

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM INOVAÇÕES TECNOLÓGICAS
MESTRADO EM INOVAÇÕES TECNOLÓGICAS

MARLONS FRANCYS DOS SANTOS

**SOFTWARE PARA QUANTIFICAÇÃO VIA IMAGENS DIGITAIS:
DETERMINAÇÃO DO TEOR DE CLOROFILA EM ÓLEO DE SOJA**

CAMPO MOURÃO

2019

MARLONS FRANCYS DOS SANTOS

**SOFTWARE PARA QUANTIFICAÇÃO VIA IMAGENS DIGITAIS:
DETERMINAÇÃO DO TEOR DE CLOROFILA EM ÓLEO DE SOJA**

Dissertação apresentada ao Curso de Pós-Graduação em Inovações Tecnológicas, Universidade Tecnológica Federal do Paraná, como parte das exigências para a obtenção do título de Mestre em Inovações Tecnológicas.

Orientador: Prof. Dr. Paulo Henrique Março

Coorientador: Prof. Dr. André Luis Schwerz

CAMPO MOURÃO

2019

Dados Internacionais de Catalogação na Publicação

S237 Santos, Marlons Francys dos

Software para quantificação via imagens digitais: determinação do teor de clorofila em óleo de soja / Marlons Francys dos Santos. – Campo Mourão, 2019.
87 f. : il. color. ; 30 cm.

Orientador: Paulo Henrique Março
Dissertação (Mestrado) – Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Inovações Tecnológicas, Campo Mourão, 2019.
Inclui bibliografia.

1. Análise espectral. 2. Clorofila – Análise. 3. Óleos vegetais. 4. Processamento de imagens. 5. Inovações tecnológicas – Dissertações. I. Março, Paulo Henrique, orient. II. Schwerz, André Luis, co-orient. III. Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Inovações Tecnológicas. IV. Título.

CDD (22. ed.) 658.514

Biblioteca da UTFPR - Câmpus Campo Mourão

Bibliotecária/Documentalista:
Andréia Del Conte de Paiva – CRB-9/1525



TERMO DE APROVAÇÃO

SOFTWARE PARA QUANTIFICAÇÃO VIA IMAGENS DIGITAIS: DETERMINAÇÃO DO TEOR DE CLOROFILA EM ÓLEO DE SOJA

por

Marlons Francys dos Santos

Essa dissertação foi apresentada às quatorze horas, do vigésimo sétimo dia de dois mil e dezenove, como requisito parcial para a obtenção do título de Mestre em Inovações Tecnológicas, Linha de Pesquisa Inovações Tecnológicas em Gestão da Produção e Qualidade, no Programa de Pós-Graduação em Inovações Tecnológicas – PPGIT, da Universidade Tecnológica Federal do Paraná. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

(Prof. Dr. Paulo Henrique Março)
Prof.(a) Orientador(a)

(Prof. Dr. Rafael Liberato Roberto)
Membro titular

(Prof. Dr. André Marcelo de Souza)
Membro Externo

Dedico esta pesquisa primeiramente à
DEUS pelo dom da vida, a minha esposa
e filho pelo amor, paciência e incentivo
durante toda a caminhada, a meus pais, e
a todos os professores que contribuíram
para que isso fosse possível.

AGRADECIMENTOS

Agradeço primeiramente a Deus, pela força e coragem durante toda a caminhada e pela oportunidade de estudar e ter a chance desta conquista.

A minha esposa Camila e meu filho Miguel pelo amor, incentivo e paciência nos momentos de ausência. Gostaria de dedicar também a toda minha família que, com muito carinho e apoio não mediram esforços para que eu concluísse esta etapa.

Agradeço ao meu coorientador Prof. Dr. André Luis Schwerz, pela sabedoria e disponibilidade com que guiou o meu trabalho durante toda a minha trajetória.

A todos os professores do PPGIT pela disposição na transmissão de conhecimento e apoio durante esse período.

Em especial agradeço ao meu orientador Prof. Dr. Paulo Henrique Março pela confiança e apoio durante todo o desenvolvimento deste trabalho que foi fundamental para a conclusão do mesmo.

Sem sonhos, a vida não tem brilho. Sem metas, os sonhos não têm alicerces. Sem prioridades, os sonhos não se tornam reais. Sonhe, trace metas, estabeleça prioridades e corra riscos para executar seus sonhos. Melhor é errar por tentar do que errar por se omitir. (CURY, 2002)

RESUMO

DOS SANTOS, Marlons Francys. **Software para quantificação via imagens digitais**: determinação do teor de clorofila em óleo de soja. Dissertação (Mestrado em Inovações Tecnológicas) - Universidade Tecnológica Federal do Paraná. Campo Mourão, 2019.

Este trabalho visa o desenvolvimento de um software para viabilizar o uso de imagens digitais na análise do teor de clorofila em óleo de soja. A principal motivação para o desenvolvimento deste invento vem do fato de que parte do conhecimento científico gerado nas instituições de ensino e pesquisa do Brasil tardam a se transformar em ferramenta aplicável, muitas vezes, devido a falta de integração entre as diferentes áreas do conhecimento. Como consequência, as prateleiras das bibliotecas das universidades estão cheias de estudos cujas aplicações podem ser viabilizadas, demandando conhecimento em diferentes áreas para gerar inovação tecnológica significativa para os mais diversos nichos. Para ilustrar este fato, os trabalhos que envolvem ferramentas quimiométricas para implementação de novas metodologias merecem destaque, principalmente nos casos de análises rápidas e menos invasivas. Avanços importantes destas aplicações podem ser vistos na indústria farmacêutica e agroalimentar, as quais se utilizam de ferramentas não invasivas para medidas *online* a partir de calibrações multivariadas associadas a espectroscopia na região do infravermelho próximo. Para este trabalho, utilizou-se como base uma dissertação de mestrado em inovações tecnológicas que fez uso de imagens digitais para quantificação de teor de clorofila em óleos de soja, na qual se propõe a aquisição de imagens a partir de câmeras digitais comuns de baixo custo para tornar o processo simples, rápido, preciso, e não destrutivo. O software computacional produzido teve a finalidade de integrar sistemas de imagens digitais com o método padrão utilizado por uma cooperativa que armazena e esmaga soja para produção de óleo. A integração das metodologias foi realizada desde a conexão da câmera fotográfica digital com a nuvem computacional, o desenvolvimento do software para os cálculos e geração dos modelos matemáticos, até os sistemas de troca de informação disponíveis para dispositivos e plataformas. Ao final, foi possível inferir que o software produzido apresenta potencial para ser expandido para uma grande diversidade de aplicações.

Palavras-chave: Espectroscopia. RGB. PLS. Agricultura. Clorofila.

ABSTRACT

DOS SANTOS, Marlons Francys. **Software para quantificação via imagens digitais**: determinação do teor de clorofila em óleo de soja. Dissertação (Mestrado em Inovações Tecnológicas) - Universidade Tecnológica Federal do Paraná. Campo Mourão, 2019.

This work aims to develop software to enable the use of digital images in the analysis of chlorophyll content in soybean oil. The main motivation for the development of this invention comes from the fact that part of the scientific knowledge generated in teaching and research institutions in Brazil is slow to become an applicable tool, often due to the lack of integration between different areas of knowledge. As a result, university library shelves are full of studies whose applications can be made feasible, requiring knowledge in different areas to generate significant technological innovation for the most diverse niches. To illustrate this fact, the work involving chemometric tools to implement new methodologies deserves attention, especially in the case of quick and less invasive analyzes. Important advances in these applications can be seen in the pharmaceutical and agri-food industry, which use non-invasive tools for online measurements from multivariate calibrations associated with near-infrared spectroscopy. For this work, a master's dissertation on technological innovations was used, which used digital images to quantify chlorophyll content in soybean oils, which proposes the acquisition of images from ordinary low-cost digital cameras. to make the process simple, fast, accurate, and non-destructive. The computational software produced was intended to integrate digital imaging systems with the standard method used by a cooperative that stores and crushes soybeans for oil production. Methodologies were integrated from the connection of the digital camera to the computational cloud, the development of software for calculations and generation of mathematical models, and the information exchange systems available for devices and platforms. In the end, it was possible to infer that the software produced has the potential to be expanded to a wide range of applications.

Keywords: Spectroscopy. RGB. PLS. Agriculture. Chlorophyll.

LISTA DE ILUSTRAÇÕES

Figura 1 - BPMN Método AOCS Cc13d-55/2009	19
Figura 2 – BPMN Construção do modelo proposto por Basseti (2019) Preparar Amostra	20
Figura 3 - BPMN Construção do modelo proposto por Basseti (2019) Carregar Imagem Matlab	20
Figura 4 - BPMN Construção do modelo proposto por Basseti (2019) Montar Matriz	21
Figura 5 - Processo para calibração do modelo proposto por Basseti (2019)	21
Figura 6 - Exemplo de Requisição no Protocolo HTTP	29
Figura 7 – Funcionamento de Serviços Web	30
Figura 8 - Representação Objeto JSON	32
Figura 9 - Representação Tabela Banco de Dados instrumentoColetor	34
Figura 10 - Representação do Relacionamento Cidade - Unidade Federativa	35
Figura 11 – Interpretação do código fonte Python	37
Figura 12 – Etapas construção software	39
Figura 13 - Imagem Capturada Amostra Óleo de Soja	42
Figura 14 – Esquema Construção da Matriz X a partir de uma imagem digital	44
Figura 15 - Diagrama de caso de uso	45
Figura 16 - Diagrama de Atividade (Processamento de Imagem)	49
Figura 17 - Esquema proposto para a arquitetura do software.	51
Figura 18 - Modelo Relacional (software proposto)	56
Figura 19 – Exemplo montar histograma Canal B (RGB)	58
Figura 20 - Algoritmo PLS2 exemplo Scikit Learn	59
Figura 21 - Página Inicial Software	60
Figura 22 - Tratamento de Erros do Software (Exceção)	61
Figura 23 - Cadastro Modelo Calibração	61
Figura 24 - Imagem selecionada para uma das amostras	62
Figura 25 - Inclusão Amostra Modelo de Dados	63
Figura 26 - Calibrar Modelo (Visão Responsiva)	64
Figura 27 – Tela Parâmetros de Mérito	65
Figura 28 - Predição de resultado em amostra nova	66
Figura 29 - Resultado predição amostra nova	66
Figura 30 - Predição amostras cadastradas	67
Figura 31 - Resultado Predição Amostra 1	68
Figura 32 - Resultado Predição Amostra 2	68
Figura 33 - Resultado Predição Amostra 3	69
Figura 34 - BPMN carregamento amostra (Software)	70
Figura 35 - BPMN calibrar modelo (Software)	71
Figura 36 - Protótipo tela inicial	82

Figura 37 - Protótipo tela upload	82
Figura 38 - Protótipo captura imagem 1	83
Figura 39 - Protótipo captura amostra 2.....	83

LISTA DE QUADROS

Quadro 1 - Descrição requisito funcional.....	40
Quadro 2 - Descrição requisitos não funcional.....	41
Quadro 3 - Exemplo Codificação Base64.....	43
Quadro 4 - Legenda Diagrama Caso de Uso	45
Quadro 5 - Lógica redimensionar imagem	57
Quadro 6 - Predição amostras valores conhecidos.....	66
Quadro 7 - Comparativo resultados Basseti(2019) x Software.....	71

LISTA DE TABELAS

Tabela 1 - Dicionário de dados ASS_AMOSTRA_CALIBRACAO	84
Tabela 2 - Dicionário de dados CAD_AMOSTRA_MODELO	85
Tabela 3 - Dicionário de dados CAD_CALIBRACAO	85
Tabela 4 - Dicionário de dados CAD_MATRIZ_X	85
Tabela 5 - Dicionário de dados CAD_MATRIZ_Y	86
Tabela 6 - Dicionário de dados CAD_MODELO.....	86
Tabela 7 - Dicionário de dados CAD_PARAMETRO_MODELO	87

LISTA DE ABREVIATURAS

API	Application Programming Interface
BPMN	Business Process Model and Notation
FIR	Far Infrared Spectroscopy
GPL	General Public License
HTTP	Hypertext Transfer Protocol
IDE	Ambiente de Desenvolvimento Integrado
JSF	Java Server Faces
MVP	Minimum Viable Product
PLS	Regressão por Mínimos Quadrados Parciais
RGB	Sistema de cores formado pelas iniciais das cores Red (vermelho), Green (verde) e Blue (azul)
SGBD	Sistema Gerenciador de Banco de Dados
REST	Representational State Transfer

SUMÁRIO

1 INTRODUÇÃO	16
1.1 PROBLEMÁTICA.....	18
1.2 OBJETIVOS.....	22
1.2.1 Objetivo Geral.....	22
1.2.2 Objetivos Específicos.....	22
1.3 JUSTIFICATIVA.....	23
2 FUNDAMENTAÇÃO TEÓRICA	24
2.1 QUIMIOMETRIA.....	24
2.1.1 Calibração Multivariada.....	24
2.1.1.1 Seleção de amostras: o algoritmo de Kennard-Stone.....	25
2.1.1.2 Detecção de amostras anômalas: os outliers.....	25
2.1.1.3 Validação: cálculo de parâmetros de mérito.....	26
2.2 DESENVOLVIMENTO DE SOFTWARE.....	26
2.2.1 Metodologia de Desenvolvimento ágil Scrum.....	27
2.2.2 Requisitos de Software.....	27
2.2.3 Arquitetura de Software.....	28
2.2.3.1 Protocolo HTTP.....	29
2.2.4 Serviços Web.....	30
2.2.4.1 Serviço REST.....	31
2.2.4.2 JSON.....	32
2.2.4.3 Front-end.....	32
2.2.4.4 Back-end.....	33
2.2.5 Banco de Dados.....	33
2.2.5.1 Banco de dados relacional.....	34
2.2.5.2 Banco de dados PostgreSQL.....	35
2.2.6 Linguagem de Programação.....	36
2.2.6.1 A linguagem Python.....	36
2.2.6.2 A linguagem Java.....	37
3 PROJETO DO SOFTWARE	39
3.1 METODOLOGIA DE DESENVOLVIMENTO.....	39
3.1.1 Levantamento de Requisitos.....	40
3.1.1.1 Descrição dos requisitos de software.....	40
3.1.1.2 Aquisição de imagens.....	42
3.1.1.3 Processamento de imagem.....	42
3.1.1.4 Transferência de Dados.....	43
3.1.1.5 Padrão Processamento (Imagem).....	43
3.1.1.6 Diagrama de caso de uso.....	44
3.1.1.7 Diagrama de Atividades.....	48

3.1.2 Prototipação.....	50
3.1.3 Definição Arquitetura	50
3.1.4 Definição de Tecnologias.....	51
3.1.4.1 Bibliotecas.....	52
3.1.5 Construção do Banco de Dados	55
3.1.6 Programação do Software	57
3.1.7 Testes	59
4 RESULTADOS E DISCUSSÃO.....	60
4.1 FUNCIONAMENTO DO SOFTWARE.....	60
4.2 BENEFÍCIOS DO SOFTWARE EM RELAÇÃO AO MÉTODO MANUAL.....	69
4.3 VALIDAÇÃO DOS MÉTODOS DESENVOLVIDOS	71
5 CONCLUSÃO	73
5.1 PERSPECTIVAS FUTURAS.....	73
REFERÊNCIAS.....	75

1 INTRODUÇÃO

As prateleiras das bibliotecas das instituições de ensino e pesquisa estão repletas de materiais interessantes e que, na maioria dos casos, demanda tempo considerável para que possam ser viabilizados como produto, disponível para um devido aproveitamento. No entanto, para a vasta maioria dos casos, a integração entre diferentes áreas do conhecimento poderia trazer soluções a ponto de transformar o conhecimento produzido em ferramenta de factível. Considerando-se as soluções propostas por meio de calibração multivariada, a integração entre um periférico com função de aquisição de medidas para análises rápidas (como espectrômetros, colorímetros e câmeras digitais) e um software para permitir os cálculos aumentaria o potencial de aplicação das propostas.

O desenvolvimento de um software possibilita automatização de processos, por meio de uma série de benefícios que a tecnologia pode oferecer: agilidade, padronização, redução de custos, segurança de informações e otimização de tempo (diminuição de retrabalho). A evolução tecnológica facilita a execução de tarefas e operações corporativas, pois a tecnologia opera a favor de produtividade. Um software visando aprimoramento em um processo fabril pode ser viabilizado visando maior segurança e controle operacional e gerencial.

Nos últimos anos, o conceito de fábricas inteligentes ou produção em grande escala vem ganhando cada vez mais espaço dentro do mundo corporativo, e qualquer benefício na produção de um produto, torna-se uma vantagem competitiva de extrema importância na entrega final. Atualmente, a inteligência artificial permite que sistemas de software tomem decisões que antes deveriam ser avaliadas exclusivamente por seres humanos. Desta forma, quando diagnosticado um padrão, é possível configurar um programa para auxiliar no processo de tomada de decisão.

A comercialização de derivados da soja exige um rigoroso controle de qualidade dos grãos e dos padrões de comercialização (ARAUJO, 2014). Entretanto, os procedimentos analíticos utilizados no controle de qualidade são lentos, dispendiosos, com custo elevado, necessitam de mão de obra qualificada e apresentam potencial elevado na geração de resíduos químicos. A fim de mitigar esses problemas, Basset e colaboradores propuseram a análise do teor de clorofila a partir de imagens digitais em lugar do método convencional em amostras de óleo de soja branqueado.

Atualmente, a técnica adotada para mensurar o teor de clorofila é fundamentada em espectroscopia na região ultravioleta-visível (UV-Vis) proposto pela sociedade americana de química de óleos (American Oil Chemists' Society – AOCS), particularmente a indicada em Cc13d-55/2009 (AOCS, 2009). Apesar de funcional, esta técnica demanda tempo considerável para obtenção dos resultados, o que limita o número de amostras analisadas, aumentando o consumo de reagentes e de custos devido à dosagem de argilas, além da geração e descarte de resíduos gerados.

O desenvolvimento de trabalhos acadêmicos que propõem o uso de modelos de análises multivariados capazes de correlacionar técnicas não invasivas/destrutivas vem ganhando destaque em todas as áreas do conhecimento. No entanto, o número de tarefas necessárias para que se possa utilizar os resultados é, atualmente, um fator determinante no que diz respeito a inviabilização do uso das propostas: os trabalhos acadêmicos seccionam as etapas de modo que a aquisição das medidas está separada do módulo de construção/alimentação do modelo, que por sua vez não está necessariamente vinculada aos bancos de algoritmos necessários para devolver as respostas solicitadas. Além disso, a maior

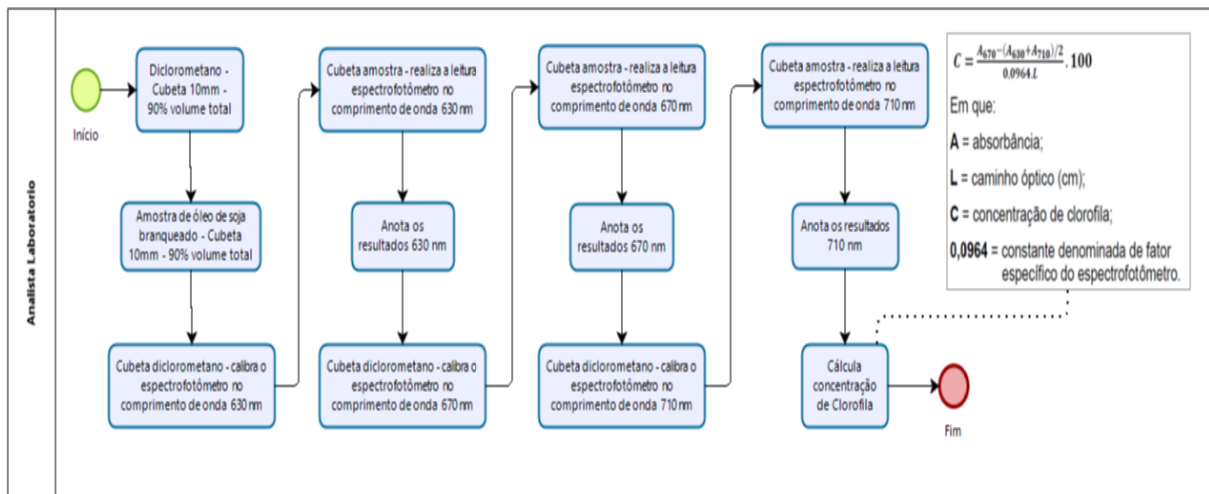
parte dos softwares disponíveis para os cálculos não é gratuita, tornando ainda mais difícil a automatização dos processos.

Desta forma, a centralização de uma tecnologia que possibilite o interfaceamento entre os métodos não invasivos/destrutivos já estudados, tais como os de espectroscopia (SANTOS, 2017; PAVIA, 2001; VALDERRAMA; BRAGA; POPPI, 2009), pode possibilitar que um maior número de laboratórios industriais tenha acesso a inovação tecnológica no sentido de aprimorar o monitoramento dos produtos, maximizando lucros e permitindo ações menos agressivas ao meio ambiente diminuindo, por consequência, o uso de reagentes químicos nos processos analíticos.

1.1 Problemática

Seguindo a descrição da metodologia proposta por Bassetti (2019), a Figura 1 ilustra o método proposto pela AOCS Cc13d-55/2009 (AOCS, 2009), utilizado como método convencional para determinação do teor de clorofila. Destaca-se o fato de que a metodologia é totalmente manual e propensa a erros que poderiam ser minimizados.

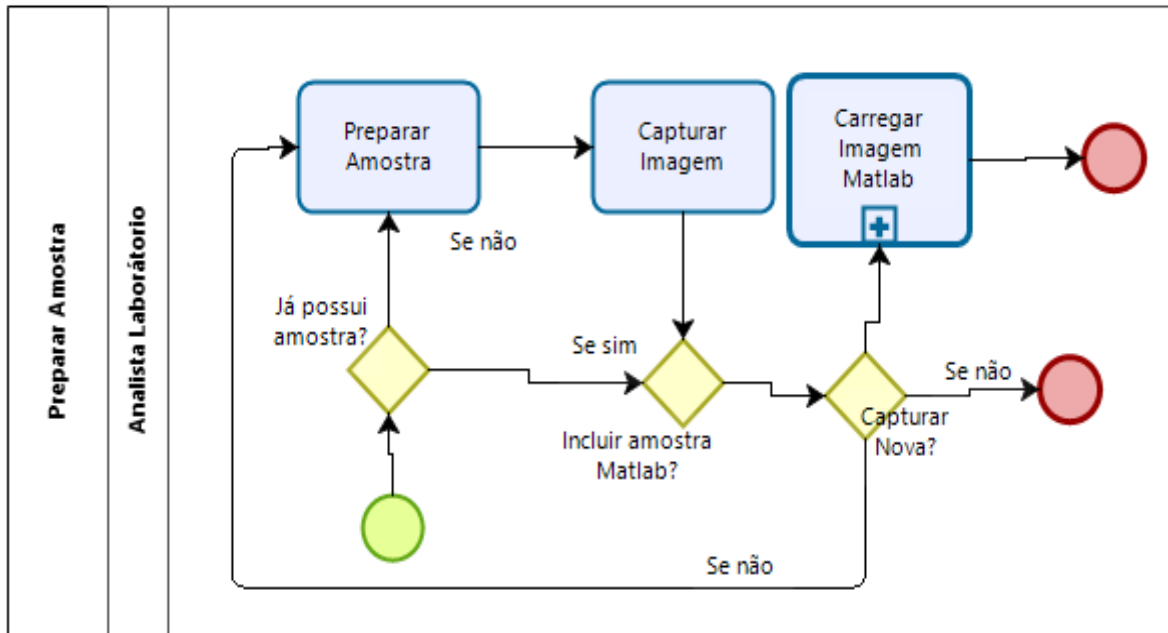
Figura 1 - BPMN Método AOCS Cc13d-55/2009



Fonte: Autoria Própria (2019)

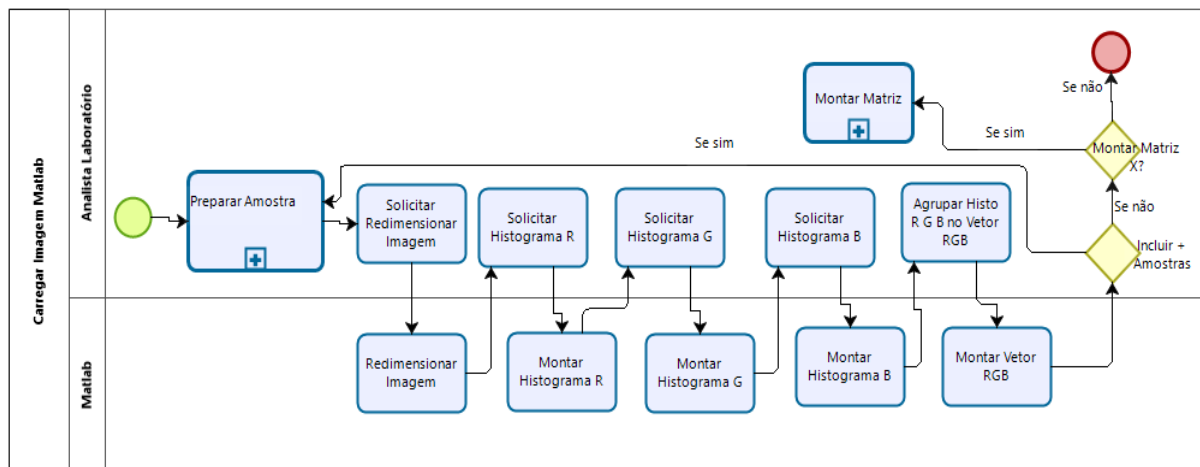
No estudo realizado por Basseti (2019), propõe-se um método alternativa para prever a clorofila presente em óleo de soja branqueado, empregando para este fim imagens digitais e calibração multivariada por mínimos quadrados alternantes (PLS, do inglês *Partial Least Squares* (GELADI; KOWALSKI; CHEMISTRY, 1986; OLIVIERI; BORTOLATO; ALLEGRINI, 2015; SOUZA; POPPI, 2012)) e utilizando os recursos da ferramenta Matlab® para implementação. No entanto, o seccionamento das tarefas para que se possa chegar no resultado inviabilizam a implementação da proposta em uma escala de demanda industrial. Para auxiliar a visualização a etapa de inclusão das amostras foi separada em 3 subetapas, sendo elas: “Preparar Amostra”, “Carregar Imagem Matlab” e “Montar Matriz” que são representadas pelas Figura 2, 3 e 4, respectivamente onde o Analista realiza o trabalho por meio do Matlab®. Assim, é possível visualizar o formato padrão que deve ser seguido para construção de modelo de calibração, mais especificamente o proposto por Basseti (2019). A Figura 5 ilustra o funcionamento da calibração do modelo.

Figura 2 – BPMN Construção do modelo proposto por Basseti (2019) Preparar Amostra



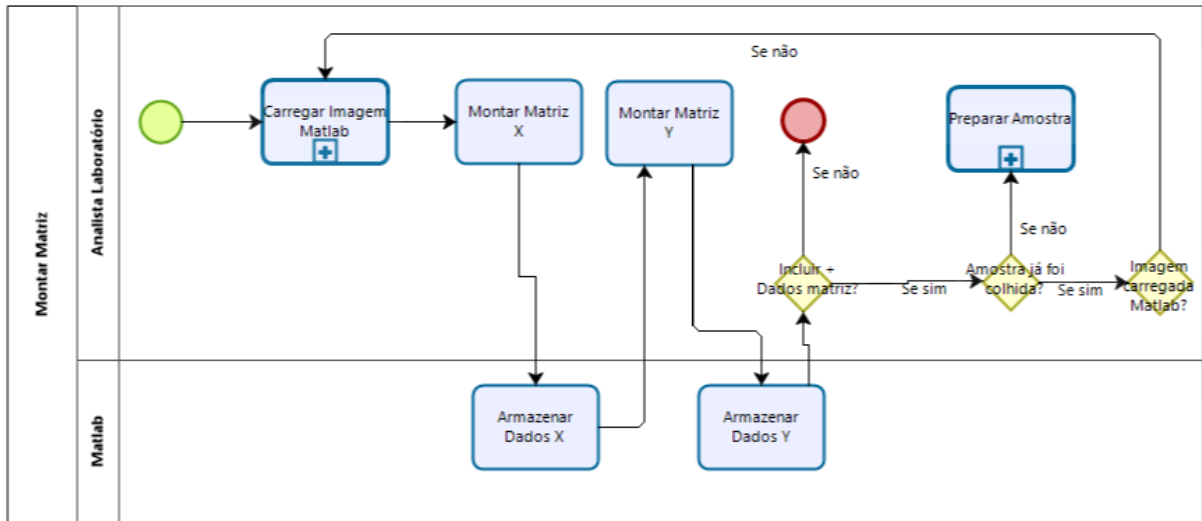
Fonte: Autoria Própria (2019)

Figura 3 - BPMN Construção do modelo proposto por Basseti (2019) Carregar Imagem Matlab



Fonte: Autoria Própria (2019)

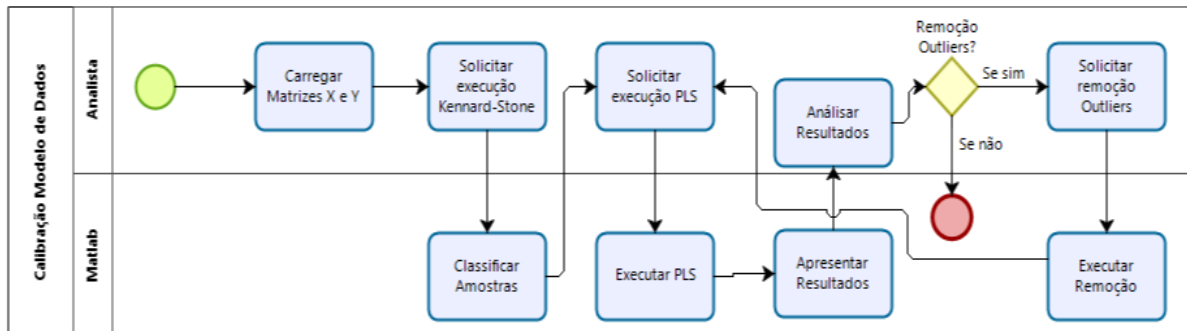
Figura 4 - BPMN Construção do modelo proposto por Basseti (2019) Montar Matriz



Powered by
bizagi
Modeler

Fonte: Autoria Própria (2019)

Figura 5 - Processo para calibração do modelo proposto por Basseti (2019)



Powered by
bizagi
Modeler

Fonte: Autoria Própria (2019)

Esse tipo de manuseio compromete a segurança do método, uma vez que um trabalhador está sujeito a cometer erros durante o processo industrial. Para que uma empresa consiga êxito em um mercado de alta concorrência com aumento da rentabilidade da produção, é essencial melhorar a performance que, conseqüentemente, tem implicações na confiabilidade do produto. Logo, a proposta deste trabalho é aprimorar o uso de um dos modelos de calibração multivariada já desenvolvido a fim de viabilizar a aplicação da ferramenta de forma prática.

1.2 Objetivos

As seções seguintes descrevem o objetivo geral, os objetivos específicos e a justificativa do trabalho realizado, visando demonstrar os limites estabelecidos para a criação do software proposto.

1.2.1 Objetivo Geral

Desenvolver um software computacional para determinação de clorofila em amostras de óleo de soja branqueado por meio do processamento de imagens digitais e calibração multivariada, integrando desde as medidas laboratoriais até as respostas quantitativas.

1.2.2 Objetivos Específicos

Para que o desenvolvimento da ferramenta proposta, observaram-se os seguintes objetivos específicos:

- Avaliar a proposta de pesquisa de Basseti e colaboradores (2019), fundamentada no uso de imagens digitais para determinação de teor de clorofila em óleo de soja, a fim de transformar a tecnologia sugerida em software computacional;
- Definir a arquitetura de software adequada para o desenvolvimento do projeto;
- Aplicar técnicas para processamento de imagens digitais por meio de um algoritmo que redimensione a imagem e construa histograma RGB;
- Desenvolver modelo de calibração multivariada a partir das imagens obtidas;
- Manter as informações de modelos e amostras centralizados para facilitar o processamento dos dados;
- Validar os métodos por meio do cálculo e avaliação dos parâmetros de mérito;

- Englobar todas as ferramentas para execução do processo em um único software.

1.3 Justificativa

Considera-se que para a aplicação da proposta de Basseti (2019) e colaboradores possa ser implementada em escala industrial, será necessária a automatização do processo como um todo, o que seria viabilizado através da utilização de uma ferramenta de software. O software deverá otimizar tempo de medida, auxiliar na redução de custos de produção e reduzir a utilização de argila branca no processo de clareamento e, conseqüentemente, diminuir os resíduos descartados sendo um forte aliado ao projeto de química verde.

O método utilizado atualmente pela AOCS Cc13d-55/2009, tem pontos que podem ser melhorados, tais como condução laboratorial totalmente manual, uso de vidrarias e reagentes cujo preparo pode acumular erros aleatórios, a execução deste estudo é justificada pela possibilidade de agilizar e melhorar diretamente o processo de tomada de decisões, diminuindo o custo da produção, bem como o uso de argila branca no tratamento do óleo de soja.

O cenário atual de morosidade na quantificação de amostras de soja nas refinarias demanda uma tecnologia que supra a necessidade de inovação na área de análise de soja, tendo em vista que por meio de um periférico que capture imagem (câmera digital, *webcam*, etc) e com acesso à ferramenta de software proposta será possível executar, em um curto período de tempo, uma análise confiável e tomar a decisão adequada nessa etapa do tratamento de óleo de soja branqueado, reduzindo a quantidade de resíduos e promovendo química sustentável.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os conceitos de quimiometria que foram utilizados no desenvolvimento do software. Assuntos como arquitetura de software, banco de dados e linguagens de programação também são abordados nesse capítulo considerando a utilidade destes tópicos para o embasamento deste trabalho.

2.1 QUIMIOMETRIA

A quimiometria é a área da química analítica que aplica métodos matemáticos e estatísticos para planejar, otimizar ou selecionar procedimentos de medidas e experimentos, buscando extrair o máximo de informação química relevante, com a análise de dados químicos (FERREIRA, 2015). Tais procedimentos começaram a ser implementados em meados dos anos 70, considerando a chegada do computador nos laboratórios químicos (NETO; SCARMINIO; BRUNS, 2006).

Considerando os avanços tecnológicos, surgiu-se a necessidade de desenvolvimento de novos métodos estatísticos e matemáticos para analisar a grande quantidade de dados produzidos por instrumentos analíticos assistidos por computadores, permitindo a utilização da estatística multivariada para maior extração de informações de dados químicos.

O software proposto neste trabalho será utilizado para quantificar clorofila nas amostras de óleo de soja branqueado, utilizando-se de imagens digitais e um método calibração multivariada, fundamentado em Mínimos Quadrados Parciais (PLS, do inglês *Partial Least Squares*) (GELADI; KOWALSKI, 1986).

2.1.1 Calibração Multivariada

A calibração multivariada visa estabelecer uma relação quantitativa entre dois conjuntos de dados de medição, sendo um aquele utilizado como referência (o que se deseja otimizar) e o outro é a metodologia capaz de otimizar os percalços observados no método de referência (GELADI, 2003; HIBBERT, 2004; VALDERRAMA, 2009a).

Os métodos de calibração multivariada utilizam toda a estrutura de uma amostra, obtida por sinais que podem variar desde um espectro e cromatogramas bidimensionais (LEME et al., 2019; MARÇO et al., 2011; SEASHOLTZ; KOWALSKI, 1993) (FERREIRA, 2010) até dados de imagens de alta complexidade (ALEXANDRINO; POPPI, 2013; MAMIÁN-LÓPEZ; POPPI, 2015).

As etapas para calibração multivariada consistem em:

1. Seleção de amostras para os conjuntos de calibração e validação;
2. Detecção de amostras anômalas (*outliers*);
3. Construção do modelo de calibração;
4. Validação a partir da previsão de concentrações de amostras;
5. Avaliação dos parâmetros de mérito.

A seguir, estão apresentadas breves explicações sobre as metodologias empregadas neste trabalho para seguimento das etapas descritas acima.

2.1.1.1 Seleção de amostras: o algoritmo de *Kennard-Stone*

Neste trabalho, a separação dos conjuntos de calibração e validação foi realizada a partir do algoritmo de *Kennard-Stone* (KENNARD; STONE, 1969). Esta metodologia seleciona as amostras com base em suas distâncias, de modo que a primeira amostra selecionada é a que apresenta a maior distância em relação à amostra mais central. A segunda amostra a ser escolhida será aquela que apresentar maior distância em relação à primeira amostra selecionada. A amostra seguinte a ser selecionada apresentará maior distância em relação à última amostra escolhida, e assim sucessivamente, até atingir o número de amostras desejadas.

2.1.1.2 Detecção de amostras anômalas: os *outliers*

Dentre as metodologias para detecção de *outliers*, as mais comuns são baseadas na identificação de amostras com grande influência no modelo (*leverage*), altos resíduos não modelados nos dados espectrais e altos resíduos na variável dependente (Valderrama & Braga, 2009). O *leverage* indica a distância que uma amostra se encontra em relação à média de conjunto de dados, demonstrando o “peso relativo” que a amostra exerce dentro do modelo. Logo, uma amostra com

elevado resíduo e *leverage* simultaneamente sugere que se trata de uma amostra muito diferente das demais. Nestes casos, é preciso verificar com cautela se há necessidade de remoção de amostras ou se há necessidade de se adicionar outras mais.

2.1.1.3 Validação: cálculo de parâmetros de mérito

Depois da construção do modelo de calibração é necessário realizar a validação, que visa avaliar a qualidade e a capacidade de se empregar a metodologia na previsão de amostras “desconhecidas”, previamente escolhidas pelo método de *Kennard-Stone*. A qualidade dos modelos é avaliada a partir dos parâmetros de mérito, dentre os quais destacam-se a raiz quadrada do erro médio quadrático de calibração (RMSEC, do inglês *Root Mean Square Error of Calibration*), raiz quadrada do erro médio quadrático de previsão (RMSEP, do inglês *Root Mean Square Error of Prediction*), linearidade, exatidão, precisão, sensibilidade, seletividade, coeficiente de correlação, limites de detecção e quantificação, relação de desvio do desempenho, robustez, intervalos de confiança, teste para erros sistemáticos, extensão da faixa de trabalho ou faixa linear dinâmica (Silva, 2011; Valderrama, 2009).

Considerando o objetivo do trabalho de se produzir o software, todos os detalhes mais aprofundados sobre os métodos e equações utilizadas estão dispostas no trabalho de Basseti (2019).

2.2 Desenvolvimento de software

Esta seção tem como objetivo apresentar resumidamente os principais conceitos que fundamentam o desenvolvimento do software proposto neste trabalho. Para o completo entendimento, esta seção é opcional a leitores familiarizados com a área.

Atualmente, sistemas de software são essenciais para automatização de melhorias de processos, buscando agilidade, otimização de tempo e aumento dos lucros. De acordo com Pressman (2011), software de computador é o produto em

que os profissionais constroem e, depois mantêm ao longo do tempo, abrange programas que executam em computadores. Para Sommerville (2011), sistemas de software são programas de computador e toda a documentação associada. Os sistemas de software podem ser desenvolvidos para um cliente específico ou para um mercado em geral.

No caso deste estudo, pretende-se resolver problemas de indústrias de óleo de soja, que necessitam de maior agilidade para quantificar clorofila presente nas amostras de óleo de soja branqueado.

2.2.1 Metodologia de Desenvolvimento ágil Scrum

O método ágil Scrum é baseado nos trabalhos de Pittman em 1993 e Booch em 1995, aderindo aos princípios do desenvolvimento ágil de software. Neste tipo de metodologia, enfatiza-se a flexibilidade, uma vez que somente as partes de início e final (planejamento e fechamento) estão bem definidas durante o projeto. As demais atividades são desenvolvidas em uma série de entregas periódicas (*sprints*) (SCHWABBER et al. 2013).

Esse tipo de abordagem visa que o produto seja entregue com uma maior probabilidade de sucesso, mesmo se enfrentar mudanças significativas durante sua concepção. Neste tipo de metodologia os riscos são reduzidos com a produção em ciclos menores e as entregas frequentes de partes prontas do produto, partido do mais importante para o menos importante transmitindo uma sensação de confiança para o cliente.

Para o desenvolvimento do software foi adotado a metodologia em questão, adaptando-se o *framework* para contemplar a pesquisa científica.

2.2.2 Requisitos de Software

Os requisitos de software podem variar de uma declaração abstrata de alto nível, de uma função que o sistema deve fornecer ou de uma restrição do sistema, até uma definição detalhada, matematicamente formal, de uma função do sistema.

Para Sommerville (2011), um requisito de software é uma descrição detalhada de uma função do sistema. De acordo com essa definição podemos definir como:

- Uma capacidade que o usuário necessita para resolver um problema ou atingir um objetivo;
- A capacidade para ser atendida ou possuída por um sistema ou componente de um sistema para satisfazer contrato, padrão, especificação ou outro documento formalmente imposto;
- Conjunto de todos os requisitos que formam a base para o desenvolvimento subsequente de um software ou componentes de um software.

Existem dois tipos de classificação de requisitos, denominados de Requisitos Funcionais (RF) e Requisitos Não-Funcionais (RNF).

Os RF preocupam-se com a funcionalidade e serviços do sistema, com as funções que o sistema deve fornecer para o cliente e como o sistema se comporta em situações específicas. Já os RNF são aqueles que declaram restrições, ou atributos de qualidade para um software e/ou para o processo de desenvolvimento de sistema. Segurança, precisão, usabilidade, performance e manutenibilidade são exemplos de RNF.

2.2.3 Arquitetura de Software

A arquitetura de software tem como objetivo dar a direção a ser tomada e também as tarefas envolvidas em cada área do desenvolvimento. Segundo Bass et al. (2003), a arquitetura de software representa a estrutura, ou o conjunto de estruturas, que compreende os elementos de software, suas propriedades externamente visíveis e seus relacionamentos (BASS et al., 2003).

A arquitetura de software é onde define-se como o software será desenvolvido, segmentando-o em partes. Desta forma, define-se os componentes do software, suas propriedades externas, relacionamentos com outros softwares, etc. Para este projeto, é utilizado a arquitetura cliente-servidor baseada em serviços web que será apresentado nas próximas seções. Além disso, define-se qual/quais os mais adequados sistemas de gerenciamento bancos de dados para o projeto. Para o desenvolvimento deste projeto é utilizado um banco de dados relacional que será abordado posteriormente. Ainda, define-se quais serão as linguagens adotadas, frameworks, padrões de projeto, entre outros.

Esta é uma etapa que deve ser muito bem avaliada, uma vez que qualquer alteração posterior nas definições acima pode ser custosa durante a realização do projeto.

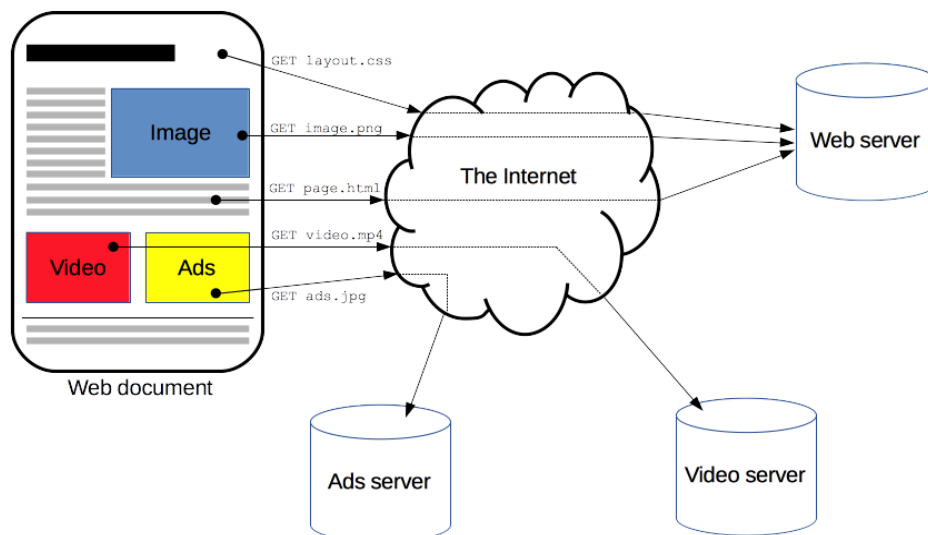
2.2.3.1 Protocolo HTTP

De acordo com Luchi (2019), o protocolo HTTP é a base de qualquer troca de dados na Web e um protocolo cliente-servidor, o que significa que as requisições são iniciadas pelo emitente.

Clientes e servidores se comunicam trocando mensagens individuais (em oposição a um fluxo de dados). As mensagens enviadas pelo cliente são chamadas de solicitações (*requests*), ou também requisições, e as mensagens enviadas pelo servidor como resposta são chamadas de respostas (*responses*).

A Figura 6 ilustra o funcionamento do protocolo HTTP em que, por meio de um documento Web (*Web document*), um cliente envia requisições (GET) solicitando, por exemplo, recursos como a estrutura da página Web, imagens e vídeo que podem estar em servidores distintos.

Figura 6 - Exemplo de Requisição no Protocolo HTTP



Fonte: Luchi (2019)

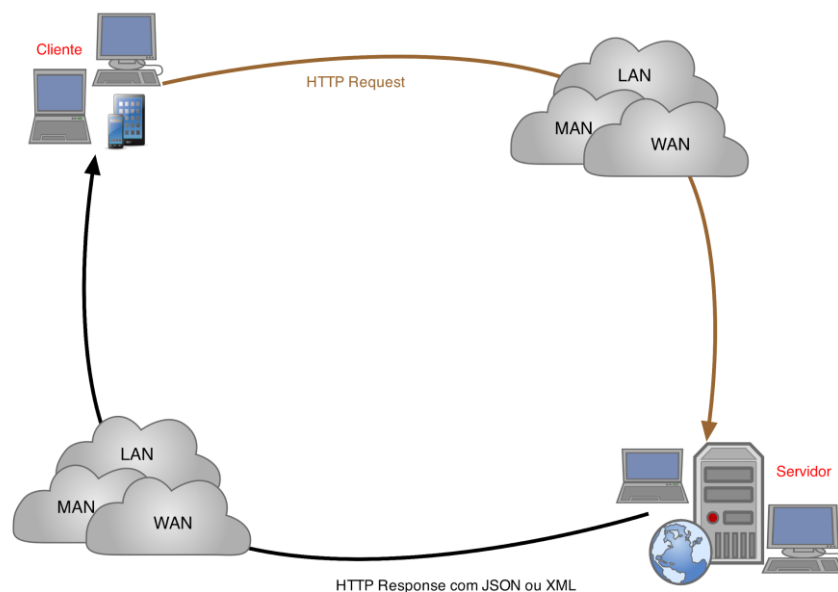
2.2.4 Serviços Web

Serviços Web são conjuntos de aplicações publicadas, localizadas e invocadas por meio da Web, independentemente do tipo de processo que será publicado. Sempre que publicado outras aplicações (até mesmo outros serviços Web) poderão acessá-lo e/ou invocá-lo para utilizar de suas funcionalidades, desde uma simples obtenção de dados até uma interação com outros serviços oferecidos.

A arquitetura de serviços Web se baseia na interação de três entidades: provedor do serviço (*service provider*), cliente do serviço (*service requestor*) e servidor de registro (*service registry*). De uma forma geral, as interações são para publicação, busca e execução de operações.

A Figura 7 esquematiza o funcionamento de Serviços Web, em que é utilizado protocolos de comunicação para transferência de dados, independente das plataformas e linguagens de programação. O cliente envia uma requisição HTTP (*request*) a um servidor, e fica aguardando a resposta em HTTP (*response*), a resposta é dada em um formato esperado pelo cliente.

Figura 7 – Funcionamento de Serviços Web



Fonte: <https://www.devmedia.com.br/web-services-em-aplicacoes-android-e-ios/28901>

Os protocolos e elementos utilizados para o desenvolvimento do software deste estudo são de serviços web baseados em REST tendo em vista a simplicidade do conjunto de regras básicas. Para o tráfego de dados, será utilizado JSON, considerando que as respostas HTTP por meio dele possibilitam grande portabilidade dos serviços oferecidos em várias plataformas que é suportado por uma gama extensa de linguagens.

2.2.4.1 Serviço REST

O padrão arquitetural Transferência de Estado Representacional (REST, acrônimo em inglês de *Representational State Transfer*) facilita a criação e organização de sistemas distribuídos (DOGLIO, 2015). O princípio básico do REST é possibilitar o desenvolvimento de serviços de boa eficiência, que sejam simples, e que permitam trabalhar com dados via HTTP.

Um sistema no estilo REST é denominado RESTful e possui as seguintes características:

- Deve ser um sistema cliente-servidor independente de estado tal que cada requisição deverá ser independente. Deve ser uniformemente acessível em que cada recurso deve ter um endereço exclusivo a um ponto de acesso válido.
- O núcleo da abordagem REST consiste na percepção de que, apesar termo transporte em seu nome, o HTTP consiste em uma Interface de Aplicação (API, do inglês *Application Programming Interface*) e não um simples protocolo de transporte. Desta forma, o HTTP possui métodos bem definidos que correspondem as operações CRUD (*Create, Read, Update, Delete*). Cada requisição HTTP indica qual a operação CRUD que deve ser realizada sobre o recurso, por meio de métodos POST (cria um novo recurso), GET (efetua leitura de um recurso), PUT (atualiza o recurso a partir da requisição), DELETE (remove um recurso).

2.2.4.2JSON

JSON (do inglês, *JavaScript Object Notation*) em tradução livre Notação de Objetos JavaScript é uma estrutura baseada em texto para armazenar e transmitir dados estruturados. Utilizando-se de uma sintaxe simples, o usuário pode facilmente armazenar qualquer informação por meio de cadeia de caracteres, matrizes e objetos usando nada além de uma sequência de texto simples. O usuário também pode alinhar matrizes e objetos, o que lhe permite criar estruturas complexas de dados.

Uma vez criado, um arquivo JSON pode facilmente ser transmitido, pois trata-se de um texto simples. Entre suas principais vantagens, destaca-se:

- Fácil entendimento tanto para leitura como para desenvolvimento;
- Mapeia estruturas de dados usadas por muitas linguagens de programação;
- As linguagens de programação possuem funções ou bibliotecas que podem ler e escrever estruturas JSON.

Para exemplificar seu formato, considere um arquivo JSON que contém o nome do aplicativo proposto (ChemoApp) e ano de sua publicação (2019), conforme ilustrado na Figura 8.

Figura 8 - Representação Objeto JSON

```
{
  "nome": "ChemoAPP",
  "ano": 2019
}
```

Fonte: Autoria Própria (2019)

2.2.4.3 Front-end

Front-end é o nome dado a camada de apresentação de um software, isto é, a interface que é apresentada ao usuário. São todos os elementos que formam a

identidade visual de uma aplicação. De acordo com Viana (2017), o *front-end* é a parte de uma aplicação que interage diretamente com o usuário.

O software proposto neste trabalho conta com um *front-end* desenvolvido em JavaEE, totalmente responsivo cuja principal característica é mudar sua aparência e disposição com base no tamanho da tela onde é exibido, os elementos se reorganizam para tornar a navegação mais simples tornando a experiência do usuário mais agradável.

2.2.4.4 *Back-end*

Back-end é nome dado a camada da aplicação que possui as funcionalidades de um software, engloba todas as regras de negócio, é responsável pela interação com a base de dados, e com qualquer outro recurso que o software tenha de se comunicar.

Para Viana (2017), o *back-end* é tudo que está atrás da aplicação, aquilo que o usuário não pode visualizar, mas que é responsável por persistência de dados, regras de negócio, segurança da informação, performance e outros.

O software produzido neste trabalho possui *back-end* desenvolvido em Python.

2.2.5 Banco de Dados

Com a evolução da informática, os bancos de dados tornaram-se uma demanda indispensável para empresas que utilizam de sistemas de informação, considerando que oferecem segurança dos dados e agilidade em buscas. Antes dessa evolução, os dados das empresas eram mantidos em papéis, pastas, entre outros.

Segundo Heuser (2010), banco de dados é uma coleção de dados persistentes, utilizados por sistemas de aplicações de empresas. Para Elmasri e Navathe (2011), um banco de dados representa algum aspecto do mundo real em que todas as modificações são refletidas e armazenadas. Em outras palavras, é uma coleção logicamente coerente de dados com algum significado inerente, pois

considera-se que uma variedade aleatória de dados não pode ser corretamente chamada de banco de dados. Um banco de dados é projetado, construído e populado com dados para uma finalidade específica, geralmente destinado a um grupo definido de usuários e algumas aplicações previamente concebidas nas quais esses usuários estão interessados.

Um banco de dados pode ser definido como uma coleção de informações relacionadas que possuem um significado implícito. Pode-se considerar como exemplo uma lista de endereços armazenados em um programa de computador tal como Microsoft Excel.

2.2.5.1 Banco de dados relacional

De acordo com Heuser (2009), um banco de dados relacional é uma composição de tabelas ou relações, em que o termo tabela é mais comum nos produtos comerciais. Cada tabela (identificada por um nome) pode armazenar um conjunto de valores em suas linhas (tuplas). Cada coluna é identificada por um atributo acompanhado pelo tipo de dado permitido a ele (por exemplo, inteiro, real, ou cadeia de caracteres) (ELMASRI; NAVATHE, 2011). No exemplo da Figura 9 apresenta-se a tabela usada para armazenar os instrumentos coletor de dados.

Figura 9 - Representação Tabela Banco de Dados instrumentoColetor

instrumentoColetor		
idInstColetor	dsInstColetor	dsInformacoesGerais
1	IMAGEM DIGITAL	
2	ESPECTROFOTOMETRO	APARELHO LABORATORIAL CUJA FUNCAO E A DE MEDIR E COMPARA A QUANTIDADE DE LUZ(ENERGIA RADIANTE) ABSORVIDA POR DETERMINADA SOLUCAO.

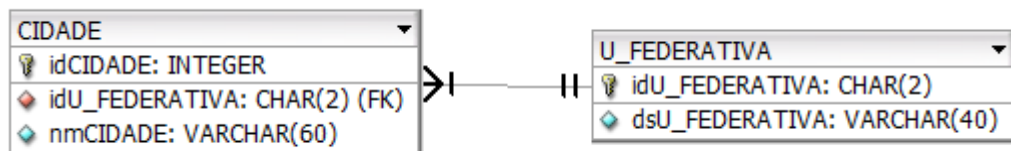
Diagram annotations: 'Atributos' points to the column headers; 'NomeTabela/Relação' points to the table title; 'Tuplas' points to the data rows.

Fonte: Autoria Própria (2019)

O modelo relacional impõe restrições para evitar a repetição de informações, a incapacidade de representar parte da informação e a perda de informação. Essas restrições são conceitualmente chamadas de:

- Restrição de integridade de entidade: o principal “artefato” para esta restrição é a chave primária, que é um conjunto de um ou mais atributos cujos valores formam combinações que não podem se repetir em diferentes linhas. A chave primária é utilizada como índice para criação de relacionamentos com as demais tabelas de um banco de dados. Uma chave primária é um atributo obrigatório na definição de uma tabela e seu valor nunca pode ser nulo.
- Restrição de integridade referencial: especifica a relação entre duas tabelas a fim de manter a consistência dos dados. Informalmente significa que para um registro dependente (filho) existir, ele deve existir na entidade pai. Por exemplo, suponha as duas tabelas chamadas de Cidade e Unidade Federativa ilustradas na Figura 10 em que a Unidade Federativa é a tabela pai do relacionamento, e a Cidade (filho) depende da existência da unidade federativa para existir. Se uma unidade federativa estiver vinculada em alguma cidade, o banco de dados passa a controlar pela integridade referencial e, conseqüentemente, ela não poderá ser excluída do banco de dados. Uma integridade referencial é também chamada de chave estrangeira.

Figura 10 - Representação do Relacionamento Cidade - Unidade Federativa



Fonte: Autoria Própria (2019)

Para o desenvolvimento do software deste trabalho foi utilizado um banco de dados relacional devido à organização da estrutura dos dados, e também ao fato de possibilitar criar referências entre os dados por meio de chaves estrangeiras (relacionamentos entre tabelas).

2.2.5.2 Banco de dados PostgreSQL

O PostgreSQL é um sistema gerenciador de banco de dados relacional que se destaca, considerando seu código fonte aberto e também por ser um dos mais

avançados em recursos neste segmento. Segundo Milani (2008), o PostgreSQL é um software livre diferente da maioria, pois não se utiliza da licença GNU e sim BSD¹. A principal diferença entre as duas licenças é que a BSD¹ pode ser utilizada gratuitamente até mesmo em produtos comerciais.

2.2.6 Linguagem de Programação

Atualmente, muitas das tarefas executadas pelos seres humanos estão inseridas em um programa de computador, seja por meio do smartphone, laptop ou computadores desktop. Desde o acesso a redes sociais e até mesmo chamada de um telefone necessitam de processamento de informação. Segundo Gotardo (2015), uma linguagem de programação é um método padronizado utilizado para expressar instruções de um programa para um computador programável.

Para um computador entender as instruções e processar, é necessário um programa de computador conhecido como compilador ou interpretador. Ele trabalha na tradução (ou interpretação) do código fonte para execução das instruções no computador. É também por meio da linguagem de programação que se especifica quais os dados que serão utilizados pelo computador, como esses dados serão tratados, armazenados e transmitidos.

2.2.6.1 A linguagem Python

A linguagem foi criada em 1990 por Guido Van Rossum, no Instituto Nacional de Pesquisa para Matemática e Ciência da Holanda (CWI), destinada para físicos e engenheiros. É um software de código aberto com licença compatível com a *General Public License* (GPL²), porém com menos restrições, permitindo que seja

¹ Licença de software que diz que os direitos dos autores originais devem ser mantidos, mas não incluem outras limitações para a licença permitindo até mesmo cópia de trechos de código fonte.

² GNU GPL ou simplesmente GPL, é a designação da licença de software idealizada por Richard Matthew Stallman em 1989, criada de acordo com as definições de software livre da Free Software Foundation.

incorporado em produtos proprietários. Atualmente, a linguagem é mantida pela *Python Software Foundation 2* (PSF).

Para Borges (2010), Python é uma linguagem de altíssimo nível (*Very High Level Language*), orientado a objetos, de tipagem e dinâmica forte, interpretada e interativa. Possui uma sintaxe clara e concisa, favorecendo a legibilidade do código fonte, tornando-a mais produtiva.

Python é uma linguagem interpretada, requerendo de um interpretador para execução do código fonte, conforme detalhado na Figura 11. O programa escrito é disponibilizado através do seu código fonte. Em seguida, um compilador é o responsável pela interpretação do seu código fonte. O compilador faz a tradução e disponibiliza um executável (código objeto), que pode ser executado e apresenta a saída do programa.

Figura 11 – Interpretação do código fonte Python



Fonte: Downey; Elkner; Meyers (2017).

Uma das principais vantagens da utilização de Python é sua vasta biblioteca padrão. Essa biblioteca possui diversos recursos poderosos que auxiliam na construção de um software. Esta característica foi primordial na escolha desta poderosa linguagem de programação para execução deste trabalho.

2.2.6.2 A linguagem Java

Segundo Furgeri (2010), foi lançada em 1995, pela Sun Microsystem, buscava ser simples, neutra, possibilitando a execução em diversos tipos de hardware, propiciando a ideia de portabilidade através da execução da mesma aplicação nos mais diversos sistemas operativos e/ou hardwares.

Para Junior (2013) e Sebesta (2011), Java é uma linguagem orientada a objetos que possui diversas características, destacam-se:

- Funcionalidades Web;

- Independência de plataforma;
- Alto desempenho;
- Segurança

A linguagem Java foi escolhida para desenvolvimento do *front-end* considerando a facilidade em: criar páginas ou telas com boa usabilidade e carregamento rápido, garantir o funcionamento nos diferentes navegadores, facilidade de integrar com os serviços do *back-end*.

3 PROJETO DO SOFTWARE

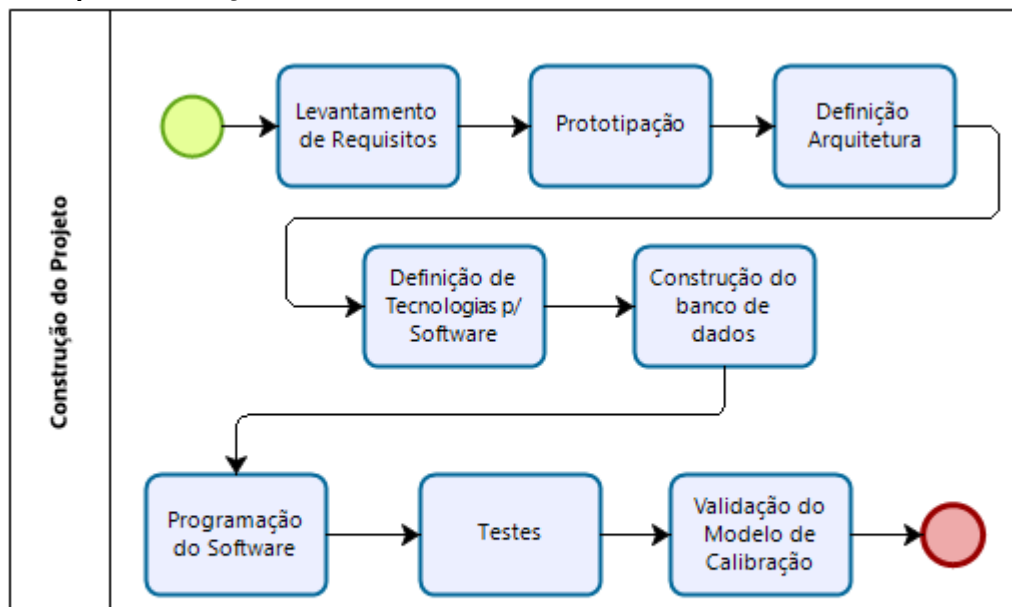
Nesta seção serão apresentadas as técnicas utilizadas para a construção do software, com informações sobre quais foram os artefatos produzidos, arquitetura utilizada, ambiente de desenvolvimento entre outros.

3.1 Metodologia de Desenvolvimento

Neste trabalho adaptou-se a metodologia Scrum visando melhor aproveitamento durante o desenvolvimento do projeto, por meio disso foi possível obter os artefatos necessários para a construção do software, e por meio do levantamento de requisitos foi possível perceber a necessidade de melhorias no processo de construção dos modelos multivariados, além da necessidade de manter os dados dos modelos multivariados em constante atualização.

A Figura 12 representa os ciclos que aconteceram durante o desenvolvimento do software.

Figura 12 – Etapas construção software



Conforme ilustrado na Figura 12, as próximas seções visam demonstrar os ciclos referentes a construção do software, onde serão apresentados os requisitos, protótipos de telas, arquitetura escolhida, tecnologias utilizadas no software, construção do banco de dados, codificação e testes. Enquanto a validação do modelo de calibração será apresentada na seção 4.1.

3.1.1 Levantamento de Requisitos

O primeiro ciclo de desenvolvimento que aconteceu foi o levantamento de requisitos. Por meio de entrevistas iniciais sobre qual era o real problema, e quais os principais requisitos que o software deveria atender.

Em seguida, foi necessário entender as técnicas de quimiometria. Nesta etapa, houve participação nas aulas da disciplina de Quimiometria do curso de Pós-graduação em Inovações Tecnológica com carga horária de 30 horas ministrada pela Professora Patrícia Valderrama.

3.1.1.1 Descrição dos requisitos de software

Os requisitos funcionais do software são apresentados no Quadro 1:

Quadro 1 - Descrição requisito funcional

Req.	Descrição Requisito
RF01	disponibilizar métodos quimiométricos desenvolvidos no sistema para seleção e montagem e processamento dos modelos.
RF02	permitir criar modelos de calibração multivariada.
RF03	permitir cadastrar as amostras de dados referente a cada modelo.
RF04	possibilitar que o usuário faça a calibração do modelo, sempre que necessário.
RF05	permitir que a entrada da amostra para análise seja por upload ou por captura da imagem por meio de dispositivo externo (webcam, câmera digital, celulares).
RF06	permitir ao usuário selecionar a área da imagem que deseja utilizar para a realização da análise.

RF07	redimensionar a imagem para o tamanho 70x70 pixels para realizar as análises.
RF08	montar o vetor RGB por meio dos pixels de uma imagem selecionada, sendo (70x70) x 3 pixels, onde 3 é pertinente ao canal de cor RGB.
RF09	disponibilizar os resultados do processamento em tela.
RF10	possibilitar a consulta das amostras processadas e os resultados do processamento.
RF11	realizar a predição de valores.

Fonte: Aatoria Própria (2019)

Os requisitos não funcionais do software são apresentados no:

Quadro 2 - Descrição requisitos não funcional

Req.	Descrição Requisito
RNF01	Para acesso ao sistema, o usuário deverá possuir conexão com a Internet, uma vez que o sistema é totalmente online
RNF02	Alta disponibilidade do sistema e dos serviços Web.
RNF03	Uso de design responsivo nas interfaces gráficas. A interface do sistema deverá se comportar adequadamente independente da forma de acesso (navegador, smartphone ou tablet) utilizada pelo usuário.
RNF04	Compatibilidade com navegadores Web mais comuns (MS Internet Explorer, Opera, Mozilla Firefox, Google Chrome, Safari).
RNF05	Construção de uma estrutura que possibilite atualizações futuras, com inclusão de funcionalidades inexistentes no projeto inicial
RNF06	Apresentação mensagens de erro que sejam claras e precisas.

Fonte: Aatoria Própria (2019)

3.1.1.2 Aquisição de imagens

As imagens utilizadas no modelo foram cedidas por uma indústria de extração de óleo de soja. Um total de 289 imagens foram obtidas na própria indústria no momento da análise pelo método de referência para determinar clorofila, conforme especificado em BASSETI (2019). Ainda, segundo BASSETI (2019), as imagens foram obtidas em ambiente climatizado a 20°C, com o uso de uma *webcam* Logitech® modelo C270, com capacidade de captura de até 3 megapixel, sendo padronizada uma distância de 20 cm (+/- 1 cm) entre amostra e *webcam*. A amostra foi acondicionada em um frasco branco com 20 mL de capacidade. A Figura 13 ilustra uma das imagens obtidas pelo sistema.

Figura 13 - Imagem Capturada Amostra Óleo de Soja



Fonte: Bruna Daniely Bassetti.(2018).

3.1.1.3 Processamento de imagem


Nesta seção, aborda-se resumidamente o processamento de imagem implementado no software. Descreve-se como a transferência do arquivo de imagem por meio do serviço Web é realizada e como a imagem é armazenada no banco de dados. Por fim, descreve-se como é o processamento da imagem para montar os dados para a **Matriz X**.

3.1.1.4 Transferência de Dados

As imagens foram convertidas para Base64 para sua transmissão. Este é um método para codificação de dados para transferência na Internet (MIME). Esse método é frequentemente utilizado na transmissão de dados binários em transmissões que lidam apenas com texto, tais como as que ocorrem no envio de e-mail. Nessa conversão a imagem é transformada em uma cadeia de caracteres delimitada à utilização de apenas 64 caracteres ([A-Za-z0-9], “/” e “+”).

O Quadro 3 exemplifica a codificação Base64 para a imagem ilustrada na Figura 12.

Quadro 3 - Exemplo Codificação Base64

Original	Texto em Base64
 (Imagem)	<pre> /9j/4AAQSkZJRgABAgAAAQABAAD/2wBDAAgGBgcGBQgHBwcJCQgKDBQNDAsLDBkSEw 8UHRofHh0aHBwgJ4nC1IslxwckDcpLDAxNDQ0Hyc5PTgyPC4zNDL/2wBDAQkJCQwLDBg NDRgyIRwhMjlyMjlyMjlyMjlyMjlyMjlyMjlyMjlyMjlyMjlyMjlyMjlyMjlyMjlyMjly PAA8DASIAAhEBAxEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QA tRAAAgEDAawIEAwUFBAAQAAAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAk M2JyggkKFhcYGRolJicoKSo0NTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1 dnd4eXqDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJyLT 1NXW19jZ2uHi4+TI5ufo6erx8vP09fb3+Pn6/8QAHwEAAwEBAQEBAQEBAQAAAAAAAA ECAwQFBgcICQoL/8QAtREAAgECBAQDBAcFBAQAAQJ3AAECAxEEBSExBhJBUQdhcRMiMo EIFEKRobHBCSMzUvAVYnLRChYkNOEl8RcYGRomJygpKjU2Nzg5OkNERUZHSElKU1RVV idYWVpjZGVmZ2hpanN0dXZ3eHl6goOEhYaHiImkkpOUlZaXmJmaoqOkpaanqKmqsrO 0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4uPk5ebn6Onq8vP09fb3+Pn6/9oADAMBAA IRAxEAPwD2DxrIzoGkxXVqkTyPOlyJQSMFWPyj0FwvdMpTavoNtfXCxrLLu3CMELwx HGSfSrGp6VZaxbrb38PnRK4cLuZcNjPBHYmpbGxttNs47S0j8uCPO1dxOMkk8nnqTXO oVfbubfuW28yrllt1P/2Q= </pre>

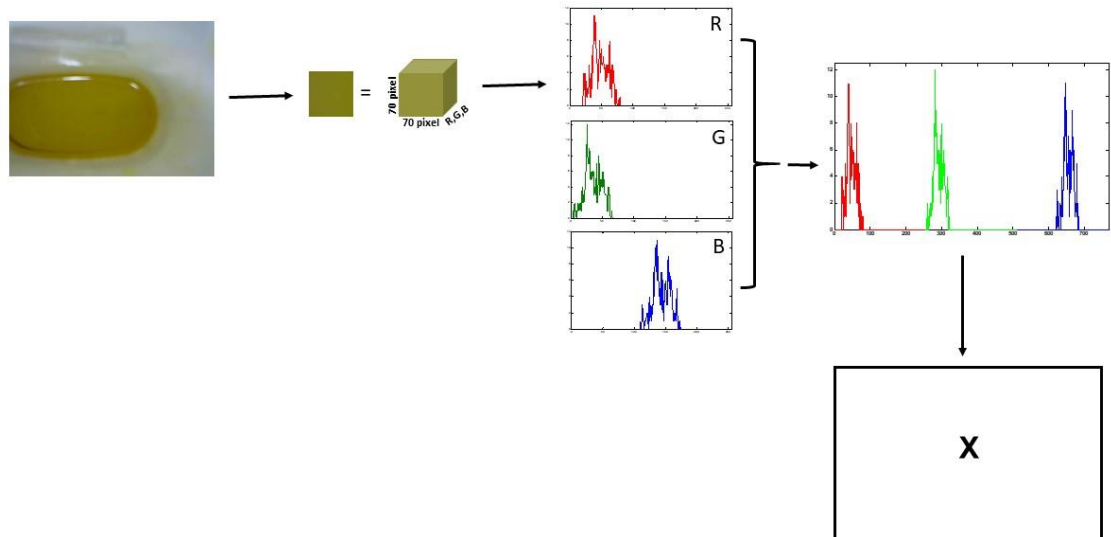
Fonte: Autoria Própria (2019)

3.1.1.5 Padrão Processamento (Imagem)

Para o processamento, o sistema redimensiona a imagem em uma resolução 70x70 pixels, extraindo posteriormente os valores dos canais R (Red) G (Green) e B (Blue), pixel a pixel (70x70x3), transformando-o em histogramas para

que os valores possam ser armazenados no banco de dados (**Matriz X**), conforme esquematizado na Figura 14.

Figura 14 – Esquema Construção da Matriz X a partir de uma imagem digital



Fonte: Adaptado de Valderrama e Valderrama (2016).

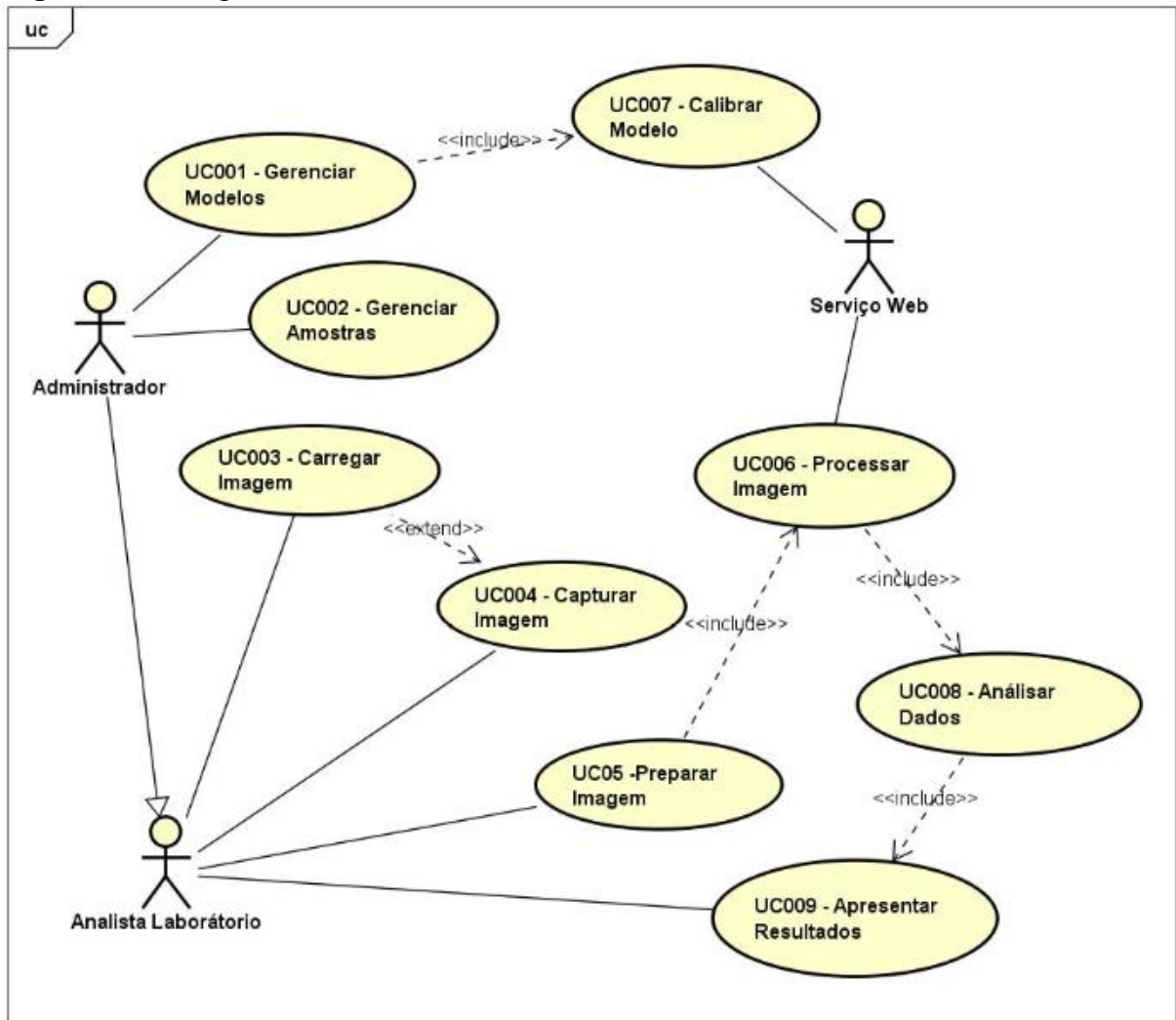
Visando demonstrar como as funcionalidades se relacionarão umas com as outras e como serão utilizadas pelos usuários uma das etapas do levantamento de requisitos é o diagrama de caso de uso.

3.1.1.6 Diagrama de caso de uso

Um diagrama de Casos de Uso contemplado na UML descreve os processos de negócio e suas interações com partes externas, tais como clientes, parceiros e serviços.

A Figura 15 apresenta o Diagrama de Caso de Uso do Software:

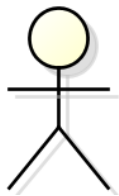
Figura 15 - Diagrama de caso de uso

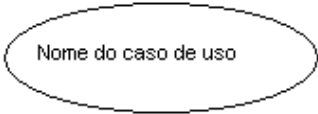


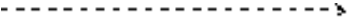



Fonte: Autoria Própria (2019)

Para entendimento do diagrama de Caso de Uso, o Quadro 4 apresenta uma legenda para seus componentes básicos (LIMA, 2005):

Quadro 4 - Legenda Diagrama Caso de Uso

 <p style="text-align: center;">ATOR</p>	<p>Ator desempenha o papel que tipicamente estimula/solicita ações/eventos do sistema e recebe reações. Cada ator pode participar de vários casos de uso.</p>
--	---

 <p>CASO DE USO</p>	<p>São os eventos feitos por um ator no uso de um sistema.</p>
 <p>GENERALIZAÇÃO</p>	<p>Um relacionamento de generalização pode acontecer entre atores ou caso de uso, uma classe de ator/caso de uso (descendentes) com outra classe de ator/caso de uso (ascendente) indicando que o descendente herda o papel que o ascendente assume.</p>
 <p>ASSOCIAÇÃO DE COMUNICAÇÃO</p>	<p>Uma associação de comunicação entre um caso de uso e um ator indica que uma instância do caso de uso e uma instância do ator devem interagir.</p>
<p><<EXTEND>></p>  <p>RELACIONAMENTO DE EXTENSÃO</p>	<p>Um relacionamento de extensão é aquele que se estabelece entre um caso de uso de extensão e um caso de uso base, especificando como o comportamento definido para o caso de uso de extensão pode ser inserido no comportamento definido para o caso de uso de base. É inserido implicitamente, ou seja, a extensão não é exibida no caso de uso base.</p>
<p><<INCLUDE>></p>  <p>RELACIONAMENTO DE INCLUSÃO</p>	<p>Um relacionamento de inclusão acontece entre um caso de uso base e um caso de uso de inclusão, especificando como o comportamento definido para o caso de uso de inclusão é inserido de forma explícita no comportamento definido para o caso de uso base.</p>

Fonte: Autoria Própria (2019)

No processo da criação do diagrama, foram identificados três atores:

- Analista de Laboratório: realizará a análise pelo método de referência para quantificar clorofila presente nas amostras de óleo de soja branqueado.
- Administrador do Sistema: herda todas as atribuições de um Analista de Laboratório. É o responsável pela fase de configuração de sistema do software.

- Serviço *Web*: responsável pelas técnicas de processamento necessárias para o resultado da análise.

Para cada caso de uso é necessária uma descrição textual complementar para auxiliar em sua interpretação. Em relação ao apresentado na Figura 15, cabe as seguintes descrições:

- **UC001 - Gerenciar Modelos:** neste caso de uso, o Administrador do Sistema deverá gerenciar o modelo de dados que o sistema executará no processo de análise, considerando que devem ser incluídos os métodos de aplicação habilitados no sistema e, conseqüentemente, montar o modelo de dados informando qual a sequência para execução das técnicas de processamento. Este caso de uso está relacionado ao RF02.
- **UC002 - Gerenciar Amostras:** o Administrador do Sistema incluirá as amostras que pertencem ao modelo de calibração. Este caso de uso está relacionado com RF03.
- **UC003 - Carregar Imagem:** o Analista de Laboratório acessa o software e define qual o tipo de carregamento de imagem que deseja fazer para iniciar o processamento, considerando que o software possibilitará que as imagens sejam incluídas tanto por *upload* quanto por captura da imagem direta, que pode ser realizada por meio de um periférico (câmera digital, *webcam*, outros) acoplado ao local em que o software está sendo acessado. Este caso de uso está relacionado com RF05.
- **UC004 - Capturar Imagem:** trata-se de uma extensão do UC003 para situações em que o dispositivo de captura de imagem estiver acoplado ao software. Visa permitir outra forma de carregamento de imagem, e este caso de uso está relacionado com RF05.
- **UC005 - Preparar Imagem:** após a conclusão do UC003, o Analista de Laboratório pode selecionar uma área específica da imagem ou manter o tamanho original tal como ela foi capturada. Este caso de uso está relacionado com RF06.
- **UC006 - Processar Imagem:** realizado após a conclusão do UC005, sendo executado pelo ator Serviço *Web*. Este caso de uso realiza o

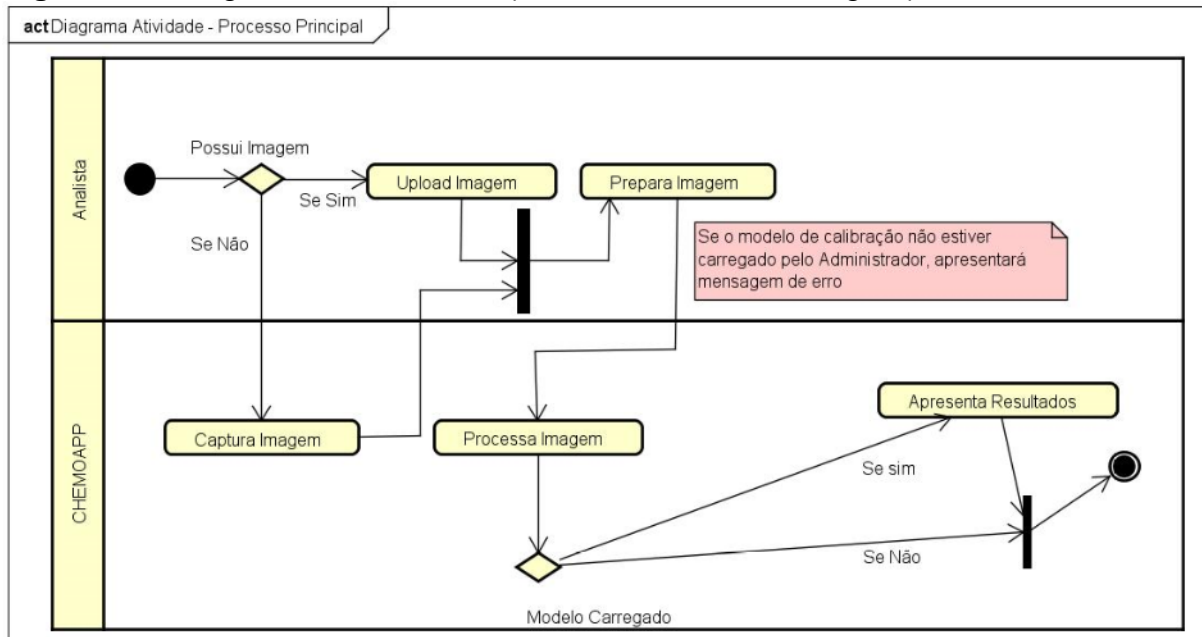
processamento da imagem capturando os valores RGB dos pixels e montando a matriz de dados. Este caso de uso está relacionado com RF07 e RF08.

- **UC007 – Calibrar Modelo:** trata-se de uma inclusão do UC001 considerando que é o Administrador do Sistema que deverá acionar a calibração do modelo. No entanto, a execução desse Caso de Uso acontecerá no Serviço *Web*. Este caso de uso está relacionado com RF02.
- **UC008 – Analisar Dados:** ocorre após a conclusão do UC006, sendo também executado pelo Serviço *Web*. Neste caso de uso, a técnica disposta no software é executada visando encontrar uma correlação entre os dados da imagem (matriz **X**) e propriedade de interesse (matriz **Y**). Este caso de uso está relacionado com RF11.
- **UC009 - Apresentar Resultados:** faz interação com dois atores: Serviço *Web* e Analista de Laboratório. Enquanto o Serviço *Web* tem a responsabilidade de fazer todo o processamento da análise, o Analista de Laboratório pode solicitar a versão impressa dos resultados. Este caso de uso está relacionado com RF09 e RF10.

3.1.1.7 Diagrama de Atividades

O software produzido neste trabalho visa auxiliar indústrias de processamento de óleo de soja na análise das amostras de óleo de soja branqueado por meio de imagens digitais. Com esse objetivo, o processo de análise é dividido em etapas: (1) captura da imagem, (2) processamento da imagem, (3) apresentação resultados. De posse dessas informações, um diagrama de atividade UML foi desenvolvido, conforme ilustrado na Figura 16.

Figura 16 - Diagrama de Atividade (Processamento de Imagem)



Fonte: Autoria Própria (2019)

Nesta figura há referências a dois módulos nas partições do diagrama de atividades. Tais módulos referem-se aos responsáveis das interações no processo de análise de imagem. O Analista de Laboratório pode carregar diretamente a imagem no sistema ou utilizar um dispositivo de captura de imagem interligado ao software.

Em seguida, o Analista de Laboratório prepara a imagem (caso seja desejável selecionar uma área específica da imagem) e, em seguida, o software realiza o processamento e busca os dados do modelo de calibração. Caso o modelo de calibração não ainda não exista, o sistema apresenta mensagem de erro e o processo é concluído. Caso contrário, o processamento da imagem é realizado e os resultados de acordo com as propriedades de interesse são apresentados, finalizando o processo.

Posteriormente, apresentou-se protótipos de telas a fim de validar a corretude do levantamento de requisitos.

3.1.2 Prototipação

A prototipação é uma etapa fundamental no desenvolvimento de software, influenciando diretamente na produtividade e na entrega final. Esta etapa tem como objetivo facilitar o entendimento dos requisitos de uma aplicação, permitindo aplicar conceitos e funcionalidades de software de uma forma simplificada.

Através desta etapa é possível propor uma solução adequada para o problema, aumentando a percepção do usuário em relação ao produto que será entregue.

Os protótipos são grandes aliados de metodologias ágeis, considerando que eles garantem um alinhamento entre equipe de desenvolvimento e cliente, necessariamente os protótipos não precisam estar idênticos ao produto, porém quanto mais fiel melhor para realizar ajustes.

Para o desenvolvimento deste software foi utilizado um tipo de prototipagem com baixo nível de fidelidade, os *wireframes*, que estão representados no Apêndice A deste documento.

3.1.3 Definição Arquitetura

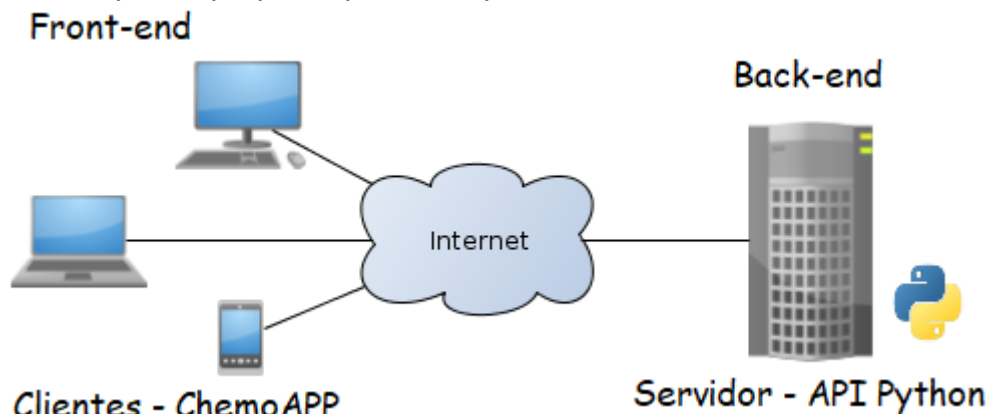
Uma das maiores preocupações no desenvolvimento deste software, era de uma arquitetura que atendesse as expectativas referente a performance tendo em vista o volume de informações necessárias para realizar a análise dos dados.

A arquitetura de um software consiste na definição dos seus componentes e de como estes devem estar organizados para que se possa utilizá-lo com eficiência. Neste caso, o projeto do software deste estudo foi desenhado para trabalhar em uma arquitetura do tipo cliente-servidor. Este tipo de arquitetura consiste em uma estrutura de aplicação distribuída que propicia a escalabilidade.

A principal vantagem da arquitetura cliente-servidor é a possibilidade de distribuir papéis e responsabilidades entre vários computadores dentro de uma rede, o que confere uma maior facilidade de manutenção. Além disso, os dados são armazenados nos servidores, que são responsáveis pelo controle de segurança de acesso aos dados.

A Figura 17 ilustra a arquitetura do software proposto. Neste caso, a estrutura está implementada em um servidor, e permite que os clientes (técnicos da indústria de óleo de soja) executem as tarefas, por meio de conexão da internet que é responsável pelo tráfego de informações.

Figura 17 - Esquema proposto para a arquitetura do software.



Fonte: Wikipedia (Adaptado) <https://pt.wikipedia.org/wiki/Cliente-servidor>

A arquitetura cliente servidor foi escolhida por satisfazer requisitos de acessibilidade, portabilidade, e também por permitir uma utilização escalável. Com isso partiu-se para a definição das tecnologias necessárias para construção do programa.

3.1.4 Definição de Tecnologias

Para o desenvolvimento da aplicação foi configurado um ambiente em que a IDE Pycharm³ Professional 2018.2 foi utilizada para a implementação do *back-end* escrito em Python. Apesar de ser uma versão proprietária, foi utilizada uma licença para fins acadêmicos válida até novembro de 2019 adquirida em nome do autor.

³ Ferramenta de desenvolvimento profissional (proprietária) para a linguagem Python, com um grande conjunto de ferramentas úteis para o desenvolvimento produtivo. Desenvolvida pela JetBrains. Disponível em <<https://www.jetbrains.com/pycharm/>>

Para o desenvolvimento *front-end* em Java foi utilizada IDE Netbeans⁴ que é uma ferramenta de desenvolvimento código aberto e oferece uma base sólida para desenvolvimento de sistemas, pois possibilita a criação de sistemas desktop, Web e dispositivos móveis por meio de diferentes linguagens como, por exemplo, Java e JavaScript.

Ferramentas auxiliares também foram utilizadas no processo de desenvolvimento. Exemplos são Dbdesigner⁵ 4.0.5.6, Balsamiq Mockups⁶ 3.5.16, ASTAH UML⁷ 7.0 e Bizagi Modeler⁸.

Atualmente, os desenvolvedores utilizam bibliotecas a fim de maximizar ou potencializar a produtividade na construção de sistemas. Estes subprogramas quando bem escolhidos, tornam-se essenciais para a construção de sistemas. Na próxima seção serão apresentadas as principais bibliotecas utilizadas na construção deste software.

3.1.4.1 Bibliotecas

No desenvolvimento de software, uma biblioteca é uma coleção de subprogramas que podem ser incorporados para a construção de um software. As bibliotecas possuem códigos auxiliares que atuam no processo de desenvolvimento.

Para o desenvolvimento do *back-end* Python, as seguintes bibliotecas foram utilizadas:

- Alembic 1.0.0: uma ferramenta de migração de banco de dados utilizada juntamente ao SQLAlchemy *Database Toolkit* Python.
- Click 6.7: pacote que auxilia na criação de interfaces por linha de comando, criando automaticamente página de ajuda para desenvolvedores.

⁴ Ferramenta de desenvolvimento, mantida pelo projeto *Apache Netbeans* que é uma parceria entre a Apache e Oracle. Disponível em <<https://netbeans.org/downloads/>>

⁵ Desenvolvido pela FabForce. Disponível em <<http://fabforce.eu/dbdesigner4/>>

⁶ Desenvolvido pela Balsamiq. Disponível em <<https://balsamiq.com/products/>>

⁷ Desenvolvido pela Apache Software Foundation. Disponível em <<http://www.apache.org/>>

⁸ Desenvolvido pela Bizagi. Disponível em <<http://www.bizagi.com>>

- Flask 1.0.2: framework escrito em Python que provê de um modelo de implementação para desenvolvimento Web.
- Flask-Migrate 2.2.1: extensão que manipula migrações de banco de dados SQLAlchemy para aplicativos Flask usando o Alembic.
- Flask-Script 2.0.6: fornece suporte para escrever scripts externos no Flask e inclui a execução de um servidor de desenvolvimento, possibilitando o desenvolvedor adicionar comandos extra.
- Flask-SQLAlchemy 2.3.2: extensão do Flask que adiciona suporte ao SQLAlchemy na aplicação.
- Gunicorn 19.9.0: servidor HTTP Python.
- Itsdangerous 0.24: assistente de transmissão de dados para ambientes não confiáveis, o qual possibilita recuperação de forma segura por meio de assinaturas que serão geradas para o tráfego dos dados.
- psycopg2-binary 2.7.5: adaptador do banco de dados PostgreSQL para Python.
- SQLAlchemy 1.2.11: biblioteca de mapeamento objeto-relacional SQL em código aberto, possibilita mapeamento entre tabelas de banco de dados com as classes Python.
- OpenCV (*Open Source Computer Vision Library*): biblioteca multiplataforma de código aberto que oferece mais de 350 algoritmos de visão computacional. Essa biblioteca foi utilizada no processamento de imagem auxiliando na montagem da matriz de RGB.
- Scikit-learn: foi necessário a utilização dessa biblioteca, tendo em vista que ela possui algoritmos voltado para aprendizado de máquina que são essenciais para a construção do software proposto, possuindo código fonte aberto e comercialmente utilizável (licença BSD¹). Possui algoritmos para métodos de classificação (tais como SVM, vizinhos mais próximos, etc.), regressão (SVR, regressão de rebordo), agrupamento (k-médias, agrupamento espectral, etc.), seleção de modelos, pré-processamento e redução de dimensionalidade (tais como PCA) e outros. Essa biblioteca é construída a partir de outras bibliotecas, dentre as quais, destaca-se:
 - NumPy: biblioteca em Python voltada para computação científica com o Python, com a mesma licença utilizada para Scikit-learn

- SciPy: biblioteca em Python com as mesmas características da NumPy, voltada para matemática, ciências e engenharia.
- Matplotlib: biblioteca de plotagem 2D que produz gráficos em diferentes formatos.

Para desenvolvimento do *front-end* em Java foram utilizadas as seguintes bibliotecas:

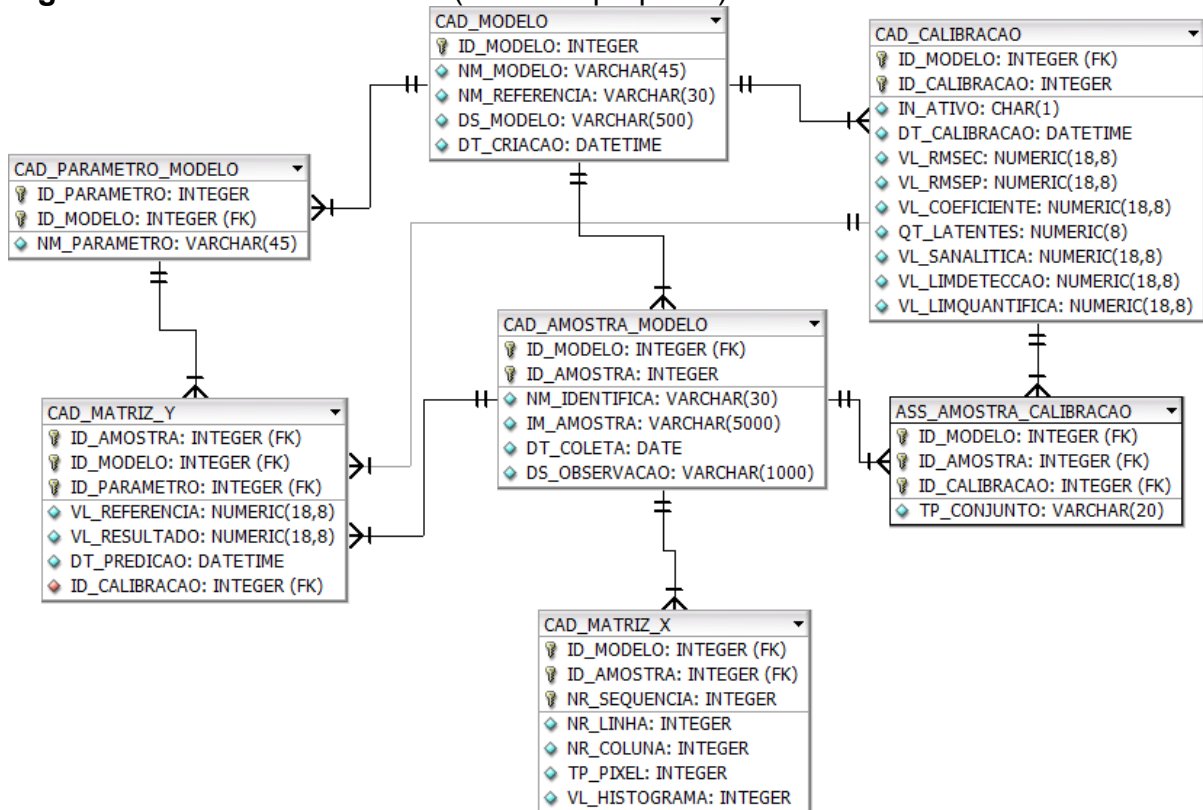
- JavaEE 7.0: especificação detalhada que demonstra como deve ser implementado um software.
- Jackson-mapper-asl 1.5.5: pacote para processamento Json.
- JSF 2.2.9: tecnologia para desenvolvimento *Web* que utiliza um modelo de interfaces gráficas baseado em eventos, ou seja, é um padrão de desenvolvimento que visa facilitar o trabalho dos desenvolvedores.
- Weld 2.4.1: implementação de referência para injeção de dependência na plataforma Java EE, ou seja, um padrão para injeção de dependência e gerenciamento de ciclo de vida contextual.
- Javax Servlet 3.1.0: responsável pelas páginas XHTML utilizada na plataforma JavaEE. Ela gerencia as requisições produzindo tais páginas dinamicamente.
- Primefaces 6.0: biblioteca de componentes ricos para aplicações criadas com JavaServer Faces. É um complemento do JSF que disponibiliza de componentes que deixam a aplicação mais atrativa.
- Omnifaces 2.5.1: biblioteca de utilitários JSF para auxiliar o desenvolvedor a fim de evitar criação de códigos validadores, conversores, etc.
- Lombok 1.16.10: biblioteca Java que, através de anotações, gera os métodos *getter*, *setter*, *equals*, etc.
- Retrofit 1.9.0: biblioteca REST que visa segurança na transferência de arquivos por meio de requisição HTTP.
- Commons-lang3 3.5: disponibiliza métodos utilitários para auxiliar no desenvolvimento de aplicações Java, tais como manipulação de cadeia de caracteres, métodos numéricos, entre outros.

- Gson 2.8.0: biblioteca da Google especializada na utilização de arquivos JSON.
- Slf4j-api: extensão do pacote Jakarta Commons Logging, utilizado para construção dos logs da aplicação.
- Jersey-Client 1.19.4: auxilia na montagem da comunicação da aplicação Java no lado Cliente com APIs REST.
- JQuery Ui: framework desenvolvido em JavaScript que conta com componentes gráficos, visando auxiliar na interação entre usuário e cliente.
- Font Awesome: conjunto de ferramentas de fontes e ícones com base em CSS e LESS.

3.1.5 Construção do Banco de Dados

O armazenamento dos dados gerados pelo sistema será por meio do Postgre SQL, para isso foi desenvolvido um projeto de banco de dados relacional representado na Figura 18. Para a construção do diagrama foi utilizada a ferramenta DB Designer 4 e a notação *crow's foot* para relacionamentos, enquanto as entidades são representadas por retângulos.

Figura 18 - Modelo Relacional (software proposto)



Fonte: Autoria Própria (2019)

As tabelas demonstradas na Figura 18, tem como base o modelo de calibração (CAD_MODELO), relacionado com as amostras que estão vinculadas ao modelo (CAD_AMOSTRA_MODELO) e os parâmetros vinculados ao modelo que são os objetos de interesse da **Matriz Y** (CAD_PARAMETRO_MODELO), e também os dados vinculados a **Matriz X** (CAD_MATRIZ_X) e a **Matriz Y** (CAD_MATRIZ_Y). Quando o modelo é calibrado os dados são criados na entidade (CAD_CALIBRACAO) e também uma tabela que associa as amostras na calibração (ASS_AMOSTRA_CALIBRACAO).

Para um entendimento das tabelas representadas na Figura 18, o dicionário de dados está representado no apêndice deste documento.

Definido todas as etapas necessárias para o desenvolvimento, então deu-se início a programação dos algoritmos do software.

3.1.6 Programação do Software

Esta seção visa apresentar os principais algoritmos utilizados na construção do software proposto. Os algoritmos demonstrados não são os mesmos construídos para o software em razão da técnica estar em processo de patente, porém foram utilizados os mesmos conceitos na construção dos algoritmos presentes no software proposto.

Para desenvolver o algoritmo de tratamento de imagem foi considerado muitos testes, visando melhoria em performance. Com isso foi escolhido a biblioteca openCV devido demonstrar bons resultados, para o tratamento de imagens foi definido que todas as imagens seriam redimensionadas para o tamanho de 70x70 pixel, conforme demonstrado:

Quadro 5 - Lógica redimensionar imagem

```
img = cv2.resize(oriimg, (70,70))
```

Fonte: Autoria Própria (2019)

O algoritmo apresentado no Quadro 5, executa o redimensionamento de imagens através da função *resize* da biblioteca openCV, onde utiliza-se da imagem original redimensionando-a em 70x70 pixels gerando uma nova imagem.

Posteriormente ao redimensionamento, é necessário capturar os valores para os canais de RGB que compõem os valores da **Matriz X**, a Figura 19 é um algoritmo que monta um histograma do canal B a fim de exemplificar o que foi desenvolvido no software proposto:

Figura 19 – Exemplo montar histograma Canal B (RGB)

```
# importing required libraries of opencv
import cv2

# importing library for plotting
from matplotlib import pyplot as plt

# reads an input image
img = cv2.imread('ex.jpg',0)

# find frequency of pixels in range 0-255
histr = cv2.calcHist([img],[0],None,[256],[0,256])

# show the plotting graph of an image
plt.plot(histr)
plt.show()
```

Fonte: <https://www.geeksforgeeks.org/opencv-python-program-analyze-image-using-histogram/>

A Figura 19 demonstra a plotagem de gráfico de histograma da imagem **ex.jpg** nesse algoritmo é utilizado a biblioteca openCV e também a matplotlib (plotagem de gráficos), a imagem é carregada no programa e então através de uma função `calcHist` do openCV, com os parâmetros de imagem, canal (0,1,2 para calcular histograma dos canais azul, verde e vermelho respectivamente), quando for utilizar a imagem completa o próximo parâmetro deve ser *none*, contagem de Bin e a gama de cores.

Para realizar a regressão foi utilizado o algoritmo PLS2 da biblioteca `scikit-learn`, conforme demonstrado na Figura 20.

Figura 20 - Algoritmo PLS2 exemplo Scikit Learn

```
>>> from sklearn.cross_decomposition import PLSRegression
>>> X = [[0., 0., 1.], [1.,0.,0.], [2.,2.,2.], [2.,5.,4.]]
>>> Y = [[0.1, -0.2], [0.9, 1.1], [6.2, 5.9], [11.9, 12.3]]
>>> pls2 = PLSRegression(n_components=2)
>>> pls2.fit(X, Y)
...
PLSRegression(copy=True, max_iter=500, n_components=2, scale=True,
               tol=1e-06)
>>> Y_pred = pls2.predict(X)
```

Fonte: [https://scikit-](https://scikit-learn.org/stable/modules/generated/sklearn.cross_decomposition.PLSRegression.html)

[learn.org/stable/modules/generated/sklearn.cross_decomposition.PLSRegression.html](https://scikit-learn.org/stable/modules/generated/sklearn.cross_decomposition.PLSRegression.html)

De acordo com a Figura 20, o algoritmo disponibilizado na documentação da biblioteca monta duas matrizes X e Y, declara o método de PLSRegression com 2 variáveis latente. Posteriormente, ajusta a Matriz X e Y (comando fit) e aplica o método PLS.

3.1.7 Testes

O teste do software é um processo que faz parte da construção de um software, o seu principal objetivo é revelar falhas para que sejam corrigidas até que o produto final atinja uma qualidade desejada.

Os testes do software proposto foram realizados durante o ciclo de desenvolvimento de código, foram executados testes unitários, integrados e de performance da aplicação pelo desenvolvedor.

Nessa etapa foram avaliadas diversas situações dentre as quais destacamos:

- Se os requisitos foram todos atendidos.
- Avaliação da interface.
- Testes de performance.

Todos os algoritmos construídos foram testados tanto individualmente, quanto combinados com outros componentes. Além disso, verificou-se a viabilidade com relação aos tempos de resposta para o momento da utilização da aplicação. Os testes executados alcançaram resultados satisfatórios.

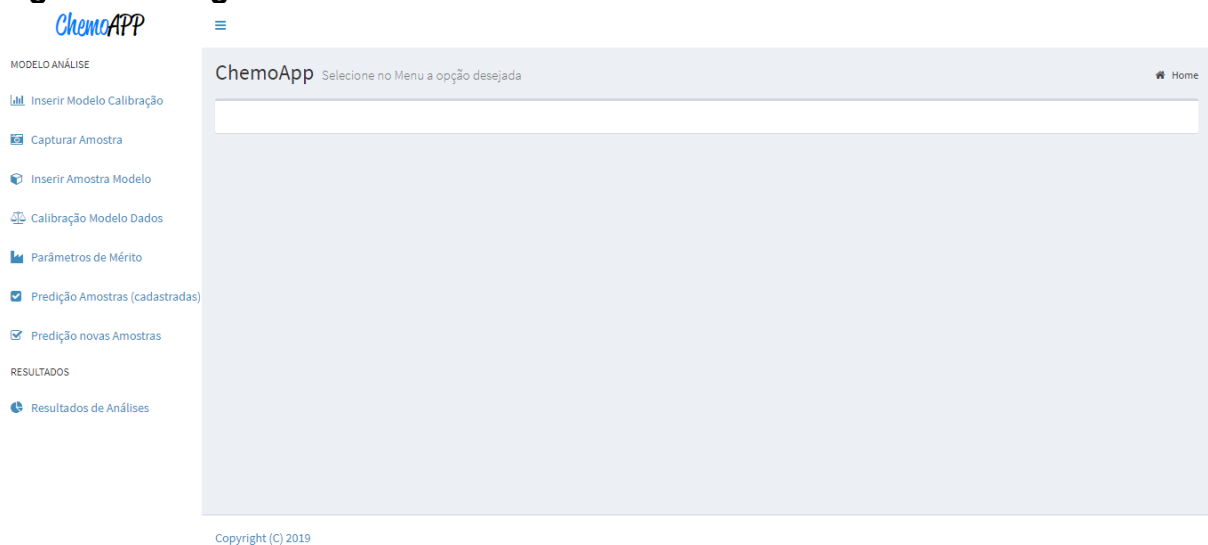
4 RESULTADOS E DISCUSSÃO

Neste capítulo são apresentados resultados obtidos com o software por meio da metodologia de quantificação de clorofila em óleo de soja branqueado desenvolvida por Basseti (2019). Além disso, será demonstrado o funcionamento do software e também será apresentado um comparativo entre os processos de quantificação com e sem o uso software.

4.1 Funcionamento do Software

A Figura 21 apresenta a tela inicial do software, com o menu lateral onde o usuário deverá escolher qual a ação que deseja executar.

Figura 21 - Página Inicial Software



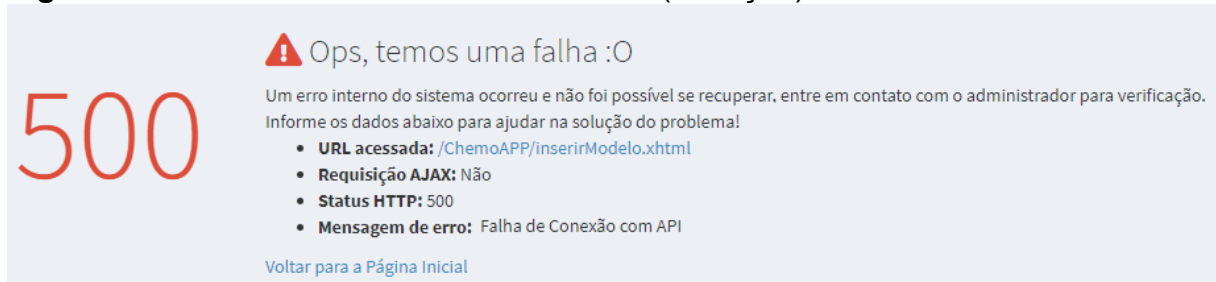
Fonte: Autoria Própria (2019)

O *front-end* do software foi desenvolvido em Java e é um projeto totalmente responsivo (as páginas se adaptam ao tamanho do navegador). O software foi homologado para uso em tablets e/ou dispositivos com resolução mínima de 1152x864 pixel. A interface principal conta com um menu lateral (expansível) possibilitando acesso à todas as telas do software como ilustrado na Figura 21.

A fim de auxiliar os usuários e desenvolvedores na resolução de problemas, o software conta com uma funcionalidade para tratamento de erros. A princípio foram catalogados os principais erros que podem acontecer e, então, criada uma

estrutura para auxílio na resolução destes. Essa funcionalidade visa minimizar o tempo de impacto aos usuários e ao desenvolvedor na correção dos problemas. A Figura 22 demonstra o layout criado para estrutura de tratamento de erros de exceção.

Figura 22 - Tratamento de Erros do Software (Exceção)



Fonte: Autoria Própria (2019)

Para a criação de um modelo de calibração, o usuário acessa ao software e seleciona a opção “Inserir Modelo Calibração”. Ao carregar a interface, o usuário informa um nome para identificar o modelo e qual o método de referência (possibilita entrada de informações complementares, caso necessário), além de qual o parâmetro que será analisado. Caso haja necessidade de se avaliar mais de um parâmetro, basta sair do campo “Nome Parâmetro” que o software possibilita o carregamento de mais uma linha em branco para alocar a nova informação. Ao concluir, deve-se clicar em “Salvar” para que o modelo seja registrado no banco de dados, conforme demonstrado na Figura 23.

Figura 23 - Cadastro Modelo Calibração

Fonte: Autoria Própria (2019)

Inicialmente, para criação do modelo de calibração, foram obtidas 289 imagens de amostras de óleo de soja branqueado, que foram registradas no banco de dados do software. As imagens que compõe o banco de dados são as mesmas que foram utilizadas por Basseti (2019), respeitando todos os passos da metodologia proposta.

Figura 24 - Imagem selecionada para uma das amostras



Fonte: Basseti (2019).

A Figura 24 representa a imagem antes do processamento realizado pelo software, quando o usuário delimita a área de imagem que será utilizada. Para inclusão de amostras no software, o usuário deve selecionar a opção “Inserir Amostra Modelo” e selecionar o Modelo de Dados no qual se deseja incluir a amostra, informando a “Data Amostragem” e “Identificação da Amostra”, considerando que muitas empresas utilizam códigos para identificar suas amostras dentro do laboratório. O campo “Descrição da Amostra” permite a informar qualquer informação adicional sobre a amostra, na sequência, é preciso “Escolher” a amostra desejada e informar os valores obtidos pelo método de referência. Por fim, clica-se em “Salvar”, conforme demonstrado na Figura 25.

Figura 25 - Inclusão Amostra Modelo de Dados

MODELO ANÁLISE

Inserir Modelo Calibração

Inserir Amostra Modelo

Predição de Resultados

RESULTADOS

Resultados de Análises

Inclusão Amostra no Modelo Carregar Amostra para o Banco de Dados

Home

Modelo de Dados/Instrumento Coletor: CLOROFILA - OLEO DE SOJA DEGOMADO / IMG

Data Amostragem: 16-07-2019

Identificação da Amostra: IMG 42047

Descrição da Amostra:
Amostra retirada para simulação do CHEMAAPP.

956 caracteres restantes.

Imagem da Amostra:
+ Escolher

Nome Parâmetro	Valor Resultado
% CLOROFILA	223.03

Salvar Limpar

Fonte: Autoria Própria (2019)

Quando o usuário toma a ação “Salvar”, demonstrado na Figura 25, o software inicia o processo de tratamento da imagem escolhida conforme descrito na Seção 3.1.1.3, e posteriormente ocorre a gravação dos dados. No total foram incluídas 289 amostras no banco de dados, posteriormente é necessário realizar a calibração do modelo de dados, dividindo as amostras em conjuntos de calibração e validação.

Para realizar a calibração de um modelo, o usuário seleciona no menu a opção “Calibração Modelo Dados”, preenche os dados conforme demonstrado na Figura 26 e clicar no botão “Calibração do Modelo”.

Figura 26 - Calibrar Modelo (Visão Responsiva)

The screenshot shows the ChemoAPP interface for data model calibration. At the top, the ChemoAPP logo is displayed. Below it is a hamburger menu icon. The main heading is "Calibração Modelo de Dados" with a subtitle "Realiza a Calibração Multivariada do Modelo de Dados". A "Home" button is visible. The main form area is titled "Modelo de Dados/Instrumento Coletor" and contains a dropdown menu with the selected value "CLOROFILA - OLEO DE SOJA DEGOMADO / IMG". Below this are three input fields: "Nr. Variável(s) Latente(s)" with a value of 20, "Quant. Remoção Outlier" with a value of 0, and "Valor Corte - Outlier" with a value of 0. At the bottom of the form are two buttons: "Calibração do Modelo" (green) and "Limpar" (red).

Fonte: Autoria Própria (2019)

Ao clicar “Calibração do Modelo” é executado o algoritmo de *Kennard-Stone* (KENNARD; STONE, 1969), onde as amostras são separadas na execução 187 amostras foram classificadas como calibração e 102 amostras como validação. Não foi removida nenhuma amostra anômala e foram utilizadas 20 variáveis latentes para a construção do modelo de calibração.

Os resultados apresentados foram satisfatórios na calibração do modelo, considerando-se a faixa de concentração utilizada no trabalho 0,10 – 477,18 ppb, conforme demonstrado na Figura 27.

Figura 27 – Tela Parâmetros de Mérito

Figuras de Mérito <small>Parâmetros de validação do modelo PLS</small>			
Modelo de Dados	Figuras de Mérito		
	Exatidão		Coeficiente de correlação
	RMSEC ^a	RMSEP ^a	
CLOROFILA - OLEO DE SOJA DEGOMADO	36.10	36.41	0.81

Fonte: Autoria Própria (2019)

A Figura 27 apresenta os resultados obtidos no software para os valores de erro médio quadrático da calibração (RMSEC) sendo 36,10 e previsão (RMSEP) 36,41 do modelo otimizado. Tais valores são indicadores de exatidão do modelo multivariado (VALDERRAMA; BRAGA; POPPI, 2009). Enquanto o coeficiente de correlação apresentou resultado de 0,81.

A metodologia empregada para construção do modelo, que empregou mínimos quadrados parciais (PLS, do inglês *Partial Least Squares*), sugere-se que o modelo possa ser melhorado pela adição de novas amostras, uma vez que o software possibilita tal operação, conforme demonstrado na Figura 26.

Ao término da calibração, o modelo foi validado a partir da previsão das amostras, separadas no conjunto de validação. No software, o usuário deve selecionar a opção “Predição novas amostras” e selecionar o modelo que deseja realizar a predição, informando a “Data Amostragem”, “Identificação da Amostra”. Além disso, é possível incluir informações referentes a amostra no campo “Descrição da Amostra”, e então “Escolher” a amostra desejada, e por fim clicar no botão “Analisar”, conforme demonstrado na Figura 28.

Figura 28 - Predição de resultado em amostra nova

Fonte: Autoria Própria (2019)

Ao término do processamento o resultado será apresentado em tela, conforme demonstrado na Figura 29.

Figura 29 - Resultado predição amostra nova

Fonte: Autoria Própria (2019)

A fim de validar o processo de predição, foi feito a previsão de 3 amostras cujo resultado já eram conhecidos (todas as amostras inclusas no modelo passaram pelo método de referência). Para entendimento segue os resultados no Quadro 6, onde apresenta os valores obtidos pelo método de referência e após aplicação no software.

Quadro 6 - Predição amostras valores conhecidos


Ident. Amostra	Valor – Método Ref.	Valor - Software
Amostra 1	228,22	224,53
Amostra 2	145,23	128,09
Amostra 3	51,90	70,10

Fonte: Autoria Própria (2019)

Os resultados apresentados no Quadro 6, são considerados bons considerando a diferença apresentada entre os resultados da análise pelo método de referência e com a utilização do software, pois os valores estão dentro do que foi calculado como erro conforme demonstrado na Figura 27.

No software quando o usuário desejar realizar a predição de amostras que estão cadastradas no sistema, deverá selecionar a opção “Predição amostras cadastradas”, selecionar o Modelo de Dados que deseja realizar a predição, informar o “Id. da Amostra”, e então clicar em “Buscar”. Conforme demonstrado na Figura 30.

Figura 30 - Predição amostras cadastradas

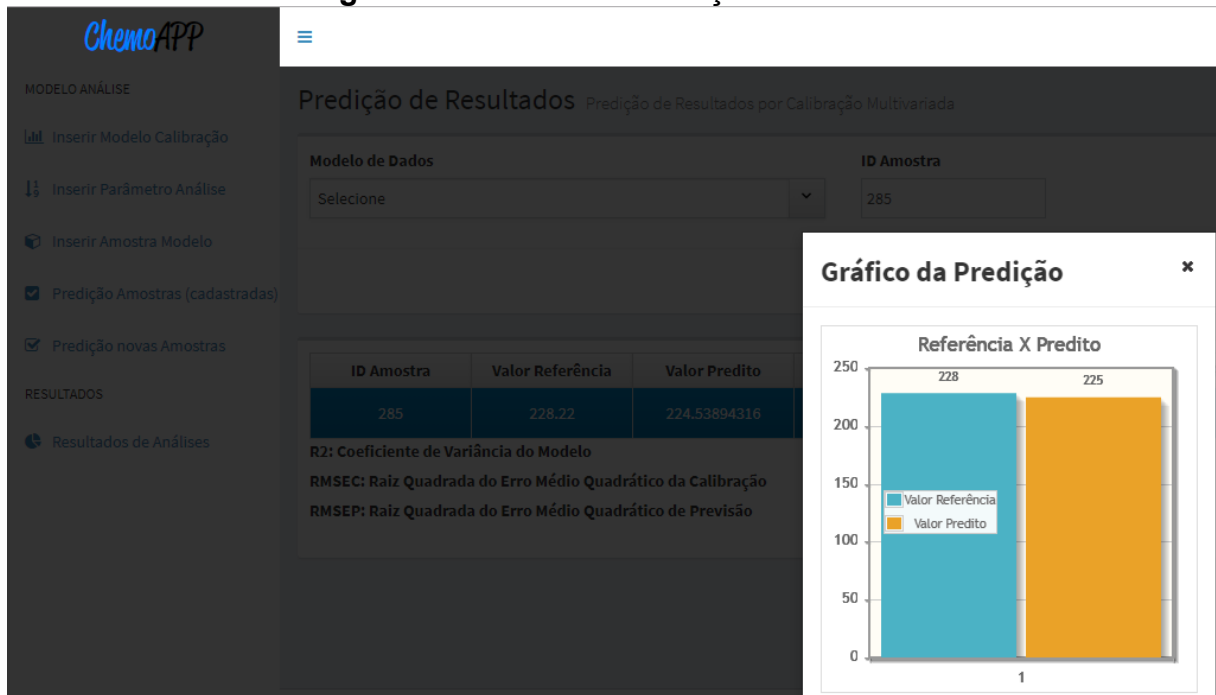


The screenshot shows a web application interface titled "Predição de Resultados" with a subtitle "Predição de Resultados por Calibração Multivariada" and a "Home" link. The main area contains two input fields: "Modelo de Dados" with a dropdown menu showing "CLOROFILA - OLEO DE SOJA DEGOMADO" and "ID Amostra" with a text input field containing "285". A blue "Buscar" button is located at the bottom right of the form.

Fonte: Autoria Própria (2019)

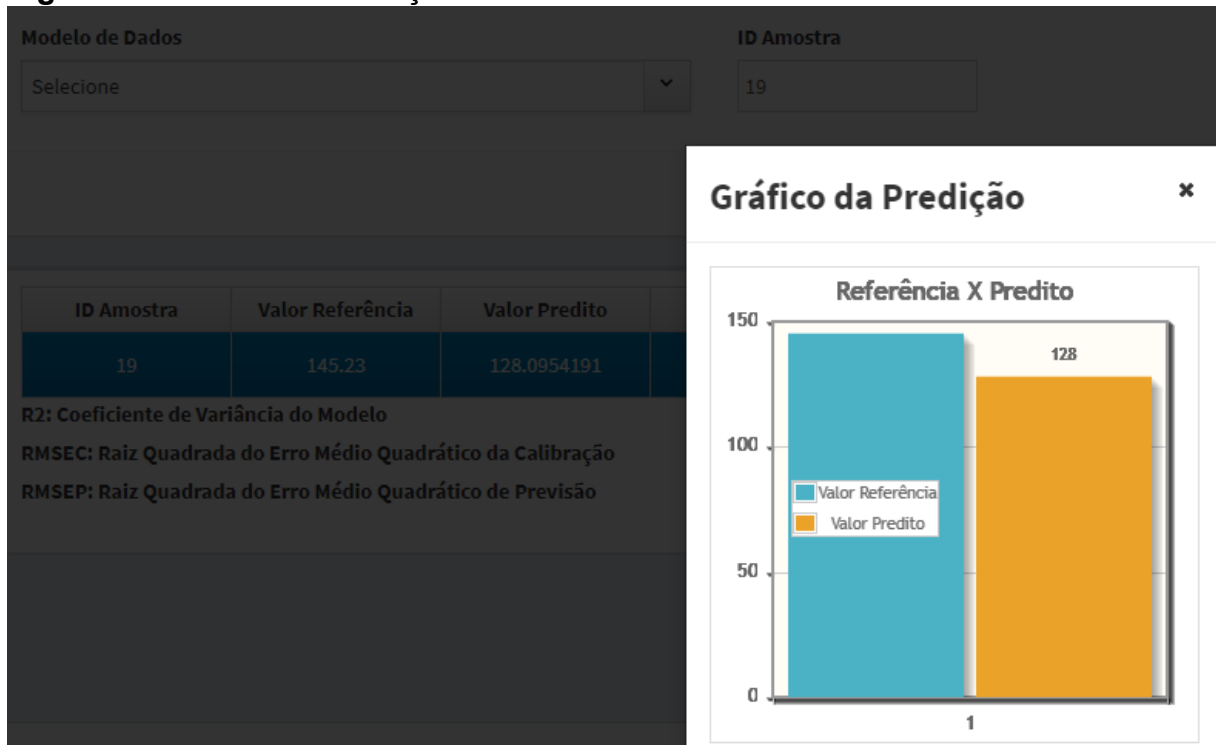
O resultado da predição será mostrado em tela, e então é possível visualizar um gráfico comparativo entre o método de referência e a predição do software. A seguir será demonstrado nas Figuras 31 a 29, o resultado de predições das amostras com valores conhecidos, para análise da margem de erro do software é apresentado em tela um gráfico comparativo.

Figura 31 - Resultado Predição Amostra 1

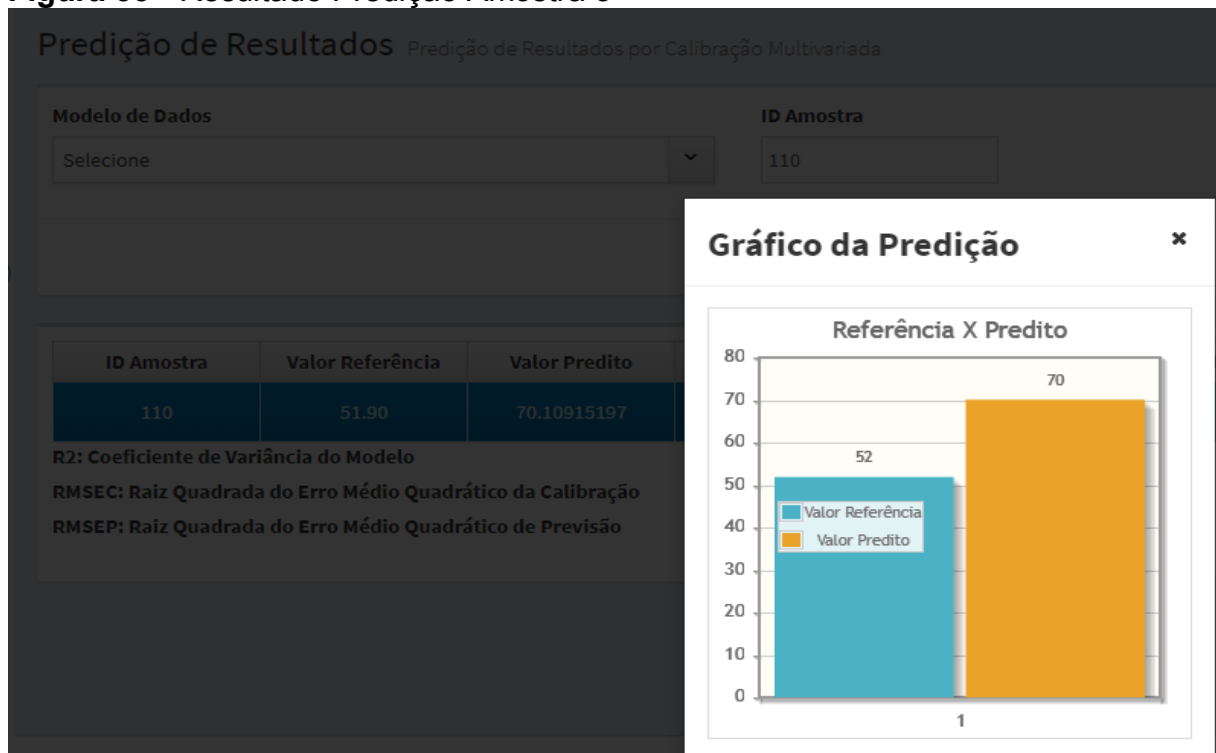


Fonte: Autoria Própria (2019)

Figura 32 - Resultado Predição Amostra 2



Fonte: Autoria Própria (2019)

Figura 33 - Resultado Predição Amostra 3

Fonte: Autoria Própria (2019)

Essas figuras ilustram resultados promissores na quantificação de clorofila na amostra de óleo de soja branqueado, considerando que ao acrescentar novas amostras e recalibrar o modelo é possível melhorar os resultados.

4.2 Benefícios do software em relação ao método manual

Os processos produtivos são essenciais para o sucesso dos negócios. A busca constante para diminuição de custos e para manter um alto índice de produção necessitam de um conjunto de processos bem estruturados para se manterem ativas no mercado.

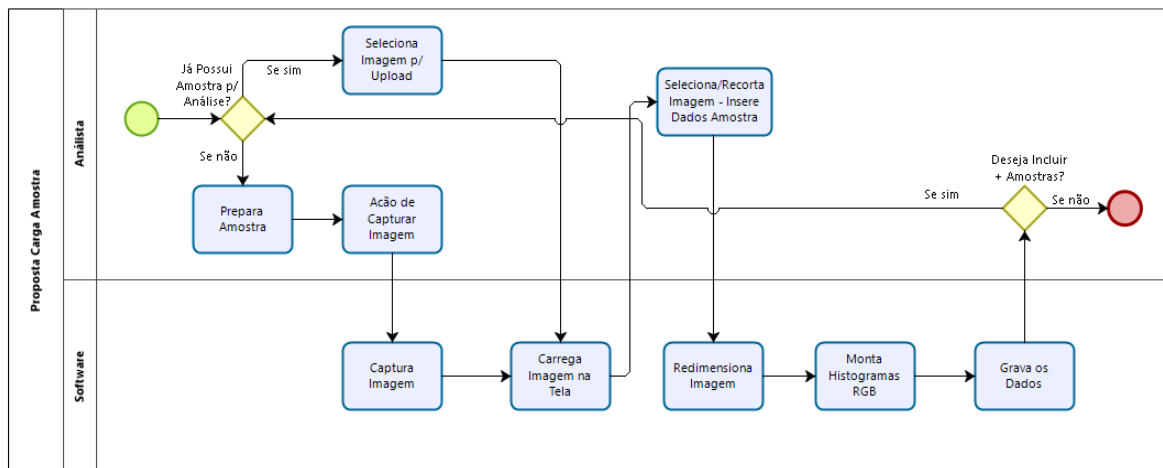
Com a introdução do software nas indústrias de óleo de soja branqueado, haverá uma otimização significativa no processo de quantificação de clorofila em óleo de soja branqueado comparando-se a metodologia proposta por Basseti (2019), reduzindo os desperdícios de recursos, tempo e custos.

O método explorado em Basseti (2019) é totalmente dependente da execução manual de um operador. Isso pode reduzir sua adoção na indústria devido ao custo operacional ou até mesmo inviabilizar a sua implantação. O software

apresentado neste trabalho tem como um dos objetivos viabilizar a adoção do método para quantificar clorofila em óleo de soja.

A fim de auxiliar no desenvolvimento e compreensão dos processos que são melhorados com a ferramenta de software proposta, foi desenvolvido diagramas de processos para demonstrar como ficou o processo automatizado. A Figura 34 demonstra o processo para carregamento de amostra no software.

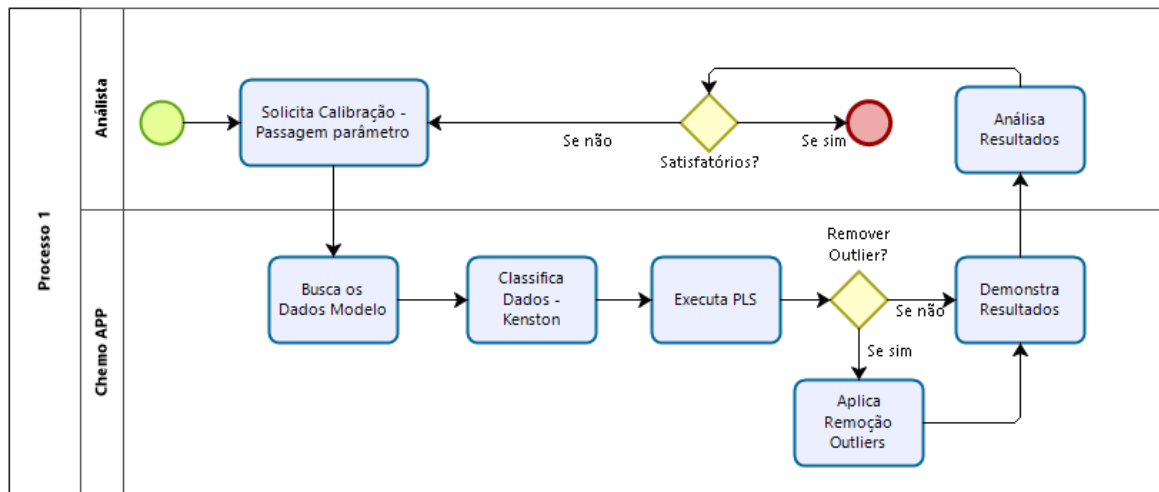
Figura 34 - BPMN carregamento amostra (Software)



Fonte: Autoria Própria (2019)

Enquanto a Figura 35 demonstra o processo de calibração multivariada que foi desenvolvido para o software proposto.

Figura 35 - BPMN calibrar modelo (Software)



Powered by
bizagi
Modeler

Fonte: Autoria Própria (2019)

Comparado o processo de inclusão de amostras proposto por Basseti (2019) e o software, na Figura 2 e Figura 34 respectivamente, é possível perceber o ganho produtivo em relação a interação do analista de laboratório com o software que foi minimizada. Assim como o processo da calibração multivariada demonstrada na Figura 5 e Figura 35 que também demonstram o antes e depois e apresentam as mesmas melhorias do processo anterior.

Analisando os processos apresentados é possível visualizar que houve diminuição na interação do usuário, considerando que foram empacotados dentro do software todas as técnicas necessárias para quantificar a clorofila em óleo de soja branqueado, e o usuário somente toma a ação de solicitar o processo, que será executado automaticamente. Tornando o processo mais rápido e confiável.

4.3 Validação dos métodos desenvolvidos

Nesta seção será apresentado um comparativo dos resultados apresentados pelo software proposto em relação a pesquisa de Basseti (2019), o objetivo é demonstrar que ambos os resultados são válidos de acordo com a literatura.

Quadro 7 - Comparativo resultados Basseti(2019) x Software

Parâmetros de Mérito	PLS Basseti (2019)	PLS Software
----------------------	--------------------	--------------

RMSECa	37,44	36,10
RMSEPa	36,96	36,41
Coefficiente de correlação	0,7505	0,81

Fonte: Autoria Própria (2019)

Segundo Basseti (2019, apud VALDERRAMA, BRAGA, POPPI, 2007; VALDERRAMA, BRAGA, POPPI, 2007b; FERREIRA, PALLONI, POPPI, 2013; SANTOS et al., 2015; SANTOS et al., 2018), mesmo com os resultados apresentados do coeficiente de correlação abaixo de 0,9 na literatura já houve trabalhos com resultados na ordem de 0,7 quando o método de referência são métodos laboriosos, que necessitam de muitas etapas.

Considerando a faixa de concentração utilizada em ambos os trabalhos, 0,10 – 337,14 ppb, suspeita-se que a forma que é feita o processamento da imagem na biblioteca openCV do Python otimizou o resultado apresentado pelo software.

Os resultados demonstram que o software atingiu o objetivo na construção de uma ferramenta de calibração multivariada confiável, tendo em vista que os resultados foram comparados com a ferramenta do software Matlab® que atualmente é uma das ferramentas mais consolidadas dentre os quimiometristas.

5 CONCLUSÃO

A união entre quimiometria e tecnologia da informação possibilitaram o desenvolvimento de um software contendo um modelo de calibração multivariada fundamentado no processamento de imagens digitais. O software desenvolvido permite que, por meio de um equipamento periférico capaz de adquirir imagens, determine-se o teor de clorofila em óleo de soja branqueado.

Conclui-se que há grande potencial para aplicação desta tecnologia em outras áreas das ciências, promovendo aprimoramento tecnológico com vantagens significativas para a medida de atributos de amostras.

O software desenvolvido neste estudo alcançou os objetivos propostos, considerando que:

- Produziu, a partir do método proposto em Basseti (2019), um software integrando os passos necessários para tornar viável a proposta de Basseti;
- A arquitetura escolhida para desenvolvimento atende as expectativas iniciais, uma vez que a funcionalidade do software está centralizada em um servidor, possibilitando implementação de novas interfaces que venham a consumir os serviços disponibilizados;
- O modelo de calibração utilizando-se o software desenvolvido permitiu a obtenção de um modelo de calibração mais eficiente que aquele demonstrado por Basseti a qual utilizou software Matlab e PLS toolbox;
- A construção do modelo de calibração a partir do software desenvolvido se mostrou tão eficiente quanto aquele desenvolvido por Basseti (2009),

Desta forma, é possível afirmar que o software proposto neste trabalho é um “Produto Viável Mínimo”, uma vez que os primeiros resultados apresentados são bastante encorajadores.

5.1 Perspectivas Futuras

Pretende-se produzir um aplicativo móvel que realize a coleta e predição das amostras, uma vez que o software foi totalmente estruturado visando portabilidade

para outras plataformas. Dentre as adaptações pretendidas, idealiza-se o controle de usuários, relatórios gerenciais, entre outros.

Sugere-se ainda a criação de um artefato físico para padronizar o processo de aquisição de imagens, onde a câmera fique acoplado a uma distância e posição correta das amostras, visando minimizar efeitos indesejados durante a coleta das imagens.

REFERÊNCIAS

- ABNT - ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6023:** informação e documentação: referências: elaboração. Rio de Janeiro, 2002. (modelo de referência com autoria coletiva).
- ALEXANDRINO, G. L.; POPPI, R. J. NIR imaging spectroscopy for quantification of constituents in polymers thin films loaded with paracetamol. **Analytica Chimica Acta**, v. 765, p. 37–44, 26 fev. 2013.
- AOCS. **Official Methods and Recommended Practices of the AOCS**. 6th. ed. 2009.
- ARAUJO, M. M. V. Indicadores de qualidade em grãos de soja submetidos a diferentes condições de armazenamento. 2014. 84 f. Dissertação (Mestrado em Agricultura Tropical). Universidade Federal de Mato Grosso, Faculdade de Agronomia e Medicina Veterinária, 2014.
- BASS, L., CLEMENTS, P., KAZMAN, R., 2003, **Software Architecture in Practice**, Second Edition, Addison Wesley.
- BASSETTI, B. D.; **Desenvolvimento de método alternativo para quantificação de clorofilas em indústria de óleo utilizando imagens digitais**. Universidade Tecnológica Federal do Paraná. Campo Mourão, 2019.
- BEEBE, K. R.; PELL, R. J.; SEASHOLTZ, M. B. **Chemometrics: a practical guide**, New York: John Wiley and Sons, Inc., 1998.
- BOCK, F. C., **Desenvolvimento de Métodos de Análise de Superfície e Calibração utilizando Imagens Digitais**. Dissertação Mestrado. Universidade de Santa Cruz do Sul: Rio Grande do Sul, 71 p., 2016.
- BORGES, L. E. **Python para Desenvolvedores**. 2ª Edição. Rio de Janeiro. 2010. Disponível em: <<https://ark4n.wordpress.com/python/>>. Acesso em 02 out. 2018.
- BORRMANN, D. **Efeito do déficit hídrico em características químicas e bioquímicas da soja e na degradação da clorofila, com ênfase na formação de metabólitos incolores**. TESE DE DOUTORADO. USP: São Paulo, 125 p., 2009.

BOTELHO, B. G.; MENDES, B. A. P.; SENA, M. M. Implementation of a robust method for quality inspection control of artisanal minas cheese. A multivariate metrological approach. **Química Nova**, v. 36, n. 9, p. 1416–1422, 2013.

BRETERON, R. G. Introduction to multivariate calibration in analytical chemistry. **The Analyst**, v. 125, n. 11, p. 2125–2154, 2000.

CRUZ, F. **Python: Escreva seus primeiros programas**. São Paulo: Casa do Código, 2015.

DAMASCENO, D.; TOLEDO, T. G.; GODINHO, M. S.; et al. Análise multivariada de imagens na química: um experimento para determinação do pH de águas potáveis. **Química Nova**, v. 38, n. 6, p. 836–841, 2015.

DIAS, M.S., VIEIRA, M.E.R., 2000, Software architecture analysis based on statechart semantics. In: **International Workshop on Software Specification and Design**, Washington, DC, USA.

DOEVEN, E. H.; BARBANTE, G. J.; HARSANT, A. J.; et al. Mobile phone-based electrochemiluminescence sensing exploiting the “USB On-The-Go” protocol. **Sensors & Actuators: B. Chemical**, v. 216, p. 608–613, 2015.

DOGLIO, F. **Pro REST API development with Node.js**. New York: Apress, 2015. ISBN 978-1-4842-0917-2.

DOWNEY, A.; ELKNER, J.; MEYERS, C. **Aprenda Computação com Python: Documentação Versão 1.1**. Disponível em: <https://aprendendo-computacao-com-python.readthedocs.io/en/latest/capitulo_01.html>. Acesso em 02 out. 2018.

DUARDA, L.; **Scrum e Métodos Ágeis: Um Guia Prático**, LuizTools, 2016

FERREIRA, M. M. C. **Quimiometria - Conceitos, Métodos e Aplicações**. 1. ed. Campinas-SP: Editora da UNICAMP, 2015.

FOCA, G.; MASINO, F.; ANTONELLI, A.; ULRICI, A. Prediction of compositional and sensory characteristics using RGB digital images and multivariate calibration techniques. **Analytica Chimica Acta**, v. 706, n. 2, p. 238–245, 2011.

FODOR, M. et al. Development of a rapid, non-destructive method for egg content determination in dry pasta using FT-NIR technique. **Journal of Food Engineering**, v. 107, n. 2, p. 195–199, 2011.

FERREIRA, D. S.; PALLONE, J. A. L.; POPPI, R. J. Fourier transform near-infrared spectroscopy (FT-NIRS) application to estimate Brazilian soybean [*Glycine max* (L.) Merrill] composition. **Food Research International**. 2013.

FERREIRA, E. C. **Softwares para cromatografia gasosa bidimensional abrangente (GCxGC)**. Campinas, SP, 2010.

FURGERI, S. **Java 7: Ensino Didático**. São Paulo: Érica, 2010.

GELADI, P. Chemometrics in spectroscopy. Part 1. Classical chemometrics. **Spectrochimica Acta Part B: Atomic Spectroscopy**, v. 58, n. 5, p. 767–782, maio 2003.

GELADI, P.; KOWALSKI, B. R.; CHEMISTRY, P. A. Partial least-squares regression: a tutorial. **Analytica Chimica Acta**, v. 185, p. 1–17, jan. 1986.

GELADI, P.; GRAHN, H. **Multivariate image analysis**. 1.ed. New York: John Wiley & Sons Ltd, 1996.

GODINHO, M. S.; PEREIRA, R.; RIBEIRO, K.; SCHIMIDT, F.; OLIVEIRA, A.; OLIVEIRA, S. Classificação de refrigerantes através de análise de imagens e análise de componentes principais (PCA). **Química Nova**, v. 31, n. 6, p. 1485-1489, 2008.

GOTARDO, R. **Linguagem de Programação I**. Rio de Janeiro: SESES, 2015.

HELLAND, I. S. On the structure of partial least squares regression. **Communications in Statistics - Simulation and Computation**, v. 17, n. 2, p. 581–607, 1988.

HELPER, G. A.; BOCK, F.; MARDER, L.; FURTADO, J. C.; COSTA, A. B.; FERRÃO, M. F. Chemostat: um software gratuito para análise exploratória de dados multivariados. **Química Nova**, v. 38, n. 4,, 2015. Disponível em <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-40422015000400575> acesso em 19 de Novembro de 2018.

HEUSER, C. A. **Projeto de banco de dados**: Volume 4 da Série Livros didáticos informática UFRGS. Bookman Editora, 2009.

HIBBERT, D. . Multivariate calibration and classification. **Chemometrics and Intelligent Laboratory Systems**, v. 70, n. 1, p. 95, jan. 2004.

JUNIOR, P. J. **Java: Guia do Programador Atualizado para Java 7**. 2. ed., São Paulo: Novatec, 2013.

KENNARD, R. W.; STONE, L. A. Computer Aided Design of Experiments. **Technometrics**, v. 11, n. 1, p. 137–148, 1969.

LEME, L. M. et al. Fast non-invasive screening to detect fraud in oil capsules. **LWT**, v. 109, p. 179–185, jul. 2019.

LI, G. J. et al. QTL Analysis for Dynamic Expression of Chlorophyll Content in Soybean (*Glycine max* L. Merr.). **Acta Agronomica Sinica**, v. 36, n. 2, p. 242–248, 2010.

LUCHI, G. **Uma visão geral do HTTP**. Disponível em <<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Overview>>. Acesso em: 28 de julho de 2019.

MAMIÁN-LÓPEZ, M. B.; POPPI, R. J. SERS hyperspectral imaging assisted by MCR-ALS for studying polymeric microfilms loaded with paracetamol. **Microchemical Journal**, v. 123, n. September, p. 243–251, nov. 2015.

MARÇO, P. H. et al. Investigation of the pH effect and UV radiation on kinetic degradation of anthocyanin mixtures extracted from *Hibiscus acetosella*. **Food Chemistry**, v. 125, n. 3, 2011.

MARQUES FILHO, O.; VIEIRA NETO, H. **Processamento digital de imagens**. Brasport Livros e Multimídia. Rio de Janeiro: 1999.

MILANI, A. **PostgreSQL - Guia do Programador**. Novatec. São Paulo - SP: 2008.

NETO, B. DE B.; SCARMINIO, I. S.; BRUNS, R. E. 25 anos de quimiometria no Brasil. **Química Nova**, v. 29, n. 6, p. 1401–1406, 2006.

NUNES, C. A.; FREITAS, M.; PINHEIRO, A.; BASTOS, S. Chemoface: a novel free user-friendly interface for chemometrics. **Journal of the Brazilian Chemical Society**, v. 23, n. 11, p. 2003-2010, 2012.

OLIVIERI, A. C.; BORTOLATO, S.; ALLEGRINI, F. Figures of Merit in Multiway Calibration. **Data Handling in Science and Technology**, v. 29, p. 541–575, 2015.

PAVIA, D. L. et al. **Introduction to spectroscopy**. Third edition, Brooks/Cole – Thomson Learning, US, 2001.

PRESSMAN, ROGER S. **Engenharia de Software**. 7^a. ed. PORTO ALEGRE: AMGH, 2011.

SANTOS, L. R. **Métodos alternativos para análise rápida de parâmetros de qualidade da soja**. Dissertação (Mestrado) – Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Tecnologia de Alimentos. Campo Mourão, 2017.

SCHWABER, K.; SUTHERLAND, J. **Um guia definitivo para o Scrum: As regras do jogo**, 2013. Disponível em:
<https://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf>

SEASHOLTZ, M. B.; KOWALSKI, B. The parsimony principle applied to multivariate calibration. v. 277, p. 165–177, 1993.

SEBESTA, R. W. **Conceitos de linguagem de programação**. 4.ed. Rio de Janeiro: Alta Books, 2000.

SEBESTA, R. **Conceitos de Linguagem de Programação**. 9. ed., Porto Alegre: Bookman, 2011.

SILVA, M. A. DE M. **Desenvolvimento e Validação de Método Quimiométrico para Determinação de Amoxicilina em Formulação Farmacêutica**. [s.l: s.n.].

SILVA, K. A. M da. **Aplicação de Sistema Computacional para a Classificação de Sementes de Soja por Tamanho Utilizando Processamento Digital de Imagens**. 2017. Universidade Federal do Rio Grande do Norte. Unidade Especializada em Ciências Agrárias. Tecnólogo em Análise e Desenvolvimento de Sistemas. Macaíba, 2017.

SOMMERVILLE, I. I. Tradução de IVAN BOSNIC e KALINKA G. DE O. GONÇALVES. 9ª. ed. SÃO PAULO: PEARSON, 2011.

SOUZA, A. M.; POPPI, R.J. Experimento didático de quimiometria para análise exploratória de óleos vegetais comestíveis por espectroscopia no infravermelho médio e análise de componentes principais: um tutorial, parte I. **Química Nova**, v. 35, n. 1, p.223-229, 2012.

SOUZA, A. M. DE et al. Teaching experiment of chemometrics for multivariate calibration in determination of paracetamol in commercial tablets using Near-Infrared Spectroscopy: A tutorial, part II. **Química Nova**, v. 36, n. 7, p. 1057–1065, 2013.

VALDERRAMA, P. **Calibração Multivariada de Primeira e Segunda Ordem e Figuras de Mérito na Quantificação de Enantiômeros por Espectroscopia**. [s.l.] Tese - Universidade Estadual de Campinas. Instituto de Química. 219 p., 2009., 2009.

VALDERRAMA, P.; BRAGA, J. W. B. State of the art of figures of merit in multivariate calibration. **Química Nova**, v. 32, n. 5, p. 1278–1287, 2009.

VALDERRAMA, L.; VALDERRAMA, P. **Nondestructive identification of blue pen inks for documentoscopy purpose using iPhone and digital image analysis including an approach for interval confidence estimation in PLS-DA models validation**. CHEMOMETRICS AND INTELLIGENT LABORATORY SYSTEMS, v. 156, p. 188-195, 2016.

VALDERRAMA, P. **Avaliação de figuras de mérito em calibração multivariada na determinação de parâmetros de controle de qualidade em indústria alcooleira por espectroscopia no infravermelho próximo**. Campinas, SP: [s.n], 2005.

VALDERRAMA, P.; BRAGA, J. W. B.; POPPI, R.J. Validation of multivariate calibration models in the determination of sugar cane quality parameters by near infrared spectroscopy. **Journal of the Brazilian Chemical Society**, v. 18, n. 2, p. 259-266, 2007.

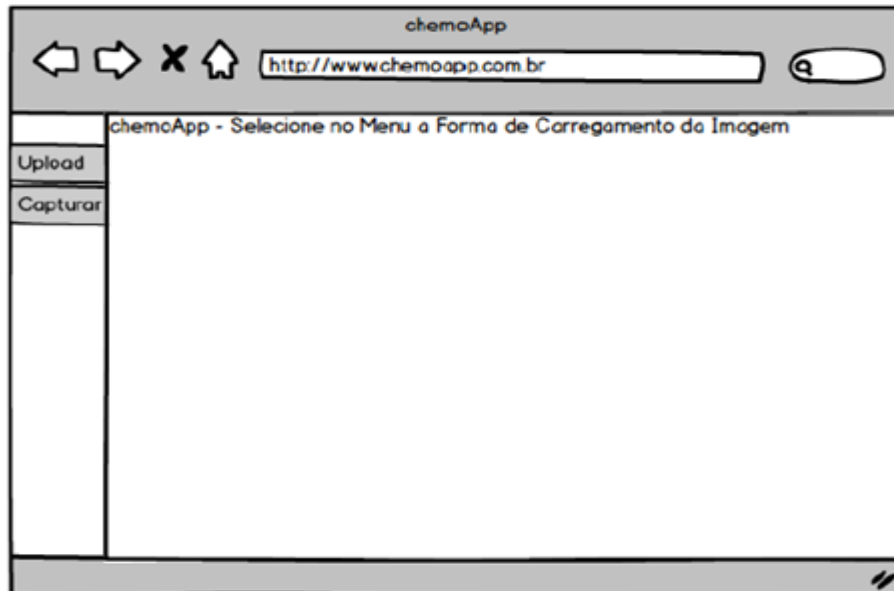
VERZELLO, Robert J.; REUTTER III, John **Processamento de dados**. São Paulo: McGraw-Hill, 1984.

VIANA, G.; **O que é front-end e back-end**. Disponível em <<https://www.treinaweb.com.br/blog/o-que-e-front-end-e-back-end/>>. Acesso em: 28 de Julho 2019

W3C. In: World Wide Consortium. Disponível em <<http://www.w3.org/>>. Acesso em: 17 de novembro de 2018.

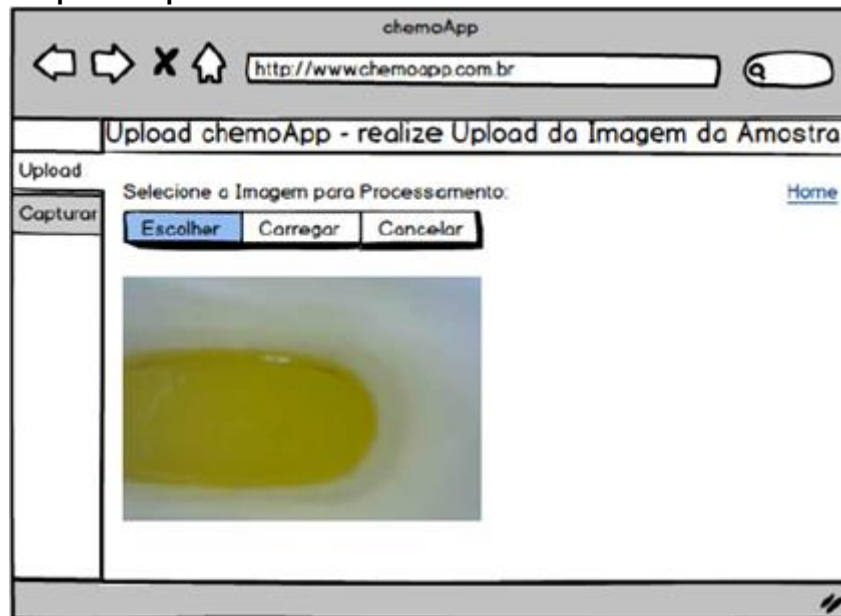
APÊNDICE A – Protótipos de Telas

Figura 36 - Protótipo tela inicial



Fonte: Autoria Própria (2018)

Figura 37 - Protótipo tela upload



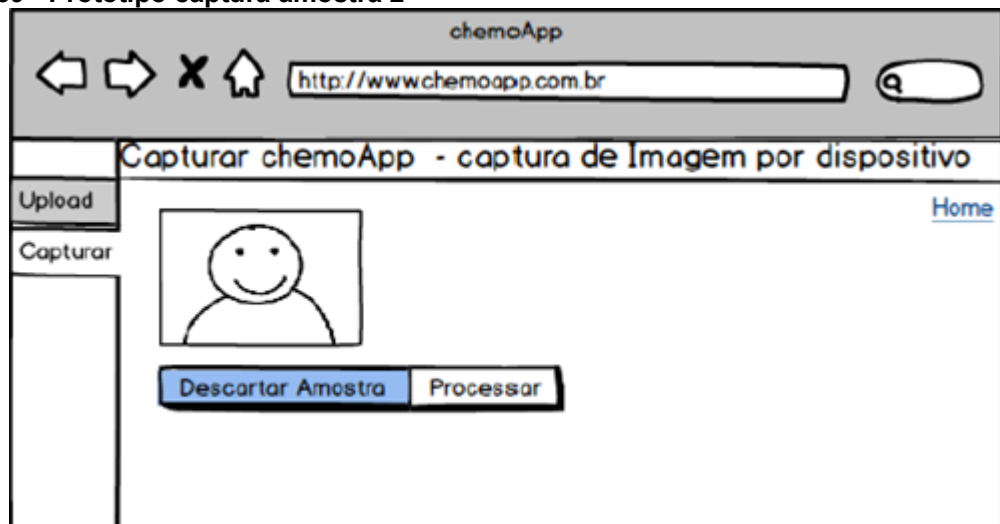
Fonte: Autoria Própria (2018)

Figura 38 - Protótipo captura imagem 1



Fonte: Autoria Própria (2018)

Figura 39 - Protótipo captura amostra 2



Fonte: Autoria Própria (2018)

APÊNDICE B – Dicionário de Dados

As relações (tabelas) da Figura 18 são descritas por meio dos dicionários de dados apresentados na Tabela 1 a 7. Para entendimento das tabelas, considere que:

- nome coluna: apresenta o nome que o atributo recebe no banco de dados.
- tipo dado: refere-se ao tipo de dado atribuído no banco de dados.
- chave primária: identifica qual atributo é a chave primária da tabela de acordo com indicador (sim/não).
- obrigatório: representa quais os atributos são obrigatórios na tabela de banco de dados de acordo com indicador (sim/não).
- descrição: representa o nome físico do atributo.
- auto-inc: indica que o atributo recebe auto-incremento automático pelo sistema, de acordo com o indicador (sim/não).

Além disso, existe a representação dos índices do banco de dados apresentada na mesma tabela.

Tabela 1 - Dicionário de dados ASS_AMOSTRA_CALIBRACAO

ASS_AMOSTRA_CALIBRACAO

Associativa amostra e calibração. Essa entidade guarda as informações referente a aplicação do algoritmo de Kennard-Stonne. Classifica as amostras em calibração, validação ou outliers.

Nome Coluna	Tipo Dado	Chave Primária	Obrigatório	Descrição	AutoInc
ID_MODELO	INTEGER	Sim	Sim	Identificador do modelo de calibração.	Não
ID_AMOSTRA	INTEGER	Sim	Sim	Identificador da amostra do modelo.	Não
ID_CALIBRACAO	INTEGER	Sim	Sim	Identificador da calibração.	Não
TP_CONJUNTO	VARCHAR(20)	Não	Sim	Tipo da amostra no conjunto de calibração. Classifica: Validação, Calibração ou Outlier.	Não

Nome do Índice	Tipo de Índice	Coluna(s)
Chave Primária	Chave Primária	ID_MODELO ID_AMOSTRA ID_CALIBRACAO
CAD_AMOSTRA_MODELO_has_CAD_CALIBRACAO_FKIndex1	Index	ID_MODELO ID_AMOSTRA
CAD_AMOSTRA_MODELO_has_CAD_CALIBRACAO_FKIndex2	Index	ID_MODELO ID_CALIBRACAO

Fonte: Autoria Própria (2019)

Tabela 2 - Dicionário de dados CAD_AMOSTRA_MODELO**CAD_AMOSTRA_MODELO**

Cadastro das amostras referente ao modelo de calibração

Nome Coluna	Tipo Dado	Chave Primária	Obrigatório	Descrição	AutoInc
ID_MODELO	INTEGER	Sim	Sim	Identificador do modelo de calibração.	Não
ID_AMOSTRA	INTEGER	Sim	Sim	Identificador da amostra do modelo.	Sim
NM_IDENTIFICA	VARCHAR(30)	Não	Sim	Nome identificador da amostra na empresa.	Não
IM_AMOSTRA	VARCHAR(5000)	Não	Sim	Imagem da amostra em Base64.	Não
DT_COLETA	DATE	Não	Não	Data de coleta da amostra.	Não
DS_OBSERVACAO	VARCHAR(1000)	Não	Não	Informações adicional da amostra.	Não

Nome do Índice	Tipo de Índice	Coluna(s)
Chave Primária	Chave Primária	ID_MODELO ID_AMOSTRA
CAD_AMOSTRA_MODELO_FKIndex1	Index	ID_MODELO

Fonte: Autoria Própria (2019)

Tabela 3 - Dicionário de dados CAD_CALIBRACAO**CAD_CALIBRACAO**

Cadastro da calibração e figuras de mérito.

Nome Coluna	Tipo Dado	Chave Primária	Obrigatório	Descrição	AutoInc
ID_MODELO	INTEGER	Sim	Sim	Identificador do modelo de calibração.	Não
ID_CALIBRACAO	INTEGER	Sim	Sim	Identificador da calibração.	Sim
IN_ATIVO	CHAR(1)	Não	Sim	Indicador de Ativo/Inativo.	Não
DT_CALIBRACAO	DATETIME	Não	Sim	Data da calibração do modelo.	Não
VL_RMSEC	NUMERIC(18,8)	Não	Sim	Valor do RMSEC ^a .	Não
VL_RMSEP	NUMERIC(18,8)	Não	Sim	Valor do RMSEP ^a .	Não
VL_COEFICIENTE	NUMERIC(18,8)	Não	Sim	Valor do Coeficiente de Correlação.	Não
QT_LATENTES	NUMERIC(8)	Não	Sim	Quantidade de Variável Latente.	Não
VL_SANALITICA	NUMERIC(18,8)	Não	Não	Valor da Sensibilidade Analítica.	Não
VL_LIMDETECCAO	NUMERIC(18,8)	Não	Não	Limite de detecção do modelo.	Não
VL_LIMQUANTIFICA	NUMERIC(18,8)	Não	Não	Limite de quantificação do modelo.	Não

Nome do Índice	Tipo de Índice	Coluna(s)
Chave Primária	Chave Primária	ID_MODELO ID_CALIBRACAO
CAD_CALIBRACAO_FKIndex1	Index	ID_MODELO

Fonte: Autoria Própria (2019)

Tabela 4 - Dicionário de dados CAD_MATRIZ_X**CAD_MATRIZ_X**

Cadastra as informações referente a Matriz X do modelo de calibração.

Nome Coluna	Tipo Dado	Chave Primária	Obrigatório	Descrição	AutoInc
ID_MODELO	INTEGER	Sim	Sim	Identificador do modelo de calibração.	Não
ID_AMOSTRA	INTEGER	Sim	Sim	Identificador da amostra do modelo de calibração.	Não
NR_SEQUENCIA	INTEGER	Sim	Sim	Número sequencial para ordenar a amostra.	Sim
TP_PIXEL	INTEGER	Não	Sim	Identifica o tipo de pixel do histograma. 1 - R, 2 - G e 3 - B	Não
VL_HISTOGRAMA	INTEGER	Não	Sim	Valor referente ao histograma.	Não

Nome do Índice	Tipo de Índice	Coluna(s)
Chave Primária	Chave Primária	ID_MODELO ID_AMOSTRA NR_SEQUENCIA
CAD_MATRIZ_X_FKIndex1	Index	ID_MODELO ID_AMOSTRA

Fonte: Autoria Própria (2019)

Tabela 5 - Dicionário de dados CAD_MATRIZ_Y

CAD_MATRIZ_Y

Cadastra as informações referente a Matriz Y do modelo de calibração.

Nome Coluna	Tipo Dado	Chave Primária	Obrigatório	Descrição	AutoInc
ID_MODELO	INTEGER	Sim	Sim	Identificador do modelo de calibração.	Não
ID_AMOSTRA	INTEGER	Sim	Sim	Identificador da amostra no modelo de calibração.	Não
ID_PARAMETRO	INTEGER	Sim	Sim	Identificador do parâmetro na Matriz Y.	Não
VL_REFERENCIA	NUMERIC(18,8)	Não	Não	Valor do resultado do teste na amostra quando aplicado método de referencia.	Não
VL_RESULTADO	NUMERIC(18,8)	Não	Não	Valor do resultado da predição.	Não
DT_PREDICAO	DATETIME	Não	Não	Data em que foi realizado a predição no sistema.	Não
ID_CALIBRACAO	INTEGER	Não	Não	Identificador da calibração utilizada para predição.	Não

Nome do Índice	Tipo de Índice	Coluna(s)
Chave Primária	Chave Primária	ID_MODELO ID_AMOSTRA ID_PARAMETRO
CAD_MATRIZ_Y_FKIndex1	Index	ID_MODELO ID_AMOSTRA
CAD_MATRIZ_Y_FKIndex2	Index	ID_PARAMETRO ID_MODELO
CAD_MATRIZ_Y_FKIndex3	Index	ID_MODELO ID_CALIBRACAO

Fonte: Autoria Própria (2019)

Tabela 6 - Dicionário de dados CAD_MODELO

CAD_MODELO

Cadastro do modelo de calibração.

Nome Coluna	Tipo Dado	Chave Primária	Obrigatório	Descrição	AutoInc
ID_MODELO	INTEGER	Sim	Sim	Identificador modelo de calibração.	Sim
NM_MODELO	VARCHAR(45)	Não	Sim	Nome identificador modelo de calibração.	Não
NM_REFERENCIA	VARCHAR(30)	Não	Sim	Nome do método de referência para obtenção das amostras.	Não
DS_MODELO	VARCHAR(500)	Não	Sim	Informações adicionais do modelo de calibração.	Não
DT_CRIACAO	DATETIME	Não	Sim	Data cadastro do modelo de calibração.	Não

Nome do Índice	Tipo de Índice	Coluna(s)
Chave Primária	Chave Primária	ID_MODELO

Fonte: Autoria Própria (2019)

Tabela 7 - Dicionário de dados CAD_PARAMETRO_MODELO**CAD_PARAMETRO_MODELO**

Cadastro dos parâmetros que serão vistos na Matriz Y. O Identificador do parâmetro representa a sequência dos valores na Matriz Y.

Nome Coluna	Tipo Dado	Chave Primária	Obrigatório	Descrição	AutoInc
ID_MODELO	INTEGER	Sim	Sim	Identificador do modelo calibração.	Não
ID_PARAMETRO	INTEGER	Sim	Sim	Identificador do parâmetro na Matriz Y.	Sim
NM_PARAMETRO	VARCHAR(45)	Não	Sim	Nome identificador do parâmetro na Matriz Y	Não

Nome do Índice	Tipo de Índice	Coluna(s)
Chave Primária	Chave Primária	ID_MODELO ID_PARAMETRO
CAD_PARAMETRO_MODELO_FKIndex1	Index	ID_MODELO

Fonte: Autoria Própria (2019)