

## IDENTIFICAÇÃO DO ESCOPO DE SOFTWARE A PARTIR DA ANÁLISE DE REQUISITOS UTILIZANDO A UML

**Anderson Fernando dos Santos**

Graduando em Tecnologia em Análise e Desenvolvimento de Sistemas  
Faculdades Integradas de Três Lagoas – FITL/AEMS

**Erick Allan de Oliveira**

Graduando em Tecnologia em Análise e Desenvolvimento de Sistemas  
Faculdades Integradas de Três Lagoas – FITL/AEMS

**Gledson Fonseca da Silva**

Graduando em Tecnologia em Análise e Desenvolvimento de Sistemas  
Faculdades Integradas de Três Lagoas – FITL/AEMS

**Alan Pinheiro de Souza**

Mestre em Sistemas de Informação  
Docente das Faculdades Integradas de Três Lagoas – FITL/AEMS

### RESUMO

O objetivo da pesquisa foi utilizar a análise de requisitos para reconhecimento dos problemas e seus impactos no ambiente corporativo. As informações coletadas nessa etapa serão utilizadas para desenvolver um sistema computacional, visando rastreamento das informações corporativas, agregação de valores aos serviços oferecidos pela empresa e automatização de tarefas operacionais e gerenciais. Esse projeto adotou técnicas engenharia de requisitos, entrevistas e ferramenta de modelagem para determinação do escopo do sistema. A finalidade do sistema é apoiar tarefas para cadastrar assistente, supervisor, motorista, destino e veículo, registrar viagem e emitir relatório de motoristas, veículos e viagens.

**PALAVRAS-CHAVE:** Escopo; Requisitos; Modelagem; UML.

### INTRODUÇÃO

Em meio a um mercado dinâmico, as empresas devem procurar melhorias contínuas de seus processos de gestão como forma de adquirir ou manter vantagem competitiva. O adequado gerenciamento de movimentação de materiais, é um fator que poderá auxiliar a organização nesse nível de armazenagem e competitividade, tornando mais fácil o controle de entrada e saída de produtos do estoque, agilizando, assim, o processo da empresa como um todo, evitando a demora das tarefas e a perda de produtividade.

A proposta para desenvolver um sistema gerencial de entrada e saída de veículos na fábrica utilizando a análise de requisitos visando à melhoria contínua no processo organizacional da equipe de colaboradores, utilizando a linguagem UML (*Unified Modeling Language*) foi possível visualizar, especificar os artefatos para modelar o *software* baseado em diagramas, onde os aspectos essenciais como funcionais estruturação e interação e não funcional como tempo e confiabilidade e processamento das informações, podendo analisar *software* através de visões diferentes, concentrando um aspecto de cada vez, atendendo as necessidades ou restrições pré-estabelecidas pelos *stakeholders*.

Além das seções de Introdução, Considerações Finais e Referências, o projeto está organizado em três seções. A primeira seção apresenta uma visão do projeto com engenharia de requisitos. A segunda seção mostra o levantamento de requisitos, apresentando os problemas encontrados, proposta do desenvolvimento e definição dos atores. A terceira seção mostra a modelagem aplicada a UML destacando o diagrama de classe e diagrama de caso de uso.

## 1 ENGENHARIA DE REQUISITOS

A engenharia de requisitos é uma etapa que começa com a necessidade em conhecer todas as variáveis, identificando possíveis problemas para realização da atividade, analisando o que o cliente deseja e quem faz a atividade, observando quais passos das atividades e qual a melhor escolha das características do *software*.

De acordo com Pressman (2006), a engenharia de requisitos, como todas as outras atividades de engenharia de *software*, precisa ser adaptada às necessidades do processo, do projeto, do produto e do pessoal que está fazendo o trabalho. Na perspectiva do processo de *software*, a engenharia de requisitos é uma ação de engenharia de *software* que começa durante a atividade de comunicação e continua durante a atividade de modelagem. A engenharia de requisitos fornece um mecanismo apropriado para entender o que o cliente deseja, incluindo o conjunto de tarefas necessárias para entendimento dos impactos do *software* sobre o negócio e como usuários finais irão interagir com sistema.

A empresa possui dois níveis hierárquicos, bem divididos para aperfeiçoar e melhorar suas tarefas. O Supervisor é responsável pelo gerenciamento de volume

expedido e fechamento de volume de carregamento diário. O Assistente é responsável por lançar todos os dados do carregamento para uma planilha realizando a conferência das notas fiscais.

## 2 LEVANTAMENTO DE REQUISITOS

Nesta seção serão apresentadas as condições atuais de trabalho e uma solução para os problemas existentes no sistema atual na empresa. Nesta fase será realizado um levantamento das dificuldades que o usuário vem enfrentando no seu dia a dia.

A empresa para qual este projeto sendo elaborados prestações de serviços logísticos, emitindo produto direto da fábrica para a estação de terminal. O registro da movimentação interna de veículos é feito manualmente com o preenchimento de uma planilha impressa. O procedimento é realizado pelo assistente. Ao final do expediente é feita a conferência e o fechamento diário das saídas dos caminhões da fábrica.

### 2.1 Problemas Encontrados

Os problemas existentes relatados pela empresa durante o levantamento de requisitos. A empresa não possui cadastro de motorista formal, o que causa dificuldade na rastreabilidade do motorista. Além disso, os documentos formais de cadastros de veículos são de difíceis acessos, causando transtorno e lentidão na hora de procurar as fichas dos veículos. O registro da saída da fábrica e os relatórios são montados manualmente no *Microsoft Office Word*.

### 2.2 Proposta de Desenvolvimento

A linguagem de modelagem UML foi adotada no desenvolvimento do projeto visando construir um sistema de qualidade para auxiliar a administração da empresa nos cadastros, com um ambiente visual e interativo de acordo com a coleta de informações realizada no levantamento de requisitos. Segue lista de eventos identificados nessa etapa:

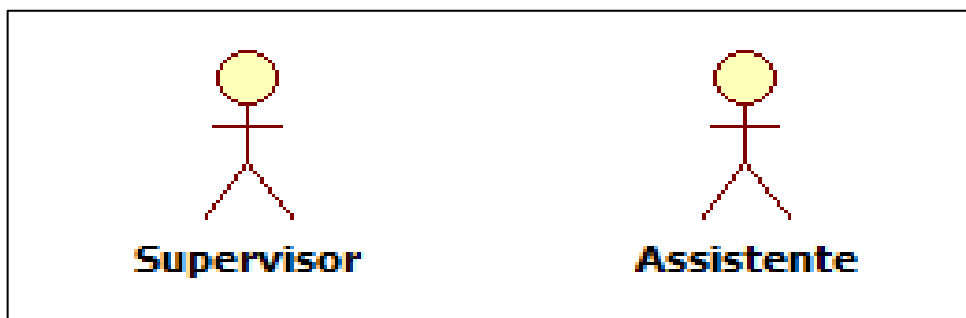
1. Cadastrar Assistente;
2. Cadastrar Supervisor;
3. Cadastrar Motorista;
4. Cadastrar Veículo;
5. Cadastrar Destino;
6. Registrar Viagem;
7. Emitir Relatório de Motoristas;
8. Emitir Relatório de Veículos;
9. Emitir Relatório de Viagens.

### 2.3 Definições de Atores

Atores são usuários (ou outros meios externos) que desenvolvem algum papel em relação ao sistema. Os meios externos são *hardwares* e/ou *softwares* que, assim como os usuários, geram informações para o sistema ou necessitam de informações geradas a partir do sistema. Segundo Furlan (1998), “um ator é um agente que interage com o sistema, um tipo de usuário ou categoria com um papel definido, podendo incluir seres humanos, máquinas, dispositivos ou outros sistemas”. Durante a etapa de levantamento de requisitos foram detectados dois principais atores conforme mostrado na Figura 1 que utilizará o sistema.

- **Supervisor:** possui privilégios e acesso a todas as funções do sistema.
- **Assistente:** possui acesso às funções necessárias para cadastrar viagem.

**Figura 1:** Descrição dos atores.



**Fonte:** Elaborado pelos autores (2015).

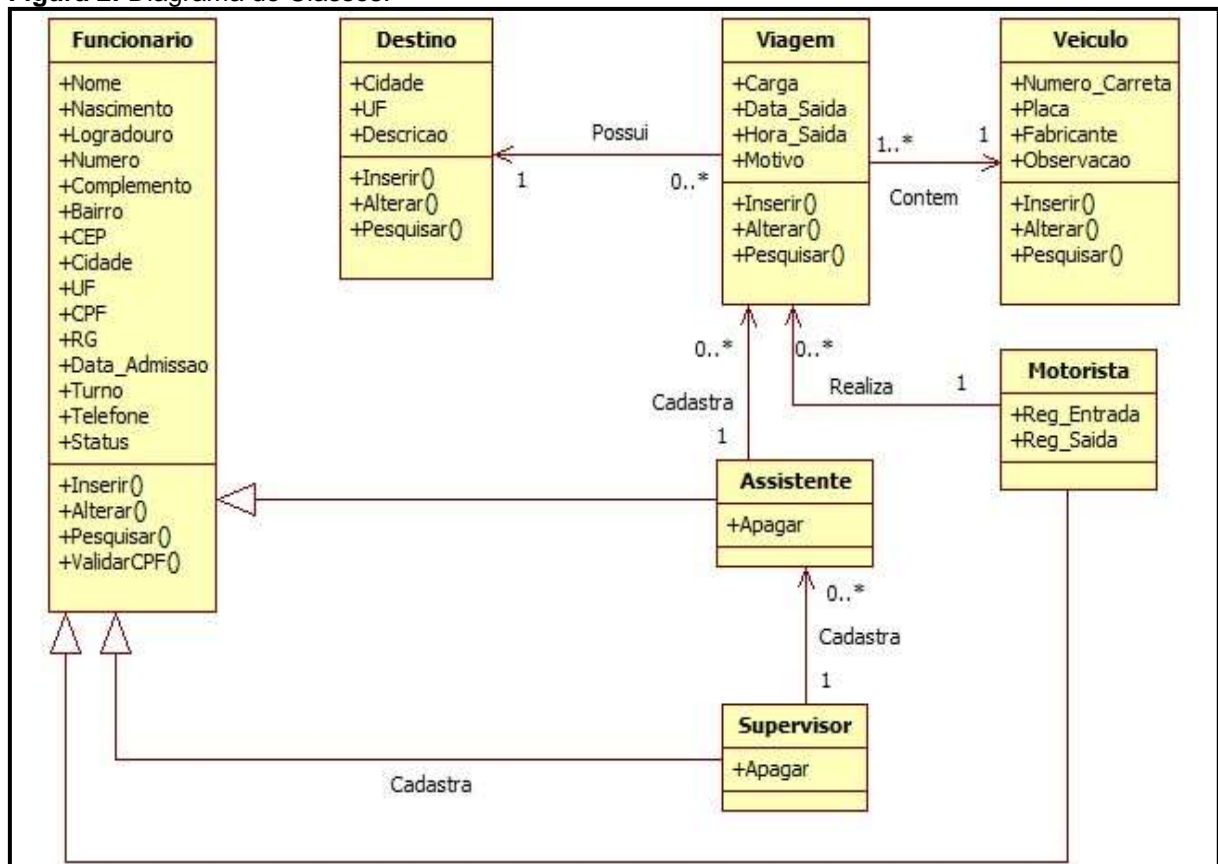
### 3 LINGUAGEM DE MODELAGEM UML

A UML (*Unified Modeling Language*) é uma linguagem-padrão para a elaboração da estrutura de projetos de *software*. Ela poderá ser empregada para visualizações, a especificações, a construção e a documentação de artefatos que façam uso de sistemas complexos de *software* (BOOCH; RUMBAUGH; JACOBSON, 2000). Segundo Furlan (1998), diagrama como uma apresentação gráfica de uma coleção de elementos de modelo, frequentemente mostrado como um gráfico conectado de arcos (relacionamentos) e vértices (os outros elementos).

#### 3.1 Diagrama de Classes

O diagrama de classes é muito utilizado na modelagem de sistemas orientados a objetos, pois possibilita a visão de todo o relacionamento e colaboração entre as classes do sistema. Diagramas de classes servem de base para outros tipos de diagramas existentes no projeto, importante para visualização dos modelos de estrutura do sistema. Segundo Furlan (1998), diagrama de classes trata-se de uma estrutura lógica estética em uma superfície de duas dimensões mostrando uma coleção de elementos declarativos de modelo, como classes, tipos e seus respectivos conteúdos e relações. A Figura 2 demonstra o diagrama de classes que foi elaborado na modelagem do sistema. As classes Assistente, Supervisor e Motorista herdam da classe Funcionário. Uma Viagem é realizada pelo Motorista e contém registros de Destino e Veículo.

**Figura 2:** Diagrama de Classes.



Fonte: Elaborado pelos autores (2015).

### 3.2 Diagramas de Caso de Uso

Casos de uso são utilizados para entender melhor e com precisão os requisitos do sistema, nele serão descritas as principais funcionalidades do sistema e a interação dessas funcionalidades com os usuários do mesmo sistema. Segundo Furlan (1998), caso de uso é um conjunto de ações que um sistema desempenha para produzir um resultado observável de valor a um ator específico. Estes diagramas criam uma ótica externa das interações do sistema com o mundo exterior, representando em alto nível de funcionalidade intencional conforme os estímulos recebidos.

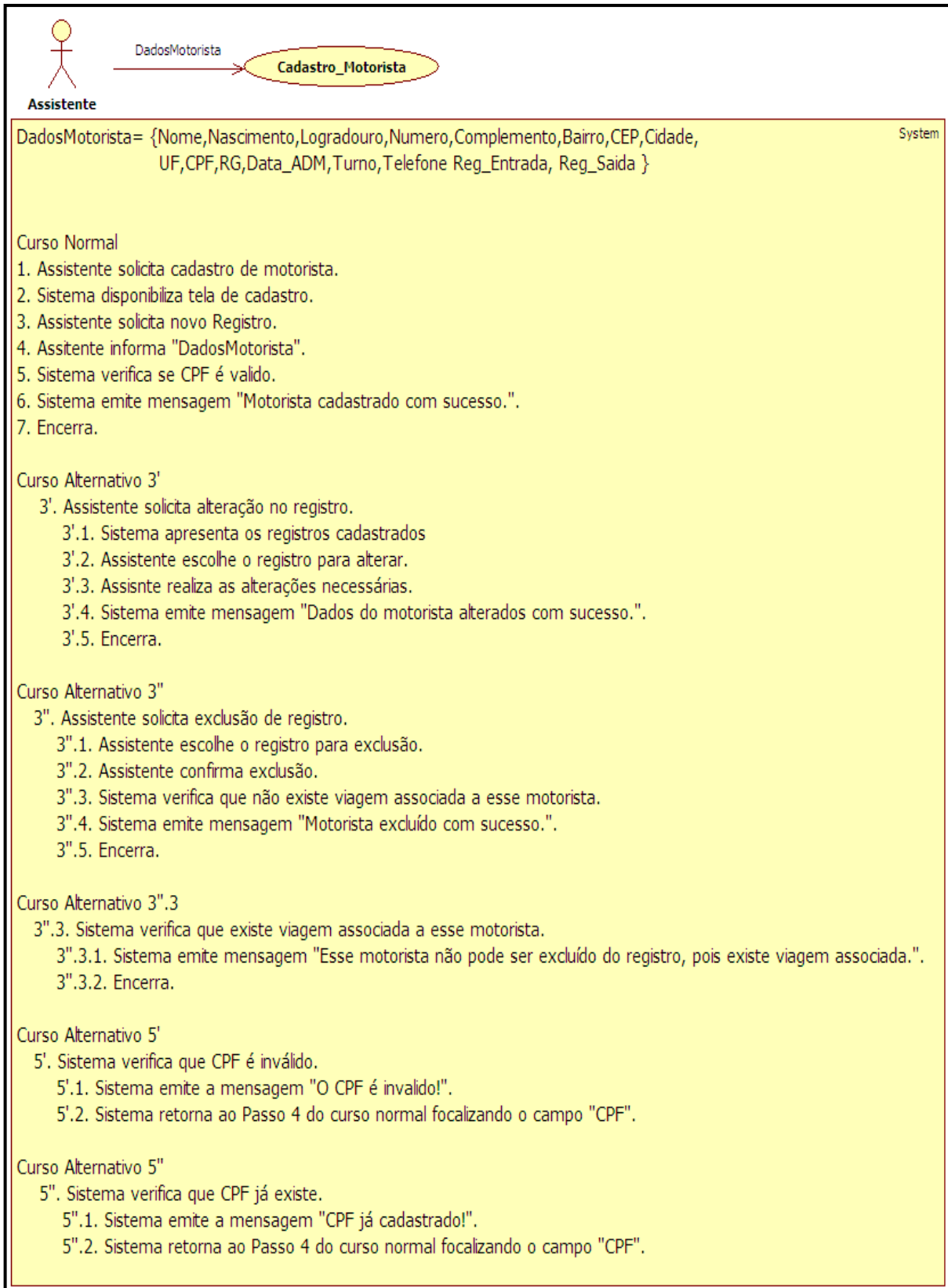
O diagrama de casos de uso serve para visualizar os relacionamentos entre os atores e os casos de uso do sistema (cenários), numa visão geral. Os diagramas de caso de uso são importantes principalmente para a organização e modelagem dos comportamentos de um sistema. É uma descrição de um conjunto de sequências de ações, inclusive variantes, que um sistema executa para produzir um

resultado de valor observável por um ator. Esses diagramas “devem conter casos de uso, atores e relacionamentos de dependência, generalização e a associação. Assim como os outros diagramas podem conter notas e restrições” (BOOCH; RUMBAUGH; JACOBSON, 2000).

Foram elaborados diagramas de casos de uso, assim como curso normal e cursos alternativos para cada evento definidos no levantamento de requisitos. O detalhamento foi necessário para poder identificar e compreender as situações de interação dos atores e os comportamentos do sistema.

A Figura 3 apresenta diagrama de caso de uso para o evento “Cadastrar Motorista”. Para realizar esse cadastro, Assistente solicita previamente novo registro na tela de cadastro de motorista. O curso normal apresenta passo a passo para cadastro dos dados do motorista, conforme registrado no pacote de dados “DadosMotorista”. Ao final desse procedimento, o sistema emite a mensagem “Motorista cadastrado com sucesso”. O curso alternativo 3’ apresenta etapas cujo supervisor solicita alteração nos registros, podendo fazer os ajustes de forma direta modificando os dados do motorista. Os cursos alternativos 3” e 3”.3 mostram passos a serem seguidos para exclusão de registro. Pode ser observado no roteiro 3”.3 que nem sempre a exclusão será permitida. Caso exista viagem associada ao motorista, o usuário do sistema não conseguirá apagar o registro. Essa verificação é importante para garantir a integridade das informações contidas no repositório de dados. Já os cursos alternativos 5’ e 5” possuem a função de verificar se o CPF informado pelo motorista é válido e consultar se o CPF informado já possui cadastro na base de dados.

**Figura 3:** Diagrama de Caso de Uso – Cadastrar Motorista.



**Fonte:** Elaborado pelos autores (2015).

## CONSIDERAÇÕES FINAIS

Através deste projeto foi possível oferecer um sistema para informatizar as principais funções de entrada e saída de veículos, facilitando o rastreamento das informações. A finalidade do sistema é gerenciar as operações para cadastrar assistente, supervisor, motorista, destino e veículo, registrar viagem e emitir relatório de motoristas, veículos e viagens.

Os diagramas criados pela ferramenta *StarUML* foram importantes, pois permitiram identificação e especificação dos requisitos, documentação das informações do projeto, além de apresentar recursos gráficos para a elaboração e editoração estruturada e orientada a objetos.

Essa modelagem será adotada para apoiar as atividades seguintes de projeto, codificação e teste do sistema a ser desenvolvido. A partir da adoção de conceitos do paradigma de orientação a objetos, espera-se construir um sistema com facilidade de manutenção e evolução das suas funcionalidades. Isso será importante para atender as necessidades de negócio futuras da empresa.

## REFERÊNCIAS

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML: guia do usuário**. 12ª Ed., Rio de Janeiro: Campus, 2000.

FURLAN, D. **Modelagem de Objetos Através da UML**. São Paulo: Makron Books, 1998.

PRESSMAN, R. **Engenharia de Software**. 6ª ed. São Paulo: MacGraw-Hill, 2006.