

SISTEMA WEB DE GESTÃO DE FROTAS PARA AUTOMAÇÃO DE PROCESSOS E AUXÍLIO AO GESTOR

Thales Marques Engel

Faculdades Integradas de Taquara – Faccat – Taquara – RS – Brasil
thalesengel@gmail.com

Marcelo Cunha de Azambuja
Professor Orientador

Faculdades Integradas de Taquara – Faccat – Taquara – RS – Brasil
azambuja@faccat.br

Resumo

Este artigo apresenta o processo e os resultados do desenvolvimento de uma aplicação web de gestão de frotas de veículos, a qual tem por finalidade, além de automatizar processos, auxiliar no controle e administração dos veículos de uma organização. Por consequência dessa administração facilitada, o trabalho visa também diminuir gastos, poupar tempo, prolongar a vida útil dos veículos e diminuir riscos. Para prover esses benefícios, o sistema desenvolvido possui funcionalidades pertinentes à gestão de veículos automotores, tais como controle de manutenção, controle de abastecimentos, apresentação de relatórios, entre outros.

Palavras-chave: frotas, gestão, auxílio.

SYSTEM WEB OF FLEET MANAGEMENT FOR PROCESS AUTOMATION AND AID TO MANAGER

Abstract

This article presents the process and results of the development of a web application for vehicle fleet management, which aims, as well as automate processes, assist in the control and management of vehicles in one organization. Consequently this easier administration, the work also aims to cut costs, save time, extend the life of vehicles and reduce risks. To provide these benefits, the developed system has features relevant to the management of vehicles, such as maintenance control, supply control, reporting, among others.

Keywords: *fleets, management, assistance.*

1. Introdução

Com o acesso cada dia mais facilitado aos recursos de informática, e a busca constante por eficiência, gestores buscam automatizar processos em suas organizações. Procuram também melhorar o controle em todos os setores da empresa, de modo que consigam, tanto eles como os demais colaboradores, estar bem informados para a tomada de decisões. Em relação à administração de frotas veiculares de uma empresa, o cenário não é diferente.

Segundo Coelho (2011) grande parte das decisões estratégicas da gestão de uma frota tem como pontos de análise a problemática do controle, a redução dos custos operacionais dos veículos, os sistemas de manutenção, bem como o planejamento e formação da frota de veículos de uma organização.

Nos dias atuais, fazer a gestão da frota de forma amadora, esperar que o carro quebre para que então seja consertado, são métodos que se tornaram obsoletos. Com a pressão constante para reduzir custos, os gestores buscam alternativas que provêm maior eficiência e praticidade. A alternativa para esta situação é a adoção de um sistema de gestão de frotas que garanta um melhor controle das informações, tornando a empresa mais competitiva e aumentando a produtividade (AVELAR, 2012).

Tendo conhecimento desses fatos, foi desenvolvido um software web de controle de frotas, capaz de monitorar e controlar os períodos de manutenção, troca de peças, os gastos com combustível e as despesas geradas pelos veículo, com a finalidade de otimizar a gestão da frota e, assim, diminuir os gastos e os riscos de acidente com os veículos.

2 Referencial teórico

2.1 Gestão de frotas de veículos

Segundo Clemente (2008), gestão de frotas de veículos é a atividade de administrar os veículos automotivos de uma determinada organização e, tipicamente, consiste na utilização de softwares ou ferramentas tecnológicas, que possibilitam as empresas obter melhor controle e monitoramento das informações referentes aos veículos. Além disso, complementando este conceito, Valente (2011) afirma que essa tarefa tem abrangência bem ampla e envolve diferentes serviços, tais como, dimensionamento, especificação de equipamentos, roteirização, manutenção e renovação de veículos, entre outros.

De fato a gestão da frotas é de grande importância para as empresas, afinal, diminuem os riscos associados com o investimento dos automóveis, melhoram a eficiência das operações, aumentam a produtividade, reduzem custos e o cumprem com mais facilidade a legislação governamental imposta pelas entidades reguladoras do mercado.

2.2 Gestão de Combustível

De fato, o combustível é um gasto frequente e geralmente alto nas frotas de automóveis, portanto é de suma importância para os gestores ter um controle dos abastecimentos de sua frota.

A principal vantagem em realizar o controle de combustível é poder fazer um planejamento financeiro. Com o gerenciamento de quanto cada veículo utilizou e quanto foi gasto com combustível, o gestor consegue ter médias de utilização e médias de despesas com abastecimento. Dessa forma, é possível elaborar um planejamento seguro e eficaz, baseando-se em informações reais. Assim, a empresa sabe exatamente quanto irá gastar com combustível em sua frota e não é surpreendida com gastos que não estavam no orçamento (SOFIT SOFTWARE SA, 2014).

Fazendo esse controle, o gestor de frota consegue saber onde está gastando mais e identificar formas de economizar nos gastos com o combustível - seja por meio da manutenção dos veículos, ou mesmo revisão das rotas empregadas (SOFIT SOFTWARE SA, 2014).

Na gestão de combustível e abastecimentos de uma frota, de fato é necessário ter o controle de todos os veículos. Para isso, o ideal, segundo Gonçalo (2014), é ter uma numeração ou um código que identifique cada um deles para que se saiba qual é o tipo de veículo, a marca, o ano etc. Dessa forma, é possível avaliar o volume de combustível utilizado por cada tipo de veículo da frota e saber a média de abastecimento das diferentes marcas. Tendo conhecimento destes valores pode-se identificar quando o número de abastecimento de um certo tipo de veículo divergir do padrão, então tomar ações para evitar desperdícios.

A quantidade de combustível abastecido pelos veículos a cada vez precisa ser controlada juntamente com a distância percorrida. Dessa forma, é possível controlar quantos quilômetros cada um está fazendo com aquela determinada quantidade de combustível e calcular as médias de cada tipo de veículo. Isso é importante para que seja identificado se

algum deles está saindo do padrão empregado, como por exemplo, mais combustível que o geralmente utilizado para percorrer certa distância. Isso pode estar acontecendo por diversos motivos, entre eles, problemas mecânicos. Com o controle correto dos abastecimentos, consegue-se identificar com antecedência qualquer tipo de anormalidade e tomar ações para resolver os problemas, dando mais segurança e economia às operações de sua frota (GONÇALO, 2014).

2.3 Considerações gerais sobre manutenção

A Associação Brasileira de Normas Técnicas (ABNT (1971)) define manutenção como sendo o conjunto de todas as ações necessárias para que um item seja conservado ou restaurado de modo a poder permanecer de acordo com uma condição especificada.

Segundo Azeredo (2007), por mais cuidado que se tenha ao utilizar um veículo, este sempre sofrerá desgaste ao longo do tempo. Por isso, os manuais de veículos indicam o período para a verificação de diversos itens. Há casos em que alguns destes itens devem ser revisados semanalmente, e outros até diariamente, como para veículos de carga e passageiros.

Existem métodos diferentes de manutenções, que são divididos em três grupos básicos: manutenção preventiva, corretiva e preditiva. A aplicação de cada método varia de acordo com algumas características da própria manutenção, como por exemplo: a situação, o tipo da peça/equipamento em questão, o nível de relevância de cada peça, entre outras. A definição de cada um é apresentada a seguir.

Manutenção Preventiva, de acordo com a Norma NBR 5462 (1994), é a manutenção efetuada em intervalos pré-determinados, ou de acordo com critérios prescritos, destinada a reduzir a probabilidade de falha ou a degradação do funcionamento de um item. Em geral, a manutenção preventiva é considerada como uma verificação periódica programada. Esta verificação não se trata de consertar algo, mas constatar se os itens estão em condições satisfatórias para o bom funcionamento do veículo. Normalmente, entre uma verificação e outra, as boas condições devem se manter. Porém, se toda vez é necessário, por exemplo, completar o óleo do motor, o nível de água do radiador, etc, o administrador poderá ver um indício de algum defeito. Neste caso, deve ser realizada uma inspeção mais apurada para verificar a causa da perda (SEITO e GILL, 2008).

Assim sendo, manutenção preventiva é aquela realizada num período pré-determinado ou sob condições específicas, com o intuito de prevenir a ocorrência de falhas ou quebra. A exemplo de manutenções preventivas de um carro, pode-se citar a verificação da correia-dentada, o estado dos pneus, suspensão, verificar se não há vazamentos, entre outros.

Em relação à manutenção corretiva, segundo a Norma NBR 5462 (1994), é aquela realizada depois da ocorrência de uma pane ou defeito constatado, destinada a recolocar um item em condições normais. O objetivo da manutenção corretiva é justamente concertar os defeitos ou falhas causados pelo desgaste ou o mau uso. Esta manutenção é considerada a mais primitiva entre as três, pois ela somente será realizada após a máquina/equipamento quebrar ou apresentar falha. A substituição de um pneu furado, troca de um farol queimado e concerto da caixa de câmbio são exemplos de manutenções corretivas de automóveis.

A terceira forma de manutenção é conhecida como manutenção preditiva. A norma NBR 5462 (1994) a define como sendo a aquela que permite garantir uma qualidade de serviço desejada, com base na aplicação sistemática de técnicas de análise, utilizando-se de meios de supervisão centralizados ou de amostragem, para reduzir ao mínimo a manutenção preventiva e diminuir a manutenção corretiva.

A manutenção preditiva é semelhante à manutenção preventiva, com a diferença de que esta é feita com o objetivo de manter o funcionamento e qualidade das peças. Portanto, manutenção predita é aquela em que se atua sobre determinados itens/peças que apresentam desgaste e que possuem grande probabilidade de quebra, como, por exemplo, a troca de óleo do motor após o veículo ter atingido o limite de quilometragem especificado, a troca de um pneu gasto, geometria e balanceamento.

2.3.1 Manutenção planejada

É essencial para uma empresa a existência de um plano de manutenção a curto prazo, o qual deve especificar o período e as condições adequadas para realização das manutenções de cada categoria de veículo. Ou seja, este plano de manutenção deve indicar a correta política de manutenção da empresa para operações de reparo, substituição e recondicionamento dos sistemas e componentes, visando, deste modo, a melhoria do processo de controle de manutenção dos veículos.

Campos e Belhot (1994) destacam que o plano de manutenção deve estar em conformidade com a estratégia de controles de recursos adotada pela empresa, uma vez que o custo de manutenção influencia sensivelmente as estruturas organizacionais e administrativas vigentes.

Um exemplo de plano de manutenção indicando a frequência (tempo ou quilometragem realizada) que devem ser realizados cada serviço descrito pode ser visto no Quadro 1.

QUADRO 1. Exemplo de plano de manutenção

Veículo:	Carro 001	
Manutenção	Período	
	Km	Meses
Troca do fluido de frio	10000	-
Troca de óleo	10000	6
Limpeza do sistema de arrefecimento	30000	-
Substituição do filtro de ar	-	12
Troca do filtro de combustível	15000	-
Alinhamento, balanceamento e rodizio de pneus	10000	-
Limpeza interna e externa	-	1

Fonte: Autoria própria (2015)

2.4 Trabalhos Relacionados

Softwares voltados para o controle de frotas de veículos não são novidade, o SISMA (Sistema de Gerenciamento e Manutenção de Frota), por exemplo, segundo a Assite (2006), proprietária do software, atua neste segmento de mercado desde 1991. Porém, com o advento dos dispositivos de computação móveis, surgiu uma demanda por aplicações que permitam que o usuário as acesse e as utilize remotamente. O SISMA é um software destinado a uso empresarial e algumas das principais funções deste sistema são: emitir ordens de serviço, manter um histórico das manutenções básicas, controlar o consumo de combustível, controlar a rotatividade dos pneus, etc. O SISMA, entretanto, é um software *desktop*, ou seja, precisa ser instalado no computador de mesa do cliente para ser utilizado, impossibilitando o uso em dispositivos móveis. Além disso, diferente de softwares hospedados na nuvem onde o risco de

falhas de hardware é praticamente nulo, devido à alta infra-estrutura de armazenamento, softwares *desktop* apresentam ricos de segurança quanto a integridade dos dados, por conta de falhas no hardware. como por exemplo, uma queda de energia, que pode desligar a maquina e causar a perda de dados.

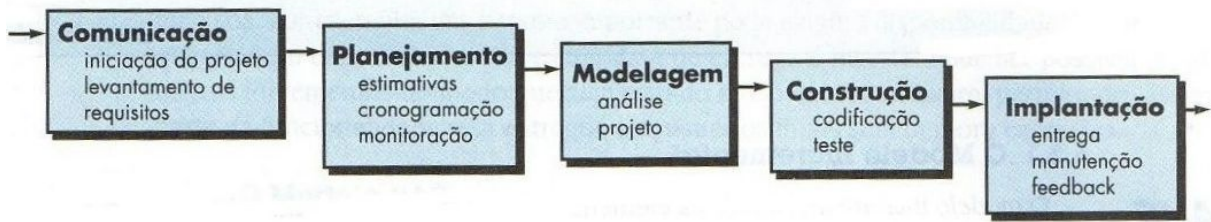
Outro sistema relacionado ao deste trabalho é o CTN Car (CTN Cardoso, 2013), que possui uma versão para uso corporativo e outra para uso pessoal. De acordo com o desenvolvedor, a versão do CTN Car para uso corporativo se trata de um sistema online que possibilita o controle sobre os veículos e condutores. Conta com módulos de gerenciamento de abastecimento, despesas e serviços. Possui ainda um módulo para dispositivos móveis, onde é possível preencher as informações de abastecimentos, despesas e serviços. Já a versão para uso pessoal é definido como um aplicativo que possibilita ao usuário ter controle sobre os abastecimento, manutenções, trocas de óleos e outros serviços realizados no veículo. As informações sobre o veículo são apresentadas através de relatórios e gráficos, tendo as informações de média de Km/litro, gastos mensais, médias por dia, combustíveis utilizados, entre outras.

Entre as características que diferenciam o sistema apresentado neste trabalho para os demais softwares de gerenciamento de frotas semelhantes é que além de ser um aplicação online sem custo algum para os usuários, foi desenvolvido com o intuito de ter a usabilidade simples, apresentar os resultados de forma clara e objetiva no auxílio a tomada de decisão.

3. Metodologia

O modelo de processo empregado para desenvolvimento do software foi o modelo sequencial, também conhecido como “Ciclo de vida clássico” ou “*Waterfall model*” (modelo cascata). Este modelo segue o processo em sequência de etapas, ou seja, uma etapa só se inicia quando a anterior já estiver finalizada. O modelo segue o fluxo com as seguintes etapas: (i) análise: onde ocorre as especificações dos requisitos, (ii) planejamento e modelagem: onde é criada uma solução e definido como ela será feita, (iii) construção: etapa onde o sistema é efetivamente codificado e testado, (iv) implantação: etapa em que o projeto é entregue e o cliente a fornece um retorno (PRESSMAN, 2009). Este processo pode ser visto na figura 1.

FIGURA 1 – Modelo Cascata (Waterfall)



Fonte: adaptado de Pressman R., Software Engineering: A Practitioner's Approach, 7/e (2009)

3.1 Análise

A primeira etapa do projeto iniciou com a identificação das principais necessidades e dificuldades existentes nas tarefas de controle e gestão de frotas veiculares de uma empresa. Para isso, foi realizada uma consulta com pessoas que trabalham, tanto diretamente como indiretamente, com a frota de veículos em suas empresas. Nessa consulta foram realizadas perguntas as quais tiveram o intuito de constatar possíveis problemas e dificuldades na gestão de uma frota de veículos, a fim de identificar possíveis funcionalidades que seriam importantes existirem no software.

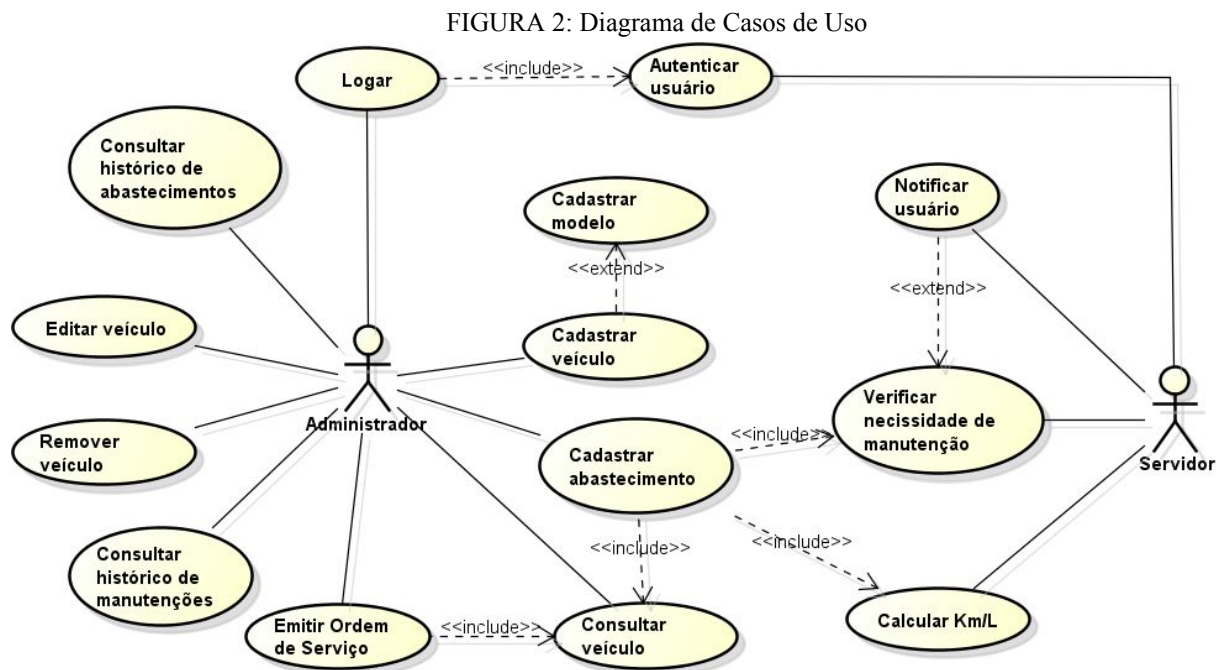
Com base nas informações obtidas, realizou-se o levantamento dos requisitos e funcionalidades que o sistema deveria possuir para que atendesse as reais necessidades desta área, e com isso, melhorar o controle e gestão de frotas veiculares.

Com os requisitos definidos, foi construído o escopo do projeto, que nada mais é do que uma descrição sucinta das funcionalidades que o software possuirá.

3.2 Modelagem

Assim que as atividades de análise do projeto foram concluídas, deu-se início a etapa de modelagem da arquitetura do software, onde foi definida toda a estrutura do sistema e como ele se comportaria para que ele atendesse aos requisitos e as funcionalidades propostas. Para isso, foram empregados os diagramas propostos pela UML (Unified Modeling Language – Linguagem de Modelagem Unificada). A UML é definida por Fowler (2003) como uma família de notações gráficas que ajudam na descrição e no projeto de sistemas de software, particularmente daqueles que utilizam linguagem Orientada a Objetos.

Como base no padrão da UML, foi feito o diagrama de casos de uso com a finalidade de definir o cenário, os atores e seus respectivos casos de uso do sistema. Atores são os usuários que interagem com o sistema, ou melhor, são os tipos de usuários, que podem ser humanos ou outro sistema computacional. Caso de uso é uma tarefa ou funcionalidade realizada pelo ator. A seguir pode ser visto o diagrama de casos de uso do software, na Figura 2.



Fonte: Autoria própria (2015)

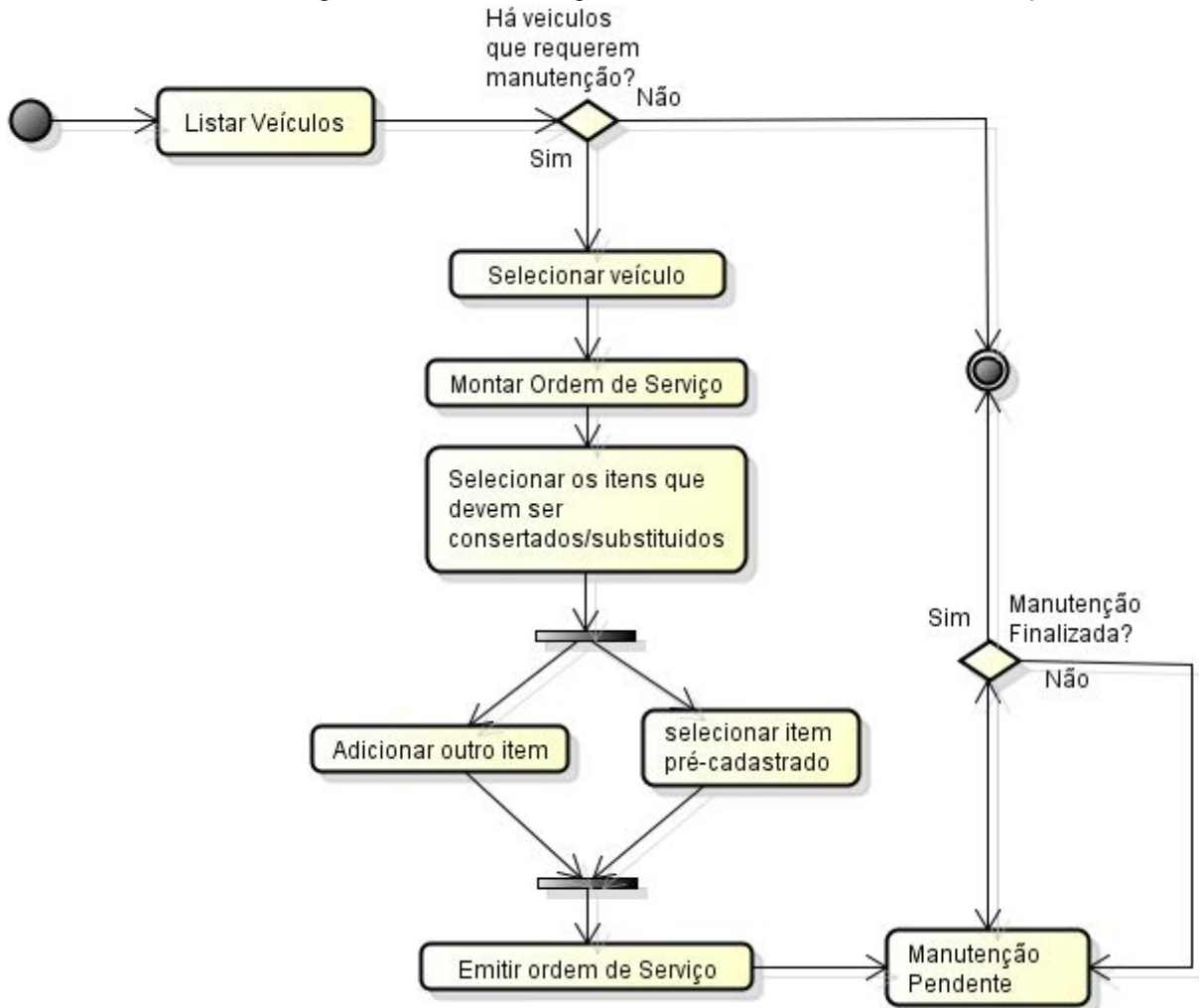
Os atores presentes no sistema são o administrador e o servidor (o próprio sistema). O administrador é o usuário responsável por popular e manter os dados no sistema, é ele quem deverá realizar os cadastros, consultas e exclusões. Além disso, ele fará a emissão das ordens de serviço para manutenção dos veículos.

Já o servidor, através das suas rotinas de software, irá atuar verificando as circunstâncias de cada veículo cadastrado, e caso um dado veículo apresente as condições que indiquem a necessidade de manutenção, o software do servidor então emite um aviso ao usuário, informando que um determinado veículo necessita manutenção.

Assim que definidos os casos de uso do sistema, foi elaborado o Diagrama de Atividades. Na definição de Fowler (2003), diagramas de atividades é uma técnica para

descrever a lógica processual, processo de negócio, e o fluxo de trabalho. Logo, o diagrama de atividades foi fundamental para modelagem do fluxo de controle de uma atividade para outra, possibilitando a identificação dos aspectos dinâmicos do sistema. A seguir, a Figura 3 apresenta o diagrama de atividades de um dos principais processos do sistema, a emissão de uma ordem de serviço.

FIGURA 3. Diagrama de Atividades do processo de emissão de um ordem de serviço



Fonte: Autoria própria (2015)

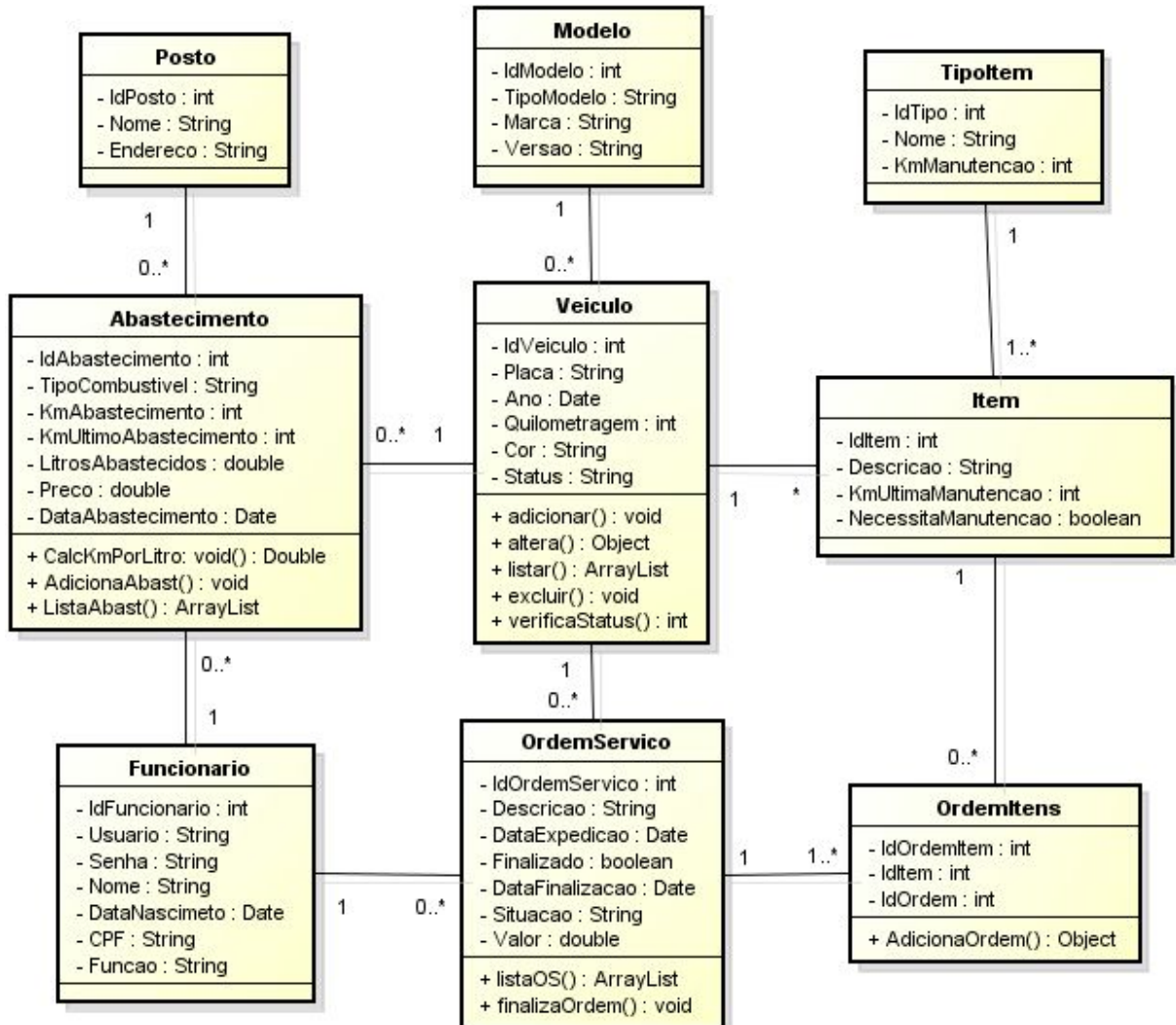
Como pode ser verificado no diagrama da Figura 3, após a listagem dos dados dos veículos cadastrados, e verificado que algum deles precisam passar por manutenção, o usuário deve então selecionar o registro de tal veículo, dando início ao processo de montagem da ordem de serviço (OS) para manutenção. O usuário poderá então escolher os itens/peças que precisam ser revisadas. Quando isso for feito, o usuário irá emitir a OS, e esta irá para a

fila de ordens pendentes. Após a realização de manutenção, a ordem é finalizada e as informações são armazenados em um histórico.

Para finalizar a etapa de modelagem do sistema, elaborou-se o Diagrama de Classes (DC). Segundo a concepção de Fowler (2003), o DC descreve os tipos de objetos no sistema e os vários tipos de relacionamentos estáticos que existem entre eles. Mostram também as propriedades e operações de uma Classe e as restrições que se aplicam à maneira como os Objetos estão conectados.

Como a Orientação a Objeto (OO) é a base do C#, linguagem de programação empregada no desenvolvimento do software (os detalhes podem ser vistos mais adiante, no item 3.3 deste artigo), o Diagrama de Classe foi essencial para o projeto, pois nele foram definidos todos os vários tipos de Objetos que compõem o sistema, assim como seus atributos. Além disso, se identificou os relacionamentos entre tais objetos, servindo, deste modo, de base para estruturar o banco de dados do sistema. O diagrama de classes pode ser visto na Figura 4.

FIGURA 4. Diagrama de Classes



Fonte: Autoria própria (2015)

O Diagrama de Classes foi utilizado não só para identificar os objetos e seus atributos, mas também como base para a modelagem da estrutura do banco de dados. Cada classe foi considerada uma tabela do banco de dados relacional, assim o relacionamento entre elas pôde ser interpretado como os relacionamentos entre as tabelas e suas respectivas cardinalidades.

Para criar os diagramas de Casos de Uso, Atividades e o de Classes, foi utilizada a ferramenta Astah Community (CHANGE VISION, 2015), que é própria para modelagem UML.

3.3 Desenvolvimento

Por se tratar de uma aplicação web, o sistema desenvolvido está dividido em duas partes: a *server-side* (lado do servidor) e o *client-side* (lado do cliente), também conhecidas com *back-end* e *front-end*, respectivamente. O *back-end* é responsável por realizar a conexão com o banco de dados, tratar os dados trazidos do banco ou recebidos do cliente e aplicar as mudanças e/ou ações sobre estes dados. O *front-end* é responsável por consumir os dados vindos do servidor e transformá-los em informações para o usuário. Além disso, coletar entrada do usuário e processá-la para adequá-la a uma especificação em que o servidor possa utilizar. Ou seja, o *front-end* é uma espécie de interface entre o usuário e o *back-end*.

As principais tecnologias utilizadas para desenvolver o software foram C# (MICROSOFT, 2015), Javascript, AngularJS (GOOGLE, 2015) e SQL Server (MICROSOFT, 2015), as quais estão descritas a seguir.

3.3.1 Visual C# e .NET Framework

A linguagem de programação base do software é a Visual C# (pronuncia-se "C sharp"), que se trata de um linguagem Orientada a Objetos criada e mantida pela Microsoft (2015). Para escrever o código do programa em C# a plataforma de desenvolvimento utilizada foi o Visual Studio 2013 (MICROSOFT, 2013). Essa ferramenta fornece diversos recursos para se desenvolver programas, entre eles está o ASP.NET MVC, um Framework específico para desenvolvimento web, o qual foi empregado no desenvolvimento do software deste trabalho.

O C# foi criado pela Microsoft em 2001 e consiste em uma linguagem de programação criada para o desenvolvimento de uma variedade de aplicações executadas através do *framework* .NET . C# é uma linguagem que, conforme pode ser verificado ao longo deste desenvolvimento, permite o desenvolvimento rápido de aplicações, conforme também atesta o seu fabricante (MICROSOFT, 2015) .

A IDE¹ Visual Studio suporta o Visual C# com um editor de código completo, compilador, modelos de projetos, designers, assistentes de código, depurador e outras ferramentas que auxiliam o trabalho do programador. A biblioteca de classes do *framework* .NET provê acesso a vários serviços do sistema operacional e outras classes úteis que

¹ IDE: Integrated Development Environment, que significa Ambiente de Desenvolvimento Integrado.

aceleram o ciclo de desenvolvimento (MICROSOFT, 2014). Estas funcionalidades puderam ser comprovadas ao longo deste trabalho.

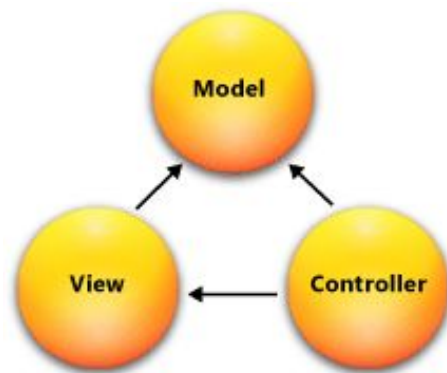
SQL Server foi o banco de dados usado para armazenar todos os registos do software. E o Entity Framework é usado para realizar a conexão com o banco e possibilitar a manipulação dos dados através do Visual Studio.

3.3.2 Arquitetura MVC

O padrão MVC² é uma arquitetura que possibilita separar em camadas as funcionalidades de um sistema de software. Com essa divisão, além de melhorar a organização, torna-se mais fácil o desenvolvimento de programas em grande escala (BASTOS, 2011). Essa arquitetura é considerada atualmente a forma indicada para desenvolvimento de softwares, e por esse motivo foi adotada neste projeto.

A Figura 5 demonstra as associações entre as camadas do padrão MVC:

FIGURA 5. Padrão MVC



Fonte: MSDN (2015)

O *model* é utilizado para manipular informações de forma mais detalhada, sendo recomendado que, sempre que possível, se utilize dos modelos para realizar consultas, cálculos e todas as regras de negócio de um site ou sistema. O acesso a todas as informações, sejam essas vindas do banco de dados, ou de arquivos XML (*Extensible Markup Language*) por exemplo, é realizado pela camada de modelo (BASTOS, 2011).

² MVC: *Model, View e Controller*

A *view* é responsável por tudo que o usuário visualiza, toda a interface e informação, não importando sua fonte de origem (BASTOS, 2011). ou seja, a *view*.

A *controller*, como o nome já sugere, é responsável por controlar todo o fluxo de informação que passa pelo sistema. Nela ocorrem as tarefas de lógica do sistema, bem como é onde são definidas quais informações devem ser geradas, quais regras devem ser acionadas e para onde as informações devem ir. Em resumo, é a controladora que executa uma regra de negócio (*model*) e repassa a informação para a visualização (*view*) (BASTOS, 2011).

3.3.3 AngularJS e Javascript

Para a criação de código no lado do cliente foi utilizado o framework Angular.js, o qual possibilita a manipulação dos dados, pela aplicação cliente, ou seja, os servidor apenas provê os dados, as ações são executadas na máquina do usuário através do interpretador Javascript (navegador Web normalmente).

O Angular é um framework baseado em Javascript mantido pelo Google. Ele pode ser estruturado nos modelos MVC, MVP (Model-Viewpresenter) e MVVM (Model View ViewModel). Pelo fato de se enquadrar em vários tipos de modelos, ele é considerado pelos seus desenvolvedores como um framework MVW (Model-View-Whatever) (SILVA e VICENTE, 2014).

Segundo a própria documentação, o Angular possibilita o uso do HTML³ como linguagem de modelo e permite estendê-lo para expressar os componentes da sua aplicação (OLIVEIRA, 2013). O Angular, através de Javascript, torna o HTML dinâmico (SILVA e VICENTE, 2014).

3.4 Testes

Após a etapa de desenvolvimento, iniciou-se a fase de testes. Pressman (2009) afirma que o teste de software é uma função de controle de qualidade que tem um principal objetivo: encontrar erros. Ainda, destaca que testes devem ser devidamente planejados e conduzidos de forma eficiente para que ele tenha a maior probabilidade de alcançar o seu objetivo principal. Dado isto, o método utilizado para realização dos testes foi o da “caixa preta”, o qual,

³HTML: abreviação para a expressão inglesa HyperText Markup Language, que significa Linguagem de Marcação de Hipertexto.

segundo Costa (2011), simula a interação do usuário com o sistema, baseado em cenários de testes extraídos dos casos de uso ou da descrição das funcionalidades.

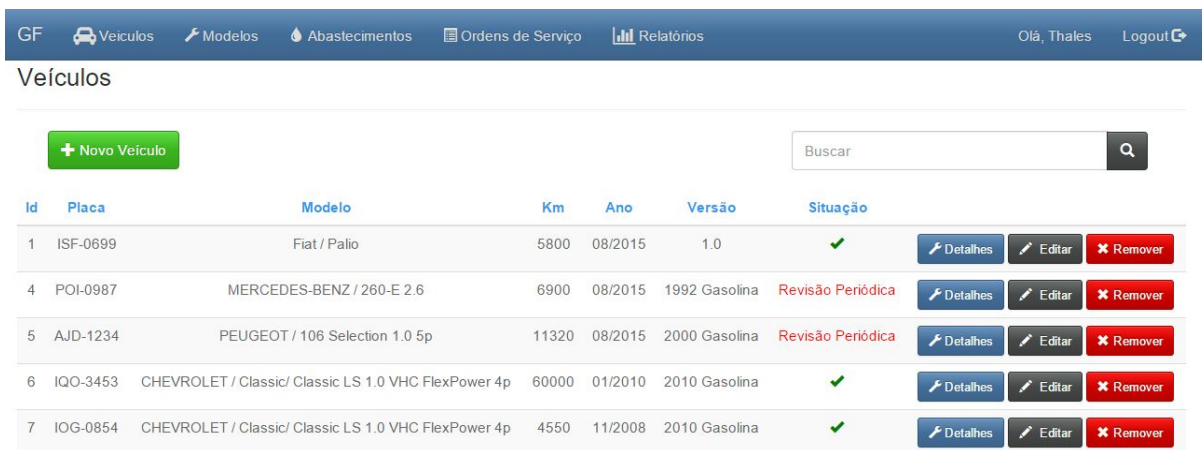
4. Resultados

Como resultados da pesquisa e desenvolvimento, se obteve o software on-line de gestão de frotas de veículos. Esse software oferece recursos e funcionalidades para manter o controle das diversas tarefas existentes na gestão de frotas veiculares, tais como manter as informações dos veículos, controlar abastecimentos, manutenções programadas, manutenções corretivas, entre outras. O funcionamento do sistema está descrito a seguir.

Primeiramente, por ser um sistema armazenado na nuvem, o usuário obrigatoriamente deve possuir registro de usuário e senha válidos para acessar o sistema. Esta é uma medida básica de segurança para impedir que pessoas não autorizadas utilizem a aplicação.

Estando o usuário autenticado corretamente, ele será redirecionado para a tela onde estarão listadas as informações de todos os veículos cadastrados. Esta tela pode ser observada na Figura 6.

FIGURA 6. Tela de listagem dos veículos cadastrados



Id	Placa	Modelo	Km	Ano	Versão	Situação	
1	ISF-0699	Fiat / Palio	5800	08/2015	1.0	✓	Detalhes Editar Remover
4	POI-0987	MERCEDES-BENZ / 260-E 2.6	6900	08/2015	1992 Gasolina	Revisão Periódica	Detalhes Editar Remover
5	AJD-1234	PEUGEOT / 106 Selection 1.0 5p	11320	08/2015	2000 Gasolina	Revisão Periódica	Detalhes Editar Remover
6	IQO-3453	CHEVROLET / Classic/ Classic LS 1.0 VHC FlexPower 4p	60000	01/2010	2010 Gasolina	✓	Detalhes Editar Remover
7	IQG-0854	CHEVROLET / Classic/ Classic LS 1.0 VHC FlexPower 4p	4550	11/2008	2010 Gasolina	✓	Detalhes Editar Remover

Fonte: Autoria própria (2015)

A Figura 6 exibe a tela com uma simulação de um cenário em que há cinco veículos cadastrados, e destes, dois estão sendo indicados que devem ser realizada as suas a revisões periódicas. A partir desta tela é possível escolher as ações que serão feitas dentro do sistema e para cada veículo, estas ações são: (i) incluir um novo registro de veículo, (ii) editar, (iii)

remover, (vi) ver mais detalhes. Também, para facilitar a busca e organização do usuário, todas as tabelas do sistema possuem um campo de busca e permitem ordenação por coluna.

Ao acessar os detalhes de algum dos veículos, o sistema permite ao usuário iniciar o processo de emissão de uma OS para manutenção. Neste processo o usuário irá selecionar os itens de manutenção (correia, cárter, motor, faróis, velas, etc) que ela deseja, podendo incluir à cada um uma descrição do serviço a ser realizado. Veja o exemplo na Figura 7:

FIGURA 7: Emissão de Ordem de Serviço

	nº	Item	Descrição do serviço
✘	1	Controle de ignição/injeção	Revisão periódica
✘	2	Velas	Revisão periódica
✘	3	Embreagem	Regulagem
✘	4	Estado da Bateria	Trocar bateria
✘	5	Correia Dentada	Trocar correia

Fonte: Autoria própria (2015)

Caso seja o momento de realizar a manutenção periódica do veículo e o usuário decida emitir uma OS para fazê-la, o próprio sistema irá sugerir os itens de manutenção, pré-cadastrados no sistema, que devem ser revisados. Essa sugestão é baseada no período e quilometragem de revisão de cada item.

Após a OS ser gerada, ela é encaminhada à fila de OS pendentes, aguardando ser finalizada. Quando houver alguma OS pendente uma sinalização aparecerá no menu superior em “Ordens de Serviço”, indicando existência e sua quantidade, como pode ser visto mais abaixo, na Figura 8, que exibe uma simulação de uma OS gerada. Elas contêm as informações pertinentes ao serviço a ser realizado, tais como, os dados do veículo, possíveis peças que

apresentam defeito ou que devem ser revisadas, descrição do serviço, data de emissão da OS, entre outras. Assim que finalizada ela é mantida em um histórico com todas as manutenções realizadas.

FIGURA 8: Ordem de Serviço

002 - POI-0987			24/10/2015
Veículo			
Código do veículo	Placa	Ano do Veículo	
4	POI-0987	08/2015	
Quilometragem	Marca	Modelo	
6900	MERCEDES-BENZ	260-E 2.6	
Versão			
1992 Gasolina			
Serviço			
Código do serviço	Data expedição da Ordem de Serviço		
002	24/10/2015		
n°	Item	Descrição do serviço	
1	Controle de ignição/injeção	Revisão periódica	
2	Velas	Revisão periódica	
3	Faróis	Troca do farol dianteiro direito queimado	
4	Freio de Estacionamento	Regulagem do freio	
5	Filtro de Ar	Limpeza	

Fonte: A autoria própria (2015)

Outra funcionalidade do sistema é a gestão de abastecimentos, que permite o usuário ter o controle dos abastecimentos realizadas com cada veículo, fornecendo, para isso, informações como: (i) data do abastecimento, (ii) quantidade de litros, (iii) quilometragem do veículo no momento do abastecimento, (iv) preço total por abastecimento, (v) média de quilometragem por litro (km/L) feita e (vi) posto onde foi feito o abastecimento. Estas informações são obtidas através de dados populados pelo usuário, por meio de cadastro. Após o cadastro, os abastecimentos são então mantidos em um histórico para fins de consulta.

5. Conclusão

Neste artigo foi apresentado os conceitos, a metodologia aplicada e os resultados do projeto e desenvolvimento de um sistema web de gestão de frotas de veículos para automação e auxílio ao gestor. Além dos conceitos e fundamentos buscados e obtidos durante o processo de criação do sistema, estão as ferramentas empregadas em sua implementação. As tecnologias e produtos oferecidas pela Microsoft, como as ferramentas presentes no Visual Studio, que, apesar de exigirem um bom domínio e conhecimento para sua manipulação, foram fundamentais na criação do software em questão, permitindo produzir bastante em pouco tempo e garantido a segurança, integridade e desempenho ao usuário final. Igualmente importante foi o *framework* AngularJS da Google, que, através do recursos e funcionalidade providos, possibilitou de forma simples e rápida a criação, tanto de uma interface limpa e dinâmica, como à logica de programação por trás dela.

De certa forma o sistema não está totalmente completo, pois existe a possibilidade e a necessidade de complementar e adicionar novas funcionalidade, assim como todos os tipos de softwares. Sempre é possível incrementar e agregar melhorias à um software. Apesar disso, o em relação ao escopo proposto, os resultados mostram os objetivos deste trabalho foram alcançados.

Referências

- ABNT NBR 5462:1994. **Confiabilidade e manutenibilidade**, 1994.
- ABNT - TB - 19 Grupo 05: **Termos fundamentais de eletricidade**, 1971.
- ASSISTE ENGENHARIA DE SOFTWARES TÉCNICOS. **SISMA**. Disponível em: <<http://www.assiste.com.br/sisma.html>> Acesso em: 13 de Setembro de 2015.
- AVELAR, G. **Manutenção da frota de dificuldade de gerenciamento**. 2012. Disponível em:<<http://engeman.com.br/pt-br/artigos-tecnicos/manutencao-de-frotadificuldade-no-gerenciamento/print/>>. Acesso em: 08 de Abril de 2015.
- AZEREDO, L. E. **Manutenção periódica e preventiva do veículo**. 2007. Disponível em:<<http://www.vias-seguras.com/>>. Acesso em: 08 de Abril de 2015.
- BASTOS, Daniel Flores. **O que é Model-view-controller (MVC)?**. 2011. Disponível em: <http://www.oficinadanet.com.br/artigo/desenvolvimento/o_que_e_model-viewcontroller_mv_c/>. Acesso em: 14 de Outubro de 2015.
- CAMPOS, Fernando Celso de; BELHOT, Renato Vairo. **Gestão de manutenção de frotas de veículos**: uma revisão. Gest. Prod., São Carlos, v. 1, n. 2, p. 171-188, Agosto de 1994.

Disponível em:
<http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0104-530X1994000200004&lng=en&nrm=iso> . Acesso em 03 de Setembro de 2015.

CHANGE VISION. **Astah Community**. 2015. Disponível em: <<http://astah.net>>. Acesso em: 15 Outubro de 2015.

CLEMENTE, Quebo Kenge. **Gestão de frota de veículos**. Lisboa: IST, 2008.

COELHO, L. C. **Gestão de frotas de veículos**. 2015. Disponível em: <<http://www.logisticadescomplicada.com/gestao-de-frotas-de-veiculos/>>. Acesso em 30 de Setembro de 2015.

COSTA, Andrew Diniz. **Testes de Software**. Disponível em: <<http://www.les.inf.pucrio.br/wiki/images/a/a8/Testes.pdf>>. Acesso em: 05 de Dezembro de 2014.

CTN CARDOSO. **CNT Car**. 2015. Disponível em: <<http://www.ctncar.com.br/>>. Acesso em 14 de Outubro de 2015.

FOWLER, M. **UML Distilled: Applying the Standard Object Modeling Language**. Addison-Wesley. Setembro de 2003.

GONÇALO, G. **Planilha de Controle de Combustível**. 2014. Disponível em: <<http://www.sofit4.com.br/blog/2014/06/planilha-de-controle-de-combustivel/>>. Acesso em 08 de Maio de 2015.

GOOGLE. **AngularJS**. Disponível em: <<https://angularjs.org/>>. Acesso em 17 de Outubro de 2015.

LIMA, A. F.; NOGUEIRA, J. C. **Conceitos de Manutenção**. Dissertação apresentada à Faculdade de Economia, Administração, Contabilidade e Ciência da Informação e Documentação (FACE) – Brasília – DF. 2006.

MICROSOFT CORPORATION. Developer Network: **C#**. 2015. Disponível em: <<http://msdn.microsoft.com/pt-br/library/kx37x362.aspx>>. Acesso em: 15 de Outubro de 2015.

MICROSOFT CORPORATION. **SQL Server 2014**. 2015. Disponível em: <<https://www.microsoft.com/pt-br/server-cloud/products/sql-server/default.aspx>>. Acesso em: 17 de Outubro de 2015.

MSDN. **Visão geral do ASP.NET MVC**. 2012a Disponível em: <[https://msdn.microsoft.com/pt-br/library/dd381412\(v=vs.108\).aspx](https://msdn.microsoft.com/pt-br/library/dd381412(v=vs.108).aspx)>. Acesso em: 01 ago. 2012.

OLIVEIRA, Eric. **Aprenda AngularJS com estes 5 Exemplos Práticos**. Disponível em: <<http://javascriptbrasil.com/2013/10/23/aprenda-angularjs-com-estes-5-exemplospraticos/>>. Acesso em: 27 de Setembro de 2015.

PRESSMAN, R. S. **Software Engineering: A Practitioner's Approach, 7/e**. McGraw-Hill, 2009.

SEITO, A. I.; GILL, A. A. **A segurança contra incêndio no Brasil**. Disponível em: <http://www.corpodebombeiros.sp.gov.br/novo/Downloads/aseguranca_contra_incendio_no_brasil.pdf>. Acesso em: 5 de Maio de 2015.

SILVA, Diego Farias; VICENTE, Guilherme Oliveira. Apresentando o Angular.js. Disponível em: <<http://www.dextra.com.br/apresentando-o-angular-js-4/>>. Acesso em: 15 de 27 Setembro de 2015.

SOFIT SOFTWARE SA. **Indicadores para Controle de Combustível**. Agosto de 2014.

VALENTE, A.M., et all. **Gerenciamento de Transporte e Frotas**. São Paulo: Cengage Learning 2008.

VALENTE, A.M., et all. **Gerenciamento de transporte e frotas - 2^a Edição**. São Paulo: Cengage Learning, 2011.