

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS

MURILO DE PAULA ARAUJO

VENDA MAIS – SISTEMA DE APRIMORAMENTO DE VENDAS EM LOJAS

CAMPINAS

2022

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS
CENTRO DE CIÊNCIAS EXATAS, AMBIENTAIS E DE TECNOLOGIAS
FACULDADE DE ENGENHARIA DE COMPUTAÇÃO**

MURILO DE PAULA ARAUJO

VENDA MAIS – SISTEMA DE APRIMORAMENTO DE VENDAS EM LOJAS

Monografia de Trabalho de Conclusão de Curso apresentada como exigência da disciplina Projeto Final II, ministrada no Curso de Engenharia de Computação, do Centro de Ciências Exatas, Ambientais e de Tecnologias da Pontifícia Universidade Católica de Campinas.

Orientador: Dr. Carlos Miguel Tobar Toledo

Coorientador: Me. Leandro Alonso Xastre

CAMPINAS

2022

RESUMO

A monografia relata um Trabalho de Conclusão de Curso (TCC), cujo foco foi possibilitar para uma loja varejista de roupas, um atendimento personalizado com os seus clientes, por meio de um sistema de *software*, nomeado Venda Mais. O objetivo do estudo é propor uma solução que potencialize o aumento nas vendas dessa loja. Para alcançar esse objetivo, o aluno escolheu uma rede neural de reconhecimento facial para analisar os rostos dos clientes que entram na loja, um algoritmo de árvore de decisão para sugerir futuras compras e um algoritmo de *matchmaking* para indicar o melhor vendedor para atender a um cliente. Para o desenvolvimento da proposta de solução, foram utilizadas as linguagens de programação Python e Java, os *frameworks* Spring Boot e React Native e dois bancos de dados MySQL. A metodologia de desenvolvimento escolhida para gerenciar a aplicação foi a Scrum Solo. A avaliação foi realizada remotamente, junto a um avaliador especializado no setor de vestuário. O estudo apresentou as principais funcionalidades do sistema, focando em atingir o seu objetivo e obteve um resultado promissor, pois o avaliador afirmou que o Venda Mais, da maneira que foi implementado, é uma solução que potencializa o aumento das vendas na loja.

Palavras chave: Reconhecimento facial; *matchmaking*; árvore de decisão.

ABSTRACT

This monograph reports an Undergraduate Project that focused on enabling a personalized service for the customers of a retail clothing store through a software system named Venda Mais. The project aimed to propose a solution to increase sales at the store. For this purpose, the student chose a facial recognition neural network to analyze the faces of customers entering the store, a decision tree algorithm to suggest future purchases, and a matchmaking algorithm to indicate the best salesperson to serve each customer. Python and Java programming languages, Spring Boot and React Native frameworks and two MySQL databases were used to develop the proposed solution. Scrum Solo was the development methodology chosen to manage the project and its assessment was carried out remotely, along with an evaluator specialized in the clothing sector. The study presented the main functionalities of the system, focusing on achieving the project objectives, and obtained a promising result considering the evaluator statement that Venda Mais, as implemented, is a solution that enhances the store sales.

Keywords: *Facial recognition; Matchmaking; Decision tree.*

LISTA DE FIGURAS

Figura 1. Diagrama de arquitetura focado no aplicativo do gerente	11
Figura 2. Diagrama de arquitetura focado no aplicativo dos vendedores	11
Figura 3. Scrum solo: processo de <i>software</i> para desenvolvimento pessoal	13
Figura 4. Montagem da câmera de segurança utilizada no <i>software</i> Venda Mais	17
Figura 5. Cadastramento de gerente	23
Figura 6. Menu da aplicação	24
Figura 7. Cadastramento de vendedor	25
Figura 8. Cadastramento de cliente	26
Figura 9. Busca de um vendedor	27
Figura 10. Remoção de vendedor	27
Figura 11. Autenticação de vendedor	28
Figura 12. Entrada de cliente na loja Campinense	29
Figura 13. Cliente reconhecido para o vendedor atender	31
Figura 14. Cliente não cadastrado para o vendedor atender	32
Figura 15. Compra de presente para pessoa já presenteada	33
Figura 16. Compra de presente para uma nova pessoa	34

SUMÁRIO

1	INTRODUÇÃO	6
1.1	Caracterização de problemas e objetivo	6
1.2	Trabalhos relacionados	7
1.3	Proposta de artefato	9
1.4	Organização da monografia	12
2	PLANEJAMENTO E PROJETO	13
2.1	Metodologia do projeto	13
2.2	Bancos de dados relacionais	14
2.3	Tecnologias e ferramentas	15
2.3.1	<i>Hardware</i>	15
2.3.2	<i>Software</i>	15
2.4	Sistemas ou integrações de <i>hardware</i> ou <i>software</i>	16
2.5	Controle de versões e os <i>backups</i>.....	17
3	VENDA MAIS	18
3.1	Algoritmos inteligentes	18
3.1.1	Reconhecimento facial	18
3.1.2	<i>Matchmaking</i>	20
3.1.3	Árvore de decisão	21
3.2	Aplicativo do gerente	23
3.2.1	Cadastramento de gerente	23
3.2.2	Menu principal	24
3.2.3	Cadastramento de vendedor e cliente.....	25
3.2.4	Busca de vendedor e cliente.....	26
3.2.5	Remoção de vendedor.....	27
3.3	Aplicativo dos vendedores	28
3.3.1	Autenticação de vendedor	28
3.3.2	Entrada de cliente.....	29
3.3.3	Realização de atendimento	30
3.3.4	Compra de presentes	32
4	AVALIAÇÃO E VALIDAÇÃO	35
4.1	Avaliação	35
4.2	Validação	37
5	CONCLUSÃO	38
5.1	Aspectos de inovação e aprimoramento	38
5.2	Qualidade e grau de complexidade do <i>software</i>.....	39
5.2.1	Qualidade	39
5.2.2	Grau de complexidade.....	41
5.3	Dificuldades enfrentadas	42
5.4	Futuras melhorias.....	43
	REFERÊNCIAS.....	45
	APÊNDICES	46

1 INTRODUÇÃO

A loja Campinense é uma loja de roupas localizada em Campinas, que atualmente funciona com cinco vendedores, um gerente e um operador de caixa. O atendimento da loja é realizado de maneira tradicional em relação às demais varejistas no setor de vestuário. De acordo com o gerente da loja, quando um cliente entra no estabelecimento, um vendedor o aborda, perguntando se pode ajudar e se o cliente gostaria de experimentar alguma roupa ou comprar um presente.

Esse vendedor pode não ser sempre o mesmo, pois, conforme o gerente explicou, o atendimento atualmente funciona como se fosse uma “roleta”. Como são cinco vendedores, define-se qual será o primeiro e este atende o primeiro cliente e assim por diante até cada vendedor realizar um atendimento. E então, o primeiro da roleta atende novamente e assim por diante, iniciando outro “ciclo” de atendimento.

Durante as abordagens, algumas vezes, acontece de um vendedor indicar um outro colega para realizar o atendimento de determinado cliente por conta de seu perfil, independente da colocação do vendedor na roleta de atendimento. Pois o vendedor percebe que tem um outro vendedor que pode atender melhor o consumidor e aumentar a sua satisfação de atendimento, por conta de suas características e preferências tanto do cliente como do vendedor.

Como o gerente da Campinense relatou, muitos clientes procuram por determinados vendedores por questões de afinidade e para agilizar a futura compra, pois o funcionário já conhece o perfil do cliente. No entanto, isso pode não ser possível, se o vendedor desejado não for mais funcionário da loja ou se não estiver disponível.

Em lojas no setor de vestuário, como a Campinense, acontece algo muito comum conhecido como *turnover*, que significa rotatividade de pessoal, pois se trata de um mercado muito grande e os funcionários recebem novas oportunidades ou por outros motivos, deixam o seu emprego e passam atuar em outras lojas.

1.1 Caracterização de problemas e objetivo

A loja de roupas Campinense costuma possuir dez mil itens de roupas em seu estoque. Entretanto, nunca vendeu mais do que 50% (cinquenta por cento) dessa quantidade em um mês.

Dia 30 do mês, o gerente realiza solicitações de novas mercadorias, preocupando-se com as quantidades de roupas não vendidas, motivo pelo qual necessita realizar liquidações e promoções, com o objetivo de vender os produtos mais antigos, pois, assim, aumenta o seu fluxo de caixa e consegue mais espaço de armazenamento para novos itens.

O gerente da loja compra algumas mercadorias selecionadas, com a preocupação acerca do espaço disponível na loja, mas a loja Campinense poderia vender muito mais roupas e alcançar uma margem de venda superior aos níveis atuais.

O objetivo do Trabalho de Conclusão de Curso (TCC), descrito nesta monografia, é propor uma solução que potencialize um aumento de vendas na loja Campinense.

1.2 Trabalhos relacionados

No mundo, existem atualmente diversas lojas no setor de vestuários. Em geral essas lojas procuram constantemente inovar e melhorar os seus atendimentos com os clientes, com o intuito de apresentar um diferencial competitivo. E, conseqüentemente, destacar-se no mercado diante das concorrentes, aumentar as vendas de seus produtos e obter maior lucro.

Dessa forma, já existem nessa área diversos *softwares* semelhantes ao Venda Mais. Contudo, esse aplicativo se diferencia por realizar sugestões de compras e notificar um melhor vendedor para realizar o atendimento de um cliente.

Foram pesquisadas documentações de sistemas semelhantes ao Venda Mais para uma melhor análise comparativa. Assim, foram realizadas comparações com três publicações:

- Recommender Vision: um sistema de recomendação para o comércio baseado em Visão Computacional (SEMINOTTI; RIEDER, 2020) – aborda um sistema que realiza sugestões de compras para os clientes, em relação aos produtos nas prateleiras que eles observam.
- Supermarket Automation with Chatbot and Face Recognition Using IoT and AI (ANGELINE *et al.*, 2018) – propõe um *software* totalmente eficiente, com o

intuito de automatizar as lojas, e conseqüentemente, remover as filas nos caixas.

- Enhancing retailer marketing with an facial recognition integrated (WU; ZENG; SHIH, 2015) – apresenta um sistema que recomenda um perfil de usuário já definido, quando um cliente irá realizar o cadastro na loja, em relação as características de seu rosto.

Uma análise comparativa entre o Venda Mais e os sistemas existentes pode ser vista no Quadro 1.

Quadro 1. Comparativo entre o Venda Mais e sistemas semelhantes.

Características	SEMINOTTI; RIEDER, 2020	ANGELINE et al., 2018	WU; ZENG; SHIH, 2015	Venda Mais
Suporta reconhecimento facial	Sim	Sim	Sim	Sim
Algoritmo LBPH para realizar o reconhecimento facial	Não	Não	Não	Sim
Algoritmo de árvore de decisão para indicar categoria de roupa	Não	Não	Não	Sim
<i>Chatbot</i>	Não	Sim	Não	Não
Indicação de produto em relação ao que o cliente está observando	Sim	Não	Não	Não
Utilização de câmera de monitoramento	Sim	Sim	Sim	Sim
Focado em lojas varejistas	Sim	Sim	Sim	Sim
Deteção de produto retirado da prateleira	Não	Sim	Não	Não
Análise do estoque da loja se alguma mercadoria foi removida ou adicionada	Sim	Sim	Não	Sim
Realização de interações com o banco de dados da loja	Sim	Sim	Não	Sim
Notificação dos vendedores com dados dos clientes	Não	Não	Não	Sim
Cadastramento dos clientes no sistema	Não	Não	Sim	Sim
Funcionamento autônomo sem a necessidade de um funcionário	Não	Sim	Sim	Sim
Renderização das imagens antes de realizar a classificação do algoritmo de reconhecimento facial	Sim	Sim	Sim	Sim
Utilização de <i>QR Code</i> para pagamento das compras e identificação do cliente	Não	Sim	Não	Não

Fonte: Elaborado pelo autor (2022).

1.3 Proposta de artefato

Para atingir o objetivo proposto, pensou-se na elaboração de um *software*, com o intuito de aprimorar o comércio de lojas, contanto com o desenvolvimento de um sistema Spring Boot (Java) para gerenciamento e organização de todo o *backend* do *software* Venda Mais.

Que realiza as comunicações com dois bancos de dados relacionais, um do Venda Mais e outro da loja, no qual são gerenciados e hospedados no PostgreSQL localmente. A Seção 2.2, desta monografia, descreve detalhadamente os dois bancos de dados.

Também foram desenvolvidos dois aplicativos em React Native, considerado o *frontend* do *software*, que disponibilizam as interfaces para interações dos funcionários (gerente e os vendedores da loja) e comunicam com o *backend*.

Um aplicativo foi implementado para *tablet*, no qual é exclusivo para o gerente da loja, que permite realizar o cadastro de clientes, cadastro e exclusão de funcionários e busca para análise de algum dado ou sua edição. A Seção 3.2 descreve detalhadamente as funcionalidades desse aplicativo.

O segundo aplicativo é exclusivo para os vendedores, e pode ser baixado em mais de um celular; eles são notificados para atender os clientes que entram na loja. O aplicativo facilita na realização de um atendimento personalizado, que utiliza um algoritmo de árvore de decisão descrito no Subitem 3.1.3, para o vendedor verificar sugestões de compras e de presentes. A Seção 3.3 apresenta com mais detalhes as funcionalidades desse aplicativo.

O Venda Mais foi integrado com uma câmera de segurança conectada via cabo no *notebook*, na Seção 2.4 pode ser verificado como foi realizada a sua montagem. Todas as vezes que uma pessoa passa pela câmera de segurança, é captada a imagem do seu rosto, e, então, o sistema utiliza uma rede neural de reconhecimento facial detalhada na Seção 3.1.1 para analisar se é um cliente cadastrado no *software* Venda Mais, cliente desconhecido ou vendedor da loja.

Se for um cliente cadastrado, o sistema utiliza um algoritmo de *matchmaking* detalhado na Seção 3.1.2, para definir qual o melhor vendedor disponível para atender o cliente, e notifica o vendedor definido com as informações e uma foto do cliente já armazenada no seu cadastro. Entretanto, se o cliente não estiver cadastrado, o

sistema notifica o vendedor disponível com o menor número de atendimentos, com uma foto do cliente que foi captada, enquanto ele passava na frente da câmera.

Caso não tenha nenhum vendedor disponível, é elaborada uma nova entidade de atendimento no banco de dados do Venda Mais, no qual é inserido o identificador do cliente se for cadastrado, se caso não for, o atributo identificador do cliente na entidade recebe o valor zero. No entanto, assim que algum vendedor estiver disponível, é inserido o identificador desse vendedor na entidade de atendimento e ele é notificado em seu celular, para realizar o atendimento do cliente.

Em relação à proposta de artefato inicial, ocorreram algumas modificações durante o desenvolvimento do aplicativo. Inicialmente, não havia sido planejado implementar a opção de compras de presentes para clientes cadastrados, que foi descrito detalhadamente na Seção 3.3.4.

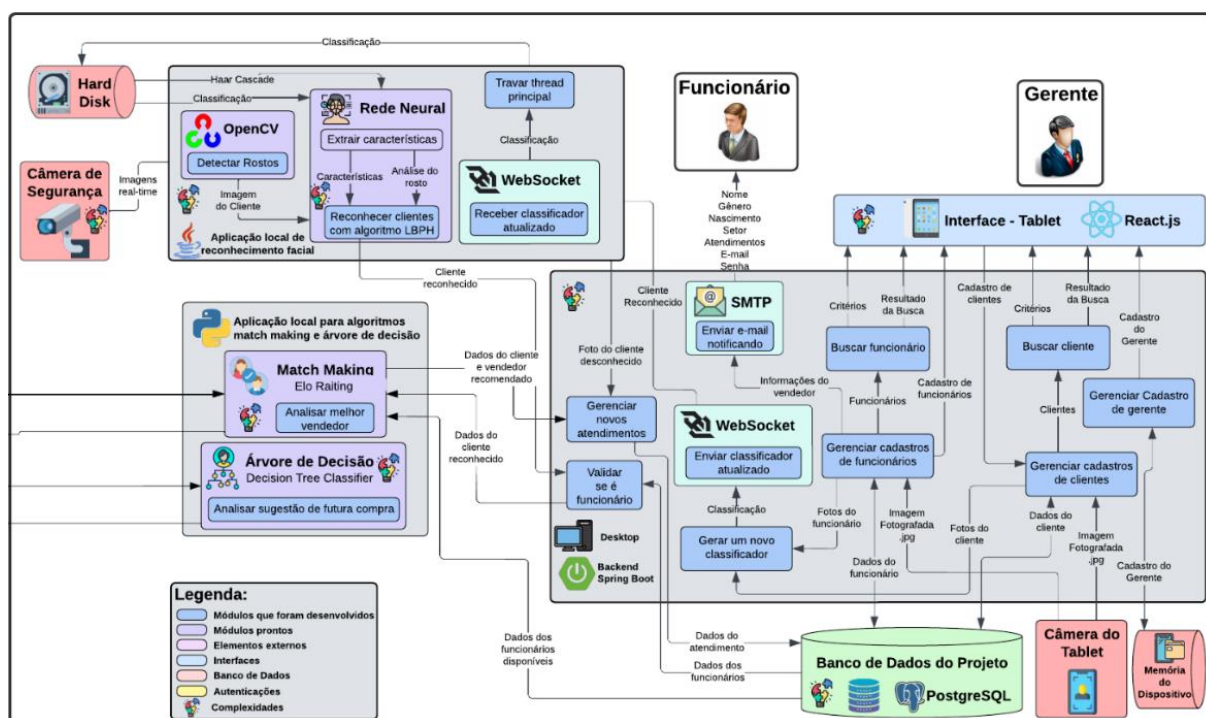
Também, no desenvolvimento do Venda Mais, foi solicitada uma implementação para notificar um vendedor durante o atendimento, entre um mês antes ou depois do aniversário do cliente. Dessa maneira, o vendedor pode desejar felicitações e propor um cupom promocional de aniversário da loja Campinense.

Uma outra solicitação, foi em relação à elaboração de um algoritmo de *matchmaking* totalmente focado em atender os requisitos do *software* Venda Mais, em vez, de utilizar um algoritmo pronto já existente. Com o intuito de recomendar um melhor vendedor para um determinado cliente, considerando apenas as características comuns entre eles, cadastradas no banco de dados relacional do Venda Mais.

Uma última alteração solicitada durante o desenvolvimento do Venda Mais, foram os envios de *e-mails* para os vendedores, no qual indica que um vendedor foi cadastrado no *software* e deseja as boas-vindas; pode ser visualizado um exemplo no Apêndice A. Caso ocorram alterações nos dados de um vendedor, o sistema também envia um *e-mail* para ele, exibindo os dados alterados, conforme apresentado no Apêndice B.

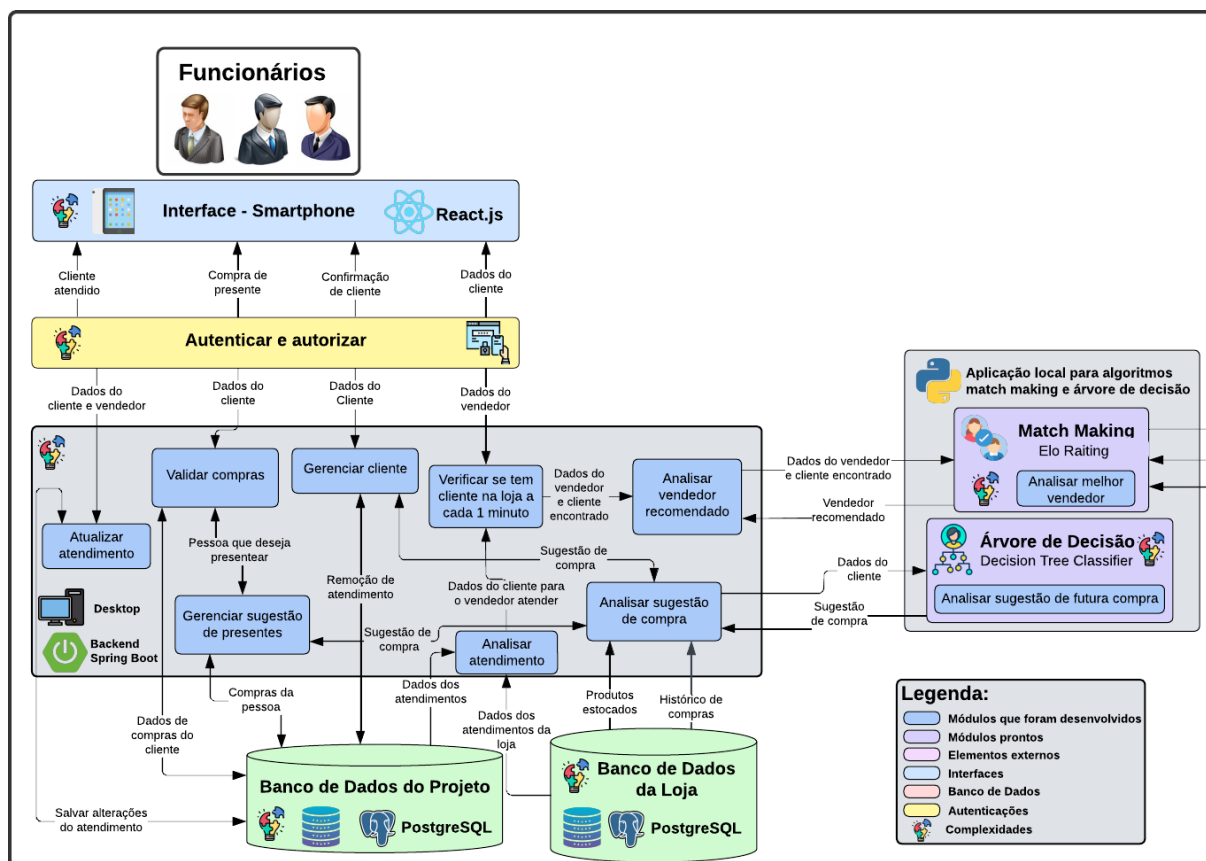
As Figuras 1 e 2 são partes do diagrama de arquitetura completo do Venda Mais, que é apresentado detalhadamente no Apêndice C. Essas figuras apresentam a mesma legenda e, por meio das cores, é verificar os módulos que foram desenvolvidos, os módulos prontos, elementos externos, interfaces, banco de dados e autenticação. E por intermédio dos ícones, pode-se verificar representações de funcionários e gerente, e também, os módulos que apresentam complexidade.

Figura 1. Diagrama de arquitetura focado no aplicativo do gerente.



Fonte: Elaborada pelo autor (2022).

Figura 2. Diagrama de arquitetura focado no aplicativo dos vendedores.



Fonte: Elaborada pelo autor (2022).

1.4 Organização da monografia

Esta monografia está organizada da seguinte maneira:

- O Capítulo 2 apresenta o desenvolvimento do *software* Venda Mais, que descreve o método de desenvolvimento escolhido, os dois bancos de dados do sistema, as tecnologias e as ferramentas utilizadas, os sistemas ou integrações de *hardware* ou *software*, e como foram realizados o controle de versões e os *backups*.
- O Capítulo 3 descreve detalhadamente o Venda Mais, os algoritmos inteligentes utilizados e os dois aplicativos desenvolvidos, um para o gerente e o outro dos vendedores.
- No Capítulo 4, são apresentados os resultados do artefato, como: a avaliação do sistema, sua execução, os seus resultados alcançados e como foi definido que o artefato atingiu o objetivo proposto no TCC, descrito nesta monografia.
- O Capítulo 5 descreve as conclusões do TCC. Nele são abordados os aspectos de inovação e aprimoramento, análise da qualidade e do grau de complexidade presentes no sistema desenvolvido, as dificuldades enfrentadas durante o desenvolvimento do TCC e como foram resolvidas e futuras melhorias para o Venda Mais.

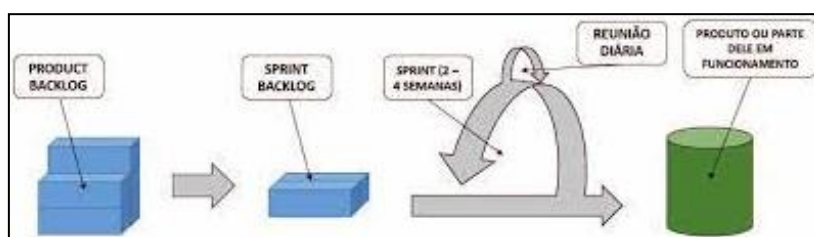
2 PLANEJAMENTO E PROJETO

Este capítulo apresenta o método de desenvolvimento escolhido para planejar e gerenciar o *software* Venda Mais, descrito nesta monografia; os dois bancos de dados relacionais do Venda Mais; as tecnologias e as ferramentas utilizadas; os sistemas ou integrações de *hardware* ou *software* e como foram realizados o controle de versões e os *backups* durante o desenvolvimento do *software*.

2.1 Metodologia do projeto

Como o *software* Venda Mais foi desenvolvido individualmente, foi escolhida a metodologia de desenvolvimento mais recomendada, que é a Scrum Solo (PAGOTTO *et al.*, 2016) para o seu desenvolvimento. A Figura 3 apresenta as etapas do processo dessa metodologia.

Figura 3. Scrum solo: processo de *software* para desenvolvimento pessoal.



Fonte: (PAGOTTO *et al.*, 2016, p. 36).

Inicialmente, houve uma extração de requisitos com o cliente do Venda Mais. Esses requisitos compuseram o *Product Backlog* inicial. Em seguida, realizou-se uma lista de histórias, a partir de uma análise de requisitos.

E depois, foram definidos os *Sprints* na ferramenta *on-line* Jira Software. Esses *Sprints* tiveram uma duração total de 128 dias, dividido em oito *Sprints* de 16 dias cada.

2.2 Bancos de dados relacionais

O *software* utiliza dois bancos de dados relacionais, um do Venda Mais e o outro da loja Campinense, o Apêndice D apresenta o modelo de entidade e relacionamento (MER) do banco de dados do Venda Mais e o Apêndice E, apresenta

o MER do banco da loja Campinense. Os dois bancos de dados são formados por cinco entidades e possuem uma entidade cliente em comum.

A entidade Clientes, contida nos dois bancos de dados, possui os seguintes atributos: identificação, nome, gênero, data de nascimento e a sugestão de compra. Por meio dos atributos nome e data de nascimento, é possível obter todas as informações de um cliente armazenadas nos dois bancos.

No banco de dados do Venda Mais, a entidade Clientes está relacionada com a tabela Imagens de Cliente, responsável por armazenar a chave primária do cliente e seis fotos de seu rosto em formato *blob*, que foram fotografadas no *tablet*.

Também possui uma tabela de Funcionários, com os seguintes atributos: identificação, nome, gênero, número de atendimento, disponibilidade, data de nascimento, *e-mail*, senha, setor que atua e faixa etária que prefere atender. Que está relacionada com uma entidade Imagens de Funcionário, responsável por armazenar seis imagens de cada vendedor no formato *blob* e a sua chave primária cadastrada na tabela Funcionários.

As tabelas Clientes e Funcionários se relacionam com uma entidade fraca, nomeada Cliente Vendedor, que é utilizada para armazenar os atendimentos. Essa entidade possui os seguintes atributos, a chave primária de um cliente e um vendedor, tempo de entrada e saída do cliente. O tempo de entrada é preenchido na criação da entidade atendimento e o atributo tempo de saída quando o vendedor finaliza o atendimento, por meio do aplicativo dos vendedores.

O banco de dados da loja também possui uma entidade para armazenar os atendimentos dos clientes que entram na loja e compram os seus produtos, sem ser atendido por um vendedor que utiliza o *software* Venda Mais. Essa entidade é preenchida apenas quando o cliente finaliza as suas compras no caixa da loja. Sendo necessário para o *software* analisar se um cliente que aguarda um vendedor ficar disponível no Venda Mais para atendê-lo, já deixou a loja ou não.

A tabela Clientes do banco de dados da loja está relacionada com a entidade Compras, que possui os seguintes atributos: chave primária do cliente, categoria, data, grau de parentesco e o nome da pessoa presenteada. Caso esses dois últimos atributos não forem preenchidos, o sistema entende que a pessoa não presenteou ninguém e a compra foi realizada para ela mesma.

A tabela Compra também está relacionada com a entidade fraca, nomeada Compra Produto, para manter o princípio de um banco de dados relacional e possuir

um relacionamento com a entidade Produtos, que contém os atributos: marca, categoria, descrição da roupa, preço, quantidade em estoque e o tamanho.

2.3 Tecnologias e ferramentas

Para a realização do trabalho descrito nesta monografia, foram utilizadas as seguintes tecnologias e ferramentas.

2.3.1 Hardware

- Câmera profissional de segurança analógica S.L.N 1.3MP modelo LS-506, para realizar o monitoramento da entrada de clientes na loja.
- Dispositivo móvel (*smartphone*) *ASUS ZenFone Max Shot*, para utilização do aplicativo dos vendedores.
- Dispositivo móvel (*smartphone*) *Samsung Galaxy Note 10*, para utilização do aplicativo dos vendedores.
- *Notebook* pessoal com processador Intel Core i5 3,70 Ghz, para execução do *software* Venda Mais.
- Placa De Captura *EasyCAPure* USB 2.0 adaptador de vídeo, para converter imagem analógica da câmera de segurança em digital para o *notebook*.
- *Tablet SM-T865*, para utilização do aplicativo do gerente.

2.3.2 Software

- Biblioteca JavaCV v1.3.3, para realizar a detecção de rostos.
- Decision Tree Classifier, para realizar as sugestões de futuras compras para o cliente.
- Figma 88.1.0, para realizar a prototipagem dos aplicativos que serão implementados nos *smartphone* e *tablet*.
- GitHub, Inc. [US], para *backup* com gerenciamento do controle de versões.
- Google Drive versão 46.0, para *backup* de documentos e arquivos;
- IntelliJ IDEA, como ambiente de desenvolvimento integrado escrito em Java para o desenvolvimento do *backend* do artefato. Além de acesso de requisições e serviços relacionados aos bancos de dados.

- Java 11, para elaborar uma aplicação local e integração da rede neural LBPH e a câmera de segurança.
- Javascript 1.7, como linguagem de programação.
- Jira *Software*, para o gerenciamento do Venda Mais.
- Lucidchart 2022, para a realização de diagramas.
- Elo Raiting, para indicar um melhor vendedor para o cliente em relação ao seu perfil.
- Microsoft Excel 2019 16.0.6742.2048, para confecção de planilhas.
- Microsoft Project 2019 16.0.6741.2048, para a realização do cronograma e distribuição de atividades do primeiro e segundo semestre.
- Microsoft Visual Studio Code 1.64, como um ambiente de desenvolvimento integrado da Microsoft para desenvolvimento do *frontend* do Venda Mais e das da aplicação local em Python.
- Python 3.9.10, para elaborar uma aplicação local e os algoritmos *matchmaking* e árvore de decisão.
- PostgreSQL v13.4, como gerenciador do banco de dados relacional.
- React Native v0.63.3, como *framework* para programação dos aplicativos.
- Rede Neural LBPH, para elaboração do reconhecimento facial.
- SMTP - *Simple Mail Transfer Protocol*, para envios de *e-mails*.
- Spring Boot v2.6.6, como *framework* para programação do *backend*.
- Spring Security, para realizar a autenticação do aplicativo dos vendedores.

2.4 Sistemas ou integrações de *hardware* ou *software*

Para o funcionamento do *software* Venda Mais, descrito nesta monografia, é necessário ser integrado com um banco de dados de uma loja que possui histórico de compra de seus clientes e os produtos em estoque, que nesse caso foi integrado com o da loja Campinense, como descrito na Seção 2.2.

Além dessa integração, o Venda Mais necessita ser integrado com um computador, que irá funcionar como um servidor local da aplicação, no qual intermediará as comunicações entre os módulos do *software* com requisições HTTP.

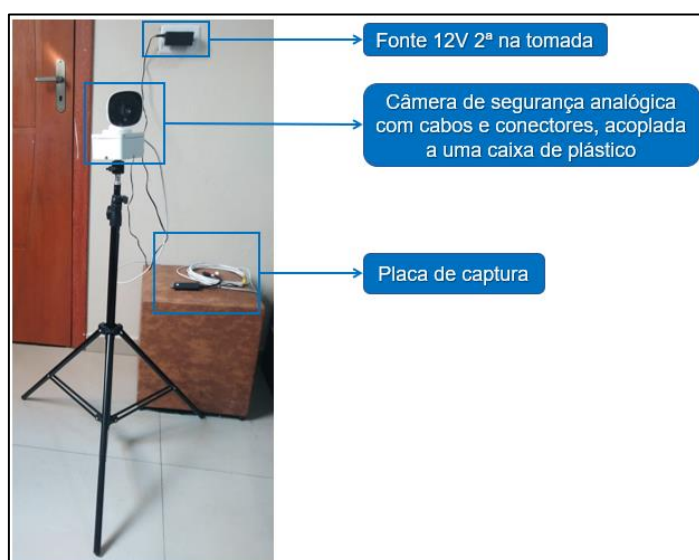
Necessita de um *tablet*, que possua uma câmera, para instalação e utilização do aplicativo do gerente da loja, apresentado na Seção 3.2. E pelo menos um celular

para a instalação e utilização do aplicativo elaborado para os vendedores, descrito na Seção 3.3.

Além disso, o sistema necessita ser integrado a uma câmera compatível com o computador que está sendo executado o Venda Mais. Nesse *software*, foi escolhida uma câmera de segurança analógica com cabos e conectores, acoplada a uma caixa de plástico, em um tripé, que precisa apenas ligar a fonte com especificações de “12V 2A” em uma tomada.

Para o computador conseguir captar as imagens em tempo real, por meio da sua entrada USB, foi necessária uma placa de captura, responsável pela conversão do sinal analógico em digital. Na Figura 4, apresenta-se detalhadamente a sua montagem e seus componentes.

Figura 4. Montagem da câmera de segurança utilizada no *software* Venda Mais.



Fonte: Elaborada pelo autor (2022).

2.5 Controle de versões e os *backups*

Para o controle de versões e os *backups*, foi utilizado o Google Drive para armazenar todos os documentos, arquivos, imagens e diagramas que foram elaborados no decorrer do estudo.

E para o controle de versionamento de todo o código do *software* Venda Mais, foi utilizado apenas um repositório na plataforma GitHub, que pode ser encontrado no Apêndice F. Foram criados diretórios e *branches* separadas, para cada módulo que constitui o *software*.

3 VENDA MAIS

Este capítulo apresenta detalhes do sistema Venda Mais, descrevendo os algoritmos inteligentes utilizados e as funcionalidades dos dois aplicativos desenvolvidos, um para o gerente e o outro dos vendedores. Diversas funcionalidades são acessíveis e podem ser interagidas pelos funcionários (gerente e vendedores) por meio das interfaces gráficas dos aplicativos.

3.1 Algoritmos inteligentes

Os principais diferenciais do *software* Venda Mais em relação aos sistemas similares apresentados na Seção 1.3, são os seus algoritmos inteligentes. O Venda Mais possui três algoritmos inteligentes, um reconhecimento facial responsável por analisar os rostos dos clientes que entram na loja, um *matchmaking* que recomenda um melhor vendedor para atender um cliente e a árvore de decisão para sugerir futuras compras.

3.1.1 Reconhecimento facial

O algoritmo de reconhecimento facial do *software* Venda Mais foi totalmente desenvolvido em Java utilizando a biblioteca JavaCV e uma rede neural, a LBPH (KHAN *et al.*, 2019). Esse algoritmo foi desenvolvido em uma aplicação local, implementado com *web socket*, no qual esta aplicação local funciona como um servidor local, que comunica com o *backend* do sistema, implementado também com *web socket*, sendo considerado o cliente dessa arquitetura cliente servidor. Na Figura 1, pode-se observar melhor as funcionalidades e os fluxos do reconhecimento facial.

A biblioteca *JavaCV* é necessária para realizar as detecções de rostos em tempo real, conforme as pessoas passam na frente da câmera de segurança. A montagem da câmera e os componentes necessários para instalação podem ser encontrados na Seção 2.4. Depois de um rosto ser detectado, o sistema utiliza a LBPH para extrair as características do rosto e compará-las com as demais cadastradas, que ficam armazenadas em um arquivo conhecido como classificador, na memória do *notebook*, nomeado “LBPHClassifier.yml”.

Sempre que um cliente ou vendedor é cadastrado no *software*, é gerado um novo arquivo de classificação, que contém dados do rosto desse novo cliente ou do vendedor. E no caso da demissão de vendedores, é gerado um novo arquivo, com os dados do vendedor removido.

Todas as vezes que acontece essa geração, é necessário sobrescrever o classificador anterior para a LBPH utilizar o atualizado. Como o *backend* é o cliente da arquitetura cliente servidor, ele encaminha o arquivo de classificação atualizado de 1024 em 1024 *bytes* para a aplicação local, conforme a aplicação local recebe o classificador atualizado, ela armazena na memória do *notebook* com o nome “LBPHClassifier2.yml”.

Depois da finalização da transferência do arquivo de classificação, o sistema utiliza o recurso de semáforo para bloquear a *thread* principal da aplicação local, exclui o classificador principal e renomeia o arquivo recebido para o mesmo nome do excluído, ou seja, para “LBPHClassifier.yml”.

Em seguida, o sistema desbloqueia a *thread* principal. Foi necessária a utilização de semáforo para evitar *race condition*, porque durante o processo de reconhecimento, o sistema não pode sobrescrever o arquivo de classificação que está sendo usado pela rede neural LBPH.

A LBPH, sempre que realiza o reconhecimento facial, indica um identificador de uma pessoa cadastrada e um valor de confiabilidade que acredita ser o rosto dessa pessoa, quanto menor o valor, maior a probabilidade de ser a pessoa indicada. Se esse valor for maior que 90, que é o valor de *threshold* configurado nesse algoritmo de reconhecimento facial, é retornado o identificador 0, que indica ser uma pessoa não cadastrada no Venda Mais, entretanto, se for menor que 90, é retornado o identificador da pessoa cadastrada no banco de dados.

Para validar a acurácia do reconhecimento facial, fotografou-se seis imagens do rosto do pesquisador e utilizou-se seis fotos dos rostos de 24 pessoas públicas distintas na *Internet*, que totalizam 150 fotos para o treinamento, sendo 25 rostos com seis fotos cada. Para realizar a validação, foram usadas 25 imagens diferentes das fotos do treinamento. Depois da aplicação de treinamento e validação, foi possível verificar que o sistema reconheceu corretamente 21 rostos dos 25 analisados, o que resulta em uma acurácia de 84%.

3.1.2 Matchmaking

O algoritmo de *matchmaking* foi totalmente elaborado para satisfazer os requisitos do *software* Venda Mais, com o intuito de recomendar um melhor vendedor para realizar o atendimento de um determinado cliente.

Em relação aos requisitos, foram solicitados três dados dos vendedores disponíveis e de um cliente, para definir um melhor vendedor. O mais prioritário é o gênero, ou seja, um cliente ser atendido por um vendedor do mesmo gênero. O segundo seria a categoria de roupa que o vendedor domina atender e a categoria de compra do cliente que a árvore de decisão sugeriu.

E o terceiro, seria em relação à faixa etária que o vendedor tem preferência para atender e à faixa etária do cliente, no qual pode ser criança (menor de 12 anos), jovem (de 12 a 18 anos), adulto (entre 18 e 65 anos) ou pessoa idosa.

No desenvolvimento do *matchmaking*, foi utilizado o método estatístico *Elo Rating* (OLMEDA, 2021) para elaborar a Equação 1 e os pesos que cada variável pode atribuir. Esse método estatístico é muito utilizado em aplicativos e sites de relacionamentos, e também, em jogos competitivos de dois jogadores, com o objetivo de definir um melhor adversário para o seu oponente.

$$f(x, y, z) = \frac{1000 * x + y}{z} \quad (1)$$

O *matchmaking* foi elaborado com a linguagem de programação Python, juntamente a uma aplicação local, que necessita de apenas dois parâmetros, uma lista com informações dos vendedores disponíveis e de um cliente.

E então, são aplicadas todas as informações dos vendedores e do cliente na Equação 1, a qual possui um valor constante 1.000, que multiplica a variável “x”, sendo representada pelo peso do gênero. O resultado, é somado com o peso da categoria que o vendedor domina ou a sugestão de compra do cliente que é a variável “y”, em seguida, divide o resultado da soma pela variável “z”, que representa o peso da informação preferência de atendimento do vendedor ou a faixa etária do cliente. O Apêndice G apresenta os pesos de cada informação, tanto do cliente como do vendedor, para os quais as variáveis podem ser atribuídas.

Depois de o *software* obter o resultado final “f(x,y,z)” de cada vendedor disponível, é elaborada uma tabela de classificação dos vendedores. Em seguida, o

software utiliza o resultado obtido com as informações do cliente, para encontrar qual vendedor da tabela de classificação possui um resultado mais próximo do cliente. E assim, indica que esse é o melhor vendedor para realizar o atendimento.

Foram elaboradas duas tabelas com todas as combinações possíveis das informações dos clientes e vendedores, que resultou em 32 combinações em cada tabela, conforme apresentado no Apêndice H. Em seguida, foram aplicadas no algoritmo *matchmaking* todas as combinações de vendedores e de apenas um cliente, sendo executados 32 vezes o algoritmo e alterando apenas o cliente em cada execução sem repetir.

Com essas aplicações, foi possível analisar que o algoritmo sempre recomendava o vendedor com a pontuação mais próxima do cliente, que no caso, seria o mais adequado para realizar o atendimento. Portanto, o algoritmo de *matchmaking* apresenta uma acurácia de 100%.

3.1.3 Árvore de decisão

O algoritmo de árvore de decisão, também foi totalmente elaborado para atender os requisitos do *software* Venda Mais, com o objetivo de sugerir uma melhor categoria de compra para um cliente. No entanto, a categoria sugerida, precisa possuir pelo menos um produto no estoque, com o tamanho que o cliente normalmente compra na loja Campinense.

Para o desenvolvimento da árvore de decisão, foi utilizado o algoritmo de *machine learning* Decision Tree Classifier (JIJO; ABDULAZEEZ, 2021), juntamente a uma aplicação local desenvolvida em Python, que necessita de seis parâmetros de entrada, sendo eles: o gênero do cliente, as categorias de roupa mais comprada, o tamanho mais comprado, o produto mais vendido da loja, o item com maior quantidade em estoque e a categoria que possui vendedores com maior disponibilidade de atendimento.

Com esses parâmetros de entrada, o algoritmo realiza uma análise e retorna uma sugestão de categoria de roupa (camiseta, *short*, vestido ou calça) mais adequada para o cliente realizar futuras compras para ele ou de um presente.

Antes da implementação da árvore de decisão, foi necessário implementar um conjunto de dados com 6.250 instâncias e sete colunas, sendo elas: gênero, comprado, tamanho, vendido, vendedor, estoque, sugestão.

Em seguida, utilizou-se um *script* de análise combinatória em Python, para inserir todas as cinco entradas possíveis no conjunto de dados, sem repetição, no qual pode receber como entradas as seguintes categorias: camiseta, vestido, *short*, calça e indeciso. A indeciso é uma categoria que o sistema não conseguiu definir, um exemplo disso seria quando o cliente não possui uma categoria mais comprada, ou seja, uma mesma quantidade de compra em duas categorias distintas.

A coluna de sugestão é de validação deste *dataset*. Nessa coluna, não aparece nenhuma instância da categoria indeciso, pois o sistema só utiliza essa instância para entradas do *dataset*.

Depois de elaborar o conjunto de dados, foi utilizada a técnica *Cross Validation* (YADAV; SHUKLA, 2016) para avaliar o desempenho do modelo da árvore de decisão, sendo possível determinar se o modelo está “subajustando” (*overfitting*), “superajustando” (*overfitting*) ou em equilíbrio (*balanced*).

O *dataset* foi particionado em 21 subconjuntos de forma aleatória com aproximadamente a mesma quantidade de amostras em cada um deles. E ocorreram 21 interações de treino e teste, no qual um subconjunto foi utilizado para teste e os restantes foram utilizados para treinamento da árvore de decisão gerando uma métrica de avaliação. Durante as interações, cada subconjunto foi utilizado para teste em algum momento da avaliação do modelo.

Depois todo o conjunto de dados ser aplicado no processo de treino e teste, foi plotado um gráfico, que é apresentado no Apêndice I, no qual o eixo das abcissas seria as interações e o eixo das ordenadas, as porcentagens de aprendizagem do modelo árvore de decisão.

Pode-se observar no gráfico, que a árvore de decisão iniciou a aprendizagem gradativamente em relação às interações da técnica *Cross Validation* e alcançou uma acurácia aproximada de 99% na última interação. Por meio desse gráfico, é possível determinar que o modelo de árvore de decisão está em equilíbrio (*balanced*), ou seja, apresenta uma excelente aprendizagem em relação aos testes e treinos realizados no conjunto de dados.

No Apêndice J, é apresentada a estrutura do algoritmo de árvore de decisão depois da aplicação da técnica *Cross Validation*, que permite verificar os seus nós de decisões e os pesos que o algoritmo atribui para cada categoria existente no conjunto.

3.2 Aplicativo do gerente

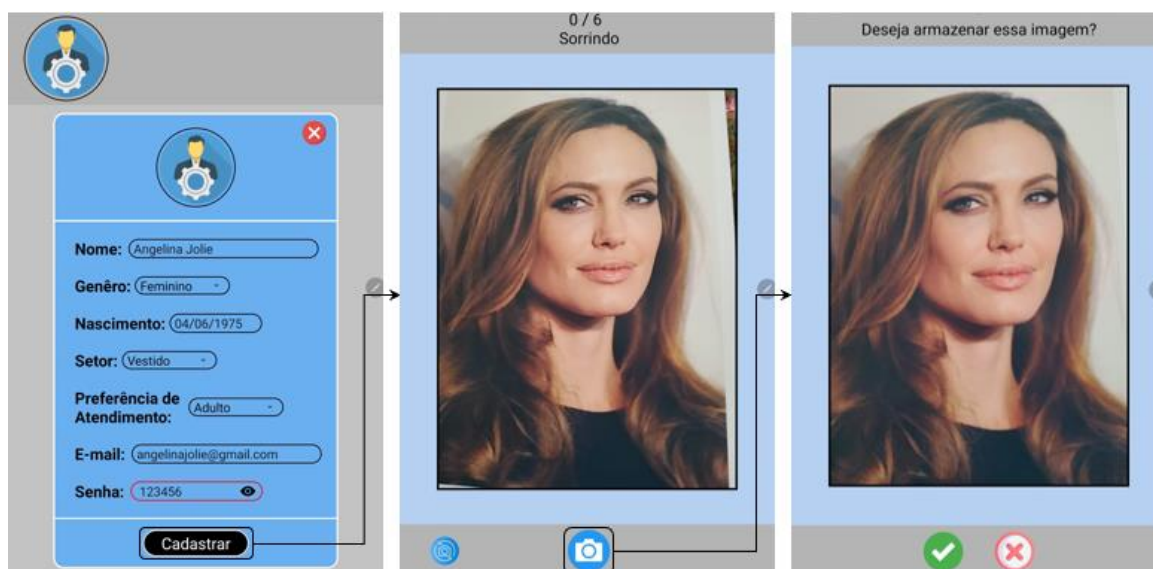
Foi desenvolvido um aplicativo *mobile* em React Native para *tablet*, exclusivo para o gerente da loja, que pode realizar o cadastramento de clientes, cadastro e exclusão de vendedores e busca para análise de algum dado ou sua edição.

Para o gerente utilizar o aplicativo, é necessário que ele se cadastre na aplicação, na qual é solicitado os mesmos dados de um vendedor, pois o gerente pode atender os clientes em um celular com o aplicativo dos vendedores, sem realizar outro cadastramento.

3.2.1 Cadastramento de gerente

Assim que o aplicativo é inicializado pela primeira vez, é mostrada uma tela para a gerente realizar o seu cadastro na aplicação, conforme apresentado o cadastramento da Angelina Jolie como gerente da loja na Figura 5. O sistema aguarda que a gerente insira o nome, gênero, data de nascimento, setor que ela domina atender, faixa etária que tem preferência de atendimento, *e-mail* e senha.

Figura 5. Cadastramento de gerente.



Fonte: Elaborada pelo autor (2022).

Se a gerente pressionar o botão Cadastrar, sem preencher algum campo, o campo não preenchido recebe um destaque em vermelho, o que indica que precisa ser preenchido para a realização do cadastramento.

Se ela preencher algum campo inválido, por exemplo, número no campo de nome ou *e-mail* sem um provedor, esses campos também recebem um destaque em vermelho, no qual indicam que precisam ser revisados, pois é realizada uma validação em todos os campos, com o intuito de evitar dados incoerentes.

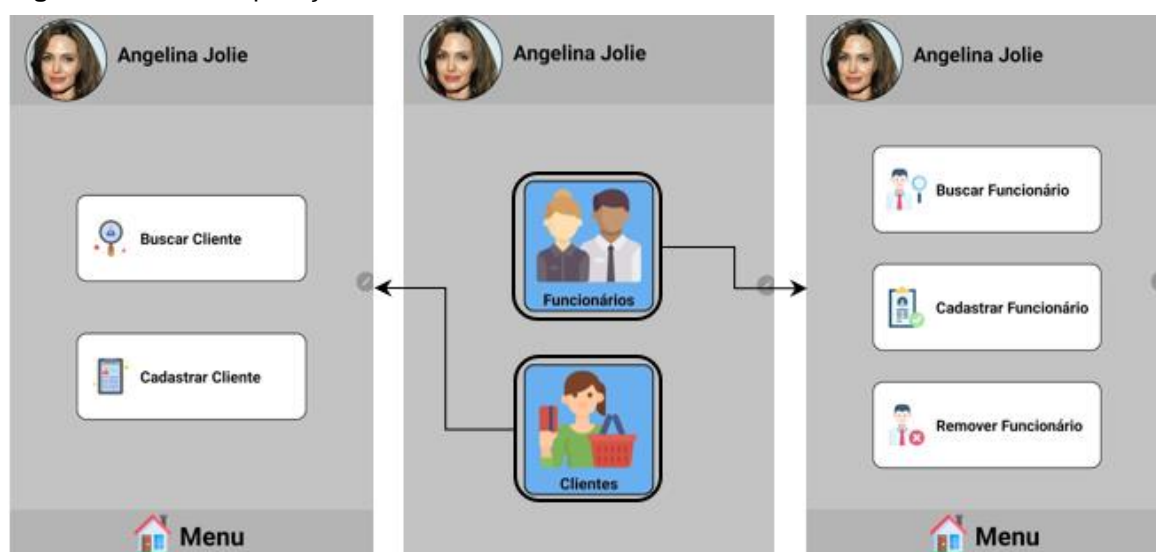
Depois, o *backend* cadastra no banco de dados do Venda Mais a Angelina Jolie na entidade Funcionários, que atribui o identificador com valor um para gerente. O identificador é auto incremental, no entanto, os vendedores cadastrados posteriormente nesta mesma entidade, receberam os identificadores subsequentes. Em seguida, o aplicativo direciona para a segunda tela apresentada na Figura 5, com isso é necessário fotografar seis fotos do rosto da gerente.

Conforme a gerente fotografa o rosto, a aplicação disponibiliza uma outra tela para ela decidir se deseja salvar ou excluir a imagem fotografada. E sempre que ocorre atualização da quantidade de fotos salvas, o aplicativo atualiza o índice de imagens armazenadas, no qual não ultrapassa o valor seis.

Depois de a gerente realizar o seu cadastro, a aplicação direciona ao menu principal descrito no próximo subitem. E todas as vezes que o aplicativo é inicializado novamente, também, é direcionado ao menu principal, sem a necessidade de realizar outro cadastramento. Se futuramente, ocorrerem alterações de gerente na loja Campinense, apenas é necessário alterar os dados do novo gerente no sistema, mantendo assim o identificador com valor um no banco de dados do Venda Mais.

3.2.2 Menu principal

Figura 6. Menu da aplicação.



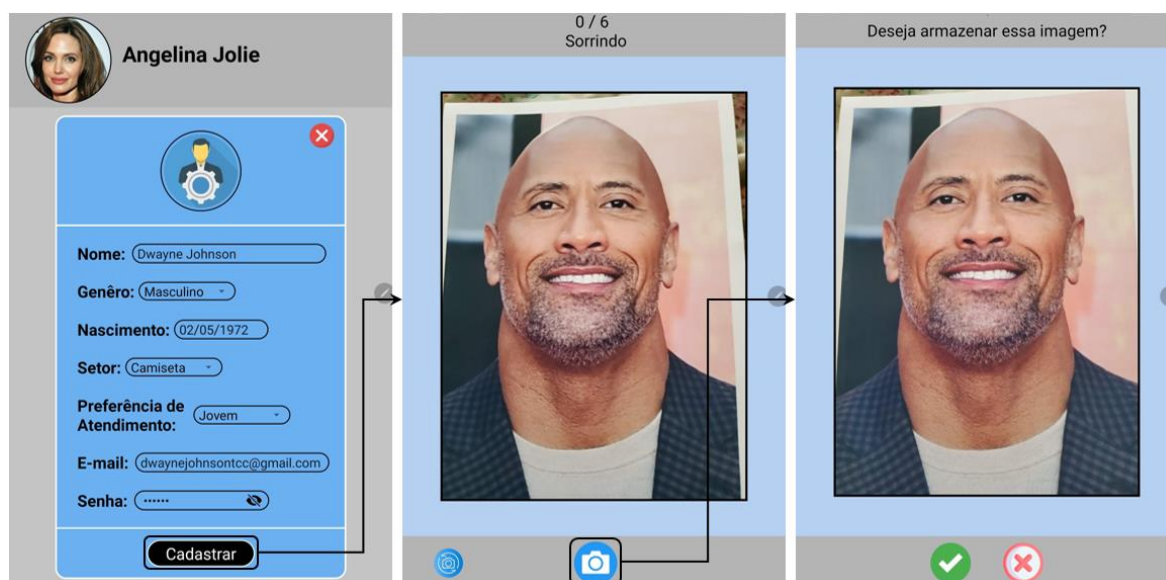
Fonte: Elaborada pelo autor (2022).

A tela de menu principal é apresentada na imagem no centro da Figura 6. No menu, são disponibilizados dois botões, um Funcionários e o outro Clientes a escolha da gerente. Quando pressionado o botão Funcionários, é direcionado para uma outra tela que permite a gerente buscar, cadastrar ou remover os vendedores. E quando pressionado o botão Clientes, em outra tela, permite a gerente buscar e cadastrar clientes.

3.2.3 Cadastramento de vendedor e cliente

A Figura 7 apresenta os fluxos relacionados ao cadastramento do vendedor Dwayne. O *software* mostra as mesmas telas e realiza os mesmos procedimentos do cadastramento da gerente Angelina.

Figura 7. Cadastramento de vendedor.



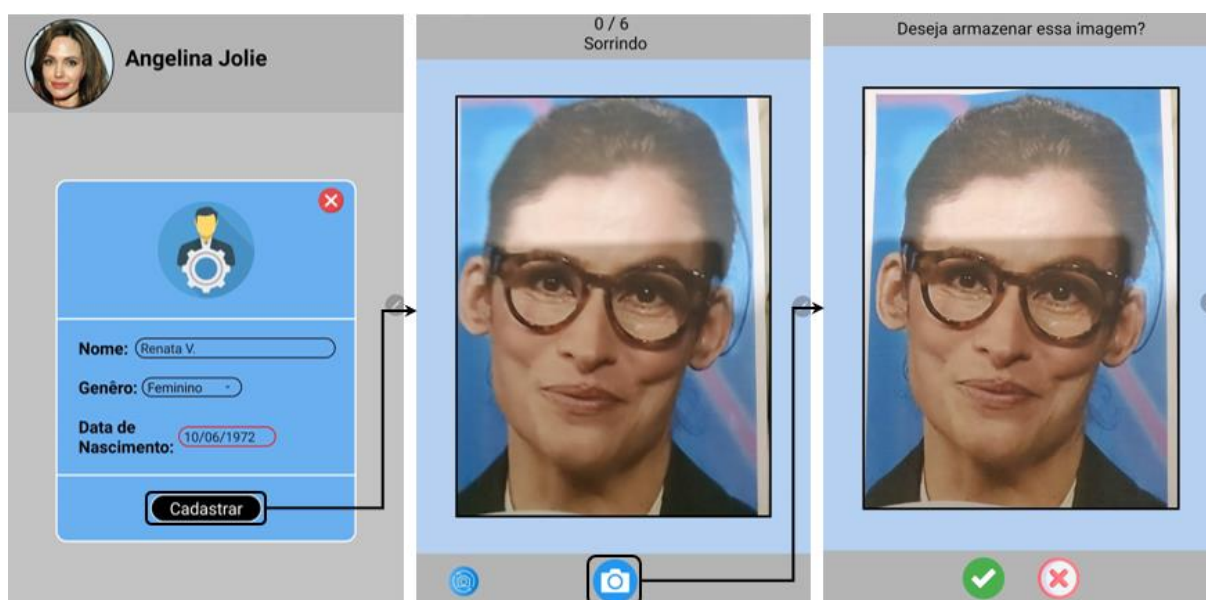
Fonte: Elaborada pelo autor (2022).

A tela para realizar o cadastramento de clientes, é diferente da dos funcionários (gerente e os vendedores da loja), pois para um cliente é solicitado menos dados e, também, tem como foco a realização de um cadastro rápido e simplificado para o cliente não aguardar muito tempo e até mesmo desistir de se cadastrar.

A Figura 8 apresenta os fluxos relacionados ao cadastramento da cliente Renata. O sistema aguarda que a gerente insira o nome, gênero e a data de nascimento. Assim como no cadastramento e armazenamento de fotos dos funcionários, o *software* realiza os mesmos procedimentos, no entanto, com

requisições HTTP própria para clientes e, também, é necessário fotografar seis fotos do rosto da cliente.

Figura 8. Cadastramento de cliente.



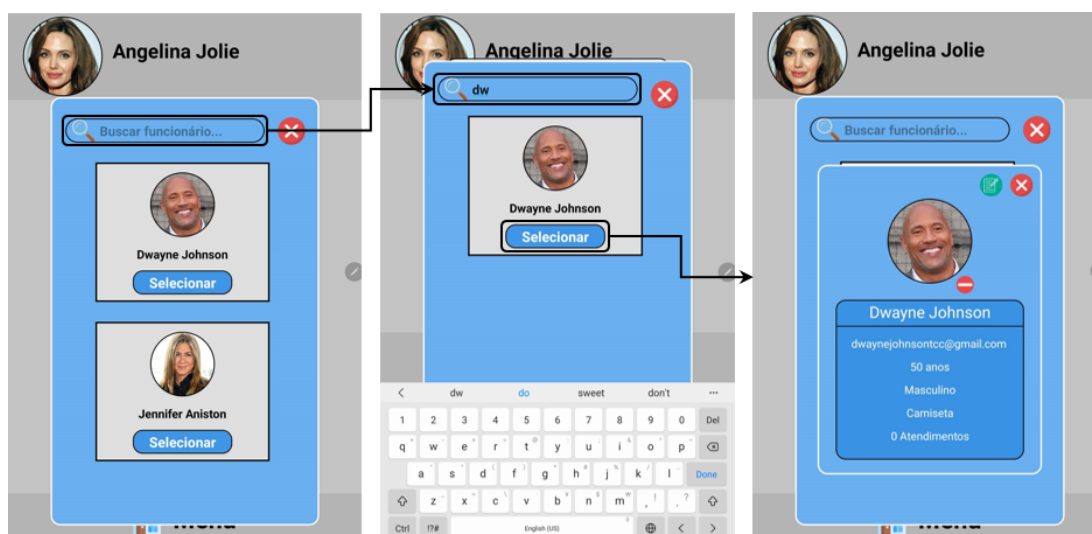
Fonte: Elaborada pelo autor (2022).

Se a gerente pressionar o botão Cadastrar, com algum campo inválido, tanto no cadastramento de vendedores como no de clientes, o sistema aplica a mesma validação descrita no Subitem 3.2.1 em todos os campos, com o mesmo intuito de evitar dados incoerentes.

3.2.4 Busca de vendedor e cliente

A Figura 9 apresenta os fluxos referentes à busca de um vendedor na aplicação. A gerente precisa inserir apenas o nome do vendedor que ela deseja encontrar, o aplicativo se comunica com o *backend* enviando o nome conforme é atualizado e recebe a lista dos vendedores cadastrados no banco de dados do Venda Mais, por meio de requisição HTTP. Essa busca não é sensível à letra maiúscula e tem um tratamento para sempre retornar os vendedores em ordem alfabética.

Da mesma maneira, o sistema Venda Mais utiliza essa funcionalidade e comportamento para buscar um cliente, no entanto, com requisições HTTP diferentes e o *backend*, retorna uma lista em ordem alfabética de clientes cadastrados no banco de dados do Venda Mais.

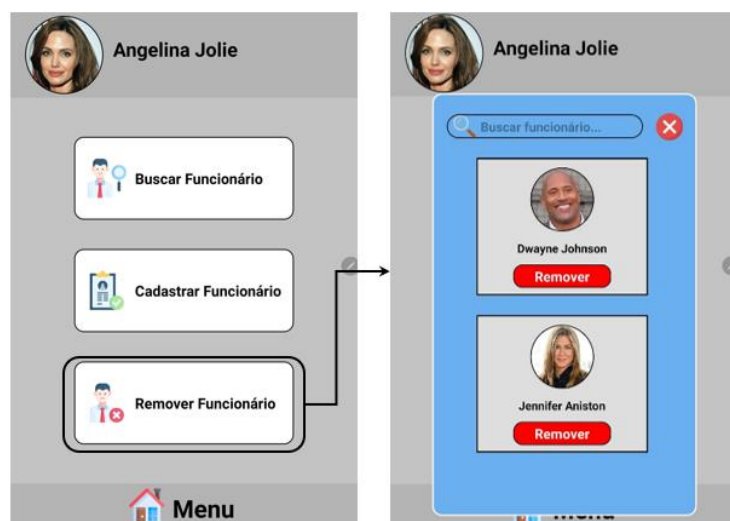
Figura 9. Busca de um vendedor.

Fonte: Elaborada pelo autor (2022).

Na Figura 9, também é possível observar que, depois de a gerente encontrar o vendedor, ela pode pressionar o botão Selecionar e verificar a foto e todos os seus dados, que podem ser editados. Os vendedores possuem um ícone abaixo da foto, que indica se eles estão disponíveis ou ocupados para realizarem um atendimento.

3.2.5 Remoção de vendedor

Da mesma maneira que a gerente procura um vendedor para analisar ou editar seus dados, ela também pode removê-lo, selecionando o botão Remover Funcionário e, após encontrá-lo, pressiona o botão Remover, conforme mostra a tela da Figura 10.

Figura 10. Remoção de vendedor.

Fonte: Elaborada pelo autor (2022).

3.3 Aplicativo dos vendedores

Foi elaborado um aplicativo *mobile* em React Native para celulares, para utilização dos vendedores da loja Campinense. Essa aplicação é focada em auxiliar o vendedor na realização de um atendimento personalizado para os clientes. Inicialmente, necessita se autenticar, para poder utilizar os recursos do aplicativo.

3.3.1 Autenticação de vendedor

A Figura 11 apresenta a primeira tela desse aplicativo, a de autenticação, onde há dois campos para serem preenchidos, o de *e-mail* e o da senha que permite o vendedor exibir ou ocultar a senha, conforme é pressionado o ícone de um olho no campo senha.

Figura 11. Autenticação de vendedor.



Fonte: Elaborada pelo autor (2022).

O vendedor precisa inserir os dados que foram cadastrados pela gerente, no momento de seu cadastramento, e depois pressionar o botão Entrar, no qual o sistema encaminha esses dados para o *backend* por meio de requisição HTTP. Onde tem

configuração, gerenciamento e o armazenamento dos *tokens* desenvolvidos com auxílio de uma biblioteca em Spring Boot, a Spring Security.

Depois da validação de autenticação do *token*, o *backend* retorna para o aplicativo se os dados inseridos estão corretos ou não, caso estiverem corretos, o vendedor é direcionado para a próxima tela da aplicação.

Entretanto, caso estiverem incorretos, o aplicativo destaca os campos em vermelho e exibe uma mensagem para o vendedor que indica incoerência em um dos dois campos. A aplicação permite que o vendedor realize várias tentativas para autenticação.

3.3.2 Entrada de cliente

A tela, apresentada na Figura 12, refere-se à segunda da aplicação, que o vendedor só consegue acessar depois de passar pela etapa de autenticação. Assim que entra nessa tela, o aplicativo utiliza o identificador do vendedor obtido por meio do *token* para enviar uma requisição HTTP para o *backend*, no qual é retornado os dados do vendedor referente ao identificador. Em seguida, armazena os dados do vendedor na memória do dispositivo móvel, atualiza as variáveis do aplicativo, mostra a sua foto e o seu nome na tela e altera o status para disponível.

Figura 12. Entrada de cliente na loja Campinense.



Fonte: Elaborada pelo autor (2022).

Sempre que chega um cliente na loja, o Venda Mais realiza a inserção de uma nova tabela de atendimento no banco de dados, que possui os campos identificador do cliente e do vendedor, horário de chegada, que é preenchido automaticamente, e horário de saída, que fica vazio nesse momento, sendo preenchido apenas quando o vendedor encerra o atendimento.

Se o cliente tiver sido reconhecido, armazena-se o seu respectivo identificador cadastrado e verifica-se se há vendedores disponíveis. Se tiver, utiliza-se o algoritmo *matchmaking*, descrito na Seção 3.1.2, para indicar um melhor vendedor para atender esse cliente e armazena o seu identificador. Se nenhum vendedor estiver disponível, deixa-se o campo desse identificador vazio.

E, se o cliente não for reconhecido, confere-se o valor zero para seu identificador, que indica ser cliente desconhecido e atribui o identificador do vendedor disponível com menor número de atendimentos. Se nenhum estiver disponível, deixa-se esse campo vazio.

Na Figura 12, é apresentado quando o aplicativo realiza um processo conhecido como *polling*, no qual envia uma requisição HTTP em um em um minuto para o *backend*, com o intuito de verificar se existe uma nova tabela de atendimento ou um cliente para ser atendido. Para isso, o *backend* analisa se tem no banco do Venda Mais nas tabelas atendimentos, algum campo horário de saída vazio e no campo identificador, o identificador do vendedor *logado* ou vazio.

Portanto, se o campo horário de saída tiver vazio, indica que o cliente ainda não foi atendido. E se constar na tabela de atendimento, o identificador do vendedor *logado*, mostra que esse vendedor foi indicado para realizar o atendimento, então o *backend* encaminha para o aplicativo os dados do cliente para ser atendido.

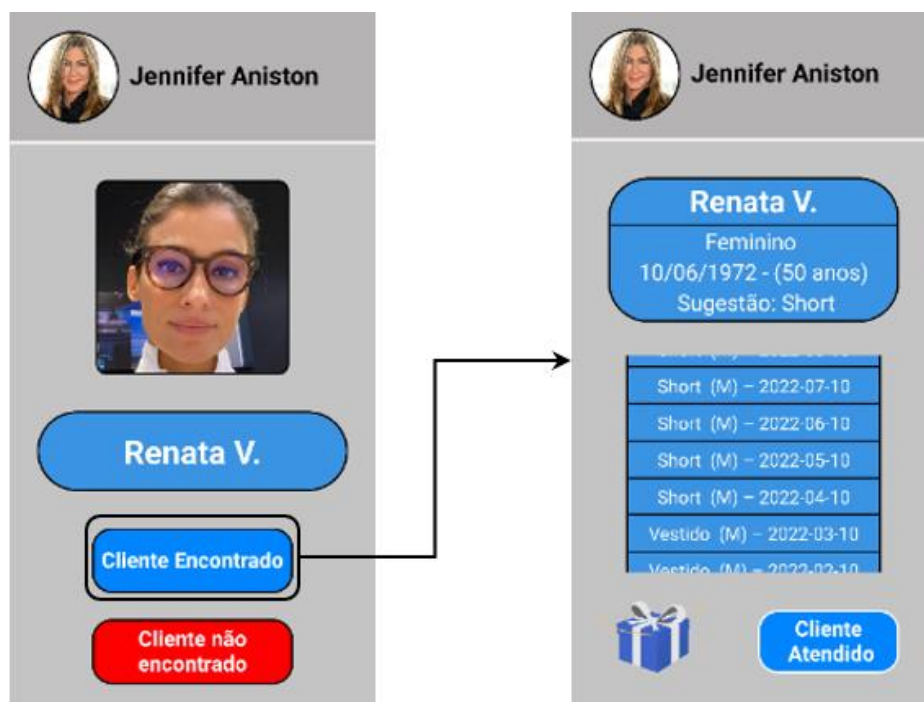
E, se o campo identificador do vendedor estiver vazio e o vendedor *logado* não estiver sido indicado para nenhum outro atendimento, o sistema atualiza esse campo com o identificador desse vendedor e envia para o aplicativo os dados do cliente, para ser realizado o atendimento.

3.3.3 Realização de atendimento

A Figura 13 apresenta uma tela que informa que a vendedora Jennifer foi indicada para realizar o atendimento de uma cliente cadastrada, que no caso foi reconhecida pela rede neural, sendo a Renata. Nessa tela, pode-se verificar que há

dois botões, o de Cliente Encontrado, em azul, e o Cliente não encontrado, em vermelho.

Figura 13. Cliente reconhecido para o vendedor atender.



Fonte: Elaborada pelo autor (2022).

O primeiro botão só é pressionado se a vendedora visualizou a foto da cliente e encontrou ela de fato na loja, aproximando-se para iniciar a realização do atendimento. Depois disso, o *status* da vendedora é alterado para ocupado e o aplicativo direciona para a segunda tela apresentada na Figura 13. Onde é mostrado os seus dados, como gênero, idade e a sua sugestão de compra que foi definida por meio do algoritmo de inteligência artificial árvore de decisão descrito na Seção 3.1.3.

O botão Cliente não encontrado é pressionado quando a vendedora visualiza a foto da cliente Renata e não encontra ela na loja. Pois pode acontecer de a cliente entrar na loja e não ter nenhum vendedor disponível no *software* Venda Mais, e ela finalizar a sua compra no caixa, sem ter sido atendida por um vendedor utilizando o *software*. Nesse caso, é excluído o atendimento do banco de dados, que não incrementa um atendimento para o vendedor e altera o seu *status* para disponível.

A Figura 14 apresenta um cliente não reconhecido pela rede neural e o aplicativo indicou o vendedor Dwayne para realizar o atendimento. Depois de pressionar o botão Cliente Encontrado, o aplicativo direciona para outra tela, no qual

aguarda o vendedor inserir os dados do cliente, como gênero, idade e tamanho e as características de roupas que ele deseja comprar para si ou presentear. Também é possível selecionar verão ou inverno, social ou casual e listrada ou estampada, e assim, solicitar uma sugestão de compra para esse cliente, depois de pressionar o botão Requisitar sugestão.

Figura 14. Cliente não cadastrado para o vendedor atender.

Fonte: Elaborada pelo autor (2022).

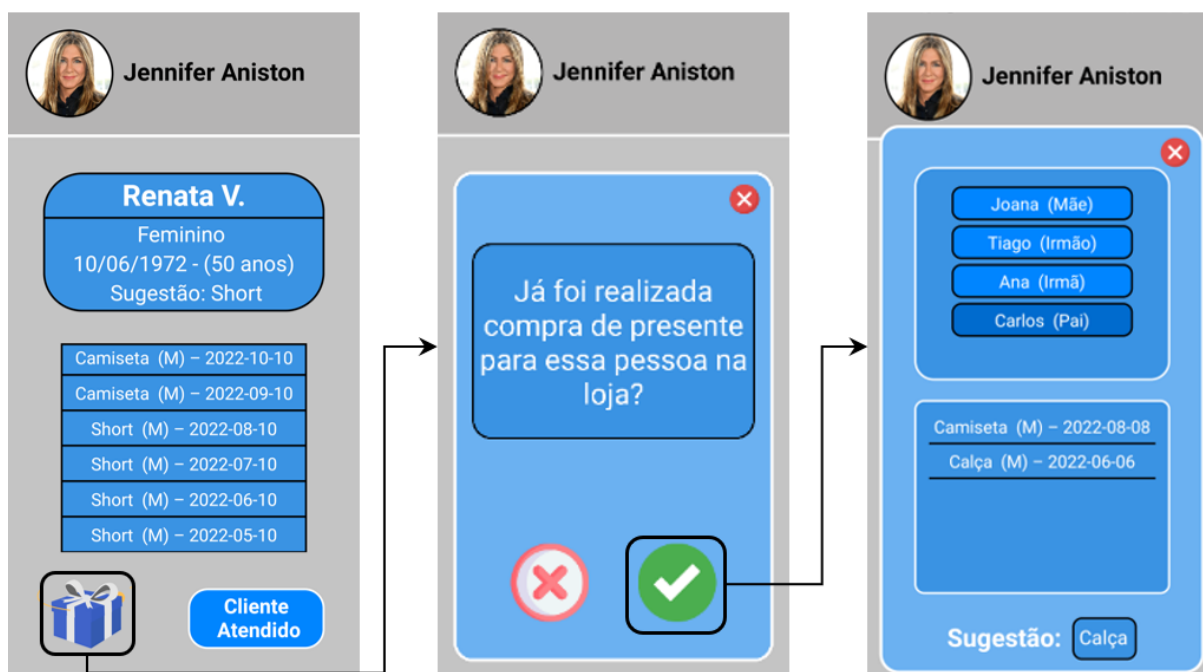
Como não há nenhum dado do cliente desconhecido, é necessário esse processo para poder inserir parâmetros no algoritmo árvore de decisão, descrito na Seção 3.1.3. Os campos referentes aos dados do cliente são obrigatórios, o que gera destaque em vermelho caso seja solicitada sugestão, e o campo não for preenchido.

3.3.4 Compra de presentes

A Figura 15 apresenta os fluxos de quando uma pessoa já comprou pelo menos um presente para um amigo ou familiar e deseja presentear novamente. Nesse caso, a vendedora pressionou o ícone de presente, sendo direcionada para uma tela que pergunta se a cliente que está sendo atendida já comprou presente para a pessoa que

deseja presentear, com a resposta afirmativa, a vendedora pressiona o botão correspondente ao sim, e é direcionada para outra tela.

Figura 15. Compra de presente para pessoa já presenteada.



Fonte: Elaborada pelo autor (2022).

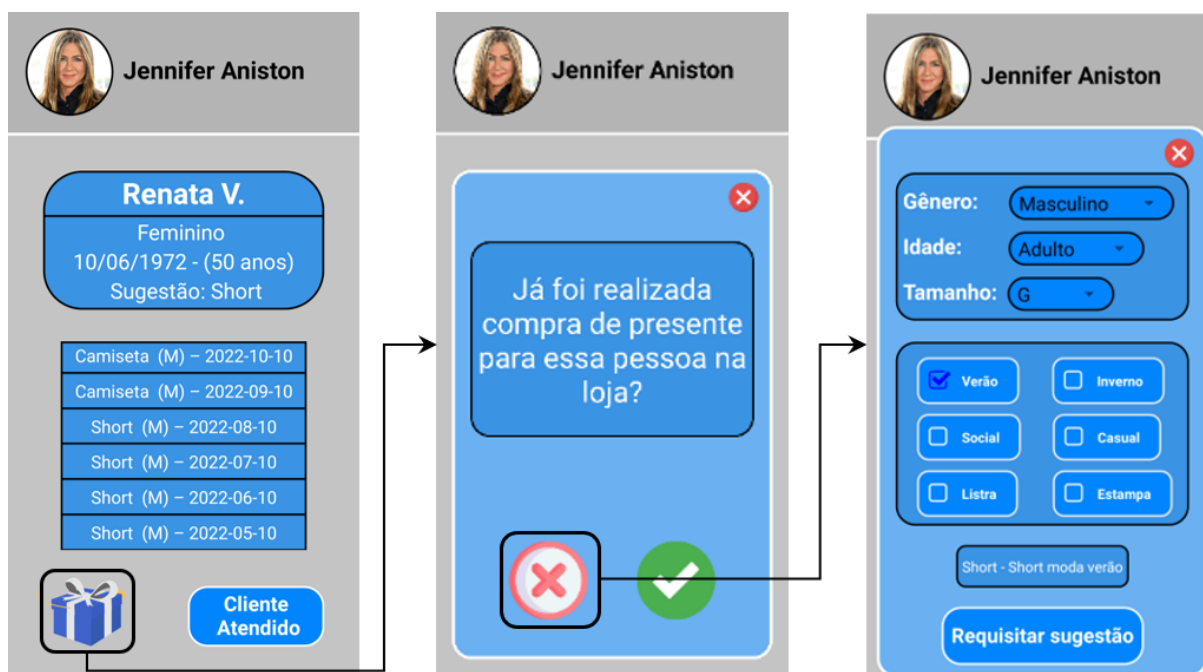
Nessa outra tela, é possível verificar as pessoas que a cliente já presenteou, no caso são quatro pessoas, Joana, Tiago, Ana e Carlos. Conforme a vendedora pressiona os botões contidos os nomes de cada pessoa, é possível visualizar o histórico de compras de presentes de cada pessoa e qual sugestão de compra o algoritmo árvore de decisão sugere para ela. No caso da Figura 15, a cliente gostaria de presentear o Carlos, o sistema faz uma análise das compras e sugere a categoria de roupa calça, como presente.

Se caso o vendedor pressionar o botão indicando que a cliente já comprou presente, para uma pessoa que ela deseja presentear, e não houver nenhum registro de compra de presente no banco de dados do Venda Mais, o sistema exibe uma mensagem em vermelho na tela escrito: "Nenhuma compra de presente encontrada no sistema".

A Figura 16 apresenta a opção que a cliente ainda não tem registro de compras de presentes para a pessoa que ela deseja presentear. Como não há nenhum dado dessa pessoa, o sistema a considera como desconhecida e solicita alguns dados pessoais, como gênero, idade e tamanho e características de roupas, como verão ou

inverno, social ou casual e listrada ou estampada. Assim, para solicitar uma sugestão de presente para essa pessoa, pressiona-se o botão Requisitar sugestão.

Figura 16. Compra de presente para uma nova pessoa.



Fonte: Elaborada pelo autor (2022).

Nesse caso, foram inseridos os seguintes dados: masculino, adulto e tamanho de roupa G e característica de roupa apenas verão. O *backend* analisa todas essas informações, percorre os produtos em estoque referente a esses dados de entrada e insere os parâmetros necessários na árvore de decisão. O algoritmo de inteligência artificial sugeriu para a cliente presentear com um produto *short* moda verão, o que atendeu os requisitos de entrada.

Essa opção compra de presente só está disponível para clientes cadastrados no *software* Venda Mais. Caso um cliente desconhecido deseje presentear alguém e o vendedor selecione o ícone de presente, o aplicativo exibe um *toast* ou uma mensagem na tela, escrita: "Opção apenas para clientes cadastrados!".

4 AVALIAÇÃO E VALIDAÇÃO

O estudo descrito nesta monografia visa possibilitar um atendimento personalizado com os clientes da loja Campinense, que permita que os vendedores realizem atendimentos com mais atenção e considere os itens mais comprados de cada cliente. Com o objetivo de propor uma solução que potencialize o aumento nas vendas da loja Campinense. Para validar o cumprimento desse objetivo, foi elaborado um roteiro para ser seguido no dia da avaliação do Venda Mais.

4.1 Avaliação

Para avaliação do objetivo descrito nesta monografia, inseriu-se dados nos dois bancos de dados relacionais, o da loja e do Venda Mais, sendo esses bancos, as entidades e cada atributo apresentados detalhadamente na Seção 2.2. E também, elaborou-se uma pasta compartilhada no Google Drive, que contém todos os anexos que foram preparados para realizar a avaliação, que pode ser encontrada no Apêndice K. Nessa pasta compartilhada, dentro do diretório Validações, encontram-se cinco imagens que foram impressas em papel fotográfico para simular a passagem de vendedor ou cliente na frente da câmera.

Inicialmente, foram adicionados dados na entidade de produtos no banco da loja, sendo cadastradas cem peças de roupas nas categorias camiseta, calça, *short* e vestido, com as descrições de verão, inverno, social, casual e estampada para cada categoria, e também, seis tamanhos de roupas, que seriam PP, P, M, G, GG, XGG, no qual totalizam 14.400 peças de roupas.

Em seguida, no banco de dados do Venda Mais, realizou-se o cadastramento da Angelina como gerente e a Jennifer e o Dwayne como vendedores em uma entidade funcionários, com as informações do nome, gênero, data de nascimento, setor predominante, disponibilidade, número de atendimentos, *e-mail*, senha e faixa etária que o indivíduo tem preferência de atender. Foram armazenadas as fotografias do rosto da gerente e dos vendedores, seis fotos por cadastramento.

Foram cadastrados dois clientes na tabela clientes no banco do Venda Mais, a Renata V. e o Leonardo DiCaprio, com as informações do nome, gênero e data de nascimento e outro campo de sugestão de compra. Foram armazenadas fotografias dos rostos dos clientes, seis fotos por cadastramento.

E depois, no banco de dados da loja, realizou-se inserções de compras dos dois clientes. Para a Renata V. apenas roupas com tamanho M e foi inserido mais compras da categoria *shorts* e para o Leonardo DiCaprio apenas roupas com tamanho G e realizou-se inserções de mais compras da categoria camiseta.

Também foram efetuadas inserções de compra de presentes para quatro familiares fictícios da cliente Renata; para o Leonardo não foi realizada nenhuma compra de presente.

Para a realização da avaliação do objetivo, o desenvolvedor e o coorientador deste estudo contaram com a participação de um especialista no setor de vestuário. O profissional trabalha na área de lojas varejistas há mais de dez anos e está familiarizado com todas as atividades pertinentes, além de dominar a arte de atender bem o cliente. O artefato foi apresentado para o especialista de maneira remota, em uma reunião *on-line* na plataforma Google Meet.

A avaliação aconteceu no dia 20 de outubro, no período da tarde, e a plataforma foi apresentada pelo *notebook* pessoal do desenvolvedor-aluno. Também foram utilizados dois celulares, os quais o desenvolvedor já deixou *logada* a conta dos vendedores, um *tablet* e a câmera de segurança montada como detalhado na Seção 2.4.

Depois da realização dessa preparação, o desenvolvedor percorreu – para que o especialista visualizasse a plataforma em funcionamento – os seguintes passos:

- 1 – passou a foto da cliente Renata V. na frente da câmera, com o intuito do sistema reconhecê-la e indicar uma sugestão de compra;

- 2 – no celular que a vendedora Jennifer estava *logada*, era esperado que ela fosse notificada com a foto da Renata para atendê-la e com uma pergunta na tela se ela foi encontrada na loja ou não. Depois de o *software* notificá-la, o desenvolvedor indicou que sim e verificou as informações da cliente e suas últimas compras;

- 3 – apresentou para o avaliador, a opção de compras de presentes para as quais ainda não tinham sido realizadas aquisições;

- 4 – apresentou para o avaliador a opção de verificar sugestão de compra de presentes, para aqueles quatro familiares fictícios da cliente Renata cadastrados no banco de dados da loja, sendo possível observar cada pessoa, histórico de compras e a sugestão de compra;

- 5 – finalizou o atendimento, o que permitiu que os dois vendedores ficassem disponíveis novamente.

6 – passou com a foto do cliente Leonardo DiCaprio na frente da câmera, com o mesmo intuito do primeiro passo;

7 – era esperado o mesmo comportamento do segundo passo, ou seja, que o sistema notificasse o vendedor Dwayne para realizar esse atendimento. Depois de o *software* notificá-lo, o desenvolvedor indicou que o cliente havia sido encontrado e verificou as informações do cliente e suas últimas compras;

8 – mostrou para o avaliador que o sistema sugeriu corretamente a categoria camiseta, pois Leonardo tem um maior cadastro no banco da loja de compra nessa categoria, e o estoque da loja possui pelo menos uma camiseta com o tamanho G, que é o tamanho mais comprado pelo cliente. Depois, foi finalizado o atendimento;

9 – o desenvolvedor, que não está cadastrado no sistema, passou com o seu rosto na frente da câmera, e simulou um cliente desconhecido;

10 – como a vendedora Jennifer possui o menor número de atendimentos entre os vendedores disponíveis, ela foi notificada com a foto do aluno para atendê-lo e com a mesma pergunta se o cliente foi encontrado na loja ou não. O desenvolvedor indicou que sim e mostrou para o avaliador que se tratava de um cliente não cadastrado no *software* Venda Mais.

4.2 Validação

No decorrer da avaliação junto com o especialista, foram seguidas as instruções descritas anteriormente, e o *software* Venda Mais pôde ser executado sem apresentar nenhum problema ou falha, como foi planejado. Depois de o especialista validar a avaliação descrita na Seção 4.1, afirmou que de fato o objetivo deste trabalho foi atingido. Pois, a loja espera um atendimento personalizado com os seus clientes e a solução possui o reconhecimento facial, indicação de melhor vendedor e sugestão de compra, o que são diferenciais que garantem destaque competitivo para loja Campinense em relação às concorrentes, que atendem os seus clientes de maneira tradicional. Podendo o aplicativo com suas especificidades aumentar potencialmente as vendas da loja Campinense.

5 CONCLUSÃO

Neste documento descreve um estudo que teve como foco possibilitar para uma loja varejista de roupas um atendimento personalizado com os seus clientes, por meio de uma solução com reconhecimento facial, sugestão de futuras compras e indicação de um melhor vendedor para realizar o atendimento de um cliente. O estudo atingiu o seu objetivo, ou seja, apresentar uma solução para potencializar o aumento de vendas da loja Campinense.

5.1 Aspectos de inovação e aprimoramento

A solução Venda Mais elaborada para este estudo apresenta diversas funcionalidades e métodos que eram pouco conhecidas ou totalmente desconhecidas pelo pesquisador. O *software* foi desenvolvido utilizando um ambiente Windows com ferramentas para desenvolvimento *Mobile*, no qual, permitiu que o aluno adquirisse bastante conhecimento técnico sobre algoritmos de inteligência artificial, controle de versionamento, requisições HTTP, banco de dados, desenvolvimento de aplicativos, *design pattern MVC*, manipulação e conversão de imagens e autenticações. A seguir, estão descritos os principais aspectos de inovação e aprimoramento do Venda Mais, e o seu nível de complexidade.

- Integração e utilização dos dois bancos de dados – Foi utilizado o PostgreSQL para gerenciar a plataforma do banco de dados da loja e do Venda Mais localmente. Esse item é considerado uma complexidade de nível 7 em 10.
- *Backend* – Foi utilizada a linguagem de programação Java com o *framework* Spring Boot para realização de toda a estruturação do sistema de *backend* do Venda Mais, no qual foi implementado no *notebook* pessoal do aluno. Essa implementação é considerada o de maior complexidade do Venda Mais, sendo 10 em 10.
- Interfaces que o usuário irá interagir com o sistema – Foi utilizada a linguagem de programação Java Script com o *framework* React Native. Como o aluno tinha pouco conhecimento dessa linguagem de programação e desse *framework*, é considerada uma complexidade de nível 8 em 10.

- Reconhecimento Facial – Foi utilizada a rede neural LBPH para realizar o reconhecimento facial dos clientes na loja. Esse item é considerado uma complexidade de nível 6 em 10.
- Autenticação de vendedores – Necessária para validar se o vendedor está cadastrado no Venda Mais. Essa complexidade seria de nível 5 em 10.
- Indicação de um vendedor para um cliente – Foi desenvolvida uma aplicação local e integrada com o algoritmo *matchmaking*, utilizando a linguagem de programação Python. Sendo considerada uma complexidade de nível 7 em 10.
- Sugestão de categoria para o cliente – Foi utilizada a mesma aplicação local desenvolvida no item anterior, integrada com o algoritmo árvore de decisão, desenvolvido na linguagem de programação Python. Essa complexidade pode ser considerada de nível 8 em 10.
- *Design* de interface de usuário – Todos os materiais relacionados a UI como ícones, cores e fontes das letras para a elaboração das interfaces que a gerente e os vendedores realizam interações, foi elaborado e escolhido pelo desenvolvedor, e também, o mesmo desenvolveu uma prototipagem das telas, com fluxos funcionais conforme aciona um botão, para auxiliar na elaboração dos aplicativos. Sendo uma complexidade relevante de nível 6 em 10.
- Câmera de segurança – Por conta de o aluno não possuir conhecimentos elétricos e eletrônicos relacionados à integração da câmera de segurança no *notebook* e ao algoritmo de reconhecimento facial, esse item pode ser considerado complexidade 6 em 10.

5.2 Qualidade e grau de complexidade do software

Nesse item descreve as análises realizadas referente ao grau de complexidade e qualidade de *software* aplicadas sobre a solução proposta para o estudo descrito nesta monografia.

5.2.1 Qualidade

Para realização da análise de qualidade do *software* desenvolvido para este TCC, foi utilizada a norma S. O. IEC 25010:2011 (ISO, 2011) sobre requisitos e

avaliação de qualidade de sistemas e *softwares*. Essa norma define oito principais métricas externas e internas de qualidades, distribuídas em oito principais categorias:

- adequação funcional: é a métrica que estabelece a capacidade do produto de *software* prover funções que atendam às necessidades explícitas e implícitas do usuário. O Venda Mais atende essa métrica, pois foi desenvolvido baseado em requisitos específicos da loja Campinense, que implementa todas as regras de negócio de maneira genérica;
- confiança: para um sistema de enquadrar nessa categoria, necessita tolerar falhas e manter um nível de desempenho. O Venda Mais não satisfaz completamente essa categoria, pois não foi realizado um plano de testes e nem testes automatizados, apenas foram aplicados testes manuais elaborados pelo desenvolvedor;
- usabilidade: é considerada a capacidade do *software* ser compreendido, operado e atraente ao usuário. O Venda Mais atende a essa métrica, pois, depois de ser realizada a avaliação, o avaliador afirmou que as interfaces dos dois aplicativos são bem intuitivas e fáceis de serem manuseadas, contendo ícones e botões didáticos. A aplicação como um todo foi planejada para que os vendedores ou gerente de loja possam interagir com os aplicativos, independentemente de um tutorial;
- eficiência de desempenho: um sistema para se enquadrar nesse atributo, necessita ter um desempenho adequado em relação ao tempo de processamento e à quantidade de recursos utilizados. O Venda Mais foi submetido apenas às situações de poucas pessoas passando juntas ou sozinhas na frente da câmera e no máximo dois vendedores disponíveis, similar com o cenário apresentado na avaliação do sistema, no Subitem 4.1. Nesse cenário, o sistema apresentou um tempo de resposta aceitável, entretanto, o *software* não foi submetido às avaliações com um grande número de clientes e vendedores, para a obtenção de uma melhor análise de seu desempenho;
- segurança: um *software* seguro necessita garantir uma confidencialidade, autenticidade, responsabilidade e proteção de funções e dados, não permitindo acesso de usuários não autorizados. O Venda Mais satisfaz essa métrica, pois possui um sistema de autenticação, no qual apenas vendedores que estão cadastrados conseguem fazer registros na aplicação, o sistema funciona

apenas localmente e não necessita de dados sensíveis dos vendedores e clientes;

- compatibilidade: é a métrica que estabelece a capacidade do *software* interagir com outros sistemas sem causar impactos negativos. O aplicativo dos vendedores pode ser instalado em diversos dispositivos diferentes. E o sistema foi elaborado com diversas *threads* e tratamentos de *race conditions*, preparado para responder diversas requisições simultâneas. Portanto, é possível dizer que o Venda Mais atende a esse requisito de qualidade;
- manutenibilidade: estabelece a capacidade do sistema ser modificado, reutilizado, testado e de detectar e corrigir erros. O Venda Mais foi desenvolvido desde o início respeitando o padrão de *design* MVC (*Model - View - Controller*) e, também, sempre foram utilizadas boas práticas de programação. Logo, é possível afirmar que o Venda Mais atende a essa métrica de qualidade;
- portabilidade: um *software* portátil é aquele que pode ser efetivamente instalado ou acessado em diversos dispositivos. O Venda Mais satisfaz essa métrica, pois o aplicativo dos vendedores pode ser utilizado em diversos celulares simultaneamente. Os dois aplicativos foram desenvolvidos em React Native, o que permite o funcionamento em dispositivos Android e IOS. E os aplicativos são responsivos para se adaptar em qualquer tela, independentemente do seu tamanho e da sua resolução.

5.2.2 Grau de complexidade

Nesse item, foi analisada a quantidade de processamento que o Venda Mais necessita, separadamente em cada módulo com maior importância para o seu funcionamento. Nessa análise foram priorizados os pontos mais críticos dos módulos.

Primeiramente, foi analisado o *backend* do *software* Venda Mais, em que pior caso ocorre no processo de comunicação com o *frontend* e o algoritmo de reconhecimento facial (LBPH), quando é realizado o cadastramento de um novo cliente ou funcionário. Pois, o *backend* precisa converter a imagem, realizar o treinamento do LBPH, gerar um novo classificador e bloquear a *thread* do algoritmo de reconhecimento facial, no qual encaminha o arquivo via *socket*. Resumidamente, é considerado uma complexidade de $O(4n) + O(n^2)$, ou seja, prevalece o $O(n^2)$.

O *frontend* foi elaborado com o intuito de apresentar um melhor desempenho para interação do usuário. No entanto, o caso mais crítico da aplicação, ocorre quando o vendedor realiza a autenticação. Pois o sistema envia os dados do vendedor para o *backend*, aguarda a validação e depois de validado, precisa armazenar os dados do vendedor na memória do dispositivo. Sendo considerada uma complexidade de $O(n)$, no qual é influenciada linearmente pela quantidade de vendedores registrados.

O processo de reconhecimento facial é inicializado assim que é detectado um rosto, e o converte pixel por pixel da imagem em escala de cinza. Depois, o sistema percorre uma matriz do tamanho da altura por largura com duas estruturas de repetições, em relação a imagem analisada, separa cada pixel em tamanho 3 por 3 e após isso, aplica o método de “*Threshold*”, no qual é um método de segmentação de imagens, que cria uma imagem binária a partir da imagem em tons de cinza.

No interior dessas duas estruturas de repetições, há mais duas responsáveis por realizar a conversão dos valores em binário para decimal, e define um novo valor decimal central para a imagem e depois realiza a extração dos histogramas. Pode-se afirmar que o algoritmo de reconhecimento facial apresenta uma complexidade de $O(n^4)$, no qual é influenciado exponencialmente em relação ao tamanho de “n”, que seria a altura e largura da imagem que o algoritmo analisa.

O algoritmo árvore de decisão tem o seu pior caso no processo de encontrar a predição, ou seja, inserção com os parâmetros e o retorno de uma sugestão, no qual envolve interações com duas estruturas de repetições juntas, ao final resulta em uma complexidade de $O(n) + O(n^2)$, logo, prevalece $O(n^2)$, sendo este o grau de complexidade desse algoritmo.

Para a indicação de um melhor vendedor, é utilizado o algoritmo de *matchmaking*, que realiza diversos cálculos, com seis estruturas de repetições, de maneira sequenciais. Portanto, o algoritmo é influenciado linearmente em relação à quantidade de vendedores, que precisa analisar todos os vendedores para sugerir o mais recomendado para um determinado cliente, onde totaliza uma complexidade de $O(6n)$, pode-se considerar $O(n)$.

5.3 Dificuldades enfrentadas

No decorrer do processo de planejamento e desenvolvimento deste estudo, o pesquisador precisou enfrentar algumas dificuldades:

- problemas de *Firewall* para executar o *software* na Universidade: o aluno não conseguia utilizar a rede de *Internet* acadêmica para demonstrar o funcionamento do sistema, no qual necessitou recorrer aos funcionários de Núcleo da Tecnologia de Informação e Comunicação (NTIC), da PUC-Campinas; e eles ajudaram a resolver essa questão da rede;
- instalação da câmera de segurança: o pesquisador não tinha experiência com instalação elétrica e nem eletrônica, no qual precisou aprender como instalar a câmera de segurança analógica com um conversor digital no *notebook* e integrar com o algoritmo de reconhecimento facial;
- conversões de imagens: o *software* Venda Mais realiza diversas manipulações com as imagens do rosto de clientes e vendedores, que converte para um determinado formato dependendo onde será armazenada ou utilizada;
- aplicação local para execução do algoritmo de reconhecimento facial: o pesquisador sentiu dificuldade maior do que a esperada, pois precisou implementar a parte de reconhecimento separada do *backend* e utilizar *socket*, *threads* e semáforos para tratar *race conditions*.

5.4 Futuras melhorias

Quando foi realizada a reunião para definições de requisitos com cliente, para o desenvolvimento do Venda Mais, muitas funcionalidades interessantes foram analisadas, entretanto, não tiveram prioridade para serem implementadas, por conta do tempo estipulado para entrega do *software*. A seguir podem ser observados os principais requisitos para o seu aprimoramento:

- hospedar remotamente todo o *software* e o banco de dados: assim não dependeria mais da execução de uma máquina local como servidor e o sistema poderia acessar as informações dos clientes em diversas lojas da mesma empresa;
- mostrar os produtos em estoque no aplicativo dos vendedores: seria interessante para cada vendedor analisar os produtos estocados na interface da aplicação, mostrar imagens para os clientes que estão sendo atendidos, e com isso evitar pegar as mercadorias que não interessam ao cliente.

- utilização de mais câmeras de segurança: o Venda Mais foi focado para loja que tem apenas uma entrada e uma única câmera monitorando a entrada dos clientes, uma possível melhoria seria a integração de mais câmeras. Pois, seria necessário para lojas com mais de uma entrada ou saída, para o sistema validar se o cliente já deixou a loja e se foi atendido, atualizando os vendedores.

Diante do exposto, o estudo mostrou-se proveitoso, e com sua conclusão disponibiliza à loja Campinense uma solução de qualidade, que gera melhor organização entre os vendedores, para realizar um atendimento personalizado e, com isso, potencializa o aumento de vendas da loja. Sendo importante para lojas varejistas que têm como objetivo o aumento de vendas de seus produtos, e também, seu lucro.

REFERÊNCIAS

ANGELINE, R. *et al.* Supermarket Automation with Chatbot and Face Recognition Using IoT and AI. *In: INTERNATIONAL CONFERENCE ON COMMUNICATION AND ELECTRONICS SYSTEMS (ICCES)*, 3rd., 2018, Coimbatore, India. **Anales** [...]. Coimbatore, India: IEEE, 2018. p. 1183-1186. Disponível em: <https://ieeexplore.ieee.org/document/8723978>. Acesso em: 23 nov. 2022.

KHAN, M. *et al.* Face Detection and Recognition Using OpenCV. *INTERNATIONAL CONFERENCE ON COMPUTING, COMMUNICATION, AND INTELLIGENT SYSTEMS (ICCCIS)*, 2019, Greater Noida, India. **Anales** [...]. Greater Noida, India: IEEE, 2019. p. 116-119. Disponível em: <https://ieeexplore.ieee.org/document/8974493>. Acesso em: 23 nov. 2022.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO/IEC 25010:2011**: systems and *software* engineering – systems and *software* quality requirements and evaluation (square) – system and *software* quality models. Geneva, 2011. v. 1. Disponível em: <https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en>. Acesso em: 23 nov. 2022.

JIJO, B. T.; ABDULAZEEZ, A. M. Classification Based on Decision Tree Algorithm for Machine Learning. **Journal of Applied Science and Technology Trends**, v. 2, n. 1, p. 20-28, 2021. Disponível em: <https://iplogger.com/2V17k4>. Acesso em: 23 nov. 2022.

OLMEDA, F. **Towards a statistical physics of dating apps**. Dresden, Alemanha: Max Planck Institute for the Physics of Complex Systems, 2021. Disponível em: <https://iplogger.com/2Vg9k4>. Acesso em: 23 nov. 2022.

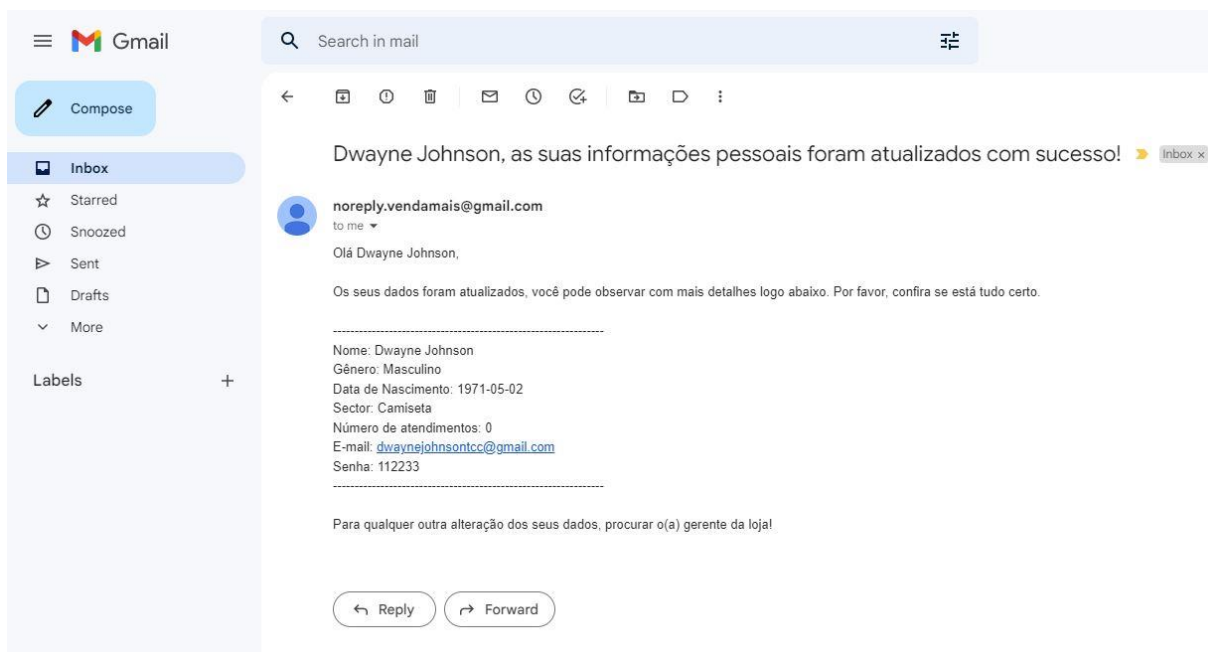
PAGOTTO, T. *et al.* Scrum solo: *software* process for individual development. 2016 *In: IBERIAN CONFERENCE ON INFORMATION SYSTEMS AND TECHNOLOGIES (CISTI)*, 11th, 2016, Gran Canaria, Spain. **Anales** [...]. Gran Canaria, Spain: IEEE, 2016. p. 1-6. Disponível em: <https://ieeexplore.ieee.org/document/7521555>. Acesso em: 23 nov. 2022.

SEMINOTTI, M. A.; RIEDER, R. Recommender Vision: um sistema de recomendação para o comércio baseado em Visão Computacional. *In: WORKSHOP DE TRABALHOS EM ANDAMENTO - CONFERENCE ON GRAPHICS, PATTERNS AND IMAGES (SIBGRAP)*, 33, 2020, Evento Online. **Anais** [...]. Porto Alegre: Sociedade Brasileira de Computação, 2020. p. 147-150. Disponível em: https://sol.sbc.org.br/index.php/sibgrapi_estendido/article/view/12998. Acesso em: 23 nov. 2022.

YADAV, S.; SHUKLA, S. Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification. *In: International Conference on Advanced Computing (IACC)*, 6th., 27-28 fev. 2016, Bhimavaram, India. **Anales** [...]. Bhimavaram, India: IEEE, 2016. Disponível em: <https://ieeexplore.ieee.org/document/7544814>. Acesso em: 23 nov. 2022.

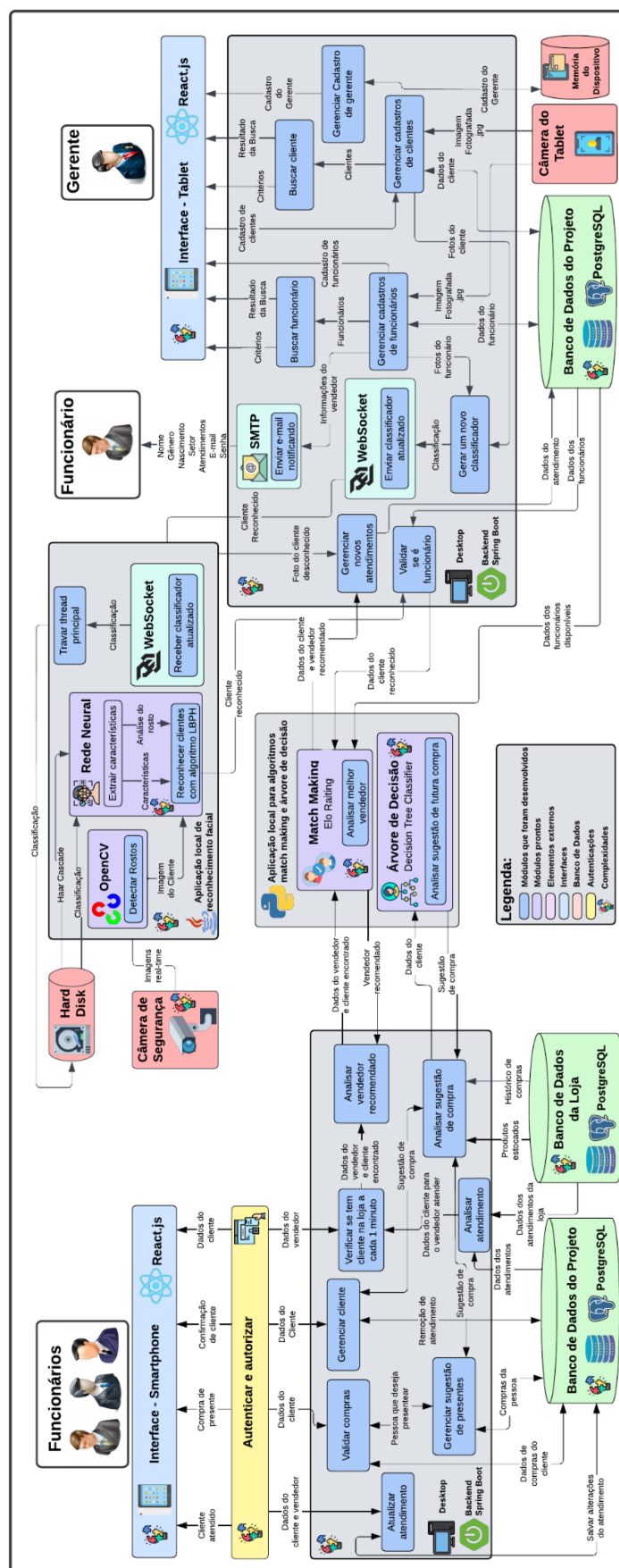
WU, C-C.; ZENG, Y-C.; SHIH, M-J. Enhancing retailer marketing with an facial recognition integrated recommender system. *INTERNATIONAL CONFERENCE ON CONSUMER ELECTRONICS – 2015*, Taipei, Taiwan. **Anales** [...]. Taipei, Taiwan: IEEE, p. 25-26. Disponível em: <https://ieeexplore.ieee.org/document/7216881>. Acesso em: 23 nov. 2022.

APÊNDICE A – NOTIFICA O VENDEDOR CADASTRADO NO SOFTWARE VENDA MAIS

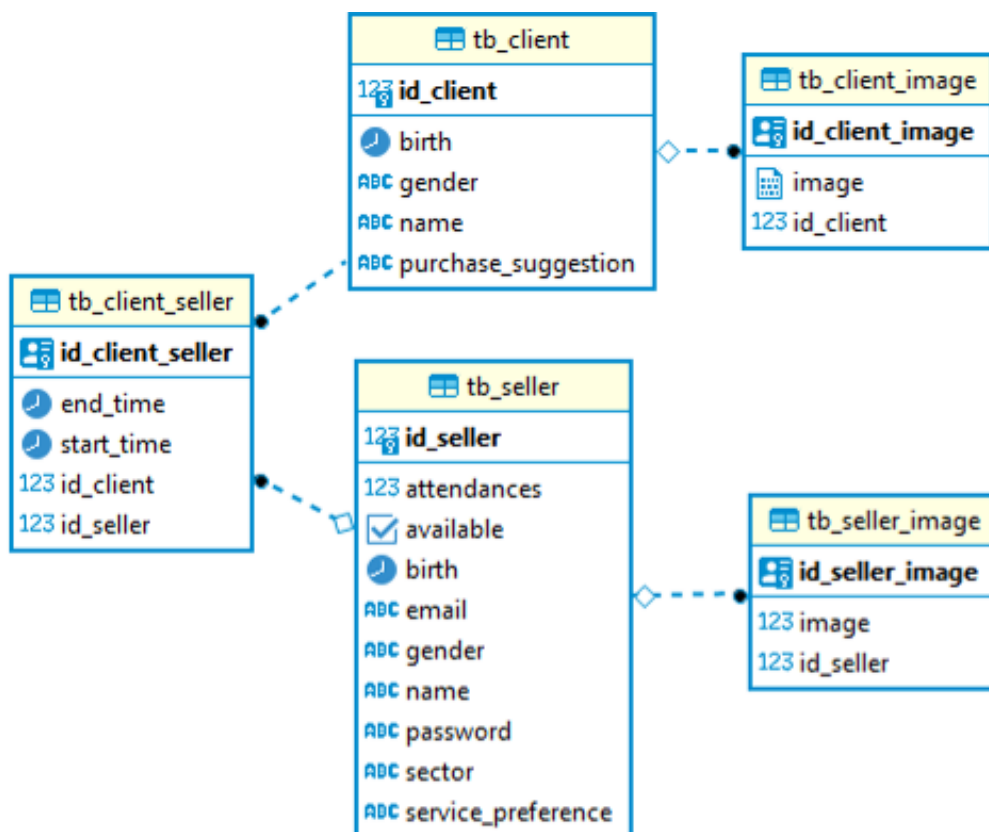


APÊNDICE B – NOTIFICA O VENDEDOR QUE OS SEUS DADOS FORAM ATUALIZADOS

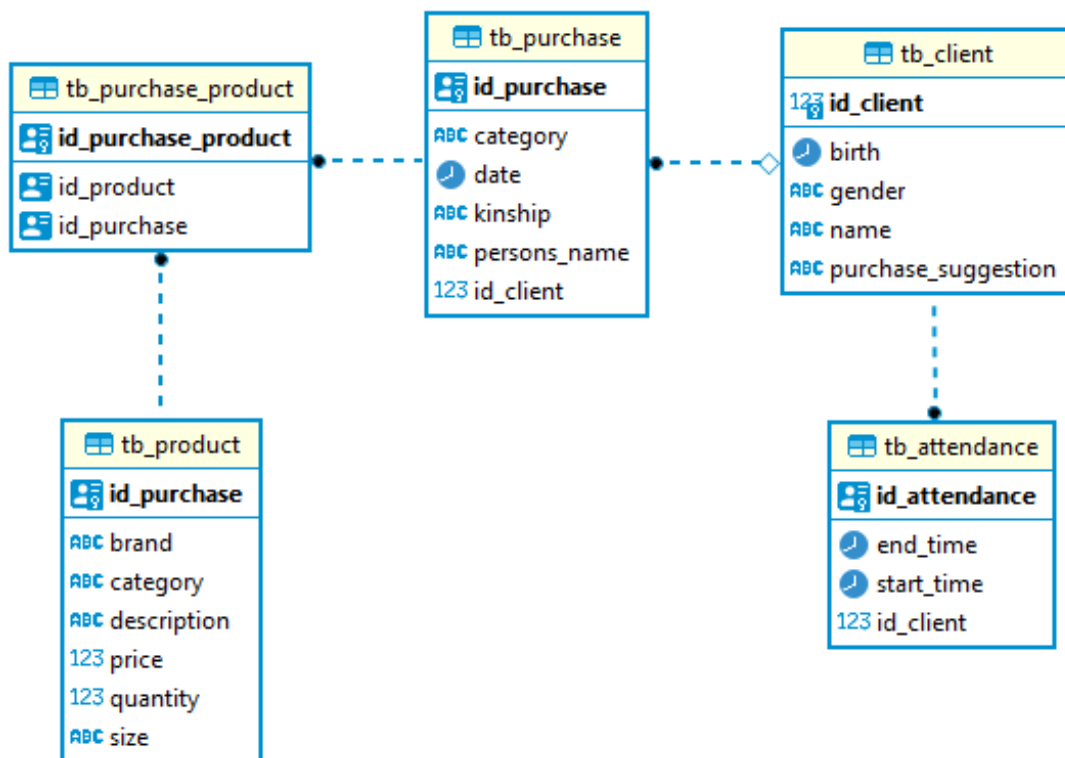




APÊNDICE D – MODELO DE ENTIDADE E RELACIONAMENTO DO BANCO DO VENDA MAIS



APÊNDICE E – MODELO DE ENTIDADE E RELACIONAMENTO DO BANCO DE DADOS DA LOJA



**APÊNDICE F – URL DO REPOSITÓRIO QUE FOI REALIZADO O CONTROLE DE
VERSIONAMENTO DE CÓDIGO DO VENDA MAIS**

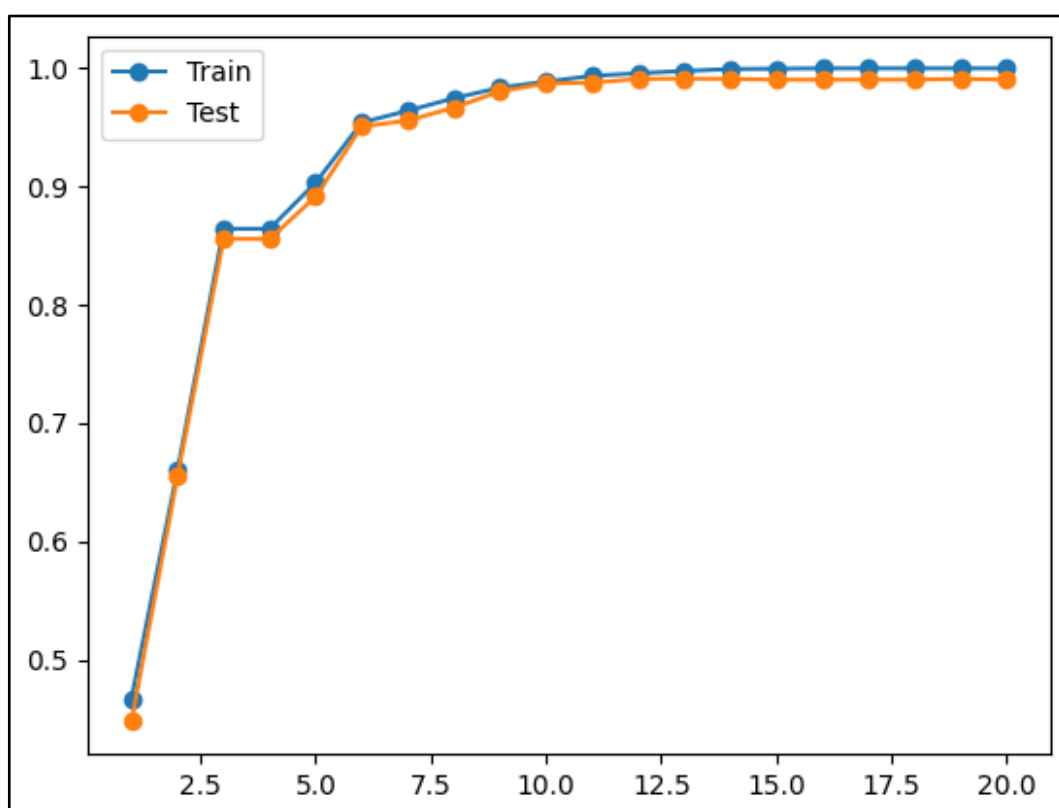
<https://github.com/murilodepa/Undergraduate-Final-Project/>

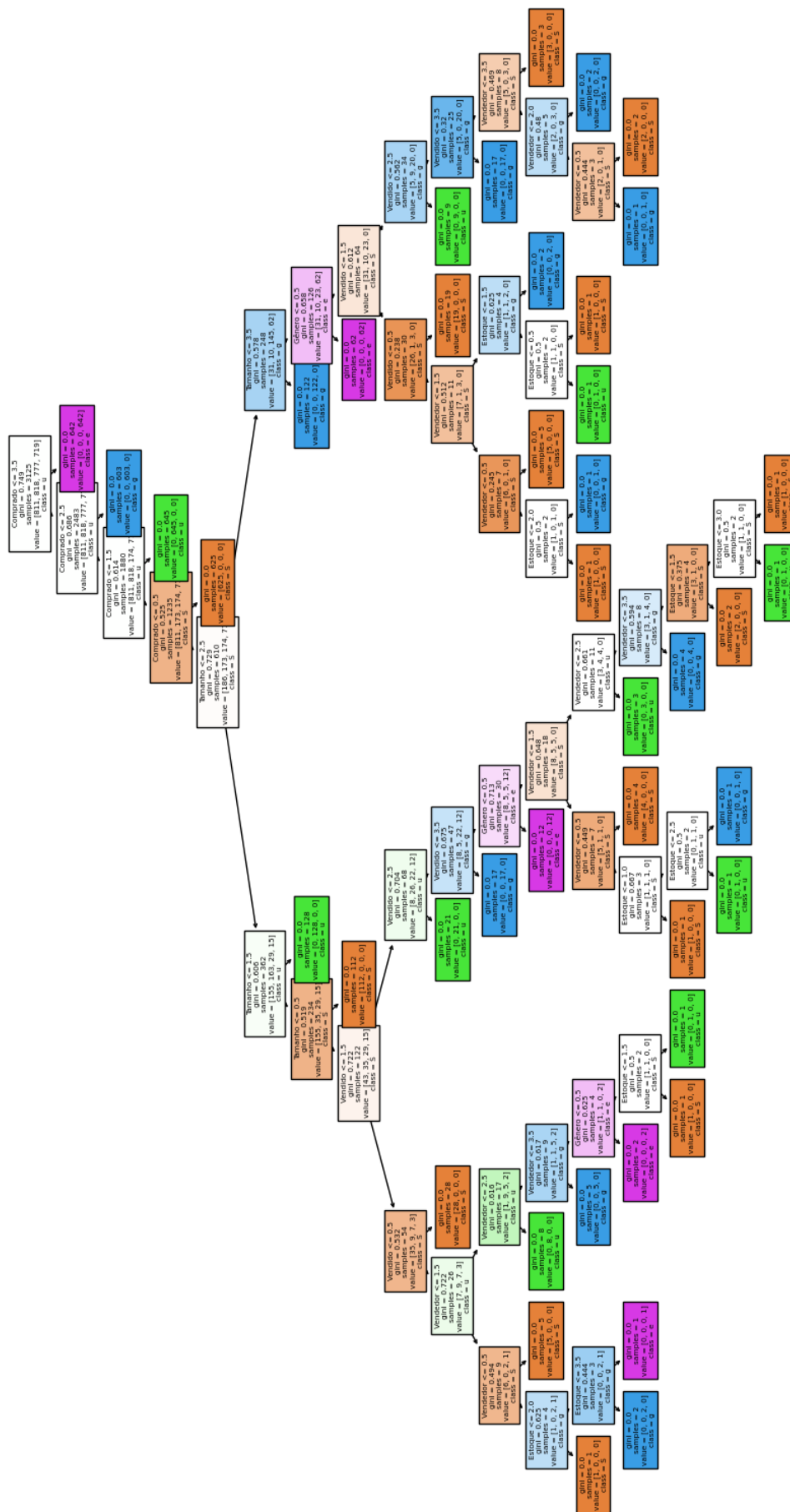
**APÊNDICE G – PESOS DE CADA CATEGORIA UTILIZADA NA FÓRMULA DO
MATCHMAKING**

		Categoria	Peso	Idade	Faixa etária	Peso
Gênero	Peso	Camiseta	-150	0-12	Criança	1,1
Feminino	1,5	Calça	200	12-18	Jovem	1,2
Masculino	0,5	Short	300	18-65	Adulto	0,8
		Vestido	-250	60+	Idosos	0,9

APÊNDICE H – TODAS AS COMBINAÇÕES DE FUNCIONÁRIOS E CLIENTES

Funcionários					Clientes				
-/-	x	y	z	f(x)	-/-	x	y	z	f(x)
Funcionário	Gênero	Setor	Atendimento	Raiting	Cliente	Gênero	Setor	Faxetária	Raiting
1	Feminino	Camiseta	Criança	1227,273	1	Feminino	Camiseta	Criança	1227,273
2	Feminino	Calça	Criança	1545,455	2	Feminino	Calça	Criança	1545,455
3	Feminino	Short	Criança	1636,364	3	Feminino	Short	Criança	1636,364
4	Feminino	Vestido	Criança	1136,364	4	Feminino	Vestido	Criança	1136,364
5	Feminino	Camiseta	Jovem	1125,000	5	Feminino	Camiseta	Jovem	1125,000
6	Feminino	Calça	Jovem	1416,667	6	Feminino	Calça	Jovem	1416,667
7	Feminino	Short	Jovem	1500,000	7	Feminino	Short	Jovem	1500,000
8	Feminino	Vestido	Jovem	1041,667	8	Feminino	Vestido	Jovem	1041,667
9	Feminino	Camiseta	Adulto	1687,500	9	Feminino	Camiseta	Adulto	1687,500
10	Feminino	Calça	Adulto	2125,000	10	Feminino	Calça	Adulto	2125,000
11	Feminino	Short	Adulto	2250,000	11	Feminino	Short	Adulto	2250,000
12	Feminino	Vestido	Adulto	1562,500	12	Feminino	Vestido	Adulto	1562,500
13	Feminino	Camiseta	Idoso	1500,000	13	Feminino	Camiseta	Idoso	1500,000
14	Feminino	Calça	Idoso	1888,889	14	Feminino	Calça	Idoso	1888,889
15	Feminino	Short	Idoso	2000,000	15	Feminino	Short	Idoso	2000,000
16	Feminino	Vestido	Idoso	1388,889	16	Feminino	Vestido	Idoso	1388,889
17	Masculino	Camiseta	Criança	318,182	17	Masculino	Camiseta	Criança	318,182
18	Masculino	Calça	Criança	636,364	18	Masculino	Calça	Criança	636,364
19	Masculino	Short	Criança	727,273	19	Masculino	Short	Criança	727,273
20	Masculino	Vestido	Criança	227,273	20	Masculino	Vestido	Criança	227,273
21	Masculino	Camiseta	Jovem	291,667	21	Masculino	Camiseta	Jovem	291,667
22	Masculino	Calça	Jovem	583,333	22	Masculino	Calça	Jovem	583,333
23	Masculino	Short	Jovem	666,667	23	Masculino	Short	Jovem	666,667
24	Masculino	Vestido	Jovem	208,333	24	Masculino	Vestido	Jovem	208,333
25	Masculino	Camiseta	Adulto	437,500	25	Masculino	Camiseta	Adulto	437,500
26	Masculino	Calça	Adulto	875,000	26	Masculino	Calça	Adulto	875,000
27	Masculino	Short	Adulto	1000,000	27	Masculino	Short	Adulto	1000,000
28	Masculino	Vestido	Adulto	312,500	28	Masculino	Vestido	Adulto	312,500
29	Masculino	Camiseta	Idoso	388,889	29	Masculino	Camiseta	Idoso	388,889
30	Masculino	Calça	Idoso	777,778	30	Masculino	Calça	Idoso	777,778
31	Masculino	Short	Idoso	888,889	31	Masculino	Short	Idoso	888,889
32	Masculino	Vestido	Idoso	277,778	32	Masculino	Vestido	Idoso	277,778

APÊNDICE I – CURVA DE APRENDIZAGEM DA ÁRVORE DE DECISÃO



**APÊNDICE K – URL DA PASTA COMPARTILHADA QUE CONTÉM OS ANEXOS
ELABORADOS PARA AVALIAÇÃO E VALIDAÇÃO DO ESTUDO**

<https://drive.google.com/drive/u/0/folders/1SJUxZO8gfLJ7HKXMmrrpGIF1VffYTv77>