

**UNIVERSIDADE FEDERAL DO PIAUÍ – UFPI**  
**CAMPUS SENADOR HELVÍDIO NUNES DE BARROS - CSHMB**  
**Curso Bacharelado em Sistemas de Informação**

Leonardo Neiva Teixeira da Silva Moura

**Protótipo de um Sistema de Agendamento de Controle de Serviços  
de Revisão em Concessionárias de Veículos de Picos-PI**

PICOS-PI,  
2015.

Leonardo Neiva Teixeira da Silva Moura

**Protótipo de um Sistema de Agendamento de Controle de Serviços  
de Revisão em Concessionárias de Veículos de Picos-PI**

Trabalho de conclusão de curso, apresentado ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal do Piauí, como parte dos requisitos para obtenção do título de Bacharel em Sistemas de Informação, sob orientação do Prof. Esp. Francisco das Chagas Imperes Filho.

PICOS-PI,  
2015.

**FICHA CATALOGRÁFICA**  
**Serviço de Processamento Técnico da Universidade Federal do Piauí**  
**Biblioteca José Albano de Macêdo**

**M929p** Moura, Leonardo Neiva Teixeira da Silva.  
Protótipo de um sistema de agendamento de controle de  
serviços de revisão em concessionárias de veículos de Picos-PI  
/ Leonardo Neiva Teixeira da Silva Moura. - 2014.  
CD-ROM : il. ; 4 ¾ pol. (60 p.)

Monografia(Bacharelado em Sistemas de Informação) –  
Universidade Federal do Piauí. Picos-PI, 2014.  
Orientador(A): Prof. Esp. Francisco das Chagas Imperes Filho

1. Sistema Web. 2. Agendamento. 3. Veículos. I. Título.

**CDD 004.21**

Leonardo Neiva Teixeira da Silva Moura

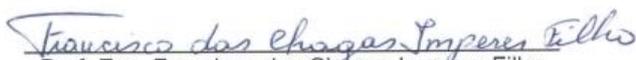
**Protótipo de um Sistema de Agendamento de Controle de Serviços  
de Revisão em Concessionárias de Veículos de Picos-PI**

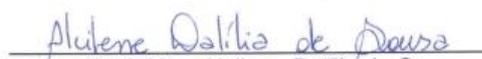
UNIVERSIDADE FEDERAL DO PIAUÍ – UFPI  
Curso de Graduação em Sistemas de Informação

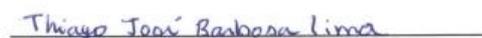
Trabalho apresentado ao Curso de  
Bacharelado em Sistemas de Informação  
da Universidade Federal do Piauí como  
requisito parcial para obtenção do título de  
Bacharel em Sistemas de Informação.

**Data de Aprovação:**

**06/01/2015**

  
Prof. Esp. Francisco das Chagas Imperes Filho  
Universidade Federal do Piauí – UFPI

  
Prof. Msc. Alcilene Dalilia de Sousa  
Universidade Federal do Piauí – UFPI

  
Prof. Thiago José Barbosa Lima  
Universidade Federal do Piauí – UFPI

**Picos – PI  
2014**

## **AGRADECIMENTOS**

A Deus por ter me dado saúde e força para superar as dificuldades.

A esta universidade, seu corpo docente, direção e administração que oportunizaram a janela que hoje vislumbro um horizonte superior, eivado pela acendrada confiança no mérito e ética aqui presentes.

Ao meu orientador Francisco das Chagas Imperes Filho, pelo suporte, orientações e incentivos.

Aos meus pais, pelo amor, incentivo e apoio incondicional.

E a todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

O único lugar aonde o sucesso vem antes  
do trabalho é no dicionário.

Albert Einstein

## RESUMO

Atualmente as Tecnologias da Informação e Comunicação (TIC) tem sido utilizadas para resolver diversos problemas, tanto em empresas de grande porte, quanto em empresas de médio e pequeno porte. Portanto, utilizando Linguagens de Programação (LP) como, por exemplo, HTML, *JavaScript*, PHP, SQL e CSS e de ferramentas que auxiliam no trabalho do programador como, *Sublime Text 2*, *Workbench*, MySQL e *Wampserver*, foi produzido um sistema capaz de atender a demanda de reserva de serviços em uma concessionária de veículos. O sistema pretende suprir a necessidade de uma concessionária na área de revisão de serviços em veículos, que permite melhorar e agilizar a reserva via internet, proporcionando aos seus clientes agendarem uma revisão comparecendo apenas no horário marcado, evitando filas e conseqüentemente demora na hora da revisão do seu veículo. O projeto também tem como meta verificar se o sistema atende as necessidades propostas, por isso foi elaborado um questionário para coleta de informações, e para atingir ao máximo de disponibilidade e usabilidade, foram reduzidos o tamanho e a qualidade das imagens (*background*, imagens de exibição do *slider* na página inicial e logotipos em geral) utilizadas, possibilitando os clientes visualizarem as páginas mais rapidamente e conseqüentemente melhorando a usabilidade do sistema.

**Palavras-chave:** sistema web, agendamento, veículos.

## **ABSTRACT**

Currently the Information and Communication Technologies (ICT) has been used to solve many problems, both in large companies, as for small and medium-sized businesses. Therefore, using programming languages (LP), for example, HTML, JavaScript, PHP, SQL and CSS and tools that help the programmer work as Sublime Text 2, Workbench, MySQL and Wampserver, was produced a system capable of meet the reserve demand for services in a car dealership. The system aims to meet the need for a dealership in the review of the service area in vehicles, which helps improve and streamline the booking via internet, providing its customers to schedule a review appearing only at the scheduled time, avoiding queues and consequently delay the review of the hour of your vehicle. The project also aims to verify that the system meets the proposed needs, so we developed a questionnaire to gather information, and to achieve the maximum availability and usability, were reduced the size and quality of the images (background, image slider display on the homepage and logos in general) used, enabling customers load pages faster and consequently improving the system's usability.

**Keywords:** web system, scheduling, vehicles.

## LISTA DE ILUSTRAÇÕES

FIGURA 1 - Diagramas desenvolvidos no projeto de banco de dados.....	19
FIGURA 2 - Representação do caso de uso Cadastrar Veículos.....	20
FIGURA 3 - Representação do ator Gerente.....	21
FIGURA 4 - Relacionamentos entre o ator Gerente e seus casos de uso.....	21
FIGURA 5 - Caso de uso do sistema.....	22
FIGURA 6 –Representação da entidade serviços.....	22
FIGURA 7 - Representação do relacionamento e dependência entre entidades.....	23
FIGURA 8 - Representação do Diagrama de Entidade Relacionamento.....	24
FIGURA 9 - Função que identifica o uso de SQL Injecton no login.....	25
FIGURA 10 - Método de utilização da função antiInject.....	25
FIGURA 11 - <i>NOT NULL</i> obriga a inserção de dados em campo do banco.....	27
FIGURA 12 - Requisitos funcionais referente ao Cliente.....	32
FIGURA 13 - Requisitos Funcionais referente ao funcionário (Gerente).....	34
FIGURA 14 - Requisitos Funcionais referente ao funcionário (Mecânico).....	36
FIGURA 15 - Requisitos Funcionais referente ao funcionário (Atendente).....	37
FIGURA 16 - Requisitos não Funcionais.....	38
FIGURA 17 - Página de cadastro de veículos.....	40
FIGURA 18 - Página de agendamento de veículos.....	41
FIGURA 19 - Página de promoções divididos por categoria.....	42
FIGURA 20 - Página de promoção geral.....	42
FIGURA 21 - Página de relatórios divididos por categoria.....	43
FIGURA 22 - Relatório do cliente.....	43
FIGURA 23 - Cadastro, Edição e Exclusão de Serviços.....	44
FIGURA 24 - Página de acompanhamento de reserva.....	45
FIGURA 25 - Código de cadastramento de veículo.....	46
FIGURA 26 - Código de alteração de veículo.....	47
FIGURA 27 - Código para desativar um veículo.....	47
FIGURA 28 - Código de busca de reservas.....	48
GRÁFICO 1 - Percentual de usuários que utilizaram o sistema na fase de testes....	49
GRÁFICO 2 - Média da velocidade de conexão utilizada pelos usuários.....	50
GRÁFICO3 - Nível de satisfação dos usuários na fase de testes de funcionalidade.	51

## **LISTA DE ABREVIATURAS E SIGLAS**

CSS - Cascading Style Sheets

HTML – Hyper Text Markup Language

IHC – Interação Humano-Computador

LP – Linguagem de Programação

MySql - My Structured Query Language

PHP - Personal Home Page

SQL - Structured Query Language

SGBD – Sistema de Gerenciamento de Banco de Dados

TICs - Tecnologias da Informação e Comunicação

UML - Unified Modeling Language

## SUMÁRIO

<b>1.</b>	<b>INTRODUÇÃO</b>	<b>12</b>
1.1	Objetivos	12
1.1.1	Objetivo Geral	13
1.1.2	Objetivos Específicos	13
1.2	Organização do Documento	13
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>14</b>
2.1	Linguagens de Programação Utilizadas	14
2.2	Softwares Utilizados	17
<b>3</b>	<b>PROTÓTIPO DO SISTEMA</b>	<b>19</b>
3.1	Projeto de Banco de Dados	19
3.1.1	Diagrama de Caso de Uso	20
3.1.2	Diagrama Entidade Relacionamento	22
3.2	Módulos do Sistema de Agendamento	24
3.2.1	Módulo I – Segurança	24
3.2.2	Módulo II - Gestão de Usuários	27
3.2.3	Módulo III - Reservas	29
3.2.4	Módulo IV - Dados	30
3.3	Requisitos do Sistema	32
3.3.1	Requisitos Funcionais	32
3.3.1.1	Cliente	32
3.3.1.2	Gerente	34
3.3.1.3	Mecânico	36
3.3.1.4	Atendente	37
3.3.2	Requisitos não funcionais	38
3.4	Interface dos Requisitos Funcionais (Tópico 3.3.1)	40
3.4.1	Cliente	40
3.4.2	Gerente	41
3.4.3	Mecânico	44
3.4.4	Atendente	44
3.5	Funções	45
<b>4</b>	<b>RESULTADOS</b>	<b>49</b>
4.1	Quantidade de Usuários (Usabilidade e Interface)	49
4.2	Velocidade de Conexão (Disponibilidade)	50
4.3	Funcionalidades (Funcionalidade)	50
<b>5.</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>52</b>
	<b>REFERÊNCIAS</b>	<b>54</b>
	<b>APÊNDICE</b>	<b>55</b>
	<b>APÊNDICE A - Questionário utilizados nos testes</b>	<b>56</b>
	<b>APÊNDICE B - Funções do sistema</b>	<b>57</b>

## 1. INTRODUÇÃO

Com o crescimento da economia na cidade de Picos-PI, as empresas estão encontrando dificuldades em gerir seu próprio negócio, com isto, estão sendo obrigadas a estabelecer sistemas internos de gestão, possibilitando o desenvolvimento de sistemas específicos para cada empresa, que buscam melhorar a eficácia na gestão empresarial.

Todo sistema é desenvolvido com a finalidade de solucionar problemas do mundo real. Neste contexto, o projeto apresenta uma opção viável de gestão de clientes na reserva de veículos de uma concessionária na cidade de Picos-PI.

Com o grande avanço das TICs (Tecnologias da Informação e Comunicação), principalmente na área da programação, hoje existem LPs (Linguagens de Programação) capazes de prover mecanismos para criação de sistemas inteligentes e organizados. Portanto, sistemas com essas características são essenciais para uma instituição, independentemente do porte e de sua área de atuação.

Com o propósito de resolver a problemática citada, foi realizado um levantamento de requisitos para obter requisitos de usuário que o sistema deve gerenciar, possibilitando o desenvolvimento de um projeto de banco de dados consistente.

Como resultado a aplicação contempla soluções como: gestão de *login* (Clientes e Funcionários); uma área de agendamento onde o cliente possa realizar o agendamento de serviços de seu veículo, permitindo escolher o dia e o horário da reserva; e, por fim, uma área administrativa para cada nível de funcionário (Gerente, Mecânico e Atendente), que possibilita a utilização de funcionalidades específicas para cada tipo de usuário.

### 1.1 Objetivos

Para que o sistema fosse desenvolvido, foi necessário realizar um estudo de caso sobre as necessidades dos usuários. Para tanto, foram realizadas entrevistas com funcionários (Gerente, Mecânico e Atendente) para coletar informações, gerando objetivos que subsidiaram o desenvolvimento do sistema.

### 1.1.1 Objetivo Geral

Implementar um sistema capaz de gerenciar o agendamento de reserva de serviços para uma concessionária de veículos na cidade de Picos-PI, podendo o cliente agendar a reserva de seu veículo na data e hora desejada, indo a concessionária somente no dia e hora marcado para realização do(s) serviço(s) solicitado(s).

### 1.1.2 Objetivos Específicos

- Desenvolver um sistema capaz de gerenciar as reservas de revisão de veículos.
- Implementar uma área onde o mecânico possa gerenciar os serviços prestados.
- Prover um sistema com interface amigável utilizando técnicas de Interface Humano-Computador(IHC).

## 1.2 Organização do Documento

Após a introdução serão apresentados os próximos capítulos, que estão organizados da seguinte forma:

- Capítulo 2 – Referencial Teórico: Fornece o embasamento teórico para o trabalho. São demonstrados conceitos relacionados as LPs (linguagens de programação) e as ferramentas de trabalho utilizadas.
- Capítulo 3 – Protótipo do Sistema: Será mostrada as etapas do desenvolvimento do sistema como: projeto de banco de dados, requisitos funcionais e não funcionais, interfaces dos requisitos funcionais e funções.
- Capítulo 4 – Resultados: São mostrados os resultados dos testes realizados no sistema, que são: teste de quantidade de usuários, teste de velocidade média de conexão e teste de nível de satisfação dos usuários.
- Capítulo 5 – Considerações Finais: Apresenta-se a conclusão do protótipo e indicações de trabalhos futuros.

## 2. REFERENCIAL TEÓRICO

O sistema foi implementado utilizando tanto materiais físicos: (computador), quanto softwares: (internet, servidor local, gerenciador de banco de dados, modelagem de banco de dados e editor de códigos), e linguagens de programação.

### 2.1 Linguagens de Programação Utilizadas

Para a produção do sistema baseado na *web* de controle de agendamento de serviços foram utilizadas diversas LPs tais como: HTML<sup>1</sup>, CSS<sup>2</sup>, PHP<sup>3</sup>, *Java Script*<sup>4</sup> e SQL<sup>5</sup>. Lançando mão dessas linguagens foi desenvolvido um sistema focando principalmente na área de agendamento de reserva de veículos. Abaixo uma breve descrição acerca das tecnologias mencionadas:

- HTML (*HyperText Markup Language* - Linguagem de Marcação de Hipertexto)

O HTML é uma linguagem base para publicação de sistemas *web*, ou seja, é essencial para que o sistema *web* funcione, dividindo a página *web* em etapas melhorando a organização e exibição do conteúdo. Com o passar dos anos a LP HTML veio evoluindo. Hoje a versão disponível é 5, que tem como objetivo mudar a forma de codificar e organizamos páginas *web*.

“O HTML modifica a forma de como escrevemos o código e organizamos as informações na página. Seria mais semântica com menos código. Teria mais interatividade sem a necessidade de instalação de plug-ins e perda de performance. É a criação de código interoperável, pronto para futuros dispositivos que facilita a reutilização da informação de diversas formas.”

FERREIRA, EIS (2010)

Visando a compatibilidade com outras linguagens, o HTML possibilita utilizar mais três linguagens: PHP, *Java Script* e CSS. O PHP com a função de gerenciar as

<sup>1</sup> - Disponível em: <http://www.w3schools.com/html>

<sup>2</sup> - Disponível em: <http://www.w3schools.com/css>

<sup>3</sup> - Disponível em: <http://php.net>

<sup>4</sup> - Disponível em: <http://www.w3schools.com/js>

<sup>5</sup> - Disponível em: <http://www.w3schools.com/sql>

conexões com o Sistema Gerenciador de Banco de Dados (SGBD) MySQL, o CSS em conjunto com a LP *Java Script*, todas incorporadas ao HTML, possuem a função de melhorar a interface e usabilidade do sistema.

- *CSS (Cascading Style Sheets – Folhas de Estilos em Cascata)*

Com as vantagens que as CSS proporcionam, foi desenvolvido um software de fácil usabilidade utilizando técnicas de IHC para que os usuários não tenha nenhuma dificuldade na hora do seu cadastro ou agendar serviços para seu veículo.

As CSS estilizam qualquer tipo de informação contida no HTML, que pode ser: imagem, texto, vídeo, áudio ou qualquer outro elemento. Geralmente a formatação é visual. (W3C, 2014).

Além dessas características as CSS podem formatar cores, background, fontes, margens e outras propriedades da estrutura da página web. Para que um sistema tenha uma interface elegante e moderna as folhas de estilos são suficientes para auxiliar a HTML na criação e apresentação de páginas web.

- *Java Script*

*Java Script* é uma LP baseada em scripts, onde estes são integrados a página web devido sua facilidade de interação. Esta linguagem possui scripts que permitem incrementar a apresentação da página tornando-a dinâmica e interativa. Utilizando *JavaScript* foram codificados alguns scripts que incorporados à página inicial do sistema proporcionaram efeitos diversos como, por exemplo, máscaras usadas nos campos (*Fields*) para inserção de dados.

- *PHP (Personal Home Page)*

Mesmo um sistema elegante e moderno precisa de uma linguagem que trate a comunicação com o banco de dados, manuseio dos dados na página web e da segurança, para isso foi utilizada a LP PHP.

Segundo Hiederauder (2011), “PHP é uma das linguagens mais utilizadas da

*web*. Milhões de sites no mundo inteiro utilizam PHP. A principal diferença em relação às outras linguagens, é a capacidade que a PHP tem de interagir com o mundo *web*, transformando totalmente os *websites* que possuem páginas estáticas.”

A PHP é responsável pela inserção (envio de informações ao banco de dados), edição (envio de informações ao banco de dados reescrevendo os dados armazenados) e exclusão das informações no banco de dados.

Além de ser responsável pela inserção, edição e exclusão das informações no banco de dados, a PHP auxilia o SGBD MySQL no armazenamento e manipulação das informações utilizando a linguagem SQL.

Uma das vantagens de se utilizar PHP é por causa da sua gratuidade, possui código-fonte aberto (*open source*) e é orientado a objetos, possibilitando a criação de páginas dinâmicas com maior qualidade e segurança para o sistema *web*.

- SQL (*Structured Query Language – Linguagem de Consulta Estruturada*)

SQL é uma linguagem de programação que possibilita a comunicação com o banco de dados. Através de comandos em SQL é possível estabelecer a comunicação com o banco de dados.

“Para o gerenciamento de informações são baseados em registros (ou *Record*, em inglês). Um registro é um conjunto simples de dados com o qual se identifica determinado elemento (uma pessoa, um objeto, um evento etc.), associando a ele uma série de atributos que o caracterizam de maneira unívoca, por exemplo: nome, sobrenome e endereço, e número de telefone, no caso de uma pessoa; local, data e hora em que certo evento será realizado; código, peso, dimensões e cor, caso se trate de um objeto ou produto. Cada informação é inserida separadamente dentro de um espaço a ela reservada, chamado de campo.” FABRÍCIO AUGUSTOS (2007).

No sistema forma utilizados comandos em SQL para inserir, alterar e buscar informações no banco de dados, auxiliando a PHP no controle de informações do banco acessando informações sempre que necessário.

## 2.2 Softwares Utilizados

- MySQL (*My Structured Query Language*)

O MySQL<sup>6</sup> é um SGBD muito utilizado devido a sua praticidade e segurança com as informações. Devido a essas propriedades, é altamente recomendado para aplicações *web*, sendo considerado um dos 5 gerenciadores de bancos de dados<sup>7</sup> mais usados no mundo.

Devido a esses atributos o MySQL foi escolhido como SGBD como repositório de dados para o sistema de controle de agendamento de serviços. Todo sistema que tenha que armazenar dados será obrigado a utilizar um banco de dados. No sistema de controle de agendamento de serviços de revisão será necessário um banco de dados para armazenar informações como: dados sobre os clientes, funcionários, veículos e reservas de veículos efetuadas.

“Com o tempo, foram sendo identificadas funcionalidades comuns a muitos programas. Por exemplo, hoje, a grande maioria dos programas comunica-se com usuários através de interfaces gráficas de janelas. Entretanto, normalmente, os programas não contêm todo o código referente à exibição dos dados na interface, mas utilizam gerenciadores de interface de usuário, conjuntos de rotinas que incluem as funcionalidades que um programador vai necessitar frequentemente ao construir uma interface de usuário. Da mesma forma, para comunicar-se com processos remotos, os programas usam gerenciadores de comunicação. Para manter grandes repositórios compartilhados de dados, ou seja, para manter banco de dados, são usados sistemas de gerência de banco de dados.” HEUSER (2009)

- *Wampserver*

*Wampserver*<sup>8</sup> é uma ferramenta utilizada como servidor de aplicações *web*, possibilitando ao programador testar seu sistema sem ter que disponibilizá-lo na *web*. Ela é compatível com PHP, HTML e *Java Script*, que são linguagens utilizadas

---

<sup>6</sup> – Disponível em: <http://www.mysql.com/>    <sup>7</sup> – Disponível em: <http://db-engines.com/en/ranking>

<sup>8</sup> – Disponível em: <http://www.wampserver.com/>

no sistema de agendamento de controle de revisão, possuindo também integração com SGBD MySQL.

- *Sublime Text 2*

*Sublime Text 2*<sup>9</sup>, é um editor de códigos que possui suporte para as mais variadas linguagens existentes, devido a sua gratuidade e design elegante ganhou uma grande quantidade de adeptos por todo o mundo.

O *Sublime Text 2* foi escolhido pela compatibilidade com as linguagens utilizadas no sistema *web*, possibilitando maior organização na criação de códigos, permitindo gerenciar todas as linguagens no mesmo editor.

- *Workbench*

O *Workbench*<sup>10</sup> é a ferramenta oficial do MySQL. Sendo ambiente completo, permite realizar consultas, criar diagramas e gerenciar um banco de dados por completo.

O *Workbench* foi utilizado devido a possibilidade de conexão com o servidor local, possibilitando em poucos cliques importar o banco modelado para o bando no servidor.

---

<sup>9</sup> – Disponível em: <http://www.sublimetext.com/>

<sup>10</sup> – Disponível em: <http://www.mysql.com/products/workbench/>

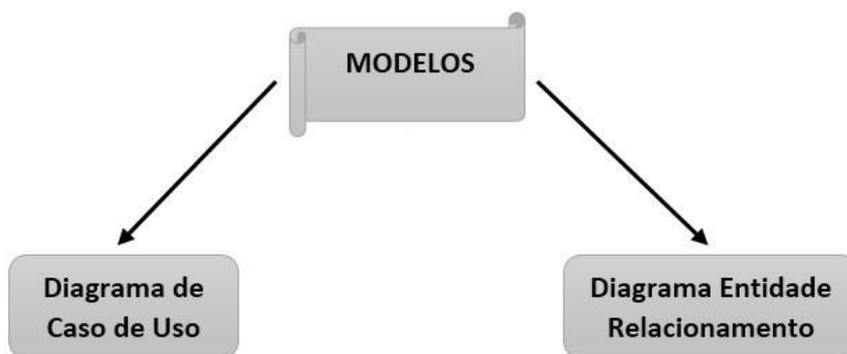
### 3. PROTÓTIPO DO SISTEMA

Neste capítulo será demonstrado o processo utilizado para desenvolver o sistema de agendamento de serviços de revisão, sendo que a exposição obedecerá a seguinte ordem: Projeto de Banco de Dados, Módulos, Requisitos, Interface e Funções.

#### 3.1 Projeto de Banco de Dados

Um projeto de banco de dados pode ser desenvolvido utilizando-se de técnicas UML (*Unified Modeling Language* - Linguagem de Modelagem Unificada) e seus diagramas. UML é um modelo que representa graficamente os elementos envolvidos em um sistema, ou como cita Heuser (1998), “a primeira etapa do projeto de um banco de dados é a construção de um modelo conceitual, a chamada modelagem conceitual”. Para o desenvolvimento do banco de dados, foram criados 2 diagramas UML: diagrama entidade relacionamento e diagrama de caso de uso, demonstrados na Figura 1.

Figura 1: Diagramas desenvolvidos no projeto de banco de dados.



Fonte: Autoria própria.

Abaixo segue uma descrição sobre os diagramas de UML utilizados na produção do software.

##### 3.1.1 Diagrama de Caso de Uso

O diagrama de caso de uso possibilita uma visão do que será desenvolvido, melhora a compreensão, visualizando o que o sistema deve realizar. O diagrama de caso de uso também facilita a comunicação com o cliente, mostrando detalhadamente como o software funciona através de elementos como: cenário, caso de uso, atores (clientes e funcionários) e relacionamentos entre as várias entidades que compõem o sistema.

- Cenário

O cenário é a sequência de eventos que acontece quando o usuário interage com o sistema, possibilitando ter uma visão ampla de tudo que o sistema é capaz de realizar, gerando os requisitos funcionais necessários para o projeto de banco de dados.

- Caso de Uso

Caso de uso é uma tarefa ou funcionalidade realizada pelo ator, como mostrado na Figura 2, o caso de uso realiza o cadastro de veículos, ou seja, um caso de uso representa qualquer funcionalidade (requisito funcional) que o sistema venha a desempenhar.

Figura 2: Representação do caso de uso Cadastrar Veículos.

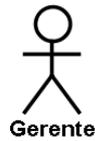


Fonte: A autoria própria.

- Atores

O ator é representado por um usuário do sistema como mostra a Figura 3, ou seja, não representam a pessoa física, mas sua regra, que é quando uma pessoa interage com o sistema de diferentes maneiras.

Figura 3: Representação do ator Gerente.

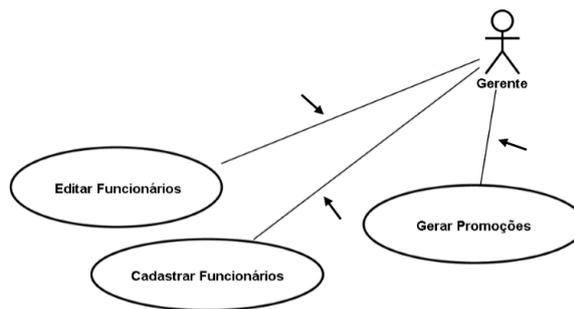


Fonte: Autoria própria.

- Relacionamentos

Os relacionamentos são ligações que unem o ator, aos seus casos de uso, formando um grupo de ações que um ator pode realizar no sistema que é demonstrado na Figura 4. Quando relacionados com outros atores, formam o diagrama de caso de uso demonstrado na Figura 5.

Figura 4: Relacionamentos entre o ator Gerente e seus casos de uso.

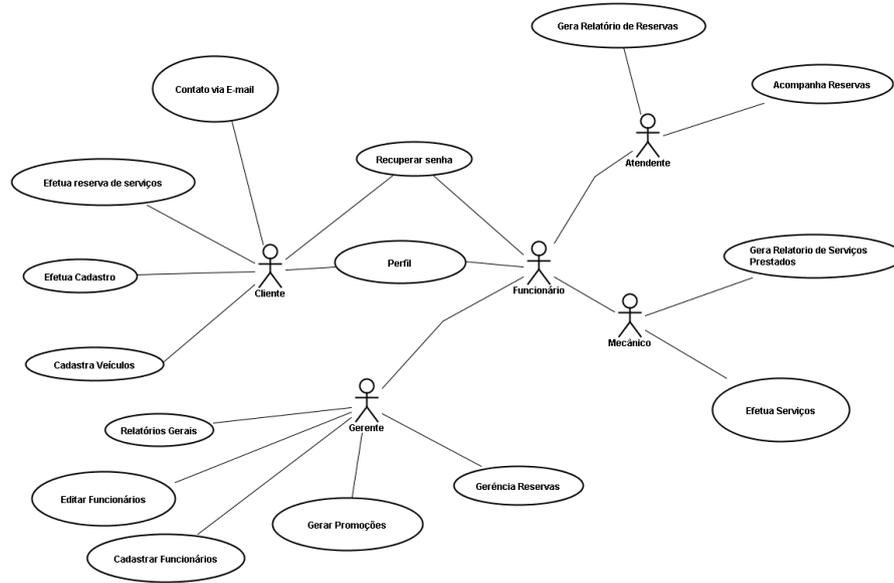


Fonte: Autoria própria.

- Caso de Uso do Sistema

O diagrama de caso de uso descreve as funcionalidades propostas, sendo utilizado para o levantamento de requisitos do sistema. Utilizando atores e relacionamentos, o diagrama de caso de uso descreve de forma geral as ações que os clientes e funcionários realizam no sistema.

Figura 5: Caso de uso do sistema.



Fonte: Autoria própria.

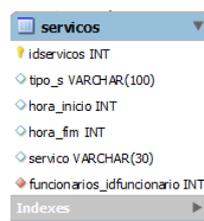
### 3.1.2 Diagrama Entidade Relacionamento

O diagrama entidade relacionamento tem como objetivo representar os modelos de dados de um sistema na forma mais próxima do mundo real. É utilizado para representar o modelo conceitual do negócio e adota o conceito de entidades e relacionamentos como exibido na Figura 9.

- Entidade

A entidade representa um grupo de objetos, que são representados por atributos (nome, senha, cpf) e seus métodos (ações e comportamentos), cada atributo é representado pelo seu tipo (*Int*, *Varchar*, *Float*, *Date*), demonstrada na Figura 6.

Figura 6: Representação da entidade serviços.

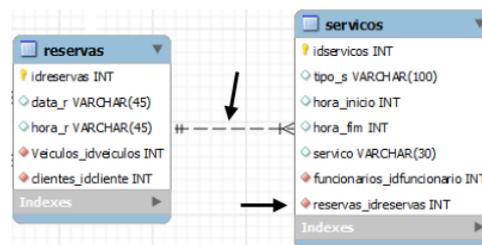


Fonte: Autoria própria.

- Relacionamentos

Uma entidade pode se relacionar com outras entidades utilizando métodos de relacionamentos (associação e dependências). Associação é quando uma classe é interligada a outra, desempenhando um relacionamento entre as mesmas. Dependência é quando uma classe possui uma chave estrangeira pertencente a outra classe, à qual possui um relacionamento, exemplo demonstrado na Figura 7.

Figura 7: Representação do relacionamento e dependência entre entidades.

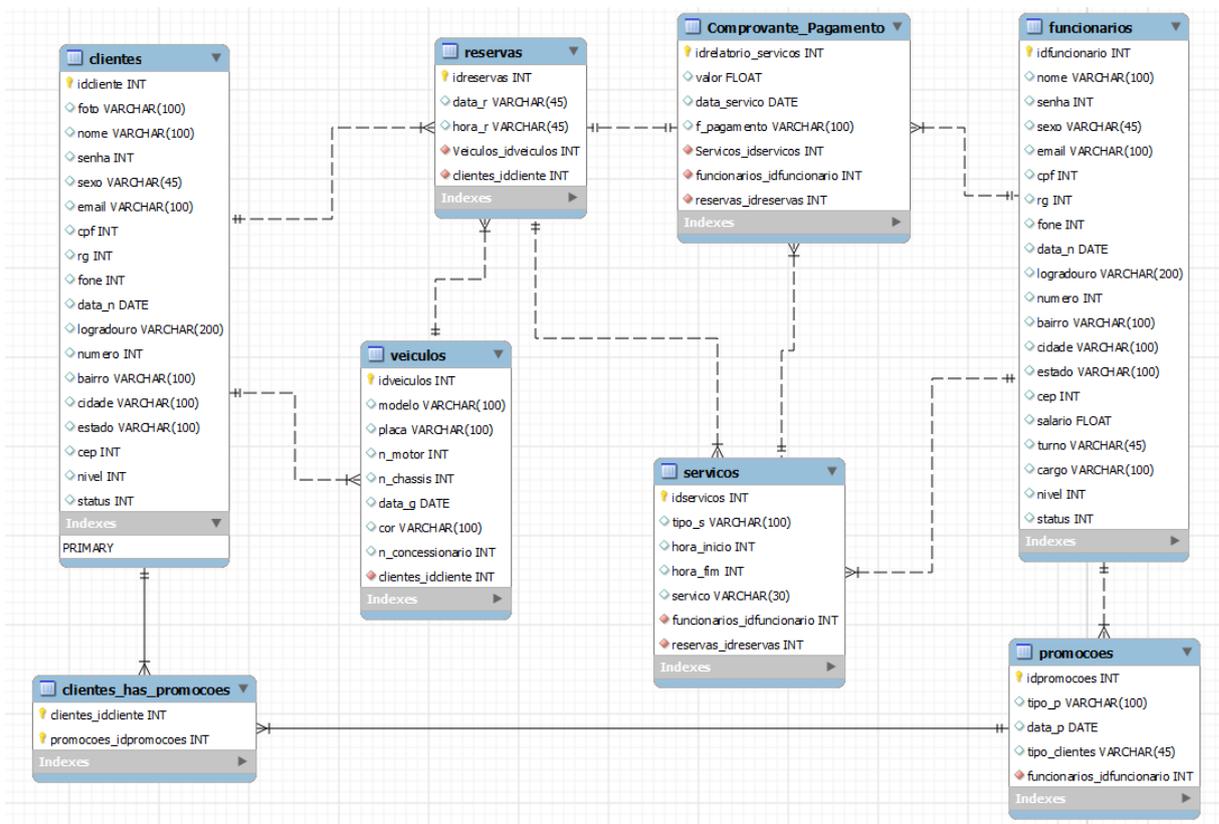


Fonte: Autoria própria.

- Diagrama Entidade Relacionamento do Sistema

O modelo entidade relacionamento é uma forma de descrever e definir um processo de negócio. Que é modelado com componentes (*entidades*) que são ligadas umas às outras por relacionamentos. O diagrama entidade relacionamento do sistema é demonstrado abaixo na Figura 8.

Figura 8: Representação do Diagrama de Classe do Sistema.



Fonte: Autoria própria.

## 3.2 Módulos do sistema de agendamento

### 3.2.1 Módulo I – Segurança

- Login

Um sistema de gestão necessita de diversos módulos, um deles é o de controle de usuários (clientes e funcionários: Gerente, Recepcionista e Mecânico). Este módulo, denominado área de *login*, tem como função gerenciar os diversos tipos de usuários. O *login* possui mecanismos para evitar uso de SQL *injecton* e outros meios utilizados para burlar o sistema, ou seja, ferramentas que se passam por um usuário existente do banco de dados, obtendo permissão e privilégios não autorizados.

O módulo de *login* verifica as informações disponibilizadas pelo usuário. Em seguida, averigua se é um cliente e só depois se é um funcionário devido a



somente as páginas respectivas aos clientes, evitando que um usuário cliente tenha acesso às páginas de administração que são restritas a usuários do tipo funcionário (gerente). Para que o nível de segurança seja ativado, o administrador do sistema, no ato do cadastro dos usuários (funcionário: atendente, mecânico e gerente, e clientes em geral) será atribuído um nível de acesso, permitindo que sempre que um usuário “logar”, será verificado o nível de acesso pelo sistema restringindo as páginas que o usuário pode acessar.

Cada página no sistema possui um código em PHP que faz parte do módulo de segurança, tendo como objetivo verificar se a sessão do usuário ativo pertence a algum nível (Níveis: 1,2,21,22) de usuários permitidos. Sempre que uma página é solicitada será verificado se existe uma sessão ativa, caso exista será verificado o nível e qual classe o usuário pertence. Caso o usuário possua nível para acessar a página solicitada, esta será carregada normalmente, caso contrário será redirecionado para a página respectiva ao seu nível previamente cadastrado. Por fim, caso o usuário não exista o sistema automaticamente redirecionará para a página inicial do sistema para que o *login* seja efetuado.

- Funções

Por motivo de segurança, o sistema armazena suas respectivas funções em uma página denominada *funcoes.php*. Essa página não é acessível a nenhum usuário, somente por solicitação via código. O usuário apenas recebe o resultado da função existente na página *funcoes.php*. Por exemplo, a página de cadastro de clientes solicita o cadastro, neste momento é enviada uma solicitação para cadastrar as informações desejadas. Feito a solicitação, a função cadastrar usuários se encarrega de inserir o usuário desejado. Esse tipo de procedimento é importante devido a usuários mal intencionados tentarem explorar falhas no sistema para realizar invasões e roubo de informações importantes para a empresa.

- HTML 5 e *Java Script*

Dentre as linguagens de programação utilizadas no sistema, a HTML 5 e *Java Script* foram adotadas com o objetivo de padronizar a inserção de dados em

campos do tipo texto, como por exemplo o campo CEP. Mesmo que os usuários tenham CEPs diferentes, todos possuem a mesma quantidade de dígitos, sendo assim uma máscara tem a finalidade de mostrar ao usuário como o campo deve ser preenchido. Para os campos que utilizam HTML 5 não é necessário a criação de uma página para armazenar os códigos das máscaras utilizadas, por existir na própria biblioteca da linguagem mecanismos para tal finalidade. Porém nos campos que utilizam *Java Script* é necessário armazenar seus códigos em uma página chamada *maskaras.js*, permitindo que as máscaras sejam utilizadas sempre que necessário.

- Banco de Dados

O banco de dados possui mecanismos de segurança próprio. Campos das tabelas que são inteiros jamais poderão armazenar informações do tipo strings (conjunto de caracteres alfanuméricos). Outra forma de segurança é obrigar que os campos da tabela não sejam nulos (Figura 11), ou seja, para armazenar informações de um usuário na tabela clientes, será necessário que todos os campos possuam o atributo *NOT NULL*, obrigando, desta forma, o armazenamento de informações nestes campos. Caso algum campo obrigatório não contenha informação para armazenar será reportado um erro, evitando inconsistências futuras no banco de dados.

Figura 11: *NOT NULL* obriga a inserção de dados em campo do banco.

```
CREATE TABLE Cliente (  
  cd_cliente INT NOT NULL AUTO INCREMENT,  
  nm_cliente VARCHAR(5) NOT NULL,  
  PRIMARY KEY(cd_cliente)  
);
```

Fonte: Aatoria própria.

### 3.2.2 Módulo II - Gestão de Usuários

Este módulo é responsável pelo gerenciamento de usuários, permitindo que cada um que acesse o sistema tenha sua própria sessão independente de quantos

usuários estejam utilizando a aplicação ao mesmo tempo. Este módulo é responsável por manter todas as sessões funcionando sem conflitos indesejados. Também é responsável pela recuperação de senha caso o usuário não lembrem sua senha de acesso e pelo contato com a empresa mesmo que ele não tenha uma conta previamente cadastrada no sistema.

- Conta de Usuário

Para que um usuário possa ter acesso ao sistema será necessário efetuar um cadastro. Após o cadastro o usuário terá uma conta com um id (identificador) próprio. A partir desse ponto, o id será único e não poderá ser alterado como são modificadas outras informações no cadastro como, por exemplo, e-mail, endereço e outros dados pessoais.

- Sessão

A sessão é indispensável para qualquer sistema que queira ter o mínimo de segurança com as informações de seus usuários. Desde o momento do *login* cada usuário terá sua própria sessão, que será criada sempre que um usuário realizar *login* no sistema. A sessão é responsável por gerenciar todos os usuários que efetuarem *login*. Informações contidas na sessão de cada usuário serão utilizadas para realizar diversas atividades como: verificar se o usuário possui nível suficiente para acessar a página, acessar dados do usuário para exibir informações de veículos, usuários, serviços, perfil de usuário, reservas e relatórios (usuários, serviços, reservas e veículos) e inserir ou alterar dados do usuário contidos no banco de dados. Cada usuário terá uma quantidade de tempo mínima que o sistema poderá ficar sem ser utilizado. Caso esse período de tempo seja atingido a aplicação será finalizada automaticamente, ou caso o usuário necessite ele mesmo poderá finalizar sua sessão.

- Áreas de Usuário

Após efetuar o cadastro e o sistema gerar sua sessão, o usuário terá acesso a sua página. Esta página contém informações de ações que o usuário pode efetuar como, cadastrar, alterar e desativar veículos, agendar reservas de veículos, alterar perfil (foto e informações de conta) e gerar relatórios. Com todas essas informações em uma única página, o cliente terá acesso total as informações que necessita para realizar suas funções.

### 3.2.3 Módulo III – Reservas

O módulo de reservas gerencia a principal função do sistema, que é o gerenciamento de reservas efetivadas pelos clientes. O limite de reservas por dia é determinada pela quantidade de mecânicos que a concessionária possui, sendo que o cliente a qualquer momento poderá desativar a reserva feita caso seja necessário, mas a reserva permanecerá armazenada no banco de dados para que o gerente tenha um maior controle sobre os clientes que realizam esse tipo de ação. Também ficarão armazenadas as reservas que foram feitas, mas não foram efetivadas devido à ausência do veículo na data marcada. Esses dados serão importantes quando o gerente for realizar alguma promoção para seus clientes, ou quando for feito uma relação de clientes que realizaram esse tipo de procedimento.

- Reserva

As reservas possui algumas restrições como: no momento da reserva o cliente só poderá realizar uma reserva por veículo; caso o dia escolhido para a reserva não esteja disponível, será disponibilizada uma nova reserva em uma data diferente, ou caso o horário escolhido não estiver disponível será pedido uma nova reserva em um horário diferente. Essas restrições são fundamentais para evitar informações duplicadas no banco de dados, evitando transtornos a concessionária na hora de realizar a manutenção dos veículos reservados, mantendo, maior integridade dos dados armazenados no banco de dados.

- Serviços Prestados

Antes de realizar qualquer reserva, o mecânico verifica no sistema quais reservas marcadas no dia, possibilitando maior controle de reservas e de serviços que o mecânico poderá realizar. No momento da reserva o cliente especifica o que deve ser realizado na manutenção do veículo. Caso seja feito algum serviço adicional o mecânico informa os serviços extras, ou caso o veículo esteja ausente na hora da revisão, o mecânico marca a reserva como ausente.

- Veículos

O sistema permite que somente uma reserva seja feita para cada veículo, mas por medida de segurança sempre que uma reserva de veículo for desativada, será verificado se existi alguma reserva marcada para o automóvel em questão. Caso exista uma reserva, a aplicação retorna uma mensagem, e o cliente pode escolher se deve desativar a reserva em espera ou cancelar a operação. Essa verificação é necessária para evitar a inserção de informações falsas e/ou inconsistentes no banco de dados.

#### 3.2.4 Módulo IV - Dados

O sistema *web* interagem com o banco de dados e este, e por sua vez, contém informações fornecidas por usuários que utilizam a aplicação. Usuários atualizam (inclusão, alteração e exclusão) dados, ou seja, todas as funcionalidades do software necessitam das informações contidas no banco de dados, desde o momento do *login* até o momento da reserva feita pelos clientes. O mecânico necessita das informações fornecidas pelos clientes, a atendente para gerar o relatório de reserva necessita que o mecânico realize os serviços necessários, e o gerente, por sua vez, para gerenciar as informações e o negócio necessita de dados consistentes para administrar e tomar decisões para garantir o sucesso da organização.

- Cadastrar

A função cadastrar tem como objetivo inserir dados no banco de dados, dados que são primordiais para o funcionamento do sistema. Com essa função é

possível inserir novos veículos, cadastrar novos usuários, inserir serviços prestados por mecânicos e realizar reservas.

- Alterar

Esta função é utilizada sempre que for necessário alterar alguma informação no banco de dados, ou seja, quando um usuário alterar sua foto, perfil, veículos, reservas ou quando o mecânico alterar o status de uma reserva para efetuada ou desativada. O gerente utilizará frequentemente esta função sempre que for necessário alterar alguma informação de perfil de algum usuário, veículo, reservas e promoções. Com esta função é possível resolver problemas que venham a ocorrer com informações contidas no banco de dados.

- Desativar

Por medida de segurança o sistema não exclui nenhuma informação do banco de dados, mas altera o status nas tabelas de ativo para não ativo, possibilitando ao usuário recuperar suas informações quando quiser. Os dados armazenados de forma consistente auxiliam na tomada de decisões do gerente. Por exemplo: quando for planejada uma promoção para os clientes que estão desativados, a ação que tem como objetivo influenciar os clientes a voltar a utilizar o sistema novamente, ou quando um usuário desativa sua conta, caso volte a usar a aplicação novamente não será necessário criar uma nova conta, apenas recuperar a conta existente no banco de dados.

- Buscar

Esta rotina tem como meta buscar informações no banco de dados, seja para alterá-las ou exibi-las. Esta função fará uma busca no banco comparando informações dadas pelo usuário, sendo utilizada frequentemente para recuperação de dados de clientes, veículos, reservas, serviços e promoções.

### 3.3 Requisitos do Sistema

Os requisitos do sistema se dividem em dois, funcionais e não funcionais. Requisitos funcionais são os requisitos que um usuário pode realizar utilizando o sistema, ou seja, quando um cliente efetua seu cadastro. Requisitos não funcionais são requisitos relacionados ao uso da aplicação como, por exemplo: confiabilidade, desempenho, segurança, disponibilidade e usabilidade.

“A atividade de requisitos consiste na tradução de um cenário complexo e coeso, descrito em diversas linguagens em meios, em uma linguagem única que possa orientar a equipe de desenvolvimento”. (Fagundes, 2011).

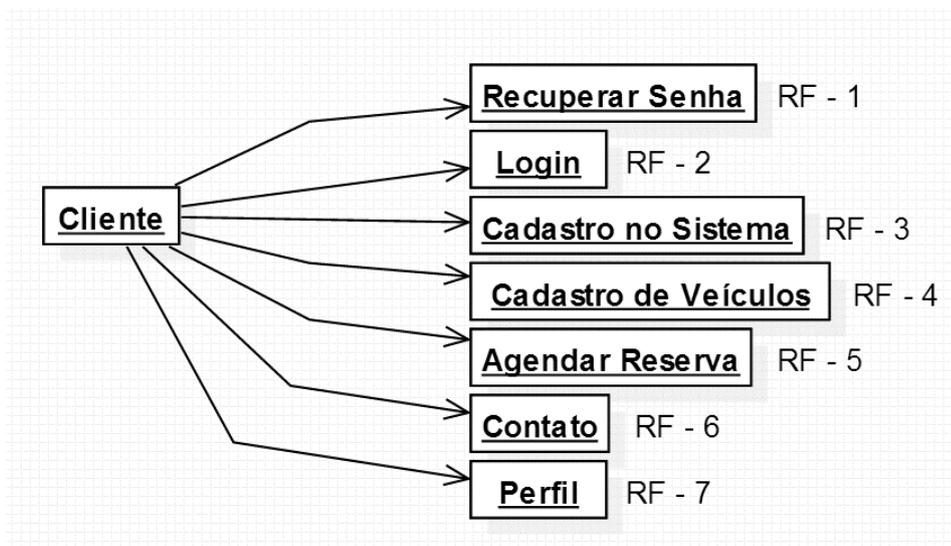
#### 3.3.1 Requisitos Funcionais

Aqui serão exibidos detalhadamente cada requisito funcional do sistema, que serão divididos pelos tipos de usuários.

##### 3.3.1.1 Cliente

Abaixo uma breve descrição acerca dos requisitos funcionais pertencentes ao usuário Cliente, demonstrados na Figura 12.

Figura 12: Requisitos funcionais referente ao Cliente.



Fonte: Autoria própria.

- Recuperar Senha

Ao iniciar a recuperação de senha, será enviado um e-mail para o e-mail do usuário cadastrado, permitindo que o usuário recupere sua senha sem a necessidade de entrar em contato com a concessionária.

- *Login*

O *login* gerencia a autenticação de usuários no sistema, dependendo do nível de acesso que é determinada pelo tipo de usuário (cliente e funcionário (Gerente, Mecânico e Atendente)) que acessar a aplicação *web*, após a autenticação o *login* direciona o usuário para a sua página inicial.

- Cadastro no Sistema

Caso o usuário não possua cadastro, o sistema disponibiliza uma área onde o usuário possa realizar seu cadastro, possibilitando realizar o *login*, e conseqüentemente utilizar das funcionalidades disponíveis.

- Cadastro de Veículos

O cadastro de veículos possibilita ao cliente efetuar o cadastro de seus veículos, disponibilizando total controle de todos os veículos cadastrados pertencentes ao cliente.

- Agendar Reserva

A reserva possibilita aos clientes efetuarem o agendamento de seus veículos, sendo permitido um agendamento por veículo cadastrado. Caso o veículo possua um agendamento pendente, só será possível efetuar um novo agendamento desse veículo se o agendamento pendente for cancelado.

- Contato

Caso o usuário queira entrar em contato com a concessionária, na área de contato o usuário poderá enviar um e-mail para a concessionária, que entrará em contato com o usuário assim que possível.

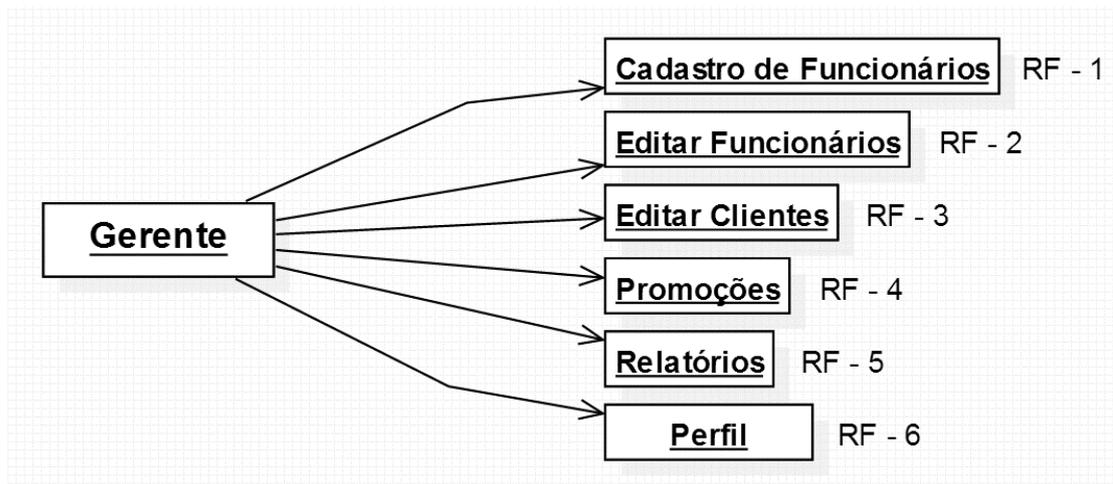
- Perfil

Esta funcionalidade permite que o cliente (usuário) altere informações de seu perfil, sem a necessidade de entrar em contato com o administrador do sistema, possibilitando um maior controle sobre suas informações armazenadas no banco de dados.

### 3.3.1.2 Gerente

Abaixo uma breve descrição acerca dos requisitos funcionais pertencentes ao usuário Gerente, demonstrados na Figura 13.

Figura 13: Requisitos Funcionais referente ao funcionário (Gerente)



Fonte: Autoria própria.

- Cadastro de Funcionários

O cadastro de funcionário permite ao gerente cadastrar novos funcionários (mecânicos e atendentes) sempre que necessário.

- Editar Funcionários

A funcionalidade editar funcionários possibilita ao gerente gerenciar melhor as informações contidas no banco de dados, alterando os dados de seus funcionários quando for necessário.

- Editar Clientes

Esta funcionalidade permite que o gerente busque, edite ou desative qualquer cliente cadastrado no sistema.

- Promoções

Esta funcionalidade permite ao gerente gerar diversos tipos de promoções, ou seja, cada promoção depende do tipo de cliente que o gerente deseja atingir. Sempre que estiver disponível um novo modelo de veículo, o gerente poderá criar uma promoção específica para os clientes que tiverem o modelo antigo, enviando uma promoção de desconto, ou criar promoções oferecendo desconto em serviços disponibilizados pela concessionária.

- Relatórios

O gerente poderá gerar diversos tipos de relatórios como: relatório de clientes (exibindo seu perfil e veículos cadastrados), de funcionários (atendente e mecânico), relatório de reserva (informações da reserva e serviços prestados) e relatório de promoções (exibindo informações da promoção), possibilitado maior controle e auxiliando na tomada de decisão.

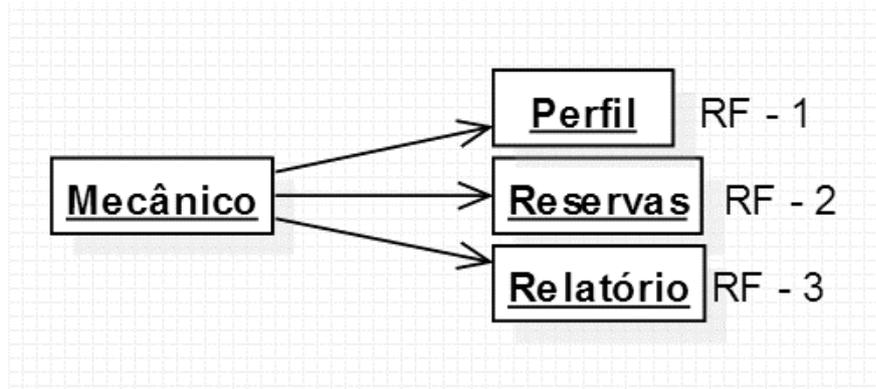
- Perfil

Funcionalidade descrita no tópico 3.3.1.1 Cliente RF - 7.

### 3.3.1.3 Mecânico

Abaixo uma breve descrição acerca dos requisitos funcionais pertencentes ao usuário Mecânico, demonstrados na Figura 14.

Figura 14: Requisitos Funcionais referente ao funcionário (Mecânico).



Fonte: Autoria própria.

- Perfil

Funcionalidade descrita no tópico 3.3.1.1 Cliente RF - 7.

- Reservas

As reservas possibilitam ao mecânico gerenciar de forma eficiente as reservas agendadas do dia, ou seja, apenas será exibido as reservas do dia contendo as seguintes informações: cliente, modelo do veículo, horário marcado e tipo de serviço solicitado. O mecânico poderá selecionar a reserva desejada e cadastrar todos os serviços realizados na reserva e, caso necessário, poderá deixar uma observação sobre a reserva e/ou serviços extras que não foram solicitados na reserva.

- Relatório

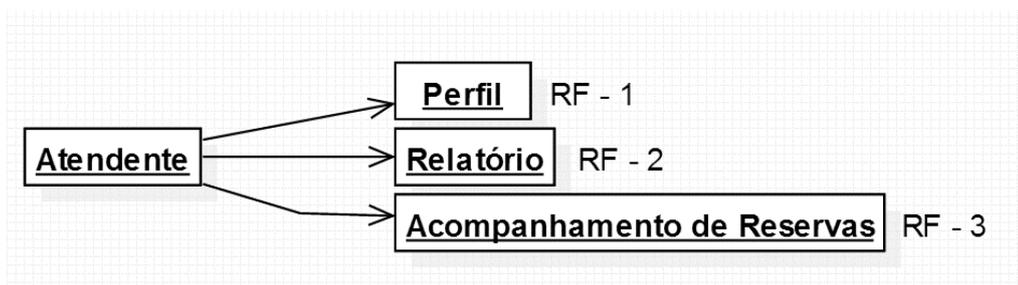
Visando uma forma de comprovar os serviços prestados, o mecânico poderá gerar relatórios exibindo os serviços prestados em cada reserva durante o dia,

possibilitando ao mecânico ter um controle dos serviços realizados em sua jornada de trabalho.

#### 3.3.1.4 Atendente

Abaixo uma breve descrição acerca dos requisitos funcionais pertencentes ao usuário Atendente, demonstrados na Figura 15.

Figura 15: Requisitos Funcionais referente ao funcionário (Atendente).



Fonte: Autoria própria.

- Perfil

Funcionalidade descrita no tópico 3.3.1.1 Cliente RF - 7.

- Relatório

Sempre que uma reserva é finalizada, um relatório será gerado pela atendente, exibindo informações da reserva e dos serviços prestados ao veículo, dessa forma, o cliente terá acesso a todas as informações necessárias sobre o que foi realizado no seu veículo.

- Acompanhamento de Reservas

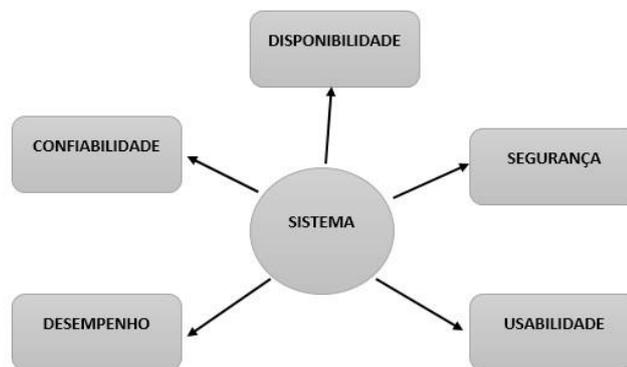
Caso necessite, a atendente poderá acompanhar a reserva de um cliente, verificando o status da reserva, que variam entre: desativada (0), em andamento (1) e finalizada (2). Dessa forma a atendente poderá informar ao cliente como está o processo de manutenção de seu veículo.

A exibição das telas dos principais requisitos funcionais será exibida no tópico 4.4 *Interface* dos requisitos funcionais.

### 3.3.2 Requisitos não funcionais

Aqui serão explicados detalhadamente cada requisito não funcional do sistema, que serão divididos em cinco tipos: confiabilidade, desempenho, segurança, disponibilidade e usabilidade (Figura 16).

Figura 16: Requisitos não Funcionais.



Fonte: A autoria própria.

- Confiabilidade

Confiabilidade é a capacidade do sistema em lidar com eventos inesperados, tratando exceções e recuperando-se de falhas, ou seja, quando uma ação é realizada, o sistema deve ser capaz de realizar a recuperação de falhas caso a ação não seja efetuada corretamente.

Um exemplo: quando um cliente deseja desativar seu veículo, caso exista um agendamento de serviços reservado para o veículo, ocorrerá uma inconsistência no banco de dados pelo motivo da reserva que foi solicitada para o veículo. Nesse momento o sistema solicitará ao cliente se o mesmo deseja desativar a solicitação de agendamento do veículo.

- Desempenho

Um grande diferencial ao se desenvolver um sistema, é seu desempenho, ou seja, nada adianta ter um sistema seguro e com diversas funcionalidades, se sua interface (*Design* do Sistema) ou suas ações (cadastro, busca e alteração de dados) não possuir um desempenho satisfatório.

Utilizando técnicas de IHC e programação orientada a objetos, foi desenvolvido um sistema capaz de realizar funcionalidades sem perda de performance, mesmo a aplicação sendo utilizado por diversos usuários.

- Segurança

A segurança é primordial para que um sistema seja capaz de desempenhar suas funções de acordo com o que foi projetado, ou seja, foram implementados diversos mecanismos de segurança como: restringir as páginas liberando-as de acordo com o perfil do usuário; proteger as funções do sistema em uma página específica, sendo solicitada apenas via código restringindo o acesso a usuários mal intencionados. Todas as técnicas de segurança adotadas são descritas no tópico 4.2.1 Módulo I – Segurança.

- Disponibilidade

Um sistema *web* possui maior disponibilidade do que outras aplicações desenvolvidos em outras plataformas. Possibilita o acesso em qualquer lugar do mundo, necessitando apenas de internet, e de qualquer dispositivo (celulares, *tablets* e notebooks). Essa característica permite uma disponibilidade que nenhum outro sistema feito em outras plataformas podem alcançar.

- Usabilidade

Um sistema possui uma boa usabilidade quando se torna fácil de operar possibilitando ao usuário realizar suas funcionalidades sem nenhum problema, tornando-o prático, ou seja, quanto mais simples for a usabilidade, maior será a aceitação dos usuários.

### 3.4 Interface dos requisitos funcionais (Tópico 3.3.1)

A seguir serão apresentadas as *interfaces* dos principais requisitos funcionais de acordo com o perfil de cada usuário.

#### 3.4.1 Cliente

- Cadastro de Veículos

Abaixo a Figura 17 demonstra o requisito funcional (Cadastro de veículos) pertencente ao usuário Cliente, que possibilita o cliente cadastrar seus veículos sem quaisquer dificuldades.

Figura 17: Página de cadastro de veículos.

Modelo	Placa	Cor	Editar	Desativar
astra	123156465	branco	Editar	Desativar
novo gol	12332132	vermelho	Editar	Desativar
uno	12332132	azul	Editar	Desativar

Fonte: Autoria própria.

- Agendar de Reserva

A Figura 18 demonstra o requisito funcional (Agendamento de veículos) pertencente ao usuário Cliente, que possibilita o cliente agendar a reserva de seu veículo no dia e horário desejado.

Figura 18: Página de agendamento de veículos.

INÍCIO VEÍCULOS AGENDAR REVISÃO SAIR

Modelo	Placa	Cor	Selecionar
astra	123156465	branco	Selecionar
gol	12332132	vermelho	Selecionar
uno	12332132	azul	Selecionar

**Reserva**

**Veículo Selecionado**

Modelo: astra Cor: branco

Placa: 123156465 Nº Motor: 1324561231

Nº Chassis: 2147483647 Nº Concessionário: 1321564321

Data Garantia: 31/12/2013

de dezembro de 2014

dom	seg	ter	qua	qui	sex	sáb
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3

dd/mm/aaaa Seleção Seleção

**Efetuar Reserva**

Fonte: Autoria própria.

### 3.4.2 Gerente

- Promoções

A Figura 19 e 20 demonstra o requisito funcional (Promoções) pertencente ao usuário Gerente, que possibilita o gerente realizar diversos tipos de promoções para cada cliente desejado.

Figura 19: Página de promoções divididos por categoria.



Fonte: Autoria própria.

Figura 20: Página de promoção geral.

A interface de usuário para a página de criação de uma promoção, intitulada "PROMOÇÃO".  
**Dados da Promoção**  
Título:   
Data de Encerramento:   
Informação da Promoção:   
No canto inferior esquerdo, há um botão azul com o texto "Enviar".

Fonte: Autoria própria.

- Relatórios

A Figura 21 e 22 demonstra o requisito funcional (Relatórios) pertencente ao usuário Gerente, que possibilita o gerente gerar relatórios com informações específicas do banco de dados.

Figura 21: Páginas de relatórios divididos por categoria.



Fonte: Autoria própria.

Figura 22: Relatório do cliente.

Relatório do Cliente

::Dados Pessoais::

Nome: Leonardo Neiva Teixeira da Silva Moura

Sexo: Masculino

Rg: 13.245.111-111

Cpf: 111.111.111-11

Data de Nascimento: 1989-07-20

::Dados de Contato::

Telefone: (01) 1111-1111

E-mail: leo@gmail.com

Endereço: Rua da Palmeira

Número: 123

Bairro: centro

Cidade: Picos

Estado: piaui

CEP: 64600000

Veiculos do Cliente

Modelo	Placa	Nº Chassis	Data Garatia	Cor	Nº Motor	Nº Concessionário
uno	PQM0001	1321323132	2013-12-30	azul	32132321	2147483647
novo gol	BFG8822	2147483647	2013-12-31	vermelho	46589764	2147483647
astra	TDG2556	2147483647	2013-12-31	branco	24561231	1321564321

Fonte: Autoria própria.

### 3.4.3 Mecânico

- Reservas

A Figura 23 demonstra o requisito funcional (Serviços) pertencente ao usuário Mecânico, que possibilita o mecânico adicionar novos serviços prestados ao veículo da reserva atual.

Figura 23: Cadastro, Edição e Exclusão de Serviços.

Serviço	Comentário	Editar	Excluir
Troca de Oleo	Oleo de 2.000 Quilômetros.	Editar	Excluir

Fonte: Aatoria própria.

Por motivos de usabilidade, em uma única página o mecânico poderá realizar três ações: cadastrar, editar e excluir serviços, possibilitando maior eficiência ao realizar a manutenção nos veículos.

### 3.4.4 Atendente

A Figura 24 demonstra o requisito funcional (Acompanhamento de reservas) pertencente ao usuário Atendente, que possibilita a atendente acompanhar o status de uma reserva.

Figura 24: Página de acompanhamento de reserva.

**STATUS DA RESERVA**

Aguardando      Em andamento      Concluído

**Dados da Reserva**

Cliente:	Leonardo Neiva Teixeira da Silva Moura	Veículo:	novo gol
Status:	Aguardando	Data da Reserva:	2014-12-17
Horário:	Manha	Serviço:	Troca de oleo

Fonte: Aatoria própria.

### 3.5 Funções e Códigos

Ao desenvolver um sistema, o programador deve ser capaz de desenvolver códigos organizados, utilizando orientação a objetos visando reaproveitamento de código. Utilizando essas técnicas, foram desenvolvidas funções que realizam as funcionalidades do sistema. Abaixo uma breve descrição sobre trechos de códigos desenvolvidos.

- Cadastrar

A função cadastrar veículo está disponível na página do usuário Cliente. Quando solicitada, ela recebe as informações da página de cadastro de veículos e envia a inserção das informações para o banco de dados utilizando instruções SQL, demonstração do código de inserção de veículos na Figura 25.

Figura 25: Código de cadastramento de veículo.

```

//-----
//INSERINDO VEÍCULO
if ($_GET['funcao']=="gravar_veiculo"){
    $get_email= $_SESSION['login'] ;
    $get_senha = $_SESSION['senha'];
    $grava_modelo = $_POST['modelo'];
    $grava_placa = $_POST['placa'];
    $grava_n_chassis = $_POST['n_chassis'];
    $grava_data_g = $_POST['data_g'];
    $grava_cor = $_POST['cor'];
    $grava_n_motor = $_POST['n_motor'];
    $grava_n_concessionario = $_POST['n_concessionario'];

    $result = mysql_query("SELECT * FROM `clientes` WHERE email = '$get_email'
        AND senha = '$get_senha' AND nivel = '1'");

    while($linha = mysql_fetch_array($result)){
        $grava_id = $linha['idcliente'];}

    $sql_gravar = mysql_query("INSERT INTO veiculos (modelo,placa,n_chassis,data_g,cor,
        n_motor,n_concessionario,status,clientes_idcliente) value ('$grava_modelo',
        '$grava_placa','$grava_n_chassis','$grava_data_g','$grava_cor','$grava_n_motor',
        '$grava_n_concessionario','1','$grava_id')");
    ?>
    <script>
        alert("Veiculo cadastrado com sucesso!")
        parent.location.href = "cliente_veiculos.php";
    </script>
    <?
}

```

Fonte: Autoria própria.

- Alterar

Ao ser solicitada, a função editar veículo recebe as informações que deve alterar no banco de dados. Em seguida é feita uma solicitação ao banco, caso o id do veículo selecionado exista no banco de dados, as informações serão alteradas através de instruções SQL, demonstração do código Alterar veículo na Figura 26.

Figura 26: Código de alteração de veículo.

```

//EDITANDO VEÍCULO
if ($_GET['funcao'] == "editar_veiculo"){
    $grava_id = $_GET['id'];
    $grava_modelo = $_POST['modelo'];
    $grava_placa = $_POST['placa'];
    $grava_n_chassis = $_POST['n_chassis'];
    $grava_data_g = $_POST['data_g'];
    $grava_cor = $_POST['cor'];
    $grava_n_motor = $_POST['n_motor'];
    $grava_n_concessionario = $_POST['n_concessionario'];
    $sql_alterar = mysql_query("UPDATE veiculos SET modelo='$grava_modelo', placa='$grava_placa',
    n_chassis='$grava_n_chassis', data_g='$grava_data_g' , cor='$grava_cor'
    , n_motor='$grava_n_motor' , n_concessionario='$grava_n_concessionario'
    WHERE idveiculos = '$grava_id'");
    ?>
    <script>
        alert("Veiculo alterado com sucesso!")
        parent.location.href = "cliente_veiculos.php";
    </script>
<?
}

```

Fonte: Autoria própria.

- Desativar

Informações contidas no banco de dados jamais devem ser excluídas, mas sim desativadas, possibilitando o cliente recuperar informações de seu veículo desativado, evitando um novo cadastro. Ao encontrar o id do veículo selecionado, será alterado apenas o status de ativo (1) para desativado (0), demonstração do código desativar veículo na Figura 27.

Figura 27: Código para desativar um veículo.

```

//DESATIVANDO VEÍCULO
if ($_GET['funcao'] == "desativar_veiculo"){
    $grava_id = $_GET['id_veiculo'];
    $sql_desativar = mysql_query("UPDATE veiculos
    SET status = '0' WHERE idveiculos = '$grava_id'");
    ?>
    <script>
        alert("Veiculo desativado!, para utilizar novamente o veiculo sera preciso ativa-lo!")
        parent.location.href = "cliente_veiculos.php";
    </script>
<?
}

```

Fonte: Autoria própria.

- Buscar

O método buscar é muito utilizado no sistema para exibir informações em tabelas, como demonstra a Figura 28. Estão disponíveis três níveis de buscas: a

primeira recupera as últimas sete reservas cadastradas no banco de dados; a segunda recupera o nome do cliente da reserva; e por último a terceira busca encontra o modelo do veículo. As buscas são importantes devido ao fato de poderem recuperar as informações desejadas, podendo exibi-las em qualquer parte da página.

Figura 28: Código de busca de reservas.

```
$id = $_GET['id'];
$sql_editar = mysql_query("SELECT * FROM reservas order by data_r desc limit 7 ");
while($linha = mysql_fetch_array($sql_editar)){
    $id_reserva = $linha['idreservas'];
    $id_cliente = $linha['clientes_idcliente'];
    $id_veiculo = $linha['Veiculos_idveiculos'];
    $data = $linha['data_r'];
    $hora = $linha['hora_r'];
    $servico = $linha['servico'];

    $sql_cliente = mysql_query("SELECT nome FROM clientes WHERE idcliente = $id_cliente");
    if($linha = mysql_fetch_array($sql_cliente)){ $id_cliente_nome = $linha['nome'];

    $sql_veiculo = mysql_query("SELECT modelo FROM veiculos WHERE idveiculos = $id_veiculo");
    if($linha = mysql_fetch_array($sql_veiculo)){ $modelo = $linha['modelo'];
```

Fonte: Autoria própria.

## 4 RESULTADOS

Todo projeto deve ter uma fase de testes. É nessa fase que são descobertos erros não encontrados no processo de desenvolvimento, possibilitando ao desenvolvedor realizar a manutenção que for necessária. Sempre que um aplicativo não funcionar corretamente, provavelmente é devido à má realização de testes funcionais, e conseqüentemente é disponibilizado no mercado um produto incompleto. A fase de testes também é utilizada para demonstrar ao cliente o produto que está adquirindo, permitindo ter uma visão de todo o funcionamento da aplicação.

Abaixo uma explanação dos testes alcançados divididos por categorias: usabilidade, interface, disponibilidade e funcionalidade.

### 4.1 Quantidade de Usuários (Usabilidade e Interface)

O teste tem como objetivo verificar a capacidade do sistema em gerenciar usuários, verificando a performance com diversas sessões ativas.

Nesse teste o sistema se mostrou capaz de gerenciar dezenas de usuários como é demonstrado no Gráfico 1, não demonstrando erros ao efetuar o *login*, redirecionando o usuário para sua respectiva área de acordo com seu perfil pré-definido.

Alguns erros foram encontrados na página pertencente ao funcionário atendente. O erro possibilitava o funcionário mecânico ao clicar em início, carregar a página inicial do funcionário atendente, ocasionando erros na sessão que obrigava o funcionário mecânico realizar o *logout*.

Gráfico 1: Percentual de usuários que utilizaram o sistema na fase de testes.

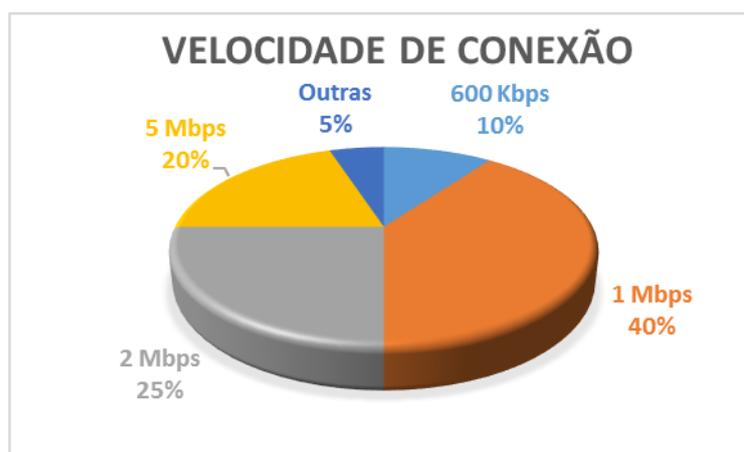


Fonte: Aatoria própria.

## 4.2 Velocidade de Conexão (Disponibilidade)

Devido ao sistema ser online houve uma preocupação adicional com relação a velocidade média de conexão utilizada pelos clientes. Por essa razão, foram realizados testes com diferentes níveis de conexão (600 Kbps, 1Mbps, 2Mbps, 5 Mbps e outras) demonstrados no Gráfico 2.

Gráfico 2: Média da velocidade de conexão utilizada pelos usuários.



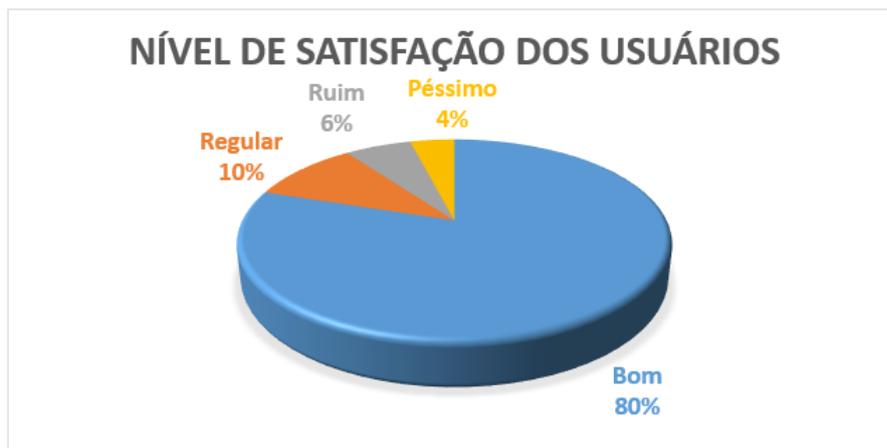
Fonte: Aatoria própria.

Ao realizar os testes, foi identificado demora no carregamento das páginas utilizando conexões de 600 Kbps e 1 Mbps. Devido a esse problema foram reduzidos o tamanho e a qualidade das imagens (*background*, imagens de exibição do *slider* na página inicial e logotipos em geral) utilizadas, possibilitando os clientes carregarem as páginas mais rapidamente.

## 4.3 Funcionalidades (Funcionalidade)

O teste de funcionalidade tem como objetivo demonstrar para os usuários se as funcionalidades disponibilizadas atendem suas necessidades, tal qual definidas no processo de levantamento de requisitos. Permite aos usuários testar, e caso tenham algo a acrescentar e/ou modificar, essas necessidades serão atendidas.

Gráfico 3: Nível de satisfação dos usuários na fase de testes de funcionalidade.



Fonte: Autoria própria.

Como demonstra o Gráfico 3, 80% dos usuários aprovaram o sistema, mas novas funcionalidades foram exigidas como o acompanhamento da reserva e aviso de término da manutenção.

- Acompanhamento de Reserva

O acompanhamento de reserva terá a função de expor o andamento do processo de manutenção do veículo. Atualmente, essa funcionalidade pertence apenas ao perfil da atendente. No futuro essa função será migrada para o perfil do cliente, auxiliando-o no acompanhamento de sua reserva.

- Aviso de Término da Manutenção

Devido há alguns serviços na manutenção demorarem mais do que o previsto, será implementado um mecanismo de envio de e-mail, ou seja, quando a manutenção da reserva for terminada, o cliente receberá uma mensagem em seu celular avisando-o para ir a concessionária buscar seu veículo.

## 5 CONSIDERAÇÕES FINAIS

O presente trabalho apresentou o processo de desenvolvimento de um sistema gerencial de agendamento e serviços para uma concessionária de veículos, possuindo uma funcionalidade diferenciada de outros sistemas utilizados no mercado, que é a área de promoções. Esta área permite ao usuário com perfil de gerente criar novas promoções para clientes específicos, possibilitando maior estratégia de *Marketing* para a concessionária.

Algumas dificuldades surgiram devido ao uso do servidor *web*. Servidores que atendessem todos os requisitos propostos eram pagos e caros. Para que os testes fossem realizados com êxito, a aplicação foi armazenado em servidores fora do Brasil, por serem gratuitos por um curto período de tempo, limitando bastante o tempo dos testes.

Com os testes realizados, foram identificados erros de redirecionamento de páginas e de sessão que foram corrigidos. Algumas implementações referentes ao cliente deverão ser implementadas futuramente, permitindo ao cliente ter um acompanhamento online de sua reserva e receber uma mensagem em seu celular quando a reserva for finalizada.

Como o sistema foi desenvolvido em módulos e possui um projeto de banco de dados, poderá ser usado sem quaisquer dificuldades, possibilitando a implementação de novas funcionalidades e permitindo ser modificado e implantado em outras concessionárias, ou servi como base para futuros trabalhos acadêmicos.

Como possíveis trabalhos futuros, pode-se apresentar:

- I. Implementar uma área, onde o cliente poderá acompanhar o processo de manutenção de seu veículo, obtendo informações de serviços antes mesmo que a manutenção termine.
- II. Implementar um novo tipo de cadastro de usuário que permita uma empresa se cadastrar e utilizar os serviços disponíveis. Assim uma empresa (locadora de veículos, por exemplo) poderá agendar serviços de todos os seus veículos, possibilitando a concessionária possuir dois tipos de clientes, físicos e jurídicos.

**III.** Desenvolver uma versão do sistema para a plataforma *mobile*.

## REFERÊNCIAS

Augusto Fabrício Ferrati **Crie Banco de Dados em Mysql**. São Paulo: Digerati Books, 2007. 6.p.

**CSS3 – Curso W3C Escritório Brasil**. Disponível em:

<<http://www.w3c.br/Cursos/CursoCSS3>> Acessado em Jul.2014.5.p.

Fagundes R. M. **Engenharia de Requisitos: do perfil do analista de requisitos ao desenvolvimento de requisitos com UML e RUP**. Salvador, 2011.

Ferreira Elcio, Eis Diego. **Html5 – Curso W3c Escritório Brasil**.2010.10.p.

Heuser, Carlos Alberto. **Projeto de Banco de Dados**. 6ª Edição. Porto Alegre: Bookman, 2009.23.p

Heuser, Carlos Alberto. **Projeto de Banco de Dados**. 4ª Edição. Porto Alegre: Sagra Luzzatto, 1998.22.p

Niederauer, Juliano. **Desenvolvendo Websites com PHP**. 2ª Edição. São Paulo: Novatec, 2011.23.p

Php.net. **Injeção de SQL**. Disponível em :<

[http://php.net/manual/pt\\_BR/security.database.sql-injection.php](http://php.net/manual/pt_BR/security.database.sql-injection.php)> Acessado em Dez. 2014.

## **APÊNDICE**

## APÊNDICE A – Questionário utilizado nos testes.

### Usuário

Nome Completo:

Sexo:

### Questionário

Tipo de usuário:

Velocidade de Conexão:

Nível de Satisfação do sistema:

## APÊNDICE B – Funções do sistema.

```

if ($_GET['funcao']=="gravar_cliente" ){
    $grava_nome = $_POST['nome'];
    $grava_senha = $_POST['senha'];
    $grava_sexo = $_POST['sexo'];
    $grava_fone = $_POST['fone'];
    $grava_email = $_POST['email'];
    $grava_data_n = $_POST['data_n'];
    $grava_cpf = $_POST['cpf'];
    $grava_rg = $_POST['rg'];
    $grava_logradouro = $_POST['logradouro'];
    $grava_numero = $_POST['numero'];
    $grava_bairro = $_POST['bairro'];
    $grava_cidade = $_POST['cidade'];
    $grava_estado = $_POST['estado'];
    $grava_cep = $_POST['cep'];

    $sql = mysql_query("INSERT INTO clientes (nome,senha,sexo,fone,email,
data_n,cpf,rg,logradouro,numero,bairro,cidade,estado,cep,nivel,status)
|value ('$grava_nome','$grava_senha','$grava_sexo','$grava_fone','$grava_email',
'$grava_data_n','$grava_cpf','$grava_rg','$grava_logradouro','$grava_numero',
'$grava_bairro','$grava_cidade','$grava_estado','$grava_cep','1','1')");
?>
<script>
    alert("Cadastro efetuado com sucesso!")
    parent.location.href = "index.php";
</script>
<?
}

```

```

if ($_GET['funcao'] == "editar_cliente"){
    $grava_id = $_GET['id'];
    $grava_nome = $_POST['nome'];
    $grava_sexo = $_POST['sexo'];
    $grava_cidade = $_POST['cidade'];
    $grava_senha = $_POST['senha'];
    $grava_email = $_POST['email'];
    $grava_data_n = $_POST['data_n'];
    $grava_fone = $_POST['fone'];
    $grava_logradouro = $_POST['logradouro'];
    $grava_estado = $_POST['estado'];
    $grava_cep = $_POST['cep'];
    $grava_cpf = $_POST['cpf'];
    $grava_rg = $_POST['rg'];
    $grava_numero = $_POST['numero'];

    $sql_alterar = mysql_query("UPDATE clientes SET nome='$grava_nome',
senha='$grava_senha', sexo='$grava_sexo', cidade='$grava_cidade',
fone='$grava_fone', email='$grava_email', data_n ='$grava_data_n',
cpf='$grava_cpf', logradouro='$grava_logradouro', estado='$grava_estado',
cep='$grava_cep', nivel = '1', status = '1', rg='$grava_rg', numero='$grava_numero'
WHERE idcliente = '$grava_id'");

    ?>
    <script>
    alert("Cliente atualizado!")
    parent.location.href = "busca_clientes.php";
    </script>
    <?
}

```

```
if ($_GET['funcao'] == "atualizar_cliente"){
    $grava_id = $_GET['id'];
    $grava_nome = $_POST['nome'];
    $grava_sexo = $_POST['sexo'];
    $grava_cidade = $_POST['cidade'];
    $grava_senha = $_POST['senha'];
    $grava_email = $_POST['email'];
    $grava_data_n = $_POST['data_n'];
    $grava_fone = $_POST['fone'];
    $grava_logradouro = $_POST['logradouro'];
    $grava_estado = $_POST['estado'];
    $grava_cep = $_POST['cep'];
    $grava_cpf = $_POST['cpf'];
    $grava_rg = $_POST['rg'];
    $grava_numero = $_POST['numero'];

    $sql_alterar = mysql_query("UPDATE clientes SET nome='$grava_nome',
    senha='$grava_senha', sexo='$grava_sexo', cidade='$grava_cidade',
    fone='$grava_fone',email='$grava_email',data_n = '$grava_data_n',
    cpf='$grava_cpf', logradouro='$grava_logradouro', estado='$grava_estado',
    cep='$grava_cep',nivel = '1', status = '1' ,rg='$grava_rg',numero='$grava_numero'
    WHERE idcliente = '$grava_id'");

    $_SESSION['login'] = $grava_email;
    $_SESSION['senha'] = $grava_senha;

    ?>
    <script>
    alert("Perfil atualizado!")
    parent.location.href = "cliente.php";
    </script>
    <?
}
```



```

if ($_GET['funcao']=="realizar_reserva"){
    $id_cliente = $_GET['id_cliente'];
    $id_veiculo = $_GET['id'];
    $get_email= $_SESSION['login'];
    $get_senha = $_SESSION['senha'];

    $data = $_POST['data_reserva'];
    $horario = $_POST['horario'];
    $servico = $_POST['servico'];

    $result = mysql_query("SELECT data_r FROM reservas where data_r = '$data'");
    while($linha = mysql_fetch_array($result)){
        $cont++;
    }
    //Verifica se está disponível o dia para a reserva
    if($cont < 2){
        if($cont == "0"){
            $sql_gravar = mysql_query("INSERT INTO reservas (data_r,hora_r,
                servico,Veiculos_idveiculos,clientes_idcliente,status) value
                ('$data','$horario','$servico','$id_veiculo','$id_cliente','1')");
            ?>
            <script>
                alert("Reserva efetuada com sucesso!")
                parent.location.href = "agendar_revisao.php";
            </script>
            <?
            //Caso exista 1 reserva no dia
        }else{
            $result = mysql_query("SELECT * FROM `reservas` WHERE data_r = '$data'");
            while($linha = mysql_fetch_array($result)){
                $hora = $linha['hora_r'];
            }
            if($hora == $horario){
                ?>
                <script>
                    alert("Este horario nao esta vago, selecione um novo horario!")
                    parent.location.href = "agendar_revisao.php";
                </script>
                <?
            }else{
                $sql_gravar = mysql_query("INSERT INTO reservas (data_r,hora_r,servico,
                    Veiculos_idveiculos,clientes_idcliente,status) value ('$data','$horario',
                    '$servico','$id_veiculo','$id_cliente','1')");
                ?>
                <script>
                    alert("Reserva efetuada com sucesso!")
                    parent.location.href = "agendar_revisao.php";
                </script>
                <?
            }
        }
    }else{
        ?>
        <script>
            alert("Esta data nao possui mais horarios vagos, selecione uma nova data!")
            parent.location.href = "agendar_revisao.php";
        </script>
        <?
    }
}

```



TERMO DE AUTORIZAÇÃO PARA PUBLICAÇÃO DIGITAL NA BIBLIOTECA  
"JOSÉ ALBANO DE MACEDO"

Identificação do Tipo de Documento

- ( ) Tese  
( ) Dissertação  
( x ) Monografia  
( ) Artigo

Eu, **Leonardo Neiva Teixeira da Silva Moura**, autorizo com base na Lei Federal nº 9.610 de 19 de Fevereiro de 1998 e na Lei nº 10.973 de 02 de dezembro de 2004, a biblioteca da Universidade Federal do Piauí a divulgar, gratuitamente, sem ressarcimento de direitos autorais, o texto integral da publicação **Protótipo de um Sistema de Agendamento de Controle de Serviços de Revisão em Concessionárias de Veículos de Picos-PI** de minha autoria, em formato PDF, para fins de leitura e/ou impressão, pela internet a título de divulgação da produção científica gerada pela Universidade.

Picos-PI 19 de Fevereiro de 2015.

  
Assinatura