**Consumo Médio - Modelagem em Texto**

**Giulia Ferraro1, Isabella Pagnoncelli1**

1Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)

Avenida Ipiranga, 6681 Partenon - Porto Alegre - RS - Brasil

1. **Introdução**

Este relatório tem como objetivo descrever o processo de modelagem em texto do problema URI 1014 - Consumo retirado do site *URI Online Judge* [URI Online Judge]. Neste documento iremos modelar o problema citado anteriormente utilizando histórias de usuário, critérios de aceitação, testes e linguagem Gherkin para compreendermos melhor o que é modelo em texto. Os arquivos de cada etapa deste relatório estão contidos no repositório do *GitHub* (<https://github.com/GiuliaFerraro/Trabalho2_ESOM>).

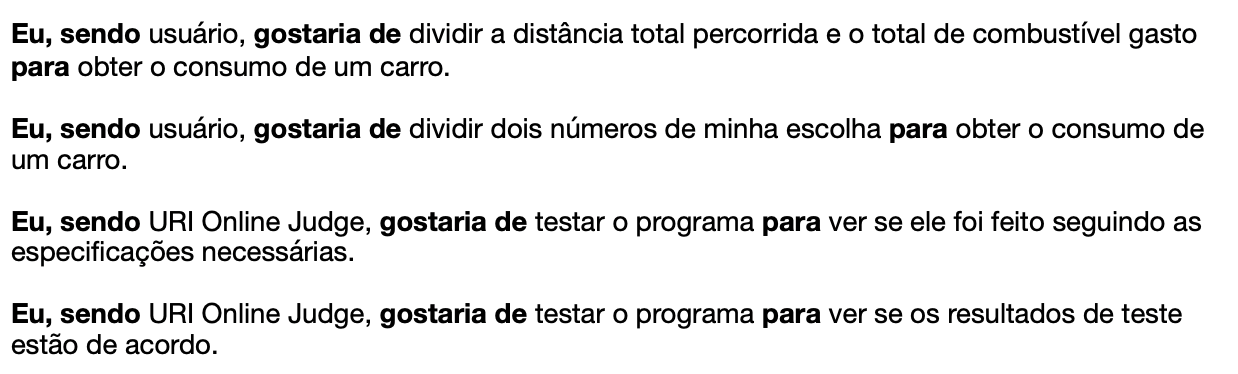
1. **Enunciado**

O problema a ser modelado neste relatório é o de número 1014 da plataforma *URI Online Judge* [URI Online Judge]. Segundo a plataforma, o programa deverá realizar o cálculo do consumo médio de um veículo, ou seja, quantos quilômetros ele consegue rodar com apenas um litro de combustível.

Neste programa, a entrada será um valor inteiro representando a distância total percorrida e um valor real representando o total de combustível gasto. Após os dados de entrada serem registrados, a distância total percorrida será dividida pelo total de combustível gasto e a saída do programa será quantos quilômetros o veículo consegue rodar com apenas um litro de combustível.

1. **Histórias de usuário**

Uma história de usuário é como se fosse uma descrição informal e geral, sem entrar em muitos detalhes técnicos, de algum recurso do *software*. Podemos dizer que histórias de usuário especificam o valor que aquele recurso pode oferecer a persona [Atlassian Agile Coach].

****

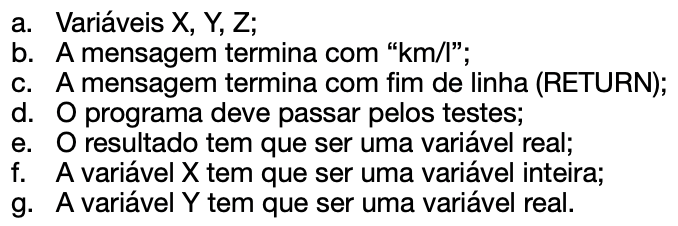
**Figura 1: Histórias de usuário para o problema de consumo.**

Acima estão descritas as histórias de usuário do problema de Consumo Médio que estamos modelando utilizando apenas modelos em texto. Como podemos assumir a partir das histórias, o valor para o usuário que o recurso dá é o de calcular o consumo médio de um veículo. O usuário gostaria de colocar dois números: um correspondente a distância total percorrida e outro correspondente ao total de combustível gasto e o programa deve dividi-los e retornar o consumo médio deste veículo.

As histórias de usuário envolvendo a *URI Online Judge* são um pouco diferentes. No problema que estamos modelando, o *URI Online Judge* tem o papel de avaliar se a solução sugerida pelo usuário está de acordo com as especificações do problema e se as saídas de teste estão certas e é exatamente isso que as histórias de usuário acima estão especificando.

1. **Descrição de critérios de aceitação**

Os critérios de aceitação verificam se o programa está sendo implementado de acordo com as histórias de usuário. É utilizando esses critérios de aceitação que será analisado se uma entrega foi aceita ou recusada [Site Campus].

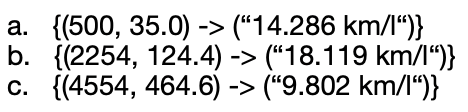


**Figura 2: Critérios de aceitação para o problema de consumo.**

No problema do consumo médio, foi identificado sete critérios de aceitação que precisam ser seguidos para que o programa tenha o comportamento que é esperado dele. E os critérios são: o programa deve ter três variáveis que são elas, respectivamente, a distância total percorrida, o total de combustível gasto e o consumo médio calculado após e todas essas variáveis devem ser dos tipos pré-determinados; após o resultado do consumo médio deve ser impresso a *String* "km/l"; o programa deve passar pelos testes do sistema. Caso um desses critérios não for atingido, a entrega será recusada.

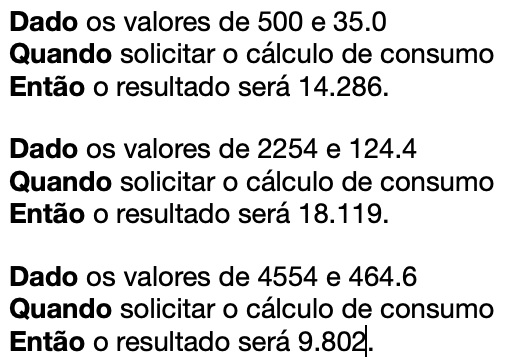
1. **Testes e Linguagem Gherkin**

Os testes e a linguagem Gherkin tem o intuito de descrever como será o comportamento do *software*. Ou seja, é simulado entradas e então é mostrada a saída que seria a certa caso o *software* esteja funcionando corretamente.



**Figura 3: Testes para o problema de consumo.**

Acima, temos três casos de teste do problema de consumo médio. São estes três casos de teste que foram disponibilizados na plataforma *URI Online Judge.* De acordo com a plataforma, dado duas entradas específicas, o único resultado correto é o que está após a vírgula.

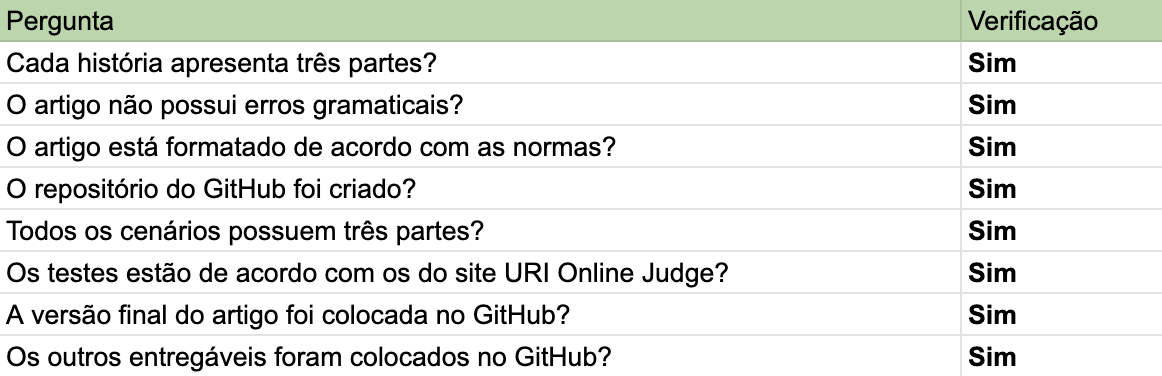


**Figura 4: Linguagem Gherkin para o problema consumo.**

Outra forma de exibir testes é utilizando a linguagem Gherkin. A linguagem Gherkin é uma forma mais textual de descrever o comportamento do sistema, e talvez mais fácil para algumas pessoas entenderem, mas ele funciona de um jeito semelhante aos casos de teste. Construímos a linguagem Gherkin em três partes: Dado, Quando e Então. No caso acima, depois do Dado é colocado as entradas que serão os parâmetros desse teste, depois de Quando é descrito o comportamento do *software* e depois do Então é colocado o resultado esperado.

1. **Lista de verificação**

Para garantir que o relatório estivesse completo e sem erros foi criado uma lista de verificação que pode ser verificada abaixo:

****

**Figura 5: Lista de verificação do trabalho.**

A lista de verificação é bem abrangente pois o trabalho não consiste apenas em um tipo de entrega. Por causa disso, foi verificado tanto a construção do artigo como se ele está de acordo com as normas e sem erros gramaticais, assim como se os modelos em texto foram feitos corretamente e se o repositório do *GitHub* foi criado com a intenção de que o trabalho estivesse completo e sem erros.

1. **Conclusão**

A criação de modelos em texto pode ser mais fácil para pessoas que não tem muito contato com termos de programação e coisas mais técnicas já que conseguimos descrever os requisitos do *software* e as suas funcionalidades de uma forma textual, utilizando a linguagem do dia a dia e isso torna o processo inicial de desenvolvimento de *software* mais acessível para todas as partes.

Com a criação de modelos em texto podemos ter uma visão geral de como seria o comportamento ideal para o sistema e assim antes mesmo de pensar em como ele será desenvolvido programaticamente e com isso, podemos otimizar o nosso trabalho na hora do desenvolvimento pois olhando para os modelos em texto do *software* é possível ter uma ideia de qual a melhor forma de desenvolvê-lo.

**Referências**

URI Online Judge. 1014 - Consumo. https://www.urionlinejudge.com.br/ judge/pt/problems/view/1014. [Online; accessed 27-August-2019].

Atlassian Agile Coach. Histórias de usuário com exemplos e templates. https://

www.atlassian.com/br/agile/project-management/user-stories. [Online; accessed 27-September-2019].

Site Campus. Scrum: Os critérios de aceitação. https://sitecampus.com.br

/scrum-os-criterios-de-aceitacao/. [Online; accessed 27-September-2019].