Engenharia de Software I - Sprint 1

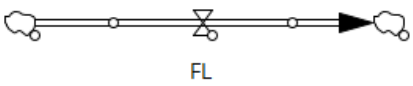
Aluno: Yasmin Altino Araújo
Professor: Tiago Garcia de Senna Carneiro

1 Funcionalidades

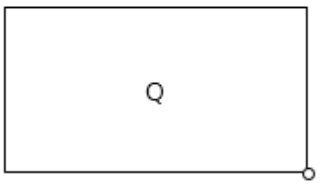
- Simulação de Fluxos Dinâmicos.
- Configuração de modelos contendo vários sistemas e fluxos.
- Execução das simulações em um número definido de iterações.
- Personalização de fluxos segundo equações matemáticas.

2 Casos de Uso

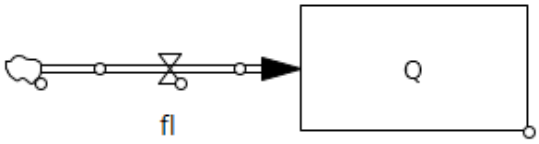
1. **Um fluxo sem origem e sem destino:**



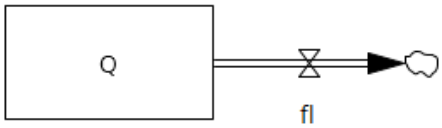
2. **Um sistema isolado:**



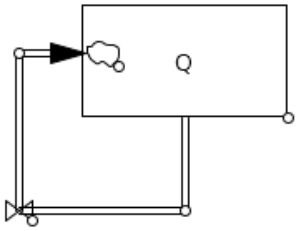
3. **Fluxo entrando em um sistema:**



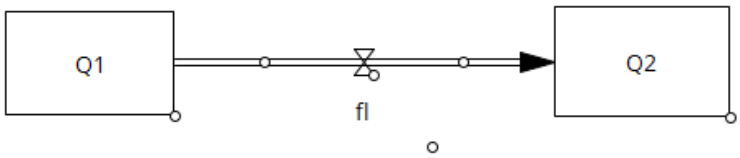
4. **Fluxo saindo de um sistema:**



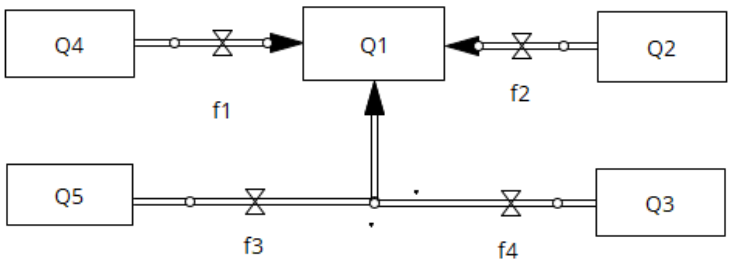
5. **Fluxo saindo e entrando no mesmo sistema:**



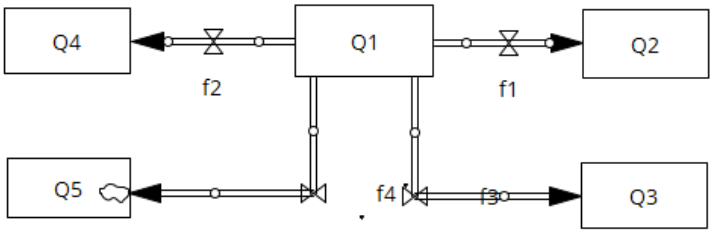
6. **Dois sistemas ligados por um fluxo:**



7. **Vários fluxos com equações logísticas entrando em um sistema:**



8. **Vários fluxos com equações exponenciais saindo de um sistema:**



9. **Definição do Fluxo Exponencial**
10. **Definição do Fluxo Logístico**

2.1 Cenários de Teste

1.

```
        Model model;
        Flow* f1 = new Flow ("f1");
        model.add(f1);

```

2.

```
        Model model;
        System* Q  = new System("Q", 100);
        model.add(Q);

```

3.

```
        Model model;
        Flow* f1 = new Flow("f1");
        System* Q = new System ("Q", 0);
        f1->setTarget(Q);
        model.add(f1);
        model.add(Q);

```

4.

```
        Model model;
        Flow* f1 = new Flow("f1");
        System* Q = new System("Q", 100);
        f1->setSource(Q);
        model.add(f1);
        model.add(Q);

```

5.

```
    Model model;
    Flow* f1  = new Flow("f1");
    System* Q = new System("Q", 100);
    f1->connect(Q, Q);
    model.add(f1);
    model.add(Q);

```

6.

```
    Model model;
    Flow* f1 = new Flow("f1");
    System* Q1 = new System("Q1", 100);
    System* Q2 = new System("Q2", 0);
    f1->connect(Q1, Q2);
    model.add(f1);
    model.add(Q1);
    model.add(Q2);

```

7.

```
    Model model;
    Logistic *flow1 = new Logistic("FlowLog1");
    Logistic *flow2 = new Logistic("FlowLog2");
    Logistic *flow3 = new Logistic("FlowLog3");
    Logistic *flow4 = new Logistic("FlowLog4");

    System* Q1 = new System("Q1", 100);
    System* Q2 = new System("Q2", 0);
    System* Q3 = new System("Q3", 50);
    System* Q4 = new System("Q4", 0);
    System* Q5 = new System("Q5", 10);

    f1->connect(Q4,Q1);
    f2->connect(Q2,Q1);
    f3->connect(Q5,Q1);
    f4->connect(Q3,Q1);

    model.add(f1);
    model.add(f2);
    model.add(f3);
    model.add(f4);
```

```
    model.add(Q1);
    model.add(Q2);
    model.add(Q3);
    model.add(Q4);
    model.add(Q5);
    model.run(100);

```

8.

```
    Model model;
    Exponential *flow1 = new Exponential("FlowExp1");
    Exponential *flow2 = new Exponential("FlowExp2");
    Exponential *flow3 = new Exponential("FlowExp3");
    Exponential *flow4 = new Exponential("FlowExp4");

    System* Q1 = new System("Q1", 100);
    System* Q2 = new System("Q2", 0);
    System* Q3 = new System("Q3", 50);
    System* Q4 = new System("Q4", 0);
    System* Q5 = new System("Q5", 10);

    flow1->connect(Q1,Q2);
    flow2->connect(Q1,Q4);
    flow3->connect(Q1,Q3);
    flow4->connect(Q1,Q5);

    model.add(f1);
    model.add(f2);
    model.add(f3);
    model.add(f4);
    model.add(Q1);
    model.add(Q2);
    model.add(Q3);
    model.add(Q4);
    model.add(Q5);
    model.run(100);

```

9.

```
    class Exponencial : public Flow

    float Exponencial::execute()
    {
        if (getSource() != nullptr)
        {
            return getSource()->getValue() * 0.01; // Exemplo de calculo exponencial
        }
        return 0.0;
    }

```

10.

```
    class Logistic : public Flow

    float Logistic::execute()
    {
        if (getTarget() != nullptr)
        {
            return getTarget()->getValue() * 0.01 * (1 - getTarget()->getValue() / pmax);
    // Exemplo de calculo logistico
        }
        return 0.0;
    }

```

3 UML

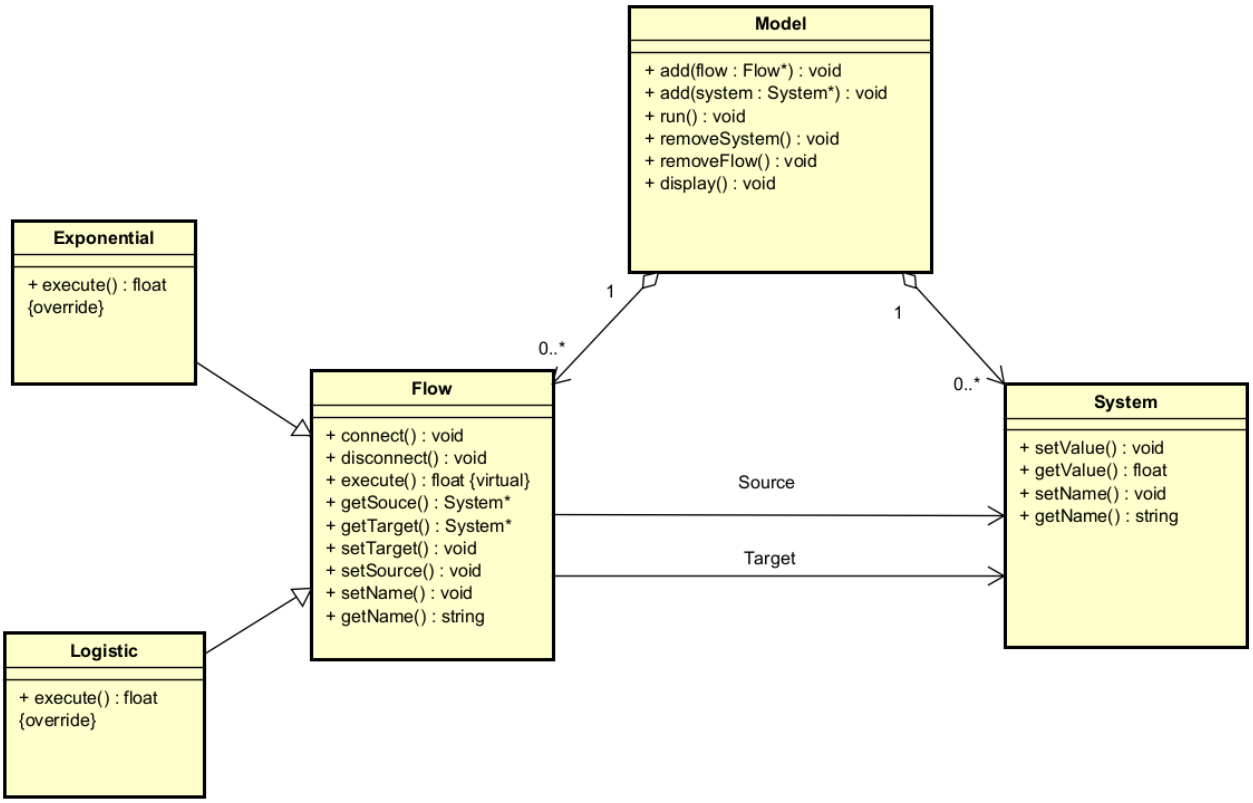


Figure 1: Diagrama UML