

# **APLICAÇÃO DA MODELAGEM UML NA FASE DE ANÁLISE DE UM PROJETO DE SOFTWARE PARA AGENDAMENTO DE USO DE VEÍCULOS INTERNOS DE UMA EMPRESA**

**ANDRE APARECIDO LEAL DE ALMEIDA**

Discente da AEMS  
Faculdades Integradas de Três Lagoas – MS

**ELISANGELA CITRO**

Docente da AEMS  
Faculdades Integradas de Três Lagoas – MS  
Mestre em Ciência da Computação  
Área de Engenharia de Software

O uso de modelagem para o desenvolvimento de *software* é fator preponderante para a elaboração de um projeto de *software* que atenda seu objetivo, a linguagem de modelagem UML (*Unified Modeling Language*) é uma técnica que apresenta diagramas que auxiliam o processo de desenvolvimento de *software*. Este artigo tem por objetivo aplicar os diagramas da UML na fase de análise de um projeto de *software*, procurando gerar solução para os requisitos extraídos em entrevista com o usuário. Como meio de apoiar esta pesquisa foi utilizado um estudo de caso real em uma empresa privada que agenda o uso de veículos da empresa para funcionários. Foram levantados os requisitos para a construção do sistema, foi gerada a análise usando a modelagem UML e foi verificado que o diagrama de caso de uso auxilia no entendimento dos requisitos do sistema e o diagrama de classe na construção de um modelo conceitual para geração da base de dados.

**Palavras-chave:** análise, diagrama de caso de uso, diagrama de classe, processo de desenvolvimento, UML.

## 1 - INTRODUÇÃO

Atualmente, é necessário ter uma ferramenta estratégica que atenda a uma determinada demanda e facilita os processos dentro de um ambiente empresarial, ajudando a reduzir e aprimorar as tarefas manuais.

Qualquer empresa de grande ou médio porte que possui uma frota de veículos, independentemente da quantidade, se não gerenciar o uso desses veículos, provavelmente começarão a surgir vários problemas como: um colaborador precisa do veículo para realizar uma tarefa muito importante para a empresa, mas, outro colaborador saiu com o veículo para realizar algo que poderia ter deixado para resolver num outro horário. Outro desafio é a manutenção preventiva dos veículos: troca de óleo, alinhamento, calibragem, balanceamento e até mesmo os abastecimentos podem ficar em segundo plano e tornar um problema bem maior, pois todos usam o veículo e há falta de gerenciamento adequado.

Considerando estes apontamentos, se faz necessário desenvolver e implementar um sistema na empresa, que garanta o gerenciamento de forma rápida, fidedigna e atua enquanto ferramenta facilitadora nos processos de gestão da empresa.

### 1.1 - Objetivo

Este trabalho tem como objetivo principal mostrar a importância de aplicar as técnicas da modelagem UML na fase de análise de um projeto de *software* para agendamento de uso de veículos internos de uma empresa e projetar o sistema por meio dos diagramas de classe e casos de uso.

## **2 – A LINGUAGEM DE MODELAGEM UML**

Este capítulo apresenta uma definição geral da UML, os diagramas mais importantes e mais utilizados na modelagem que são: diagrama de classe e diagrama de casos de uso. Mostra também que é uma linguagem facilitadora, ajudando compreender o projeto em todos os processos de desenvolvimento.

### **2.1 – Definições Gerais da UML**

A UML - *Unified Modeling Language* é uma linguagem gráfica para visualizar, especificar, construir e documentar um *software* através dos diagramas que são compostos de gráficos, símbolos e texto permitindo compreender de forma detalhada, conclusiva, sem ambiguidades e seus modelos pode ser conectados a diferentes linguagens de programação.

Em outras palavras, diagrama são imagens, figuras ou esboço do projeto desenhado no papel.

### **2.2 - Diagrama de classes**

Grady Booch, James Rumbaugh, Ivar Jacobson em [BOOCH; RUMBAUGH; JACOBSON], escreve que um diagrama de classe mostra um conjunto de classes, interfaces e colaboração e seus relacionamentos.

São os diagramas mais encontrados e um dos mais importantes quando o assunto é orientação a objeto. É onde se abstrai as informações estruturais mínimas do *software*.

A Figura 2.1 apresenta um diagrama de classe com duas classes onde a classe condutor herda atributos da classe funcionário.

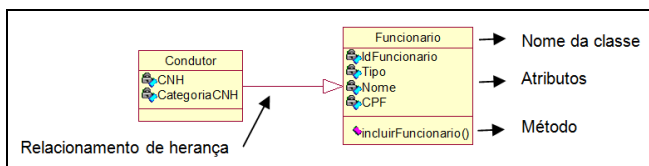


Figura 2.1 – Diagrama de classe

## 2.4 - Diagramas de caso de uso

FURLAN (1998) escreve que um diagrama de caso de uso é um diagrama da UML para descrever a visão externa de um sistema e suas interações com o mundo exterior.

Especifica todo o comportamento e descreve as interações entre os usuários do sistema e o sistema em si. Não é necessário saber o que deverá acontecer dentro do sistema, pois os casos de uso devem ser suficientemente claros ao descrever o que um sistema faz. Deve mostrar também o início e o fim do fluxo de eventos mostrando o seu caminho normal e alternativo. Todo meio externo que interage com o sistema recebe o nome de ator. Um ator pode ser um *hardware*, um ser humano e ou outro. Um diagrama de casos de uso é composto por dois blocos: o primeiro bloco é composta por: diagramas, símbolos gráficos.

Conforme mostra a Figura 2.2, é possível verificar com mais detalhes a parte gráfica de um caso de uso onde o ator supervisor esta enviando “Dados Veiculo” para o sistema “Cadastrar veiculo”.

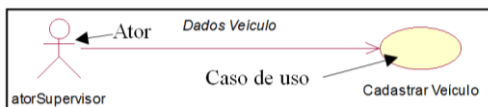


Figura 2.2 – Caso de uso

A Figura 2.3 contém a segunda parte, a textual que é a especificação do diagrama de caso de uso, descrevendo o curso normal e os cursos alternativos desse cadastro de veículo.

Nesta figura podemos perceber toda descrição textual através de um único curso normal onde é feito um cadastro de um veículo que nunca foi cadastrado e os outros cursos alternativos é para quando o veículo já esta cadastrado, podendo alterar ou excluir o referido cadastro.

```
DadosVeiculo = {IDVeiculo , Marca, Modelo, Cor, Tipo, Km, Placa, Status}
Curso normal
    1. Supervisor solicita cadastro de veículo.
    2. Sistema disponibiliza tela de cadastro.
    3. Supervisor solicita novo registro e informa placa.
    4. Sistema verifica que a placa não esta cadastrada.
    5. Supervisor informa dados do veículo.
    6. Sistema cria instância veiculo1 e emite mensagem "gravação ok".
    7. Encerra use case.

Curso alternativo 3.
    3. Supervisor solicita pesquisa e verifica que o veículo esta cadastrado
    3.1. Supervisor solicita alteração e informa novos dados
    3.2. Sistema atualiza veiculo1 e emite mensagem "gravação ok".
    3.3. Encerrar use case.

Curso alternativo 3.1
    3.1 Supervisor solicita exclusão do veículo
    3.1.1. Sistema verifica se o veículo não está associado a algum
    percurso, exclui registro e emite mensagem "exclusão ok".
    3.1.2. Encerrar use case.

Curso alternativo 4
    4- Sistema verifica que a placa existe.
    4.1- Ir para o passo 3.1 do curso alternativo 3.
```

Figura 2.3 - Especificação do caso de uso

### 3 – ESTUDO DE CASO

Com base em estudo de caso realizado em uma empresa privada de ensino, este capítulo vai abordar o levantamento de requisitos que ajudará a compreender como são coletados os dados de cada percurso realizado pelos veículos da empresa, quais os problemas existentes com este controle e com base nestas informações, é criado à lista de eventos. Na análise, através do diagrama de classe e diagrama de

casos de uso da UML, é apresentado o aspecto comportamental, os atores, eventos e relacionamentos necessários para o sistema de agendamento de veículos.

### **3.1 – Objetivo do Sistema**

Este sistema será utilizado para o agendamento e gerenciamento de veículos. A empresa, em estudo, possui uma frota de veículos, caso algum funcionário queira sair de imediato da empresa com algum veículo, o mesmo precisa ter autorização para dirigir os veículos da empresa, pois nem todo funcionário pode ser um condutor, mas todo condutor deve ser funcionário da empresa. O condutor precisará informar dados do percurso tanto na saída como na chegada do veículo: data, hora, Km de saída/chegada e local (aonde vai com o veículo). Caso o veículo não esteja disponível, o condutor agendará esse veículo informando os dados do percurso: o dia, horário e local. Poderá ter vários agendamentos para o mesmo período, cabendo ao supervisor administrativo definir e ordenar o percurso de maior prioridade.

### **3.2 - Levantamento de Requisitos**

SOMMERVILLE (2003) avalia a UML como sendo um padrão para modelagem orientada a objetos, e aponta a visão dos casos de uso como sendo uma fonte de obtenção de requisitos importantíssima.

#### **3.2.1 - Descrição do Sistema Atual**

O supervisor administrativo imprime uma tabela com data, nome do condutor e km. É preenchida, manualmente, antes de o veículo sair do pátio da empresa e depois que chega.

### **3.2.2 - Problemas existentes**

Em virtude de este controle ser feito manualmente, os problemas encontrados foram:

- O condutor pode informar dados que não correspondem a um determinado trajeto, e anotar, acidentalmente, um Km inicial ou final e passar uma informação de percurso e deslocamento inconsistente, ou seja, um Km muito inferior ou superior ao trajeto efetuado.
- Dificuldades em obter um relatório preciso e rápido.

### **3.2.3 - Desejos do Usuário**

O usuário deseja um sistema que armazene os dados de todos os trajetos feitos com os veículos a fim de gerar relatórios e ter controle de manutenção, abastecimento e gerenciar os trajetos realizados pelos condutores.

## **3.3 - Análise**

É durante a análise que devemos descrever o que o cliente exige e estabelecer a base para a criação de um projeto de *software*, definindo um conjunto de requisitos a serem validados após a construção do *software* com a finalidade de detalhar através de diagramas.

PRESSMAN (1995) defende que o objetivo da análise é fornecer uma descrição dos domínios informal, funcional, e comportamental necessário a um sistema baseado em computador.

### **3.3.1 - Lista de Eventos**

Após analisar os requisitos do sistema, a equipe de desenvolvimento identificou os seguintes eventos, conforme Tabela 3.1

Tabela 3.1 - Lista de eventos

1. Cadastrar funcionário	10. Relatório de funcionário
2. Cadastrar condutor	11. Relatório de condutor
3. Cadastrar veículo	12. Relatório de veículo
4. Cadastrar serviço	13. Relatório de serviço
5. Movimento ordem serviço	14. Relatório de ordem de serviço
6. Movimento reserva percurso	15. Relatório de reserva percurso
7. Movimento aprovar percurso	16. Relatório de aprovar percurso
8. Movimento finalizar percurso	17. Relatório de finalizar percurso
9. Movimento de abastecimento	18. Relatório de abastecimentos

### 3.3.2 - Atores do Sistema

Conforme análise, concluiu-se que serão utilizados três atores, conforme ilustrado na Figura 3.2. A função do ator condutor é agendar, realizar percursos. O ator funcionário finaliza os percursos e o ator supervisor realiza todas as inclusões, alterações, exclusões de dados, gerar todos os relatórios, além de aprovar ou não aprovar os percursos solicitados pelos condutores.

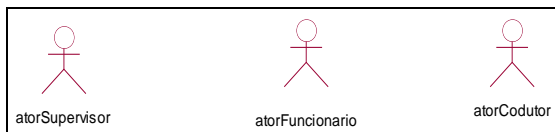


Figura 3.2 - Atores do sistema

### 3.3.3 - Diagrama de Classe

Após analisar foi modelado o diagrama de classe, Figura 3.3 que mostra um conjunto de itens, classes e relacionamentos, estabelecidos a partir da implementação do agendamento de veículos. A classe “Funcionario” tem um atributo em particular, chamado “Tipo” para identificar se esse funcionário é ou não condutor. A classe “Condutor” herda os atributos e métodos da classe “Funcionario” e se relaciona com a

classe “Veiculo” podendo um condutor reservar vários veículos, a partir deste relacionamento surge a classe “Percurso” onde o condutor irá agendar ou realizar o percurso. Quando um veículo estiver sem combustível ou precisar de reparos é registrado, respectivamente, nas classes “Abastecimento” e “Servico” as referidas informações do abastecimento ou da ordem de serviço. As classes “Marca”, “TipoVeiculo” e “Modelo” também se relaciona com a classe Veiculo e o objetivo principal é fazer com que o usuário do sistema cadastra separadamente os respectivos atributos referente a cada uma dessas classes evitando entradas no banco de informações inconsistentes.

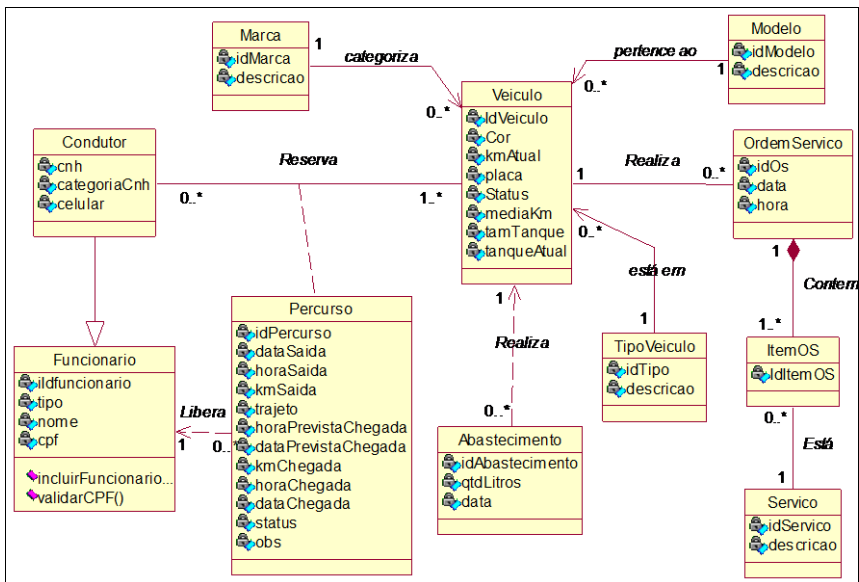


Figura 3.3 – Diagrama de Classe

### 3.3.4 - Diagrama de Casos de Uso

A abstração dos requisitos sobre o agendamento e gerenciamento de veículos, é apresentada pela Figura 3.4 que mostra todo esse levantamento modelado por meio do diagrama de casos de uso. Neste

diagrama verifica-se a presença dos seguintes atores: ator condutor, o ator funcionário e o ator supervisor em seus respectivos cenários.

Perceba que o diagrama de casos de uso, Figura 3.4 mostra também todo cenário do sistema. Os três atores realizar um evento em comum que é “Cadastrar serviços”. O ator supervisor é o que mais interage com o sistema.

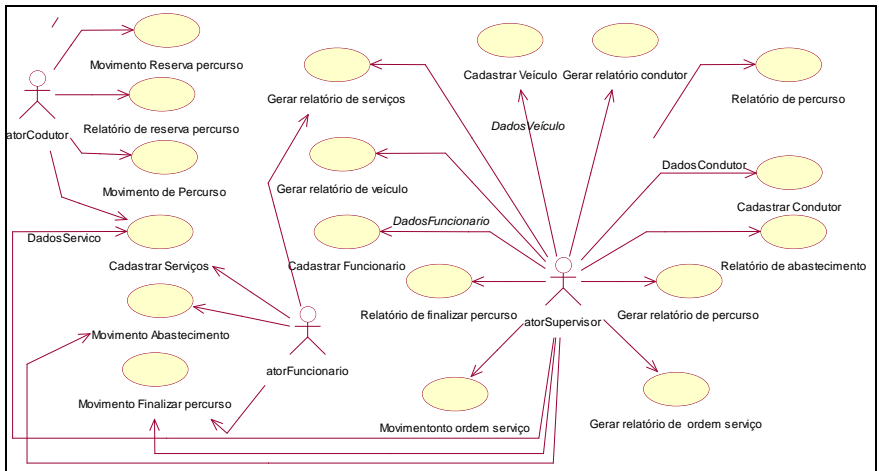


Figura 3.4 - Diagrama de Casos de Uso

O ator condutor, conforme o caso de uso acima realiza o movimento de reserva e relatórios de percurso, movimento de percurso e cadastra serviços nos veículos.

A função principal do ator condutor é além de realizar quase todas as entradas, alterações e exclusões ele também tem um papel extremamente importante que é aprovar ou não os percursos solicitados pelos condutores.

O ator funcionário cadastra e gera relatórios de serviços e finaliza os percursos realizados pelos condutores.

#### **4 – CONCLUSÃO**

Construir um *software* de computador não é uma tarefa simples, fácil e rápida se não fossem levadas em consideração as etapas de construção que são: engenharia de requisitos, de modelagem, e de análise. Levando em consideração o que Pressman (1995) diz logo no início do primeiro capítulo, ele defende que um *software* afeta a vida de milhões e milhões de pessoas no mundo inteiro, pois, este está presente direta ou indiretamente em tudo o que vemos. Portanto, apenas desenvolver linhas de códigos, sem antes submeter toda a construção deste projeto de desenvolvimento, às etapas de engenharia de *software*, o mesmo, poderá facilmente, tornar-se um projeto complicado, difícil e cronologicamente demorado e no final o desenvolvedor poderá entregar um produto sem qualidade, que não satisfaça as reais necessidades do cliente, além de poder gerar um sistema incompleto. E isso sem dúvidas estará afetando, negativamente, a vida das pessoas.

A implantação de modelos usando a UML para visualizar, documentar e representar o sistema, é sem dúvida alguma, uma ferramenta eficiente, pois com os diagramas da UML é possível verificar todo comportamento do sistema, através de símbolos, gráfico e figuras, tornando-o fácil, em qualquer ponto do seu desenvolvimento, ler, entender e acertar os detalhes com o cliente e ver se a sua vontade esta sendo contemplada. Modelar com a UML é, essencialmente, o primeiro passo para o sucesso e mesmo que surgindo os problemas , ficaria fácil de ser tratados. Um sistema alicerçado na UML é um sistema com mais qualidade e menos manutenção.

## **5 – REFERÊNCIAS BIBLIOGRÁFICAS**

BOOCH, Grady; RAMBAUGH, James; JACOBSON, Ivar. **UML: Guia do Usuário**. Rio de Janeiro: Campus, 2000. 473 p.

FURLAN, Furlan, José Davi. **Modelagem de Objetos através da UML**. São Paulo: MAKRON Books, 1998. 329 p.

SOMMERVILLE, Ian. **Engenharia de Software**. 6<sup>a</sup> ed., São Paulo: Addison Wesley, 2003. 580 p.

PRESSMAN, Roger. **Engenharia de Software**. 6<sup>a</sup> ed., São Paulo: Makron Books, 1995. 1056 p.