Trabalho 2 - Engenharia de Software Orientada a Modelos

Pedro Chem e Rafael Almeida de Bem

29 de setembro de 2020

Sumário

Introdução

O presente relatório tem como objetivo documentar o processo de modelagem de dados para um enunciado de programação. Este projeto foi requisitado pelo Profesor Marco Aurélio Souza Mangan para a disciplina de Engenharia de Software Orientada a Modelos no curso de Ciência da Computação, Escola Politécnica, PUC-RS. Todos as entregas deste projeto estão disponíveis individualmente no repositório compartilhado: `https://github.com/Pedrochem/Modelagem-UML2`.

O trabalho envolve dois modelos. O primeiro é um modelo inclusivo, mais especificamente histórias de usuário. O segundo é um modelo independente de plataforma, respectivamente, a correspondente descrição dos critérios de aceitação, testes e cenários em linguagem Gherkin.

Modelos

Para a descrição dos modelos à seguir, será levado em conta o seguinte enunciado de programação: "Write an algorithm that reads two floating values (x and y), which should represent the coordinates of a point in a plane. Next, determine which quadrant the point belongs, or if you are over one of the Cartesian axes or the origin (x = y = 0)."Retirado do site URI Judge[3], com código: 1041.

2.1 Modelo Inclusivo

Um modelo inclusivo é um modelo que foi esboçado rapidamente utilizando ferramentas simples como um quadro branco ou cartões [1]. Um exemplo desse modelo são as histórias de usuário.

2.1.1 Histórias de usuário

- Eu como um usuário, gostaria de saber em qual quadrante está localizado a coordenada de um ponto para obter o resultado correto.
- Eu como usuário, gostaria de saber quando um ponto está na origem para não confundir com os quadrantes ou eixos.
- Eu como usuário, gostaria de saber quando um ponto está em um dos eixos, para não confundir com os quadrantes ou origem.

2.2 Modelo Independente de Plataforma

Um modelo independente de plataforma, ou PIM (*Platform Independent Model*), é uma descrição das funcionalidades de um sistema independentemente das características de implementação em plataformas específicas [2].

2.2.1 Descrição

O modelo a ser implementado, deve ser capaz de informar, a partir da coordenada de um ponto, em qual quadrante este ponto está. Se o ponto estiver na origem, ou em um dos eixos, isso deve ser informado ao usuário. O resultado deve sempre vir em forma de uma mensagem.

2.2.2 Critérios de aceitação

- Se o ponto está na origem, então o resultado deve ser "Origem".
- Se o ponto está no eixo X, então o resultado deve ser "Eixo X".
- Se o ponto está no eixo Y, então o resultado deve ser "Eixo Y".

- Se o ponto não está na origem, e nem em um dos eixos, então o resultado deve ser o uma mensagem que identifica o quadrante em que o ponto está, como por exemplo "Q4".

2.2.3 Testes e cenários

Dado os valores de :x e :y, quando solicitar o quadrante, então o resultado será :res

x	y	res
0	0	"Origem"
0	10	"Eixo X"
10	0	"Eixo Y"
10	10	Q1
4.5	-2.2	Q4

Lista de Verificação

Para contribuir com a qualidade deste trabalho, foi elaborado uma lista de validação. A lista é composta por 7 itens que tem o propósito de validar que os modelos e demais outras entregas estão corretas.

1. O trabalho apresenta corretamente o modelo inclusivo. ✓
2. O trabalho apresenta corretamente o modelo independente de plataforma. ✓
3. As histórias de usuário foram implementadas seguindo o padrão "Eu como x, gostaria de x, para x". ✓
4. O trabalho apresenta o enunciado de programação. ✓
5. Todos os cenários tem uma cláusula "então"✓
6. O trabalho apresenta referências às fontes consultadas. ✓
7. O trabalho está disponível em um repositório. ✓

Conclusão

Neste trabalho foram apresentados dois modelos com o propósito de modelar um enunciado de programação retirado do site URI Online Judge [3]. Além disso, foi elaborado uma lista de verificação para controlar a qualidade do trabalho.

Referências Bibliográficas

[1] Scott W. Ambler. Approaches to agile model driven development (amdd). `http://agilemodeling.com/essays/amddApproaches.htm`. Acessado: 29-08-2020. 3

[2] Leopoldo Vinicio Venegas Loor. Arquitectura manejada por modelos. *Revista San Gregorio*, 2014. 3

[3] Neilor Tonin. Uri online judge. `https://www.urionlinejudge.com.br`, Agosto 2020. 3, 6