

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

STEVEN KARISTON LOUBACK DE CARVALHO

**SISTEMA DE SOFTWARE PARA ANÁLISE DE PARÂMETROS  
SANGUÍNEOS DE FORMA NÃO INVASIVA**

DISSERTAÇÃO DE MESTRADO

CAMPO MOURÃO

2019

STEVEN KARISTON LOUBACK DE CARVALHO

**SISTEMA DE SOFTWARE PARA ANÁLISE DE PARÂMETROS  
SANGUÍNEOS DE FORMA NÃO INVASIVA**

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

---

C331 Carvalho, Steven Kariston Louback de

Sistem de software para análise de parâmetros sanguíneos de forma não invasiva / Steven Kariston Louback de Carvalho. – Campo Mourão, 2019.  
70 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. Diabetes. 2. Espectroscopia de infravermelho. 3. Quimiometria. 4. 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**

### **SISTEMA DE SOFTWARE PARA ANÁLISE DE PARÂMETROS SANGUÍNEOS DE FORMA NÃO INVASIVA**

por

Steven Kariston Louback de Carvalho

Essa dissertação foi apresentada às dezenove 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

- O Termo de Aprovação assinado encontra-se na Coordenação do Curso -

## RESUMO

CARVALHO, Steven. **SISTEMA DE SOFTWARE PARA ANÁLISE DE PARÂMETROS SANGUÍNEOS DE FORMA NÃO INVASIVA**. 2019. 66. Dissertação (Mestrado em Inovações Tecnológicas) - Universidade Tecnológica Federal do Paraná. Campo Mourão, 2019.

Laboratórios de análises clínicas podem diagnosticar, rastrear e monitorar a diabetes mellitus, por meio de exames que usam métodos de coleta amostral desconfortáveis para o paciente por serem invasivos além de expor o analista e o meio ambiente a riscos de contaminação. Por isso, o desenvolvimento de formas de análises clínicas fundamentadas na espectroscopia para análises menos invasivas poderia diminuir a quantidade de exames sanguíneos convencionais e trazer vantagens como baixo custo, simplicidade operacional, mínimo de preparo das amostras (quando necessário), sem geração de resíduos, além de diminuir a exposição dos analistas ao material biológico. A fim de se promover este desenvolvimento, testes preliminares para análises sanguíneas por espectroscopia foram realizados diretamente na mucosa bucal de pacientes, sendo que os resultados obtidos foram muito encorajadores. No entanto, apesar da excelente expectativa, observou-se a dependência de intervenções humanas no processo que vai desde a aquisição dos espectros até que se forneçam as respostas de interesse, retardando sua aplicação. Considerando a possibilidade de se otimizar o processo, este trabalho propõe um sistema de software para integrar o processo desde a medida espectral até a resposta quantitativa sobre o teor glicêmico. O processo de análise proposto por meio do software inicia com a integração automatizada do equipamento responsável por obter as medidas espectrais na região do infravermelho próximo com um *smartphone* ou computador pessoal. Um módulo cliente deste software será responsável por transmitir a informação coletada pelo equipamento para uma nuvem computacional, na qual um serviço *Web* responsável executa a calibração multivariada (por Mínimos Quadrados Parciais). Este serviço *Web* possibilita a utilização e atualização de um modelo de calibração multivariada, responsável por estabelecer a correlação entre os espectros obtidos e as medidas convencionais, permitindo um monitoramento em tempo real de pacientes e, conseqüentemente, intervenções mais assertivas. Desta forma, o diagnóstico de diabetes mellitus pode ser realizado por uma metodologia menos desconfortável e extremamente ágil, tornando-se acessível a um número maior de pessoas e sendo ainda mais representativo por permitir medidas diárias. Ressalta-se ainda o aspecto sustentável da metodologia que, por não gerar resíduos biológicos e lixo hospitalares como agulhas e seringas, beneficia diretamente o meio ambiente, além da possibilidade de expansão para medida de outros parâmetros sanguíneos.

**Palavras-chave:** NIR. PLS. Calibração Multivariada. Software. Serviço *Web*.

## ABSTRACT

CARVALHO, Steven. **SOFTWARE SYSTEM FOR NON-INVASIVE BLOOD PARAMETERS ANALYSIS**. 2019. 66. Dissertation (Master degree on Technological Innovations) – Federal university of technology of Paraná. Campo Mourão, 2019.

Clinical analysis laboratories can diagnose, track, and monitor diabetes mellitus through tests that use patient collection methods that are uncomfortable for the patient because they are invasive and expose the analyst and the environment to contamination risks. Therefore, the development of spectroscopy-based forms of clinical analysis for less invasive analyzes could reduce the number of conventional blood tests and bring advantages such as low cost, operational simplicity, minimal sample preparation (when necessary), no waste generation, as well as reducing analyst exposure to biological material. To promote this development, preliminary tests for spectroscopic blood tests were performed directly on the oral mucosa of patients, and the results obtained were very encouraging. However, despite the excellent expectation, it was observed the dependence of human interventions in the process that goes from the acquisition of spectra until the answers of interest are provided, delaying their application. Considering the possibility of optimizing the process, this work proposes a software system to integrate the process from the spectral measurement to the quantitative response on the glycemic content. The proposed analysis process through the software begins with the automated integration of the equipment responsible for obtaining near-infrared spectral measurements with a smartphone or personal computer. A client module of this software will be responsible for transmitting the information collected by the equipment to a computational cloud, in which a responsible web service performs multivariate calibration (by Partial Least Squares). This web service enables the use and updating of a multivariate calibration model, responsible for establishing the correlation between the obtained spectra and conventional measurements, allowing real-time monitoring of patients and, consequently, more assertive interventions. Thus, the diagnosis of diabetes mellitus can be made by a less uncomfortable and extremely agile methodology, making it accessible to a larger number of people and being even more representative by allowing daily measurements. It is also emphasized the sustainable aspect of the methodology that, by not generating biological waste and hospital waste such as needles and syringes, directly benefits the environment, besides the possibility of expansion to measure other blood parameters.

**Keywords:** NIR. PLS. Multivariate Calibration. Software. Web Service.

## LISTA DE ILUSTRAÇÕES

<b>Figura 1</b> - Fluxograma com os diferentes métodos empregados na quimiometria ...	19
<b>Figura 2</b> - Montagem da Matriz de Espectros.....	21
<b>Figura 3</b> - Construção da Matriz <b>X</b> e vetor <b>y</b> .....	22
<b>Figura 4</b> - Esquema do Método PLS .....	24
<b>Figura 5</b> - Arquitetura de uma Relação no Banco de Dados Relacional.....	28
<b>Figura 6</b> - Arquitetura do ambiente de compilação e interpretação da linguagem Python .....	30
<b>Figura 7</b> - Arquitetura de um serviço <i>Web</i> .....	32
<b>Figura 8</b> - Fluxograma do Desenvolvimento do software .....	37
<b>Figura 9</b> - Processo de calibração feito manualmente.....	44
<b>Figura 10</b> - Processo de predição de amostra manual .....	45
<b>Figura 11</b> - Diagrama de Caso de Uso .....	46
<b>Figura 12</b> - Diagrama de Atividade do Processo de Predição da Amostra .....	47
<b>Figura 13</b> - Algoritmo <i>Kennard-Stone</i> .....	48
<b>Figura 14</b> - Algoritmo KNN.....	49
<b>Figura 15</b> - Arquitetura do Software.....	51
<b>Figura 16</b> - Modelo Relacional.....	53
<b>Figura 17</b> - Protótipo da Interface de Acesso ao Sistema .....	54
<b>Figura 18</b> - Protótipo da Interface Principal .....	55
<b>Figura 19</b> - Interface de Gerenciamento de Modelos .....	56
<b>Figura 20</b> - Interface de Inclusão de Amostras .....	56
<b>Figura 21</b> - Interface para a Predição .....	57
<b>Figura 22</b> - Gráfico de Predição.....	58
<b>Figura 23</b> - Interfaces de login e o menu principal do aplicativo GlicoLab .....	59
<b>Figura 24</b> - Interface de coleta e resultados da Análise no Aplicativo .....	60
<b>Figura 25</b> - Processo de construção do modelo de calibração automatizado .....	63

## LISTA DE TABELAS

<b>Quadro 1</b> - Métodos HTTP e Operações CRUD.....	33
<b>Quadro 2</b> - Figuras de mérito do modelo de calibração multivariada .....	61



## 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
IBM	International Business Machines;
IDE	Ambiente de Desenvolvimento Integrado
JSF	Java Server Faces
KNN	K Nearest Neighbor
MIR	Mid Infrared Spectroscopy
MVP	Minimum Viable Product
NIR	Near Infrared Spectroscopy
OASIS	Advancing Open Standards for the Information Society
PCA	Análise por Componentes Principais
PLS.	Regressão por Mínimos Quadrados Parciais
SGBD	Sistema Gerenciador de Bando de Dados
REST	Representational State Transfer
SOA	Arquitetura Orientada a Serviços
SOAP	Simple Object Access Protocol
UDDI	Universal Description, Discovery and Integration
XML	Extensible Markup Language
WSDL	Web Services Description Language
W3C	World Wide Web Consortium

## **LISTA DE SIGLAS**

ABIPTI	Associação Brasileira das Instituições de Pesquisa Tecnológica
ACID	Atomicidade, Consistência, Isolamento e Durabilidade
BSC	Balanced Scorecard
CH	Capital Humano
CRUD	Create, Read, Update, Delete

## **LISTA DE ACRÔNIMOS**

CWI	Instituto Nacional de Pesquisa para Matemática e Ciência da Computação da Holanda
PSF	Python Software Foundation
OHA	Open Handset Alliance

## SUMÁRIO

<b>1 INTRODUÇÃO</b>	<b>12</b>
<b>2 REFERÊNCIAL TEÓRICO</b>	<b>17</b>
2.1 ESPECTROSCOPIA DE INFRAVERMELHO PRÓXIMO	17
2.1.1 História e Aplicação	17
2.1.2 Princípios Teóricos do NIR	18
2.2 QUIMIOMETRIA	18
2.2.1 Calibração Multivariada	20
2.2.2 Validação de Modelos de Calibração Multivariada	22
2.2.3 Mínimos Quadrados Parciais (PLS)	23
2.2.4 Algoritmo de Kennard-Stone	24
2.3 DESENVOLVIMENTO DE SOFTWARE	25
2.3.1 Banco de Dados Relacional	26
2.3.2 Banco de Dados PostgreSQL	28
2.3.3 Linguagem de Programação Python	29
2.3.4 Linguagem de Programação Java	30
2.3.5 Serviços <i>Web</i>	31
2.3.6 UML (Unified Modeling Language)	34
2.3.7 Desenvolvimento ágil de software	34
2.3.8 Metodologia de Desenvolvimento ágil SCRUM	35
<b>3 PROCEDIMENTOS</b>	<b>37</b>
3.1 LEVANTAMENTO DOS REQUISITOS	37
3.2 PROJETO DO SISTEMA	38
3.3 IMPLEMENTAÇÃO	40
<b>4 RESULTADOS E DISCUSSÕES</b>	<b>41</b>
4.1 REQUISITOS FUNCIONAIS	41
4.1.1 Processo de criação e calibração do modelo multivariado	43
4.1.2 Processo de Predição de Amostra	44
4.1.3 Diagrama de Caso de Uso	45
4.1.4 Diagrama de Atividades	46
4.2 ALGORITMOS E MÉTODOS	48
4.2.1 Algoritmo <i>Kennard-Stone</i> (Kennard & Stone, 1969)	48
4.2.2 Algoritmo KNN ( <i>K Nearest Neighbor</i> )	49
4.3 ARQUITETURA DO SOFTWARE	50
4.3.1 Projeto do Banco de Dados	52
4.3.2 Interfaces do software	54
4.4 VALIDAÇÃO DO MODELO DE CALIBRAÇÃO MULTIVARIADA	61
4.5 PROCESSOS MELHORADOS	61
4.5.1 Processo de criação e calibração do modelo	62

4.5.2 Processo de Predição de Amostra.....	64
<b>5 CONCLUSÃO.....</b>	<b>65</b>
<b>6 REFERÊNCIAS .....</b>	<b>67</b>

## 1 INTRODUÇÃO

Laboratórios de análises clínicas são utilizados para diagnosticar, rastrear e monitorar a diabetes mellitus, por meio de exames convencionais. A qualidade dos resultados dos exames laboratoriais está intimamente relacionada à fase pré-analítica e, principalmente, às condições de coleta de sangue venoso (BRASIL, 2005). Esta etapa é responsável por cerca de 70% do total de erros ocorridos nos laboratórios clínicos que possuem um sistema de controle da qualidade bem estabelecido (BRASIL, 2014). São consideradas como condições pré-analíticas a variação cronobiológica, gênero, idade, posição, atividade física, jejum, dieta, uso de drogas para fins terapêuticos ou não, e até mesmo a aplicação de torniquete. Outros pontos que também podem causar variação dos resultados são os aspectos do tubo de coleta, como o uso de gel separador, anticoagulantes e conservantes e características da amostra, como hemólise e lipemia (BRASIL, 2005).

Os métodos tradicionais de análises sanguíneas demandam um tempo significativo para revelação dos resultados, devido ao processo ser feito através de reagentes, análise quantitativas e visuais por meio de um microscópio, além de exporem os analistas a riscos de contaminação durante a manipulação do material biológico e gerarem resíduos críticos de manipulação. Para minimizar os contratempos apresentados pelas análises convencionais, uma alternativa viável seria a aplicação de métodos espectroscópicos, também chamados de métodos ópticos. A aplicação de métodos ópticos apresenta vantagens em relação aos métodos tradicionais tais como baixo custo a médio e longo prazo, simplicidade operacional, não destruição da amostra analisada, além de oferecer análises rápidas, com um mínimo de preparo das amostras (quando necessário) e não gerarem resíduos. Dentre os métodos ópticos mais adequados para este fim, pode-se citar a espectroscopia na região do infravermelho próximo, que se baseia na absorção de energia radiante por moléculas que transformam a energia absorvida em vibração molecular. A frequência de uma vibração depende das massas relativas dos átomos ligados, das constantes de força das ligações e da geometria dos ligantes. Desta forma, cada frequência de vibração pode ser associada a um tipo específico de ligação química, permitindo diferenciar substâncias, desde que a

concentração esteja, no mínimo, como 0,1% da constituição da amostra (SILVERSTEIN et al., 2010).

Em uma proposta de análise realizada durante o mês de abril de 2016, executada por Larissa Rocha dos Santos e Luiza Mariano Leme, na época discentes do Programa de Pós Graduação em Tecnologia de Alimentos da Universidade Tecnológica Federal do Paraná do Campus Campo Mourão, foram coletados espectros da mucosa bucal de 165 pessoas que foram até o “Laboratório de Análises Clínicas Santa Cecília”, da cidade de Campo Mourão. Os pacientes apresentavam idade variando de 18 até 72 anos e estavam na clínica para realização de análises sanguíneas convencionais, via punção venosa. Todos os avaliados assinaram termo de consentimento concordando com a liberação dos dados para fins de cálculos desde que não houvesse exposição nominal. Na ocasião, os resultados obtidos a partir do método convencional foram correlacionados com as análises espectrais, mostrando uma correlação (R) de 0,8866. No entanto, a dificuldade de sistematização das etapas inviabilizou as análises por exigirem uma quantidade de trabalho manual prolongada e morosa no que diz respeito a organização dos dados. Por outro lado, o trabalho inicial mostrou que, caso um sistema computacional automatizado permitisse a integração entre os espectros produzidos, o modelo de calibração e os interessados (paciente e ou médico), a metodologia proposta poderia trazer benefícios significativos, tais com diminuição das análises sanguíneas convencionais.

Na análise baseada em espectroscopia, a coleta da amostra não é realizada via punção venosa e sim por meio de radiação eletromagnética de baixa energia (menos energética que a radiação visível) focalizando-se um espectrômetro portátil e forma direta na mucosa bucal de pacientes. O equipamento utilizado para as medidas de infravermelho são produzidas por um sistema que possui interfaces de integração que permitem a exportação dos resultados para planilhas eletrônicas ou programas estatísticos para futura manipulação. Desta forma, pode-se utilizar os resultados obtidos por este tipo de sistema para diversos fins, tais como a construção de modelos de calibração a partir de métodos multivariados para análises, mais conhecidos como métodos quimiométricos para análises quantitativas (BARROS NETO et al., 2006; MARÇO, 2009; VALDERRAMA, 2009).

Considerando-se a medida de parâmetros sanguíneos por espectroscopia por meio de calibração multivariada, em que são correlacionadas as medidas espectrais com o método de referência utilizado convencionalmente, é possível construir modelos de predição dos resultados de diabetes mellitus de uma determinada amostra, evitando o desconforto da punção venosa no paciente, diminuindo os riscos produzidos pelos materiais biológicos expostos aos profissionais envolvidos no processo e ao meio ambiente.

Atualmente, todas as etapas desta análise sanguínea baseada em espectroscopia são realizadas em subetapas que requerem intervenções humanas durante o processo, criando uma dependência direta da capacidade e disponibilidade dos profissionais envolvidos no processo. Para transferir as informações coletadas a partir do equipamento de espectroscopia para um dispositivo eletrônico, é necessário a utilização de softwares *proprietários* (softwares licenciados com os direitos exclusivos do desenvolvedor) e um operador especializado. A confecção e a validação do modelo de calibração e a predição de uma determinada amostra, atualmente, são tarefas realizadas manualmente com o auxílio de softwares estatísticos, tais como o MatLab<sup>1</sup> e R<sup>2</sup>, responsáveis pela execução dos métodos quimiométricos necessários para o tratamento dos dados, tais como a Análise de Componentes Principais (PCA, do inglês, *Principal Component Analysis*), métodos para identificar amostras anômalas (outliers), métodos de regressão, tais como o mínimos quadrados parciais (PLS, do inglês, *Partial Least Squares*).

Mediante ao fato de que já existem as bases de comparações e os modelos matemáticos conhecidos que produzem resultados confiáveis frente aos exames convencionais, neste trabalho propõe-se um sistema de software capaz de unificar o processo de análise de diabetes mellitus por meio da espectroscopia no

---

<sup>1</sup> MATLAB (MATrix LABoratory) trata-se de um software interativo de alta performance voltado para o cálculo numérico. O MATLAB integra análise numérica, cálculo com matrizes, processamento de sinais e construção de gráficos em ambiente fácil de usar onde problemas e soluções são expressas somente escritas matematicamente, ao contrário da programação tradicional.

<sup>2</sup> R é uma linguagem e também um ambiente de desenvolvimento integrado para cálculos estatísticos e gráficos. Foi criada originalmente por Ross Ihaka e por Robert Gentleman no departamento de Estatística da universidade de Auckland, Nova Zelândia, e foi desenvolvido em um esforço colaborativo de pessoas em vários locais do mundo.

infravermelho próximo, com o objetivo de viabilizar o uso deste método de análise para fins práticos. O processo de análise proposto por meio do software, que deverá ser o produto final deste trabalho, se inicia com a integração automatizada do equipamento responsável por obter as medidas espectrais com um smartphone ou computador pessoal. O módulo cliente deste software será responsável por transmitir os dados produzidos por espectroscopia para uma nuvem computacional (serviço *Web* responsável por receber os dados espectrais e executar os métodos matemáticos, de acordo com a parametrização do padrão estabelecido pelo especialista).

Os espectros transferidos ficarão armazenados em um banco de dados para que sejam utilizados em uma análise preditiva ou incluso no modelo de calibração multivariada. Estas integrações e automatizações serão extremamente úteis para reduzir a carga de trabalho do analista, tarefas que hoje são realizadas manualmente. O sistema proposto neste trabalho deve possibilitar a construção, validação e manutenção de um modelo de calibração multivariada, em que o usuário responsável pela parametrização tenha a possibilidade de definir previamente os parâmetros para o tipo de análise que deseja executar. Ainda, por meio do sistema proposto, será possível correlacionar os resultados das análises espectroscópicas com os resultados obtidos nos métodos de referência (exames laboratoriais padrão).

O objetivo principal deste trabalho é propor um MVP (*Minimum Viable Product*), fruto da interação de conhecimentos das áreas de química e tecnologia da informação, capaz de trazer benefícios para pessoas envolvidas em um exame laboratorial, para o meio ambiente e para a sociedade. Para os **pacientes** que necessitam fazer exames sanguíneos, além de ser uma técnica não invasiva e menos desconfortável, será menos custosa financeiramente além de ser extremamente ágil, tornado este exame de diabetes mellitus acessível para um maior número de pessoas e permitindo um monitoramento mais representativo.

Considerando que os **profissionais responsáveis** por realizar as análises sanguíneas estão constantemente sujeitos a riscos, a utilização deste software possibilitará minimizar significativamente a exposição a materiais biológicos coletados dos pacientes e objetos utilizados. O **aspecto sustentável** deste projeto será a viabilização técnica e operacional de uma metodologia de análise sanguínea que não gera resíduos biológicos e lixo hospitalares como agulhas e seringas,



possibilitando o aumento do monitoramento de pessoas sem prejuízo ao meio ambiente.

Esta dissertação está dividida na forma de capítulos, sendo que o Capítulo 2 traz uma revisão bibliográfica sobre os principais temas deste trabalho, destacando-se a calibração multivariada e métodos quimiométricos. Também são apresentadas as metodologias e tecnologias utilizadas para o desenvolvimento do sistema que compõe este trabalho. No Capítulo 3 encontram-se descritos os procedimentos que são as etapas, materiais, bibliotecas e equipamentos utilizados na execução do trabalho. No Capítulo 4 apresentam-se discutidos os resultados, tendo em vista o cumprimento dos objetivos propostos. No Capítulo 5 são apresentadas as conclusões deste trabalho.

## 2 REFERÊNCIAL TEÓRICO

Neste capítulo são apresentados os conceitos principais deste trabalho, construídos por autores reconhecidos nas respectivas áreas do conhecimento que atuam, sendo estes extremamente úteis para o embasamento deste trabalho. Os conceitos usados nesta dissertação envolvem: quimiometria, métodos quimiométricos, arquiteturas de softwares, linguagens de programação e banco de dados.

### 2.1 Espectroscopia de Infravermelho Próximo

#### 2.1.1 História e Aplicação

A região do infravermelho próximo foi descoberta no ano de 1800, pelo astrônomo Sir William Herschel, quando estudava a contribuição de cada uma das cores da luz solar no aumento da temperatura das substâncias expostas a radiação. Por se encontrar a um comprimento de onda próximo do espectro visível, foi denominada como espectroscopia de infravermelho próximo (NIR, do inglês *Near Infrared Spectroscopy*) (Pasquini 2003). No entanto, a espectroscopia de infravermelho teve um período de latência, entre os anos de 1800 e 1950, devido à existência de outros métodos analíticos que mostravam capacidade de fornecer resultados menos contestáveis. Assim, o NIR teve o seu reconhecimento no setor agroalimentar, apenas na década de 1950, especialmente na medição direta de amostras sólidas (Workman e Weyer 2012).

O sistema de análises por infravermelho próximo pode vir a aprimorar grande parte das metodologias convencionais de análises em laboratório, com qualidade e as especificidades necessárias para cada análise. Dentre as vantagens em relação aos métodos tradicionais, destacam-se a análise múltipla dos constituintes, o curto período de tempo necessário para se medir cada amostra, a menor necessidade de mão-de-obra aplicada (na maioria dos casos não há necessidade de preparo de amostra), o menor custo a médio prazo além do fato de não ser poluente por não utilizar produtos químicos ou reagentes (Amorim, 1996).

### 2.1.2 Princípios Teóricos do NIR

A espectroscopia de infravermelho pode ser dividida em três regiões distintas, sendo que o infravermelho próximo (NIR) se encontra compreendido entre os 780-2500 nm, infravermelho médio (MIR, do inglês *Mid Infrared Spectroscopy*), compreendido entre 2500-40000 nm e o infravermelho distante (FIR, do inglês *Far Infrared Spectroscopy*), compreendido entre os 40000-100000 nm (Workman 2006; Sun 2008). O espectro de um determinado material obtido com radiação infravermelha é o resultado da absorção de energia, na forma de luz, por moléculas orgânicas, particularmente aquelas que possuem ligações entre átomos de carbono e entre carbono e hidrogênio, oxigênio, nitrogênio e até enxofre, dependendo da geometria e estrutura das moléculas produzidas (Amorim, 1996).

Quando a radiação de infravermelho é incidida na amostra, será absorvida a frequência da radiação que corresponder à frequência de vibração da ligação correspondente àquela determinada frequência. A frequência é dependente da massa entre os ligantes e da energia da ligação, fazendo com que as vibrações sejam características para cada tipo de molécula, permitindo assim diferenciações e até mesmo quantificações. No entanto, para que ocorra vibração molecular, é necessário que a radiação infravermelha consiga interagir com a molécula de modo a produzir uma alteração em seu momento dipolar (Pasquini 2003).

## 2.2 Quimiometria

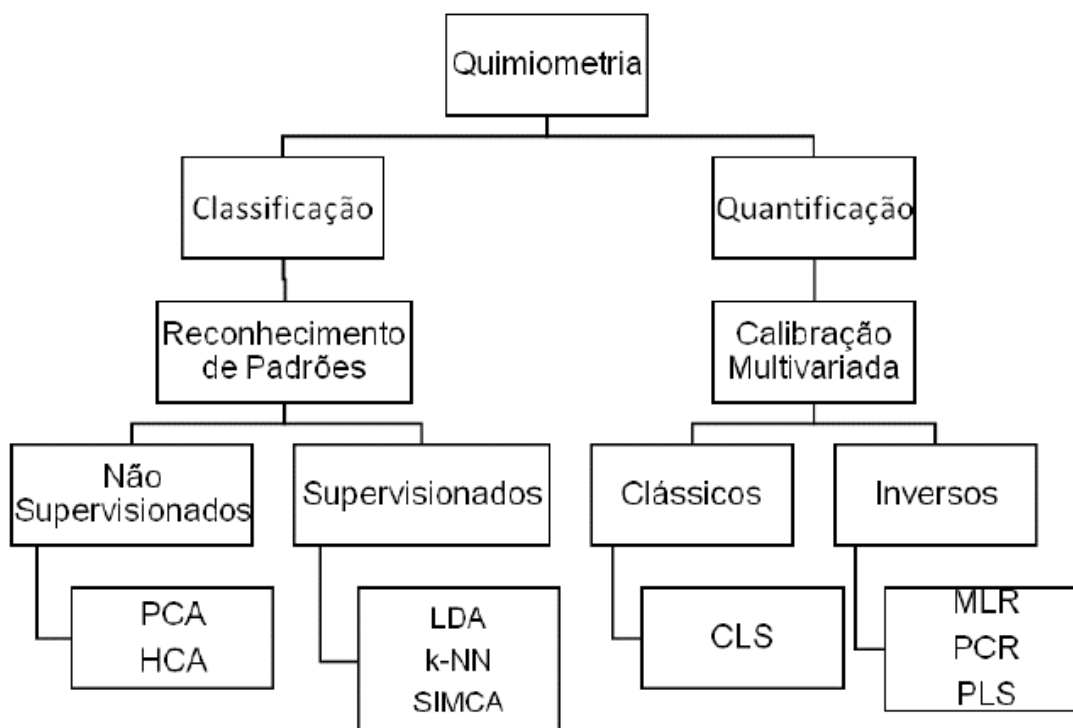
A quimiometria surgiu devido à necessidade de novos métodos estatísticos e matemáticos para extrair o máximo de informação química de dados cada vez mais complexos e com um número crescente de variáveis. Barros Neto et al. (2006) ressaltam que os trabalhos em quimiometria no Brasil podem ser agrupados em três áreas: planejamento e otimização de experimentos, reconhecimento de padrões (métodos de análise exploratória e classificação) e calibração multivariada.

Segundo Ferreira Neto (2012), a quimiometria trabalha com análise multivariada, por meio de métodos que são escolhidos de acordo com os objetivos

da pesquisa, pois sabe-se que a análise multivariada é uma análise exploratória de dados, prestando-se a gerar hipóteses, e não tecer confirmações a respeito dos mesmos, o que seria uma técnica confirmatória como àquelas dos testes de hipótese, embora em algumas situações possa ser utilizada para confirmação dos eventos. Portanto, pode-se dizer que a quimiometria surge face à necessidade de extração de informações de dados multivariados utilizando-se métodos matemáticos e estatísticos de modo a processar os dados das medições efetuadas (Roussel et al., 2014).

Na década de 1970, pesquisadores que utilizavam métodos estatísticos para cálculos necessários em suas pesquisas. Porém, somente após a evolução dos sistemas computacionais a quimiometria tornou-se acessível aos pesquisadores de diversas áreas, dentre as quais destaca-se a química analítica (Neto et al., 2006). A Figura 1 representa distribuição de técnicas qualitativas e quantitativas utilizada na quimiometria (Ferreira, 2015).

**Figura 1** - Fluxograma com os diferentes métodos empregados na quimiometria



Fonte: Valderrama (2005)

Para Valderrama (2005), a principal linha de pesquisa da quimiometria aplicada à química analítica tem sido a construção de modelos de regressão a partir de dados de primeira ordem, ou seja, dados que podem ser representados por meio de um vetor para cada amostra, sendo a construção desses modelos denominada de calibração multivariada.

### 2.2.1 Calibração Multivariada

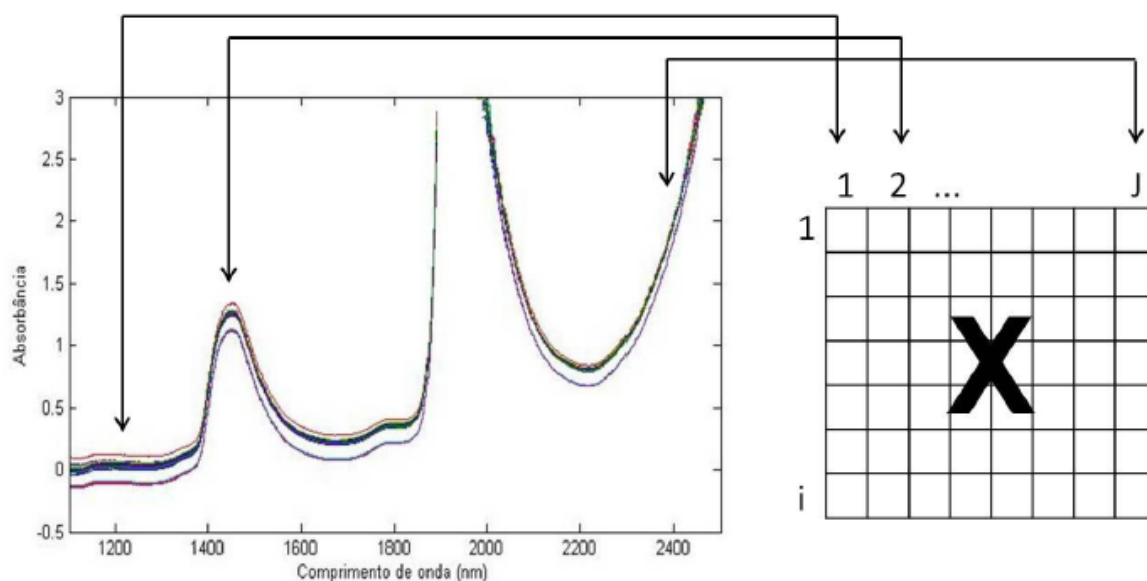
Um dos grandes desafios da química analítica é a estimativa do número e a concentração das espécies em misturas por meio de espectros. Uma série de técnicas estatísticas tem sido utilizada para desenvolver metodologias multivariadas e para extrair informação dos espectros com o objetivo de identificar as espécies presentes e fazer determinações quantitativas. A aplicabilidade de cada metodologia depende do conjunto de dados (informação experimental) submetidos à análise. Um requisito importante para aplicação dessas metodologias é que os espectros das misturas devem ser uma combinação linear dos sinais das espécies puras, ponderados por suas concentrações (Scarminio, 1998). O uso dos métodos de análise estatística tanto na química como em outras ciências experimentais está intimamente ligado ao desenvolvimento dos computadores e de equipamentos capazes de realizar diversas medidas em uma amostra simultânea ou sequencialmente, permitindo assim a aquisição e o tratamento multivariado destes dados (Martens, 1994).

A calibração multivariada é uma parte importante da quimiometria, que inclui métodos de análise estatística para construção de modelos matemáticos para a quantificação de parâmetros de uma amostra (Coscione, 2001). Neste tipo de calibração, se estabelece uma relação entre dois blocos de dados de informação química disponível: o bloco das medidas instrumentais e o bloco da propriedade calibrada (Valderrama, 2005). No entanto, a aquisição de um grande volume de dados, mesmo que os experimentos tenham sido planejados de alguma forma, não significa necessariamente que estes contenham a informação necessária para descrever a(s) propriedade(s) de interesse nas amostras, nem que o modelo construído descreva este sistema de forma otimizada. O desenvolvimento de

modelos com calibração multivariada consiste de duas etapas: o desenvolvimento (ou calibração) e a validação (previsão) (Ferreira, 1999).

Na etapa de calibração, a partir de um padrão representativo para o conjunto de amostras a serem analisadas, são criadas as relações entre sinais espectrais das amostras (Matriz  $\mathbf{X}$ ) e concentrações dos analitos (vetor  $y$ ). Os dados utilizados nesta etapa constituem o conjunto de treinamento (Fortes, 2006). A propriedade de interesse nesses padrões deve ser quantificada por meio de um método de referência consolidado, com precisão e exatidão conhecidas. Antes de iniciar os cálculos, esses devem ser agrupados em matrizes (Coscione, 2001). A matriz das variáveis independentes  $\mathbf{X}$  ( $m$ -linhas,  $n$ -colunas) é composta por sinais analíticos em que as linhas representam as  $m$ -amostras e as colunas representam as  $n$ -variáveis (Fortes, 2006), como mostra o esquema apresentado na Figura 2.

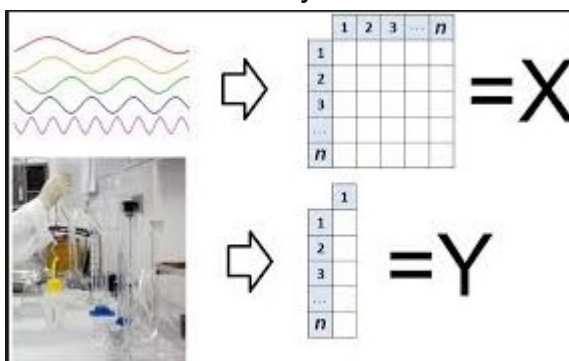
**Figura 2** - Montagem da Matriz de Espectros



**Fonte:** Valderrama (2015)

Assim como o primeiro agrupamento de dados representado pela matriz de espectros, temos um outro agrupamento de dados, constituídos pelas variáveis dependentes que compõem o vetor  $y$  ( $m$ -linhas, 1-coluna), onde a coluna ( $p$ ) correspondem às concentrações dos analitos e cada linha ( $m$ ) se refere a amostra em questão. Este agrupamento pode ser observado na Figura 3.

**Figura 3** – Construção da Matriz  $X$  e vetor  $y$



**Fonte:** Autoria Própria (2019).

A matriz de dados originais  $X$  e o vetor de respostas proveniente da metodologia padrão  $y$  a ser calibrada são decompostas em vetores de escores e pesos e uma matriz de resíduos de dados não modelados (ruídos estatísticos). O produto destes vetores origina os denominados auto-vetores, componentes principais ou ainda, variáveis latentes (dependendo da metodologia), que são capazes de representar as amostras em poucas dimensões e descrever a direção de máxima variância correlacionada com a propriedade de interesse. Os escores são as coordenadas das amostras no novo sistema de eixos. Seu gráfico permite a identificação destas, bem como a análise de semelhanças, agrupamentos e *outliers*. Analogamente, os pesos se referem às variáveis presentes nos conjuntos de dados, e permitem a identificação das variáveis importantes. Eles contêm ainda informações sobre a relevância de cada variável original na formação dos novos eixos (Fortes, 2006).

### 2.2.2 Validação de Modelos de Calibração Multivariada

Após a construção do modelo de calibração multivariada é necessário testar sua capacidade de predição de amostras desconhecidas para garantir a consistência dos resultados, ou seja, amostras que não estavam presentes no conjunto de calibração. O resultado emitido pelo modelo de calibração deve ser confrontado com o resultado de um método referência já consolidado (Broad et al., 2006; Roberts e Cozzolino 2016). Existem dois tipos de validação: validação independente, também conhecida como validação externa e a validação cruzada. A

validação externa requer um conjunto de validação separado do conjunto de calibração, que deve ser representativo em relação aos parâmetros em estudo de modo a fornecer estimativas relevantes e confiáveis da capacidade de previsão do modelo de calibração. Já a validação cruzada procura validar o modelo de calibração sem usar dados externos. Para que se efetue uma calibração cruzada, são removidas amostras ou um grupo de amostras do conjunto de calibração de forma sucessiva na tentativa da previsão da amostra removida, sendo que após cada remoção é feita uma nova calibração, e assim sucessivamente até todas as amostras terem sido utilizadas para a validação (Sun 2008).

### 2.2.3 Mínimos Quadrados Parciais (PLS)

O método dos Mínimos Quadrados Parciais, ou simplesmente PLS (do inglês *Partial least Squares*) é o mais amplamente utilizado para a construção de modelos de regressão em dados que apresentam comportamento linear, ou seja, a variação de uma propriedade medida provoca alteração proporcional de mesma magnitude de forma direta ou inversa no sinal da amostra. O principal objetivo é aproximar o espaço das medidas originais por meio de um espaço vetorial reduzido, com alguma proposta de restrição, para que a matriz de dados seja decomposta direcionando a solução para uma propriedade de interesse qualquer.

As informações referentes a propriedade de interesse que se deseja estudar são inseridas no cálculo das variáveis latentes (VL). Para esse modelo de regressão cada VL está relacionada à matriz de dados por meio de outra matriz, em que estão inseridas as propriedades de interesse. O esquema desse modelo pode ser observado na Figura 4 (Ferreira, 2015).



**Figura 4** – Esquema do Método PLS

$$\begin{array}{c}
 \begin{array}{ccc}
 & p & \\
 & \boxed{\text{X}} & \\
 n & & 
 \end{array}
 =
 \begin{array}{ccc}
 & h & \\
 & \boxed{\text{T}} & \\
 n & & 
 \end{array}
 \begin{array}{ccc}
 & p & \\
 h & \boxed{\text{P}} & \\
 & & 
 \end{array}
 +
 \begin{array}{ccc}
 & p & \\
 & \boxed{\text{E}} & \\
 n & & 
 \end{array}
 \\
 \\
 \begin{array}{ccc}
 & m & \\
 & \boxed{\text{C}} & \\
 n & & 
 \end{array}
 =
 \begin{array}{ccc}
 & h & \\
 & \boxed{\text{U}} & \\
 n & & 
 \end{array}
 \begin{array}{ccc}
 & m & \\
 h & \boxed{\text{Q}} & \\
 & & 
 \end{array}
 +
 \begin{array}{ccc}
 & m & \\
 & \boxed{\text{F}} & \\
 n & & 
 \end{array}
 \end{array}$$

**Fonte:** Ferreira (2015)

Em que: **X** é a matriz de dados; **C** é a matriz de concentração; **E** = erro; **Q** = vetor análogo ao vetor *loadings* PCA; **F** = função erro; **P** = *loadings*; **U** = *scores*

O PLS é semelhante ao PCA, exceto ao fato que a decomposição executada por ele é feita simultaneamente em duas matrizes (X e Y) e gera duas novas (T e U) com a maior covariância possível (Fachin, 2005). As matrizes X e Y são decompostas em *scores* e *loadings* simultaneamente, enquanto cada componente principal sofre uma pequena modificação (rotação) para buscar a máxima covariância entre X e Y. Assim, os componentes principais, que são obrigatoriamente ortogonais, no método PLS recebem a terminologia de Variáveis Latentes por perderem sua ortogonalidade (Valderrama et al., 2014). Ainda, de acordo com Valderrama (2009), o PLS estende o conceito do modelo inverso (propriedade como função da resposta instrumental) trocando as variáveis originais por um subconjunto truncado das variáveis latentes dos dados originais.

#### 2.2.4 Algoritmo de Kennard-Stone

No software proposto neste trabalho, a separação dos conjuntos de amostras entre calibração e validação será realizada por meio do algoritmo de Kennard-Stone,

que normalmente é aplicado para fazer a seleção das amostras pertinentes ao conjunto de calibração.

No algoritmo de Kennard-Stone, a primeira amostra selecionada é a que apresenta a maior distância em relação à média das amostras. A segunda amostra a ser selecionada será a que apresentar maior distância em relação à primeira amostra selecionada. A próxima amostra a ser selecionada apresentará maior distância em relação à última amostra selecionada, e assim sucessivamente até atingir o número de amostras desejadas (Kennard & Stone, 1969).

### **2.3 Desenvolvimento de Software**

O desenvolvimento de software é o processo de conversão de uma especificação de um sistema em executável. Isso significa que todo projeto de software é a descrição da estrutura de software a ser implementada, dos dados que são partes do sistema, das interfaces entre os componentes do sistema e, às vezes, dos algoritmos usados (Sommerville, 2008). Um software é uma ferramenta como qualquer outra, em que a sua qualidade é medida por meio do “número” de atividades que o usuário utilizador considera bem-sucedidas graças a sua utilização. O caso de sistemas de software de análise/simulação não é diferente.

Dentre as metodologias de desenvolvimento de software existentes atualmente, desenvolvimento ágil de software é uma maneira menos complexa, mais eficiente e focada em resultados e na colaboração entre a equipe de projeto e os demais interessados (Sabbagh, 2014).

No contexto de desenvolvimento de um software, a escolha de uma arquitetura de banco de dados a ser utilizada depende primordialmente da sua aplicação. O objetivo principal de um sistema de banco de dados é prover um ambiente que seja adequado e eficiente para uso na recuperação e armazenamento de informações.

### 2.3.1 Banco de Dados Relacional

Um banco de dados é uma coleção de informações relacionadas. Tais informações são dados ou fatos que podem ser gravados e que possuem um significado implícito. Por exemplo, considere nomes, números telefônicos e endereços de pessoas que você conhece. Esses dados podem ter sido escritos em uma agenda de telefones ou armazenados em um computador, por meio de programas como o Microsoft Access<sup>3</sup> ou Microsoft Excel<sup>4</sup>. Essas informações são uma coleção de dados com um significado implícito, conseqüentemente, um banco de dados (Elmasri e Navathe, 2011). Ainda, segundo Elmasri e Navathe (2011), os bancos de dados relacionais foram originalmente projetados para separar o armazenamento físico dos dados da sua representação conceitual, provendo de uma fundamentação matemática para os bancos de dados. O modelo relacional é matematicamente conciso, completo, anti-redundante e consistente internamente. Ainda, de acordo com Neves (2002), o banco de dados relacional é amplamente utilizado comercialmente para resolver diferentes problemas cotidianos. O modelo relacional surgiu devido às necessidades de aumentar a independência de dados nos sistemas gerenciadores de banco de dados; prover um conjunto de funções apoiadas em álgebra relacional<sup>5</sup> para armazenamento e recuperação de dados e permitir processamento *ad hoc*<sup>6</sup>. Este modelo foi resultado de um estudo teórico realizado por Codd (1970), tendo por base a teoria dos conjuntos e álgebra relacional. O modelo foi apresentado num artigo publicado em 1970, mas que só nos anos 80, foi implementado.

---

<sup>3</sup> Microsoft Access, também conhecido por MSAccess, é um sistema de gerenciamento de banco de dados da Microsoft, incluído no pacote do Microsoft Office Professional.

<sup>4</sup> O Microsoft Office Excel é um editor de planilhas produzido pela Microsoft para computadores que utilizam o sistema operacional Microsoft Windows, além de computadores Macintosh da Apple Inc. e dispositivos móveis como o Windows Phone, Android ou o iOS.

<sup>5</sup> Álgebra relacional é uma derivação descendente da lógica de primeira ordem e da álgebra de conjuntos em relação das operações sobre a relação finítimo, que auxilia o trabalho ao identificar os componentes de uma tupla por nome (chamado o atributo) ao invés de uma coluna de chaves numéricas, o qual é chamada a relação na terminologia de banco de dados.

<sup>6</sup> *ad hoc* é uma ligação temporária entre vários computadores e dispositivos utilizada para uma finalidade específica, por exemplo: jogos em rede, compartilhamento de documentos, compartilhamento de impressora e de internet com os utilizadores da rede etc.

O modelo relacional revelou-se ser flexível e adequado ao solucionar os vários problemas que se colocam ao nível da concepção e implementação da base de dados. A estrutura fundamental do modelo relacional é a relação. Uma relação é constituída por um ou mais atributos (campos), que traduzem o tipo de dados a armazenar. Cada instância da relação (linha) designa-se por uma tupla (registro). O modelo relacional implementa estruturas de dados organizadas em relações (tabelas). O modelo relacional impõe algumas restrições para evitar aspectos indesejáveis, tais como, repetição de informação, incapacidade de representar parte da informação e perda de informação. Essas restrições são: restrições de integridade referencial e restrições de integridade de entidade.

Segundo Date (2000), termo integridade refere-se à precisão, ou seja, consistência dos dados no banco de dados. Nesse contexto integridade significa semântica e são as restrições de integridade que representam o significado dos dados.

A restrição de integridade de entidade possui um importante artefato para garantir sua existência, as chaves primárias, que são sob o ponto de vista de um banco de dados relacional, o conjunto de um ou mais campos, cujos valores, considerando a combinação de valores em caso de mais de um campo compondo a chave primária, que nunca se repetem na mesma tabela e, desta forma, podem ser usados como um índice de referência para criar relacionamentos com as demais tabelas do banco de dados. Portanto, uma chave primária nunca pode ter valor nulo, nem valores repetidos.

A restrição de integridade de entidade é especificada em relações individuais e declara que nenhum valor da chave primária pode ser nulo. Isso se justifica porque o valor da chave primária sendo nula implica que não é possível identificar algumas tuplas (Elmasri e Navathe 2005).

A restrição de integridade referencial é especificada entre duas relações e é utilizada para manter a consistência entre tuplas destas relações. Informalmente, a restrição de integridade referencial declara que uma tupla em uma relação que se refere a uma outra relação deve se referir a uma tupla existente naquela relação (Elmasri e Navathe, 2011).

A Figura 5 apresenta a estrutura de uma relação em um banco de dados relacional.

**Figura 5** – Arquitetura de uma Relação no Banco de Dados Relacional



**Fonte:** Autoria Própria (2019).

### 2.3.2 Banco de Dados PostgreSQL

Para desenvolver este produto proposto, procurou-se, entre os Sistemas Gerenciadores de Bancos de Dados (SGBDs) disponíveis no mercado, aqueles que possuem uma arquitetura robusta e principalmente têm seu código fonte disponível e aberto.

Segundo Lane (2003), PostgreSQL é um sistema gerenciador de banco de dados objeto-relacional, desenvolvido no departamento de ciência da computação Berkeley, na Universidade da Califórnia, estável desde 1996. Tem mais de vinte anos de desenvolvimento ativo e uma arquitetura com forte reputação devido a sua confiabilidade, integridade de dados e exatidão. É bastante portátil podendo ser executado nos principais sistemas operacionais, incluindo Linux, UNIX e Windows. Mantém as propriedades ACID (Atomicidade, Consistência, Isolamento e Durabilidade) e vários recursos comuns aos tradicionais SGBDs comerciais, como suporte a consultas complexas, gatilhos (*triggers*), chaves estrangeiras (*foreign keys*), visões (*views*), controle de concorrência e integridade relacional. É atualmente o banco de dados de código aberto disponível no mercado que contém o conjunto mais completo de recursos.

O banco de dados utilizado para armazenar as informações do serviço *web* que será construído foi o PostgreSQL na versão 9.6. Foi escolhido por ser um Sistema Gerenciador de Banco de Dados Objeto-Relacional (SGBDOR) livre, ou seja, dá suporte aos modelos de banco de dados relacional e objeto-relacional, sem custo para sua utilização.

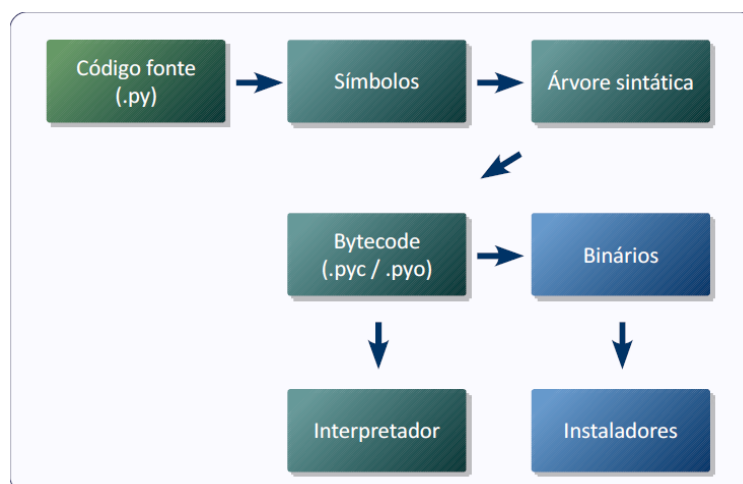
### 2.3.3 Linguagem de Programação Python

Segundo Borges (2010), Python é uma linguagem de altíssimo nível de abstração (em inglês, *Very High Level Language*) orientada a objeto, de tipagem dinâmica e forte, interpretada e interativa. A linguagem foi criada em 1990 por Guido van Rossum, no Instituto Nacional de Pesquisa para Matemática e Ciência da Computação da Holanda (CWI) e tinha originalmente foco em usuários como físicos e engenheiros. O Python foi concebido a partir de outra linguagem existente na época, chamada ABC (Borges, 2010). A linguagem inclui diversas estruturas de dados de alto nível de abstração e uma vasta coleção de bibliotecas, além de *frameworks* de terceiros que podem ser adicionados. Também possui recursos encontrados em outras linguagens modernas, tais como: geradores, introspecção, persistência, metaclasses e unidades de teste. Ela suporta programação modular e funcional, além da orientação a objetos. Mesmo os tipos básicos no Python são objetos. A linguagem é interpretada por meio de *bytecode* pela máquina virtual Python, tornando o código portátil, na Figura 6 é apresentada uma visão geral sobre a arquitetura do Python. Com isso, é possível compilar aplicações em uma plataforma e executar em outros sistemas ou executar direto do código fonte. Python é um software de código aberto com licença compatível com a *General Public License* (GPL)<sup>7</sup>, porém menos restritiva, permitindo inclusive a incorporação em produtos proprietários. A especificação da linguagem é mantida pela *Python Software Foundation*<sup>2</sup> (PSF).

---

<sup>7</sup> GNU General Public License (Licença Pública Geral GNU), GNU GPL ou simplesmente GPL, é a designação da [licença de software](#) para [software](#) idealizada por [Richard Matthew Stallman](#) em 1989. Richard Stallman criou a licença de acordo com as [definições de software livre](#) da Free Software Foundation

**Figura 6** - Arquitetura do ambiente de compilação e interpretação da linguagem Python



**Fonte:** Borges (2010).

Segundo Grus (2015) existe debate saudável sobre a melhor linguagem para o aprendizado de dados científicos. O Python tem vários recursos que o tornam adequado para aprender (e fazer) ciência de dados:

- É gratuito.
- É relativamente simples codificar (e, em particular, entender).
- Ele tem muitas bibliotecas úteis relacionadas à ciência de dados.

Diante das vantagens supracitadas acima, a linguagem de programação Python será utilizada para a codificação das rotinas responsáveis pelos processamentos das análises como: tratamento das amostras, remoção de outliers, etc, além de métodos quimiométricos, serviços *web* e responsável também pela persistência de toda informação proveniente do software no banco de dados.

#### 2.3.4 Linguagem de Programação Java

A linguagem de programação Java foi concebida para ser aplicada no controle de eletrodomésticos como TV, CD players, microcomputadores e etc., como simples ferramenta de programação, porém mostrou ser uma linguagem muito poderosa, superando as expectativas de sua criação, sendo assim passou a ser amplamente utilizada (LEITE, 2006).

Segundo Horstmann (2003), a linguagem Java possui muitos recursos adicionais, chamados de bibliotecas, que incluem funcionalidades extras para desenvolvimento de aplicações. Além disso, como a linguagem Java foi projetada para Internet ela possui dois atributos que a tornam muito adequadas: segurança e portabilidade. Outra grande vantagem é a portabilidade que permite um mesmo programa ser executado em sistemas operacionais diferentes, com pouca ou nenhuma adaptação, como Linux, Windows, Unix ou Macintosh. Isso é possível devido à máquina virtual Java, que é instalada no computador executor (HORSTMANN, 2003).

Visto que o período para o desenvolvimento do software é extremamente curto, a utilização de uma tecnologia foi escolhida o (JSF *Java Server Faces*) que é um *framework* Java de fácil manuseio. Conseqüentemente a codificação do *front end*, ou seja, a interface frontal da plataforma *web* será escrita na linguagem Java.

### 2.3.5 Serviços *Web*

O conceito sobre Serviços *Web* (do inglês, *Web Service*) surgiu nos anos 2000. Desde então, diferentes autores e empresas apresentaram as mais diferentes definições. Algumas delas são mostradas a seguir.

De acordo com a IBM (2004), um serviço *Web* é uma aplicação modular autossuficiente que pode ser descrita, publicada, localizada e invocada por meio de uma rede, em geral, a *World Wide Web*. Sua arquitetura é constituída de três partes: provedores de serviços, solicitante do serviço e agente de serviços. A implementação da arquitetura de um serviço *Web* deve permitir a segurança e a qualidade dos modelos de serviços facilitada pela configuração de um conjunto de pré-requisitos ambientais para controlar e gerenciar as interações.

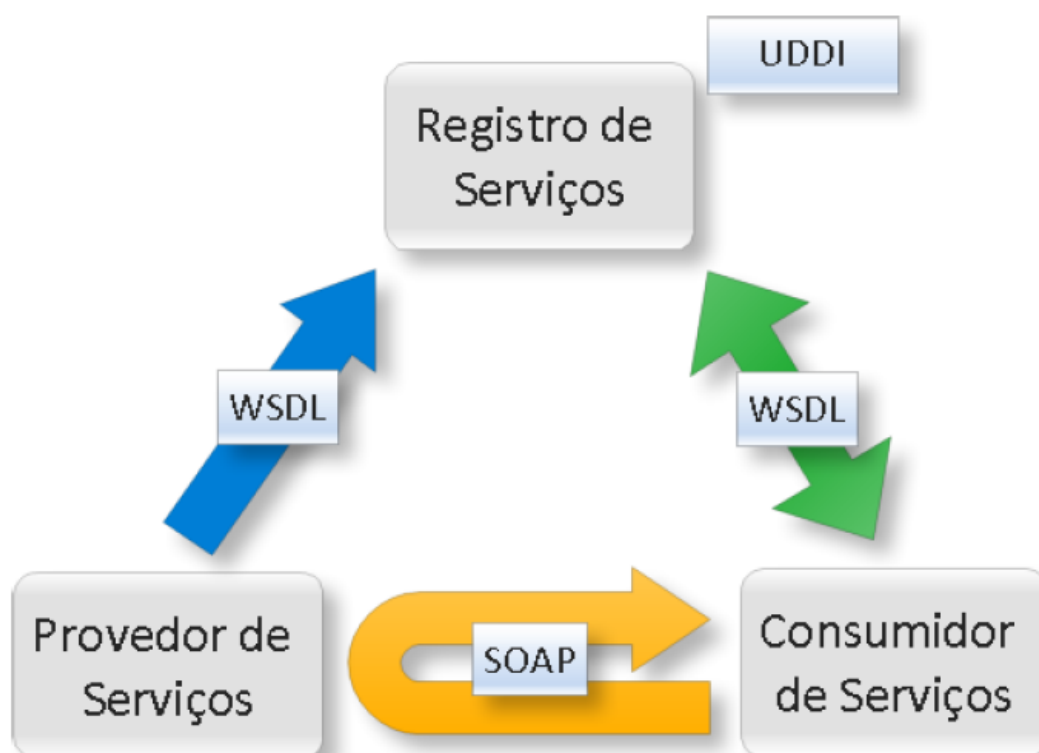
Segundo o consórcio W3C (2014), um serviço *Web* é uma solução utilizada na integração de sistemas e na comunicação entre aplicações diferentes. A implementação é baseada em tecnologias e protocolos, como HTTP, XML, SOAP, WSDL, entre outros. Usuários de serviços *Web* não precisam conhecer sua estrutura ou sua linguagem de programação para poder fazer uso dos serviços oferecidos.



Estes usuários somente precisam saber do que é necessário para a solicitação e obtenção da resposta sobre o serviço solicitado.

A arquitetura de um serviço *Web* pode também ser definida como um software que permite trocar informações. Destacam-se três elementos, o provedor de serviços, o registro de serviços e o consumidor de serviços descritos a seguir (Silva, 2009). Na Figura 7 é apresentada a arquitetura de um serviço *Web*.

**Figura 7** - Arquitetura de um serviço *Web*



Fonte: Silva (2009).

- Provedor de Serviços é responsável por conter as funcionalidades e disponibilizar esta operação como um serviço para que seja encontrada e consumida por outro sistema.
- Consumidor de Serviços – é responsável por consumir os diferentes serviços oferecidos pelos serviços *Web*.
- Registro de Serviços – é um local centralizado em que o provedor disponibiliza seus serviços *Web* e no qual um consumidor pode utilizar os serviços disponíveis.

Segundo a W3C (2014), o Oasis (2014) e a Odlis (2014), pode-se conceituar os protocolos e elementos necessários para criação de um serviço *Web*, dentre os protocolos existentes podemos destacar o REST (*Representational State Transfer*), este não representa uma arquitetura, mas sim um conjunto de restrições que, quando aplicadas na concepção de um sistema, cria um estilo de arquitetura de software. Um sistema no estilo REST é denominado RESTful e possui as seguintes características: (i) deve ser um sistema cliente-servidor, (ii) precisa ser independente de estado, ou seja, cada requisição deverá ser independente, (iii) deve ser uniformemente acessível, cada recurso deve ter um endereço exclusivo a um ponto de acesso válido. O núcleo da abordagem REST consiste na percepção de que, apesar do termo transporte em seu nome, o HTTP consiste em uma API e não um simples protocolo de transporte. Desta forma, o HTTP possui métodos bem definidos que correspondem às operações CRUD (*Create, Read, Update, Delete*). Cada requisição HTTP inclui um dos métodos apresentados no Quadro 1 para indicar qual a operação CRUD que deve ser realizada sobre o recurso.

**Quadro 1 - Métodos HTTP e Operações CRUD**

Método HTTP	Operação
POST	Criar um novo recurso a partir dos dados requisitados
GET	Lê um recurso
PUT	Atualiza um recurso
DELETE	Remove um recurso

**Fonte:** Autoria Própria (2019).

Provavelmente a maior razão por trás do sucesso atual dos serviços REST é a simplicidade do seu conjunto de regras básicas, portanto frente as necessidades e desafios existentes na implementação do software referido neste trabalho, esta arquitetura se apresenta como uma arquitetura viável para a construção dos serviços *web* que serão necessários neste trabalho.

### 2.3.6 UML (Unified Modeling Language)

A UML (sigla em inglês para *Unified Modeling Language*) é uma linguagem que se presta à modelagem de estruturas que irão compor uma aplicação, estando fortemente amparada em conceitos de Orientação a Objetos.

Uma boa modelagem é um princípio básico para o alcance de um bom software. Dentre as funções da modelagem, eles enfatizam as seguintes: comunicar a estrutura e o comportamento desejados do sistema, permitir uma melhor compreensão do sistema e gerenciar os riscos (Booch, 2005).

A UML é uma linguagem-padrão para a elaboração da estrutura de projetos de software. Ela pode ser empregada para a visualização, a especificação, a construção e a documentação de artefatos que façam uso de sistemas complexos de software (Booch, 2005).

Para modelagem do software, serão utilizados diversos artefatos da linguagem UML, que é amplamente empregada no desenvolvimento de sistemas orientados a objetos.

### 2.3.7 Desenvolvimento ágil de software

Assim que os métodos tradicionais de desenvolvimento de software passaram a não obter o nível de sucesso necessário na concretização dos projetos, surgiu a necessidade de criar metodologias para direcionar as equipes de desenvolvimento. Essas novas práticas são orientadas às pessoas pertencentes ao projeto e procuram trazer flexibilidade para sobreviver a um ambiente com mudanças constantes (Cockburn, 2002).

Os métodos ágeis surgiram como alternativas aos métodos tradicionais de desenvolvimento, que são baseados no seguimento de um plano bem definido, excesso de documentação e rigorosa padronização (Nerur et al., 2005). Portanto, em 2001, um grupo de profissionais reuniu-se com o objetivo de debater sobre formas de desenvolvimento de software, procurando uma alternativa aos processos excessivamente baseados em documentação e formalismo. Desse encontro se

origina o Manifesto pelo Desenvolvimento Ágil de Software, ou simplesmente Manifesto Ágil, o qual possui os seguintes valores (Beck et al., 2001):

- Indivíduos e interações acima de processos e ferramentas;
- Software funcionando acima de documentação abrangente;
- Colaboração com o cliente acima de negociação de contratos;
- Resposta a mudanças acima de seguir um plano.

Portanto, as metodologias ágeis se desenvolveram em um esforço para sanar fraquezas reais e perceptíveis da engenharia de software convencional. Apesar de oferecer benefícios importantes, o desenvolvimento ágil não é indicado para todos os projetos, produtos, pessoas e situações, também não é antítese da prática de engenharia de software consistente e pode ser aplicado como uma filosofia em geral para todos os trabalhos de software (Pressman, 2011).

Dentre as metodologias ágeis utilizadas para o desenvolvimento de software, está o SCRUM, esta escolhida como metodologia a ser seguida neste trabalho.

### 2.3.8 Metodologia de Desenvolvimento ágil SCRUM

O método de desenvolvimento SCRUM foi proposto em 1995, por Ken Schwaber (2004), num momento em que ficou claro para a maioria dos profissionais que o desenvolvimento de software não era algo que poderia ser planejado, ou simplesmente algo manufaturado, estimado e concluído com sucesso usando um método comum e burocrático. O método SCRUM baseia-se no trabalho de Pittman (1993) e Booch (1995) e adere aos princípios do desenvolvimento ágil de software.

Para o SCRUM, a centralidade da atividade está na maioria dos processos durante o desenvolvimento que não pode ser previsto. Por isso, aborda o software em desenvolvimento de maneira flexível. As únicas duas partes que estão totalmente definidas durante um projeto de desenvolvimento de software são a primeira e a última fase (planejamento e fechamento). No centro, o produto final é desenvolvido por várias equipes em uma série de entregas periódicas e flexíveis chamadas *sprints*. Não há novos requisitos anexados durante as *sprints*. Isso garante que o produto final esteja sendo desenvolvido com uma alta probabilidade

de sucesso, mesmo dentro de uma mudança constante no ambiente inserido. Este ambiente, que inclui fatores como concorrência, tempo e pressão financeira, mantém sua influência em desenvolvimento até a fase de encerramento.

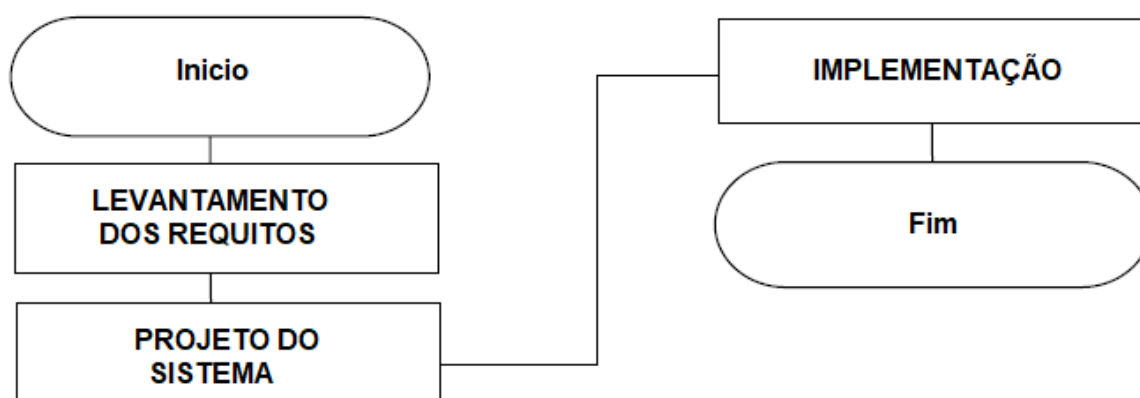
O SCRUM é uma ferramenta que permite controlar de forma eficaz o fluxo de trabalho, potencializando a busca de um objetivo em comum. O SCRUM visa à redução dos riscos de negócios do projeto pela colaboração com os clientes e demais partes interessadas durante todo o seu decorrer. Os riscos também são reduzidos com a produção em ciclos curtos e entregas frequentes de partes prontas do produto, partindo-se das mais importantes em direção às menos importantes (Sabbagh, 2014).

No decorrer do processo de desenvolvimento do deste trabalho, serão aplicados os artefatos pertinentes a metodologia de desenvolvimento ágil Scrum, que é uma ferramenta que permite controlar de forma eficaz o fluxo de trabalho, potencializando a busca de um objetivo em comum.

### 3 PROCEDIMENTOS

Neste capítulo serão descritas as ações pertinentes e necessárias ao desenvolvimento do software proposto neste trabalho. Para o desenvolvimento do software de análises sanguíneas por espectroscopia foram definidas algumas etapas observadas na Figura 8 para conduzir a execução do projeto. Estas etapas serão detalhadas a seguir.

**Figura 8** - Fluxograma do Desenvolvimento do software



Fonte: Autoria Própria (2019).

#### 3.1 Levantamento dos requisitos

O levantamento de requisitos é uma etapa que precede a codificação do software em si, porém é tão importante quanto. Consiste em obter e mapear informações relevantes relativas ao contexto do software a ser desenvolvido junto aos seus clientes ou usuários finais. Para um software que se propõe a automatizar um processo é de suma importância documentar o que o sistema espera receber como dados de entrada, como o sistema deve lidar com estes dados e, por fim, como deve fornecê-los.

Os requisitos funcionais de usuários definem recursos específicos que devem ser fornecidos pelo sistema (Sommerville, 2008). Já requisitos não funcionais estão

relacionados a características de utilização da aplicação, como segurança, desempenho e confiabilidade.

Devido a utilização da metodologia ágil SCRUM, os requisitos não foram todos coletados antes do início da codificação do software, eles foram coletados também no decorrer do desenvolvimento, assim que necessários, pois um dos principais objetivos do SCRUM é a entrega contínua, sempre buscando entregar valor para o cliente ou usuário.

Os artefatos resultantes do levantamento de requisitos são alguns diagramas da UML, nesse trabalho serão produzidos o diagrama de atividades e o diagrama de caso de uso, apresentados na seção seguinte. Para a construção destes diagramas foi utilizado o **Astah Community**. Assim como os diagramas, também são apresentados os protótipos das interfaces *web* e *mobile* do software. Os Protótipos podem ser utilizados na fase de testes ou planejamento de um software. Na engenharia de software, protótipo é um sistema modelo sem funcionalidades inteligentes como o acesso ao banco de dados, é um artefato que pode conter apenas a representação gráfica das possíveis funcionalidades de uma interface.

Do mesmo modo que os requisitos coletados, foi necessário identificar também quais métodos estatísticos e algoritmos são atualmente utilizados na análise sanguínea baseada em espectroscopia. Assim que explicados pelos usuários chave que são responsáveis por repassar este conhecimento, foi necessário compreender os conceitos e o funcionamentos destes métodos e algoritmos. Os métodos PLS e PCA, serão chamados a partir da biblioteca Scikit-learn, outros algoritmos necessários como o KNN (*K Nearest Neighbor*) e o *Kennard-stone* serão reescritos neste software por meio linguagem de programação Python, por sua vez escolhida para compor todas as rotinas da análise baseada em espectroscopia. O código escrito para os algoritmos KNN e *Kennard-stone* estão dispostos no capítulo resultados e discussões.

### 3.2 Projeto do Sistema

Tratando-se de desenvolvimento de software, é importante definir cautelosamente toda estrutura do projeto desde o princípio, isto é, sua arquitetura.

Um projeto bem arquitetado pode aumentar a performance da aplicação, pois permite o aumento de escala de processamento e do volume de dados gerados em suas operações. A escalabilidade é fundamental, pois expõe a capacidade de crescimento uniforme da carga de trabalho submetida a um software, ou a capacidade do software estar preparado para um possível crescimento.

Para a modelagem dos processos de negócios utilizando a BPMN do inglês (*Business Process Model and Notation*), foi utilizado o software **Bizagi Modeler**. A notação BPMN empregada nos diagramas, especifica o processo de negócio em um diagrama de fácil leitura, tanto para os usuários leigos quanto para os usuários que possuem uma sensibilidade analítica no processo representado. A diagramação BPMN foi escolhida por quê é intuitiva e permite a representação de detalhes complexos do processo. A simbologia BPMN serve como uma linguagem padrão, colocando um fim na lacuna de comunicação entre a modelagem do processo e sua execução. Os processos levantados na etapa de requisitos, estão dispostos na subseção dos requisitos funcionais no próximo capítulo, estes estão representados da forma que são executados atualmente, posteriormente são apresentados os fluxos destes processos, após a melhoria aplicada por meio do software proposto neste trabalho.

Após a leitura e modelagem dos principais processos a serem melhorados, foi desenvolvido o projeto do banco de dados. Um projeto de banco de dados é subdividido em etapas onde o objetivo é a criação de um banco de dados otimizado que atenda as expectativas dos usuários do software, portanto nesse contexto, os modelos de dados são muito importantes para a transmissão de ideias entre o usuário e o desenvolvedor, bem como facilitar a manutenção e crescimento do banco de dados no futuro. Para a construção do modelo relacional, que é o projeto lógico do banco de dados foi utilizado o **MySQL Workbench**, por meio dessa ferramenta foi construído o diagrama do modelo relacional, que será apresentado no próximo capítulo. Para a representação do banco de dados físico, foi escolhido **pgAdmin 4** como plataforma de administração e desenvolvimento dos códigos responsáveis por fazer a construção e a manipulação do banco de dados do software.



### 3.3 Implementação

Nessa etapa, o software é codificado a partir de uma descrição computacional das funcionalidades desejadas em uma linguagem de programação, onde se torna possível a compilação e geração do código executável. Em um processo de desenvolvimento orientado a objetos, o qual será utilizado neste projeto, a implementação se dá, definindo as classes de objetos, pode-se também utilizar na implementação ferramentas de software e bibliotecas de classes preexistentes para agilizar a atividade, como também o uso de ferramentas CASE, que dinamizam o processo de desenvolvimento, nas várias atividades, onde inclui-se geração de código-fonte, documentação, etc.

Para a codificação das rotinas que serão escritas na linguagem Python, Java e JavaScript, foram escolhidas as IDE's (ambientes de desenvolvimento integrado) **PyCharm, NetBeans e Visual Studio Code**, respectivamente.

Quanto as bibliotecas utilizadas no desenvolvimento do software, destaca-se a **Scikit-Learn** que apoia a implementação de vários algoritmos de regressão, classificação e agrupamento utilizados neste trabalho. Essa biblioteca também é usada para a construção dos modelos de calibração por meio do método PLS. Além de oferecer suporte e robustez exigidos por ambientes de produção, a biblioteca Scikit-Learn também foca em usabilidade, clareza de código, colaboração, documentação e desempenho. Neste último quesito, as funções/rotinas disponibilizadas dispõem das facilidades da linguagem Python como a interação com as bibliotecas numéricas e científicas.

## 4 RESULTADOS E DISCUSSÕES

Este capítulo traz a descrição das etapas que constituem o processo de desenvolvimento do software abordado neste trabalho, tratando desde rotinas, arquitetura e diagramação, até detalhes de usabilidade da interface do projeto. Além disso, apresenta-se uma discussão sobre os resultados obtidos pela execução das rotinas envolvidas no processo de construção, calibração e validação (previsão) no modelo proposto dentro do software.

### 4.1 Requisitos Funcionais

Nesta subseção é apresentado o levantamento de requisitos para cada uma das funcionalidades disponíveis na aplicação que também serão utilizados para validar o software após a conclusão das implementações. A seguir, são apresentadas as descrições dos requisitos funcionais catalogados para o desenvolvimento deste sistema, descrevendo detalhes das tarefas que devem ser executadas.

Por convenção, os requisitos serão identificados e enumerados a partir da formatação a seguir:

[identificador do tipo de requisito] + [número do requisito]

RF1 - O sistema deve possuir uma interface de acesso em que os usuários podem autenticar-se fornecendo suas credenciais para ter acesso às funcionalidades disponíveis.

RF2 – O sistema deve possuir uma tela principal, com um menu lateral, em que seja possível acessar as demais funcionalidades.

RF3: O sistema deve permitir a inclusão, alteração ou exclusão de modelos de calibração. O usuário responsável por gerir estes modelos deve informar qual o método de análise de referência que está sendo correlacionado com o modelo em questão.

RF4 O sistema deve possuir um controle sobre cada amostra inserida no seu banco de dados. Armazena-se informações como: data de coleta, identificação e informações adicionais.

RF5: O sistema deve permitir que o usuário responsável por gerenciar o modelo inclua novas amostras no modelo de calibração, para tal, deve-se informar os valores que foram obtidos por meio do método de referência.

RF6: O sistema deve permitir o controle de parâmetros que são analisados para cada modelo de calibração previamente registrado.

RF7: O sistema deve permitir uma predição de uma amostra por meio de uma interface, informando apenas o modelo e o identificador da amostra analisada.

RF8: O sistema deve conter uma consulta dos resultados da análise espectral realizada por meio da calibração multivariada, em que o usuário consiga visualizar os resultados obtidos pelo método de referência e pelo modelo de calibração multivariada. Também deverá ser apresentado um percentual de assertividade entre a análise proposta por este trabalho frente ao método convencional.

RF9: O aplicativo para o dispositivo móvel deve ser integrado com leitor infravermelho portátil Scio fabricado pela empresa *Consumer Physics*, tal que o aplicativo consiga capturar os vetores espectrais coletados pelo equipamento.

RF10: O aplicativo para o dispositivo móvel deve fornecer ao usuário a possibilidade de selecionar um determinado paciente e fazer a coleta do espectro. Após coletar o

vetor emitido pelo NIR, o aplicativo requisita o serviço *web*, enviando as informações coletadas para o processamento.

RF11: O aplicativo para o dispositivo móvel deve conter uma interface de consulta dos resultados das análises feitas anteriormente.

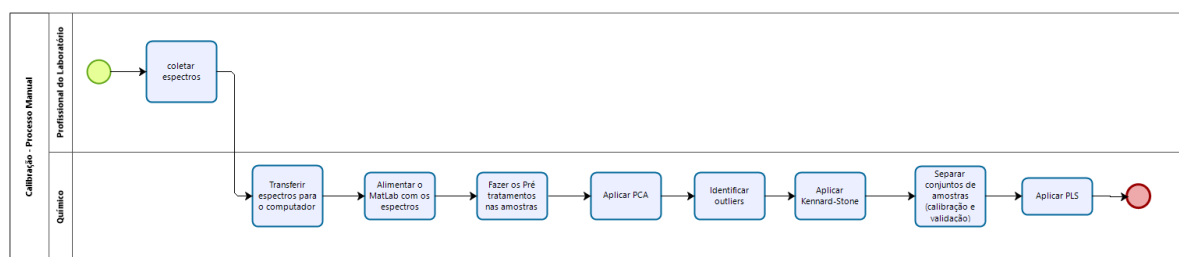
Durante a coleta dos requisitos funcionais também foram conhecidos os processos que precisavam ser melhorados. Foi observado que os principais processos de uma análise sanguínea eram a calibração do modelo multivariado e a predição de uma amostra não conhecida. Estes processos necessitavam ser automatizados por meio do software, pois eram etapas complexas e lentas para se realizar manualmente. A análise do processo atual é necessária para detecção de atividades que podem ser melhoradas, como ineficiências e gargalos, com o objetivo de definir suas metas e objetivos, o fluxo de trabalho, controles e integração com outros processos para que ele contribua de forma significativa na entrega de valor ao usuário final.

#### 4.1.1 Processo de criação e calibração do modelo multivariado

O processo de calibração de um modelo demanda um conhecimento quimiométrico solidificado, uma vez que o modelo pode prover erros significativos e levar a decisões erradas. No entanto, para que o software seja útil, é necessário que se considere a realização de medidas a partir da metodologia utilizada como referência nos moldes atuais (ou seja, via punção venosa), caso o resultado produzido indique anormalidade. Outro ponto importante a ser salientado é que quanto mais medidas forem realizadas, melhor será a capacidade preditiva do modelo. Isso deve-se pelo fato de se incluir uma maior variabilidade no modelo, o que demanda atualizações rotineiras a fim de se incorporarem as novas amostras no conjunto de calibração.

Como observado na Figura 9, atualmente todas as etapas são realizadas manualmente, iterando com os métodos e amostras em cada uma delas.

**Figura 9** - Processo de calibração feito manualmente



Powered by  
bizagi  
Modeler

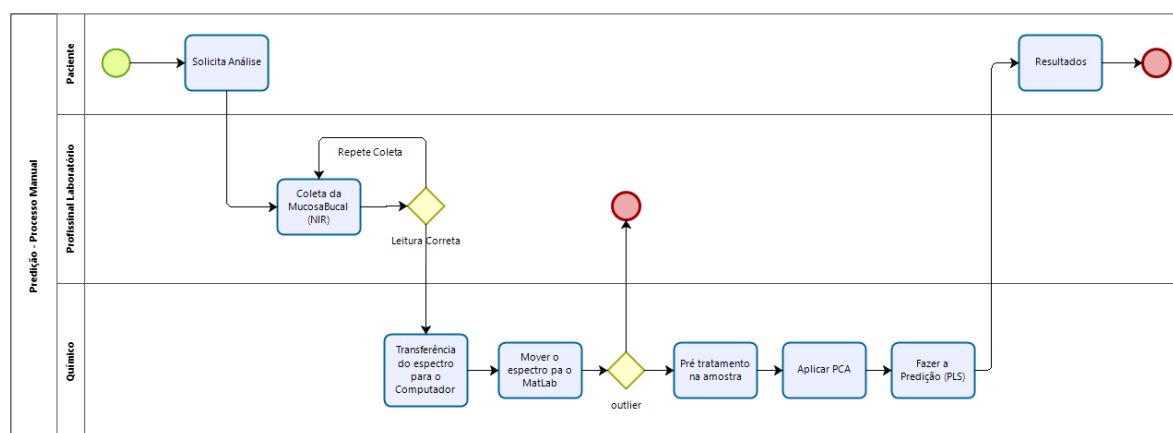
**Fonte:** Autoria Própria (2019).

Durante o levantamento deste processo, foi muito importante o diálogo com os desenvolvedores do método de análise baseado em espectroscopia, pois são conhecedores do processo atual, e possuem sensibilidade crítica para dizer onde estão as dificuldades. Devido ao fato de que as etapas estão do processo de calibração são chamadas isoladamente, pelo usuário executor do processo, exige que o profissional execute uma sequência específica das etapas representada no diagrama acima, porém, como o processo não está normalizado, atualmente está vulnerável a falhas humanas, além de ser custoso operacionalmente.

#### 4.1.2 Processo de Predição de Amostra

O processo de predição é manual, similarmente ao processo de calibração do modelo. exige que as amostras estejam tratadas da mesma forma que as amostras já existentes no modelo. A dificuldade de execução deste processo é a complexidade técnica necessária para estabelecer uma conexão entre o modelo, o equipamento de medida e as demandas operacionais de manutenção do modelo, sendo que cada uma delas possui uma ou mais etapas internas para ser executada por completo.

**Figura 10** - Processo de predição de amostra manual



Powered by  
bizagi  
Modeler

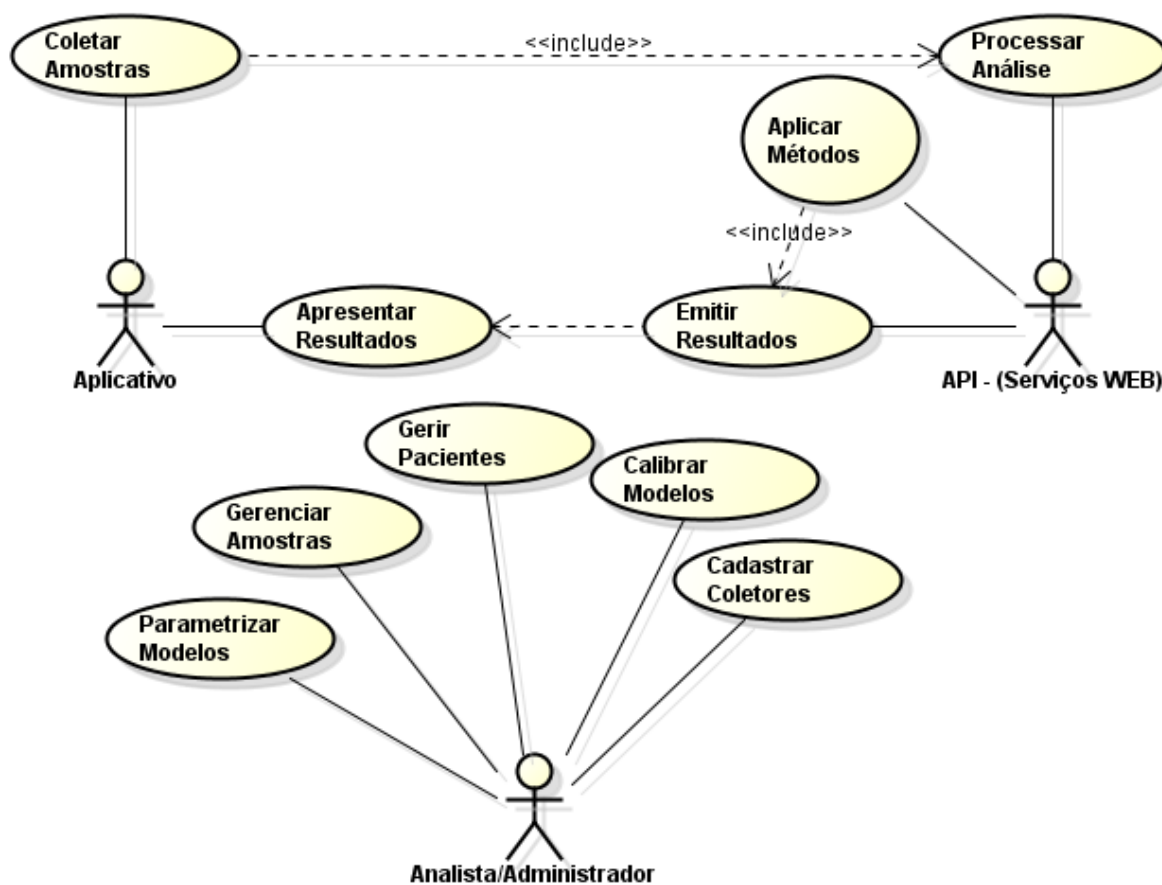
**Fonte:** Autoria Própria (2019).

Durante a coleta dos requisitos por meio de entrevistas e outros materiais de apoio foram construídos diagramas de caso de uso e o diagrama atividades com o propósito de elucidar as necessidades requisitadas e responder dúvidas dos usuários e desenvolvedores. A seguir estão apresentados os diagramas elaborados para o desenvolvimento do software em questão.

#### 4.1.3 Diagrama de Caso de Uso

Os casos de uso são utilizados para mapear funcionalidades que o sistema deve possuir e como os atores devem interagir com elas. Após especificadas as funcionalidades a serem desenvolvidas para o sistema por meio da utilização do diagrama de casos de uso, foram validadas junto aos usuários as interações e requisitos necessários para o funcionamento do software. A diagramação provém uma visão geral da forma que os atores vão interagir com o sistema assim como as respostas que o sistema deve prover em cada interação. A Figura 11 ilustra o diagrama de caso de uso desenvolvido para o software proposto neste trabalho.

**Figura 11** - Diagrama de Caso de Uso



**Fonte:** Autoria Própria (2019).

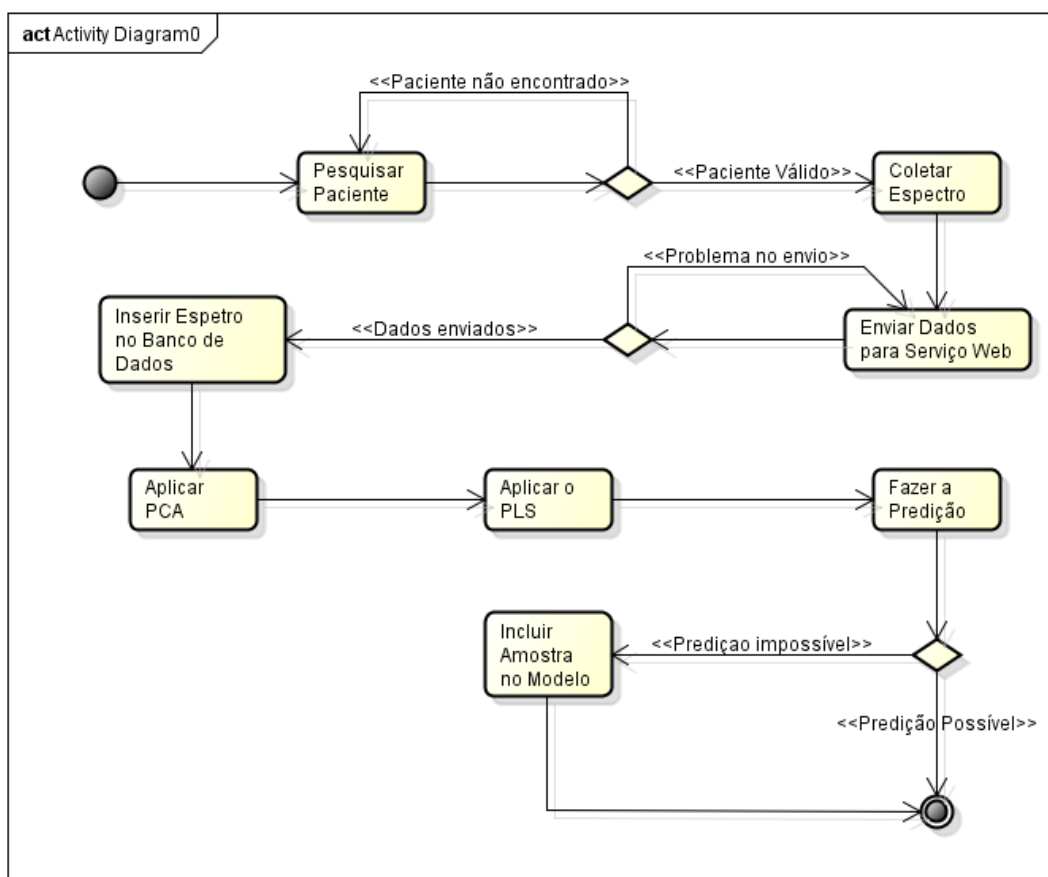
#### 4.1.4 Diagrama de Atividades

O diagrama de atividade definido pela UML representa os fluxos conduzidos por processamentos. É essencialmente um gráfico de fluxo que apresenta a sequência de controle entre as atividades. Segundo Lima (2005), este diagrama representa a visão dinâmica do sistema. A principal característica que o diferencia de um fluxograma é o fato dele poder representar atividades paralelas.

Os diagramas de atividade permitem modelar os caminhos lógicos que o sistema pode seguir e a interdependência das atividades, além de representar atividades que podem acontecer paralelamente. Este diagrama também é utilizado na modelagem de processos de negócio para fornecer aos envolvidos no desenvolvimento do software a compreensão do funcionamento de processos de

trabalho da empresa. A Figura 12 apresenta o diagrama com levantamento do fluxo das atividades exercidas pelo software durante o processo de análise.

**Figura 12 - Diagrama de Atividade do Processo de Predição da Amostra**



**Fonte:** Autoria Própria (2019).

Acima pode ser observado de forma simples, o fluxo do sistema, onde uma amostra após ser coletada é enviada automaticamente para o serviço *web*, posteriormente, esta amostra é submetida a tarefas que são parametrizáveis, e dependendo do resultado da avaliação (feita através do software) o fluxo pode tomar caminhos diferentes.



## 4.2 Algoritmos e métodos

### 4.2.1 Algoritmo *Kennard-Stone* (Kennard & Stone, 1969)

Durante a calibração do modelo multivariado é necessário separar as amostras em dois conjuntos, calibração e validação, para esta classificação foi escrito na linguagem Python o algoritmo de *Kennard-Stone*, que recebe como parâmetros de entrada a matriz com os espectros e o número de amostras que devem conter no conjunto de calibração. O retorno emitido pelo algoritmo são quais amostras deverão estar no conjunto de calibração.

**Figura 13** - Algoritmo *Kennard-Stone*

```

600 def kennardStone(X, k, precomputed=False):
601     n = len(X) # number of samples
602     # print("Input Size:", n, "Desired Size:", k)
603     now = datetime.now()
604     print("Executando KennStone")
605     print(now)
606     assert n >= 2 and n >= k and k >= 2, "Error: number of rows must >= 2"
607
608     # pair-wise distance matrix
609     dist = metrics.pairwise_distances(X, metric='euclidean', n_jobs=-1)
610
611     # get the first two samples
612     i0, il = np.unravel_index(np.argmax(dist, axis=None), dist.shape)
613     selected = set([i0, il])
614     k -= 2
615     # iterate find the rest
616     minj = i0
617     while k > 0 and len(selected) < n:
618         mindist = 0.0
619         for j in range(n):
620             if j not in selected:
621                 mindistj = min([dist[j][i] for i in selected])
622                 if mindistj > mindist:
623                     minj = j
624                     mindist = mindistj
625             # print(selected, minj, [dist[minj][i] for i in selected])
626             selected.add(np.int(minj))
627             k -= 1
628
629     now = datetime.now()
630     print(now)
631     return selected

```

Fonte: Autoria Própria (2019).

#### 4.2.2 Algoritmo KNN (*K Nearest Neighbor*)

O algoritmo KNN (*K Nearest Neighbor*) é um dos algoritmos mais utilizados em aprendizado de máquina e também um dos mais simplistas, analisando seu processo de cálculo de aproximação. Este algoritmo pode ser aplicado em diversos segmentos de negócio, logo também se aplica em diversos problemas como finanças, saúde, ciência política, reconhecimento de imagem e reconhecimento de vídeos. Permite-se também a sua utilização para classificação e regressão. Ele também é conhecido como de aprendizado lento, pois não necessita de dados de treinamento para se gerar o modelo resultante. O funcionamento do algoritmo requer que se informe o parâmetro *k* que direcionará a quantidade de vizinhos próximos (*neighbor* em inglês).

Por meio da linguagem Python, foi reescrito o algoritmo KNN que é apresentado na Figura 14.

**Figura 14** - Algoritmo KNN

```
def fit(self, X, y=None):
    """Fit detector. y is optional for unsupervised methods.

    Parameters
    -----
    X : numpy array of shape (n_samples, n_features)
        The input samples.

    y : numpy array of shape (n_samples,) , optional (default=None)
        The ground truth of the input samples (labels).
    """

    # validate inputs X and y (optional)
    X = check_array(X)
    self._set_n_classes(y)

    if self.metric_params is not None:
        self.tree_ = BallTree(X, leaf_size=self.leaf_size,
                              metric=self.metric,
                              **self.metric_params)
    else:
        self.tree_ = BallTree(X, leaf_size=self.leaf_size,
                              metric=self.metric)
    self.neigh_.fit(X)

    dist_arr, _ = self.neigh_.kneighbors(n_neighbors=self.n_neighbors,
                                        return_distance=True)
    dist = self._get_dist_by_method(dist_arr)

    self.decision_scores_ = dist.ravel()
    self._process_decision_scores()

    return self
```

Fonte: Autoria Própria (2019).

Este algoritmo foi aplicado na detecção das amostras anômalas existentes no modelo.

### 4.3 Arquitetura do software

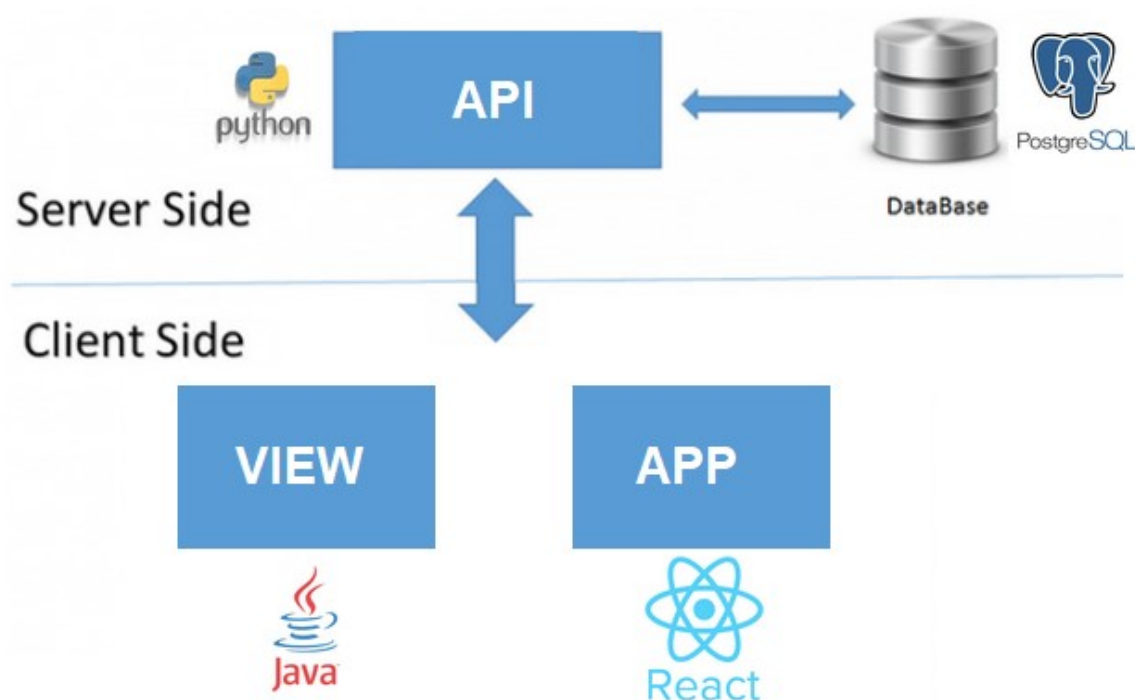
Para este projeto optou-se pela implementação do clássico modelo de arquitetura cliente-servidor. O modelo cliente-servidor é uma estrutura distribuída em que a aplicação é particionada em dois módulos, um sendo cliente (*client-side*), o qual tem o papel de solicitar um determinado recurso; e o outro sendo o servidor (*server-side*), que é responsável por fornecer ao cliente o recurso requerido. Esta arquitetura satisfaz requisitos de acessibilidade, portabilidade, permite uma utilização escalável do software

Neste contexto, o software em questão segue o modelo supracitado alocando no módulo cliente a parte de interface o qual o usuário pode interagir e inserir dados de entrada que alimentam o sistema. Este projeto possui duas implementações de interface dado a necessidade de viabilizar o acesso a aplicação tanto por meio de navegadores *web* quanto para dispositivos móveis. As interfaces *web* e dos dispositivos móveis foram desenvolvidas utilizando-se as tecnologias JavaServer Faces (JSF) e React Native, respectivamente.

No caso do *server-side* encontra-se a API (do inglês, *Application Programming Interface*) deste sistema, que é a ponte de conexão entre o *front end* do software e o banco de dados, todavia ela é transparente ao usuário comum. Ela é implementada utilizando a linguagem Python e usufrui de recursos de bibliotecas populares no universo da ciência de dados como a Scikit-learn, Numpy e Pandas. A API, além de ser responsável por disponibilizar os recursos requeridos, contém todos os métodos quimiométricos, algoritmos de pré-processamento, classificação, aprendizado de máquina e modelos estatístico que são executados de forma procedural e automatizada a fim de prover os dados requeridos pelo *client-side*. Cada requisição do usuário (no *front-end*) produz um processo que pode produzir resultados que são enviados para o autor da requisição e (em alguns casos) persistidos em um banco de dados PostgreSQL. É importante destacar que na arquitetura proposta o acesso ao banco de dados é estritamente restrito a esta API.

O fluxo de execução desse sistema inicia-se pela interface, em que o usuário insere os dados de entrada pelo *front-end*. Estes são formatados e enviados ao servidor por meio de requisições HTTP, que atuam por meio de solicitações para um determinado recurso de um serviço *web*. O servidor por sua vez processa os dados recebidos e envia uma resposta em formato JSON para requisitante, que são exibidos ao usuário na interface *web* ou no aplicativo do dispositivo móvel. A Figura 15 ilustra a arquitetura com suas respectivas tecnologias que são utilizadas na construção do software.

**Figura 15** - Arquitetura do Software



**Fonte:** Autoria Própria (2019).

Após a definição da arquitetura do software foram formulados os fluxos de interação que o software pode seguir e definidos detalhadamente o comportamento esperado diante de eventuais cenários. Portanto, nas próximas seções são apresentados os diagramas que complementam o entendimento das características do software.

### 4.3.1 Projeto do Banco de Dados

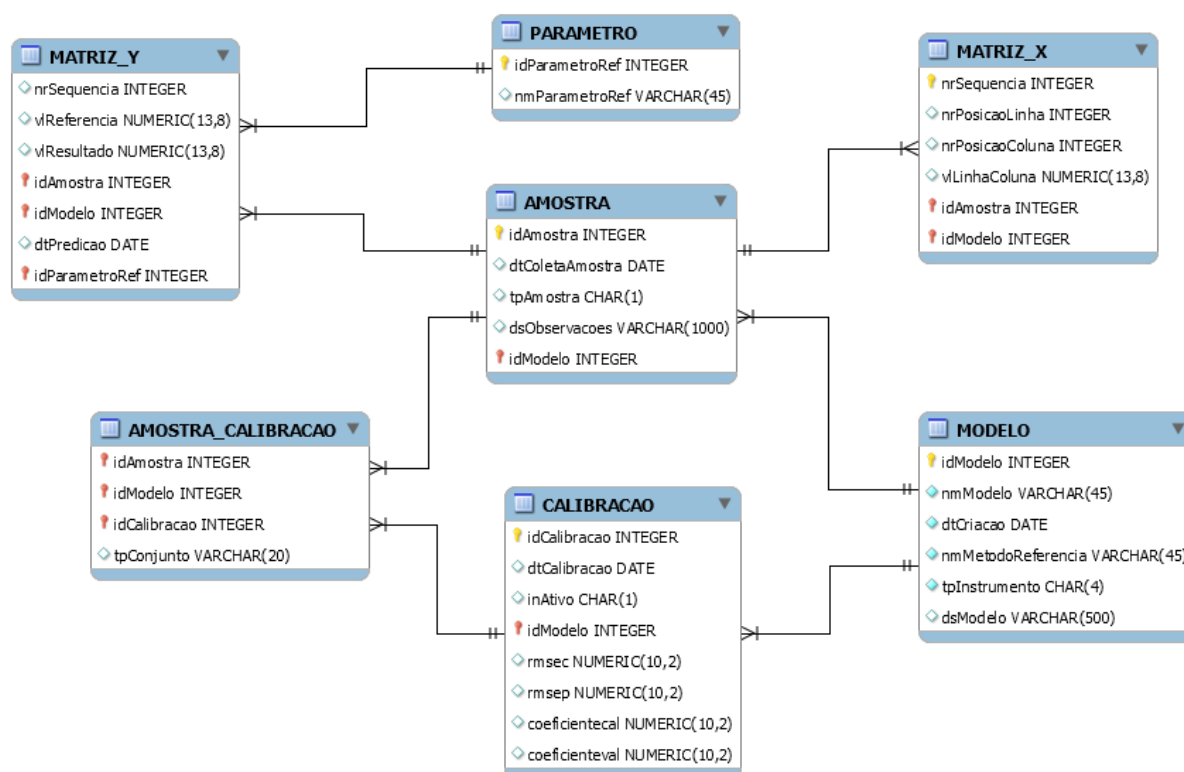
Para o armazenamento dos dados que serão coletados ou gerados pelo sistema, foi desenvolvido o projeto de banco de dados relacional, que está ilustrado na Figura 16. Esse diagrama do banco de dados foi projetado com o auxílio da ferramenta de modelagem MySQL Workbench 6.3 CE. Foi utilizada a notação de modelo simplificada da ferramenta e a notação crow's foot (IE) para os relacionamentos. As entidades são representadas graficamente como retângulos com os cantos arredondados. A chave primária de uma entidade é formada por um ou mais atributos cujo valor identifica unicamente uma ocorrência da entidade. Os relacionamentos são classificados com fortes e fracos, onde o relacionamento forte é empregado em relacionamentos que são necessários a utilização da chave primária da tabela de origem como chave primária/estrangeira na entidade de destino, assim também, os relacionamentos fracos são utilizados quando a chave primária da entidade de origem é utilizada apenas como chave estrangeira na entidade de destino.

As relações do banco de dados principal desenvolvido neste trabalho estão organizadas da seguinte forma:

- **AMOSTRA**: Responsável por armazenar todas as amostras que compõe os modelos, independentemente de sua classificação.
- **PARAMETRO**: Relação onde serão armazenados os parâmetros avaliados em cada calibração.
- **MODELO**: Entidade onde serão armazenadas as informações básicas de cada modelo de calibração multivariado gerido pelo software.
- **MATRIZ\_X**: Entidade onde serão armazenados os espectros, por meio do relacionamento com a entidade AMOSTRA, será possível armazenar diversas medidas para uma única amostra, de modo que um espectro que possui 125 posições possa ser armazenado por meio destas duas entidades aninhadas.
- **MATRIZ\_Y**: Entidade responsável pelo armazenamento do valor de referência do método de análise padrão da amostra (se conhecida esta medida), juntamente com o valor predito e a data de sua predição.

- **CALIBRACAO** e **AMOSTRA\_CALIBRACAO**: por meio destas entidades será possível identificar as principais informações de uma calibração de um modelo, quais amostras participaram de uma determinada calibração, assim como sua classificação na corrente calibração, pois uma amostra pode ser classificada como, amostra de calibração, amostra de validação ou *outlier*.

Figura 16 - Modelo Relacional



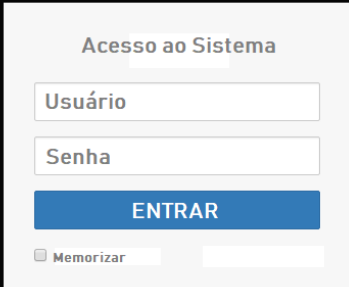
Fonte: Autoria Própria (2019).

Em projetos de software é comum a utilização de modelos para representar tanto a estrutura quanto o comportamento do sistema, de modo que, com base neles possa ser construído e desenvolvido o modelo executável do projeto, que seria basicamente o sistema materializado. A estrutura representa o que é estático, que não muda de estado com a utilização do sistema. O comportamento representa o que é dinâmico em um sistema, isto é, reage de maneiras diferentes de acordo com ações do próprio sistema ou do usuário.

### 4.3.2 Interfaces do software

Após o processo de levantamento de requisitos, foram elaborados protótipos de cada uma das telas a serem desenvolvidas. Os protótipos tiveram como finalidade viabilizar uma pré-visualização do resultado final do sistema e, conseqüentemente, possibilitar que o desenvolvedor constate possíveis problemas (estética, usabilidade, entre outros) e redefina os requisitos antes da codificação. Durante a fase de codificação do software foram construídas as interfaces definitivas do software proposto.

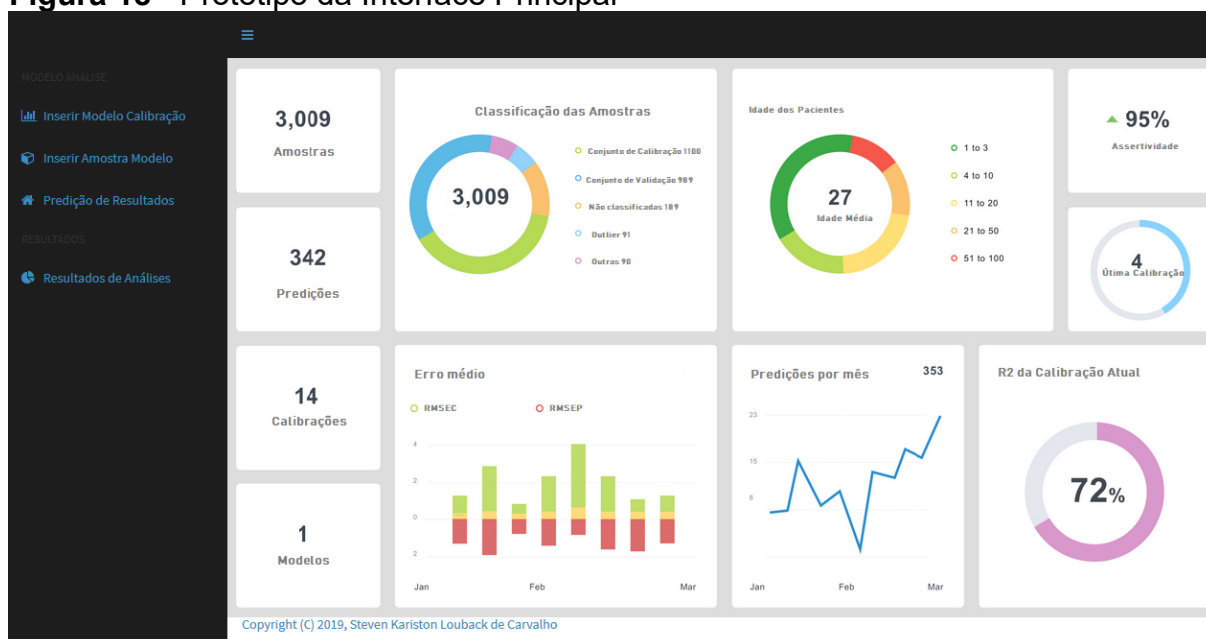
**Figura 17** - Protótipo da Interface de Acesso ao Sistema

A imagem mostra um protótipo de uma interface de acesso ao sistema. O formulário é centralizado em um fundo preto. No topo do formulário, o título "Acesso ao Sistema" está em uma fonte cinza. Abaixo dele, há dois campos de entrada de texto: "Usuário" e "Senha", ambos com bordas cinzas e fundo branco. Abaixo dos campos, há um botão azul com o texto "ENTRAR" em letras brancas maiúsculas. Na base do formulário, há uma opção "Memorizar" com uma caixa de seleção desmarcada e um campo de entrada de texto adjacente.

**Fonte:** Autoria Própria (2019).

O protótipo da interface de acesso ao sistema exibida pela Figura 17, será a primeira interface em que o usuário terá contato. Por meio de credenciais únicas para cada usuário, pode-se realizar o acesso ao software. Esta rotina é essencial para a segurança e confiabilidade do software, pois o acesso é permitido somente após a validação das credenciais de cada usuário.

**Figura 18 - Protótipo da Interface Principal**



**Fonte:** Autoria Própria (2019).

A interface principal do software foi desenvolvida com o objetivo de ser simples e objetiva. Por meio do menu lateral, as principais funcionalidades do software podem ser acessadas, enquanto na área central serão dispostas informações importantes para o usuário, como a quantidade e as classificações das amostras registradas no banco de dados, a quantidade de predições realizadas e os modelos de calibração existentes. Com este recurso o usuário pode rapidamente visualizar as informações importantes sem a necessidade de acessar outras telas.



**Figura 19 - Interface de Gerenciamento de Modelos**

**Dados Modelo** Cadastrar Modelo de Dados

**Cód. Modelo:**

**Nome Modelo:**

**Método Ref.:**

**Informações do Modelo:**  
  
 1000 caracteres restantes.

**Nome Parâmetro**

Cód. Modelo	Nome Modelo	Método Ref.	Instrumento Coletor	Ações
2	CLOROFILA - OLEO DE SOJA DEGOMADO	AOCS CC13D-55/2009	IMG	<input type="button" value="✎"/> <input type="button" value="🗑"/>
3	GLICOSE - SALIVA	REAGENTE	NIR	<input type="button" value="✎"/> <input type="button" value="🗑"/>

Fonte: Autoria Própria (2019).

Para a inclusão ou manutenção de um modelo de calibração multivariada, foi construída uma interface que permite que usuário descreva o nome do modelo, selecione o método de coleta dos dados, descreva qual o método de análise utilizado como referência para fazer a calibração do modelo e, por fim, informe qual o parâmetro a ser analisado.

**Figura 20 - Interface de Inclusão de Amostras**

**Inclusão Amostra no Modelo** Carregar Amostra para o Banco de Dados

**Modelo de Dados/Instrumento Coletor:**

**Data Amostragem:**

**Identificação da Amostra:**

**Descrição da Amostra:**  
  
 987 caracteres restantes.

**Espectro da Amostra:**

7759 caracteres restantes.

Fonte: Autoria Própria (2019).

Para a inclusão das amostras por meio da plataforma *web*, foi desenvolvida uma interface conforme a Figura 20. Por meio dela, o usuário faz a inclusão da amostra individualmente de acordo com o modelo selecionado, informando a data de amostragem, a identificação da amostra, a descrição e o espectro coletado pelo NIR.

**Figura 21** - Interface para a Predição

Predição de Resultados Predição de Resultados por Calibração Multivariada

Modelo de Dados: Seleccione | ID Amostra: 103

Buscar

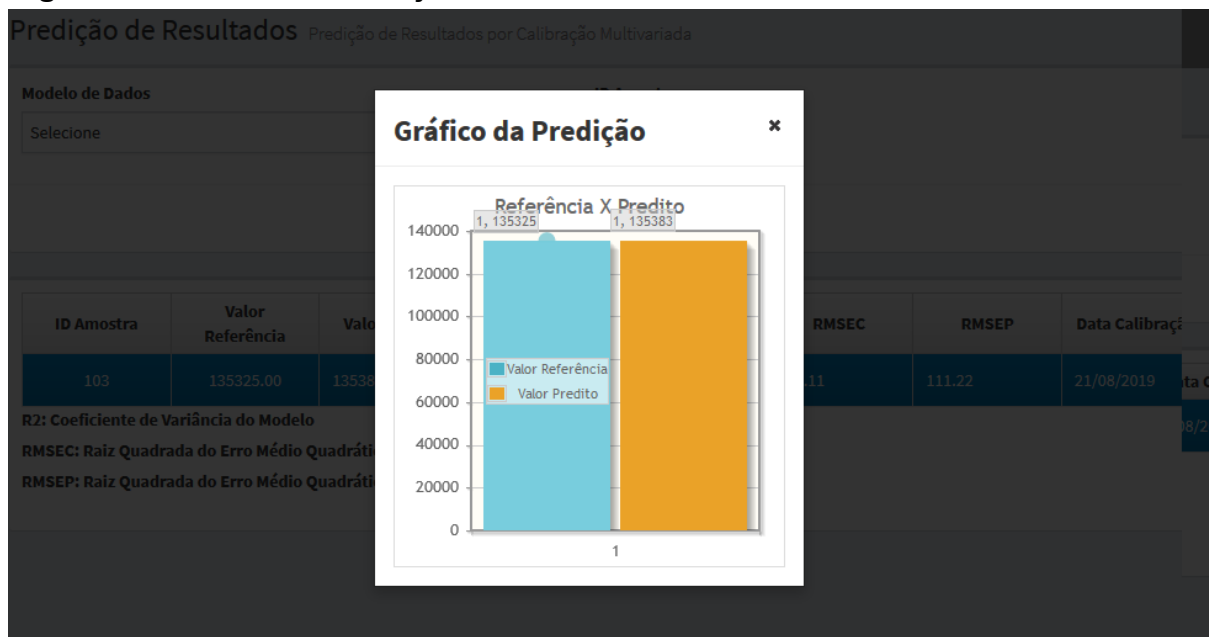
ID Amostra	Valor Referência	Valor Predito	R2 Cal	R2 Val	RMSEC	RMSEP	Data Calibração	Gráfico
103	135325.00	135383.10876722	0.88	0.89	149.11	111.22	21/08/2019	

R2: Coeficiente de Variância do Modelo  
 RMSEC: Raiz Quadrada do Erro Médio Quadrático da Calibração  
 RMSEP: Raiz Quadrada do Erro Médio Quadrático de Previsão

Copyright (C) 2019, Steven Kariston Louback de Carvalho

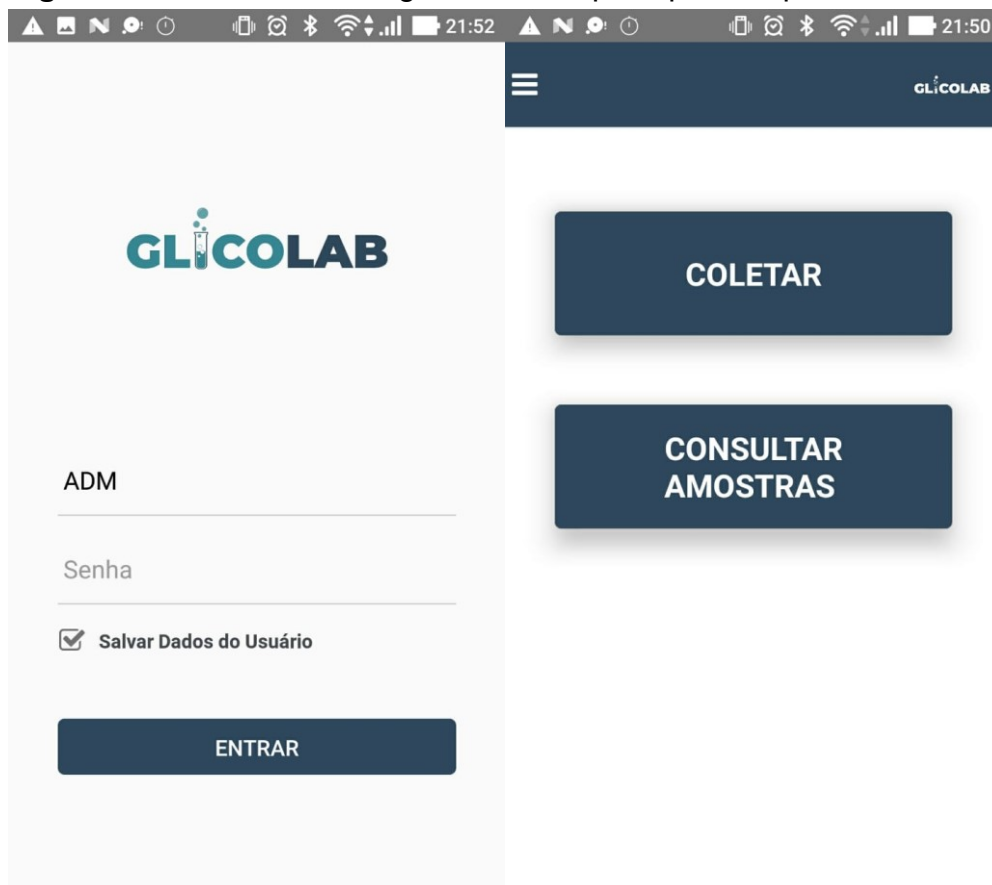
Fonte: Autoria Própria (2019).

Para a predição das amostras não conhecidas, foi desenvolvida uma interface em que o usuário seleciona qual o modelo de calibração multivariada deseja utilizar e informa o identificador da amostra a ser predita. Após informar os dados obrigatórios, o usuário solicita a predição por meio do botão buscar. Nesse momento o software faz uma requisição para o serviço *web* responsável pela predição que responde com o valor predito e algumas informações importantes como:  $R^2$  do modelo que é coeficiente de variância do modelo, RMSEC e RMSEP que representam o erro médio quadrático dos conjuntos de calibração e validação respectivamente.

**Figura 22 - Gráfico de Predição**

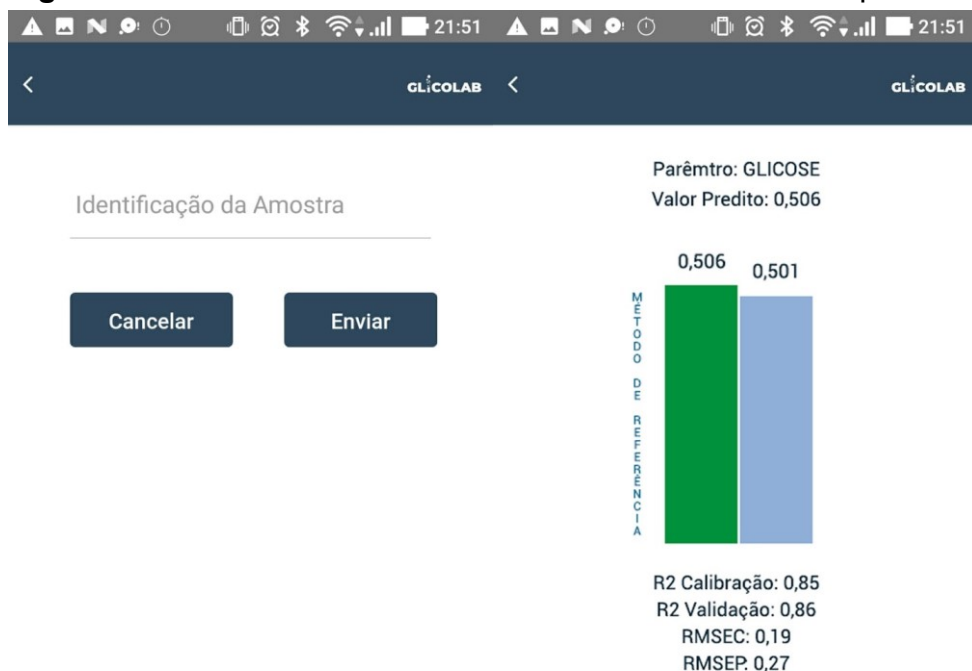
Fonte: Autoria Própria (2019).

Após fazer a predição, o usuário pode acessar o gráfico para visualizar qual o valor predito pelo modelo de calibração multivariada e o valor resultante do método de referência.

**Figura 23** - Interfaces de login e o menu principal do aplicativo GlicoLab

**Fonte:** Autoria Própria (2019).

O aplicativo não será disponibilizado para os pacientes que realizam exames sanguíneos nos laboratórios, pois apesar de ser um artefato de simples utilização, não poderá ser acessado ou operado por usuários que não receberam uma capacitação prévia, devido a necessidade que o método proposto de análise possui alguns padrões que devem ser seguidos como o preparo amostra coletada do paciente, para que todas as amostras coletadas pelo NIR estejam em condições iguais e ideais para o processo. Após feito este processo de preparo da amostra, o usuário poderá fazer a coleta do espectro por meio do aplicativo, onde o aplicativo irá coletar as informações do espectro diretamente do NIR.

**Figura 24 - Interface de coleta e resultados da Análise no Aplicativo**

**Fonte:** Autoria Própria (2019).

O principal propósito do desenvolvimento do aplicativo para fazer as predições é utilizar as vantagens que a plataforma móvel possui como a mobilidade, interatividade e simplicidade. Portanto, devido à simplicidade operacional do aplicativo é esperado que haja uma boa aceitação por parte dos usuários finais. Na rotina de predição, por exemplo, o usuário deverá informar a identificação do proprietário da amostra e acionar o botão coletar, conforme a representação na Figura 24.

Após enviar a amostra recém coletada para a API, o aplicativo deve ficar na espera da resposta e, assim que recebê-la por meio da interface de resultados da análise, serão apresentados os resultados da solicitação.

#### 4.4 Validação do Modelo de Calibração Multivariada

O modelo de calibração multivariada utilizado para aferir a eficiência e consistência das rotinas implementadas contemplou um total de 165 amostras. Para otimizar o modelo foi necessário identificar e eliminar as amostras anômalas (outliers) – que eram aquelas com comportamento muito diferente das demais amostras do conjunto de dados. A separação do conjunto de calibração e de validação foi feita a partir do algoritmo de Kennard-Stone e resultou em 107 amostras para calibração e 58 amostras para validação. O modelo construído empregou um total de 12 variáveis latentes. A partir desse modelo, testou-se a capacidade de previsão do conjunto de amostras de validação como mostra o Quadro 2.

**Quadro 2** - Figuras de mérito do modelo de calibração multivariada

Figuras de Mérito		Modelo PLS
Exatidão	RMSEC	149,1
	RMSEP	111,2
Coeficiente de Correlação - Calibração		0,88
Coeficiente de Correlação - Validação		0,88

Fonte: Autoria Própria (2019).

#### 4.5 Processos Melhorados

Considerando o objetivo de automatização de um processo de análise laboratorial que atualmente é trabalhoso e relativamente demorado, é de suma importância que este processo seja compreendido em sua totalidade e os principais conceitos respeitados, de modo que a precisão do resultado final gerado pelo sistema seja idealmente igual àquela observada processo atual de análise.

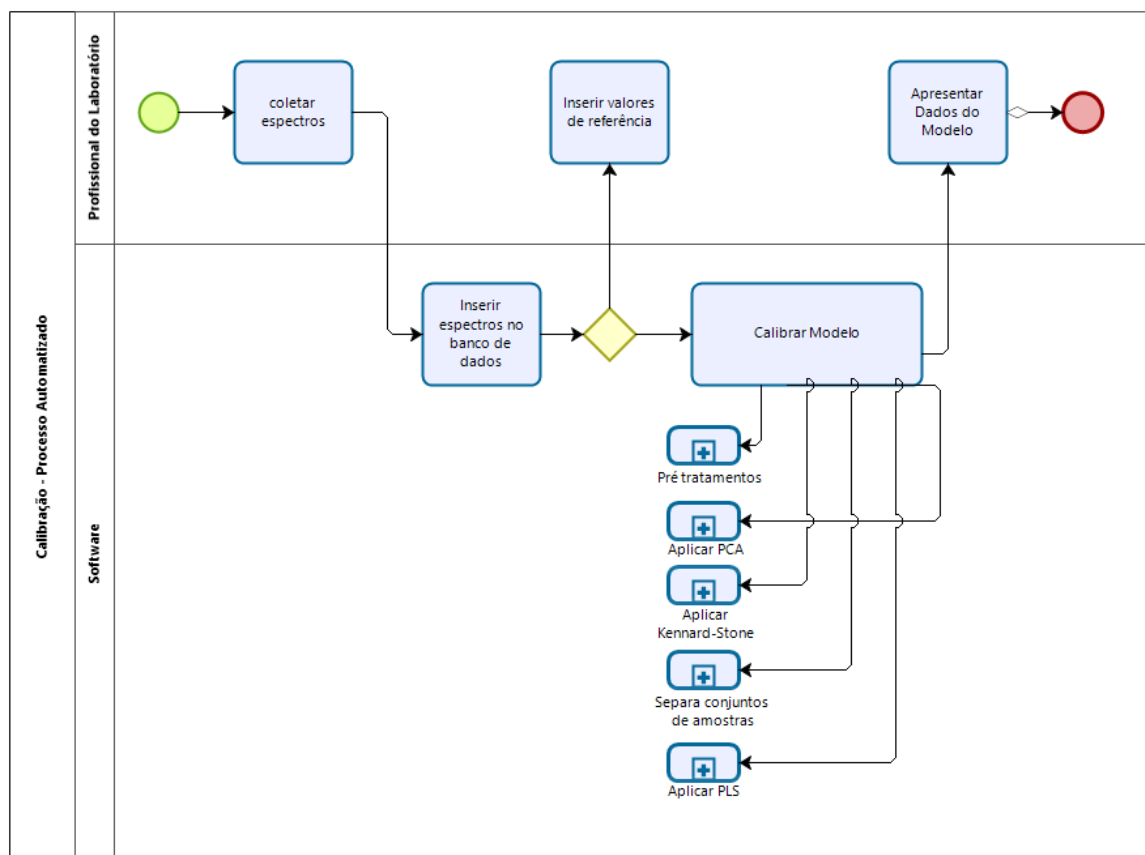
A fim de auxiliar no desenvolvimento e compreensão dos processos envolvidos, a seguir serão apresentados os diagramas concebidos para

representação dos processos pertinentes a confecção de um modelo preditivo, a calibração deste modelo e, finalmente, a predição de uma dada amostra. Com o objetivo de comparação são apresentados os processos manuais e automatizados por meio do software. Estes diagramas sustentam as conclusões sobre os benefícios providos com o desenvolvimento deste produto e dão indícios sobre o que pode ser futuramente aperfeiçoado.

#### 4.5.1 Processo de criação e calibração do modelo

O principal objetivo de melhoria empregada neste processo foi automatizar as etapas sequenciais utilizadas no modelo de calibração multivariada para fazer a predição de diabetes mellitus. O processo de calibração do modelo foi automatizado pela implementação dos métodos quimiométricos. Com essa automatização, o processo de calibração com 165 amostras foi realizado no espaço de tempo de apenas 16 segundos.

**Figura 25** - Processo de construção do modelo de calibração automatizado



**Fonte:** Autoria Própria (2019).

Os principais fatores que influenciam diretamente na redução do tempo gasto para fazer a construção de um modelo de calibração multivariada são as chamadas dos métodos, as quais demandam o repasse de diversos parâmetros como o conjunto de dados a ser processado, a automatização da coleta do espectro e os tratamentos nas amostras.

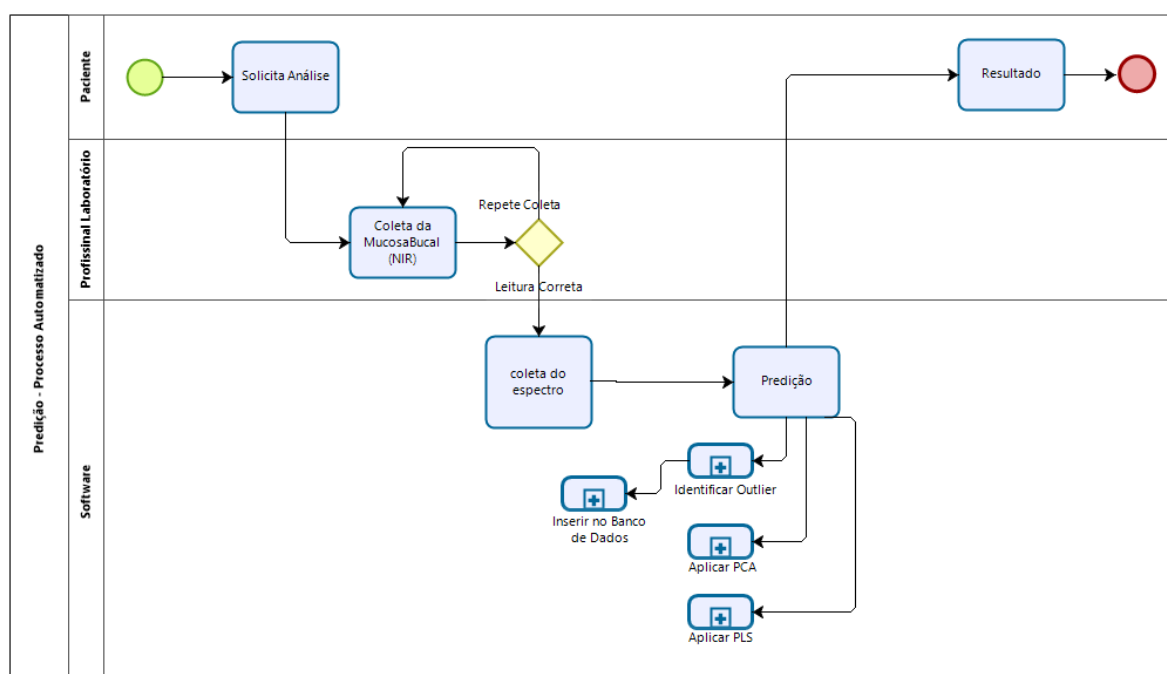
Com a implementação deste software estabelece-se padrões que podem ser personalizados de acordo com a necessidade do modelo de calibração multivariada, em que o responsável designa quais tratamentos e métodos quimiométricos o conjunto de amostras deve ser submetido.



#### 4.5.2 Processo de Predição de Amostra

O principal benefício ao processo de predição quando realizado por intermédio do software proposto está em possibilitar que qualquer profissional capacitado a operar o software possa fazer as predições sem a necessidade da intervenção do analista de laboratório, uma vez que todas as etapas complexas foram sistematizadas.

**Figura 24** - Processo de predição de amostras automatizado



**Fonte:** Autoria Própria (2019).

O processo de predição de uma amostra, antes inviável devido ao tempo necessário para cada etapa, pode ser realizado em tempo significativamente reduzido (os testes demandaram 5 segundos), o que permite uma melhoria significativa na frequência de obtenção dos resultados.

## 5 CONCLUSÃO

O desafio deste trabalho foi compreender os conceitos dos assuntos principais abordados nesta pesquisa e construir as integrações entre o equipamento coletor, métodos quimiométricos, dados previamente coletados, artefatos envolvidos da construção de software como rotinas, bibliotecas, banco de dados e linguagens de programação, etc. Conseqüentemente, após a compreensão dos conceitos supracitados e com a viabilização técnica, sem a utilização de ferramentas proprietárias, o software proposto foi construído, atendendo os requisitos apresentados pelos usuários.

Logo, por meio da integração entre áreas interdisciplinares do conhecimento foi possível oferecer uma alternativa viável para medida de glicemia em humanos de forma não invasiva e com aprimoramento da velocidade de análise empregando, para isso, fundamentos de espectroscopia NIR e quimiometria. O conhecimento em tecnologia da informação proporcionou o desenvolvimento de um software capaz de integrar resultados obtidos por equipamento para espectroscopia na região do infravermelho próximo com resultados obtidos por medidas atualmente utilizadas (via punção venosa), mostrando um potencial encorajador para utilização do formato proposto como uma forma de inovação tecnológica que pode beneficiar tanto analistas quanto pacientes e ao meio ambiente.

De modo geral, podemos afirmar que este trabalho resultou em um Produto Viável Mínimo (MVP) do software proposto, uma vez que, quando submetido a validações com modelos de calibração multivariada, apresentou um resultado satisfatório. A automatização de etapas como a de calibração e de validação (por meio do software) foi capaz de reduzir o tempo de análise para a ordem de segundos.

A redução do tempo foi viabilizada a partir da codificação de todos algoritmos necessários, invocação dos métodos quimiométricos necessários, serviços *web* e demais rotinas, todas unificadas em uma solução, sem a necessidade de intervenções humanas ou software de terceiros para a execução de todas as etapas. Portanto, pode se concluir que o objetivo principal de construir um software capaz de

operacionalizar e tornar escalável este novo método proposto de análise foi alcançado.

## 6 REFERÊNCIAS

AMORIM, Henrique.; **Manual de Métodos Analíticos para o Controle da Produção de Álcool e Açúcar**. 2ª Ed. Piracicaba: Editora Fermentec/Fealq/Esalq-USP, 1996. 230p.

BARROS NETO, Benicio; SCARMINIO, Ieda; BRUNS, Roy. 25 anos de Quimiometria no Brasil. **Química Nova**, v. 29: 6, p. 1401-1406, 2006.

BECK, Kant. **Extreme Programming Explained: Embrace Change**. Addison-Wesley, San Francisco, 2001.

BOLYARD, Elizabeth; TABLAN, Ofelia; WILLIAMS, Walter; PEARSON, Michele. Guideline for infection control in healthcare personnel. **Infection Control & Hospital Epidemiology**, v. 19, n. 6, p. 407-63, 1998.

BOOCH, Grady; RUMBAUGH, James.; JACOBSON, Ivar. **UML Guia do Usuário**. Tradução de Fábio Freitas Silva e Cristina de Amorim Machado. 2. ed. Rio de Janeiro: Campus, 2005.

BORGES, Luiz. **Python para Desenvolvedores**, 2 Ed, Edição do Próprio Autor, Rio de Janeiro, 2010.

BRASIL. **Recomendações da Sociedade Brasileira de Patologia Clínica / Medicina Laboratorial para Coleta de Sangue Venoso**. Elaborado pelo Comitê de Coleta de Sangue da SBPC/ML e BD Diagnostics - Preanalytical Systems. 1ª.ed, 76 p., São Paulo, 2005.

BRASIL. **Recomendações da Sociedade Brasileira de Patologia Clínica/Medicina Laboratorial (SBPC/ML): coleta e preparo da amostra biológica**. Editora Manole, Barueri, SP, 2014.

BROAD, Neville; GRAHAM, Paul; HAILEY, Perry; HARDY, Allison; HOLLAND, Steve; HUGHES, Stephen, LEE, David; PREBBLE, Ken; SALTON, Neale; WARREN, Paul. **Guidelines for the Development and Validation of Near-infrared Spectroscopic Methods in the Pharmaceutical Industry**. In: Chalmers JM, Griffiths PR. Handbook of Vibrational Spectroscopy. 2006; 610-3590 p.

BROWN, Steven. Chemical systems under indirect observation: Latent properties and chemometrics. **Appl. Spectrosc., Baltimore**, v.49, n.12, p.14A-31A, 1995.

CHARTIER, Yves; PRÜSS, Annette; EMMANUEL, Jorge. Cataloguing-in-Publication Data Safe management of wastes from health-care activities. 2ª ed. **Who Library Handbook**, 2013.

COCKBURN, Alistair. **Agile Software Development**. Addison-Wesley, UK. 2002.

COSCIONE, Aline. **O Uso de Calibração Multivariada para a Determinação Espectrofotométrica Simultânea de Alumínio e Ferro: Aplicação na Análise de Plantas e Solos**. 2001. 129 f. Tese (Doutorado em Química) - UNICAMP, Campinas, 2001.

DATE, Christopher. **Introdução a Sistemas de Bancos de Dados**, tradução da 7a. Edição Americana, Rio de Janeiro. Campus, 2000.

ELMASRI, Ramez; NAVATHE, Shamkant. **Sistemas de banco de dados**. 6. ed. São Paulo: Pearson, 2011.

FACHIN, Samuel. **Técnicas de análise multivariável aplicadas ao desenvolvimento de analisadores virtuais**. 2005. Dissertação (Mestrado em Engenharia Química) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2005.

FERREIRA, M.M.C. **Multivariate QSAR**. *J. Braz. Chem. Soc.*, São Paulo, v.13, n.6, p.742-753, 2002.

FERREIRA, Márcia; ANTUNES, Alexandre; MELGO, Marisa; VOLPE, Pedro. **Quimiometria I: Calibração Multivariada, um Tutorial**. *Química Nova*, v. 22, n. 5, p. 724-731, 1999.

FERREIRA, Márcia. **Quimiometria: conceitos, métodos e aplicações**. Campinas, SP: Editora da Unicamp, p. 493, 2015.

FERREIRA NETO, Francisco. **Determinação do teor de diclofenaco sódico em comprimidos por espectroscopia no infravermelho próximo - NIR com calibração multivariada - PLS**. 2012. Dissertação (Mestrado em Química) - Universidade Federal do Rio Grande do Norte, Natal, 2012.

FORTES, Paula. **Calibração Multivariada e Cinética Diferencial em Sistemas de Análises em Fluxo com Detecção Espectrofotométrica**. 2006. Dissertação (Mestrado em Ciências), Centro de Energia Nuclear na Agricultura da Universidade de São Paulo, São Paulo, 2012..

HORSTMANN, Cay. **Conceitos de programação com o essencial de Java**. 3 ed. Porto Alegre: Editora Bookman, 2003. 779 p.

IBM. **Patterns: Implementing an SOA Using an Enterprise Service Bus.**

Disponível em: <http://www.redbooks.ibm.com/redbooks/pdfs/sg246346.pdf>. Acesso em: 20 out de 2018.

JACONI, Angélica. **O uso da espectroscopia no infravermelho próximo na quantificação de carbono em solos sob o cultivo de cana-de-açúcar.** 2011. Dissertação (Mestrado em Ciências) - Universidade de São Paulo, Centro de Energia Nuclear na Agricultura, Piracicaba, 2011.

KENNARD, Wiggins. **Computer aided desing of experiments**, Technometrics. 1969, 11, 1, 137-148.

LEITE, Mário. **Técnicas de programação: Uma abordagem moderna.** 1 ed. Rio de Janeiro: Editora Brasport, 2006. 399 p.

LIMA, Adilson. **UML 2.0 - Do Requisito à Solução.** 1. ed. São Paulo: Érica, 2005.

MARÇO, Paulo. **Estudo da influência da radiação e pH no comportamento cinético de antocianinas de plantas do gênero hibiscos por métodos quimiométricos.** Tese(Doutorado em Química) – Universidade Estadual de Campinas, Instituto de Química, Campinas-SP, 2009.

MARTENS, Harald; NAES, T. **Multivariate Calibration.** 2 ed. Biddles Ltd, Great Britain, p. 419, 1994.

MISRA, Anoop; KHURANA, Lokesh. Obesity and the metabolic syndrome in developing countries. **Journal of Endocrinology and Metabolism**, v. 93, n. 11, p. 9-30, 2008.

MOITA NETO, José; MOITA, Graziella. **Uma introdução à análise exploratória de dados multivariados.** Departamento de Química - Universidade Federal do Piauí, 1997.

NERUR, Sridhar; MAHAPATRA, RadhaKanta; MANGALARAJ, George. **Challenges of migrating to agile methodologies.** Communications of the ACM: p. 72-78, 2005.

OASIS. In: Advancing open Standards for the information society. Disponível em <https://www.oasis-open.org/committees>. Acesso em: 20 de out de 2018.

ODLIS. In: Online Dictionary for Library and Information Science. Disponível em [http://www.abc-clio.com/ODLIS/odlis\\_I.aspx](http://www.abc-clio.com/ODLIS/odlis_I.aspx). Acesso em: 20 out de 2018.

PASQUINI Célio. Review Near Infrared Spectroscopy: Fundamentals , Practical Aspects and Analytical Applications. Journal of the Brazilian Chemical Society. 2003;14:198-219.

PRESSMAN, Roger. **Engenharia de Software**. 7. ed. Porto Alegre: Pearson Makron Books, 2011.

RIBEIRO, Fabiana. **Aplicação de Métodos de Análise Multivariada no Estudo de Hidrocarbonetos Policíclicos Aromáticos**. 2001. 174f. Dissertação (Mestrado em Química). UNICAMP, Campinas, 2001.

ROBERTS, J.J.; COZZOLINO, Daniel. An Overview on the Application of Chemometrics in Food Science and Technology — An Approach to Quantitative Data Analysis. Food Analytical Methods. 2016;9:3258–3267.

ROUSSEL Sylvie; PREYS Sébastien; CHAUCHARD, Fabien; LALLEMAND Jordane. Multivariate Data Analysis (Chemometrics). Process Analytical Technology for the Food Industry. In: O'Donnell C, Cullen PJ, Fagan C. Process Analytical Technology for the Food Industry. New York; Springer: 2014:7-17 p.

SABBAGH, Rafael. **Scrum: gestão ágil para projetos de sucesso. Casa do Código**, 2014.

SILVERSTEIN, Robert; WEBSTER, Francis; KIEMLE, David. **Espectrometria no infravermelho. In: Identificação espectrométrica de compostos orgânicos**. 7. ed. Cap. 2, p. 70-104. Rio de Janeiro: LTC, 2010.

SCARMINIO, Ieda; ISHIKAWA, Dilson; BARRETO, Wagner; PACZKOWSKI, Édio; ARRUDA, Iara. Calibração Multivariada para Sistemas Com Bandas Sobrepostas Através da Análise de Fatores do Tipo Q. **Química Nova**, v. 21, n. 5, p.590-596, 1998.

SMITH, Gordon. Principal component analysis: an introduction. **Analytical Proceedings**. v. 28, n. 5, p. 150-151, 1991.

SOMMERVILLE, Ian. **Engenharia de Software**. Tradução: Selma Shin Shimuzu Melnikoff, Reginaldo Arakaki, Edilson de Andrade Barbosa. São Paulo: Pearson Addison Wesley, 2008.

Sun DW. **Modern Techniques for Food Authentication**. 1th ed. Ireland: Elsevier Inc.; 2008:682 p.

TORRES, Maricy; ANDRADE, Denise; SANTOS, Claudia. Punção venosa periférica: avaliação de desempenho dos profissionais de enfermagem. **Revista Latino-americana de Enfermagem**, v. 13, n. 3, p. 299-304, 2005.

VALDERRAMA, Patrícia. **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, Patrícia. **Calibração multivariada de primeira e segunda ordem e figuras de mérito na quantificação de enantiômeros por espectroscopia**. Tese (Doutorado em Química Analítica) Universidade Estadual de Campinas, Instituto de Química. Campinas, SP, 2009.

VALDERRAMA, Leonardo; GONÇALVES, Rhayanna; MARÇO, Paulo; ; VALDERRAMA, Patricia. Espectroscopia Uv-Vis e Método Quimiométrico na Avaliação de Adultrações e Fraudes em Azeite de Oliva Extra Virgem. **Bras. Pesq. Alim.**, v. 5, n. 2, p. 32-40, 2014.

W3C. In: World Wide Consortium. Disponível em <http://www.w3.org/>. Acesso em: 20 de out de 2018.

Workman JR JJ. Interpretive Spectroscopy for Near Infrared. *Applied Spectroscopy Reviews*. 2006; 31:251-320.

Workman Jr J, Weyer L. History of Near-Infrared Applications. In: *Practical Guide and Spectral Atlas for Interpretive Near-Infrared Spectroscopy*. 2th ed. New York: CRC Press; 2012:101 p.