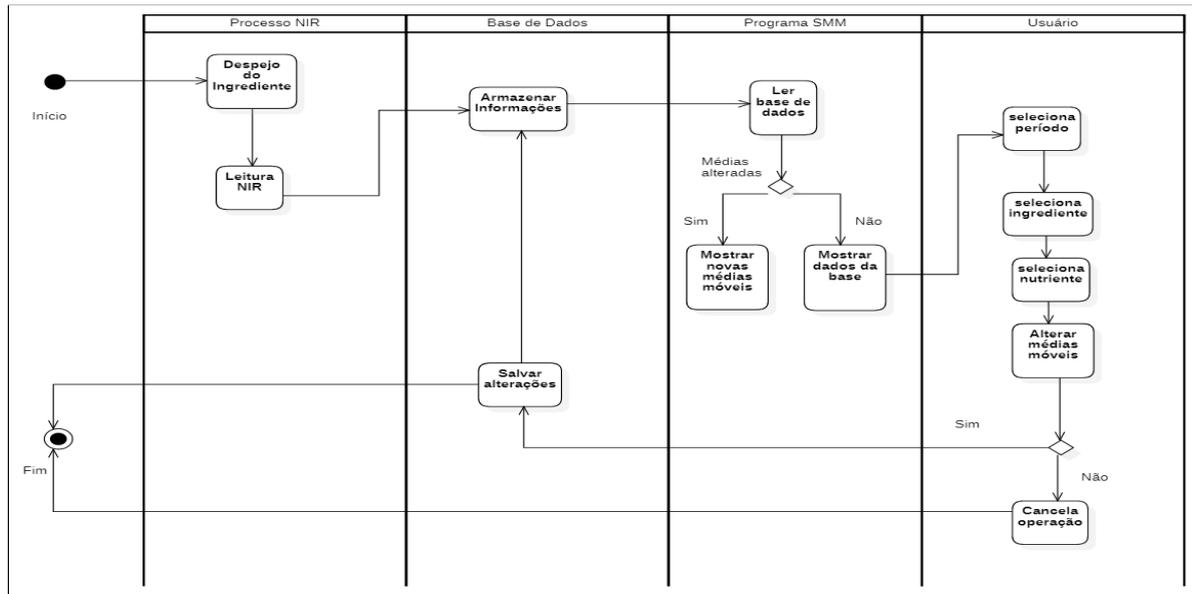


Figura 9. Diagrama de Atividade

4.3. Diagrama Entidade-Relacionamento

O diagrama da Figura 10, representa a estrutura dos dados que são gerados pela leitura do equipamento *NIR* e depositados na base de dados em um modelo relacional.

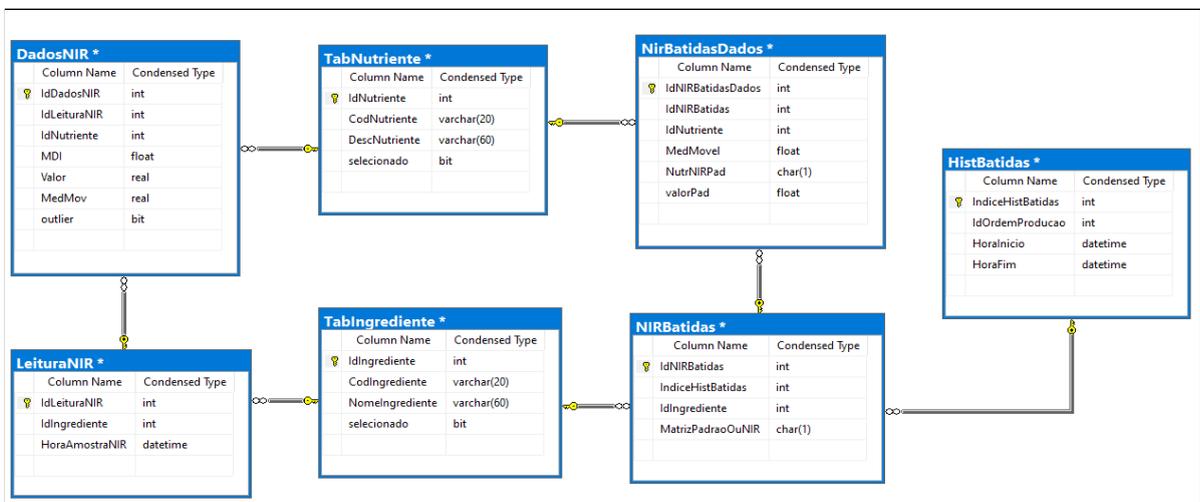


Figura 10. DER responsável pelo armazenamento dos dados através da leitura pelo *NIR*

Já a Figura 11, representa as tabelas dos valores mínimos, máximos e o fator MDI. As tabelas descritas são espelhadas para que a simulação dos parâmetros alterados, a princípio, sejam visualizados temporariamente até que se conclua as análises, e por fim, validar ou não os valores. As tabelas espelho são diferenciadas pelo mesmo nome com adicional “trab” ao final, e o propósito é criar o cenário da simulação colocando os valores editados junto aos valores oficiais da produção na tela do SimulaMM.

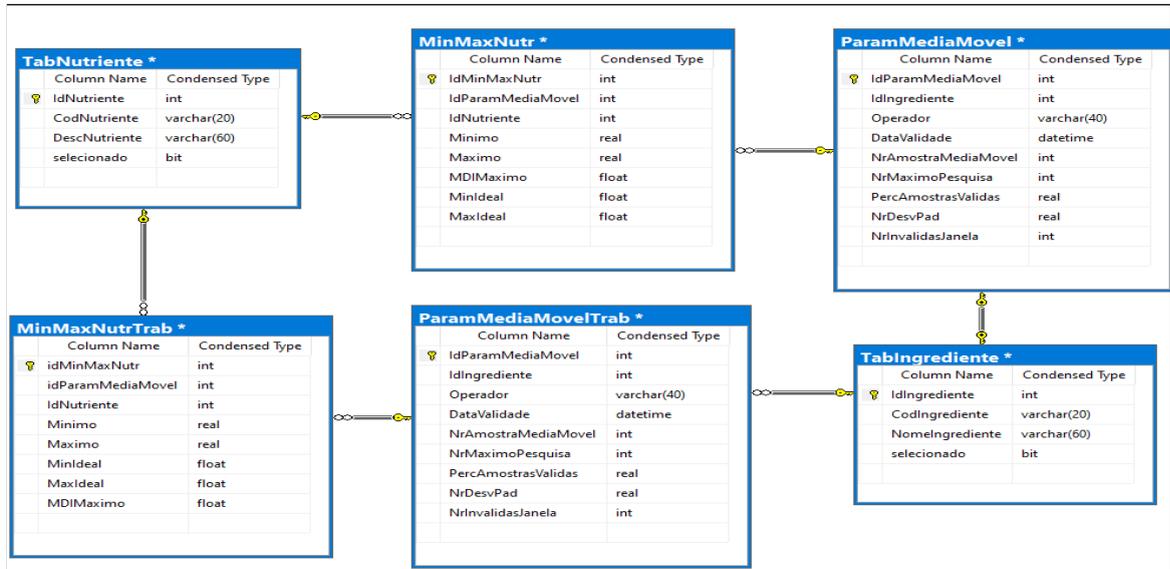


Figura 11. DER usado para armazenamento dos dados para gerar as médias móveis

4.4. Desenvolvimento das Funcionalidades

A escala evolutiva do programa seguiu o modelo incremental, considerando o projeto, a implementação e os testes, até a apresentação dos resultados gerados com a conclusão do produto.

A primeira funcionalidade a ser apresentada é a Figura 12. Demonstra uma simulação dos dados da seleção do ingrediente soja, com o nutriente extrato etéreo (Figura 12.2), no período de 13/05/2019 a 14/05/2019 (Figura 12.1). Diante destes dados parametrizados, cria-se o gráfico de dispersão dos pontos com as médias móveis (Figura 12.4) e o histograma correspondente (Figura 12.5). A Figura 12 pode ser definida como funcionalidade principal, pois é através dela que as demais opções da aplicação serão chamadas e executadas.

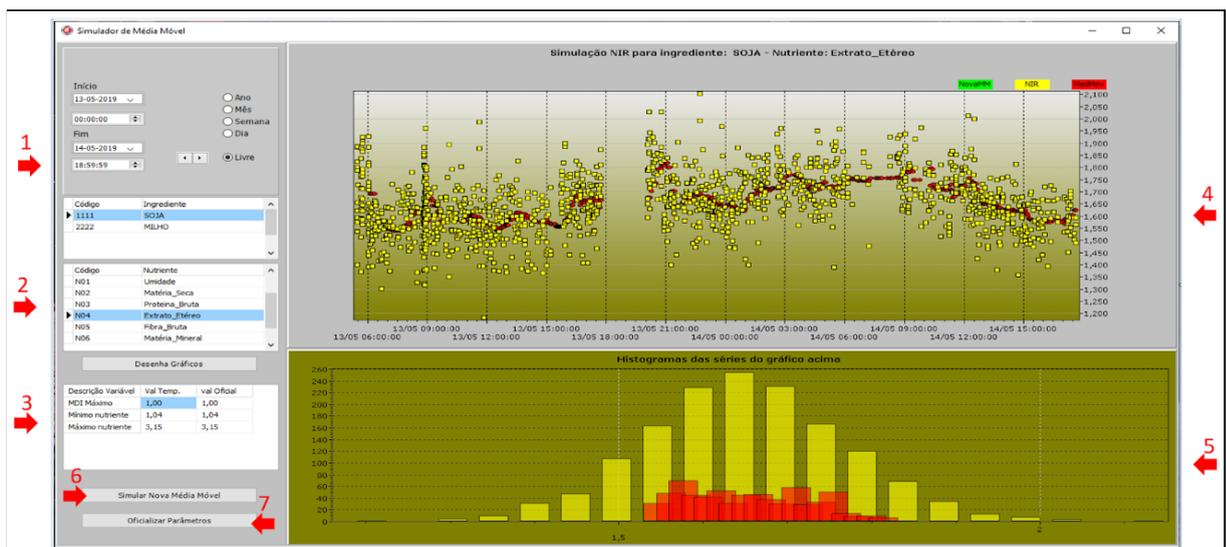
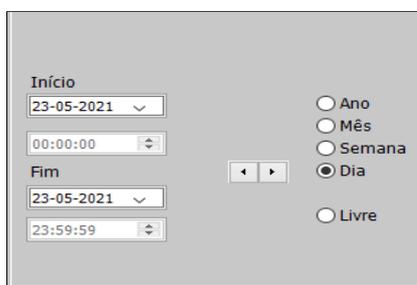


Figura 12. Tela principal do SimulaMM

A funcionalidade “selecionar período”, como é representada na Figura 13 (também pode ser observada na Figura 12.1), traz dinamismo, possibilitando ao usuário escolher livremente qual data deseja observar os dados.



Formulário de definição de período para gerar os dados no gráfico de dispersão. O formulário contém campos para 'Início' (data e hora) e 'Fim' (data e hora), além de opções de seleção para o período: Ano, Mês, Semana, Dia (selecionado) e Livre.

Figura 13. Definição do período para gerar os dados no gráfico de dispersão

Um objeto *DBGrid*⁵ funciona como uma planilha com linhas e colunas que retornam os componentes que estão armazenados no banco de dados de acordo com a requisição. Como é apresentado na Figura 14 (também pode ser observada na Figura 12.2), a consulta do *DBGrid* mostra os ingredientes e nutrientes existentes na base. O *DBGrid* é dinâmico e clicável, retornando um procedimento que atualiza a visualização de acordo com o componente em foco.



Código	Ingrediente
1111	SOJA
2222	MILHO

Código	Nutriente
N01	Umidade
N02	Matéria_Seca
N03	Proteína_Bruta
N04	Extrato_Etéreo
N05	Fibra_Bruta
N06	Matéria_Mineral

Figura 14. *DBGrid* de seleção de ingrediente e nutriente

Nas próximas Figuras, considerando a numeração 15 ao 20, conforme o nutriente selecionado, são apresentados os valores nutricionais acurados de acordo com a composição bromatológica (também pode ser observada na Figura 12.3). Essa composição bromatológica é exemplificada na Figura 1, sendo que os campos dos mínimos e máximos que estão representados nas próximas figuras, trazem noções de medidas do ingrediente observado. A importância de se apresentar essas medidas se dá pelo fato de que uma produção de ração deve obedecer limites aceitáveis para obter um resultado balanceado e que tenha valor nutricional adequado ao consumo. A alteração desses limites sem uma análise técnica, pode trazer prejuízo nutricional e financeiro, influenciando no resultado do crescimento e engorda do animal e um adicional de outros ingredientes para compensar o que faltou no balanceamento para uma ração de ótima qualidade⁶.

⁵ *DBGrid*: *DB* significa *DataBase* (Base de Dados); *Grid* vem de grade ou tabela com linhas e colunas.

⁶ Informações oriundas do levantamento de requisitos para entender a problemática da proposta.

Descrição Variável	Val Temp.	val Oficial
MDI Máximo	1,00	1,00
Mínimo nutriente	11,26	11,26
Máximo nutriente	13,72	13,73

Figura 15. Filtro umidade

Descrição Variável	Val Temp.	val Oficial
MDI Máximo	1,00	1,00
Mínimo nutriente	86,26	86,27
Máximo nutriente	88,73	88,74

Figura 16. Filtro matéria seca

Descrição Variável	Val Temp.	val Oficial
MDI Máximo	1,00	1,00
Mínimo nutriente	43,15	43,16
Máximo nutriente	48,13	48,13

Figura 17. Filtro proteína bruta.

Descrição Variável	Val Temp.	val Oficial
MDI Máximo	1,00	1,00
Mínimo nutriente	1,00	1,04
Máximo nutriente	2,50	3,15

Figura 18. Filtro extrato etéreo.

Descrição Variável	Val Temp.	val Oficial
MDI Máximo	1,00	1,00
Mínimo nutriente	1,50	1,56
Máximo nutriente	4,86	4,86

Figura 19. Filtro fibra bruta.

Descrição Variável	Val Temp.	val Oficial
MDI Máximo	1,00	1,00
Mínimo nutriente	5,19	5,19
Máximo nutriente	6,17	6,17

Figura 20. Filtro matéria mineral.

Exemplificando, a tela do gráfico de dispersão na Figura 21 (também pode ser observada na Figura 12.4), mostra em amarelo os pontos da leitura do nutriente extrato etéreo ou gordura bruta e os pontos em vermelho das médias móveis calculadas a partir dessa leitura. No eixo vertical direito, a escala reflete o valor mínimo e máximo nutricional que de acordo com a Figura 18 no valor oficial vai de 1,04 para o mínimo e 3,15 para o máximo dentro dos parâmetros de acurácia aferidos pelo laboratório. A escala horizontal se refere ao período em que os pontos são apresentados no gráfico. Ao centro superior, há uma descrição do ingrediente e nutriente escolhidos e no canto superior direito, 3 bandeiras nas quais as cores e o texto representam os pontos que também são botões para acionar a apresentação dos pontos, facilitando ao usuário uma visualização mais detalhada do parâmetro.

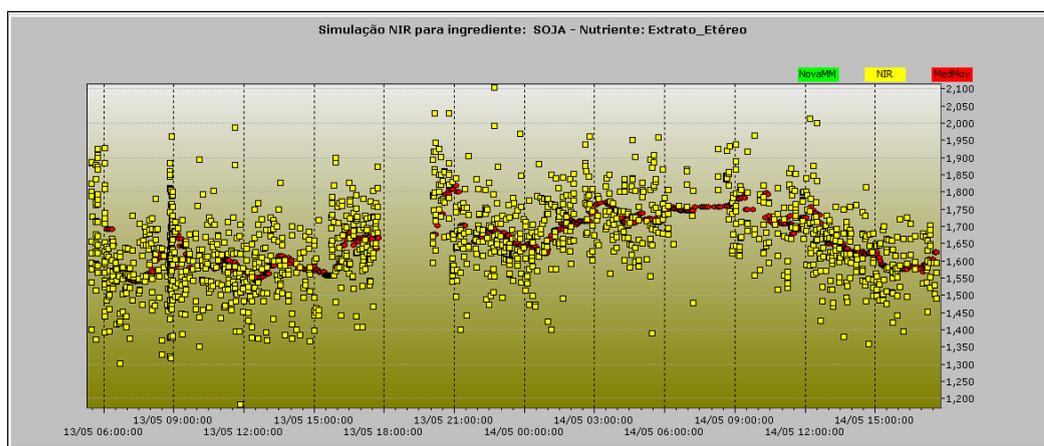


Figura 21. Gráfico de dispersão

A simulação das novas médias móveis é feita através de uma tela, conforme a Figura 22, após o usuário clicar no botão “Simular Nova Média Móvel” (o acesso desta funcionalidade pode ser observada através da Figura 10.6). Enfatiza-se que para esta simulação, os dados são depositados nas tabelas denominadas “trab”, conforme apresenta a Figura 11.

A tela mostra uma lista dos nutrientes e seus valores mínimos, máximos e o fator MDI com campos editáveis.

Nutrientes:	N01 - Umidade	N02 - Matéria_Seca	N03 - Proteína_Bruta	N04 - Extrato_Etéreo	N05 - Fibra_Bruta	N06 - Matéria_Mineral
MDI máximo	1	1	1	1	1	1
Mínimo nutriente	11,26	86,26	43,15	1	1,5	5,19
Máximo nutriente	13,72	88,73	46,13	2,5	4,86	6,17

Figura 22. Tela para alteração dos parâmetros da média móvel

Ao final das alterações concluídas, o usuário clica no botão "Simular" e uma chamada de procedimento executa um algoritmo que analisa a nuvem de pontos e com base nos cálculos para plotar os novos valores obtidos no gráfico, conforme pode ser visualizado na Figura 23 através dos pontos na cor verde.

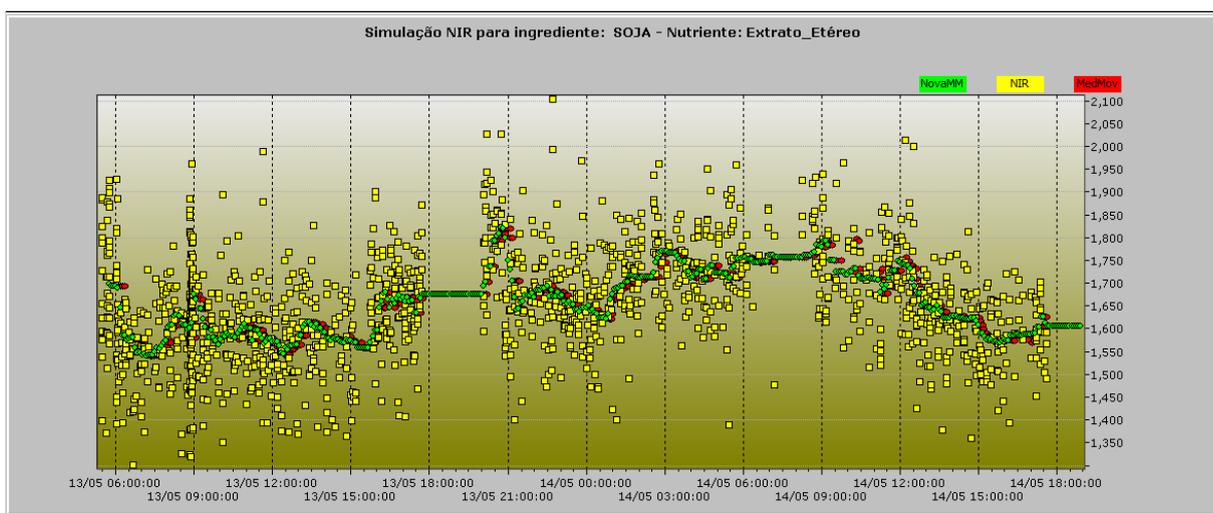


Figura 23. Gráfico de dispersão com as novas médias calculadas

A alteração dos parâmetros para novas médias móveis, pode fazer com que algumas leituras dos pontos sejam descartadas ou inseridas no gráfico. Esses limites determinam se a amostra poderá ser considerada aceitável ou não e, assim, obter uma nova simulação com critérios mais amplos ou mais restritos. A Figura 24, mostra um exemplo de descarte anterior ao limite mínimo ser diminuído.

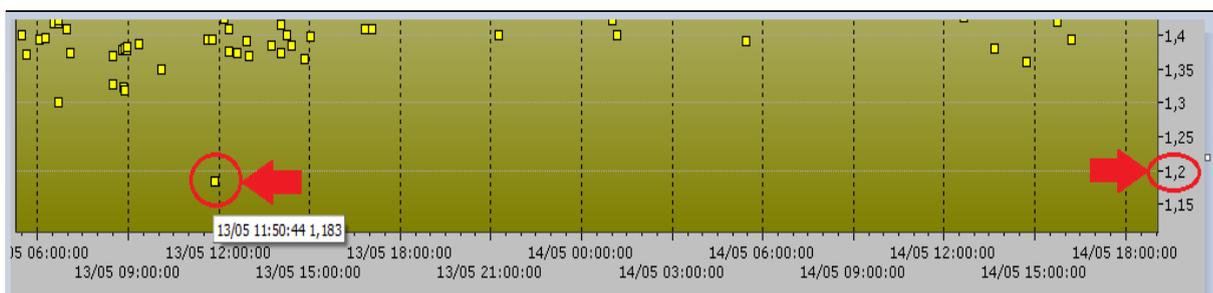


Figura 24. Observação de leitura antes da alteração do parâmetro

Logo após a obtenção das novas médias móveis, o ponto em evidência na Figura 22 foi descartado e não faz mais parte da leitura. O menor valor observado passa a ser o ponto em destaque, conforme demonstrado na Figura 25.

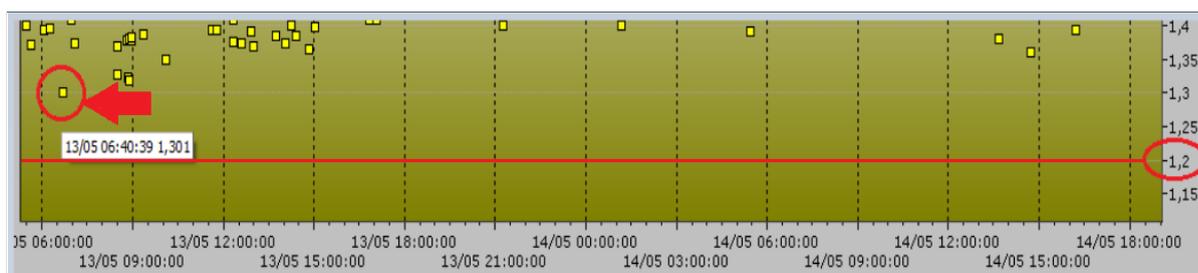


Figura 25. Observação de leitura após a alteração do parâmetro

A Figura 26 apresenta as mesmas séries visualizadas na Figura 21 mas na forma de histograma que também é um gráfico de dispersão no formato da frequência quantitativa. O eixo vertical esquerdo mostra a quantidade dos valores observados nos intervalos dos pontos.

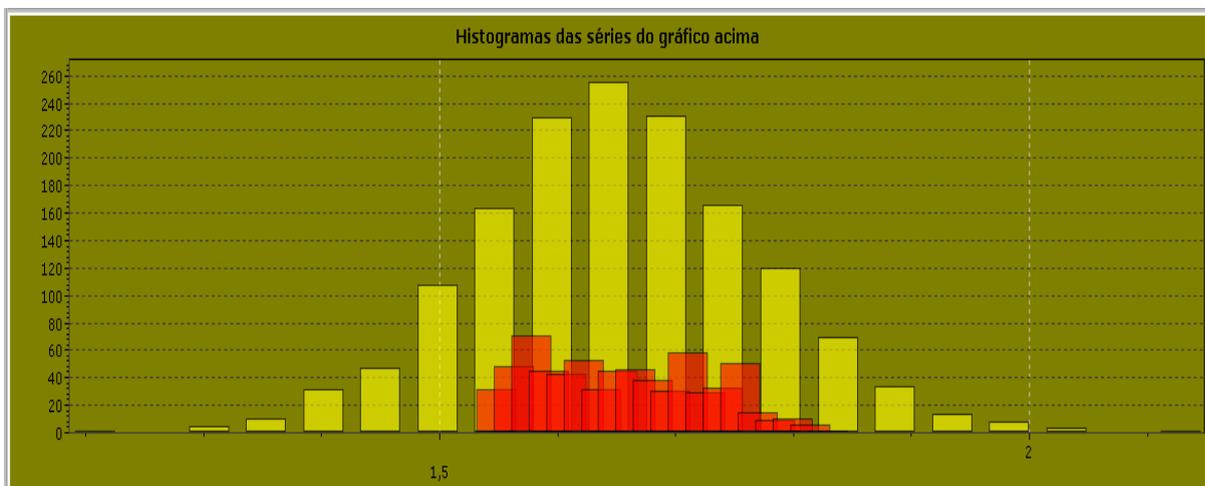


Figura 26. Histograma das séries no gráfico de dispersão

O trecho de código da Figura 27 utiliza uma matriz 3X3 para obter o código do ingrediente selecionado, os códigos dos nutrientes e as médias móveis. Na Figura 28 é possível observar em tempo de execução a obtenção desses parâmetros.

Considerando a Figura 26, na linha de código 676, um cálculo é executado a partir de uma consulta, obtendo a menor e a maior data dos pontos e as divide no intervalo de 5 minutos. Posteriormente nas linhas 685 e 686 esse resultado é usado pelas variáveis para inserir a nova média móvel nesse intervalo obtido de 5 minutos.

```

674 //1440 é a qtd de minutos = 24 horas
675 // a divisão por 5 coloca nova média móvel a cada 5 minutos
676 num:=(Trunc(MaxHora*1440.0)-Trunc(MinHora*1440.0))div 5;
677 query.Close;
678
679 for indiceNutrienteMediaMovel:=0 to numNutrientes-1 do
680 begin
681   IdNutriente:=vetorIngredientesMediaMovel[indiceIngredienteMediaMovel].VetorNutrientes[indiceNutrienteMediaMovel].IdNutriente;
682
683   SetLength(vetorIngredientesMediaMovel[indiceIngredienteMediaMovel].VetorNutrientes[indiceNutrienteMediaMovel].vetorPontosMediaMovel,
684     i:=0;
685     Hora:=(Trunc(MinHora*1440.0));
686     Hora:=Hora-(Hora mod 5);
687     while i<num do
688     begin
689       vetorIngredientesMediaMovel[indiceIngredienteMediaMovel].VetorNutrientes[indiceNutrienteMediaMovel].vetorPontosMediaMovel[i].Hr:=
690       vetorIngredientesMediaMovel[indiceIngredienteMediaMovel].VetorNutrientes[indiceNutrienteMediaMovel].vetorPontosMediaMovel[i].Hr;
691       vetorIngredientesMediaMovel[indiceIngredienteMediaMovel].VetorNutrientes[indiceNutrienteMediaMovel].vetorPontosMediaMovel[i].Mn:=
692       vetorIngredientesMediaMovel[indiceIngredienteMediaMovel].VetorNutrientes[indiceNutrienteMediaMovel].vetorPontosMediaMovel[i].Mn;
693       vetorIngredientesMediaMovel[indiceIngredienteMediaMovel].VetorNutrientes[indiceNutrienteMediaMovel].vetorPontosMediaMovel[i].Mk:=
694       Inc(i);
695       Hora:=Hora+5;
696     end;
697   setLength(vetorDadosNIR, 0);

```

Figura 27. Matriz 3X3 para obtenção do ingrediente, nutriente e médias móveis

```

for indiceNutrienteMediaMovel:=0 to numNutrientes-1 do
begin
  IdNutriente:=vetorIngredientesMediaMovel[indiceIngredienteMediaMovel].VetorNutri
  SetLength(vet
  i:=0;
  Hora:=(Trunc(MinHo
  Hora:=Hora-(Hora m
  while i<num do
  begin
    vetorIngredientes [0] (1, 'N01', 0)
    vetorIngredientes [1] (2, 'N02', 0)
    vetorIngredientes [2] (3, 'N03', 0)
    vetorIngredientes [3] (4, 'N04', 0)
    vetorIngredientes [4] (6, 'N05', 0)
  Inc(i);
  Hora:=Hora+5;
end;

```

Figura 28. Matriz 3X3 em tempo de execução

5. Resultados

Para executar os testes da ferramenta, foi utilizada uma base de dados com 222.915 registros, considerando as tabelas que contém os ingredientes e os nutrientes.

O tempo estimado para gerar as médias e o novo gráfico da simulação sofre uma variação conforme a quantidade de pontos existentes no período. O período de teste para um dia apresentou uma média de 43 a 50 segundos e um mês de 1:16 a 1:22 minutos em um computador com a seguinte configuração:

- Processador Intel(R) Core(TM) i7 CPU @ 1.80GHz/1.99 GHz
- RAM instalada 12,0 GB
- Sistema operacional de 64 bits, processador baseado em x64
- Banco de Dados *SQL Server (SQLEXPRESS 2019) Versão 15.0.18206.0*

6. Conclusão

O programa SimulaMM facilita as análises técnicas para a observação dos dados bromatológicos que, utilizando a estatística, possibilita acompanhar se há muitas variações de leituras e ajustar as médias móveis para criar um cenário de otimização da ração.

Diante do cenário estudado e do levantamento de requisitos funcionais, a ferramenta proposta atendeu as expectativas do usuário através de testes práticos com a dispersão das amostras na esteira e captação das leituras do aparelho *NIR* com a execução da ferramenta SimulaMM.

Para desenvolvimento deste trabalho e até mesmo destacando as competências e habilidades adquiridas no decorrer do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, houve aplicação de conceitos associados às disciplinas de Algoritmo e Lógica de Programação, Programação Orientada a Objetos, Banco de Dados,

Estatística, Engenharia de *Software*, além da Metodologia de Pesquisa e Projeto de Sistemas para planejamento e desenvolvimento teórico-técnico.

7. Trabalhos Futuros

Com a capacidade de se analisar a variabilidade das leituras *NIR* dos pontos obtidos, uma implementação de relatórios e alarmes trariam assistência ao técnico na ponta do processo, que tomaria ciência de eventuais anomalias dos parâmetros.

Outra funcionalidade que agregaria a esta proposta seria uma mensagem de tela apresentando o tempo de execução da simulação para que o usuário aguarde seu término.

Neste trabalho não foram documentados os testes, sendo assim, esta também pode ser apresentada como uma proposta futura, além de avaliar a qualidade do *software* desenvolvido e implantado.

Referências

Bonafini, F. C. (2015). “Probabilidade e Estatística”. Editora Pearson.

Cantú, M. (2003). “Dominando o Delphi 7 – A Bíblia”. Editora Makron Books.

Castanheira, N. P. (2020). “Matemática Aplicada à Economia”. Editora Pearson.

Farber, B., Larson, R. (2009). "Estatística Aplicada". Editora Pearson, 4ª Edição.

Félix, R. (2016). “Programação Orientada a Objetos”. Editora Pearson.

Fidêncio, P. H. “Análise de solos por espectroscopia no infravermelho próximo e aplicação de métodos quimiométricos. 2001. 138p. Tese (doutorado) - Universidade Estadual de Campinas, Instituto de Química, Campinas, SP ”. <http://www.repositorio.unicamp.br/handle/REPOSIP/249824>. [Online: acessado em 29 de julho de 2021].

Filzmoser, P., Gregorich, M. (2020). “Multivariate Outlier Detection in Applied Data Analysis: Global, Local, Compositional and Cellwise Outliers”. <https://doi.org/10.1007/s11004-020-09861-6>. [Online: acessado em 21 de setembro de 2020].

Guedes, G. T. A. (2009). “UML 2 : uma abordagem prática”. Editora Novatec.

Maciel, R. (2015). “Métodos de avaliação dos alimentos”. http://www.dzo.ufla.br/Roberto/analise_avaliacao_alimentos.pdf. [Online: acessado em 21 de março de 2021].

- Kasvi. (2016). “Espectrofotometria – Princípios e aplicações”.
<https://kasvi.com.br/espectrofotometria-principios-aplicacoes/>. [Online: acessado em 28 de julho de 2021].
- Puga, S., França, E., Goya, M. (2013). “Banco de Dados: Implementação em SQL, PL SQL e Oracle 11g”. Editora Pearson.
- Salman, A. K. D. Ferreira, A. C. D., Soares, J. P. G., Souza, J. P. (2010). “Metodologias para avaliação de alimentos para ruminantes domésticos”.
<https://www.infoteca.cnptia.embrapa.br/infoteca/bitstream/doc/884369/1/doc136alimentacaoderuminantes.pdf>. [Online: acessado em 21 de março de 2021].
- Sommerville, I. (2011). "Engenharia de Software". Editora Pearson, 9ª Edição.
- StarUML. (2021). “A sophisticated software modeler for agile and concise modeling”.
<https://staruml.io/>. [Online: acessado em 25 de março de 2021].
- Teixeira, A. S. (2003). “Alimentos e alimentação dos animais”. UFLA/Faepe, Volume I, 5ª Edição.