

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO

SISTEMA PARA GERENCIAMENTO DE DISTRIBUIDORAS
DE SORVETE

GUSTAVO EGER

BLUMENAU
2018

GUSTAVO EGER

SISTEMA PARA GERENCIAMENTO DE DISTRIBUIDORAS

DE SORVETE

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Sistemas de Informação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Profa. Luciana Pereira de Araújo Kohler, Mestre - Orientadora

**BLUMENAU
2018**

SISTEMA PARA GERENCIAMENTO DE DISTRIBUIDORAS DE SORVETE

Por

GUSTAVO EGER

Trabalho de Conclusão de Curso aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II pela banca examinadora formada por:

Presidente:

Profa. Luciana Pereira de Araújo Kohler, Mestre – Orientadora, FURB

Membro:

Profa. Simone Erbs da Costa, Especialista – FURB

Membro:

Prof. Francisco Adell Péricas, Mestre – FURB

Blumenau, 12 de julho de 2018

Dedico este trabalho à minha família e amigos que me apoiaram do início ao fim deste trabalho.

AGRADECIMENTOS

Aos meus pais pelo apoio que sempre tive e pelo incentivo de nunca desistir.

Aos meus colegas de faculdade, pela amizade e companheirismo durante estes anos em que convivemos.

A minha orientadora Prof.^a Luciana Pereira de Araújo Kohler, por seu apoio, interesse e colaboração com este trabalho.

Ama-se mais o que se conquista com esforço.

Benjamim Disraeli

RESUMO

Este trabalho apresenta o desenvolvimento de um sistema web e um aplicativo móvel que permite a gestão de vendas de sorvetes por ambulantes. O sistema web visa permitir o controle da venda dos produtos para o gestor do estabelecimento. O aplicativo móvel, por sua vez, tem como objetivo demonstrar quais produtos o ambulante tem para vender, além de mostrar os valores de suas viagens. O sistema web foi desenvolvido com a linguagem PHP utilizando o SQL Server como banco de dados e o *framework* Bootstrap para a realização do *front-end*. O aplicativo móvel foi desenvolvido fazendo uso do *framework* Apache Cordova que é utilizado para o desenvolvimento de aplicativos híbridos para dispositivos móveis. Por fim, é apresentada a análise dos resultados obtidos com testes de performance e compatibilidade. Para os testes de compatibilidade foi utilizada a ferramenta Browsera comprovando que o sistema web é acessível em todos browsers. Através da ferramenta Google PageSpeed Insight foi possível identificar sugestões de melhorias de desempenho.

Palavras-chave: Gestão de vendas de sorvetes. Desenvolvimento híbrido. Apache Cordova.

ABSTRACT

This work presents the development of a web system and a mobile application that allows the management of ice cream sales by street vendors. The web system aims to allow the control of the products sale to the manager of the establishment. The mobile application in turn aims to demonstrate what products it has to sell besides showing the values of its travels. The web system was developed using the PHP language using SQL Server as the database and the Bootstrap framework for the front end. The mobile application was developed using the Apache Cordova framework that is used for the development of hybrid applications for mobile devices. Finally, it is presented the analysis of the results obtained with tests of performance and compatibility. For the compatibility tests, the Browsera tool was used, proving that the web system is accessible in all browsers. Through the Google PageSpeed Insight tool, it was possible to identify suggestions for performance improvements.

Key-words: Sales management of ice cream. Hybrid development. Apache Cordova.

LISTA DE FIGURAS

Figura 1 – Comparativo dos diferentes tipos de aplicações	17
Figura 2 – Visualização de produtos	18
Figura 3 – Tela de pedidos	19
Figura 4 – Mapa de ambulantes.....	20
Figura 5 – Solicitação de venda.....	20
Figura 6 – Tela de pagamento	21
Figura 7 – Tela de solicitação de ambulante	22
Figura 8 – Manutenção de Estoque	23
Figura 9 – Diagrama de Casos de Uso do sistema web.....	27
Figura 10 – Diagrama de casos de uso do aplicativo móvel	28
Figura 11 – Diagrama de pacotes	29
Figura 12 – Modelo Entidade Relacionamento	30
Figura 13 – Arquitetura do sistema web.....	32
Figura 14 – Diagrama de atividades	38
Figura 15 – Tela de listagem de viagens	39
Figura 16 – Tela de cadastro de viagens.....	39
Figura 17 – Cardápio de produtos	40
Figura 18 – Tela de finalização da viagem.....	41
Figura 19 – Relatório de viagens no aplicativo móvel	42
Figura 20 – Resultado dos testes com a ferramenta Browsera no Google Chrome	43
Figura 21 – Resultado dos testes com a ferramenta Browsera no Internet Explorer.....	44
Figura 22 – Resultado dos testes com a ferramenta Browsera para o Firefox e Safari	44
Figura 23 – Resultado dos testes com a ferramenta Google Speed para a versão mobile.....	45
Figura 24 – Resultado dos testes com a ferramenta Google Speed para a versão desktop	46

LISTA DE QUADROS

Quadro 1 – Matriz de rastreabilidade do sistema web.....	25
Quadro 2 – Requisitos não funcionais do sistema web	25
Quadro 3 – Matriz de rastreabilidade do aplicativo móvel	25
Quadro 4 – Requisitos não funcionais do aplicativo móvel	26
Quadro 5 – Classe ControllerProduto.....	33
Quadro 6 – Classe PersistenciaProduto	34
Quadro 7 – Classe PersistenciaPadrao.....	35
Quadro 8 – Arquivo de configuração do Apache Cordova	36
Quadro 9 – Requisição para a listagem dos produtos.....	37
Quadro 10 – Características dos trabalhos correlatos e do trabalho desenvolvido	46
Quadro 11 – Descrição do caso de uso Movimentar o estoque dos produtos (UC08).....	52
Quadro 12 – Descrição do caso de uso Cadastrar viagem (UC10)	53
Quadro 13 – Descrição do caso de uso Confirma Vendas (UC02)	54
Quadro 14 – Descrição do caso de uso Consultar viagens realizadas (UC03).....	55

LISTA DE ABREVIATURAS E SIGLAS

API – Application Programming Interface

CSS – Cascading Style Sheets

HTML – HyperText Markup Language

JSON – JavaScript Object Notation

MER – Modelo de Entidade Relacionamento

MVC – Model View Controller

PHP – Hypertext Preprocessor

REST – Representational State Transfer

RF – Requisito Funcional

RNF – Requisitos Não Funcionais

REST - Representational State Transfer

SQL – Structured Query Language

UC – Diagrama de Casos de Uso

SUMÁRIO

1 INTRODUÇÃO.....	12
1.1 OBJETIVOS.....	12
1.2 ESTRUTURA.....	13
2 FUNDAMENTAÇÃO TEÓRICA	14
2.1 COTIDIANO DE UM AMBULANTE	14
2.2 CONTROLE DE ESTOQUE	15
2.3 SISTEMAS HÍBRIDOS	16
2.4 TRABALHOS CORRELATOS	17
2.4.1 Salebox.....	18
2.4.2 Ambulante	19
2.4.3 NaPraia.....	21
2.4.4 Sistema de gestão de estoque para distribuidoras de fumo	22
3 DESENVOLVIMENTO	24
3.1 LEVANTAMENTO DE INFORMAÇÕES	24
3.2 ESPECIFICAÇÃO	24
3.2.1 Diagrama de casos de uso do sistema web	26
3.2.2 Diagrama de casos de uso do aplicativo móvel	27
3.2.3 Diagrama de pacotes	28
3.2.4 Modelo Entidade Relacionamento	29
3.3 IMPLEMENTAÇÃO	31
3.3.1 Técnicas e ferramentas utilizadas.....	31
3.3.2 Desenvolvimento.....	32
3.3.3 Operacionalidade da implementação	37
3.4 RESULTADOS E DISCUSSÕES.....	42
4 CONCLUSÕES.....	48
4.1 EXTENSÕES	49
REFERÊNCIAS	50
APÊNDICE A – DESCRIÇÃO DOS CASOS DE USO DO SISTEMA WEB.....	52
APÊNDICE B – DESCRIÇÃO DOS CASOS DE USO DO APLICATIVO MÓVEL.....	54

1 INTRODUÇÃO

Com o intuito de conseguir uma renda extra, ou até mesmo porque estão desempregados, o número de trabalhadores informais cresce cada vez mais no Brasil (PAPP, 2016). Segundo dados da Pesquisa Nacional por Amostra de Domicílios (PNAD), este número chegou a 10 milhões no ano de 2016 (PAPP, 2016). De acordo com a pesquisa da Associação Brasileira das Indústrias e do Setor de Sorvetes, o volume de consumo de sorvetes no Brasil saltou de 685 milhões de litros em 2003 para 1 bilhão em 2016 (ABIS, 2016). Mais de 8 mil empresas do setor de sorvetes, sendo 90% delas micro ou pequenas empresas, são responsáveis por cerca de 80 mil empregos diretos e um faturamento que gira em torno de R\$ 4 bilhões anualmente (RODRIGUES, 2015, p.5).

A beleza das praias e a presença constante de sol e calor nos períodos de verão em algumas regiões do país colaboram para que o Brasil se destaque como destino turístico (SCHERER, 2013). Ainda segundo Scherer (2013), é justamente nestes períodos que cresce o consumo dos sorvetes e as oportunidades para os serviços como ambulante. Porém, para se tornar um vendedor ambulante não se pode apenas sair pela cidade ou pela praia vendendo seus produtos. Cada município possui regras e normas com relação ao comércio ambulante, além de que a maioria das prefeituras possui convênio com distribuidoras de alimentos e bebidas para a distribuição destes produtos (NETO, 2015).

Com um cenário competitivo cada vez mais acirrado, as empresas tendem a recorrer ao uso dos sistemas de informações para garantir maior produtividade e eficácia. Segundo Batista (2004, p. 39), “o objetivo de usar os sistemas de informação é a criação de um ambiente empresarial em que as informações sejam confiáveis e possam fluir na estrutura organizacional.”.

Diante do exposto, é apresentado o desenvolvimento de um protótipo para gerir a venda de sorvetes por ambulantes. Por meio dele, o gestor do estabelecimento terá controle do estoque de seus produtos, além de acompanhar as vendas de seus ambulantes e tomar decisões através dos relatórios de apoio. Ainda, é apresentado o desenvolvimento de um aplicativo móvel no qual o ambulante poderá assegurar que suas vendas não serão desviadas.

1.1 OBJETIVOS

O objetivo deste trabalho é disponibilizar um sistema híbrido que permita a gestão de vendas de sorvetes por ambulantes em praias.

Os objetivos específicos são:

- a) disponibilizar um aplicativo móvel para o ambulante contendo os produtos que ele

irá vender;

- b) desenvolver um sistema web para facilitar o controle dos produtos vendidos pelos gestores dos restaurantes e lanchonetes que vendem sorvetes.

1.2 ESTRUTURA

Este trabalho está dividido em quatro capítulos. O primeiro capítulo é composto pela introdução que relata a inspiração para o desenvolvimento deste, dos objetivos definidos para o mesmo e a apresentação de sua estrutura.

O segundo capítulo apresenta a fundamentação teórica, abordando os conceitos sobre o cotidiano de um ambulante, controle de estoque e os trabalhos correlatos.

No terceiro capítulo é apresentado o desenvolvimento da ferramenta, trazendo os requisitos dos sistemas, os diagramas de casos de uso, diagrama de pacotes, modelo conceitual de dados, além das técnicas e ferramentas utilizadas, operacionalidade da implementação e os resultados e discussões.

Por fim, o quarto capítulo apresenta as conclusões e sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo apresentar os principais temas relacionados com este trabalho. São apresentados o cotidiano de um ambulante, o que é controle de estoque e sistemas híbridos. Além disso na seção 2.4 são apresentados os trabalhos correlatos.

2.1 COTIDIANO DE UM AMBULANTE

Vendedor ambulante caracteriza-se por trabalhar por conta própria em vias e logradouros públicos, portando a devida autorização com determinado prazo de validade (GONÇALVES, 2012). Ainda segundo Gonçalves (2012) em tempos de crise econômica, o comércio informal cresce cada vez mais no Brasil e embora o negócio pareça fácil, para poder oferecer esse tipo de serviço também há burocracia e necessidade de um bom plano de administração.

Um dado relevante a ser considerado no setor do trabalho informal, aonde se encaixam os ambulantes, diz respeito a exploração e a precarização desta força de trabalho. A forma de inserção no trabalho informal é de forma precária e caracteriza-se por uma renda bastante baixa, além de não garantir os direitos trabalhistas básicos como, tais como a aposentadoria, o Fundo de Garantia do Tempo de Serviço (FGTS) e demais auxílios (FERNANDES, 2008). Fernandes (2008) ainda salienta que os trabalhadores deste setor, em sua maioria, não possuem horário de trabalho fixo, sendo assim quase sempre acabam aumentando sua jornada de trabalho, possibilitando produzir mais nas horas vagas.

Segundo Mallaguti (2000 apud FERNANDES, 2008) o setor formal e o informal possuem dependência mútua, passando a existir trabalhadores informais no setor formal e trabalhadores formais no setor informal. Porém o que é do conhecimento de todos é que os trabalhadores do setor informal estão por toda parte, constituem um cenário que cresce diariamente chamando a atenção da sociedade e gerando discussões polêmicas.

Para se obter sucesso no ramo, algumas práticas são importantes, entre elas um bom atendimento ao cliente e trabalhar rápido satisfazendo as expectativas deles. Os empreendedores do ramo de ambulante devem expor seu produto com qualidade e enfatizar que estão sempre dispostos a solucionar o problema do cliente (GONÇALVES, 2012).

O comércio ambulante utiliza quiosques, bancas, camelôs e estruturas parecidas como sede de seus serviços. O lugar aonde o ambulante poderá comercializar seus produtos depende também da prefeitura, onde é necessário retirar uma licença para poder realizar este tipo de produto que é um dos maiores problemas atuais de trabalhadores deste ramo. Além disto,

correm riscos diários, como por exemplo atropelamentos, exposição em excesso ao sol, violência verbal e física (GONÇALVES, 2012).

2.2 CONTROLE DE ESTOQUE

A estocagem é uma prática mantida no processo de armazenamento de produtos e é fundamental na manutenção da organização da distribuidora, bem como no aumento da produção operacional. A administração dos estoques deve ser muito bem avaliada, uma vez que podem ser armazenadas em diferentes etapas do processo, apresentando características diversas (POZO, 2008).

Moura (2004 apud SANTANA, 2014) define estoque como um conjunto de bens armazenados com características e funções próprias que atendem aos objetivos e necessidade de uma determinada empresa. Ainda salienta que os estoques proporcionam as organizações a possibilidade da produção em massa e contínua, e estão presentes em nosso cotidiano em inúmeros lugares como, por exemplo, nas fábricas, escritórios e comércio.

O controle de estoque foi criado a partir da necessidade de controlar e organizar os estoques. Antigamente era feito por método manual, com fichas de prateleiras e de controle. Hoje em dia, com o constante crescimento da tecnologia e da informática, tornou-se mais fácil e rápido controlar os estoques (RAIMUNDO, 2011).

De acordo com Pozo (2008), o papel principal do controle de estoque, dentro da logística, é assegurar os mais variados níveis de produtos e materiais que a empresa deve manter dentro dos padrões econômicos, visando ter um custo relativo ao que se vende e se tem em estoque. Além disso, administrar um estoque é também balancear a disponibilidade dos produtos ou serviços ao consumidor, com os custos de abastecimento que são necessários para atender a essa disponibilidade (POZO, 2008).

Já para Pimenta (2003), o controle do estoque é um elemento básico e fundamental em todas as etapas dos sistemas produtivos, desde o planejamento da produção até a expedição de seu produto. Ainda salienta que o processo de estocagem visa dar ao gestor total conhecimento de todas as etapas de sua empresa, desde o planejamento de compras até a venda do seu produto, de forma a otimizar o investimento no estoque, diminuindo as necessidades de capital investido bem como evitando a falta da matéria-prima.

Ao focar o objetivo de aumentar a lucratividade e diminuir custos, as empresas estão, cada vez mais, buscando meios de desenvolver suas áreas através da melhoria de seus processos (OLIVEIRA, 2016). Neste contexto, Viana (2002 apud OLIVEIRA, 2016) cita que

a gestão do estoque passa a ser um fator crítico para as organizações, pois entende-se que os estoques desnecessários requerem um alto nível de investimento.

O estoque geralmente é gerenciado por meio de sistemas de informações computadorizados, de forma que atualizem os registros de estoque e pedido, gere relatórios de situação e de previsão de demanda. Estes sistemas são completamente dependentes da manutenção frequente das informações que ali são inseridas. A falta de apuração dos dados do estoque é prejudicial para a gestão da empresa, já que pode comprometer o fluxo do sistema. Se o sistema acusa que um determinado produto está disponível no estoque da empresa, mas ele não está, no momento de despachar este produto pode-se não haver mais tempo para corrigir o problema. Estes fatores aumentam a necessidade do aperfeiçoamento do controle de estoque e um administrador bem preparado para gerenciar-lo (SLACK; CHAMBERS; JOHNSTON, 2009).

2.3 SISTEMAS HÍBRIDOS

O conceito de sistemas híbridos é bastante amplo e pode englobar diversos tipos de abordagens (OSÓRIO; VIEIRA, 1999). De uma forma geral, conforme Osório e Vieira (1999), pode-se afirmar que sistema híbrido é todo sistema que integra dois ou mais métodos para a resolução de um problema. Os autores ainda destacam que o uso de sistemas híbridos é a integração de diversos componentes que interagem entre si, cada um com suas características, visando atingir um objetivo comum. Segundo Osório e Vieira (1999, p. 33), as principais vantagens do uso dos sistemas híbridos são:

- a) a integração de duas técnicas complementares permite que uma complete as deficiências da outra de forma a obter um melhor desempenho;
- b) ao usarmos diferentes técnicas de aquisição e representação de conhecimento amplia-se a capacidade do sistema de adquirir novas informações;
- c) tarefas complexas podem ser quebradas em subproblemas, e assim tratadas de forma individual, chegando em uma solução para o problema em geral;
- d) o processamento em paralelo das informações irá permitir um sistema com maior performance e maior tolerância a falhas.

O conceito também se aplica ao desenvolvimento de aplicativos multiplataformas. Estes aplicativos possuem como finalidade funcionar em qualquer dispositivo, sendo que para as diferentes plataformas, será utilizado o mesmo código-fonte (PREZOTTO; BONIATI, 2014). Os aplicativos são desenvolvidos com tecnologias web como HTML5 e JavaScript e um *container* nativo que permite acessar os recursos do dispositivo. Com a utilização das

tecnologias web junto aos recursos nativos das plataformas, o desenvolvimento fica mais prático e econômico pelo fato de poder ser compilado para várias plataformas, além do CSS proporcionar customizações ilimitados com o *front-end* da aplicação (MOURA, 2016).

A Figura 1 traz uma comparação realizada por Prezotto e Boniati (2014), demonstrando quais quesitos cada tipo de aplicação oferece suporte. Aplicações nativas e híbridas podem ser disponibilizadas através das lojas de aplicativos, uma vez que elas ficam instaladas no aparelho, ao contrário de uma *web app* que nada mais é que um site para dispositivos móveis. As aplicações híbridas se destacam pois, em geral, independem da plataforma escolhida.

Figura 1 – Comparativo dos diferentes tipos de aplicações

	Acesso ao Dispositivo	Performance	Tempo Desenvolvimento	App Store	Multi Plataforma
Nativo	Sim	Sim	Caro	Sim	Não
Web	Parcial	Sim*	Ótimo	Não	Sim
Híbrido	Sim	Sim*	Ótimo*	Sim	Sim*

Fonte: Prezotto e Boniati (2014).

A solução mais comum atualmente para a construção de aplicativos multiplataforma é o Cordova, que utiliza linguagens padronizadas web para a construção dos aplicativos (LOPES, 2016). O Cordova permite que um aplicativo desenvolvido em HTML execute de forma nativa em Android e iOS, contudo não consegue imitar a usabilidade e aparência dos dispositivos nativos. Além disso, o Cordova possui *plug-ins* que permitem acessar funcionalidades específicas do aparelho em que o aplicativo sendo é executado. Esses devem ser instalados diretamente no projeto do aplicativo (BEZERRA; SCHIMIGUEL, 2016).

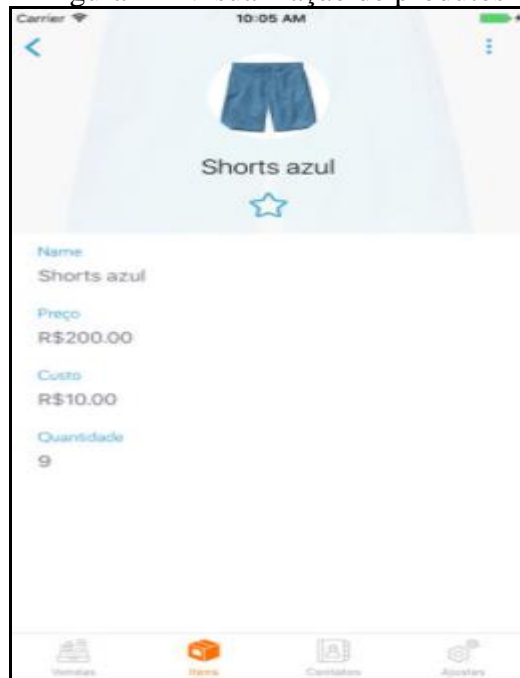
2.4 TRABALHOS CORRELATOS

A seguir são apresentados quatro trabalhos correlatos, com características e objetivos semelhantes ao presente trabalho. A subseção 2.4.1 detalha a aplicação Salebox, que tem como objetivo o gerenciamento de vendas. A subseção 2.4.2 traz o aplicativo Ambulante, desenvolvido pela empresa Ideia no Ar. A subseção 2.4.3 contempla o aplicativo NaPraia, desenvolvido com objetivo de conectar praieiros e ambulantes. Na subseção 2.4.4 é apresentado o trabalho de conclusão de curso de Luiz (2011), que consiste de um sistema para controle de estoque de uma distribuidora de fumo.

2.4.1 Salebox

A aplicação Salebox possui como objetivo principal o controle de vendas diretamente no celular. Também visa gerenciar contatos e catálogos de qualquer lugar, podendo compartilhar informações entre vários aparelhos (SILVA, 2016). Com o Salebox, pode-se cadastrar e editar seus clientes, podendo incluir todos os produtos que se deseja vender com fotos, informando quanto que há em estoque e outras informações importantes sobre o mesmo (SILVA, 2016). Na Figura 2 é ilustrada a tela de detalhes de um produto cadastrado, onde além da foto é apresentado ao usuário o nome e preço dos produtos, assim como a quantidade disponível para venda.

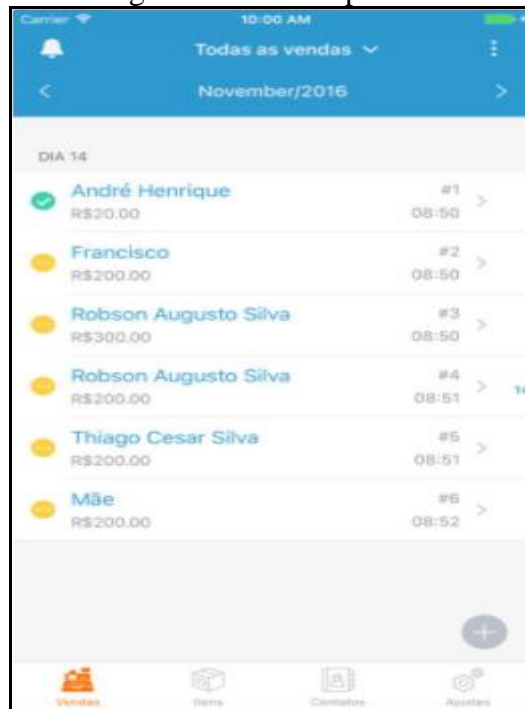
Figura 2 – Visualização de produtos



Fonte: Silva (2016).

A Figura 3 detalha a tela de vendas, na qual todas as vendas feitas pelo usuário são listadas. Para visualizar os detalhes da venda, pode-se clicar sobre o item afim de visualizar os itens comprados naquela venda. Todas as informações ficam armazenadas no aparelho móvel, ou seja, não precisa de internet para poder utilizá-lo. O aplicativo possui sistema de busca de transações, clientes e produtos, organizados em uma interface fácil de utilizar seguindo os padrões de usabilidade e design indicados pela Google (SILVA, 2016).

Figura 3 – Tela de pedidos



Fonte: Silva (2016).

2.4.2 Ambulante

Desenvolvido pela empresa Ideia no Ar, o software ambulante visa localizar produtos oferecidos por ambulantes, podendo ser filtrados por região e tipo de produto desejado (NO AR, 2016). O aplicativo pode ser utilizado tanto pelo cliente, quanto pelo ambulante, pois além de localizar os tipos de produtos comercializados, o cliente pode chamar o ambulante até ele, podendo ainda pagar a compra via aplicativo com cartão de crédito. Por sua vez, o ambulante pode receber o pagamento pelo aplicativo, evitando andar com dinheiro pelas ruas, além de não precisar alugar máquinas de cartão de crédito (NO AR, 2016).

Ao entrar no aplicativo, o cliente escolhe qual o produto que deseja comprar, e conforme a Figura 4, são apresentados no mapa os vendedores disponíveis na região. Ao clicar sobre um deles são apresentados quais produtos está vendendo. Ao ser solicitado, o ambulante poderá aceitar ou não ir até o cliente, podendo verificar qual a distância entre eles, como é ilustrado na Figura 5.

Figura 4 – Mapa de ambulantes



Fonte: No ar (2016).

Figura 5 – Solicitação de venda



Fonte: No ar (2016).

2.4.3 NaPraia

Conforme NaPraia (2016), o objetivo principal do aplicativo NaPraia é facilitar o comércio nas praias, tanto para os clientes quanto para os vendedores. O cliente compra o produto direto pelo smartphone, sendo que o pagamento é feito no aplicativo, por cartão de crédito, e o vendedor vem até a pessoa e entrega o pedido. O aplicativo, disponível para Android e iOS, mostra tudo o que os ambulantes podem vender, permite comparar preços, fazer reservas e pagar tudo eletronicamente (NAPRAIA, 2016).

Um diferencial do aplicativo é a avaliação da qualidade dos serviços prestados e produtos oferecidos. Conforme a Figura 6, ao finalizar um pagamento o cliente pode avaliar o ambulante, classificando-o entre 1 e 5 estrelas. Essa informação estará visível para os outros clientes. Com isso, outros usuários do aplicativo, ao acessar o perfil do vendedor, poderão decidir se irão chama-lo ou não, como mostra a Figura 7.

Figura 6 – Tela de pagamento



Fonte: NaPraia (2016).

Figura 7 – Tela de solicitação de ambulante



Fonte: NaPraia (2016).

2.4.4 Sistema de gestão de estoque para distribuidoras de fumo

O trabalho de Luiz (2011) tem como objetivo apresentar o desenvolvimento de um sistema web para o controle de estoques de distribuidoras de fumo. Este permite controlar a entrada e saída de materiais para assim manter informações sobre estoques e vendas. O sistema foi desenvolvido com auxílio da ferramenta e-Gen, que permite gerar aplicações em páginas JavaServer Pages (JSP). O banco de dados utilizado foi o MySQL, fazendo uso do DBDesigner 4.

A aplicação disponibiliza o cadastro da classificação e subgrupos de fumos, para após, com estas informações poder ser cadastrado um novo produto. Na Figura 8 tem-se a tela da manutenção de estoque. Na mesma é informado o produto, a quantidade que atual em estoque, a quantidade mínima e máxima que se pode ter no estoque para o produto que está sendo incluído.

Figura 8 – Manutenção de Estoque

s-Gen Developer 06/07/2011 10:34:14

Cadastro de Estoque

Todos os campos com ▼ devem ser preenchidos.

Código

Quantidade

Quantidade Mínima

Quantidade Máxima

◀ ◁ ▷ ▶ Total:

Produtos em Estoque

Código Produto	Quantidade	Quantidade Mínima	Quantidade Máxima
2 Virginia	19,00	2,00	30,00
3 Burley	4,00	1,00	10,00
4 Comum	20,00	2,00	19,00

Menu | Sair | Imprimir

Fonte: Luiz (2011).

3 DESENVOLVIMENTO

Neste capítulo é apresentado o detalhamento do desenvolvimento do sistema assim como suas especificações. A primeira seção apresenta um levantamento das informações. A segunda seção especifica o sistema desenvolvido, apresentando os Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF) do sistema web assim como do aplicativo móvel. Nesta seção ainda é apresentado o diagrama de pacotes, diagramada de atividades e o Modelo de Entidade Relacionamento (MER) do sistema. A terceira seção demonstra as técnicas e ferramentas utilizadas no desenvolvimento do trabalho, assim como trechos de códigos-fontes da implementação e a operacionalidade do sistema. Por fim, a quarta seção apresenta os resultados obtidos com o desenvolvimento deste trabalho.

3.1 LEVANTAMENTO DE INFORMAÇÕES

Este trabalho apresenta o desenvolvimento de um sistema para gestão de venda de picolés por ambulantes utilizando os conceitos de sistemas híbridos. O sistema é composto por duas partes. A primeira é um sistema web para uso dos gestores dos estabelecimentos que fornecem produtos aos ambulantes para as vendas. A segunda é um aplicativo móvel para uso dos próprios ambulantes, de modo que possam realizar suas vendas e acompanhar os valores de suas viagens.

O sistema web permite ao gestor do estabelecimento gerir suas marcas, produtos, destinos de vendas e a movimentação do estoque dos produtos. No momento em que o ambulante chega na loja para a retirada dos produtos para a venda é cadastrado uma viagem, informando quais produtos que serão levados e a quantidade dos mesmos. Com isso, o sistema já realiza um cálculo do valor que se espera receber do ambulante ao final da venda.

Ao fazer *login* no aplicativo móvel, o ambulante visualiza um cardápio com todos os produtos que tem disponível para a venda. Conforme ele realiza a venda dos mesmos, o relatório com informações das viagens em andamento é atualizado. Dessa forma, o ambulante consegue ter noção do quanto já vendeu e qual o valor que terá que retornar para o gestor do estabelecimento.

3.2 ESPECIFICAÇÃO

Na seção a seguir é apresentada a especificação do sistema, contendo os requisitos funcionais e não funcionais tanto do sistema web quanto do aplicativo móvel, além dos diagramas de casos de uso, diagrama de pacotes, diagrama de atividade e o Modelo de Entidade Relacionamento (MER). Para o desenvolvimento dos diagramas foi utilizada a

ferramenta Astah Community e para o desenvolvimento do MER foi utilizado o MySQL Workbench.

O Quadro 1 apresenta os Requisitos Funcionais (RF) do sistema web e sua rastreabilidade, ou seja, vinculação com o Caso de Uso (UC) associado. Os UCs associados neste quadro encontram-se no diagrama da Figura 9.

Quadro 1 – Matriz de rastreabilidade do sistema web

Requisitos Funcionais	Caso de Uso
RF01: O sistema web deve permitir manter lojas.	UC01
RF02: O sistema web deve permitir cadastrar os gestores.	UC02
RF03: O sistema web deve permitir manter ambulantes.	UC03
RF04: O sistema web deve permitir manter marcas.	UC04
RF05: O sistema web deve permitir manter produtos.	UC05
RF06: O sistema web deve permitir manter usuários.	UC06
RF07: O sistema web deve permitir manter destinos.	UC07
RF08: O sistema web deve permitir a movimentação de entrada e saída de produtos	UC08
RF09: O sistema web deve permitir o cadastro de viagens.	UC09
RF10: O sistema web deve permitir a finalização de viagens.	UC10
RF11: O sistema web deve permitir emitir relatórios gerenciais.	UC11, UC12, UC13

Fonte: elaborado pelo autor.

O Quadro 2 apresenta os Requisitos Não Funcionais (RNF) do sistema web.

Quadro 2 – Requisitos não funcionais do sistema web

Requisitos Não Funcionais
RNF01: O sistema web deverá ser desenvolvido utilizando a linguagem de programação PHP.
RNF02: O sistema web deverá se comunicar com o banco de dados MySQL.
RNF03: O sistema web só permitirá acesso após efetuar <i>login</i> com usuário e senha.
RNF04: O sistema web deve possuir interface simples e eficiente.

Fonte: elaborado pelo autor.

O Quadro 3 apresenta os Requisitos Funcionais (RF) do aplicativo móvel e sua rastreabilidade, ou seja, vinculação com o Caso de Uso (UC) associado. Os UCs associados neste quadro encontram-se no diagrama da Figura 10.

Quadro 3 – Matriz de rastreabilidade do aplicativo móvel

Requisitos Funcionais	Caso de Uso
RF01: O aplicativo deve permitir o ambulante efetuar <i>login</i> .	UC01
RF02: O aplicativo deve apresentar um cardápio digital.	UC02
RF03: O aplicativo deve permitir a venda de produtos em tempo real.	UC03
RF04: O aplicativo deve emitir relatório de viagens realizadas.	UC04
RF05: O aplicativo deve apresentar os valores das viagens em andamento.	UC05

Fonte: elaborado pelo autor.

O Quadro 4 apresenta os Requisitos Não Funcionais (RNF) do aplicativo móvel.

Quadro 4 – Requisitos não funcionais do aplicativo móvel

Requisitos Não Funcionais
RNF01: O aplicativo deve ser construído utilizando o <i>framework</i> Cordova.
RNF02: O aplicativo só permitirá acesso após efetuar <i>login</i> com usuário e senha.
RNF03: O tempo de espera nas consultas não deve exceder 30 segundos.

Fonte: elaborado pelo autor.

3.2.1 Diagrama de casos de uso do sistema web

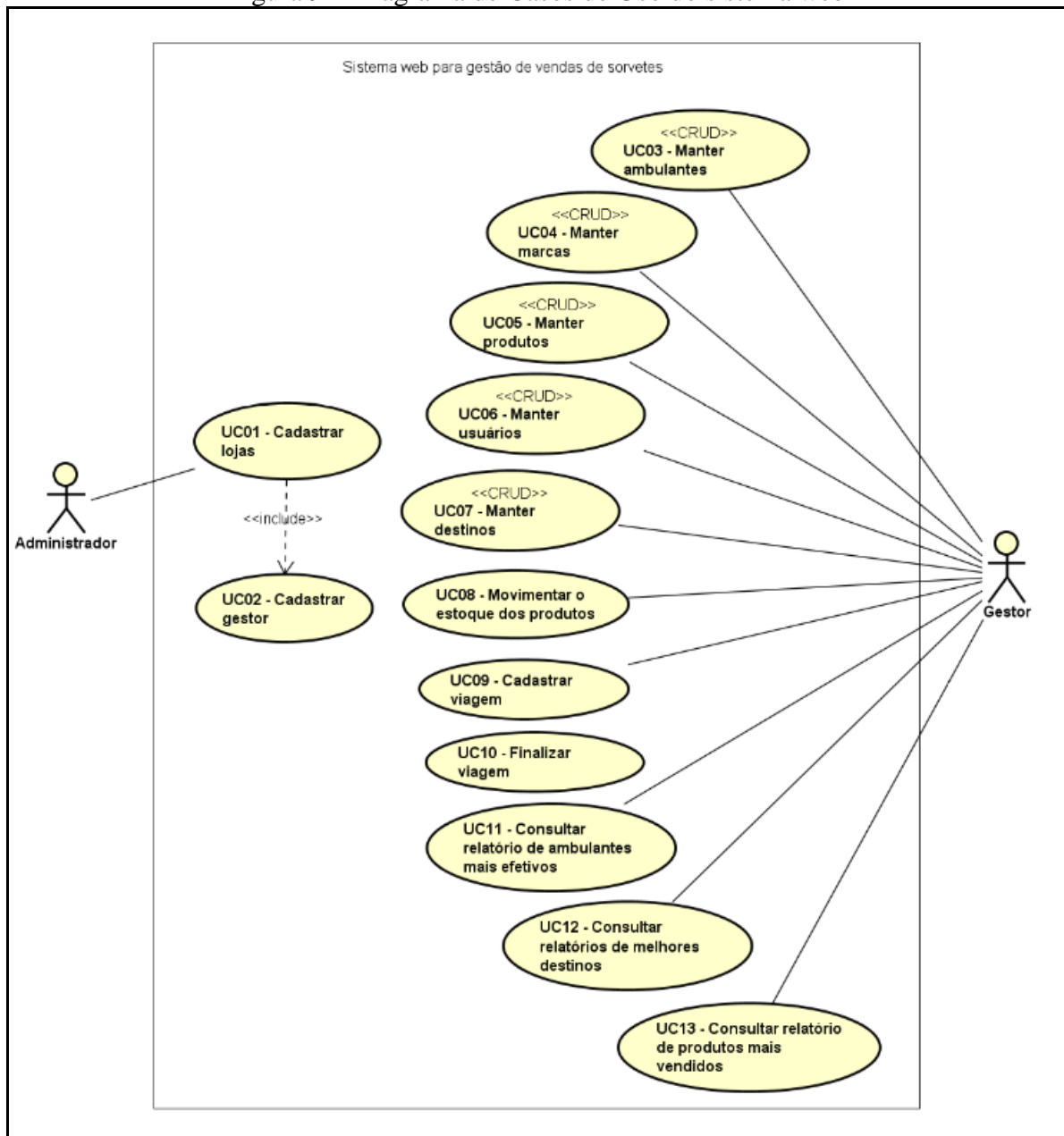
A Figura 9 apresenta o digrama de casos de uso do sistema web desenvolvido. No diagrama são apresentados os atores que persistem no sistema desenvolvido, sendo eles: *administrador* e *gestor*. O detalhamento dos principais casos uso do sistema web encontram-se no Apêndice A.

O ator *administrador* é quem realiza o cadastro das lojas através do UC01 - Cadastrar lojas, juntamente com o cadastro inicial do gestor do estabelecimento através do UC02 - Cadastrar gestor. Enquanto o ator *gestor* é quem tem maior interação com o sistema web, pois ele quem irá manter a sua loja atualizada. Ele pode manter suas marcas, produtos, destinos, usuários e ambulantes. O UC08 - Movimentar o estoque dos produtos permite cadastrar uma entrada ou saída do produto do estoque. Caso o tipo da movimentação for uma saída, o sistema valida para que a quantidade seja menor do que a quantidade em estoque, evitando erros na contabilização do estoque.

Quando um *ambulante* chega ao estabelecimento para retirada dos produtos para vender o *gestor* realiza o cadastro de uma viagem por meio do caso de uso UC09 - Cadastrar viagem. Ao fim das vendas o *ambulante* retorna ao estabelecimento e caso os valores da viagem forem validados, o *gestor* realiza a finalização da viagem através do caso de uso UC10 - Finalizar viagem.

Conforme as viagens vão sendo realizadas e finalizadas os dados dos relatórios vão sendo incrementados. O UC11 - Consultar relatório de ambulantes mais efetivos disponibiliza ao *gestor* um relatório listando os ambulantes e a quantidade de produtos vendida por cada um. Com o UC12 - Consultar relatório de melhores destinos pode-se consultar os destinos em que o número de vendas é maior. Por sua vez o UC13 - Consultar relatório de produtos mais vendidos lista os produtos ordenando por aqueles que tiveram maior número de vendas.

Figura 9 – Diagrama de Casos de Uso do sistema web

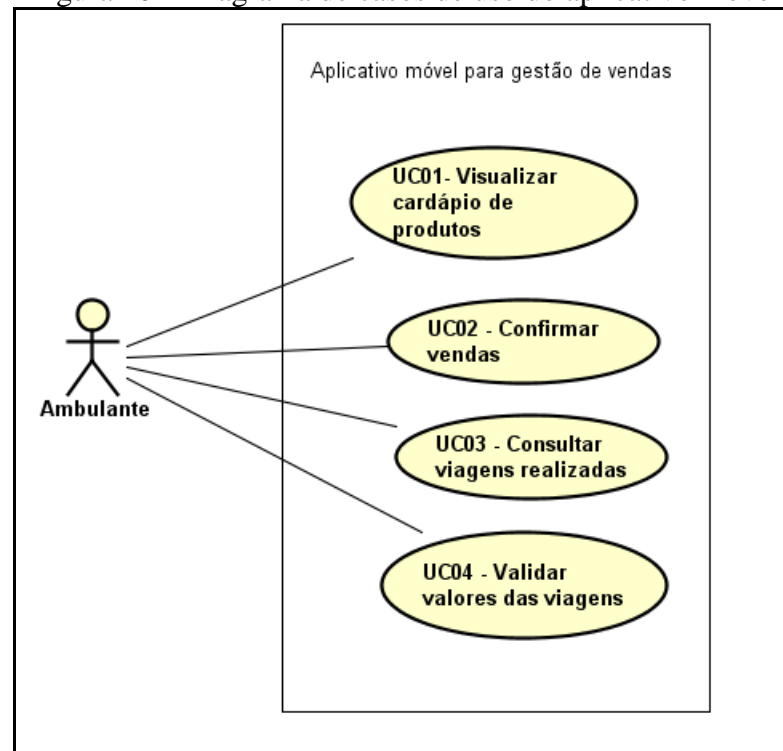


Fonte: elaborado pelo autor.

3.2.2 Diagrama de casos de uso do aplicativo móvel

A Figura 10 apresenta o diagrama de casos de uso do aplicativo móvel desenvolvido. No diagrama é apresentado o ator que persiste no aplicativo que é o ambulante. O detalhamento dos principais casos de uso do aplicativo móvel encontram-se no Apêndice B.

Figura 10 – Diagrama de casos de uso do aplicativo móvel



Fonte: elaborado pelo autor.

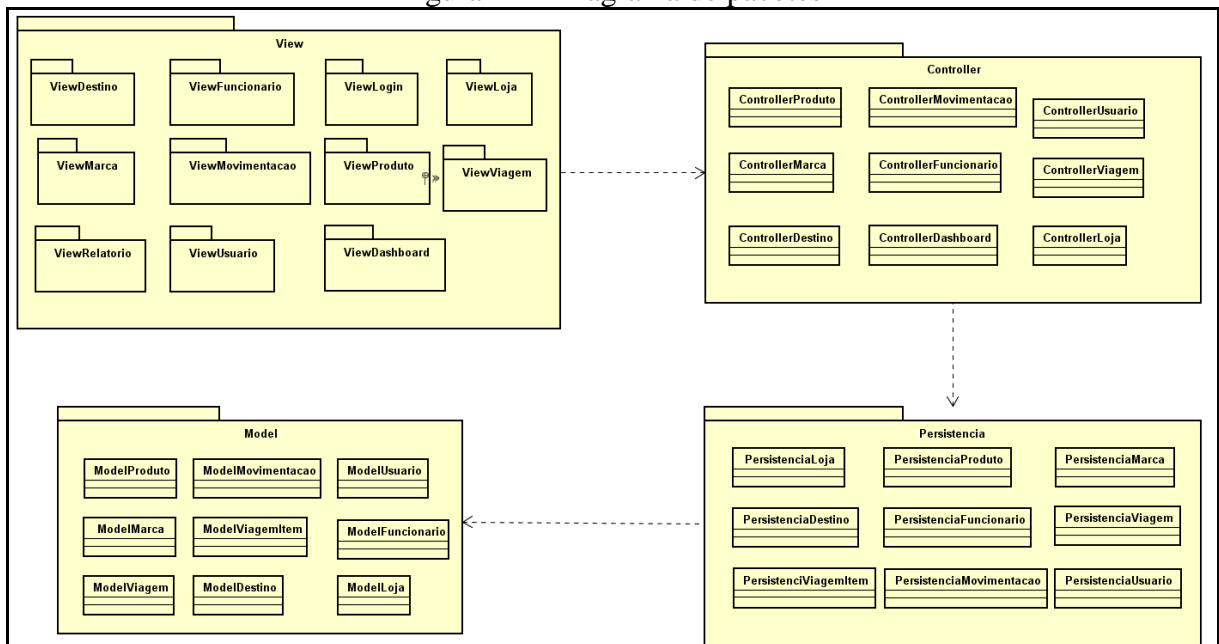
Ao efetuar login no aplicativo móvel, é executado o UC01 - Visualizar cardápio de produtos, em que é apresentado ao ambulante o cardápio de produtos contendo os produtos que estão com ele disponíveis para a venda. O UC02 - Confirmar vendas permite que o ambulante confirme ao sistema a venda de um produto. Para isto na listagem dos produtos preenche-se o campo com a quantidade vendida e clica-se no botão ao lado do campo. O sistema irá realizar o cálculo do total da venda e solicitar a confirmação da mesma.

O UC03 - Consultar viagens realizadas disponibiliza um relatório com todas as viagens já realizadas pelo ambulante. Neste relatório possuem informações como a data de início da viagem, quantidade de produtos levados, total de produtos vendidos e os valores da viagem que permitem a validação das viagens com o UC04 - Validar valores das viagens.

3.2.3 Diagrama de pacotes

Nesta seção é descrita a arquitetura do sistema mostrando o agrupamento das classes criadas para a implementação do sistema web. O diagrama está dividido em 4 pacotes, sendo eles Model, View, Controller e Persistencia como pode ser visto na Figura 11.

Figura 11 – Diagrama de pacotes



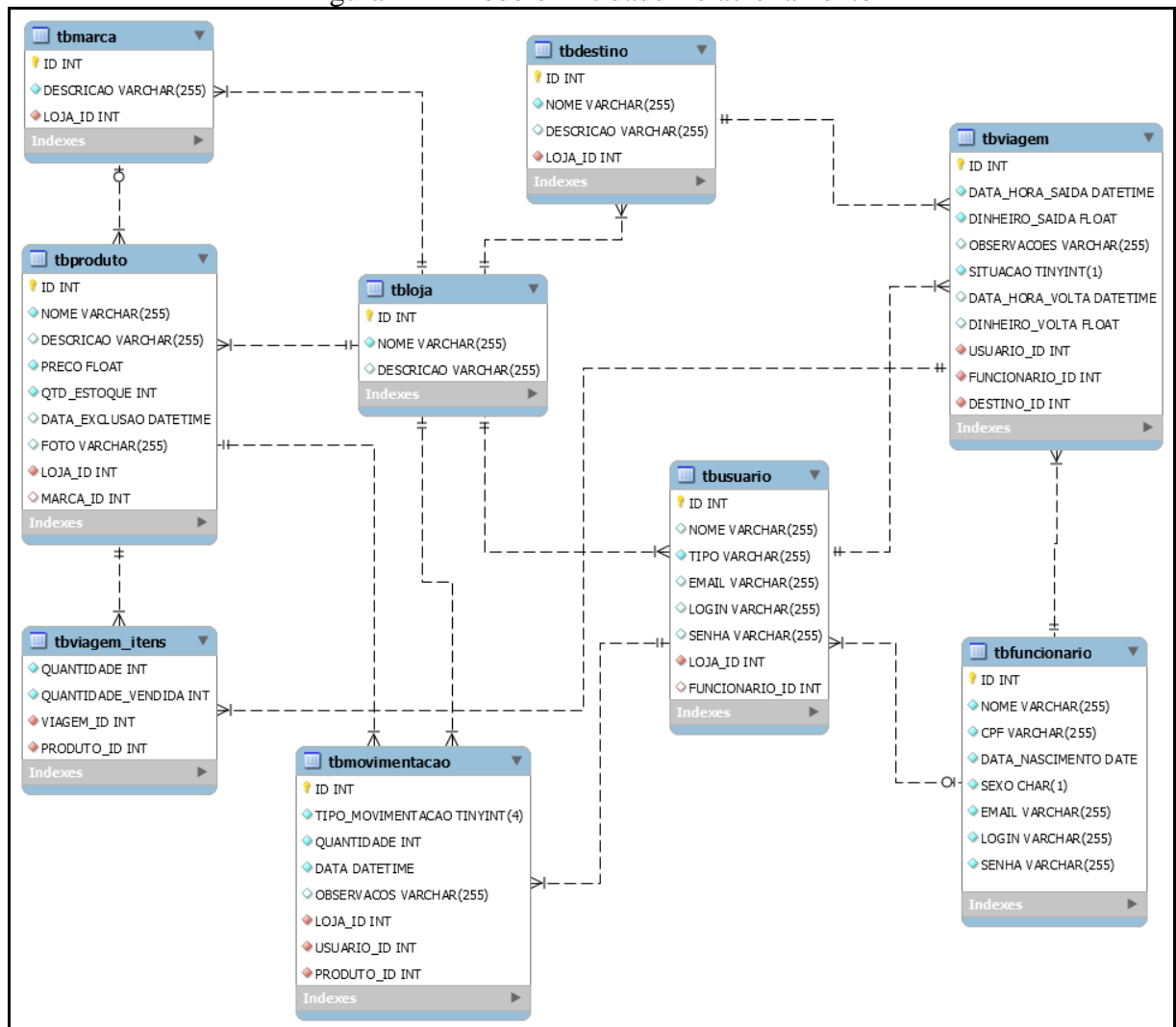
Fonte: elaborado pelo autor.

Os pacotes apresentados na Figura 11 são divididos em subpacotes. Cada subpacote possui as classes referente a uma rotina do sistema. No pacote `Model` estão as classes que representam as entidades do sistema. Nestas classes são realizadas as entradas, leituras e validações dos dados do sistema. As classes presentes no pacote `Controller` são responsáveis por estabelecer a comunicação entre a interface do sistema (`View`) e `Persistencia`. E por fim a camada `Persistencia` é responsável por fazer a comunicação entre a aplicação e o banco de dados.

3.2.4 Modelo Entidade Relacionamento

A Figura 12 apresenta o MER completo do sistema desenvolvido. Para a geração do mesmo foi utilizado o software MySQL Workbench.

Figura 12 – Modelo Entidade Relacionamento



Fonte: elaborado pelo autor.

Cada entidade é representada no banco de dados como uma tabela, a seguir é apresentada uma breve descrição das entidades criadas para o desenvolvimento do sistema:

- tbloja: entidade que armazena as lojas cadastradas no sistema;
- tbmarca: entidade que armazena as marcas cadastradas por cada loja;
- tbproduto: entidade que armazena os produtos cadastrados por cada loja;
- tbdestino: entidade que armazena os destinos em que os ambulantes de cada loja irão realizar as vendas;
- tbviagem: entidade que armazena as viagens cadastradas por cada loja. O relacionamento com a loja se dá pelo atributo `USUARIO_ID`, pois cada usuário está ligado a uma loja;
- tbfuncionario: entidade que armazena o cadastro dos ambulantes no sistema;
- tbusuario: tabela que armazena os usuários do sistema. O atributo `TIPO` define se este usuário é um gestor do estabelecimento ou se é um ambulante que trabalha

para a loja;

- h) `tbviagem_itens`: entidade que armazena os produtos que serão vendidos em cada viagem;
- i) `tbmovimentacao`: entidade que armazena as movimentações nos estoques dos produtos.

3.3 IMPLEMENTAÇÃO

Nesta seção são apresentadas e detalhadas as técnicas e ferramentas utilizadas para o desenvolvimento do sistema, assim como é apresentado como foi feita a implementação apresentando trechos de códigos das principais rotinas. Por fim, é detalhada a operacionalidade do sistema.

3.3.1 Técnicas e ferramentas utilizadas

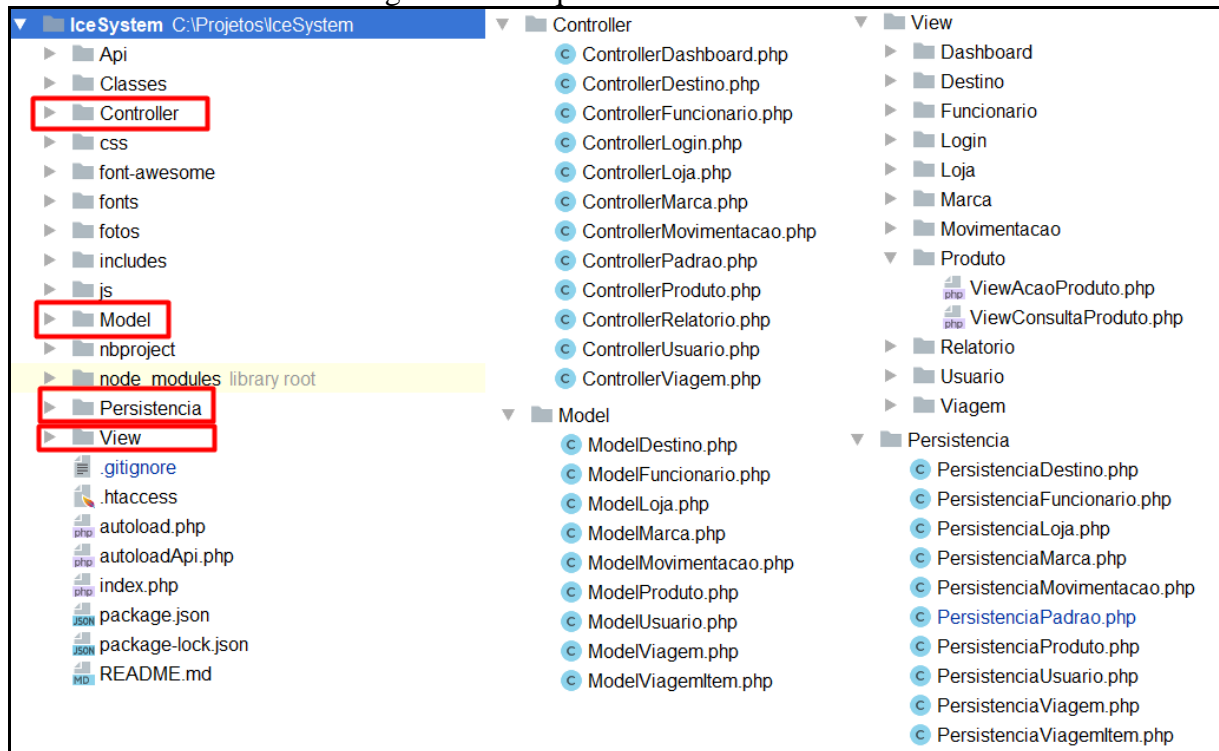
Esta subseção apresenta as técnicas e ferramentas utilizadas para a construção do sistema. Para o desenvolvimento deste trabalho foram utilizadas as seguintes ferramentas:

- a) JetBrains PhpStorm como plataforma de desenvolvimento;
- b) MySQL como banco de dados para armazenamento de dados;
- c) HeidiSQL como gerenciador de banco de dados;
- d) Git para o gerenciamento dos códigos fontes do sistema;
- e) Hypertext Preprocessor (PHP) como linguagem de programação do sistema web;
- f) Arquitetura *Model-View-Controller* (MVC) no desenvolvimento do sistema web;
- g) Bootstrap como *framework* front-end do sistema web e do aplicativo móvel;
- h) Apache Cordova como *framework* para o desenvolvimento do aplicativo móvel;
- i) Representational State Transfer (REST) para comunicação do aplicativo móvel com o sistema web;
- j) Android Studio para os testes do aplicativo móvel;
- k) MySQL Workbench para geração do MER;
- l) Astah Community para criação dos diagramas;
- m) Browsera e Google PageSpeed Insight para realização de testes de desempenho.

O sistema web foi desenvolvido utilizando o banco de dados MySQL e o padrão de arquitetura *Model-View-Controller* (MVC). Este padrão divide a aplicação em três componentes: `model`, `view` e `controller`. O `model` (modelo) é o núcleo de dados da aplicação, o `view` (visualização) apresenta as informações do modelo para o usuário e o `controller` (controlador) gerencia as informações conforme os eventos acionados pelos

usuários (BARROS, 2007). Além desses três componentes, foi criada uma camada com nome de *Persistencia* em que é realizado todas as interações com o banco de dados. A Figura 13 demonstra como o projeto do sistema web ficou estruturado e organizado com o padrão adotado.

Figura 13 – Arquitetura do sistema web



Fonte: elaborado pelo autor.

Para o desenvolvimento do aplicativo móvel foi utilizado o *framework* Apache Cordova que permite o uso de linguagens web para o desenvolvimento de aplicativos multiplataforma para dispositivos móveis. Sendo assim, o aplicativo é desenvolvido como um sistema web e ao final pode ser traduzido para as plataformas de dispositivos móveis. O aplicativo final é executado como um programa nativo do dispositivo móvel e não como um sistema web executado no navegador (Apache Software Foundation, 2018).

3.3.2 Desenvolvimento

Nesta subseção será detalhado como foi realizado o desenvolvimento do sistema. Serão apresentados os trechos de códigos mais significativos tanto do sistema web quanto do aplicativo móvel.

3.3.2.1 Implementação do sistema web

Como relatado na subseção 3.3.1, o sistema web foi implementado utilizando o padrão MVC, com adição de uma nova camada chamada *Persistencia* para o gerenciamento das

conexões com o banco de dados. As classes do pacote Controller são responsáveis por receber todas as requisições do usuário. O Quadro 5 demonstra o método `inserirRegistro` (linha 12) do `ControllerProduto`. A variável `valores` (linha 12) recebe todos os dados enviados pelo usuário no formulário de inserção do produto. Nas linhas a seguir, os dados são atribuídos na classe `ModelProduto` e é chamado o método `insertInto` (linha 27) da `PersistenciaProduto` para o processamento da inserção do registro.

Quadro 5 – Classe ControllerProduto

```

ControllerProduto  alterarRegistro()
1  <?php
2
3  class ControllerProduto {
4
5      /* @var $persistencia PersistenciaProduto */
6      private $persistencia = null;
7
8      function __construct() {
9          $this->persistencia = new PersistenciaProduto();
10     }
11
12     public function inserirRegistro($valores) {
13         $this->persistencia->getModel()->setNome($valores['nome']);
14         $this->persistencia->getModel()->setDescricao($valores['descricao']);
15         $this->persistencia->getModel()->getModelMarca()->setId($valores['marca']);
16         $this->persistencia->getModel()->setPreco(formatNumberFromMaskMoney($valores['preco']));
17         $this->persistencia->getModel()->setQtdEstoque($valores['estoque']);
18         $this->persistencia->getModel()->getModelLoja()->setId($_SESSION['lojaID']);
19
20         if(isset($valores['files'])){
21             $file = $valores['files']['file_image'];
22             $this->persistencia->getModel()->setFoto($file['name']);
23
24             move_uploaded_file($file['tmp_name'], __DIR__.'../fotos/'.$file['name']);
25         }
26
27         $this->persistencia->insertInto();
28     }

```

Fonte: elaborado pelo autor.

No Quadro 6 é detalhado o método `insertInto` (linha 20) que é chamada na classe `ControllerProduto`. É criado um *array*, sendo que sua chave representa o nome do campo no banco de dados e o valor é a informação enviada pelo usuário que foi atribuído ao modelo na classe `controller`. Este *array* é passado juntamente com a mensagem que será exibida ao usuário caso não ocorram erros para o método `insertInto` (linha 29) da classe pai `PersistenciaPadrao` que possui métodos padrões para as interações principais com o banco de dados. O Quadro 7 detalha o método de inserção.

Quadro 6 – Classe PersistenciaProduto

```

PersistenciaProduto setModel()
1  <?php
2
3  class PersistenciaProduto extends PersistenciaPadrao {
4
5      /* @var $model ModelProduto */
6      private $model = null;
7
8      function __construct() {
9          $this->model = new ModelProduto();
10     }
11
12     protected function getTabela() {
13         return 'tbproduto';
14     }
15
16     protected function getNomeRotina(){
17         return 'Produto';
18     }
19
20     public function insertInto() {
21         $valores = array('id'           => null,
22                         'nome'        => $this->getModel()->getNome(),
23                         'descricao'    => $this->getModel()->getDescricao(),
24                         'marca_id'     => $this->getModel()->getModelMarca()->getId(),
25                         'preco'        => $this->getModel()->getPreco(),
26                         'qtd_estoque'  => $this->getModel()->getQtdEstoque(),
27                         'loja_id'      => $this->getModel()->getModelLoja()->getId(),
28                         'foto'        => $this->getModel()->getFoto());
29         parent::insertInto($valores, 'Produto inserido com sucesso.');
30     }

```

Fonte: elaborado pelo autor.

No Quadro 7 é apresentado o método padrão para a inserção dos registros no banco de dados. A função, declarada na linha 37 recebe os valores e a mensagem a ser apresentada ao usuário caso não ocorra erros. Após a execução do SQL de inserção do registro (linha 43) o usuário é direcionado para a página de listagem dos registros em que será exibida a mensagem (linha 49).

Quadro 7 – Classe PersistenciaPadrao

```
PersistenciaPadrao insertInto()
37 protected function insertInto($valores,$message){
38     $pdo = $this->getConnection();
39
40     $campos = implode(',', array_keys($valores));
41     $values = ':' . str_replace(',', ':', $campos);
42
43     $stmt = $pdo->prepare('INSERT INTO '.$this->getTabela().' ('.$campos.') values (.'.$values.')');
44     foreach ($valores as $key => $valor ) {
45         $stmt->bindValue($key, $valor );
46     }
47
48     if($stmt->execute()){
49         FlashMessages::addMessage('success', $message);
50     }else{
51         FlashMessages::addMessage('error', 'Erro ao inserir registro');
52     }
53
54     header('Location: /'.NOME_PROJETO.'/'.strtolower($this->getNomeRotina()).'/listar');
55 }
```

Fonte: elaborado pelo autor.

Desta forma, funcionam todas as rotinas do sistema web. O usuário faz a requisição os controllers, informam os dados nos models e as persistências fazem a interação com o banco de dados.

3.3.2.2 Implementação do aplicativo móvel

Como relatado na subseção 3.3.1 o aplicativo móvel foi desenvolvido utilizando o *framework* Apache Cordova. O Quadro 8 apresenta o arquivo `config.xml` que contém as configurações do aplicativo desenvolvido. Este arquivo deve estar presente na raiz no projeto, caso contrário ocorrerá erro na construção do aplicativo.

Quadro 8 – Arquivo de configuração do Apache Cordova

```

1 <?xml version='1.0' encoding='utf-8' ?>
2 <widget id="io.cordova.hellocordova" version="1.0.0" xmlns="http://www.w3.org/ns/widgets"
3     xmlns:cdv="http://cordova.apache.org/ns/1.0">
4     <name>Ice Cream System</name>
5     <description>
6         Sistema mobile para vendas de ambulantes.
7     </description>
8     <author email="guga.egez@hotmail.com" href="http://cordova.io" </author>
9     <content src="index.html" />
10    <access origin="*" />
11    <access origin="http://*" />
12    <access origin="https://*" />
13    <allow-intent href="tel:*" />
14    <allow-intent href="sms:*" />
15    <allow-intent href="mailto:*" />
16    <allow-intent href="geo:*" />
17    <platform name="android">
18        <allow-intent href="market:*" />
19    </platform>
20    <platform name="ios">
21        <allow-intent href="itms:*" />
22        <allow-intent href="itms-apps:*" />
23    </platform>
24    <plugin name="cordova-plugin-dialogs" spec="^2.0.1" />
25 </widget>

```

Fonte: elaborado pelo autor.

Dentre as configurações que estão no arquivo apresentado no Quadro 8 duas tem importância maior. A *tag* *name* (linha 4) define qual será o nome que aparecerá nos dispositivos em que ele for instalado e a *tag* *content* (linha 9) define qual a página inicial quando o aplicativo for iniciado. Neste arquivo também é configurado em quais plataformas será disponibilizado o aplicativo e quais *plug-ins* que serão utilizados. No caso do aplicativo desenvolvido neste trabalho, foi disponibilizado para Android e utilizou-se o *plug-in* *cordova-plugins-dialogs* para fazer as notificações ao usuário.

Para fazer comunicação com o sistema web é utilizado o protocolo de comunicação Representational State Transfer (REST). O Quadro 9 mostra como é apresentado os produtos ao acessar a tela do cardápio no aplicativo. É realizado uma requisição do tipo *GET* (linha 70) para a API desenvolvida no sistema web, que busca os produtos do funcionário que está autenticado no aplicativo (linhas 71 e 72) e retorna um JSON com o HTML da listagem dos produtos (linha 76).

Quadro 9 – Requisição para a listagem dos produtos

```

68  function carregaListaProdutos() {
69      $.ajax({
70          type: "GET",
71          url: url + "/IceSystem/Api/busca-itens-funcionario.php",
72          data: 'funcionario_id=' + funcionarioId+'&urlAtual='+url,
73          success: function (data) {
74              var retorno = JSON.parse(data);
75              if (retorno.html) {
76                  $(''.product-catalog').html($.parseHTML(retorno.html));
77                  initTouchSpin();
78              }
79          },
80          error: function () {
81              alert("Errrr is occured");
82          }
83      });
84  }

```

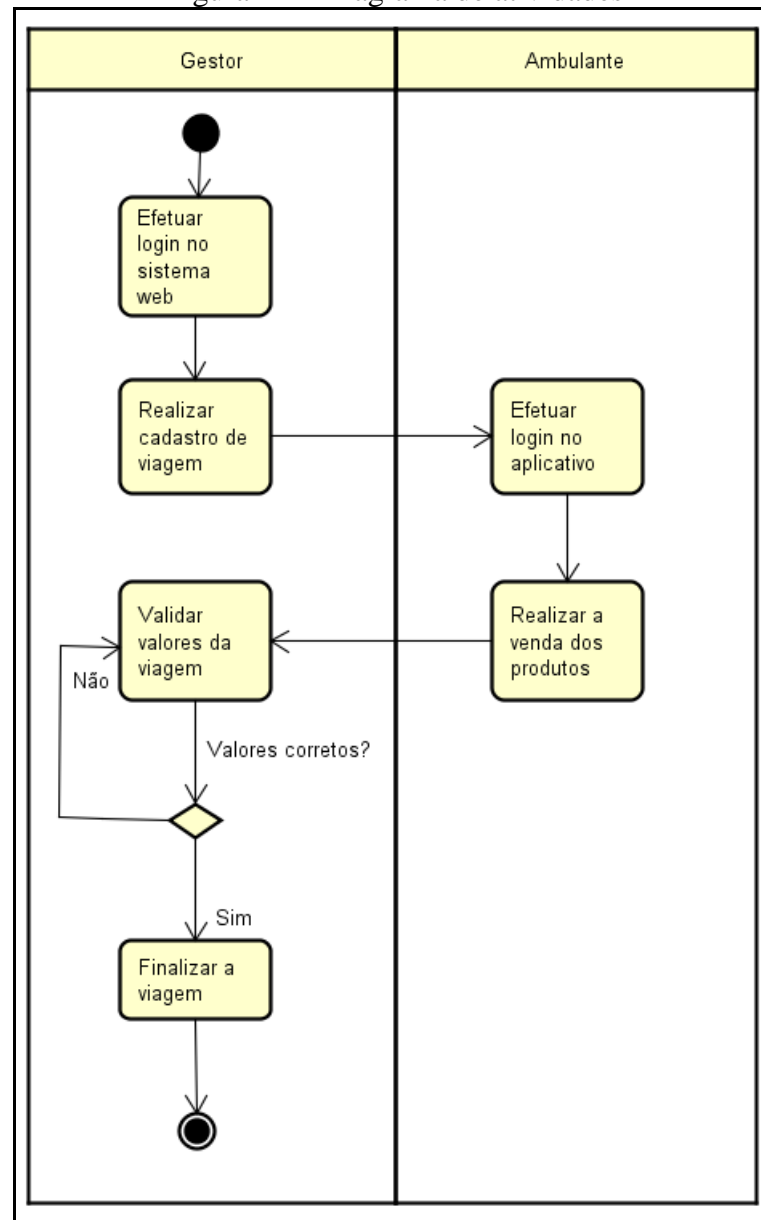
Fonte: elaborado pelo autor.

3.3.3 Operacionalidade da implementação

Nesta subseção é apresentada a operacionalidade do sistema, demonstrando o fluxo principal através do diagrama de atividades. A Figura 14 ilustra o diagrama de atividades que contém os dois principais atores do sistema: Gestor e o Ambulante.

Primeiramente o gestor deverá realizar o *login* no sistema web e acessar a tela de cadastro de viagens, preencher os dados solicitados e finalizar o cadastro da viagem. Com isso o ambulante, ao efetuar *login* no aplicativo móvel terá listado todos os produtos que estão com ele disponíveis para a venda. Conforme o ambulante realiza a venda dos produtos, ele confirma pelo aplicativo a quantidade que foi vendida. Ao término das vendas, o ambulante retorna ao estabelecimento e juntamente com o gestor realiza a validação dos valores da viagem. Caso os valores estejam corretos a viagem é finalizada, incrementando as informações disponibilizadas nos relatórios.

Figura 14 – Diagrama de atividades



Fonte: elaborado pelo autor.

Para melhor compreensão do fluxo, a Figura 15 ilustra a tela de listagem de viagens, tela que deve ser acessada pelo gestor após efetuar o *login* no sistema web. Nesta tela estão todas as viagens cadastradas no sistema, contendo: *ambulante* (nome do ambulante responsável pela viagem); a data e hora de saída do ambulante do estabelecimento; e, a situação da viagem que pode ser *Em andamento* ou *Finalizada*. Caso a viagem esteja com o status *Finalizada* significa que o ambulante já retornou ao estabelecimento após realizar as suas vendas e os valores das viagens foram validados. Ao retornar para o estabelecimento, o gestor pode finalizar a viagem clicando no botão *Finalizar* localizado na linha da viagem correspondente. Ainda, para qualquer situação, é possível visualizar os detalhes da viagem

clicando no botão `Detalhes`. Para cadastro de uma nova viagem, o gestor deverá clicar no botão `Adicionar Novo` localizado no canto superior direito da tela.

Figura 15 – Tela de listagem de viagens

Código	Ambulante	Data/Hora Saída	Situação	Ações
18	Thomas Pereira	07/06/2018 21:07:04	Em andamento	Finalizar Detalhes
17	Thomas Pereira	28/05/2018 09:02:35	Finalizada	Detalhes
15	Michael da Silva	28/05/2018 10:07:55	Em andamento	Finalizar Detalhes
14	Thomas Pereira	28/05/2018 03:07:05	Finalizada	Detalhes

Fonte: elaborado pelo autor.

A Figura 16 apresenta a tela de cadastro de viagem. O gestor deve informar qual o ambulante que irá realizar a viagem, qual o destino da viagem e quais produtos que serão levados. Na tela os campos obrigatórios são indicados pelo símbolo de asterisco (*) na frente do campo, seguindo padrões de usabilidade. Ainda são divididos em partes, sendo a primeira parte dados da viagem, a segunda parte são os dados do produto e a última parte informações relacionadas a valores.

Figura 16 – Tela de cadastro de viagens

Produto	Valor unitário (R\$)	Quantidade	Total (R\$)
Magnum Branco	7,50	10	75,00
Magnum Clássico	6,00	5	30,00
Magnum Red Velvet	7,00	8	56,00

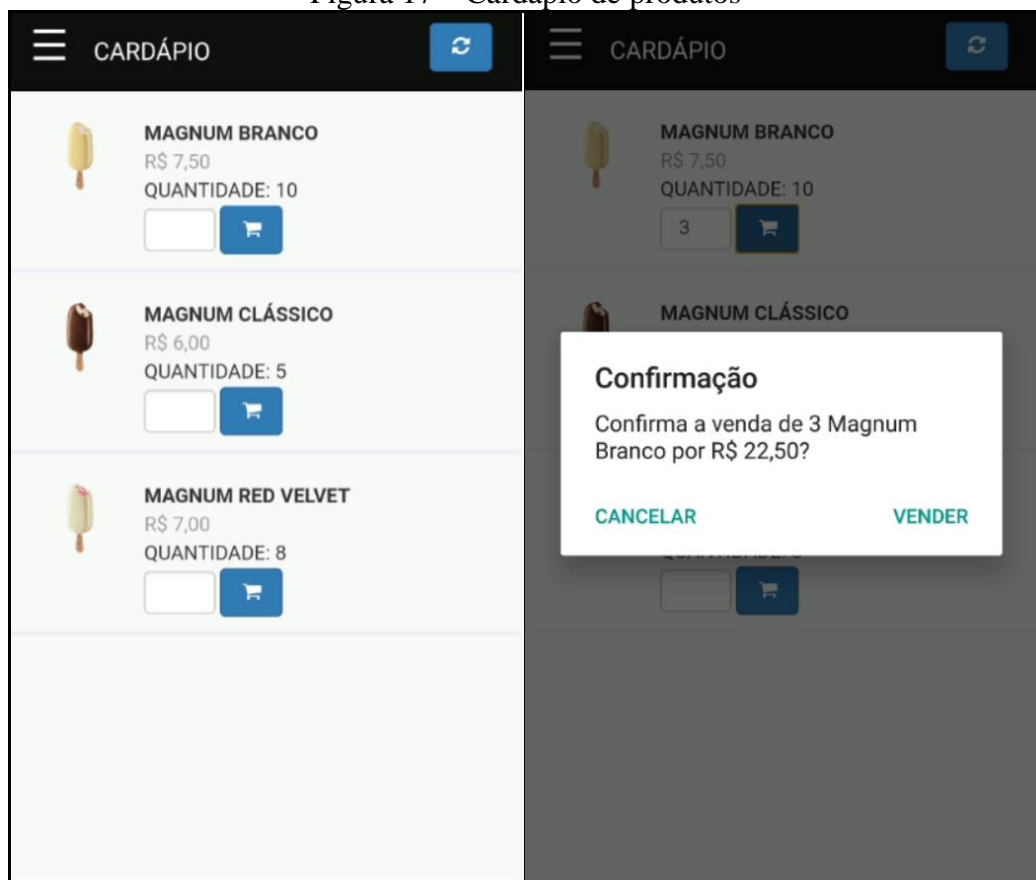
Fonte: elaborado pelo autor.

No quadro de `Produto`, ficam os dados dos produtos que estão sendo levados pelo ambulante para venda, junto com o valor unitário dos produtos, a quantidade e o valor

total em reais correspondente aos produtos levados. O campo de seleção permite ao gestor selecionar um produto cadastrado no sistema, sendo que após escolhido já é preenchido o campo valor unitário com o seu valor. Ao informar a quantidade que está sendo levada o sistema já realiza o cálculo do total unitário do produto e atualiza os valores da viagem. Conforme são adicionados mais produtos, o sistema atualiza em tempo real os valores da viagem. Após informar os produtos, o gestor deve informar qual valor o ambulante estará levando da empresa para trocos e afins. Com todos os dados preenchidos o cadastro da viagem pode ser concluído clicando no botão enviar.

Com a viagem cadastrada, quando o ambulante realiza o login no aplicativo móvel é apresentado o cardápio dos produtos disponíveis para a venda. Caso já esteja autenticado no aplicativo, pode-se atualizar o cardápio através do botão no canto superior a direita (Figura 17). O cardápio contém o preço e a quantidade de cada produto. Após o ambulante preencher a quantidade e clicar no botão com o símbolo de carrinho, o aplicativo solicita a confirmação informando o valor total da venda. A Figura 17 demonstra a listagem dos produtos no cardápio de vendas e a confirmação da venda de um produto.

Figura 17 – Cardápio de produtos



Fonte: elaborado pelo autor.

Ao término das vendas, o ambulante retorna ao estabelecimento e o gestor pode finalizar a viagem clicando no botão Finalizar na listagem da viagem, que pode ser vista na Figura 15. A Figura 18 demonstra a tela de finalização da viagem, em que é apresentado todos os valores da viagem.

Figura 18 – Tela de finalização da viagem

ID	Produto	Quantidade Levada	Valor Unitário (R\$)	Quantidade Vendida	Total Vendido (R\$)
16	Magnum Branco	10	7,50	8	60,00
17	Magnum Clássico	5	6,00	5	30,00
18	Magnum Red Velvet	8	7,00	8	56,00

Total em produtos levado (R\$) 161,00 Dinheiro levado (R\$) 25,00
 Total Vendido (R\$) 146,00 Valor a ser retornado (R\$) 171,00
 Valor retornado (R\$)

Fonte: elaborado pelo autor.

Conforme ilustrado na Figura 18 a tela de finalização da viagem já vem com todos os valores finais da viagem preenchidos. A caixa de produtos e valores da viagem contém um campo a mais que no cadastro da viagem que é a quantidade vendida. Com base nos produtos vendidos pelo ambulante o sistema realiza o cálculo do valor que deve ser retornado para o gestor, que é o total em produtos vendido mais o valor levado para troco e afins. Para realizar a finalização da venda o gestor informa o valor retornado pelo ambulante. Com os valores validados a viagem é finalizada e os dados dos relatórios gerenciais são incrementados. Caso haja alguma diferença nos valores ou o ambulante retorne um valor menor do que o esperado, o sistema exibe um alerta para que se verifique o caso. Os valores finais da viagem devem bater com os valores apresentados ao ambulante no aplicativo móvel no relatório de viagens em andamento, conforme Figura 19.

Figura 19 – Relatório de viagens no aplicativo móvel

VIAGENS EM ANDAMENTO
Loja: Distribuidora Kibon
Início: 28/05/2018 09:02:35
Dinheiro levado: R\$ 25,00
Quantidade de itens levados: 23
Quantidade de itens vendidos: 21
Total em itens vendidos: R\$ 146,00
Total a retornar: R\$ 171,00
VIAGENS FINALIZADAS
Loja: Distribuidora Kibon
Data de Início: 28/05/2018 03:07:05
Dinheiro levado: R\$ 10,00
Quantidade de itens levados: 1
Quantidade de itens vendidos: 1
Total em itens vendidos: R\$ 6,00
Total retornado: R\$ 15,00
Data do Término: 27/05/2018 22:07:17

Fonte: elaborado pelo autor.

3.4 RESULTADOS E DISCUSSÕES

O objetivo geral do trabalho foi atendido, apresentando um sistema web em que o gestor do estabelecimento consegue ter controle dos produtos que são comercializados pelos ambulantes que trabalham para em sua empresa, além de ter um controle do estoque de seus produtos. Por sua vez, com o aplicativo móvel desenvolvido, o ambulante consegue

visualizar quais produtos tem disponíveis para venda, além de poder acompanhar os valores de suas vendas.

Para realizar testes de performance e desempenho, o sistema web desenvolvido foi publicado utilizando dos serviços de hospedagem da empresa 000webhost. Esta hospedagem é totalmente grátis e livre de anúncios, possuindo suporte para a linguagem PHP e banco de dados MySQL. O primeiro teste realizado foi utilizando a ferramenta Browsera, que é uma ferramenta para testes de compatibilidade de *websites* em diversos browsers. O Browsera realiza uma varredura dentro das páginas do *website* e registra erros de JavaScript e outros problemas com relação aos leiautes. A Figura 20 apresenta os testes no browser Google Chrome.

Figura 20 – Resultado dos testes com a ferramenta Browsera no Google Chrome

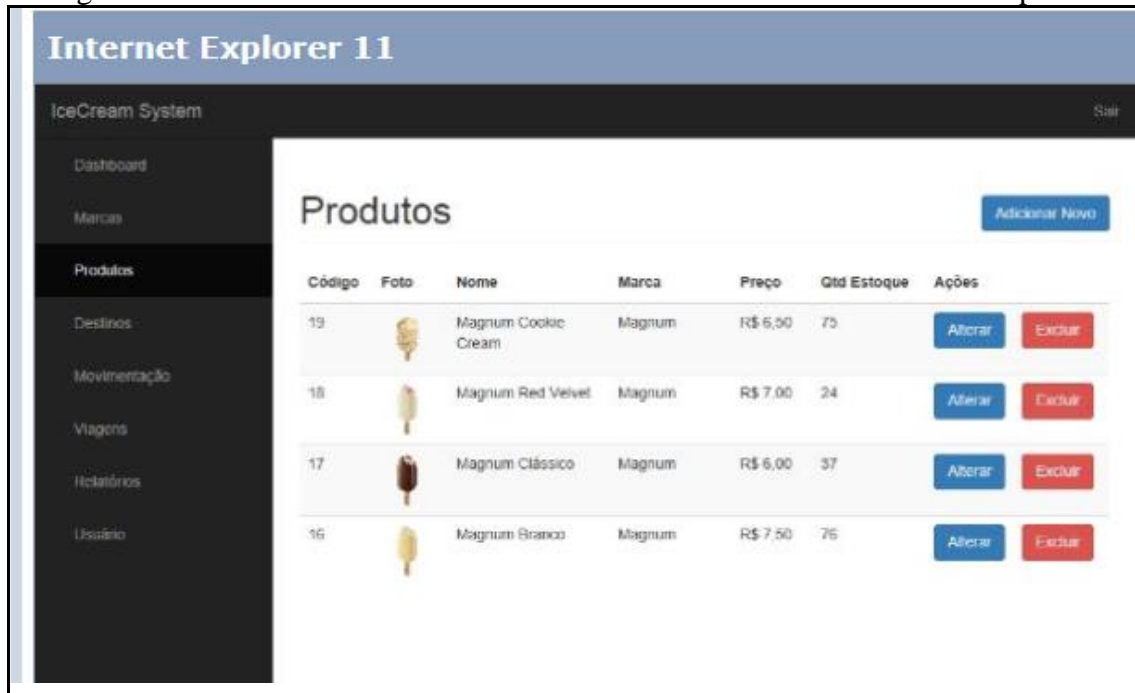
The screenshot shows the Browsera interface. At the top, there is a red 'X' icon and the text 'Layout Differences'. Below this, a section titled 'Difference Details' contains the text 'No Problems Detected'. The main part of the screenshot is a preview of the 'IceCream System' website in Chrome 64. The website has a dark sidebar menu with options like 'Dashboard', 'Marcas', 'Produtos', 'Destinos', 'Movimentação', 'Viagens', 'Relatórios', and 'Usuário'. The 'Produtos' section is active, displaying a table of products.

Código	Foto	Nome	Marca	Preço	Qty Estoque	Ações
19		Magnum Cookie Cream	Magnum	R\$ 6,50	75	Alterar Excluir
18		Magnum Red Velvet	Magnum	R\$ 7,00	24	Alterar Excluir
17		Magnum Clássico	Magnum	R\$ 5,00	37	Alterar Excluir
16		Magnum Branco	Magnum	R\$ 7,50	76	Alterar Excluir

Fonte: elaborado pelo autor.

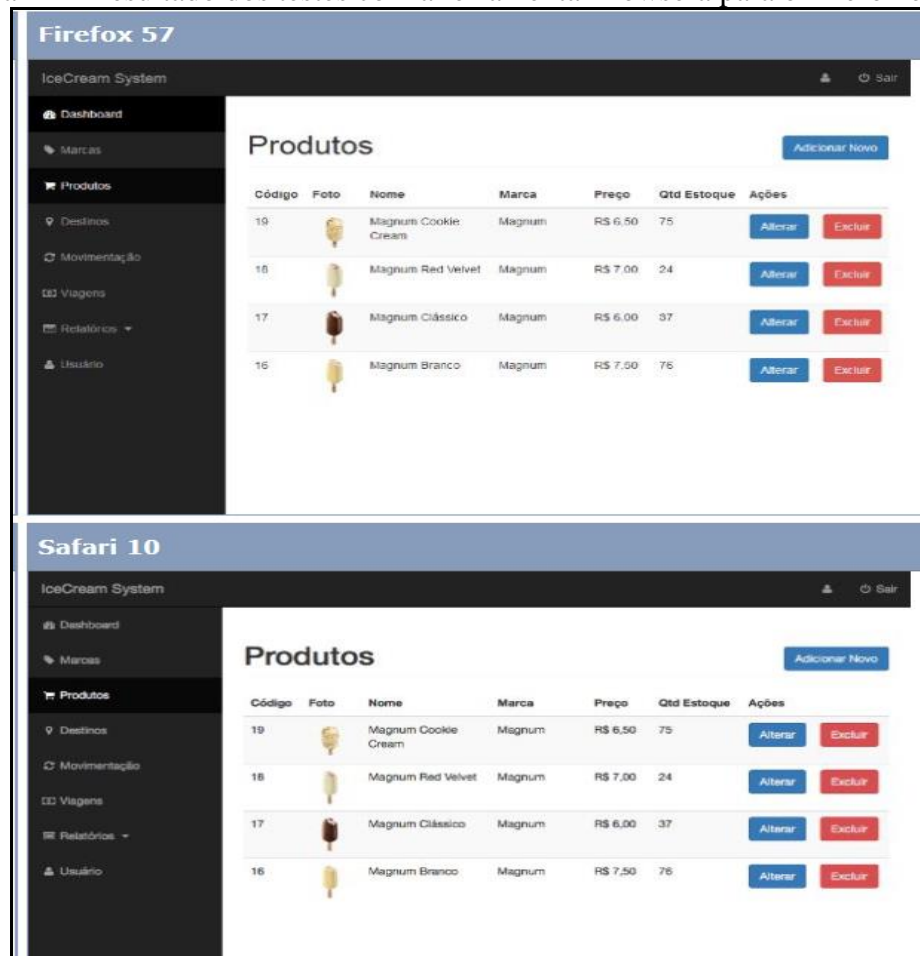
A Figura 21 apresenta os resultados dos testes para o browser Internet Explorer. Por sua vez, a Figura 22 apresenta os resultados para os browsers Firefox e Safari. Como pode-se observar, os testes não apontaram erros, garantindo que o sistema é acessível em todos os browsers.

Figura 21 – Resultado dos testes com a ferramenta Browsera no Internet Explorer



Fonte: elaborado pelo autor.

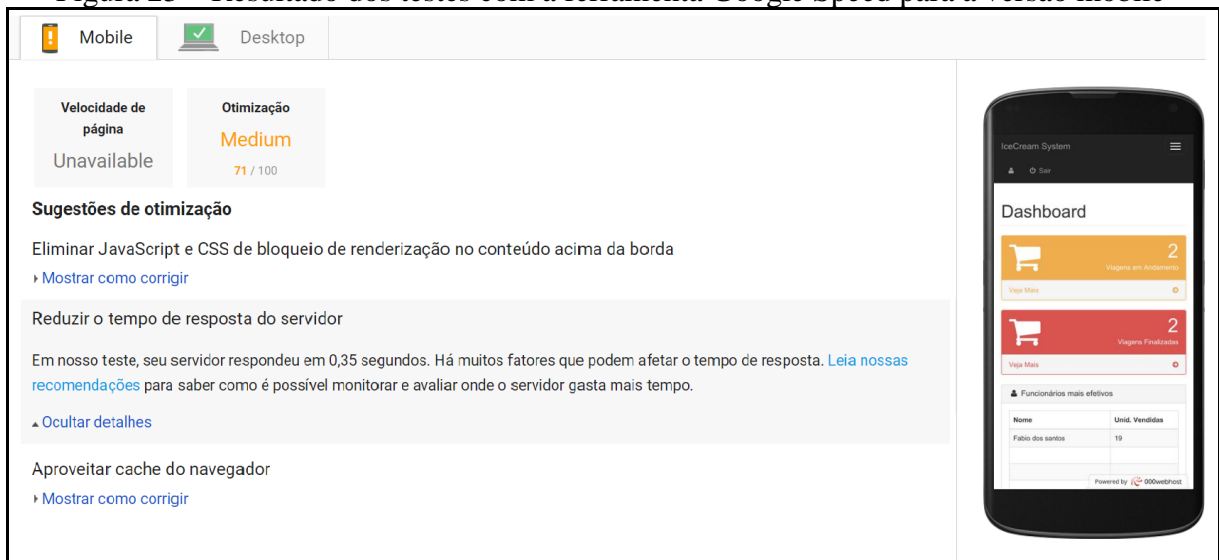
Figura 22 – Resultado dos testes com a ferramenta Browsera para o Firefox e Safari



Fonte: elaborado pelo autor.

Outro teste realizado foi com a ferramenta Google PageSpeed Insights que visa analisar o desempenho de um site e conforme os problemas são apresentados, sugestões para correção e otimizações são mostrados. A avaliação é feita principalmente quanto às imagens, JavaScript, CSS e outros atributos *front-end* que afetam o carregamento das páginas. A Figura 23 demonstra os resultados do teste para a versão móvel do sistema web.

Figura 23 – Resultado dos testes com a ferramenta Google Speed para a versão mobile



Fonte: elaborado pelo autor.

Conforme a Figura 23 a versão móvel atingiu a pontuação de 71/100 que se encaixa na pontuação categorizada como média, que significa que algumas otimizações comuns de desempenho não foram aplicadas às páginas. Dentre as sugestões de otimização estão as opções de eliminar JavaScript e CSS de bloqueio de renderização no conteúdo acima da borda, reduzir o tempo de resposta do servidor e aproveitar o cache do navegador. A Figura 24 apresenta os resultados para a versão desktop do sistema web.

A Figura 24 apresenta os testes para versão desktop, que atingiu a pontuação de 87/100 que se encaixa na pontuação categorizada como boa, que significa que a maioria das práticas recomendações de desempenho foram aplicadas às páginas. Além das sugestões de otimização sugeridas para a versão móvel, para a versão desktop foi sugerido compactar os arquivos de CSS e JavaScript.

Figura 24 – Resultado dos testes com a ferramenta Google Speed para a versão desktop



Fonte: elaborado pelo autor.

O Quadro 10 apresenta a comparação entre o trabalho desenvolvido e os trabalhos correlatos, em que as linhas representam as características contempladas por cada um e as colunas os trabalhos.

Quadro 10 – Características dos trabalhos correlatos e do trabalho desenvolvido

	Silva (2016)	No Ar (2016)	NaPraia (2016)	Luiz (2011)	Trabalho desenvolvido
Controle de vendas	X	X	X	X	X
Cadastro de produtos	X			X	X
Controle de estoque				X	X
Pagamento da venda online		X	X		
Avaliação do vendedor			X		
Plataforma	iOS	Android	Android/iOS	Web	Web/Mobile

Fonte: elaborado pelo autor.

Conforme ilustrado no Quadro 10, todos os trabalhos possuem um controle de vendas. O aplicativo de Silva (2016) e o sistema de Luiz (2011) permitem o cadastro dos produtos a serem gerenciados no sistema, enquanto os aplicativos de No Ar (2016) e NaPraia (2016) apenas permitem que o vendedor se associe a uma categoria de produtos e informe a descrição de seu serviço. Estes últimos dois possuem um sistema de geolocalização acoplado, permitindo o usuário encontrar o ambulante mais perto de si.

O grande diferencial do trabalho desenvolvido é o fato de ser um sistema híbrido, em que o sistema web desenvolvido para uso dos gestores dos estabelecimentos de vendas de sorvetes e o aplicativo móvel desenvolvido para uso dos ambulantes compartilham os mesmos dados para se chegar ao objetivo final que é a gestão das vendas de sorvetes por ambulantes. Percebe-se através do Quadro 10 a falta das funcionalidades de pagamento de venda online e avaliação do vendedor que não estavam entre os objetivos do trabalho.

4 CONCLUSÕES

Neste trabalho é apresentado o desenvolvimento de um sistema híbrido para permitir a gestão de vendas de sorvetes por ambulantes. Pode-se afirmar que os objetivos foram alcançados, pois o sistema web desenvolvido permite que os gestores dos estabelecimentos que realizam a venda de sorvetes por ambulantes tenham um controle dos produtos que estão sendo comercializados, além de possuir um controle de estoque e relatórios que podem ajudar na tomada de decisões na empresa. Ainda, com o aplicativo móvel o ambulante consegue visualizar os produtos que tem para venda, além de confirmar a venda dos produtos e acompanhar os valores de suas viagens.

Para o desenvolvimento do sistema web foi utilizada a linguagem de programação PHP e a plataforma JetBrains PhpStorm para a edição dos códigos, nas quais não foi encontrada dificuldades no uso, devido a possuir um conhecimento prévio sobre as tecnologias. O aplicativo móvel foi desenvolvido fazendo uso do *framework* Apache Cordova que permite a criação de aplicativos híbridos. Apesar da possibilidade da criação do aplicativo para iOS com o mesmo código, o mesmo foi desenvolvido e disponibilizado somente para Android, pois seria necessário a utilização de equipamentos da Apple para fazê-lo e testá-lo.

Os testes realizados em cima do sistema web que foi publicado na hospedagem gratuita da empresa 000webhost foram importantes para garantir que o sistema possa ser acessado em qualquer browser, além de identificar sugestões de melhorias de desempenho para o sistema, que serão sugeridos como melhorias futuras.

As maiores dificuldades encontradas no desenvolvimento do trabalho foi a falta de experiência no desenvolvimento para aplicativos móveis, o que resultou em muito tempo de pesquisa para a configuração do ambiente de desenvolvimento com o Apache Cordova. O processo de comunicação entre o aplicativo móvel e a API disponibilizada no sistema web foi um processo complicado devido a configurações de permissão que o autor desta monografia não tinha conhecimento. A utilização de um emulador para os testes também foi um fator que levou tempo até conseguir ser configurado corretamente.

Como contribuição social, com o desenvolvimento deste trabalho espera-se uma facilidade no gerenciamento da venda dos sorvetes por ambulantes, permitindo ao gestor do estabelecimento o acompanhamento das vendas que estão acontecendo, além de ter um controle do estoque de seus produtos. Por sua vez, o ambulante poderá ter mais agilidade nas vendas com a visualização do cardápio de produtos disponíveis para a venda no aplicativo móvel, além de ter a certeza de que o gestor não lhe cobrará um retorno maior do que deve

por uma viagem, através dos relatórios de viagens, aonde pode visualizar os valores das viagens que estão em andamento e das viagens finalizadas.

Como limitações tem-se a falta de filtros nas páginas de listagem dos registros e nos relatórios disponíveis no sistema web e no aplicativo móvel desenvolvidos. Assim como a falta do recurso de recuperação de senhas, caso o gestor não tenha mais acesso ao sistema web ou o ambulante ao aplicativo móvel.

4.1 EXTENSÕES

Para extensões deste trabalho propõe-se:

- a) disponibilização de filtros nas telas de consultas do sistema web;
- b) realizar as melhorias sugeridas nos resultados dos testes com a ferramenta Google PageSpeed Insight;
- c) gerar novos relatórios no sistema web para um maior auxílio ao gestor na tomada de decisões da empresa;
- d) implementação dos recursos de recuperação de senha, tanto no sistema web quanto no aplicativo móvel;
- e) permitir o ambulante informar o valor que o cliente pagou pela venda e informar quanto de troco terá que ser devolvido;
- f) incluir um sistema de localização no aplicativo móvel para saber aonde o ambulante se encontra realizando as vendas;
- g) criar um recurso de avaliação dos ambulantes;
- h) criar um relatório mais completo no aplicativo móvel contendo os produtos;
- i) gerar e testar versão do aplicativo móvel em dispositivos iOS.

REFERÊNCIAS

- ABIS, ASSOCIAÇÃO BRASILEIRA DAS INDÚSTRIAS E DO SETOR DE SORVETES. **Produção e consumo de Sorvetes no Brasil**. São Paulo, 2016. Disponível em: <http://www.abis.com.br/estatistica_producaoconsumodesorvetesnobrasil.html/>. Acesso em: 01 set 2017.
- APACHE SOFTWARE FOUNDATION, **Cordova Documentation**, 2015. Disponível em: <<https://cordova.apache.org/docs/en/latest/guide/overview/>>. Acesso em: 23 mai 2018.
- BATISTA, Emerson de Oliveira. **Sistema de Informação: o uso consciente da tecnologia para o gerenciamento**. São Paulo: Saraiva, 2004.
- BARROS, Tiago. **State MVC: Estendendo o padrão MVC para uso no desenvolvimento de aplicações**. SugarLoaf-PLoP, 2007. Disponível em: <<http://www.tiagobarros.org/docs/SMVC.pdf>>. Acesso em: 23 mai 2018.
- BEZERRA, P. T.; SCHIMIGUEL, J. **Desenvolvimento de aplicações mobile cross-platform utilizando phonegap**. 2016. Disponível em: <<http://eumed.net/cursecon/ecolat/br/16/phonegap.html>>. Acesso em: 25 jun 2018.
- FERNANDES, Gilberto Oliveira. **Setor informal da economia e a dinâmica dos territórios nas praias da cidade do Natal/RN**. 2008. 157 f. Dissertação (Mestrado em Dinâmica e Reestruturação do Território) - Universidade Federal do Rio Grande do Norte, Natal, 2008.
- GONÇALVES, Vinicius. **Como Montar Um Comércio Ambulante**. 2012. Disponível em: <<http://www.novonegocio.com.br/ideias-de-negocios/como-montar-um-comercio-ambulante/>>. Acesso em: 09 set 2017.
- LOPES, Sergio. **Aplicações mobile híbridas com Cordova e PhoneGap**. São Paulo: Casa do Código, 2016.
- LUIZ, Graciela M. **Sistema de gestão de estoques para distribuidoras de fumo**. 2011. 57 f. Trabalho de Conclusão de Curso (Sistemas de Informação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.
- MOURA, Igor Silva de. **Desenvolvimento de Aplicativos Híbridos baseado em Arquitetura Orientada a Serviços**. 2016. 60 f. Monografia (Especialização em Engenharia de Software) - Universidade Estadual do Ceará, Fortaleza. 2016.
- NAPRAIA. **NaPraia: Peça, receba e pague na praia**. 2016. Disponível em: <<https://www.napraiaapp.com/>>. Acesso em: 05 set 2017.
- NETO, José. **Vendedor ambulante: Veja como começar, legalizar e ganhar dinheiro**. 2015. Disponível em: <<https://www.montarumnegocio.com/vendedor-ambulante/>>. Acesso em: 17 set 2017.
- NO AR, Ideia. **Ambulante: A melhor forma de encontrar um ambulante**. 2016. Disponível em: <<http://ambulantelandingpage.azurewebsites.net/>>. Acesso em 05 set 2017.
- OLIVEIRA, Pablo Benquerer de. **A importância de um sistema de informação gerencial no controle de estoques em uma empresa de locação de fantasias**. 2016. 49 f. Monografia (Graduação em Engenharia de Produção) – Instituto de Ciências Exatas e Aplicadas, Universidade Federal de Ouro Preto, João Monlevade, 2016.
- OSÓRIO, Fernando Santos; VIEIRA, Renata. **Sistemas híbridos inteligentes**. XIX congresso da SBC. ENIA'99 - Encontro Nacional de Inteligência Artificial, Rio de Janeiro, 1999. Disponível em: <<http://osorio.wait4.org/oldsite/enia99/enia99.pdf>>. Acesso em: 09 set 2017.

PAPP, Anna Carolina. **Trabalhadores informais chegam a 10 milhões no País**. 2016. Disponível em: <<http://economia.estadao.com.br/noticias/geral,trabalhadores-informais-chegam-a-10-milhoes-no-pais,10000071200> />. Acesso em: 05 set 2017.

PIMENTA, Renata Faria. **Implantação de controle de estoque em uma clínica odontológica**: o caso da sorriso & Cia. Ltda. Ouro Preto, MG: Universidade Federal de Ouro Preto (UFOP), 2003.

POZO, H. **Administração de recursos materiais e patrimoniais**: uma abordagem logística. 6. ed. São Paulo: Atlas, 2008.

PREZOTTO, Ezequiel; BONIATI, Bruno. **Estudo de Frameworks Multiplataforma Para Desenvolvimento de Aplicações Mobile Híbridas**. 2014. Disponível em: <<http://www.eati.info/eati/2014/assets/anais/artigo8.pdf> />. Acesso em: 27 jun 2018.

RAIMUNDO, Marcos Rosso. **Gestão de recursos materiais**: controle de estoque de um supermercado localizado em Criciúma – SC. 2011. 119 p. Monografia do Curso de Administração, da Universidade do Extremo Sul Catarinense – UNESC, Criciúma.

RODRIGUES, Isis. **Indústria de gelados comestíveis no Brasil**. Informe Agropecuário, Belo Horizonte, 2015. Disponível em: <<http://cozinhafitefat.com.br/wp-content/uploads/2017/01/aqui-3.pdf> />. Acesso em: 01 set 2017.

SANTANA, Silvio Leonardo. **Gestão de estoque**: um estudo de caso numa indústria alimentícia. 2014. 100 f. Trabalho de Conclusão de Curso (Especialização) - Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2014.

SCHERER, Marinez. **Gestão de Praias no Brasil**: Subsídios para uma reflexão. Revista de Gestão Costeira Integrada - Journal of Integrated Coastal Zone Management, 2013. Disponível em: <<http://www.redalyc.org/articulo.oa?id=388340140001> />. Acesso em: 25 jun 2018.

SILVA, André. **Salebox**: Aplicativo de vendas. São Paulo, 2016. Disponível em: <<https://itunes.apple.com/br/app/salebox-o-aplicativo-de-vendas/id1175764051?mt=8&ign-mpt=uo%3D4> />. Acesso em: 05 set 2017.

SLACK, N; CHAMBERS, S; JOHNSTON, R. **Administração da produção**. 3. ed. São Paulo: Atlas, 2009.

APÊNDICE A – Descrição dos Casos de Uso do sistema web

Este Apêndice apresenta a descrição dos casos de usos UC08 e UC10 do sistema web. O primeiro caso de uso descrito é o UC08 - Movimentar o estoque dos produtos que se refere a rotina em que o gestor do estabelecimento pode adicionar entradas ou saídas de estoque de determinado produto. O Quadro 11 explica mais detalhadamente o caso de uso.

Quadro 11 – Descrição do caso de uso Movimentar o estoque dos produtos (UC08)

Caso de Uso	Movimentar o estoque dos produtos
Descrição	O sistema web deve permitir a movimentação de entrada e saída de produtos
Ator	Gestor
Pré-condição	O gestor deverá estar autenticado
Fluxo principal	<ol style="list-style-type: none"> 1. O gestor acessa a tela de movimentação de estoque; 2. O gestor seleciona o produto que deseja movimentar; 3. O gestor informa o tipo da movimentação; 4. O gestor informa a quantidade a ser movimentada; 5. O gestor finaliza a movimentação do estoque do produto.
Fluxo alternativo	<ol style="list-style-type: none"> 4. O gestor informa uma quantidade inválida para um produto; 5. O sistema informa que a quantidade informada não é permitida; 6. O gestor informa uma quantidade válida.
Pós-condição	O gestor realiza a movimentação do estoque do produto.

Fonte: elaborado pelo autor.

Conforme o Quadro 11, para que seja possível o cadastro da viagem o gestor necessita efetuar o *login* no sistema. Estando autenticado, o gestor acessa a tela de movimentação de estoque e seleciona qual o produto que deseja movimentar o estoque. Na tela seguinte é informado se é uma entrada ou saída do produto e a quantidade. Caso a movimentação seja de saída e o gestor informar uma quantidade maior do que o produto tem em estoque o sistema informa qual o máximo permitido. Com os dados todos validados é possível realizar a movimentação de estoque do produto.

O segundo caso de uso descrito é o UC10 - Cadastrar viagem que se refere ao momento em que o gestor inicia uma nova viagem para um determinado ambulante. O Quadro 12 detalha os passos para que a viagem seja cadastrada.

Quadro 12 – Descrição do caso de uso Cadastrar viagem (UC10)

Caso de Uso	Cadastrar viagem
Descrição	O sistema web deve permitir o cadastro de viagens.
Ator	Gestor
Pré-condição	O gestor deverá estar autenticado
Fluxo principal	<ol style="list-style-type: none"> 1. O gestor acessa a tela de cadastro de viagens; 2. O gestor seleciona o ambulante que irá realizar a viagem; 3. O gestor informa os produtos que o ambulante está levando; 4. O sistema realiza o cálculo do valor dos produtos informados; 5. O gestor informa o valor que está sendo levado pelo ambulante; 6. O gestor realiza o cadastro da viagem.
Fluxo alternativo	<ol style="list-style-type: none"> 3. O gestor informa uma quantidade inválida para um produto; 4. O módulo servidor retorna o erro ao aplicativo; 5. O gestor informa uma quantidade válida.
Pós-condição	<p>O gestor iniciou uma viagem para o ambulante.</p> <p>O ambulante pode visualizar os produtos no cardápio aplicativo móvel.</p>

Fonte: elaborado pelo autor.

Conforme o Quadro 12, para que seja possível o cadastro da viagem o gestor necessita efetuar o *login* no sistema. Estando autenticado, acessa-se a tela de cadastro de viagens, informa qual o ambulante que estará realizando a viagem e informa quais são os produtos que estão sendo levados. Caso o gestor informe uma quantidade maior do que o produto tem disponível em estoque o sistema irá alertá-lo e solicitar que informe um novo valor. Com todos os dados preenchidos e validados é permitido o cadastro da viagem.

APÊNDICE B – Descrição dos Casos de Uso do aplicativo móvel

Este Apêndice apresenta a descrição dos casos de usos UC02 e UC03 do aplicativo móvel. O primeiro caso de uso descrito é o UC02 - Confirmar vendas referente ao aplicativo móvel desenvolvido. Este caso de uso detalha os passos a serem seguidos para que o ambulante consiga confirmar a venda de seus produtos.

Quadro 13 – Descrição do caso de uso Confirma Vendas (UC02)

Caso de Uso	Cadastrar viagem
Descrição	O aplicativo permitir a venda de produtos em tempo real.
Ator	Ambulante
Pré-condição	O ambulante deverá estar autenticado
Fluxo principal	<ol style="list-style-type: none"> 1. O ambulante informa a quantidade que deseja vender de um produto; 2. O ambulante clica sobre o carrinho de vendas; 3. O sistema solicita confirmação da venda; 4. O ambulante confirma a venda.
Fluxo alternativo	<ol style="list-style-type: none"> 1. O ambulante informa uma quantidade inválida de um produto; 2. O aplicativo alerta que não se pode vender mais do que o disponível; 3. O ambulante informa uma quantidade válida.
Pós-condição	O sistema confirma a venda do produto.

Fonte: elaborado pelo autor.

Conforme o Quadro 13, o ambulante deverá estar autenticado no aplicativo móvel para realizar a confirmação da venda dos produtos. Ao efetuar o *login* no aplicativo é apresentado todos os produtos com suas quantidades disponíveis a venda. Ao confirmar a venda de um produto, o registro da venda do produto fica salvo tanto para o gestor quanto para o ambulante, que no momento em que finalizar sua viagem e voltar ao estabelecimento, irá validar as informações junto ao aplicativo. O fluxo alternativo refere-se a ocasião em que o ambulante tenta vender uma quantidade inválida para a confirmação de venda. Ao informar uma quantidade maior que a quantidade disponível o sistema faz a notificação e solicita que informe um valor válido para a operação.

O próximo caso de uso descrito é o UC03 - Consultar viagens realizadas que disponibiliza ao ambulante um histórico de todas viagens realizadas. O Quadro 14 detalha os passos necessários se ter acesso ao histórico das viagens.

Quadro 14 – Descrição do caso de uso Consultar viagens realizadas (UC03)

Caso de Uso	Consultar viagens realizadas
Descrição	O aplicativo deve emitir relatório de viagens realizadas.
Ator	Ambulante
Pré-condição	O ambulante deverá estar autenticado
Fluxo principal	1. O ambulante acessa o menu do aplicativo; 2. O ambulante clica a opção Viagens; 3. O sistema exibe as viagens realizadas pelo ambulante.
Fluxo alternativo	3. O ambulante não possui viagens cadastradas; 4. O aplicativo informa em tela que não possui viagens cadastradas.
Pós-condição	O aplicativo exibe as viagens realizadas pelo ambulante.

Fonte: elaborado pelo autor.

Conforme o Quadro 14, o ambulante deverá estar autenticado no aplicativo móvel para realizar a consulta de suas viagens realizadas. Ao efetuar o *login*, por meio do menu disponibilizado ao lado superior esquerdo do aplicativo, acessa a opção Viagens onde estarão listadas suas viagens com as informações das mesmas. O relatório está dividido entre as viagens em andamento e as viagens finalizadas. Pode-se observar neste relatório para qual loja a viagem é realizada, a data de início, a quantidade de itens e a quantidade em dinheiro para troco e afins levado e os valores totais vendido.