

Sprint 1



Aluno: Fábio Henrique Alves Fernandes

Matrícula: 19.1.4128

Professor: Tiago Garcia de Senna Carneiro

Observação: O trabalho foi feito através de discussões entre colegas de turma, de forma a ajudar no entendimento do problema proposto. Desta forma, pode ser que haja semelhanças entre alguns trabalhos.

Funcionalidades

A API criada é capaz de executar sistemas dinâmicos baseados na Teoria Geral de Sistemas, a partir da criação de sistemas. Ela funciona a partir de associação de fórmulas matemáticas a fluxos que representam variações de atributos entre diferentes sistemas.

Sistema: representa um estoque de qualquer coisa;

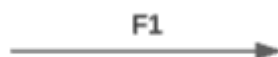
Fluxo: representa uma variação de qualquer coisa, associada a uma fórmula algébrica.

Casos de uso:

Antes de tudo, precisamos projetar a estrutura da API, estudando os casos e analisando o comportamento em diferentes situações.

Caso 1

Fluxo isolado, sem entrada nem saída.



```
Model m;  
Flow f1("F1", NULL, NULL);
```

```
m.add(f1);  
m.execute();
```

Caso 2

Sistema isolado, sem fluxos nem na entrada, nem na saída.



```
Model m;  
System s1("S1", 0);  
m.add(s1);  
m.execute();
```

Caso 3

Sistema que tem um fluxo em sua entrada.



```
Model m;  
System s1("S1", 0);  
Flow f1("F1", NULL, &f1);  
m.add(s1);  
m.add(f1);  
m.execute();
```

Caso 4

Sistema que tem um fluxo em sua saída.

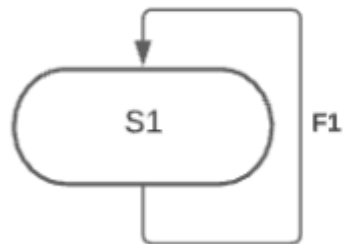


```
Model m;  
System s1("S1", 0);  
Flow f1("F1", &s1, NULL);  
m.add(s1);
```

```
m.add(f1);  
m.execute();
```

Caso 5

Sistema cíclico, ou seja, tem o mesmo fluxo tanto para entrada quanto para saída.



```
Model m;  
System s1("S1", 0);  
Flow f1("F1", &s1, &s1);  
m.add(s1);  
m.add(f1);  
m.execute();
```

Caso 6

Sistema que possui um ciclo de entrada e um ciclo de saída.



```
Model m;  
System s1("S1", 0);  
Flow f1("F1", NULL, &s1);  
Flow f2("F2", &s1, NULL);  
m.add(s1);  
m.add(f1);  
m.add(f2);  
m.execute();
```

Caso 7

Fluxo que tem um sistema na entrada e um outro sistema na saída.



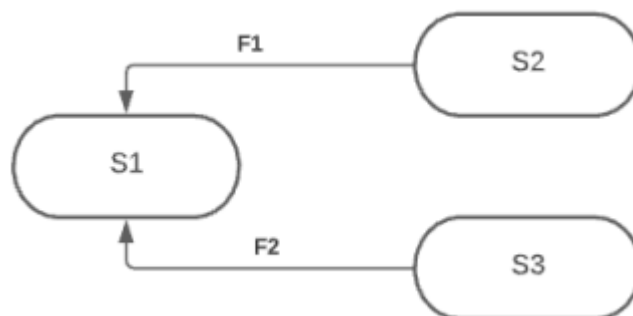
```

Model m;
System s1("S1", 0);
System s2("S2", 0);
Flow f1("F1", &s1, &s2);
m.add(s1);
m.add(s2);
m.add(f1);
m.execute();

```

Caso 8

Sistema conectado a dois ou mais sistemas a partir de fluxos de entrada.



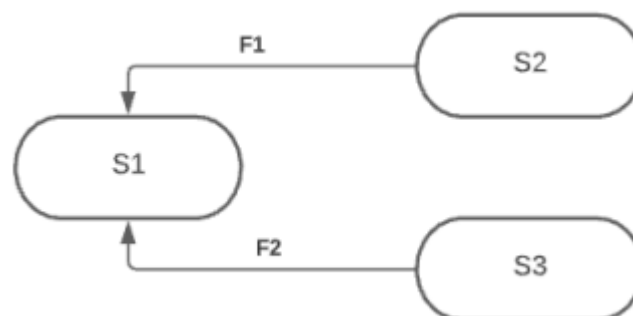
```

Model m;
System s1("S1",0);
System s2("S2",0);
System s3("S3",0);
Flow f1("F1",&s2,&s1);
Flow f2("F2",&s3,&s1);
m.add(s1);
m.add(s2);
m.add(s3);
m.add(f1);
m.add(f2);
m.execute();

```

Caso 9

Sistema conectado a dois ou mais sistemas a partir de fluxos de saída.



```

Model m;
System s1("S1",0);
System s2("S2",0);
System s3("S3",0);
Flow f1("F1",&s1,&s2);
Flow f2("F2",&s1,&s3);
m.add(s1);
m.add(s2);
m.add(s3);
m.add(f1);
m.add(f2);
m.execute();

```

Diagrama UML

