

# Revisão de UML

## Problema do Gasto de Combustível

Vinícius S. Branco<sup>1</sup>

<sup>1</sup>Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)

`vinicius.branco@acad.pucrs.br`

**Resumo.** *Este artigo traz uma breve introdução e descrição de modelos UML, o enunciado de um problema que serviu de base para a modelagem de dois diagramas, casos de uso e classes, e uma lista com as definições de pronto do trabalho. Esse trabalho foi proposto na disciplina de Engenharia de Software Orientada a Modelos, do período de 2020/2.*

### 1. UML

A UML, Linguagem Unificada de Modelagem, é uma linguagem gráfica para visualização, especificação, construção e documentação de artefatos do sistemas complexos de software. Seu primeiro esboço sendo lançado em 1995 com os motivos de: com o aumento de métodos de análise de projetos orientados a objetos, fazia mais sentido continuar essa evolução de forma conjunta. Com a unificação traria mais estabilidade ao mercado de orientada a objetos. Por fim, com a criação de um novo método seria aprimorado com o aprendizado adquirido nos métodos passados. [Booch et al. 2006]

A UML proporciona um padrão para construção e documentação de arquitetura de projetos de sistemas, incluindo aspectos conceituais tais como processos de negócios e funções do sistema, além de itens concretos como as classes escritas em determinada linguagem de programação, esquemas de banco de dados e componentes de software reutilizáveis.[Booch et al. 2006] Trazendo a versatilidade de trabalhar a especificação de diferentes partes do sistema, aumentando a capacidade de organização do sistema, facilidade de garantir a conformidade com requisitos, trabalhar a escalabilidade, reutilização e manutenção do software. Pode ser utilizado como o principal artefato para a implementação de um sistema de alta qualidade.

A modelagem é uma parte central de todas as atividades que levam à implementação de um bom software. Construímos modelos para comunicar a estrutura e o comportamento desejados do sistema, conformidade com requisitos. Construímos modelos para visualizar e controlar a arquitetura do sistema. Construímos modelos para compreender melhor o sistema que estamos elaborando, muitas vezes expondo oportunidades de simplificação e reaproveitamento, diminuição da complexidade de desenvolvimento do sistema. Construímos modelos para gerenciar os riscos e documentar decisões tomadas.

### 2. Enunciado e Modelos

O trabalho 1 da disciplina de Engenharia de Software Orientada a Modelos traz a proposta de com base em um enunciado escolhido previamente, desenvolver uma comunicação organizada, para isso era necessário desenvolver dois diagramas UML, o de casos de uso

e de classes. Também como uma forma de organização era necessário desenvolver uma lista de verificação, com pontos que deviam ser feitos nas tarefas para estar pronto, pode-se dizer que são as *definition of done*. Para desenvolvimento dos diagramas foi utilizada a ferramenta UML Astah, todos os artefatos gerados no trabalho estão no repositório <https://github.com/vinisbranco/t1ESmodelos>.

## 2.1. Enunciado

O Joãozinho quer calcular e mostrar a quantidade de litros de combustível gasto em uma viagem, usando um carro que faz 12Km/L. Para isso, ele gostaria que você o ajudasse por meio de um programa simples. Para realizar o cálculo, é necessário ler o tempo gasto (em horas) e a mesma velocidade média (km / h). Desta forma, você pode obter a distância e então calcular quantos litros seriam necessários. Mostre o valor com três casas decimais após o ponto. [Judge 2020]

## 2.2. Diagrama de Casos de Uso

Seguindo a especificação do problema, o caso de uso do sistema é calcular o quanto de gasolina foi consumida em um determinado trajeto usando um determinado carro, por isso no diagrama é descrito apenas uma associação do ator com o caso de uso *Calculate Fuel Spent* representado na Figura 1. Como o cálculo do gasto de combustível inclui outros dois cálculos, o de cálculo de distância percorrida e cálculo de litros necessários para percorrer essa distância, foi descrito no diagrama esses outros dois casos incluídos no primeiro.

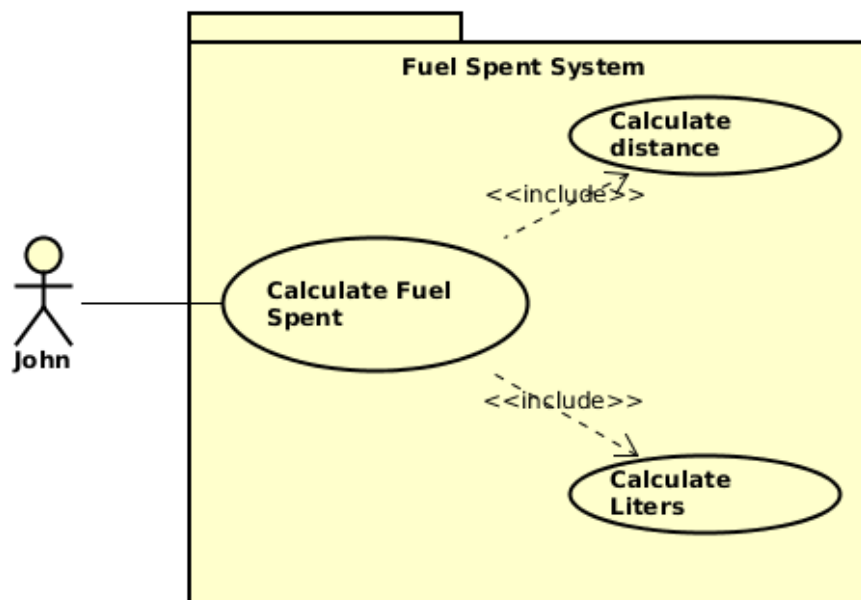


Figura 1. Diagrama de Casos de Uso

## 3. Diagrama de Classes

Com o objetivo de separar responsabilidades na arquitetura do sistema, facilitando possíveis incrementos no desenvolvimento da solução. Foram modeladas duas classes

no diagrama apresentado na Figura 2. A classe *travelInformation* que é responsável por ter os dados de uma viagem e calcular as informações necessárias sobre esta viagem específica, e a classe *CalculateFuelSpent* que recebe os dados do usuário, e gerencia os cálculos e informações de entrada e saída do sistema.

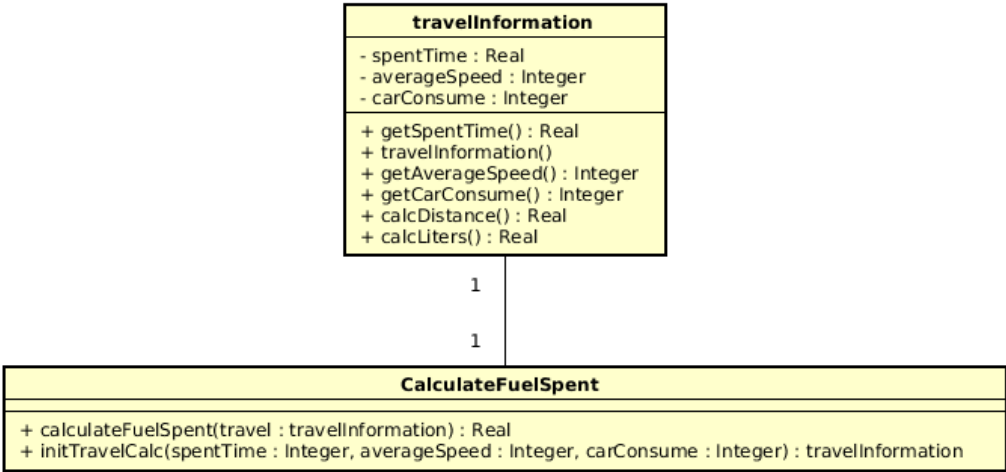


Figura 2. Diagrama de Classes

#### 4. Lista de Verificação

Os seguintes itens da Figura 3 foram considerados essenciais para executar a tarefa proposta, e depois de realizadas podendo considerar a tarefa pronta. São itens tanto de tarefas que deviam ser executadas e também de pontos a serem avaliados para garantir a qualidade do que foi desenvolvido.

1	Utiliza tipos primitivos?	<input checked="" type="checkbox"/>
2	Conformidade com requisitos?	<input checked="" type="checkbox"/>
3	Separação de responsabilidades nas classes?	<input checked="" type="checkbox"/>
4	Todos casos de uso cobertos?	<input checked="" type="checkbox"/>
5	Possibilidade de calculo de combustivel com diferentes circunstancias	<input checked="" type="checkbox"/>
6	Arquitetura escalavel	<input checked="" type="checkbox"/>
5	Lista de Verificação com no minimo 7 itens	<input checked="" type="checkbox"/>
6	Repositório com todos artefatos	<input checked="" type="checkbox"/>
7	Diagrama de Classes	<input checked="" type="checkbox"/>
8	Diagrama de Casos de Uso	<input checked="" type="checkbox"/>
9	Relatório no formato SBC	<input checked="" type="checkbox"/>
10	Referências	<input checked="" type="checkbox"/>

Figura 3. Lista de Verificação

#### Referências

Booch, G., Rumbaugh, J., and Jacobson, I. (2006). *UML: guia do usuário*. Elsevier Brasil.

Judge, O. (2020). Fuel spent problem.