Universidade Federal de Ouro Preto - UFOP
Instituto de Ciências Exatas e Biológicas - ICEB
Departamento de Computação - DECOM

Ciência da Computação - Bacharelado

Trabalho Individual

Engenharia de Software I

Victor Emmanuel Susko Guimarães

Ouro Preto
Dezembro de 2024

Conteúdo

I Resumo

Este trabalho constitui um documento do projeto de uma API para o trabalho prático individual da disciplina de Engenharia de Software I. Consta aqui o projeto das funcionalidades (casos de uso) da API, dos testes em pseudocódigo, bem como um diagrama UML representativo das classes do software, que posteriormente será implementado em código.

O objetivo do projeto é construir uma API que consiga realizar simulações de eventos em sistemas ao longo do tempo, para que seja possível prever diferentes cenários condicionados a uma situação inicial do sistema, podendo este ou não sofrer interferências de fatores externos.

Dessa forma, para este projeto, existem 3 principais componentes da API: o sistema, o fluxo e o modelo.

- O **Sistema** constitui um agente que está sujeito a mudanças ao longo do tempo. Ele pode representar seres vivos, conjuntos, locais, cidades, ou qualquer entidade que possa sofrer algum tipo de alteração de estado. Ex: o sistema poderia ser um conjunto de galinhas, que começa com uma população inicial e pode sofrer ação de predadores, alimentos disponíveis, espaço, etc. É possível que um sistema contenha outros sistemas.

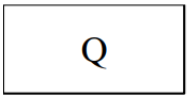
- O **Fluxo** é o componente que promove uma alteração no estado de um sistema, e o usuário deve definir, em cada fluxo, qual a sua expressão matemática associada. Como o fluxo possui um sentido, ou seja, de onde os recursos vêm, e para onde vão, a expressão matemática em conjunto com o sentido do fluxo definirão se um sistema perde ou ganha recursos ao longo do tempo. Vale ressaltar que é possível que um fluxo não esteja necessariamente associado a um sistema, ou seja, a API deve permitir que a simulação aconteça mesmo que não haja sentido lógico de existir um fluxo sozinho.

- O **Modelo** representa um conjunto de sistemas e de fluxos, cabendo a ele realizar o gerenciamento de uma aplicação inteira. Por isso, ele possui o controle do tempo decorrido durante a simulação, do lapso temporal, das ordens de execução, bem como adicionar ou remover fluxos e sistemas.

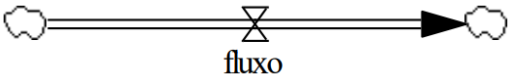
II Casos de uso

Definirão-se, aqui, quais são os possíveis usos da API, ou seja, quais as possíveis interações entre os componentes listados. São eles:

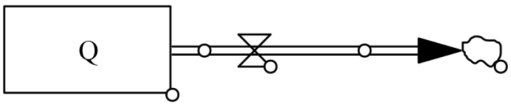
Caso 1: Sistema isolado



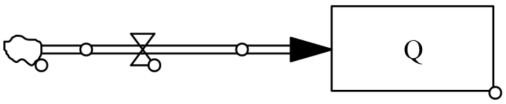
Caso 2: Fluxo isolado



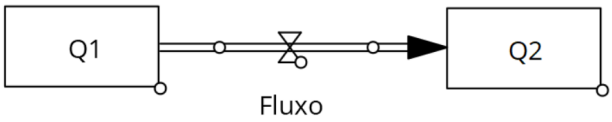
Caso 3: Fluxo com origem em um sistema, mas sem destino



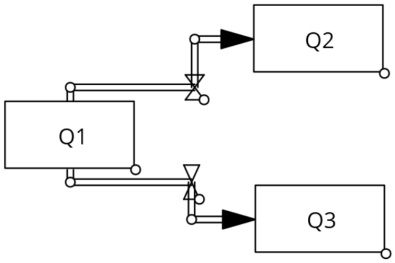
Caso 4: Fluxo com destino em um sistema, mas sem origem



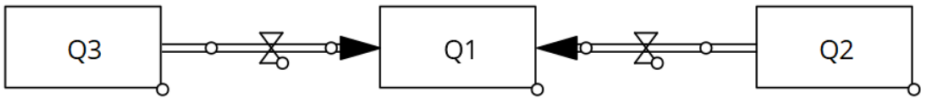
Caso 5: Fluxo conectando sistemas



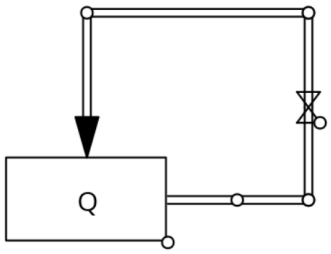
Caso 6: Múltiplos fluxos saindo de um sistema



Caso 7: Múltiplos fluxos entrando em um sistema



Caso 8: Sistema conectado em si próprio



III Cenários de teste

Caso 1:

```
//creating model and system
Model model();
System q();

//adding the system to the model
model.add(q);
```

Caso 2:

```
//creating model and system
Model model();
Flow f();

//adding flow to the model
model.add(f);
```

Caso 3:

```
//creating model, flow and system
Model model();
Flow f();
System q();

//adding flow and system to the model
model.add(q);
model.add(f);

//setting flow source
f.setSource(q);
```

Caso 4:

```
//creating model, flow and system
Model model();
Flow f();
System q();

//adding flow and system to the model
model.add(q);
```

```
model.add(f);

//setting flow target
f.setTarget(q);
```

Caso 5:

```
//creating model, flow and systems
Model model();
Flow f();
System q1();
System q2();

//adding flow and both systems
model.add(q1);
model.add(q2);
model.add(f);

//setting flow target and source
f.setTarget(q2);
f.setSource(q1);
```

Caso 6:

```
//creating model, flows and systems
Model model();
Flow f1();
Flow f2();
System q1();
System q2();
System q3();

//adding existing elements to the model
model.add(f1);
model.add(f2);
model.add(q1);
model.add(q2);
model.add(q3);

//setting flow1 and flow2 sources and targets
f1.setSource(q1);
f2.setSource(q1);
f1.setTarget(q2);
f2.setTarget(q3);
```

Caso 7:

```
//creating model, flows and systems
Model model();
Flow f1();
Flow f2();
System q1();
System q2();
System q3();

//adding existing elements to the model
model.add(f1);
model.add(f2);
model.add(q1);
model.add(q2);
model.add(q3);

//setting flow1 and flow2 sources and targets
f1.setSource(q3);
f2.setSource(q2);
f1.setTarget(q1);
f2.setTarget(q1);
```

Caso 8:

```
//creating model, flow and system
Model model();
Flow f();
System q();

//adding flow and system to the model
model.add(f);
model.add(q);

//setting flow target and source
f.setSource(q);
f.setTarget(q);
```

IV Diagrama UML