

**UNIVERSIDADE FEDERAL DO PIAUÍ
CAMPUS SENADOR HELVÍDIO NUNES DE BARROS
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**DESENVOLVIMENTO DE UM SISTEMA *WEB* PARA UMA OFICINA
AUTOMOTIVA**

WESLEY VIEIRA DE MOURA FÉ

WESLEY VIEIRA DE MOURA FÉ

**DESENVOLVIMENTO DE UM SISTEMA *WEB* PARA UMA OFICINA
AUTOMOTIVA**

Trabalho de conclusão de curso apresentado ao Curso de Bacharelado em Sistemas de Informação do *Campus* Senador Helvídio Nunes de Barros da Universidade Federal do Piauí como parte dos requisitos para obtenção do grau de Bacharel em Sistemas de Informação.

Orientadora: Prof^a Ma. Patricia Medyna Lauritzen de Lucena Drumond.

FICHA CATALOGRÁFICA

Serviço de Processamento Técnico da Universidade Federal do Piauí

Biblioteca José Albano de Macêdo

F288d Fé, Wesley Vieira de Moura.

Desenvolvimento de um sistema web para uma oficina
automotiva / Wesley Vieira de Moura. – 2016.

CD-ROM : il.; 4 ¾ pol. (34 f.)

Monografia(Bacharelado em Sistemas de Informação) –
Universidade Federal do Piauí, Picos, 2016.

Orientador(A): Profª Ma. Patrícia Medyna Lauritzen de Lucena
Drumond.

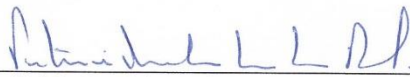
DESENVOLVIMENTO DE UM SISTEMA WEB PARA OFICINA AUTOMOTIVA

WESLEY VIEIRA DE MOURA FÉ

Monografia aprovada como exigência parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Data de Aprovação

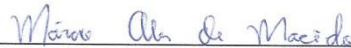
Picos – PI, 18 de fevereiro de 2016



Prof.^a. Ma. Patrícia Medyna Lauritzen de Lucena Drumond
Orientador



Prof. Esp. Allán Jheyson Ramos Gonçalves
Membro



Prof. Me. Márcio Alves de Macêdo
Membro

Dedico este trabalho em primeiro lugar a Deus e em segundo lugar aos meus pais e amigos que sempre me apoiaram em tudo o que faço, tanto nos momentos bons, como também nos momentos ruins da vida. A vocês eu devo tudo isso.

AGRADECIMENTOS

Agradeço primeiramente a Deus. Agradeço á minha família por sempre acreditar e investir nos meus estudos. A minha mãe, por estar sempre do meu lado, me ajudando nas horas mais necessitadas. Ao meu pai, por mostrar segurança e certeza de que sempre que eu precisar ele estará ao meu lado. Também agradeço a Universidade Federal do Piauí e todos os professores que me ajudaram nesta longa caminhada, como também os professores da banca e a minha orientadora.

“Para se ter sucesso, é necessário amar de verdade o que se faz. Caso contrário, levando em conta apenas o lado racional, você simplesmente desiste. É o que acontece com a maioria das pessoas.”

(Steve Jobs)

RESUMO

Este projeto visa à criação de um sistema *web* para oficinas automotivas, que necessitam de uma maior interação com seus clientes e um melhor gerenciamento de dados sobre orçamentos, clientes, funcionários, seguradoras e relatórios. A implantação do sistema que foi desenvolvido deverá facilitar a vida dos funcionários da oficina e dos clientes que sempre serão informados de todo o processo feito em seu veículo. O sistema proposto foi desenvolvido na linguagem PHP, onde a parte visual foi implementada utilizando HTML, CCS e *Javascript*.

Palavras-chave: Sistemas de Informação. Oficina. Desenvolvimento em PHP.

ABSTRACT

This project aims to create a web system for automotive workshops, which require a great interaction with its customers, and better data management on budgets, customers, employees, insurers and reports. The implementation of the system that was developed should make life easier for workshop staff and clients will always be informed of the whole process done on your vehicle. The proposed system was developed in PHP, where the visual part was implemented using HTML, CCS and Javascript.

Keywords: Information Systems. Workshop. PHP Developing.

LISTA DE FIGURAS

Figura 1 – Diagrama de Caso de Uso.....	23
Figura 2 - Classe Clientes	24
Figura 3 - Representação de associação entre classes	25
Figura 4 - Diagrama de Classes	25
Figura 5 - Modelo entidade relacionamento.....	27
Figura 6 - Página principal	28
Figura 7 - Formulário de Cadastro	29
Figura 8 - Tela de Login.....	29
Figura 9 - Painel do Administrador	30
Figura 10 - Painel do Cliente	30
Figura 11 - Exemplo de Orçamento Gerado.....	31
Figura 12 - Exemplo de Relatório Gerado	31

LISTA DE TABELAS

Tabela 1 - Requisitos Funcionais	21
Tabela 2 - Requisitos Não Funcionais	22
Tabela 3 - Notação de Relacionamento	26

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
CSS	<i>Cascading Style Sheets</i>
DW/DM	<i>Data Warehouse/Data mining</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
MYSQL	<i>My Structured Query Language</i>
PDF	<i>Portable Document Format</i>
PHP	<i>Personal Home Page</i>
POO	Programação Orientada a Objetos
RF	Requisitos Funcionais
RNF	Requisitos Não Funcionais
SAD	Sistema de Apoio à Decisão
SGBD	Sistema Gerenciador de Banco de Dados
SGE	Sistema de Gestão Empresarial
SQL	<i>Structured Query Language</i>
URL	<i>Uniform Resource Locator</i>
UML	<i>Unified Modeling Language</i>

SUMÁRIO

1 INTRODUÇÃO.....	13
1.1 Objetivo	13
1.2 Organização do Documento	14
2 REFERENCIAL TEÓRICO	15
2.1 Sistemas de Informação	15
2.2 Sistemas <i>web</i>	16
2.3 Linguagem <i>Personal Home Page</i>	16
2.4 Sistema Gerenciador de Banco de Dados	17
2.5 <i>Framework Bootstrap</i>	18
3 O DESENVOLVIMENTO DO SISTEMA.....	20
3.1 Análise e Especificação de Requisitos	20
3.1.1 Requisitos Funcionais	21
3.1.2 <i>Requisitos Não Funcionais</i>	22
3.2 Análise e <i>Design</i>	22
3.3.1 <i>Diagramas de Caso de Uso</i>	23
3.3.2 <i>Diagrama de Classe</i>	24
3.3.2.1 <i>Descrição Textual do Diagrama de Classes</i>	26
3.3 Modelagem de Dados	26
3.4 Principais telas do sistema	27
4 CONCLUSÕES E TRABALHOS FUTUROS	32
REFERÊNCIAS.....	33
APÊNDICES	34
APÊNDICE A – ATA DE REUNIÃO DE ENTREVISTA	35

1 INTRODUÇÃO

Atualmente, a *internet* é um dos meios de comunicação mais avançados e utilizados mundialmente. Devido ao alto grau tecnológico que a envolve e a sua capacidade de comunicação a longas distâncias, a *internet* consolidou-se como uma fonte completa de informação. Com o uso mundial da rede, a criação de aplicações *web* para a *internet* vem crescendo e tomando espaço entre as atividades desenvolvidas pelos profissionais de informática e em várias empresas, por abranger não só pequenos espaços físicos, mas podendo ser usado por todo o mundo.

Ao observar oficinas automotivas de Oeiras-PI, foi verificado que a maioria não usa sistemas computacionais para gerenciar seus serviços, elas utilizam somente anotações em cadernos para calcular orçamentos, cadastrar clientes, e verificar os débitos e veículos ainda em estado de conserto. No âmbito das oficinas automotivas já existem sistemas que gerenciam algumas coisas como cadastro de clientes, vendas e orçamentos, porém esses sistemas são bastante incompletos.

Muitos clientes ficam receosos ao sair de suas casas para levar seus automóveis a oficinas automotivas, mesmo que essas sejam bem qualificadas. O principal receio dos clientes é chegar nesta oficina e não ter como pagar o conserto, ou mesmo por não confiar em deixar seus automóveis com o dono da oficina. Outro caso que ocorre bastante é o atraso na data de entrega do veículo, onde o cliente aparece no dia determinado da entrega e seu serviço ainda não foi concluído, frustrando o cliente que perde tempo e dinheiro.

Pensando em uma forma de melhorar o atendimento a estes clientes, neste projeto foi criado um sistema *web* que interage com o cliente de algumas formas, como por exemplo: acompanhamento dos serviços que estão sendo feitos no veículo; solicitação de orçamento antes de ir até oficina, onde há o envio de fotos pelos clientes de algum veículo com a descrição de quais problemas deverão ser resolvidos; dentre outros tipos de interações.

Com o uso de um sistema *web* com essas funcionalidades o cliente já recebe um orçamento prévio de quanto irá custar o serviço. O cliente também tem acesso a cada uma das etapas que o seu automóvel já passou, quais peças foram trocadas e quais os valores de cada uma delas. Assim, o uso de um sistema desse tipo dá ao cliente maior confiança na oficina.

1.1 Objetivo

O objetivo desse trabalho foi desenvolver um sistema voltado para plataforma *web*, para uma oficina automotiva, onde o cliente possa interagir com a oficina, sem precisar estar presente na mesma.

1.2 Organização do Documento

As etapas de desenvolvimento do projeto estão bem descritas nos capítulos seguintes, sendo organizados em:

- Capítulo 2 - corresponde ao embasamento teórico necessário para desenvolvimento do projeto, contendo estudos e trabalhos realizados por outros autores e estudiosos das áreas que esse trabalho engloba.
- Capítulo 3 - apresenta os requisitos necessários para a realização do sistema *web* da oficina, oferecendo as informações necessárias para o desenvolvimento, bem como para a realização dos testes do sistema.
- Capítulo 4 - apresenta a conclusão do sistema e sugestões de trabalhos futuros.

2 REFERENCIAL TEÓRICO

No referencial teórico são apresentados alguns assuntos que foram utilizados para que o projeto pudesse ser realizado. A seguir serão apresentados os seguintes assuntos: sistemas de informação, sistemas *web*, linguagem PHP, SGBD e *framework bootstrap*.

2.1 Sistemas de Informação

De acordo com Laudon e Laudon (2004), a tecnologia da informação é composta pelo *hardware* e *software* que uma empresa usa para atingir seus objetivos e um sistema de informação consiste nos componentes que dão suporte à tomada de decisão e ao controle e auxiliam com análise, visualização e criação de produto. Para Stair e Reynolds (1998), um Sistema de Informação (SI) é um conjunto de componentes inter-relacionados que coletam, manipulam e disseminam dados e informações para proporcionar um mecanismo de realimentação para atingir um objetivo. E estes sistemas estão mudando constantemente a forma como as organizações conduzem seus negócios.

Sistemas de informação, portanto, é uma base para todo desenvolvimento de aplicações, pois ele fornece todos os requisitos que são necessários para ter sucesso nesse processo. Segundo Faoro, Rodrigues e Vargas (2010), os sistemas de informação são classificados em:

1. **Sistema de Apoio à Decisão (SAD)** - foco no suporte às decisões através de simulações, com a utilização modelos, que normalmente dão suporte às decisões de problemas semiestruturados e não-estruturados. Segundo Turban, McLean e Wetherbe (1996), as decisões para problemas semiestruturados, envolvem a combinação de soluções e procedimentos que não mudam e são baseadas nas experiências individuais. Já os não-estruturados, são processos vagos e problemas complexos, onde a intuição humana é frequentemente utilizada para tomar tais decisões.
2. **Sistema de Gestão Empresarial (SGE)** - foco na integração das informações em uma organização. Os sistemas SGE têm a finalidade de administrar partes importantes da empresa, tais como o planejamento do produto, compra de componentes, manutenção de estoques, interação com fornecedores, entre outros. Fornecendo assim, informações importantes para os negócios *online* e o intercâmbio automático;
3. **Data warehouse/Data mining (DW/DM)** - foco na exploração dos dados gerados pela empresa. Que segundo Harrison (1998, p. 155), *data mining*

“é a exploração e análise, por meio automático ou semiautomático, de grandes quantidades de dados para descobrir padrões e regras significativas.”. Este permite a empresa ter uma melhor compreensão dos clientes, um aumento na proporção das vendas, e apoio ao mesmo;

2.2 Sistemas *web*

Em computação, sistema *web* designa, de forma geral, sistemas de informática projetados para utilização através de um navegador, através da *internet* ou aplicativos desenvolvidos utilizando tecnologias *web*, *HyperText Markup Language* (HTML - Linguagem de Marcação de Hipertexto), *JavaScript* (linguagem de programação interpretada) e *Cascading Style Sheets* (CSS - linguagem de folhas de estilo). Pode ser executado a partir de um servidor *Hypertext Transfer Protocol* (HTTP - um protocolo de comunicação utilizado para sistemas de informação de hipermídia, distribuídos e colaborativos) ou localmente, no dispositivo do usuário (NATIONS, 2015).

Uma aplicação *web* também é definida em tudo que se é processado em algum servidor, por exemplo, quando se entra em um *e-commerce* a página que se acessa antes de vir até seu navegador é processada em um computador ligado a *internet* que retorna o processamento das regras de negócio nele contido. Por isso se chama aplicação e não simplesmente um conjunto de páginas *web* compostas por textos, imagens, animações e, eventualmente, sons. A função do servidor *web* é receber uma solicitação (requisição) e devolver (resposta) algo para o cliente, por algumas características não consegue processar tudo sozinho como: criações de páginas dinâmicas e o armazenamento de dados em um banco de dados. O *browser* permite ao usuário solicitar um recurso e quando o servidor responde a uma solicitação são encontrados recursos como: páginas HTML, figuras e documento *Portable Document Format* (PDF) que são exibidos depois para o usuário. Geralmente os servidores enviam instruções para o *browser* escritas em HTML. O HTML diz ao *browser* como apresentar conteúdo ao usuário (NATIONS, 2015).

Uma página *web* esta envolvida por uma serie de tecnologias, como protocolos, serviços e aplicativos integrados cada um com suas particularidades, essas podem ajudar o desenvolvedor caso saiba como fazer uso delas.

O protocolo HTTP usa um modelo de solicitações e respostas. Uma solicitação ocorre quando o usuário faz uma solicitação HTTP e o servidor *web* devolve uma resposta HTTP, sendo que o *browser* verifica como tratar esse conteúdo. Existem dois tipos de solicitações que são *get* e *post*, o *get* anexa dados do formulário no final da *url* e o *post* inclui dados do formulário no corpo da solicitação (NATIONS,2015).

2.3 Linguagem *Personal Home Page*

Segundo ACHOUR.*et.al* (2015), *Personal Home Page* (PHP) é uma linguagem de programação de ampla utilização, interpretada, que é especialmente interessante para desenvolvimento para a *web* e pode ser mesclada dentro do código HTML. A sintaxe da linguagem lembra C, Java e Perl, e é fácil de aprender. O objetivo principal da linguagem é permitir a desenvolvedores escreverem páginas que serão geradas dinamicamente rapidamente, mas você pode fazer muito mais do que isso com PHP.

A linguagem PHP foi concebida durante o outono de 1994 por Rasmus Lerdorf. As primeiras versões não foram disponibilizadas, tendo sido utilizadas em sua *home-page* apenas para que ele pudesse ter informações sobre as visitas que estavam sendo feitas. A primeira versão utilizada por outras pessoas foi disponibilizada em 1995, e ficou conhecida como “*Personal Home Page Tools*” (ferramentas para página pessoal). Era composta por um sistema bastante simples que interpretava algumas macros e alguns utilitários que rodavam “por trás” das *home-pages*: um livro de visitas, um contador e algumas outras coisas. Estima-se que em 1996 PHP/FI estava sendo usado por cerca de 15.000 sites pelo mundo, e em meados de 1997 esse número subiu para mais de 50.000. Nessa época houve uma mudança no desenvolvimento do PHP. Ele deixou de ser um projeto de Rasmus com contribuições de outras pessoas para ter uma equipe de desenvolvimento mais organizada. O interpretador foi reescrito por Zeev Suraski e Andi Gutmans, e esse novo interpretador foi a base para a versão 3 (BARRETO, 2015).

O PHP é uma linguagem de fácil utilização e aprendizado para pessoas que estão iniciando no desenvolvimento de *softwares* e ainda mais fáceis para profissionais que já trabalham com outro tipo de linguagem.

2.4 Sistema Gerenciador de Banco de Dados

Um banco de dados é um local onde se armazenam informações, com por exemplo, dados pessoais de clientes de uma determinada loja, números de contas bancárias, relatórios financeiros de uma empresa, entre outros. Segundo Magalhães (2008), para acessar um banco de dados é necessário a utilização de um *software* denominado de Sistema Gerenciador de Banco de Dados.

Um Sistema Gerenciador de Banco de Dados é o conjunto de programas de computador (*softwares*) responsáveis pelo gerenciamento de uma base de dados. O principal objetivo é retirar da aplicação cliente a responsabilidade de gerenciar o acesso, manipulação e organização dos dados. O SGBD disponibiliza uma *interface* para que os seus clientes possam incluir, alterar ou consultar dados. Em bancos de dados relacionais a *interface* é constituída pelas APIs ou *drivers* do SGBD, que executam comandos na linguagem *Structured Query Language* (SQL). Um sistema gerenciador de banco de dados é um sistema extremamente complexo, responsável pela persistência, organização e recuperação dos dados. Os SGBDs são *softwares* que permitem a definição de estruturas para armazenamento de informações e fornecimento de mecanismos para manipulá-las. As características de um SGBD são

integridade no sentido dos dados, a integridade do conteúdo dos dados e a validação dos dados, vinculado ao domínio da respectiva coluna (MAGALHÃES, 2008).

Existem vários bancos de dados, podemos citar alguns que são MYSQL, PostgreSQL e SQLite. Podemos observar que “O MySQL é talvez o banco que mais evoluiu nos últimos tempos, passando de um banco frágil e pouco seguro a um banco no mínimo respeitável. Eu mesmo não gostava do MySQL e tinha motivos para isso, minha preferência era o SQL Server, não que ela tenha mudada, mas o MySQL conseguiu provar que é um banco seguro e bastante indicado para aplicações web”^[1].

2.5 Framework Bootstrap

Framework é um conjunto de informações que servem como modelo para resolver um determinado problema. O *Bootstrap* é um *framework front-end* de código aberto (*opensource*). Em palavras simples, é um conjunto de ferramentas criadas para facilitar o desenvolvimento de *sites*, um conjunto de páginas *web* compostas por textos, imagens, animações e, eventualmente, sons, e sistemas *web*.

Compatível com HTML5 e CSS3, o *framework* possibilita a criação de *layouts* responsivos e o uso de *grids*, esses *grids* são um composto de até 12 colunas para organizar o conteúdo do *site* para se comportar de maneira diferente para cada resolução.

De acordo com Nascimento (2015), qualquer ferramenta possui suas vantagens e desvantagens. É importante conhecer e entender suas funcionalidades para saber os momentos certos de utilizá-lo.

Vantagens:

- Possui documentação detalhada e de fácil entendimento;
- É otimizado para o desenvolvimento de *layouts* responsivos;
- Possui componentes suficientes para o desenvolvimento de qualquer *site* ou sistema *web* com *interface* simples;
- Facilita a criação e edição de *layouts* por manter padrões;
- Funciona em todos os navegadores atuais (*Chrome, Safari, Firefox, IE, Opera*).

Desvantagens:

- Seu código terá de seguir os “padrões de desenvolvimento *Bootstrap*”;

¹ LUIS. Banco de dados Oracle, Mysql, Access e SQL Server. Disponível em: < <http://www.luis.blog.br/o-que-e-banco-de-dados.aspx> >. Acesso em 23 de fev de 2016

- Tema padrão e comum do *Bootstrap* (caso não faça ajustes visuais, seu projeto se parecerá com outros que também utilizam o *Bootstrap*).

3 O DESENVOLVIMENTO DO SISTEMA

Este capítulo apresenta a descrição do desenvolvimento do sistema proposto, contendo os requisitos de *software*, os estudos de caso de uso e o funcionamento do sistema, para que seja possível identificar os usuários e as suas funcionalidades.

O sistema para uma oficina automotiva foi desenvolvido a partir da necessidade de se criar um sistema que pudesse gerenciar as atividades da oficina como também aumentar a facilidade de atendimento ao cliente, por meios alternativos. Foram realizadas entrevistas em uma oficina automotiva com os funcionários e o dono para analisar as necessidades do cliente, como também para recolher requisitos que ajudassem a desenvolver o sistema.

Para que o programa fosse desenvolvido foi utilizada a linguagem PHP e o *framework Bootstrap*. Uma linguagem de fácil aprendizado e um *framework* bastante ágil para criação de interfaces com o usuário. Foram utilizados também conceitos de Programação Orientada a Objetos (POO) que facilita a vida do programador e também organiza melhor a estrutura do código.

O desenvolvimento do projeto passou por diversas etapas. Iniciando pela análise e especificação necessária para levantar as reais necessidades do cliente em relação às funcionalidades do sistema. Seguido pela Análise e *Design* utilizados para proporcionar uma visão geral da arquitetura do sistema, através da modelagem da solução, com o uso de diagramas UML. Após essas fases, foi realizada a modelagem de dados, que contém o diagrama de entidade relacionamento. Esse diagrama tem como finalidade descrever, de maneira conceitual, os dados utilizados no desenvolvimento do sistema. E por fim os formulários do sistema, mostrando algumas telas das principais funcionalidades desenvolvidas.

3.1 Análise e Especificação de Requisitos

Segundo Pressman (2006), requisito de *software* é uma condição ou uma capacidade com o qual o sistema deve estar de acordo, expressando as necessidades do cliente. Esses requisitos foram divididos em duas categorias, requisitos funcionais e não funcionais.

O processo de especificação de requisitos foi realizado por meio de entrevistas (apêndice A) junto ao dono da oficina automotiva. Durante essas entrevistas foram expostos os processos do referido setor, identificando assim suas dificuldades e necessidades.

3.1.1 Requisitos Funcionais

Os requisitos funcionais (RF) são as funcionalidades ou serviços que se espera que o sistema possua. Eles podem variar de acordo com o tipo de *software* que será desenvolvido, podem ser expressos de diversas maneiras e em diferentes níveis de detalhamento. Os requisitos funcionais levantados podem ser vistos na Tabela 1, que apresenta uma coluna com os requisitos funcionais, uma coluna com a descrição de cada requisito funciona e outra coluna com a dependência.

Tabela 1 - Requisitos Funcionais

Requisitos Funcionais	Descrição	Depende de
RF01 - O sistema deverá possuir controle de usuários.	O controle de usuários deve ser feito por meio de <i>login</i> e senha. O sistema diferencia os usuários em: Os usuários clientes que podem visualizar suas solicitações, seus dados cadastrados e seus veículos; O Usuário administrador que poderá fazer o gerenciamento das solicitações, criar e gerenciar orçamentos, alterar dados dos clientes e seus veículos, e pode também gerar relatórios financeiros.	
RF02 – Cadastro e gerenciamento de veículos.	Os veículos dos clientes devem ser cadastrados para que os dados desses veículos sejam analisados para geração de um orçamento.	RF01
RF03 - O sistema deve conter um local para solicitação de orçamento.	O cliente pode fazer solicitação de orçamento informando alguns dados que são: Veículo cadastrado, Descrição do problema, Tipo de pagamento	RF01, RF02
RF04-Aprovação e Reprovação de Solicitação.	O sistema deve se comportar da seguinte forma, caso a solicitação seja aprovada, deve aparecer no painel do cliente um botão para visualizar o orçamento prévio que pode ser modificado futuramente, para notificar o cliente sobre a decisão do administrador, caso a solicitação seja reprovada, deve ser excluída a solicitação no painel do cliente.	RF01, RF02,RF03
RF05-Cadastro e gerenciamento de orçamentos.	Os orçamentos cadastrados podem ser de dois tipos: Sem seguradora, onde o orçamento deve conter os dados do cliente, funcionários que trabalharam no veículo, as peças e os serviços e preços, Com seguradora, difere do anterior por não conter os valores das peças e dos serviços, acrescentando somente qual seguradora vai ser utilizada.	RF01, RF02,RF03, RF04,RF08
RF06-Cadastro e gerenciamento de funcionários.	Os funcionários devem ser cadastrados para que sejam inseridos nos orçamentos.	RF01
RF07-O sistema deverá possuir acompanhamento de solicitações e de veículo.	No acompanhamento de solicitações deve apresentar as solicitações realizadas e os orçamentos lançados pelo administrador. No acompanhamento de veículo deve apresentar o status do veículo até o presente momento.	RF01, RF02,RF03, RF04,RF08
RF08- Gerar relatórios	O sistema deve emitir relatórios financeiros de acordo com o ano ou mês ou status de pagamento.	RF01, RF02,RF03

Fonte: Autor (2015)

3.1.2 Requisitos Não Funcionais

Os requisitos não funcionais (RNF) são aqueles que estão relacionados às propriedades que o sistema possua, como confiabilidade, segurança, usabilidade, eficiência e outros atributos encontrados na qualidade de *software*. Os requisitos não funcionais levantados podem ser vistos na Tabela 2, a qual apresenta uma coluna com o identificador, uma coluna com a descrição e a outra com a categoria.

Tabela 2 - Requisitos Não Funcionais

Identificador	Descrição	Categoria
RNF01	O sistema deverá ter métricas de usabilidade bem definidas para facilitar ao usuário o uso do sistema, além de possuir design responsivo, caso a oficina queira utilizar dispositivos diferentes para acesso ao sistema.	Usabilidade
RNF02	O sistema deve atender de forma rápida qualquer solicitação de orçamento feita pelos usuários.	Eficiência em relação ao tempo
RNF03	O sistema deve possuir métricas de autenticação de forma a garantir a segurança dos dados dos clientes, funcionários da empresa.	Segurança de Acesso
RNF04	O sistema deve permanecer disponível a todo o momento, principalmente por se tratar de um sistema Web.	Disponibilidade
RNF05	O sistema deve ser desenvolvido usando boas práticas de programação, para garantir uma maior facilidade na manutenção.	Manutenibilidade
RNF06	Os dados armazenados deverão permanecer com sua integridade intacta e de forma segura.	Confiabilidade

Fonte: Autor (2015)

3.2 Análise e Design

Esta seção tem como objetivo analisar e detalhar a solução do sistema de acordo com os requisitos levantados e validados na seção 3.1. Para isso, deve-se ter uma visão geral da arquitetura do sistema e a modelagem da solução através de diagramas UML. Para o desenvolvimento do projeto foi utilizado o diagrama de caso de uso, que expressa o comportamento em alto nível que o sistema deve executar para cada ator; O diagrama de

classe que representa as classes do sistema, assim como os seus relacionamentos; e o modelo de entidade e relacionamento, cuja finalidade é descrever os dados que foram utilizados no sistema proposto.

3.3.1 Diagramas de Caso de Uso

Os diagramas de Caso de Uso são utilizados para serem entendidos por qualquer pessoa da área de desenvolvimento de sistemas ou não, proporcionando assim que tanto o cliente quanto o desenvolvedor possam entender o que será desenvolvido.

Segundo Cockburn (2001) para se escrever um caso de uso o autor tem que dominar três conceitos que se aplicam a toda sentença em um caso de uso e ao caso de uso como um todo. Os três conceitos são:

- Escopo: Qual é realmente o sistema sob discussão?
- Ator Primário: Quem tem o objetivo?
- Nível: Quão alto ou baixo é o nível do objetivo?

Os elementos que compõem um diagrama de caso de uso são os atores, o próprio caso de uso e o sistema. O ator interage com sistema, requisitando uma ação e recebendo uma reação do mesmo. Os atores são identificados por quem inicializa o fluxo de atividades do sistema, fornecendo dados e recebendo informações.

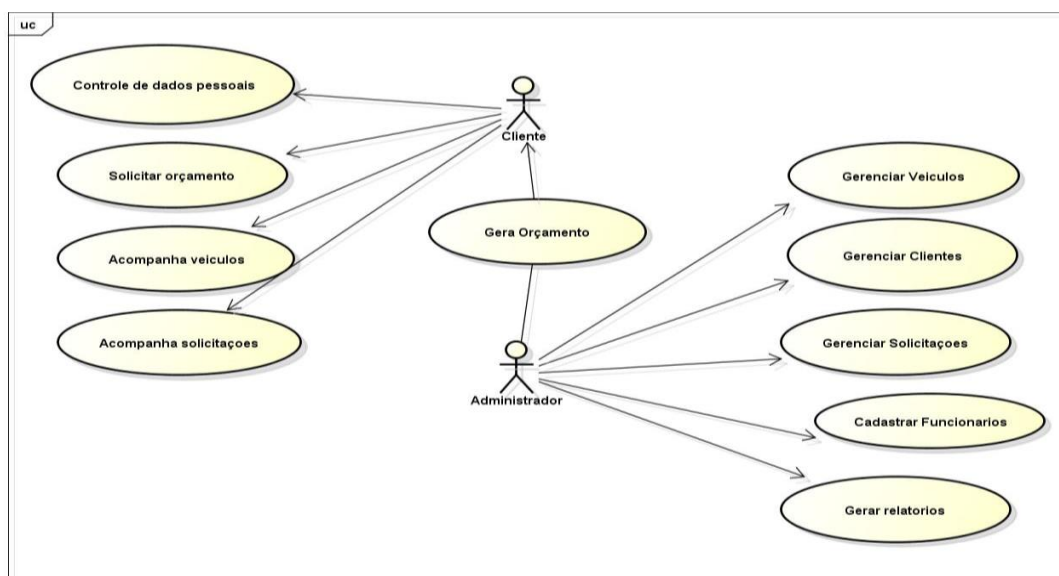


Figura 1 – Diagrama de Caso de Uso

A Figura 1 mostra o caso de uso, apresentando os seus atores: o Cliente e o Administrador, e suas ações e restrições no sistema. O ator Cliente pode realizar várias ações

como, controlar seus dados pessoais, solicitar orçamento, acompanhar veículos e acompanhar solicitações. Já o ator Administrador pode gerar o orçamento para o ator Cliente, como também gerenciar veículos desses clientes, solicitações, dados do cliente, relatórios financeiros e cadastra funcionários.

3.3.2 Diagrama de Classe

Segundo Guedes (2009), o objetivo do diagrama de classes é mostrar os relacionamentos existentes entre as classes que são abstraídas no projeto, e como esses relacionamentos colaboram para a execução de um processo específico.

No diagrama de Classe, uma classe é representada por um retângulo, que possui duas linhas que separam três partes. A primeira parte representa o nome da classe, a segunda os atributos da classe e por fim os métodos como podem ser visto na Figura 2, que representa a classe Clientes.

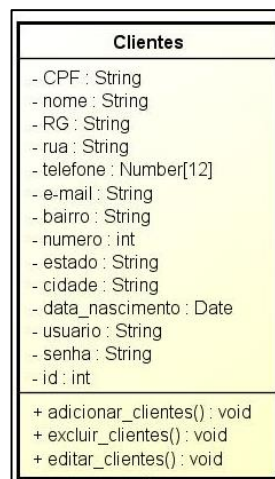


Figura 2 - Classe Clientes

Além disso, o diagrama de classe possui associação entre classes, que permite especificar quais classes estão se relacionando com outras classes. As associações são representadas por uma reta utilizada para unir as duas classes, como podem ser visto na Figura 3. Os valores nas extremidades da reta indicam a multiplicidade das associações, seguindo a notação, que pode ser observado na Tabela 3.

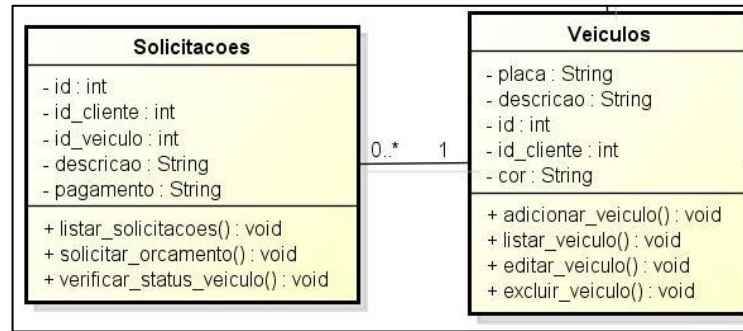


Figura 3 - Representação de associação entre classes

Um veículo pode estar associado a várias solicitações, ou a nenhuma. Uma solicitação necessariamente tem que estar associada a um veículo e a não mais do que uma, conforme mostra a figura 3.

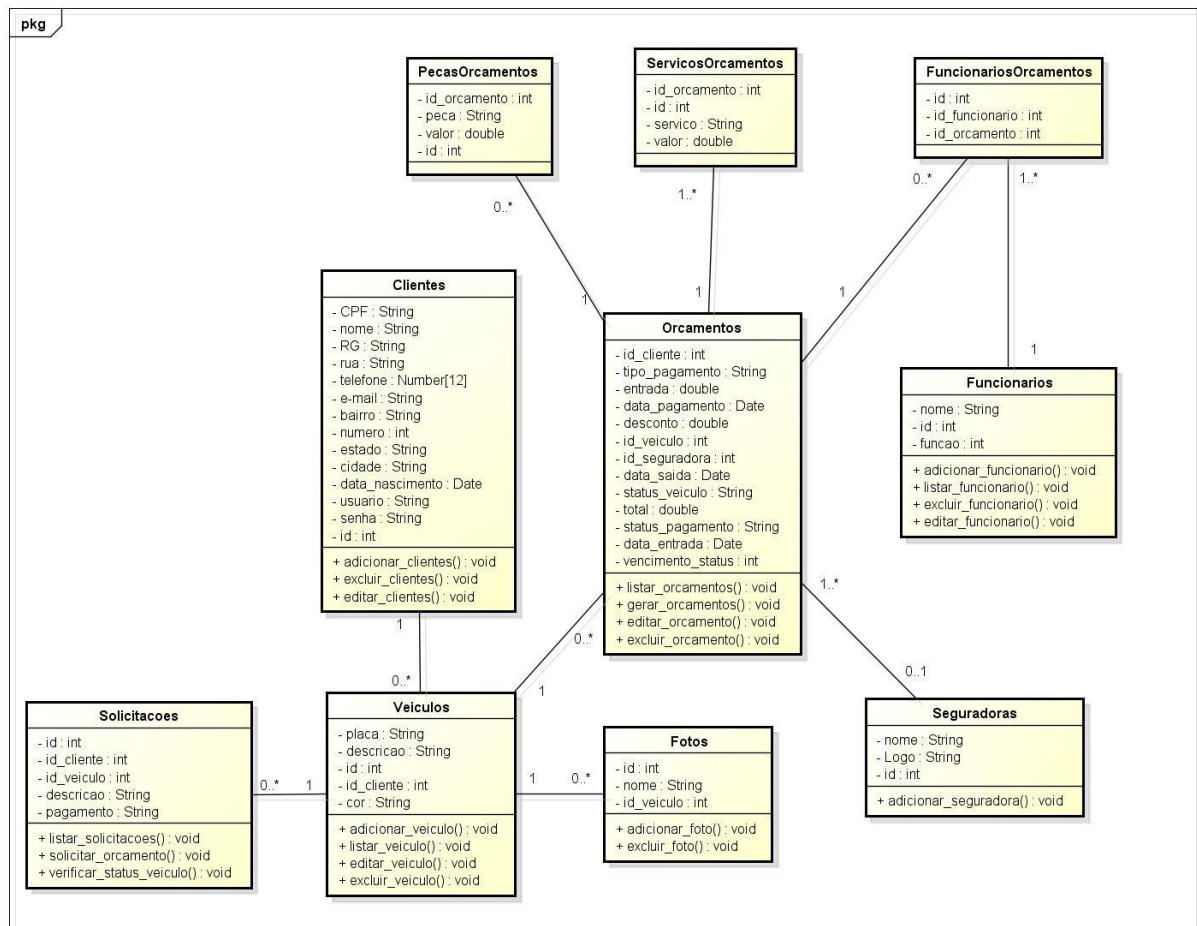


Figura 4 - Diagrama de Classes

No diagrama de classes foram utilizados os tipos de associações apresentadas anteriormente, assim como a representação de uma classe por meio de um retângulo. Ele foi dividido em três partes, sendo apenas a primeira parte referente ao nome da classe, e tem que ser preenchido obrigatoriamente, como pode ser visto na Figura 4.

Tabela 3 - Notação de Relacionamento

Tipo	Significado
0..1	Zero ou um registro.
0..*	Não existe limite para o número de registros.
1	Exatamente um registro.

Fonte: Autor (2015)

3.3.2.1 Descrição Textual do Diagrama de Classes

Para um melhor entendimento do diagrama de classes criado, é apresentada a descrição das classes de forma textual.

- a) A classe Clientes armazena as informações dos usuários do sistema.
- b) A classe Veículos armazena as informações dos veículos que o usuário tenha no sistema.
- c) Fotos: classe que armazena as fotos do veículo que serão necessários para análise da oficina.
- d) Solicitações: é a classe que armazena todas as solicitações de orçamentos feitas pelo cliente.
- e) A classe Orçamentos serve para o administrador cadastrar todas as informações de um orçamento pedido pelo cliente para seu veículo.
- f) Seguradoras: classe que armazena todas as seguradoras vinculadas a oficina.
- g) Funcionários: classe utilizada pelo administrador para gerenciar as informações dos funcionários da oficina.
- h) PeçasOrçamentos, ServiçosOrçamentos e FuncionariosOrçamentos: são classes que armazenam as informações para controlar as peças, serviços e funcionários que iram realizar o concerto do automóvel. Essas classes estão relacionadas com a classe Orçamentos.

3.3 Modelagem de Dados

No projeto foi utilizado o banco de dados MYSQL e para criar o banco foi utilizado o diagrama entidade relacionamento pode ser visto na figura 5. Segundo Martin (1990) o diagrama de entidade relacionamento é um modelo abstrato cuja finalidade é descrever, de maneira conceitual, os dados a serem utilizados em sistemas de informação.

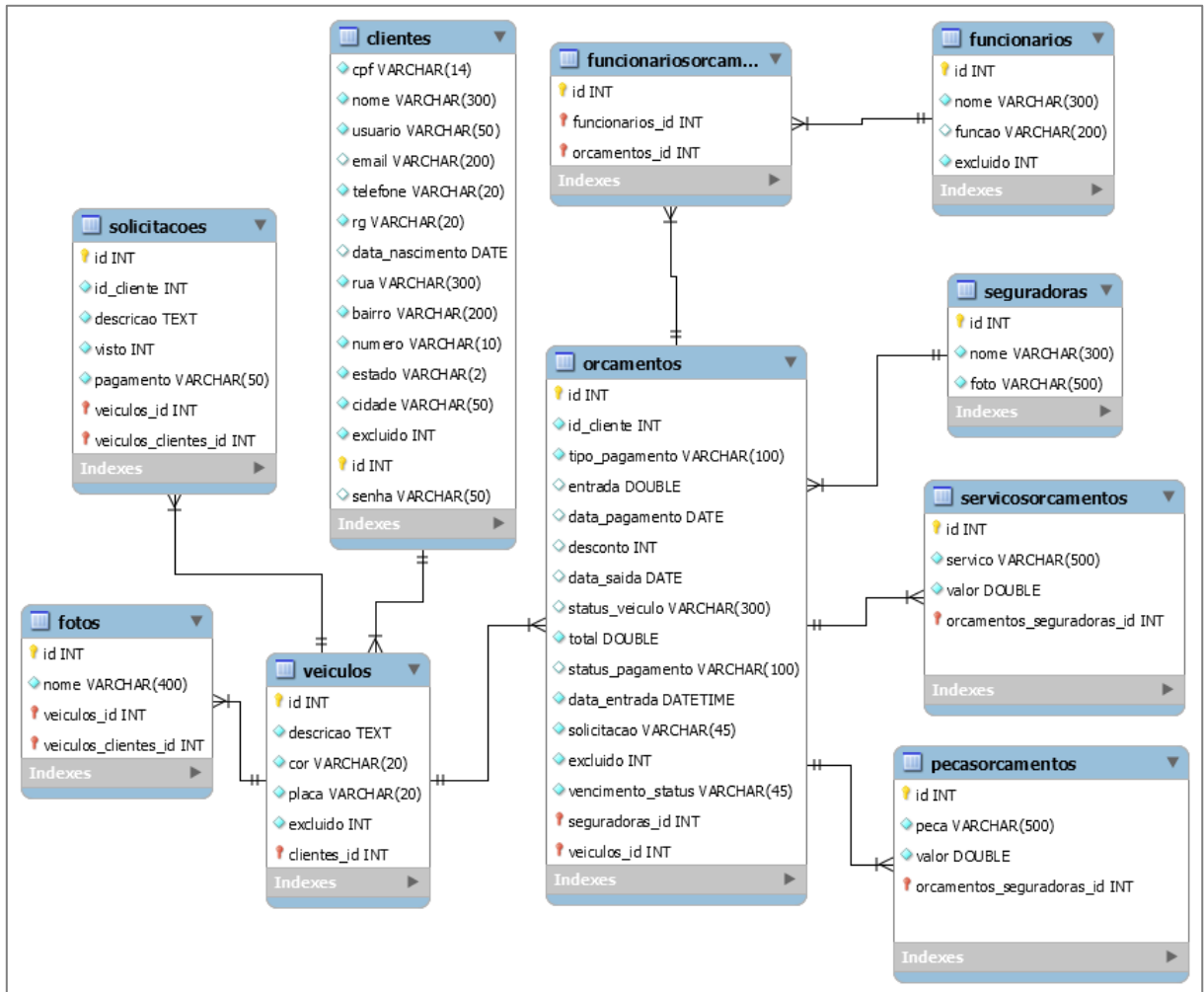


Figura 5 - Modelo entidade relacionamento

3.4 Principais telas do sistema

Nessa seção serão apresentadas as telas do sistema, mostrando as funcionalidades citadas em seções anteriores. O sistema consiste em três partes, a página inicial (Figura 6) que apresenta um *site* modelo para qualquer oficina.

AUTO CENTER FERRARA Inicio Sobre a empresa Serviços Solicitar orçamento Painel Cliente

WORKS SEAMLESSLY WELL ON ALL DEVICES

LOREM IPSUM DOLOR SIT AMET. CONSECTETUR ADIPISCING ELIT. FUSCE VITAE EUISMOD LACUS. MAECENAS IN TEMPOR. LECTUS. NAM MATTIS. ODIO UT DAPIBUS ORNARE, LIBERO.

Bem vindo(s) a **FERRARA** - Cadastre-se e solicite um orçamento para seu automóvel [Cadastrar-se](#)

Interaja com a oficina de casa
De sua própria casa você pode solicitar um orçamento prévio e acompanhar o que está sendo feito em seu veículo sem sair de casa.

Previsão de entrega
Após feito o orçamento definitivo, há uma previsão da data de término do(s) serviço(s) feitos no carro, permitindo o veículo ser liberado antes da data ou após esta data prevista então acompanhe seu carro pelo painel do cliente.

Ganhe tempo e dinheiro
Escolhendo nossa oficina você ganha tempo não precisando ir na oficina verificar como está o estado do carro, e como consequência você não perde seu dinheiro pagando algum veículo para se deslocar a oficina antes do veículo estar pronto.

» PARCEIROS

Clique em algum dos parceiros e veja seus produtos e formas de contato.

» FOTOS DA OFICINA

» SEGURADORAS

© 2016 Ferrara Auto Center. Desenvolvido por Wabney Vieira

Figura 6 - Página principal

Para acessar o painel administrativo o usuário deve inserir o usuário e senha criados previamente pelo desenvolvedor. Já para acessar o painel do cliente devem-se inserir os dados cadastrados no formulário de cadastro (Figura 7).

CADASTRAR-SE

Observação:
Primeiramente faça seu cadastro, depois entre no painel do cliente e solicite seu orçamento.
Preencha todos os campos obrigatórios(*)

Nome*

Usuario*

Senha*

Telefone*

Cpf*

Rg*

Rua*

Numero*

Bairro*

Cidade* Estado

Data de Nascimento

Email

Cadastrar

© 2015 Ferrara Auto Center. Desenvolvido por Wesley Vieira

Figura 7 - Formulário de Cadastro

Para ter acesso ao painel administrativo ou do cliente, o usuário deve realizar o *login* (Figura 8) digitando o seu usuário e senha cadastrados respondendo ao RF01 e ao RNF03, das tabelas 1 e 2, respectivamente.

Fazer login

Usuario

Senha

Entrar

Ferrara Auto Center 2015-Todos os direitos 2015. Desenvolvido por: Wesley Vieira.

Figura 8 - Tela de Login

No painel do administrador (Figura 9) existem varias funções que são:

- Listar clientes, veículos, funcionários, orçamentos e solicitações;
- Adicionar clientes, veículos e funcionários;
- Gerar orçamentos e relatórios.

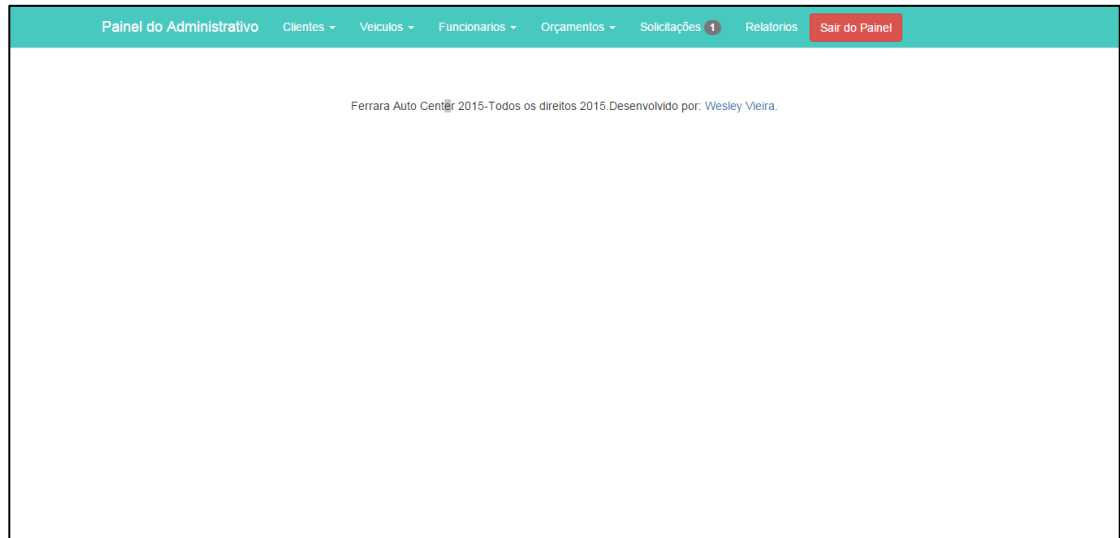


Figura 7 - Painel do Administrador

No painel do cliente (Figura 10) existem outras funções que são:

- Solicitar orçamento;
- Acompanhar solicitações e status do veículo;
- Editar dados pessoais e exibí-los;
- Listar veículos do cliente;
- Adicionar novos veículos.

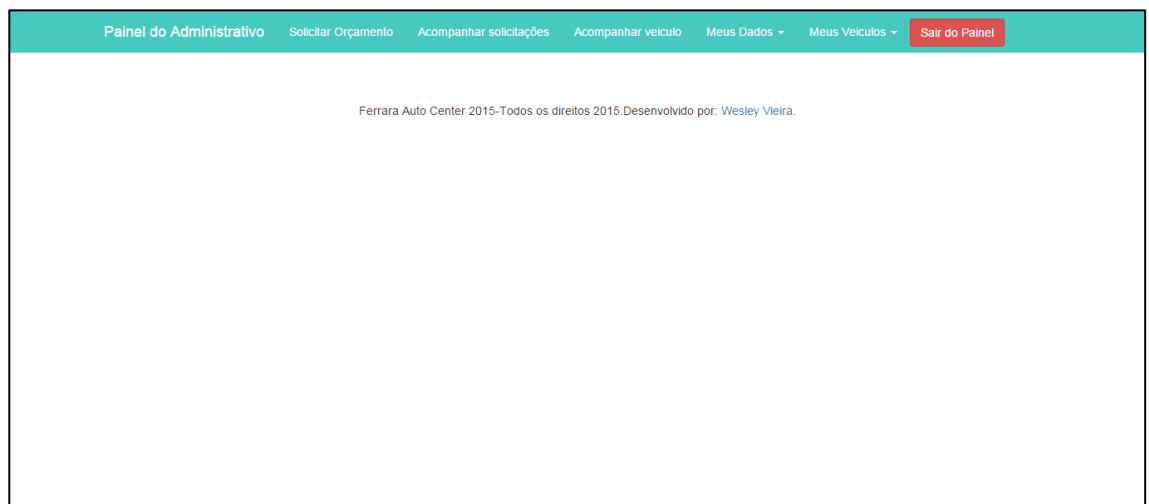


Figura 8 - Painel do Cliente

A figura 11 mostra um exemplo de orçamento que pode ser gerado pelo sistema, após uma solicitação de orçamento ou na própria loja.

07/07/2015			
			
ORÇAMENTO			
CLIENTE	Vanderley		
VEICULO	Onix	PLACA	oei-1234
COR	Branco	FONE	(89)9444-9804
TROCA DE PEÇAS			
1	porta		R\$ 300
SERVIÇOS			
1	desamassar		R\$ 100
2	pintura		R\$ 150
VALOR DA MÃO DE OBRA			R\$ 250
VALOR DAS PEÇAS			R\$ 300
VALOR DE ENTRADA			R\$ 0
VALOR EM DEBITO			R\$ 250
VALOR TOTAL DO ORÇAMENTO			R\$ 250

Figura 9 - Exemplo de Orçamento Gerado

Tanto no painel do cliente como no do administrador, podem ser adicionadas fotos aos veículos, para fazer análise do estado que o mesmo se encontra. A cada novo processo de conserto do veículo o status do veículo é modificado e pode-se verificar qual funcionário está encarregado no momento do seu veículo. A figura 12 apresenta um exemplo de relatório financeiro de um determinado período de tempo que apresenta todos os carros que tem orçamento aprovado, valor em débito e valor total do orçamento de cada veículo de um determinado cliente, valor total de todos os débitos e valor total de todos os orçamentos dos clientes.


07/07/2015					
					
RELATORIO					
id	CLIENTE	PLACA	VEICULO	VALOR EM DEBITO	VALOR ORÇADO
1	Wesley Vieira2	oei-3939	Ford ka	44	44
2	Wesley Vieira2	kkkk-22131	Moto cg 125 20141	1865.2	1865.2
3	Wesley Vieira2	kkkk-22131	Moto cg 125 20141	309.12	309.12
4	Wesley Vieira2	kkkk-22131	Moto cg 125 20141	0	0
5	Wesley Vieira2	kkkk-22131	Moto cg 125 20141	2.156	2.156
6	Albertino De Moura Fé	oei-9495	Moto cg 125 2014	2	2
7	Vanderley	oei-1234	Onix	800.5	800.5
8	Stanley de sousa saucenzelos	stw-2030	Celta 2008	0	0
9	Vanderley	oei-1234	Onix	2.94	2.94
10	Vanderley	oei-1234	Onix	0	0
11	Vanderley	oei-1234	Onix	0	0
12	Fabiana	oei-3452	gol	475	475
13	Fabiana	ssdf	dsfddf	0	0
14	Vanderley	oei-1234	Onix	22	23
15	Vanderley	oei-1234	Onix	4	4
16	Vanderley	oei-1234	Onix	250	250
VALOR TOTAL DE DEBITO				3776.916	
VALOR TOTAL ORÇADO NO MÊS				3777.916	

Figura 10 - Exemplo de Relatório financeiro Gerado

4 CONCLUSÕES E TRABALHOS FUTUROS

Em toda área a tecnologia está presente, não só com sistemas lógicos, mas também aparatos físicos que aumentam o rendimento e facilitam o trabalho das pessoas. Pensando nisso um sistema que ajude a gerenciar uma oficina seria um grande passo para aumento de produtividade, como também traria maior visibilidade, por ser um sistema *web*.

O sistema da oficina é um grande diferencial, pois nem toda oficina mecânica o possui, dando assim maior confiabilidade ao cliente da loja, por ter um acompanhamento do que está sendo feito no seu veículo, sem precisar estar presente todos os dias na loja. Outro ponto importante é a criação de relatórios que pode ajudar na parte financeira, para tomada de decisões na hora de comprar novos equipamentos, fazer cobranças, e verificar os lucros gerados pela empresa.

O sistema desenvolvido é um sistema *web*, foi testado na oficina Ferrara em Oeiras-PI e obteve-se bons resultados de acordo com o dono da loja e seus funcionários, como o aumento de clientes, rapidez no processo de conserto e de tomada de decisão na compra de peças e monitoramento mas ágil dos funcionários. Como trabalho futuro pode ser desenvolvido um aplicativo para celular *Android*, para verificar informações rápidas do seu(s) veículo(s) presente(s) na loja, facilitando ainda mais a vida do cliente por ser um dispositivo de fácil acesso e maior praticidade. Além de poder ser implementada novas funções como a criação de informações rápidas no início do painel administrativo do cliente e do administrador e um sistema de estoque e venda de produtos *online* para oficinas que não só consertam os veículos como também vendem produtos individualmente.

REFERÊNCIAS

- ACHOUR, M. *et. al.* **Manual do PHP**. Disponível em: <http://php.net/manual/pt_BR/index.php >. Acesso em 14 fev. 2015.
- BARRETO, M.V. S. **Curso de linguagem PHP**. Disponível em: <<http://www.etelg.com.br/paginaete/downloads/informatica/php.pdf> >. Acesso em 20 fev. 2015.
- COCKBURN, A., **Writing Effective Use Cases**. Addison-Wesley, 2001. Disponível em: <<http://alistair.cockburn.us/get/2465> >. Acesso em 16 fev. 2015.
- FAORO, M. A.; RODRIGUES, F. R; VARGAS, R. F.; **Um estudo sobre os principais tipos de sistemas de informação**. UCS VACARIA – RS. Dezembro de 2010.
- GUEDES, G. T. A. **UML 2: uma abordagem prática**. São Paulo: Novatec Editora, 2009.
- HARRISON, T. H. **Intranet Data Warehouse**. São Paulo: Berkeley, 1998. Disponível em: <<http://www.lume.ufrgs.br/bitstream/handle/10183/19461/000308562.pdf?..> >. Acesso em 19 fev. 2015.
- LAUDON, K. C.; LAUDON, J. P. **Sistemas de Informações Gerenciais: administrando a empresa digital**. 5. ed. Prentice Hall. 2004.
- MAGALHÃES, C. E. G.; **Fundamentos de Banco de Dados**. 2008. Disponível em: <<http://www.ebah.com.br/content/ABAAAjFQAH/gti-103-fundamentos-banco-dados-prof-kadu>>. Acesso em 14 fev. 2015.
- MARTIN, J. **Information Engineering (volume 1)** – Editora Prentice Hall, 1990.
- NASCIMENTO, T. **O fantástico mundo do desenvolvimento web**. Disponível em: <<http://thiagonasc.com/desenvolvimento-web/desenvolvendo-com-bootstrap-3-um-framework-front-end-que-vale-a-pena> >. Acesso em 28 jan. 2016.
- NATIONS, D. **What is a Web Application?** Disponível em: <http://webtrends.about.com/od/webapplications/a/web_application.htm >. Acesso em 11 jan. 2015.
- PRESSMAN, R.S. **Engenharia de Software**. 6ª Ed, McGraw-Hill, 2006.
- STAIR, R.; REYNOLDS, G., **Princípios de sistemas de informação**. LTC, 1998.
- TURBAN, E.; McLEAN, E.; WETHERBE, J. **Information technology for management**. New York: John Wiley & Sons, 1996. 801p. Disponível em: <http://www.avm.edu.br/docpdf/monografias_publicadas/k211396.pdf >. Acesso em 11 jan. 2015.

APÊNDICES

APÊNDICE A – ATA DE REUNIÃO DE ENTREVISTA

Data: 12/12/14	Entrevistados: Gerente; Dono; Recepcionista; Administrador;	Entrevistador: Wesley	Duração: 02:00 horas
Assunto: O assunto principal abordado na reunião foi a necessidade que a empresa tem de interagir com os seus clientes de uma forma virtual, em vez de somente fisicamente. Onde os clientes não precisariam sair de suas casas para ir nem a oficina para saber quanto custaria um determinado orçamento.			
Objetivos: · Verificar a necessidade da empresa, que poderia ser suprida por um programa de computador. · Fazer o levantamento dos requisitos · Fazer o estudo da viabilidade do sistema			
Perguntas Formuladas: · O que a oficina necessita? · Quais problemas deseja solucionar com o programa que será implementado? · Quais características este programa deve apresentar? · Ele deve ser <i>Web</i> ou <i>Desktop</i> ? Caso seja <i>Desktop</i> em qual plataforma ele deve funcionar? · Quais os dados que serão armazenados no banco de dados do programa? · De quem devem ser armazenados os dados? · O que vai ser solucionado com este programa? · Quem pode usar? · Terá nível de acesso? · O programa precisa de segurança? · A empresa tem máquinas para gerenciar o programa a ser criado? · Quais funcionários serão treinados para usar o programa? · O sistema necessita de backup?			
Pontos Discutidos: · Interação com o cliente; · Nem todos os funcionários terão acessos a todas as funções do programa; · A empresa necessita da compra de computadores e tablets para poder acessar o sistema que será criado; · A segurança do programa deve ser bem implementada, pois apresentam dados pessoais dos funcionários e clientes, como também informações financeiras da empresa; · O sistema poderá ser acessado por qualquer tipo de sistema operacional. · O sistema deve possuir uma funcionalidade para que possa ser criado um backup dos dados armazenados.			



**TERMO DE AUTORIZAÇÃO PARA PUBLICAÇÃO DIGITAL NA BIBLIOTECA
“JOSÉ ALBANO DE MACEDO”**

Identificação do Tipo de Documento

- () Tese
 () Dissertação
 (X) Monografia
 () Artigo

Eu, Wesley Vieira de Moura Fé,
 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
Desenvolvimento de um sistema web para uma
oficina automotiva
 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 01 de Março de 20 16.

Wesley Vieira de Moura Fé
 Assinatura

Wesley Vieira de Moura Fé
 Assinatura