

Artefatos de modelos representado por texto

Guilherme Rangel¹

¹Escola Politécnica - Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS

Porto Alegre – RS – Brazil

guilherme.rangel@acad.pucrs.br

Resumo. *O presente relatório visa apresentar como o modelo inclusivo sendo ele histórias de usuário e o modelo independente de plataforma pode ajudar no processo de desenvolvimento de software também será apresentado critérios de aceitação e testes de cenário utilizando como base a linguagem Gherkin.*

1. Informacao geral

Este documento tem o intuito apresentar a construção de dois modelos o modelo *inclusivo* que ira ser representado por **histórias de usuário** e o modelo idempedente de plataforma para isso vamo utilizar a **linguagem Gherkin** para ajudar a compreender melhor o escopo que deve ser desenvolvido, critérios de aceitacao e testes de cenarios também serao abordados. Também será apresentado um diagrama de casos de uso para ajudar na compreensão do problema.

2. O Problema

O problema a ser resolvido pode ser interpretado através do enunciado a seguir "Leia três números inteiros e classifique-os em ordem crescente. Em seguida, imprima esses valores em ordem crescente, uma linha em branco e a seguir os valores na seqüência conforme foram lidos." esse problema pode ser encontrado em [urionlinejudge](#) problema 1042.

Para resolver esse problemas algumas etapas foram feitas inicialmente foi modelado um diagrama de casos de uso para representar os atores do sistema, após foi modelado um diagrama de classes de maneira abstrata para representar o sistema e servir de artefato e poder apoiar quem quiser saber como modelar um sistema para esse problema.

Além disso também foi utilizado no *modelo inclusivo* através de histórias de usuários para melhorar a compreensão do problema, a linguagem Gherkin também foi utilizada

com a criação de cenários para ajudar no entendimento.

3. Diagrama

Para o modelo inclusivo, o modelo de caso de uso representa os atores do sistema conforme figura 1, esse diagrama é representado por um usuário que tem uma generalização para um iniciante que é um usuário qualquer com um nível baixo de aderência ao sistema e outra entidade chamada URIJudge que é o ator que pode executar o sistema de forma automatizada, a ação de ordenar números é um método na qual recebe os valores e devolve os mesmos ordenado de forma ascendente .

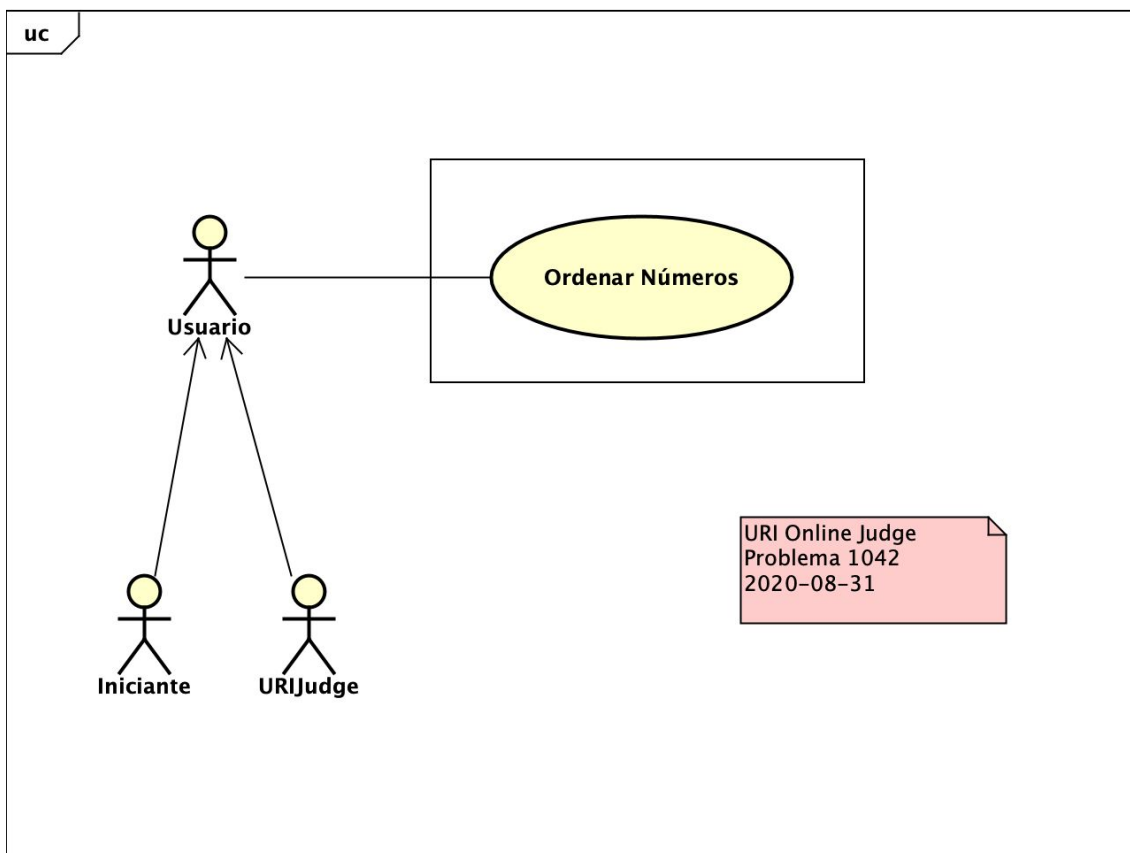


Figura 1. Diagrama de casos de Uso

Para representar esse modelo inclusivo como uma história de usuário basta seguir uma estrutura "fixa" onde é iniciado com a identificação do perfil do usuário que vai utilizar a funcionalidade, logo após vem a responsabilidade do que a funcionalidade em si se propõe a resolver e vai ser desenvolvida e por fim a principal motivação da funcionalidade incluindo principalmente o valor que trás para o negócio.

Representacao da historia de usuario:

Sendo um usuário do sistema **Quero** poder inserir números inteiros e apresentá-los em ordem crescente **Para Que** eu possa localizar um determinado número com mais facilidade.

A linguagem Gherkin, esse modelo é utilizado tendo três principais palavras chaves, **Dado que** serve para indicar o cenário atual , **Quando** serve para indicar a ação do usuário e o **Então** para indicar como o software vai reagir, ainda podemos ter o "E" e o "Ou" com eles os cenários de testes ficam ainda mais ricos e mais detalhados, esses cenários também são conhecidos como critérios de aceitação e eles podem ser ligados diretamente a como o seu software ir a ser testado, alguns cenário para o problema descrito seriam:

Cenário 1: Insercao dos valores

Dado que: O sistema só aceita números pertencentes aos conjuntos dos Inteiros

Quando: for digitado um numero que nao esteja contido nos conjuntos dos Inteiros

Então: O sistema não deve permitir a inclusão do mesmo

E deve avisar o usuário que o sistema deve ser preenchido em formato de inteiro.

Cenário 2: Exibição das informações

Dado que: O sistema deve classificar números inteiros inseridos aleatoriamente em ordem crescente e exibir os números digitados na ordem que foram inseridos no sistema.

E já foi digitado os números corretamente, números contidos no conjuntos dos números inteiros.

Quando: Eu executar o programa

Então: deve ser exibido os números em ordem crescente seguido de uma linha em branco e a seguir os números digitados na ordem que foram inseridos.

4.Consideracoes finais

Em suma percebe-se que fazer histórias de usuário e criar cenários de testes agregam valor sempre que se deseja ter mais compreensão executar os mesmo também é uma forma de auto documentar o projeto como disse o **Rafael Helm** autor do livro **Histórias de Usuário** "[Qualidade de software começa na especificação](#)". Para a execução desse trabalho foi utilizado um lista de verificação (checklist) com ela foi possível ter uma melhor assertividade na hora da execução das tarefas pois a mesma serviu de guia e me ajudou a validar alguns pontos que poderiam ter passado despercebido, todos os

artefatos podem ser encontrados no repositório a seguir
<https://github.com/guilhermeRangel/Classifica-oSimples> .

Referencias

Rafael Helm e Daniel Wildt “Histórias de Usuário”, Por que e como escrever requisitos de forma ágil? 3. edicao ano 2014

BDD desenvolvimento orientado a comportamento. Cucumber, disponivel em
<<https://cucumber.io/docs/bdd/>> Acesso em 30 de set de 2020.