

N. CLASS.	M005.1
CUTTER	A 4740
ANO/EDIÇÃO	2015

**CENTRO UNIVERSITÁRIO DO SUL DE MINAS - UNIS-MG**  
**BACHARELADO EM SISTEMAS DE INFORMAÇÃO**  
**TONIMUILLER ALVES**

**DESENVOLVIMENTO DE UM SISTEMA MÓVEL PARA RASTREAMENTO DE**  
**CARGAS E FRETES: IBFrete**

**Varginha/MG**  
**2015**



**TONIMULLER ALVES**

**DESENVOLVIMENTO DE UM SISTEMA MÓVEL PARA RASTREAMENTO DE  
CARGAS E FRETES: IBFrete**

Monografia apresentada ao curso de Bacharelado em  
Sistemas de Informação do Centro Universitário do Sul  
de Minas – UNIS/MG como pré-requisito para obtenção  
do grau de bacharel, sob orientação do Prof. Rodrigo  
Franklin Frogeri.

**Varginha/MG  
2015**



**TONIMULLER ALVES**

**DESENVOLVIMENTO DE UM SISTEMA MÓVEL PARA RASTREAMENTO DE  
CARGAS E FRETES: IBFrete**

Monografia apresentada ao curso de Bacharelado em Sistemas de Informação do Centro Universitário do Sul de Minas – UNIS/MG, como pré-requisito para obtenção do grau de bacharel pela Banca Examinadora composta pelos membros:

Aprovado em     /     /

---

Prof. Msc. Rodrigo Franklin Frogeri

---

Prof. Esp. Rodrigo Gomes da Silva

---

Prof. Msc. Rodrigo Franklin Frogeri

OBS.:

Dedico este trabalho primeiramente ao Deus criador de todas as coisas por me proporcionar esta conquista. Também dedico este à minha família e ao meu orientador por ter me apoiado prontamente em todas as ocasiões.

## RESUMO

O transporte rodoviário é de fato um dos fatores mais importantes para o sucesso do setor de logística de um país. Portanto, este trabalho tem como objetivo mostrar os passos para o desenvolvimento de um sistema para gerenciamento de cargas e fretes, visando ser mais uma ferramenta para o setor. Serão apresentadas a escolha das ferramentas e metodologias necessárias para o desenvolvimento do sistema. O sistema será composto por um sistema web, aplicativo para a plataforma *Android* e *WebServices*, visto que se tratam de softwares construídos com ferramentas e tecnologias diferentes. Será abordada a integração das partes para que o sistema funcione harmoniosamente. E com base em pesquisas bibliográficas que darão subsídio teórico ao desenvolvimento do sistema, será possível desenvolver as funcionalidades requeridas para o sistema.

**Palavras-chave:** Sistemas. Aplicativos. Sistemas Web. *Android*. *WebServices*.

## ***ABSTRACT***

Road transport is indeed one of the most important factors for the success of the logistics industry of a country. Therefore, this work aims to show the steps for developing a system for managing cargo and freight, in order to be more a tool for the sector. It is presented the choice of tools and methodologies needed to develop the system. The system will consist of a web system, application for the Android platform and Web Services. It is seen that these are software built with different tools and technologies will address the integration of the parties so that the system works smoothly. And based on library research that will benefit the development of the theoretical system, you can develop the functionality required for the system.

**Keywords:** Systems. Applications. Web systems. Android. Web Services.

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	<b>7</b>
1.1	Tema específico .....	8
1.2	Problema de pesquisa.....	8
1.3	Problematização.....	8
1.4	Hipóteses.....	9
1.5	Objetivo geral .....	9
1.6	Objetivos Específicos .....	9
1.7	Justificativa .....	10
1.8	Estudo de Viabilidade.....	10
<b>2</b>	<b>REVISÃO DA LITERATURA</b> .....	<b>14</b>
2.1	PHP .....	14
2.2	WebServices .....	16
2.3	Android .....	18
2.4	Linguagem de programação Java.....	21
2.5	SGBD - MySQL .....	22
<b>3</b>	<b>MATERIAL E MÉTODOS</b> .....	<b>24</b>
<b>4</b>	<b>RESULTADOS</b> .....	<b>29</b>
<b>5</b>	<b>CONCLUSÃO</b> .....	<b>42</b>
	<b>REFERÊNCIAS</b> .....	<b>43</b>
	<b>APÊNDICE A – ESPECIFICAÇÃO DE REQUISITOS DO SISTEMA IBFRETE.</b> .....	<b>45</b>
	<b>APÊNDICE B - DIAGRAMAS DE CASO DE USO</b> .....	<b>47</b>
	<b>APÊNDICE C - DIAGRAMA DE CLASSES</b> .....	<b>49</b>
	<b>APÊNDICE D - DIAGRAMA ENTIDADE RELACIONAMENTO</b> .....	<b>50</b>

## 1 INTRODUÇÃO

O transporte rodoviário de cargas é o modal de logística que mais movimenta produtos no país. Os transportes rodoviários realizados por caminhoneiros são responsáveis por cerca de 58% do transporte de produtos primários (RIBEIRO; DAMASCENO, 2013).

O DNIT qualifica o transporte rodoviário nacional com algumas características. Dentre elas podemos citar algumas como (DNIT, 2014):

- Possuir a maior relevância entre os modais existentes;
- Alto custo de manutenção;
- Flexibilidade com extensão da malha;
- Transporte com velocidade moderada;
- Custos altos para grandes distâncias, integra todos os estados brasileiros;

Esses dados revelam a importância dos transportes rodoviários para a economia nacional. Buscar desenvolver ferramentas para que o transporte possa ser efetuado com mais eficiência pode refletir em ganhos em diversos aspectos para as organizações e pessoas envolvidas com o transporte rodoviário.

Este trabalho tem como finalidade descrever o processo de desenvolvimento de um sistema de rastreamento de cargas e fretes. Visto que hoje existe uma grande necessidade do transporte de mercadorias pelo mundo todo, uma ferramenta que possa auxiliar as empresas e os profissionais do ramo de transportes pode ser de grande valor. E também, a difusão dos smartphones e do acesso à internet, podem subsidiar a criação de sistemas de informação que permitem que informações de valor possam ser acessadas de praticamente qualquer lugar.

Os dispositivos móveis atualmente são capazes de processar e armazenar uma quantidade de informação considerável, o que permite aos seus usuários possuir informação relevante de forma bastante prática. Além disso, hoje existem recursos embutidos nos dispositivos móveis que aumentam ainda mais as possibilidades do desenvolvimento de aplicativos úteis em diversas tarefas do cotidiano. Um bom exemplo é o GPS<sup>1</sup>, que é de grande ajuda na localização de pessoas e locais, facilitando a locomoção.

---

<sup>1</sup> GPS é a sigla para Sistema Global de Posicionamento. Trata-se de um sistema de navegação à rádio que permite que pessoas na terra, mar e ar determinem sua localização exata, velocidade e horário em qualquer lugar do mundo. Fonte: <http://www.gis2gps.com/GPS/GPSDEF/gpsdef.html>

O acesso à internet por dispositivos móveis também está bastante difundido. Abrindo possibilidade para que informações possam ser consumidas e enviadas de qualquer lugar onde haja conexão de dados disponível. Com isso é possível que sistemas web sejam acessados por aplicativos executando em dispositivos móveis.

Os sistemas web permitem que uma grande quantidade de dados sejam armazenados, processados e também posteriormente acessados de qualquer lugar. Podendo ser acessados por computadores e dispositivos móveis.

Para integrar dois sistemas diferentes em suas estruturas, como um sistema para dispositivos móveis e um sistema web, é necessário o uso de WebServices<sup>2</sup>. Os WebServices serão responsáveis por receber e disponibilizar informações de forma transparente usando de padrões previamente testados e com diretrizes e padrões definidos. Estes WebServices são softwares que tem como função integrar sistemas computacionais independente da tecnologia empregada no desenvolvimento de cada sistema (W3C, 2014).

### **1.1 Tema específico**

Desenvolvimento de um sistema de informação móvel para rastreamento de cargas e fretes.

### **1.2 Problema de pesquisa**

Qual o processo de desenvolvimento de um sistema de informação móvel para o gerenciamento de entrega de cargas e fretes?

### **1.3 Problematização**

Para desenvolver o sistema é necessário escolher uma plataforma para dispositivos móveis alvo. Visto que hoje existem diversas plataformas e cada uma contendo características distintas, cada plataforma pode se adequar melhor a algumas circunstâncias específicas.

---

<sup>2</sup> WebService é um software que executa em um computador servidor e tem como objetivo auxiliar a integração entre softwares desenvolvidos com tecnologias diferentes.

Além disso o sistema contará com um aplicativo web que disponibilizará às empresas um cadastro de fretes. Então será necessário desenvolver o aplicativo web e também modelar e implementar o banco de dados que irá armazenar as informações.

O aplicativo desenvolvido buscará os fretes cadastrados no aplicativo web. Para isso serão usados WebServices que irão realizar consultas no banco de dados onde os dados estão armazenados e retornar os dados para o aplicativo.

#### **1.4 Hipóteses**

Os sistemas, para dispositivo móvel e web, podem se integrar trocando dados entre si utilizando WebServices, mesmo sendo construídos utilizando tecnologias completamente distintas.

O aplicativo será desenvolvido para o sistema operacional Android, utilizando a linguagem Java. O Webservice será desenvolvido utilizando a linguagem PHP. Eles poderão se integrar utilizando o padrão arquitetural REST, trocando dados no formato JSON.

Os dados do usuário do aplicativo, armazenados em um banco de dados no dispositivo, sempre serão atualizados no banco de dados do aplicativo web quando uma conexão com a internet estiver disponível no momento da execução do aplicativo.

#### **1.5 Objetivo geral**

Desenvolver um sistema para controle e busca de fretes que seja composto por um sistema web, um sistema móvel, além da integração do sistema móvel e do sistema web via Webservice.

#### **1.6 Objetivos Específicos**

- Analisar as tecnologias disponíveis para desenvolvimento dos sistemas;
- Desenvolver os sistemas web, móvel e o Webservice;
- Integrar os sistema móvel e web ao Webservice.

## 1.7 Justificativa

Os dispositivos móveis processam e armazenam cada vez mais dados. E também os sistemas para internet são cada vez mais usados pelas pessoas e organizações. A possibilidade de integração dessas duas tecnologias permite que informações sejam acessadas de diversos lugares e em formatos diversificados de acordo com as necessidades.

Porém, como trata-se de tecnologias heterogêneas, sistemas para dispositivos móveis e sistemas para internet, é necessário que haja um intermediador que faça a integração ou comunicação entre esses sistemas. O uso de WebServices para fazer a função de intermediação entre sistemas heterogêneos de diversos tipos está cada dia mais difundido.

Portanto, este trabalho será motivado pela crescente utilização de dispositivos móveis com um alto poder de processamento e armazenamento e também de sistemas para internet. Tendo como foco como integrar esses sistemas utilizando WebServices.

## 1.8 Estudo de Viabilidade

Já existem disponíveis na loja oficial de aplicativos para Android, alguns aplicativos que propõem satisfazer a demanda de busca por fretes e cargas. Entre eles podemos citar: Fretebras, Quero Frete, TruckPad, Sontra Cargo e Frete Livre Web.

Todos os aplicativos citados são semelhantes em suas funcionalidades no geral, mas cada um possui características únicas que se mostram um diferencial. A seguir é apresentado pontos positivos e negativos dos aplicativos avaliados.

Alguns pontos positivos do aplicativo Fretebras são:

- Um grande número de usuários, tanto empresas que buscam caminhões para realizar fretes quanto donos de meios de transporte que buscam cargas para transportar.
- Interface bonita e de fácil utilização;
- Uma vez que o usuário informe seus dados de identificação não é necessário informar novamente ao abrir o aplicativo, isso agiliza o processo de busca de cargas;
- A empresa pode entrar em contato com o dono do meio de transporte por meio do WhatsApp<sup>3</sup>;

---

<sup>3</sup> WhatsApp é um aplicativo de mensagens multi-plataforma que permite trocar mensagens pelo celular sem pagar por SMS.

No entanto, o aplicativo Fretebras apresenta um ponto negativo:

- Para que o motorista veja as cargas disponíveis ele deve realizar o check-in, informando que está disponível em um determinado local. Para realizar o check-in deve ser informado o estado e a cidade que são escolhidos por meio de caixas de seleção, o que pode não ser tão prático e também pode ser lento se comparado à possibilidade de digitar uma determinada cidade e o aplicativo auto completar o valor para o usuário. Se o usuário escolher o estado de Minas Gerais e depois tiver que encontrar a cidade de Varginha por exemplo terá que rolar por várias cidades antes de escolher a opção;

Com relação ao aplicativo Quero Frete, são considerados pontos positivos:

- Possui bastantes usuários, tanto donos de meio de transportes quanto empresas que disponibilizam fretes;
- Agrega as mesmas características do aplicativo Fretebras e ainda permite que as pesquisas de cargas sejam feitas digitando as cidades livremente, e não usando caixas de seleção como no caso do Fretebras;
- Permite que haja uma comunicação entre as empresas e os caminhoneiros dentro do aplicativo;
- A empresa tem em tempo real, a localização do frete enquanto o transporte está sendo realizado;
- O usuário pode criar uma lista de fretes favoritos, análogo aos favoritos que existem nos navegadores de internet;
- Possibilidade de se buscar fretes de forma livre, ou seja, fretes que não se enquadram nas características do veículo cadastrado no aplicativo possibilitando que um caminhoneiro possa buscar fretes para um outro caminhoneiro que não tenha acesso ao aplicativo e que tenha um veículo com características diferentes;

Porém, o aplicativo Quero Frete:

- não permite que as cargas sejam filtradas por uma cidade de destino específica, somente por estados, listando todas as cargas e fretes de um estado ou região;

Enumerando os pontos positivos do aplicativo TruckPad, temos:

- Assim como os outros aplicativos anteriormente avaliados, é bastante prático no que se refere à identificação do usuário. Uma vez identificado, ao abrir o aplicativo ele não necessita que as informações de identificação sejam informadas novamente;
- Ao se cadastrar nesse aplicativo o usuário informa as características do seu veículo. Baseado nessas características o aplicativo busca as cargas disponíveis de acordo com a posição atual do GPS;
- O TruckPad além de dar detalhes do frete, tenta estimar os custos da viagem incluindo gastos com combustível e pedágios;

Esse aplicativo apresenta, no entanto, dois pontos negativos:

- Esse aplicativo, assim como o Fretebras, usa na pesquisa de fretes, para informar o local de origem, caixas de combinação, o que torna a entrada de dados lenta, mais uma vez se o usuário necessitar procurar uma carga com origem em Varginha – MG, ele terá que escolher o estado Minas Gerais e depois rolar a caixa de combinação na cidade Varginha;
- Não é possível filtrar os fretes por cidade de destino, somente por estados;

Os pontos positivos do aplicativo Sontra Cargo são:

- Os fretes podem ser buscados tanto por cidade como estado tanto na origem quanto no destino;
- Permite ao usuário digitar de maneira livre os dados e também com função de auto completar os dados;
- Oferece uma estimativa de custo de pedágios do trajeto que será realizado;
- Os fretes podem ser buscados independentemente do tipo de veículo que foi cadastrado pelo usuário;
- Notifica o usuário quando um frete que se adequa às características do veículo cadastrado, mesmo quando o aplicativo não está aberto na tela do dispositivo;

Como ponto negativo, o aplicativo Sontra Cargo:

- lista os fretes que estão a até 200 km de raio do destino e da origem, não dando ao usuário a opção de um raio diferente ou até mesmo desconsiderar o raio, buscando somente nas localidades de origem e destino;

Ponto negativo do aplicativo FreteLivreWeb:

- O aplicativo FreteLivreWeb assim como todos os outros, possui uma interface web também. E essa interface foi a única possível de ser testada, pois o aplicativo para Android não se adequa a todos os tamanhos de telas de dispositivos. Visto que o dispositivo em que foram realizados os testes possui uma tela de 240X320 pixels, alguns controles não apareceram corretamente, inviabilizando o uso do aplicativo.

## 2 REVISÃO DA LITERATURA

### 2.1 PHP

PHP (*Hypertext Preprocessor*) é uma linguagem de programação altamente difundida e utilizada em diversas aplicações em sistemas de informações computacionais. Sua primeira versão foi lançada em 1994 e ela foi criada por Rasmus Lerdorf. Segundo Lerdorf et. al (2013), PHP é uma simples e poderosa linguagem de programação, projetada para criar programas que serão executados em servidores, com o intuito de gerar conteúdo dinâmico na internet.

Além disso, pode ser utilizada para criar programas que rodam sobre interface de linha de comando e também para criar programas com interface gráfica que rodam localmente no computador.

Para gerar conteúdo dinâmico da web com PHP é necessário rodar os scripts em um servidor que tenha instalado um interpretador para a linguagem. O conteúdo que pode ser gerado pode ser documentos XML, gravuras, animações em flash, arquivos PDF e muitos outros. Além disso o PHP pode ser usado para rodar programas em linha de comando.

Os programas em linha de comando escritos em PHP podem ser utilizados para tarefas administrativas e também para análise de arquivos de logs dos servidores. Porém, muitas vezes programas que rodam em linha de comando não são adequados por isso a linguagem PHP também suporta a construção de programas que tenha interface gráfica.

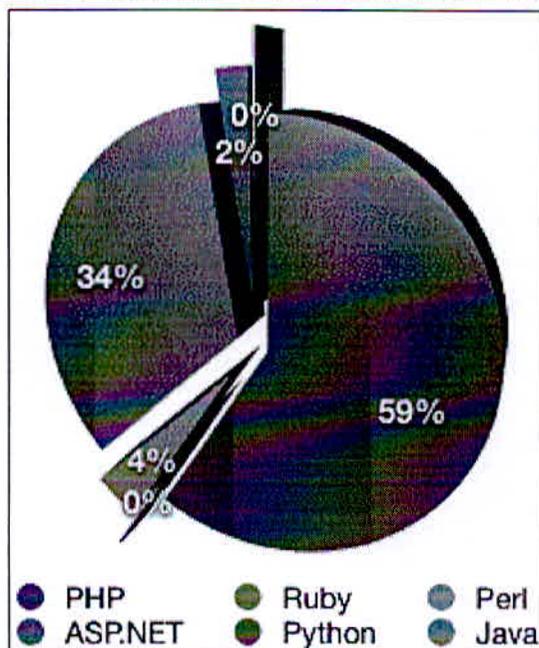
Os programas com interface gráfica escritos em PHP usam bibliotecas como o PHP-Gtk e rodam em mais de um sistema operacional sem a necessidade de alterar o código fonte. Isso porque os programas escritos em PHP podem ser executados no Windows, Linux e Mac OS X.

Outra característica relevante da linguagem PHP é o suporte a diversos sistemas gerenciadores de bancos de dados. Dentre eles estão o MySQL, SQL Server, PostgreSQL, Oracle, Sybase, Firebird, DB2 e outros.

Finalmente, o PHP oferece uma biblioteca de funções básicas para tarefas comuns como, conexão com bancos de dados, tratamento de erros, manipulação de arquivos entre outros (LERDORF et. al, 2013).

A linguagem PHP é a linguagem mais utilizada na atualidade para gerar conteúdo na web. Na figura seguinte é possível notar a distribuição das tecnologias utilizadas:

Figura 1 - Distribuição do uso de linguagens para web.



Fonte: IMASTERS, 2011.

De acordo com o gráfico o PHP é a linguagem de programação mais utilizada na web. Ficando à frente de plataformas como ASP.NET da Microsoft e Java da Oracle.

Potencier (2012) afirma que os três maiores sistemas gerenciadores de conteúdo do mercado – WordPress, Joomla e Drupal são desenvolvidos utilizando PHP. E isso contribui para o aumento do uso da linguagem.

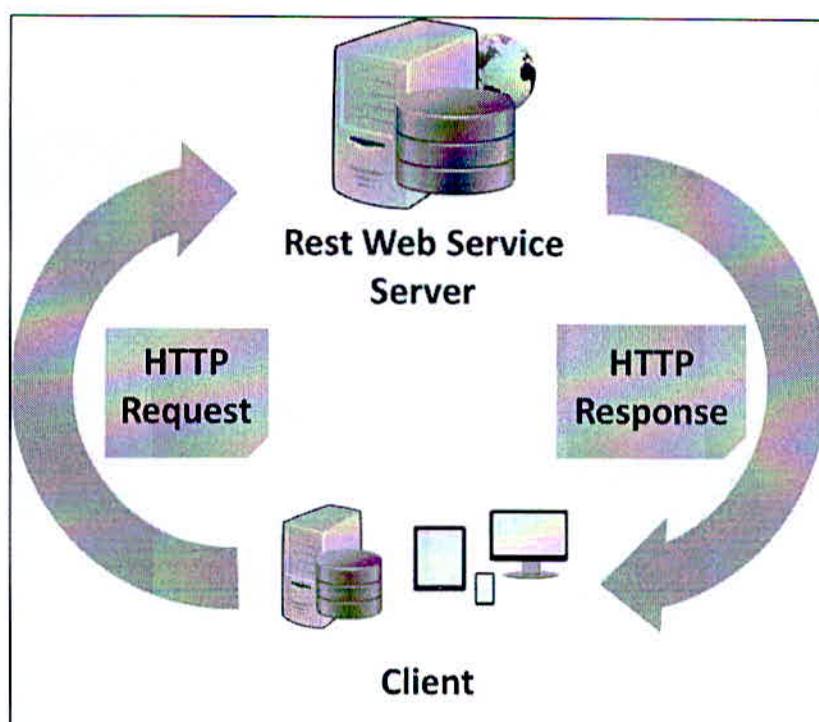
Além da popularidade, Bento (2013) ainda aponta outros fatores que contribuem para a escolha do PHP em projetos web. São eles:

- PHP nasceu para web e sua integração com servidores web é simples;
- Curva de aprendizagem pequena;
- É uma tecnologia livre;
- Facilidade em encontrar serviços de hospedagem;
- Os serviços de hospedagem para PHP são mais baratos se comparados com os serviços de hospedagem para tecnologias semelhantes;

## 2.2 WebServices

WebService é um software desenvolvido para suportar a comunicação entre computadores em uma rede. Os WebServices usam um mecanismo de interface para troca de mensagens que os computadores são capazes de entender. Os sistemas interagem com os WebServices utilizando de padrões de comunicação para internet regulamentados (W3C, 2004). A figura abaixo representa o modelo de funcionamento dos WebServices:

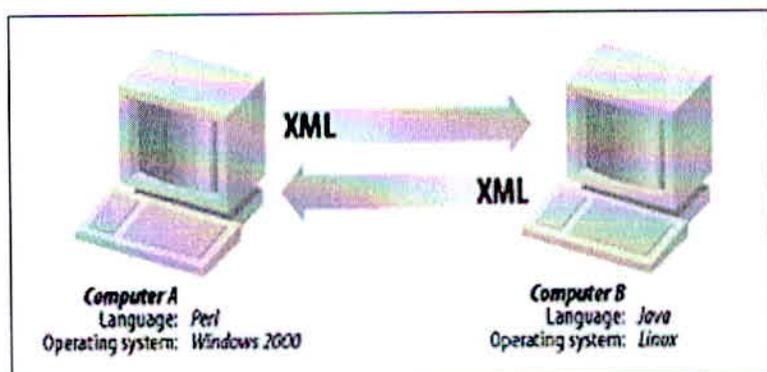
Figura 2 - Funcionamento de um Webservice



Fonte: CHEMAXON, 2015.

O principal objetivo dos WebServices é proporcionar interoperabilidade entre sistemas distribuídos. Independente da tecnologia utilizada na construção desses sistemas, disponibilizando uma melhor interligação das aplicações. Essa interligação tem como objetivo facilitar os processos de negócios, proporcionando a softwares isolados a capacidade de se comunicar com os demais. Esse processo está ilustrado na próxima figura:

Figura 3 - Comunicação entre softwares diferentes



Fonte: IMASTERS, 2006.

Os padrões de comunicação para internet mais utilizados para comunicação com WebServices na atualidade são o protocolo SOAP e o REST.

SOAP é a sigla de *Simple Object Access Protocol*, ou Protocolo Simples de Acesso a Objetos. O protocolo SOAP encontra-se na sua versão 1.2. Segundo o W3C (2007), o SOAP é um protocolo leve para ser utilizado para troca de informação estruturada em um ambiente descentralizado, em ambientes distribuídos. E ele foi projetado para ser utilizado independentemente de qualquer modelo de programação ou padrões de implementação alheios.

As mensagens do protocolo SOAP são compostas utilizando o padrão da linguagem de marcação XML. A sigla XML significa *eXtensible Markup Language*, ou em português, linguagem de marcação extensível. XML é uma linguagem de marcação que tem como função compor documentos que contêm dados e também instruções de como os softwares que interpretam estes documentos devem processar estes dados contidos no documento (W3C, 2008).

Um outro padrão disponível para comunicação com WebServices é o REST (*Representational State Transfer*), que é um estilo arquitetural para comunicação em sistemas distribuídos. Logo REST não se trata de um protocolo assim como SOAP, mas sim de um padrão de comunicação. O REST utiliza o protocolo HTTP para transmitir as informações (FIELDING, 2000).

HTTP é a sigla para *Hypertext Transfer Protocol*, ou Protocolo de Transferência de Hipertexto. O HTTP é um protocolo de nível de aplicação para ser usado em sistemas distribuídos, colaborativos e também que transportam conteúdo multimídia. Trata-se de um protocolo que está em uso na internet desde 1990 (W3C; MIT, 1999).

Diferente do protocolo SOAP, o REST pode utilizar tanto XML quanto JSON ou outros formatos de dados para troca de informações.

JSON é a sigla para *JavaScript Object Notation*, ou Notação de Objeto JavaScript. É um formato de troca e armazenamento de informações leve. É de fácil leitura e escrita para os seres humanos e também para os computadores é um formato fácil de interpretar e gerar. É independente de linguagem de programação podendo ser usado em qualquer ambiente de desenvolvimento de software (ECMA, 2013).

### 2.3 Android

O *Android* é um sistema operacional desenvolvido especialmente para dispositivos móveis como celulares e tablets. É composto por *middlewares* e alguns aplicativos principais como Contatos, Navegador de Internet e Telefone (MONTEIRO, 2013).

Em Outubro de 2003 é fundada em Palo Alto, Califórnia, EUA; a *Android Inc.*, por Andry Rubin, Rich Miner, Nick Sears e Chris White. Com a repercussão que estava proporcionando, o Google resolve adquirir a empresa em Agosto de 2005. Após dois anos de trabalho e várias empresas vendo o real potencialmente que estaria por surgir, é criado em 5 de Novembro de 2007 o Consórcio *Open Handset Alliance* no qual será disposto abaixo mais especificado. Já em 12 de Novembro de 2007 após sete dias do consorcio formado, é liberado o tão aguardado SDK para os desenvolvedores, era a primeira versão de muitos que veriam e estão por vir. A *HTC* com o seu celular Dream (G1) é a primeira empresa a lançar um celular com *Android* em 22 de Outubro de 2008, da qual sua versão era a 1.0. A partir disto, as atualizações começaram a ser constantes. No ano de 2009, foi atualizado o *Kernel* do Linux da versão 2.6.27 para 2.6.29 e a versão do *Android* foi 1.1, 1.5, a partir da versão 1.5 começaram a ser batizadas com nomes de doces americanos, veio ainda o 1.6 e a versão 2.0 encerrou o ciclo de atualizações de data coerente (MONTEIRO, 2013).

Como o *Android* é livre, licenciado sobre a licença *Apache Software Foundation* (ASF) 2.0 (Que é uma licença permissiva de software livre sem *copyleft*) (Torres, 2012), poderia ocorrer inconsistência por parte das empresas, e o que tornaria prejudicial quando fosse haver atualizações para várias versões *Android* de várias empresas, pensando nisso criou-se a *Open Handset Alliance* é uma aliança de várias empresas que se formaram para padronizar a plataforma *Android*.

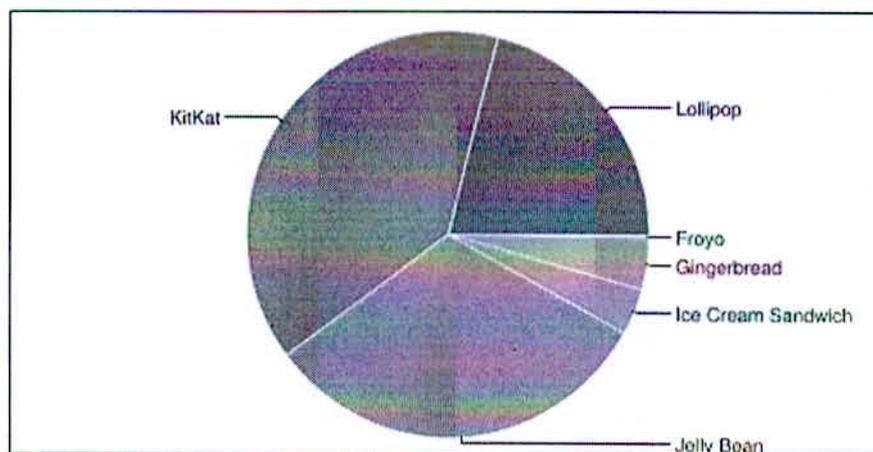
A seguir segue uma imagem, com as empresas que compõem a aliança, separadas por categorias:

Figura 4 - Empresas que compõem a *Open Handset Alliance*.

Operator	Handset Makers	Software Companies	Commercialization Companies	Semiconductor Companies
SoftBank	htc	Ascender Corporation	Aplica Corporation	Audience
Do Co Mo	LG Electronics	Google	BORQS	intel
Sprint	Motorola	Living Image	noser	Intel
vodafone	Samsung	Nuance	tat	NVIDIA
TELECOM	Garmin	SawPop	TELECA	Qualcomm
China Unicom	Huawei	OMRON	WIND RIVER	Synaptics
	Sony Ericsson	EVOX III		Texas Instruments
	acer			AKM
	ASUS			ARM
	TOSHIBA			

Fonte: OPEN HANDSET ALIANCE, 2011.

Atualmente a distribuição de utilização das versões do *Android* pode ser observado no gráfico seguinte:

Figura 5 - Distribuição das versões do *Android*.

Fonte: ANDROID DEVELOPER PORTAL, 2015.

Uma tradição adotada pela Google foi batizar as versões *Android* com nome de doces, a primeira a receber foi a 1.5, o software foi batizado de *Cupcake*. A letra “C”, inicial do doce, é também a terceira do alfabeto. Desde então essa ordem vem sendo mantida e todas as outras versões do SO ganharam nomes de doces: *Donut*, *Eclair*, *Froyo*, *Gingerbread* e *Honeycomb*,

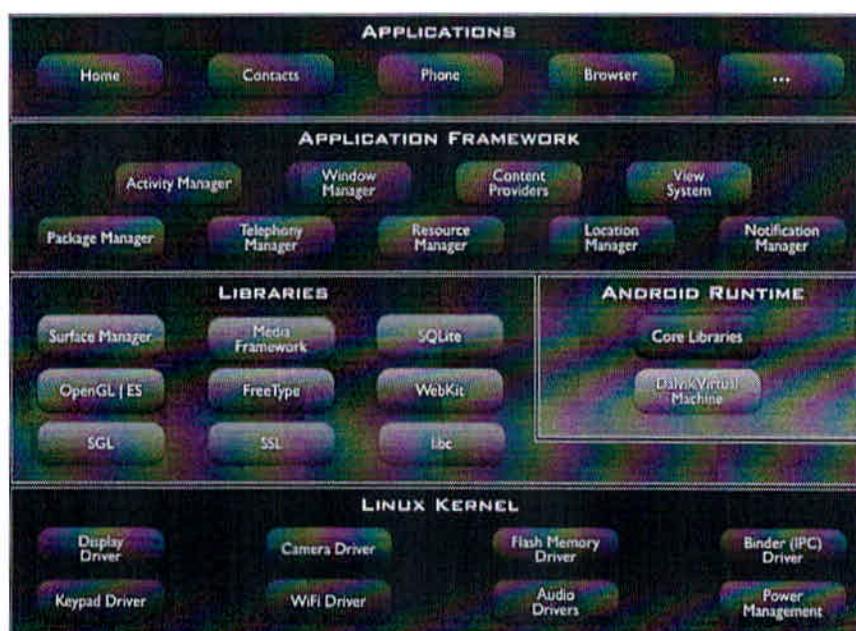
*Ice Cream Sandwich*, que podemos notar que cada palavra à primeira ordem segue a ordem do alfabeto.

- Versão 1.5 – *Cupcake* (Bolo de Caneca)
- Versão 1.6 – *Donut* (Rosquinha)
- Versão 2.0 – *Éclair* (Bomba de Chocolate)
- Versão 2.2 – *Froyo* (Iogurte Congelado)
- Versão 2.3 – *Gingerbread* (Pão de Gengibre)
- Versão 3.0 – *Honeycomb* (Favo de Mel)
- Versão 4.0 – *Ice Cream Sandwich* (Sanduiche de Sorvete)
- Versão 4.1, 4.2 e 4.3 - *Jelly Bean*. (Jujuba)
- Versão 4.4 – *Kitkat*. (Trata-se de um famoso chocolate)
- Versão 5.0 e 5.1 – *Lollipop* (Pirulito)

O sistema operacional Android tem grande relevância no mercado, isso se deve pelo número de dispositivos que são baseados nele, uma API com muitos recursos, além de boas ferramentas de desenvolvimento (MONTEIRO, 2012).

A arquitetura do sistema operacional *Android* e sua plataforma pode ser observada na figura seguinte:

Figura 6 - Arquitetura do sistema operacional *Android*.



Fonte: SÉRGIO PRADO, 2011.

As cinco camadas do sistema operacional *Android* têm as seguintes funções:

- **Camada de aplicação** – Última camada, onde situam-se os aplicativos. Softwares como agendas, telefone, aplicativos de mensagens e etc;
- **Framework de aplicação** – Camada de alto nível que tem como intuito facilitar a produtividade e o desempenho das aplicações nativas e de terceiros. Ela foi projetada visando a reutilização de recursos do sistema operacional;
- **Bibliotecas** – Conjunto de bibliotecas escritas em C/C++ que dão suporte ao Sistema Operacional.
- **Kernel Linux** – Camada responsável por controlar o hardware do dispositivo, se encarrega de controlar o driver Wireless, Áudio Drivers, Câmera Drivers, Bluetooth Driver, além da segurança, gerenciamento de memória e processos.
- **Android Runtime** – Toda aplicação *Android* roda em seu próprio processo, com sua própria instancia da máquina virtual Dalvik. O Dalvik foi escrito de forma a executar várias máquinas virtuais eficientemente. Ele executa os arquivos “.dex”, que é otimizado para consumo mínimo de memória. A máquina virtual é baseada em registros e roda classes compiladas pela linguagem Java que foram transformadas em arquivos “.dex”, através da ferramenta “dx” incluída no SDK. Praticamente todas as bibliotecas oferecidas em Java, são compatíveis no *Android* (PORTAL DA EDUCAÇÃO, 2015).

Ente outros fatores que contribuem para o aumento da popularidade do sistema operacional Android, Monteiro (2012) cita o desenvolvimento de aplicativos utilizando uma linguagem de programação bastante disseminada. A linguagem de programação Java.

## 2.4 Linguagem de programação Java

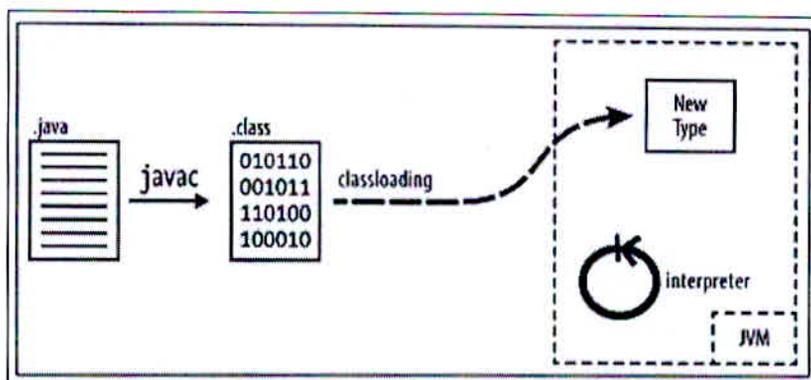
Java é uma linguagem de programação de propósito geral, baseada em classes e orientada a objetos. Ela foi projetada para que o maior número possível de programadores possa adquirir fluência na sua utilização. Ela é baseada nas linguagens de programação C e C++ mas é organizada de maneira diferente, omitindo algumas características de C e C++ e também adicionando algumas características de outras linguagens (GOSLING et. al, 2015).

A linguagem de programação Java é somente uma parte da plataforma Java. Além da linguagem, compõe a plataforma, um ambiente de execução conhecido como *Java Virtual Machine* (JVM) ou Máquina Virtual Java (EVANS; FLANAGAN, 2015).

A JVM é um programa responsável por executar os programas desenvolvidos em Java. Sem a JVM adequada em um determinado sistema operacional, não é possível executar um programa desenvolvido em Java (EVANS; FLANAGAN, 2015).

Os programas executados na JVM não são programas em linguagem Java. Antes de serem executados, os programas desenvolvidos em java são convertidos, ou compilados, para um formato conhecido como *java bytecode*. O *bytecode* é fornecido para a JVM em arquivos com extensão “.class”, e a JVM executa as instruções contidas nesses arquivos (EVANS; FLANAGAN, 2015). A figura 7 mostra o processo de compilação e execução de um programa em Java:

Figura 7 - Compilação e Execução de um programa em Java.



Fonte: EVANS; FLANAGAN, 2015, p. 9.

Uma das características mais notáveis da plataforma Java é descrita pela frase “escreva uma vez, execute em qualquer lugar”. Isso porque os arquivos compilados “class” podem ser executados em qualquer sistema operacional que tenha uma JVM instalada, sem a necessidade de alterar o software para se adequar ao sistema operacional (EVANS; FLANAGAN, 2015).

## 2.5 SGBD - MySQL

Para entender o que é um SGBD primeiro é necessário definir o que é um banco de dados. Heuser (1998) definiu um banco de dados como sendo grandes repositórios compartilhados de dados.

Um SGBD, ou Sistema de Gerenciamento de Banco de Dados, segundo Heuser (1998) é um software que incorpora as funções de definição, recuperação e alteração de dados em um banco de dados.

Um dos sistemas gerenciadores de bancos de dados disponíveis é o MySQL, que é livre e amplamente utilizado. Ele é livre pois além do programador poder utiliza-lo em seus projetos, ele pode contribuir com o desenvolvimento de funcionalidades no MySQL (BENTO, 2013).

Ainda segundo Bento (2013), a ampla utilização do MySQL ocorre pois ele pode ser utilizado juntamente com diversas linguagens de programação e também diversos sistemas operacionais.

O MySQL usa a linguagem *Structured Query Language* ou Linguagem de Consulta Estruturada, que é uma linguagem declarativa utilizada em diversos bancos de dados para definição e consulta de dados (HEUSER, 1998).

### 3 MATERIAL E MÉTODOS

A pesquisa realizada se enquadra nas metodologias pesquisa bibliográfica e prática. Segundo Mascarenhas (2012) a pesquisa bibliográfica busca conteúdo em livros, artigos, dicionários e enciclopédias.

A metodologia de pesquisa bibliográfica foi escolhida porque o desenvolvimento de um software deve se embasar em teorias já estudadas anteriormente e práticas de desenvolvimento de software maduras e bem desenvolvidas. Além disso, uma das vantagens da pesquisa bibliográfica é a grande quantidade de informação que é possível reunir (MASCARENHAS, 2012).

As informações capturadas para a realização da pesquisa foram obtidos através de livros e artigos em sites oficiais e também sites de tecnologia que tratam do assunto de desenvolvimento de software.

Os livros e artigos escolhidos deram o suporte necessário para entender e aplicar os conceitos utilizados no desenvolvimento do software proposto. Os sites proporcionaram informações e tutoriais que auxiliaram na implementação de trechos de códigos do sistema. E também, as informações em sites oficiais foram usadas para motivar o desenvolvimento da ferramenta além de dar segurança quanto à veracidade das informações obtidas.

A pesquisa foi prática pois o seu objetivo foi desenvolver um sistema, envolvendo atividades práticas de implementação do sistema.

#### 3.1 Passos do Desenvolvimento do Sistema

Foi efetuada uma pesquisa em livros e artigos para auxiliar na escolha da metodologia e das tecnologias utilizadas para o desenvolvimento do sistema.

Dentre a bibliografia pesquisada, foram apresentados alguns modelos ou métodos de processo de desenvolvimento de softwares. Um modelo de processo de desenvolvimento segundo Sommerville (2011) é uma representação de um processo de software.

Um modelo de processo de software representa uma parte do processo, não cobrem todos os detalhes do processo, são utilizados para descrever uma abordagem de desenvolvimento de software (SOMMERVILLE, 2011).

Cada modelo de processo de software tem suas características específicas. Portanto foi necessário avaliar a característica de alguns modelos de processo de software disponíveis para aplicação no desenvolvimento do sistema. Dentre os modelos elucidados pela pesquisa estão:

1. *O modelo em cascata*. Esse modelo considera as atividades fundamentais do processo de especificação, desenvolvimento, validação e evolução, e representa cada uma delas como fases distintas, como: especificação de requisitos, projeto de software, implementação, teste e assim por diante.
2. *Desenvolvimento Incremental*. Essa abordagem intercala as atividades de especificação, desenvolvimento e validação. O sistema é desenvolvido como uma série de versões (incrementos), de maneira que cada versão adiciona funcionalidade à anterior.
3. *Engenharia de software orientada a reuso*. Essa abordagem é baseada na existência de um número significativo de componentes reusáveis. O processo de desenvolvimento do sistema concentra-se na integração desses componentes em um sistema já existente em vez desenvolver um sistema a partir do zero. (SOMMERVILLE, 2011, p. 19, 20).

Pressman (2011) ainda cita outros modelos de processo de software:

- **Modelo de processo evolucionário**. É um modelo voltado para assimilar a evolução dos processos de negócio e conseqüentemente a evolução do software.
- **Modelo de processo concorrente**. Este modelo parte do princípio de que qualquer uma das atividades do processo de desenvolvimento de software podem ocorrer paralelamente.

Para o desenvolvimento do sistema proposto na pesquisa foi escolhido o modelo de processo em cascata. Este modelo foi escolhido, pois Pressman (2011, p. 59) afirma que “há casos em que o problema é bem compreendido [...]”. E o sistema proposto tem seu objetivo bem definido, com funções com poucas ou nenhuma possível mudança até o final do projeto.

Os passos fundamentais do modelo de processo em cascata seguem um método sequencial e sistemático. Seus passos passam pelo levantamento de requisitos, projeto do software, modelagem, programação, implantação e manutenção (PRESSMAN, 2011).

O levantamento de requisitos ou especificação de requisitos trata das funcionalidades que o sistema deverá disponibilizar que são informadas pelos clientes ou usuários do sistema (SOMMERVILLE, 2011).

O projeto do software trata das estimativas de custos e também do cronograma para o desenvolvimento do software (PRESSMAN, 2011).

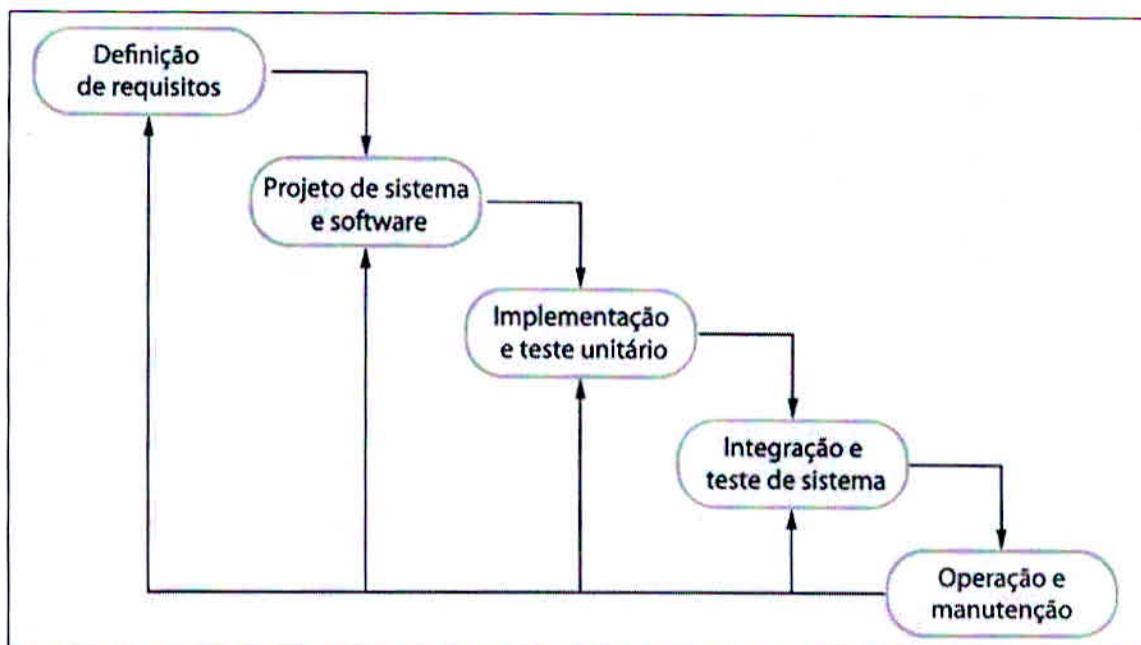
A modelagem do software diz respeito a como o sistema será organizado, sua arquitetura, seus componentes e o relacionamento entre eles. Quais partes irão compor o software e como elas irão se relacionar entre si para o funcionamento do sistema.

A programação é quando o software é codificado ou implementado como programa de computador.

Na implantação o sistema é instalado e colocado em ambiente de produção.

E por fim, a manutenção é o estágio onde os erros não encontrados inicialmente são corrigidos e também a implementação de novos requisitos descobertos (SOMMERVILLE, 2011). A figura abaixo resume o modelo de processo de desenvolvimento em cascata:

Figura 8 - Modelo de processo em cascata.



Fonte: SOMMERVILLE, 2011, p. 20.

Na figura, no primeiro quadrado, está o levantamento de necessidades ou os requisitos do sistema. Esses requisitos são levantados tendo em mente a necessidade dos usuários. Eles são registrados na especificação de requisitos de software.

Segundo Sommerville (2011), o documento de especificação de requisitos de software reúne todos os requisitos que o usuário deseja que o sistema satisfaça. Esse documento deve considerar os requisitos não funcionais e funcionais. E deve ser escrito de maneira clara, inequívoca, de fácil compreensão, completa e consistente.

É possível observar o documento de especificação de requisitos do IBFrete no “Apêndice A” deste documento.

Como foi dito anteriormente, o documento de especificação de requisitos é composto pelos requisitos funcionais e não funcionais.

Os requisitos funcionais são características que o sistema deve ter, funções que ele deve oferecer. Eles descrevem como o sistema deve reagir em determinadas circunstâncias e também como não reagir. Exemplos de requisitos funcionais:

- O sistema deverá permitir o cadastramento de dados pessoais do cliente;
- O sistema deve emitir relatórios gerenciais;
- O sistema deve permitir efetuar baixa automática no estoque quando executar uma venda de produto.

Os requisitos não funcionais se referem às restrições ou limitações do sistema. Podem ser advindos de normas externas e requerimentos de usuários. Geralmente se referem à características do sistema como um todo (SOMMERVILLE, 2011).

Alguns exemplos de requisitos não funcionais:

- O tempo de resposta do sistema não pode ultrapassar 10 segundos;
- O aplicativo deverá ser executado no sistema operacional *Android*;
- O banco de dados utilizado deverá ser o MySQL;

O terceiro quadro da figura 8, é o próximo item de interesse para o desenvolvimento do projeto. Nele é abordado a modelagem do sistema. Para realizar a modelagem do IBFrete foi utilizado o diagrama de caso de uso da UML.

A UML - *Unified Modeling Language* ou Linguagem de Modelagem Unificada é uma linguagem que usa recursos visuais para modelar softwares. Ela pode ser usada na construção de softwares para qualquer propósito em qualquer domínio de aplicação.

A UML não é uma linguagem de programação assim como Java, PHP, C e outras. Trata-se de uma linguagem de modelagem que tem como principal objetivo auxiliar os engenheiros de softwares a definirem as características de um determinado sistema (GUEDES, 2011).

A UML oferece vários diagramas para modelagem de software. A modelagem de um software pode contar com um ou mais dos diagramas oferecidos, ficando à cargo dos desenvolvedores eleger quais os diagramas são relevantes para o seu projeto. Para o desenvolvimento do IBFrete foi escolhido o diagrama de caso de usos.

O diagrama de casos de uso é utilizado para auxiliar o levantamento de requisitos. Embora ele possa ser utilizado em outras fases da modelagem do software servindo como base para outros diagramas. Ele tem como características linguagem simples e de fácil compreensão para que os usuários possam compreender o sistema. Nele são identificados os atores que irão utilizar o sistema. Esses atores podem ser pessoas, outros softwares ou até mesmo algum

dispositivo de hardware. Ele também identifica as funcionalidades que o software deve oferecer aos atores.

O apêndice B contém dois diagramas de caso de uso desenvolvidos para o sistema proposto na pesquisa.

Realizada essas atividades foi implementado o sistema. Como ferramentas para o desenvolvimento do aplicativo foi escolhido o ambiente de desenvolvimento integrado Eclipse Luna que é livre, juntamente com uma extensão para suporte ao desenvolvimento de aplicativos *Android* chamada ADT. A ferramenta Eclipse também foi utilizada para o desenvolvimento do sistema web.

Para a criação do sistema web foi escolhido o software XAMPP. Que se trata de uma coleção de softwares incluindo o servidor para PHP Apache e o banco de dados MySQL. Esse software daria o suporte necessário para simular o ambiente real de execução do sistema desenvolvido, tendo como principal objetivo o testes das funções desenvolvidas.

## 4 RESULTADOS

### 4.1 Análise de Requisitos

Foi realizado o estudo de viabilidade já apresentado no início dessa pesquisa. Com isso, foi possível verificar se o desenvolvimento do sistema era viável e também quais características o sistema poderia oferecer aos usuários como diferencial aos seus concorrentes.

O estudo de viabilidade fez com que a proposta do aplicativo desenvolvido nesta pesquisa enquadre as características positivas encontradas nos aplicativos avaliados e além disso possua alguns diferenciais. Entre eles podemos citar:

- interface adaptável a qualquer tamanho de tela, se valendo de barras de rolagens quando o conteúdo a ser exibido ocupar mais espaço do que o disponibilizado pela tela do dispositivo.
- Opções de busca de fretes detalhada, permitindo ao usuário ter uma lista de cargas disponíveis bastante específica às suas circunstâncias.

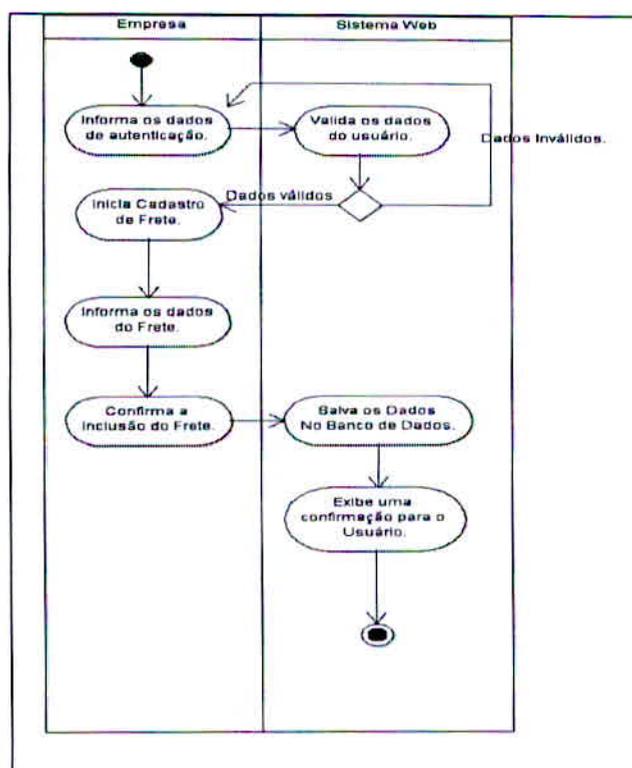
As principais opções são:

- Pesquisar por local de origem e destino. Podendo o destino ser opcional. Nesses campos poderão ser informados estados ou cidades. Serão usados campos de texto livre com recurso de auto completar quando o usuário estiver com a conexão de com a internet ativa.
- Informar um raio de origem e de destino em quilômetros sendo estes opcionais. Que irá filtrar as cargas que estejam dentro do raio de origem ou de destino de acordo com os locais informados nos parâmetros local de origem e local de destino.
- Informar um raio de trajeto em quilômetros. Para que o usuário tenha acesso a cargas a um determinado raio da sua posição atual. Esta informação também será opcional.
- Peso. O usuário poderá definir o peso máximo que a carga deverá ter. Também opcional. Este parâmetro é importante pois o veículo pode estar parcialmente ocupado, podendo levar somente uma quantidade de carga. E essa opção de filtro irá retirar da pesquisa as cargas que têm peso maior do que a capacidade disponível no veículo no momento.

Depois de levantados esses requisitos principais, foi desenhado o diagrama de caso de uso. O diagrama de caso de uso foi útil para o desenvolvimento do sistema, pois permitiu que as funcionalidades que seriam desenvolvidas fosse vista de forma gráfica, aumentando a clareza dos objetivos do sistema. É possível ver os diagramas de caso de uso no apêndice B deste documento.

O sistema web para cadastros dos fretes foi modelado de maneira simples, apenas com uma tela de identificação do usuário, uma tela de listagem de fretes e uma tela de cadastro de fretes. O processo de cadastro de fretes pode ser visualizado no diagrama de sequência seguinte:

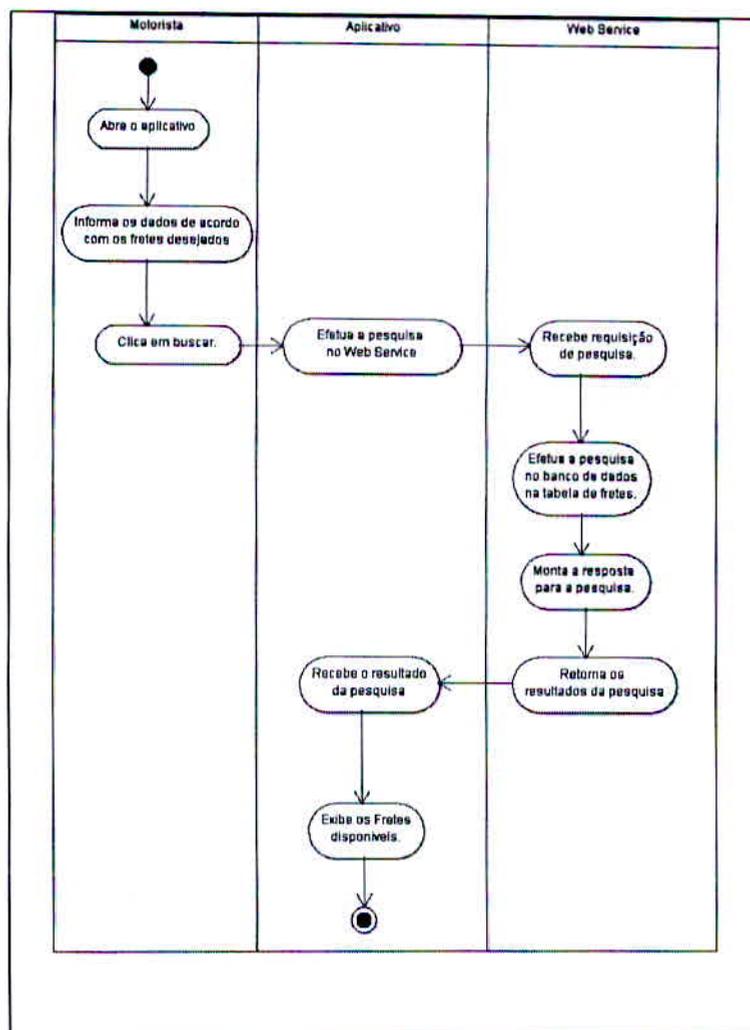
Figura 9 - Cadastro de Fretes



Fonte: Desenvolvido pelo autor.

O aplicativo móvel busca os dados cadastrados no sistema web através de *WebServices*. O fluxo de comunicação entre o aplicativo móvel e os *WebServices* podem ser observados no diagrama de sequência seguinte:

Figura 10 - Comunicação entre o aplicativo e o Webservice



Fonte: Desenvolvido pelo autor.

## 4.2 Implementação

Nessa sessão, serão apresentados as imagens das implementações do sistema e comentado alguns trechos mais relevantes.

Para desenvolver as telas do aplicativo havia duas opções, codificar os elementos manualmente utilizando linguagem XML ou arrastando e soltando utilizando o editor visual do Eclipse. Na próxima imagem é possível ver a implementação da tela de *login* do aplicativo para *Android* em linguagem XML.

Figura 11- Código XML da implementação da interface da tela de *login* do IBFrete.

```

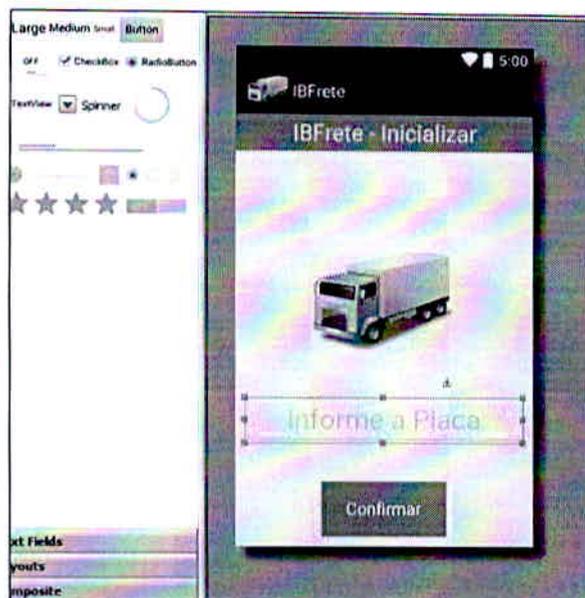
1 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   android:background="@color/CinzaIBFrete">
6
7   <TextView
8     android:id="@+id/tvTituloTela"
9     android:layout_width="match_parent"
10    android:layout_height="wrap_content"
11    android:padding="5dp"
12    android:text="@string/str_Inicializar"
13    android:textColor="@color/White"
14    android:background="@color/AzulIBFrete"
15    android:textSize="24sp"
16    android:gravity="center"/>
17
18   <LinearLayout
19     android:layout_width="wrap_content"
20     android:layout_height="wrap_content"
21     android:layout_centerInParent="true"
22     android:orientation="vertical">
23
24
25     <ImageView
26       android:layout_width="150dp"
27       android:layout_height="wrap_content"
28       android:layout_gravity="center"
29       android:src="@drawable/logo"/>
30
31     <EditText
32       android:id="@+id/editPlaca"
33       android:layout_width="100dp"
34       android:layout_height="wrap_content"
35       android:layout_marginTop="10dp"
36       android:layout_marginBottom="10dp"
37       android:layout_gravity="center"
38       android:inputType="textCapCharacters"
39       android:hint="@string/str_InformarPlaca"
40       android:textColorHint="@color/CinzaEscuroIBFrete"
41       android:textSize="18sp"/>
42
43   </LinearLayout>
44
45   <LinearLayout
46     android:layout_width="wrap_content"
47

```

Fonte: Desenvolvido pelo autor.

A figura 11 representa apenas um trecho do código. O resultado de todo o código de implementação pode ser visto na figura 12. A figura representa o modo visual de criação de telas para aplicativos *Android*.

Figura 12 - Tela de login do aplicativo IBFrete



Fonte: Desenvolvido pelo autor.

Foi necessário desenvolver diversas classes que representassem os dados do domínio da aplicação. O domínio é o conjunto de entidades conceituais que irão compor o aplicativo. Abaixo é possível visualizar a classe Carga que representa os fretes que serão manipulados pelo aplicativo.

Figura 13 - Classe Carga.

```

1 package br.inf.dataaccess.ibfrete.model;
2
3 import java.util.Calendar;
4
5 public class Carga {
6
7     private int idCarga;
8     private TipoCarroceria tipoCarga;
9     private TipoCapacidade tipoCapacidade;
10    private String descricao;
11    private Double peso_volume;
12    private Double valor;
13    private String tipoValor;
14    private String Origem;
15    private String Destino;
16    private char disponibilidade;
17    private Calendar DataHoraMaxDescarga;
18
19
20#    public Carga() {}
34
35#    public Carga(int idCarga, TipoCarroceria tipoCarga, String descricao, Double peso_volume,[]
51
52#    public int getIdCarga() {}
55#    public void setIdCarga(int idCarga) {}
58#    public TipoCarroceria getTipoCarga() {}
61#    public void setTipoCarga(TipoCarroceria tipoCarga) {}
64#    public Double getPeso_volume() {}
67#    public void setPeso_volume(Double peso_volume) {}
70#    public Double getValor() {}
73#    public void setValor(Double valor) {}
76#    public String getTipoValor() {}
79#    public void setTipoValor(String tipoValor) {}
82#    public String getOrigem() {}
85#    public void setOrigem(String origem) {}
88#    public String getDestino() {}
91#    public void setDestino(String destino) {}
94#    public Calendar getDataHoraMaxDescarga() {}
97#    public void setDataHoraMaxDescarga(Calendar dataHoraMaxDescarga) {}
100#    public String getDescricao() {}
103#    public void setDescricao(String descricao) {}
106#    public char getDisponibilidade() {}
109#    public void setDisponibilidade(char disponibilidade) {}
112#    public TipoCapacidade getTipoCapacidade() {}
115#    public void setTipoCapacidade(TipoCapacidade tipoCapacidade) {}
118
119 }
120
121

```

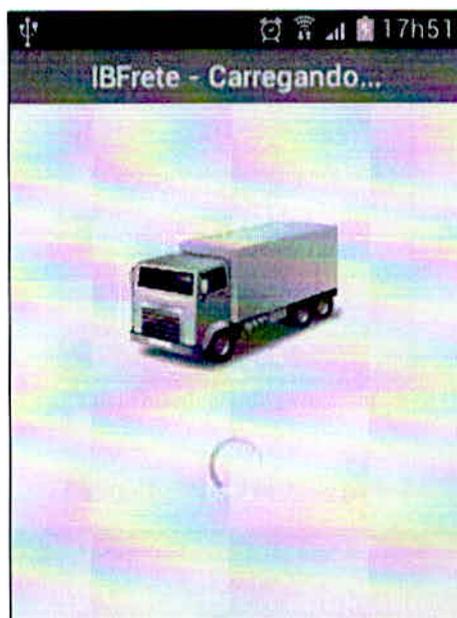
Fonte: Desenvolvido pelo autor.

Na figura podemos ver os atributos ou propriedades da classe que são eles: descrição, peso/volume, valor, Origem, Destino, Data e hora máxima da descarga. Além disso, é na classe é necessário ter os métodos que são responsáveis por manipular os valores dos atributos da classe.

Com a implementação das telas e dos códigos de classes e de eventos das telas foi obtido o aplicativo final. A figura 12 mostra a primeira tela do aplicativo, a tela de identificação do usuário. Nas imagens seguintes serão apresentadas as outras telas do aplicativo desenvolvido.

A próxima figura representa a tela de carregamento do aplicativo.

Figura 14 - Tela de Carregamento



Fonte: Desenvolvido pelo autor.

A tela de carregamento irá aparecer sempre que o aplicativo necessitar consultar dados através da internet.

As próximas duas imagens, são as imagens que representam a tela para buscar os fretes, visto que o dispositivo que foi utilizado para testes tinha uma tela menor do que o conteúdo apresentado na tela foi necessário repartir a tela em duas visualizações, rolando a barra de rolagem presente na tela. Isso prova que o requisito de poder utilizar o aplicativo em dispositivos com telas pequenas foi alcançado.

Figura 15 - Pesquisar Frete parte 1.

The screenshot shows the top portion of a mobile application interface. At the top, there is a status bar with signal strength, Wi-Fi, and battery icons, and the time 17h52. Below the status bar is a dark header with the text "IBFrete - Disponibilidade". The main content area is white and contains the following elements: a checked checkbox labeled "Disponível"; a section titled "Capacidade" with radio buttons for "Total" (selected) and "Parcial", and a numerical input field containing "4.0"; a section titled "Local de Origem" with a text input field containing "R. Joaquim Goularte Júnior, 528"; a section titled "Raio de Carregamento" with a text input field containing "Raio em Km"; a section titled "Raio Trajeto" with a text input field containing "Raio em Km"; and the beginning of a section titled "Local de Destino" with a text input field.

Fonte: Desenvolvido pelo autor.

Figura 16 - Pesquisar Frete parte 2

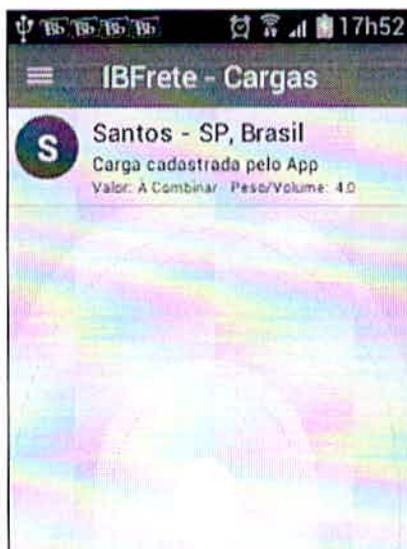
The screenshot shows the bottom portion of the mobile application interface. It continues from the previous screen with the same header and status bar. The main content area is white and contains: the "Raio Trajeto" section with a text input field containing "Raio em Km"; the "Local de Destino" section with a text input field; the "Raio Descarregamento" section with a text input field containing "Raio em Km"; the "Data e Hora de Descarga" section with two text input fields labeled "Data" and "Hora"; and a large, dark blue button at the bottom with the white text "Buscar".

Fonte: Desenvolvido pelo autor.

Para pesquisar fretes o usuário informa a capacidade total que pode carregar de pessoas, ponto de origem e destino de maneira livre, digitando o texto, distâncias de cargas desejadas a partir do ponto de origem ou destino e também do ponto atual do usuário e finalmente, a data e a hora máxima que ele poderá realizar o descarregamento do frete.

A figura a seguir é a tela de listagem de fretes pesquisados. São os fretes retornados pela pesquisa efetuada.

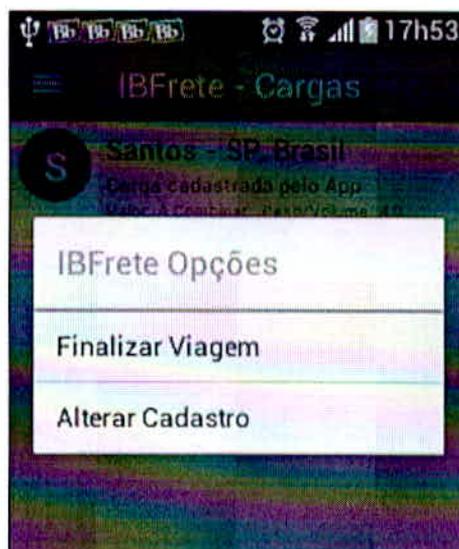
Figura 17 - Listagem de Fretes



Fonte: Desenvolvido pelo autor.

Na figura 17, no canto superior esquerdo da tela o aplicativo conta com um menu que dá opções ao usuário. As opções são as apresentadas pela próxima imagem:

Figura 18 - Opções da tela de listagem de fretes.

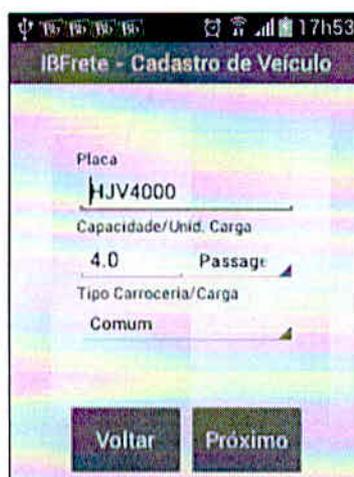


Fonte: Desenvolvido pelo autor.

A opção finalizar viagem volta na tela de pesquisa de frete permitindo que o usuário efetue nova pesquisa. A opção alterar cadastro permite que o usuário altere os seus dados e os dados do veículo cadastrado no aplicativo.

As próximas imagens mostram as telas de alterar os dados do veículo e os dados do motorista.

Figura 19 - Tela para informar dados do veículo.



IBFrete - Cadastro de Veículo

Placa  
HJV4000

Capacidade/Unid. Carga  
4.0 Passage

Tipo Carroceria/Carga  
Comum

Voltar Próximo

Fonte: Desenvolvido pelo autor.

Figura 20 - Tela para informar dados do motorista.



IBFrete - Cadastro de Motorista

\* - Campos Obrigatórios.

Nome do Condutor\*  
Tonimuiller Alves

CNH\*  
123456789

Sexo  
 Masculino  Feminino

Cursos Especiais  
ingles, informatica

Voltar Salvar

Fonte: Desenvolvido pelo autor.

A próxima imagem representa um dos WebServices desenvolvido no sistema para a comunicação com o aplicativo. Trata-se do WebService responsável por receber a requisição de pesquisa de fretes e retornar os fretes cadastrados no banco de dados.

Figura 21 - Webservice para pesquisa de fretes.

```

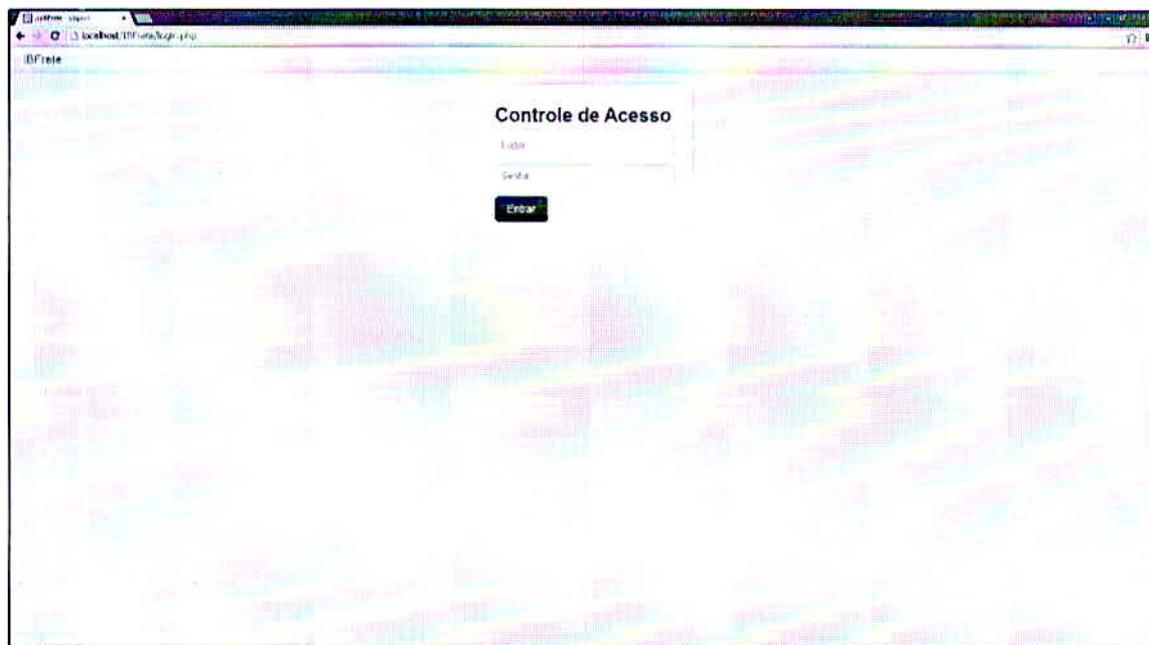
1 <?php
2
3 include("conexao.php");
4 include("CargaClass.php");
5 include("TipoCarrroceriaClass.php");
6 include("LocalClass.php");
7
8 try{
9
10  if(isset($_POST['dados'])){
11      $jsonData = json_decode($_POST['dados']);
12
13      $capacidade = $jsonData->{'capacidade'};
14      $origem = $jsonData->{'origem'};
15      $destino = $jsonData->{'destino'};
16      $dataDescarga = $jsonData->{'dataDescarga'};
17      $horaDescarga = $jsonData->{'horaDescarga'};
18      $tpCarrroceriaId = $jsonData->{'tpCarrroceriaId'};
19      $tpCapacidadeId = $jsonData->{'tpCapacidadeId'};
20      $raioOrigem = $jsonData->{'raioOrigem'};
21      $raioDestino = $jsonData->{'raioDestino'};
22      $raioTrajeto = $jsonData->{'raioTrajeto'};
23      $latAtual = $jsonData->{'latAtual'};
24      $lngAtual = $jsonData->{'lngAtual'};
25
26      echo $localOrigem->cidade;
27
28      $localOrigem = obterLocal($origem);
29      $localDestino = obterLocal($destino);
30
31      $nparams = 0;
32      $sqlCargaWhere = "HERE ";
33      if(!empty($capacidade)){
34          $sqlCargaWhere .= "(peso_volume <= $capacidade) ";
35          $nparams++;
36      }
37      if(!empty($origem)){
38          if($nparams != 0){
39              $sqlCargaWhere .= "AND ";
40          }
41          $origem = strtoupper($localOrigem->cidade);
42          $ufOrigem = strtoupper($localOrigem->uf);
43          if(!empty($raioOrigem)){
44              if(!empty($raioTrajeto)){
45                  $sqlCargaWhere .= "(((UPPER(cidadeOrigem) = '$origem') AND (UPPER(ufOrigem) = '$uforigem')) OR ";
46                  $sqlCargaWhere .= "(lat_lng_distance(latOrigem, longOrigem, $localOrigem->lat, $localOrigem->lng) <= $raioOrigem) OR ";
47                  $sqlCargaWhere .= "(lat_lng_distance(latOrigem, longOrigem, $latAtual, $lngAtual) <= $raioTrajeto)) ";
48              }else{
49                  $sqlCargaWhere .= "(((UPPER(cidadeOrigem) = '$origem') AND (UPPER(ufOrigem) = '$uforigem')) OR ";
50                  $sqlCargaWhere .= "(lat_lng_distance(latOrigem, longOrigem, $localOrigem->lat, $localOrigem->lng) <= $raioOrigem)) ";
51              }
52          }else{
53              if(!empty($raioTrajeto)){
54                  $sqlCargaWhere .= "(((UPPER(cidadeOrigem) = '$origem') AND (UPPER(ufOrigem) = '$uforigem')) OR ";

```

Fonte: Desenvolvido pelo autor.

A imagem seguinte mostra a tela de autenticação de usuários do sistema web do IBFrete.

Figura 22 - Tela de autenticação de usuários do IBFrete



Fonte: Desenvolvido pelo autor.

Nessa tela o usuário irá informar o seu *login* ou usuário, e sua senha de acesso para obter acesso às funções do sistema.

As outras telas importantes do sistema web são a tela de listagem de fretes onde o usuário poderá visualizar todos os fretes cadastrados e também a tela de cadastro e edição de fretes onde o usuário poderá incluir e alterar os dados dos fretes previamente cadastrados.

As imagens a seguir mostram as telas de listagem de fretes e a tela de cadastro de fretes.

Figura 23 - Tela de listagem de Fretes

Origem	Destino	Peso/Quilos	Valor	Ações
Santa Luzia	Zepi Paulo	5	1008	Excluir
Curvelo	Atopu Arinos	4	1504	Excluir
Paranaguá	Itaboraí Paraíba	5	1209	Excluir
Caratinga - MG, Brasil	Santa - SP, Brasil	5	0	Excluir
Caratinga - MG, Brasil	Santa - SP, Brasil	5	0	Excluir
Caratinga - MG, Brasil	Santa - SP, Brasil	4	0	Excluir
Itapecuru - MG, Brasil	Natuv - SP, Brasil	9	0	Excluir
Santa Luzia - MG, Brasil	Itapecuru - MG, Brasil	5	0	Excluir
Santa Luzia - MG, Brasil	Itapecuru - MG, Brasil	5	9	Excluir
Curvelo - MG, Brasil	Divinópolis - MG, Brasil	1	250	Excluir
Caratinga	Itapecuru Paulo	4	2	Excluir
Itapecuru São Neves - MG, Brasil	Itapecuru Paulo	5	0	Excluir
Itapecuru - MG, Brasil	Curvelo - MG, Brasil	13	10	Excluir
Itapecuru - MG, Brasil	Itapecuru São Neves - MG, Brasil	3	0	Excluir
Itapecuru São Neves - MG, Brasil	Itapecuru Paulo	4	4	Excluir
Caratinga - MG, Brasil	Paranaguá - MG, Brasil	6	6	Excluir
Itapecuru - MG, Brasil	Itapecuru Paulo	7	7	Excluir
Santa Luzia - MG, Brasil	Itapecuru - MG, Brasil	1	0	Excluir
Santa Luzia - MG, Brasil	Itapecuru - MG, Brasil	1	0	Excluir
Santa Luzia - MG, Brasil	Santa Luzia - MG, Brasil	1	0	Excluir
Caratinga - MG, Brasil	Pau Paulo - SP, Brasil	5	0	Excluir
Paranaguá - MG, Brasil	Santa Luzia - Brasil	113	12	Excluir
Curvelo - PI, Brasil	Santa Luzia - Brasil	1100	1	Excluir
Curvelo - PI, Brasil	Santa Luzia - Brasil	1100	1	Excluir
Itapecuru - MG, Brasil	Santa Luzia - MG, Brasil	1	1	Excluir
Itapecuru - MG, Brasil	Paranaguá - MG, Brasil	2500	1000	Excluir
Santa Luzia - SP, Brasil	Itapecuru - RJ, Brasil	25	0	Excluir
Caratinga - SP, Brasil	PI, Brasil	25	0	Excluir
Santa Luzia	Itapecuru	3000	0	Excluir
Santa Luzia	Itapecuru	3000	0	Excluir
Itapecuru - SP, Brasil	Caratinga - MG, Brasil	32	0	Excluir
Itapecuru - SP, Brasil	Paranaguá - MG, Brasil	32	0	Excluir
Itapecuru - RJ, Brasil	Itapecuru	24	0	Excluir
Itapecuru - RJ, Brasil	Itapecuru	25	0	Excluir

Fonte: Desenvolvido pelo autor.

Figura 24 - Tela de Cadastro/Edição de Fretes

**Cadastro de Frete**

Destino:

Origem:

Destino:

Peso/Quilos:

Tipo CATEGORIA:

Data Hora Max:

Fonte: Desenvolvido pelo autor.

## 5 CONCLUSÃO

O desenvolvimento do sistema proposto na pesquisa mostrou que existem ferramentas e técnicas capazes de suportar o desenvolvimento de sistemas que são compostos por softwares desenvolvidos com diferentes tecnologias.

As ferramentas disponíveis para o desenvolvimento de aplicativos para o sistema operacional *Android* se mostram maduras e de fácil utilização. Com isso, as organizações têm possibilidade de desenvolver aplicativos de diversos níveis de complexidade com um custo baixo ou zero. Desenvolver com a linguagem java também torna atraente a plataforma *Android*, pois trata-se de uma linguagem de programação altamente difundida e com um vasto conteúdo disponível para pesquisas.

Além disso, a plataforma *Android* se mostrou madura para as funções de integração de dados com *WebServices* utilizando especificações atuais como o *JSON*.

Os *WebServices* também proporcionaram uma interface de comunicação entre o aplicativo *Android* e o sistema web que supriu as necessidades do domínio do problema proposto na pesquisa.

Como os *WebServices* e o sistema web foram desenvolvidos em *PHP* foi possível angariar um grande número de material de pesquisa e auxílio para o desenvolvimento. É importante notar também que o *PHP* proporcionou um custo zero no desenvolvimento e também tem um custo baixo quando colocado em produção.

Quanto aos objetivos do sistema, foi possível desenvolver as funcionalidades de forma que o aplicativo tenha acesso aos dados de fretes cadastrados no site e também que o aplicativo fornecesse aos usuários um cadastro de suas informações. O sistema web por sua vez permite que os donos de fretes possam cadastrar suas cargas para que sejam visualizadas no aplicativo.

O sistema desenvolvido é bastante interessante para os dias de hoje. E também as ferramentas disponíveis para o desenvolvimento dão o suporte necessário aos requisitos de desenvolvimento de um sistema como o *IBFrete*. Serão pertinentes futuras melhorias no sistema como uma possível função para rastreamento de veículos e frotas com o auxílio do *GPS* dos dispositivos móveis, além de outras futuras funcionalidades a serem desenvolvidas.

## REFERÊNCIAS

BENTO, Evaldo Junior. **Desenvolvimento web com PHP e MySQL**. Brasil: Casa do Código, 2013.

DAMASCENO, Arthur; RIBEIRO, Jane. **Caminhoneiro ajuda a impulsionar a economia**. 2013. Disponível em: <<http://www.fmanha.com.br/economia/caminhoneiro-ajuda-a-impulsionar-economia>>. Acesso em: 18. Ago. 2015.

DEPARTAMENTO NACIONAL DE INFRAESTRUTURA DE TRANSPORTES. **Transporte Rodoviário**. 2014. Disponível em: <<http://transportes.gov.br/transporte-rodoviario-relevancia.html>>. Acesso em: 18. Ago. 2015.

ECMA. **Standard ECMA-404: The Json Data Interchange Format**. 1. Ed. 2013. Disponível em: <<http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>>. Acesso em: 18. Ago. de 2015.

EVANS, Benjamin J; FLANAGAN, David. **Java in a Nutshell. A Desktop quick reference**. 6. Ed. Estados Unidos da América: O'Reilly, 2015.

FIELDING, Roy Thomas. **Architectural Styles and the Design of Network-based Software Architectures**. Doctoral dissertation, University of California, Irvine, 2000. Disponível em: <[https://www.ics.uci.edu/~fielding/pubs/dissertation/rest\\_arch\\_style.htm](https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm)>. Acesso em: 18. Ago. 2015.

GOSLING, James; JOY, Bill; STEELE, Guy; BRACHA, Gilad; BUCKLEY, Alex. **The Java® Language Specification: Java SE 8 Edition**. 2015. Disponível em: <<https://docs.oracle.com/javase/specs/jls/se8/html/index.html>>. Acesso em: 27. Ago. 2015.

GUEDES, Gilleanes T. A. **UML 2: uma abordagem prática**. 2. Ed. São Paulo: Novatec, 2011.

MASCARENHAS, Sidney A. **Metodologia Científica**. 2012. São Paulo: Pearson, 2012.

MONTEIRO, João Bosco. **Google Android: crie aplicações para celulares e tablets**. 1. Ed. Brasil: Casa do Código, 2012.

PORTAL DA EDUCAÇÃO. **Plataforma Android – Arquitetura da plataforma**. 2015. Disponível em: <<http://www.portaleducacao.com.br/informatica/artigos/63391/plataforma-android-arquitetura-da-plataforma>>. Acesso em: 29. Set. 2015.

POTENCIER, Fabien. **PHP é muito melhor do que você pensa**. 2012. Disponível em: <<http://imasters.com.br/linguagens/php/php-e-muito-melhor-do-que-voce-pensa/?trace=1519021197&source=single>>. Acesso em: 28. Set. 2015.

PRESSMAN, Roger S. **Engenharia de software: Uma abordagem profissional**. 7. Ed. Porto Alegre: AMGH, 2011.

RECKZIEGEL, Mauricio. **Entendendo os WebServices**. 2006. Disponível em: <<http://imasters.com.br/artigo/4245/web-services/entendendo-os-webservices>>. Acesso em: 29. Set. 2015.

SOMMERVILLE, Ian. **Entenharia de Software**. 9. Ed. São Paulo: Pearson, 2011.

TATROE, Kevin; MACINTYRE, Peter; LERDORF, Rasmus. **Programming PHP**. 3. Ed. United States of America: O'Reilly, 2013.

W3C, World Wide Web Consortium; MIT, Massachusetts Institute of Technology. **Hypertext Transfer Protocol -- HTTP/1.1**. 1999. Disponível em: <<http://www.w3.org/Protocols/rfc2616/rfc2616.html>>. Acesso em: 18. Ago. 2015.

W3C, World Wide Web Consortium. **Web Services Architecture**. 2004. Disponível em: <<http://www.w3.org/TR/ws-arch/#whatis>>. Acesso em: 18. Ago. 2015.

W3C, World Wide Web Consortium. **SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)**. 2007. Disponível em: <<http://www.w3.org/TR/soap12/#intro>>. Acesso em: 18. Ago. 2015.

W3C, World Wide Web Consortium. **Extensible Markup Language (XML) 1.0**. 5. Ed. 2008. Disponível em: <<http://www.w3.org/TR/2008/REC-xml-20081126/#sec-intro>>. Acesso em: 18. Ago. 2015.

## APÊNDICE A – ESPECIFICAÇÃO DE REQUISITOS DO SISTEMA IBFRETE.

Este documento apresenta a especificação de requisitos para o sistema de rastreamento de cargas e fretes IBFrete. Que terá como principal facilitar a pesquisa de fretes pelos caminhoneiros e empresas de transporte de cargas.

### 1.1 REQUISITOS FUNCIONAIS

RF01. O sistema deverá ser composto por um aplicativo para dispositivos móveis e um sistema web, e eles irão se integrar por meio de Webservice.

RF02. O aplicativo móvel deverá permitir o usuário se identificar por meio da placa do seu veículo.

RF03. O aplicativo móvel deverá permitir o usuário cadastrar seus dados de identificação caso ainda não tenha cadastrado.

RF04. O aplicativo móvel deverá possuir uma tela para consulta de fretes onde o usuário especifique um local de origem (Obrigatório), um local de destino (Obrigatório), um raio de procura de carga de acordo com o local de origem, um raio de procura de carga de acordo com a posição atual na rota, um raio de procura de carga com um local de destino, o tipo de carroceria do veículo e a capacidade disponível.

RF05. Na tela para consulta de fretes o local de origem do usuário deve ser preenchido automaticamente de acordo com as coordenadas do GPS do dispositivo caso haja conexão de dados disponível.

RF06. Na tela para consulta de fretes os campos local de origem e destino devem ser do tipo texto para livre digitação com recurso de auto completar o texto caso haja conexão de dados disponível.

RF07. O aplicativo móvel deverá exibir uma listagem com o resultado da pesquisa efetuada na tela citada na [RF04].

RF08. Caso não haja conexão com internet disponível em qualquer operação o aplicativo móvel deve exibir uma mensagem amigável ao usuário informando o mesmo.

RF09. Na tela de resultados de pesquisa de carga deverá haver uma opção para iniciar nova pesquisa.

RF10. Na tela de resultados de pesquisa de carga deverá haver opção para o usuário abrir uma tela para alterar os seus dados.

RF11. Caso o usuário encerre o aplicativo com a tela de pesquisa aberta, ao iniciar o aplicativo novamente, após a identificação do usuário, ele deve abrir a listagem com os resultados da última pesquisa efetuada.

RF12. O sistema web deverá ter uma tela para autenticação de usuários.

RF13. O sistema web deverá ter uma tela para cadastro de usuários.

RF14. O sistema web deverá ter uma listagem de fretes cadastrados pelo usuário.

RF15. O sistema web deverá ter opções de filtro de fretes que serão listados.

RF16. O sistema web deverá ter opções de manipulação de registros de fretes cadastrados pelo usuário. (Adicionar, Alterar, Remover, Marcar Frete como Encerrado).

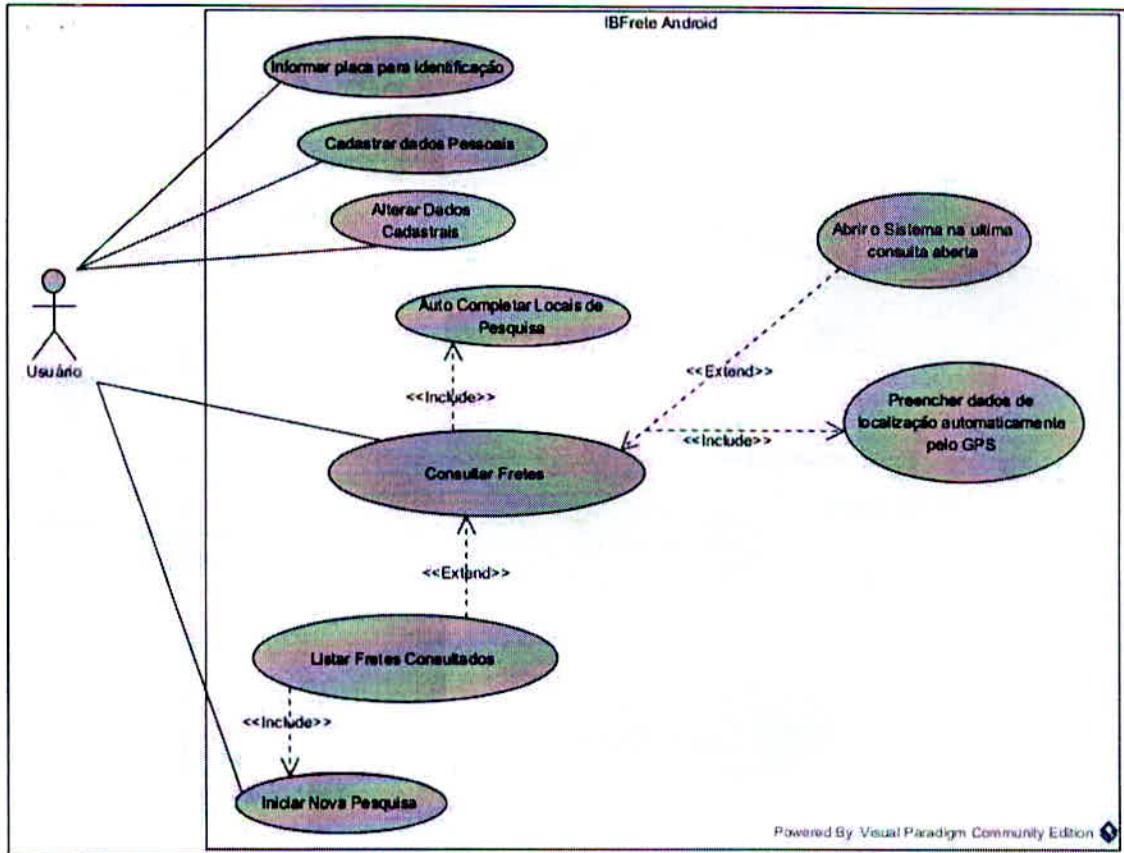
## 1.2 REQUISITOS NÃO FUNCIONAIS

RNF01. O aplicativo deve funcionar em conexão de dados móveis das operadoras de celulares.

RNF02. A consulta de fretes deve retornar os dados em menos de um minuto.

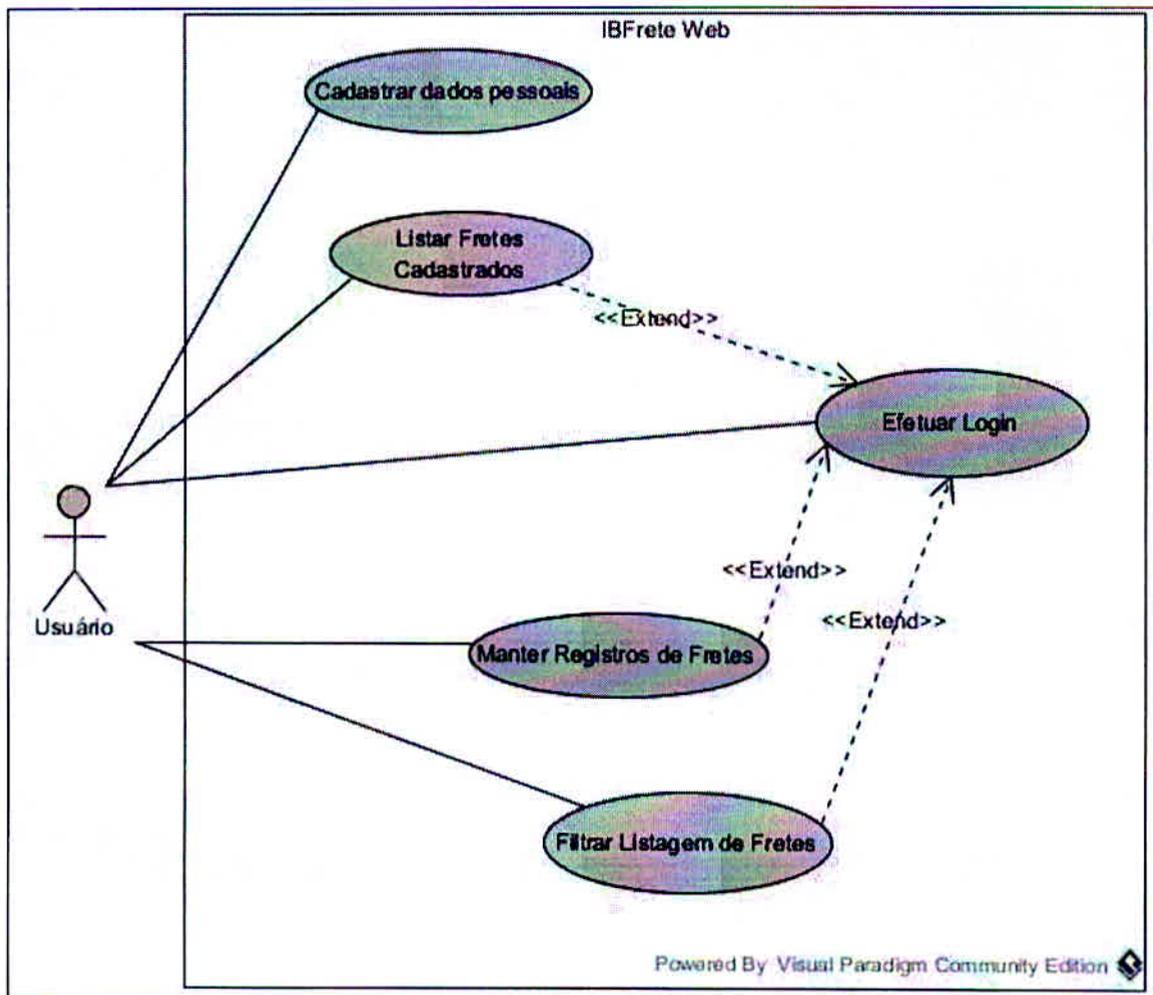
## APÊNDICE B - DIAGRAMAS DE CASO DE USO

Figura 25 - Diagrama de caso de uso IBFrete para *Android*.



Fonte: Desenvolvido pelo autor.

Figura 26 - Diagrama de caso de uso IBFrete Web



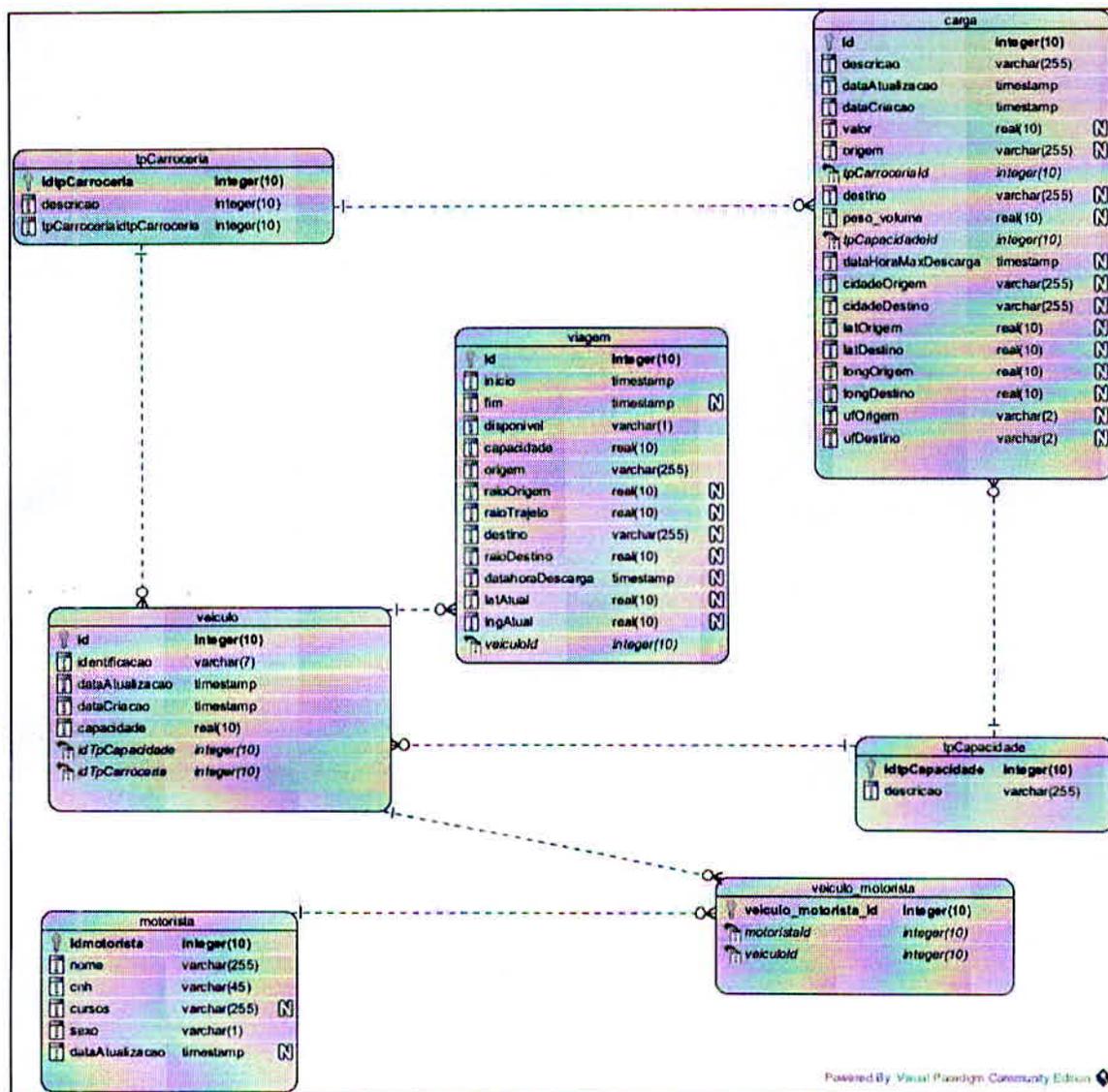
Fonte: Desenvolvido pelo autor.

# APÊNDICE C - DIAGRAMA DE CLASSES



Fonte: Desenvolvido pelo autor.

## APÊNDICE D - DIAGRAMA ENTIDADE RELACIONAMENTO



Fonte: Desenvolvido pelo autor.