

**UNIVERSIDADE FEDERAL DE OURO PRETO – UFOP**

**Ciência da Computação**



**ENGENHARIA DE SOFTWARE**

**SPRINT 1 - Projeto da API do simulador de sistemas dinâmicos**

Nathann Zini dos Reis

19.2.4007

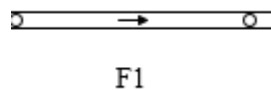
**Ouro Preto**

**2021**

## Casos de Uso

Antes de começar a projetar a estrutura da API, é necessário realizar os estudos de caso e analisar como ela deveria se comportar em diferentes situações. Para isso foram pensadas algumas situações, oito no total, que estão listadas abaixo.

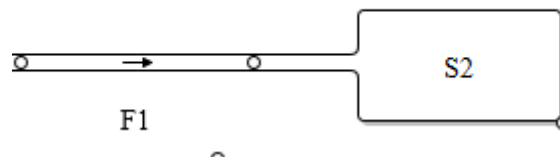
**Caso 1:** Apenas um fluxo, sem entrada ou saída



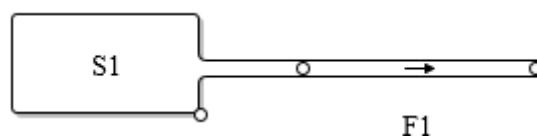
**Caso 2:** Apenas um sistema, sem fluxo de entrada ou saída



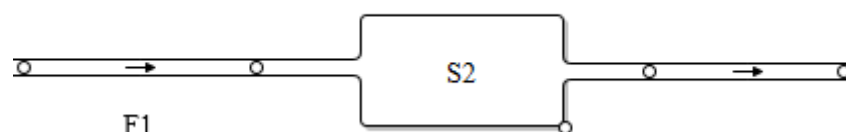
**Caso 3:** Um sistema com um fluxo de entrada



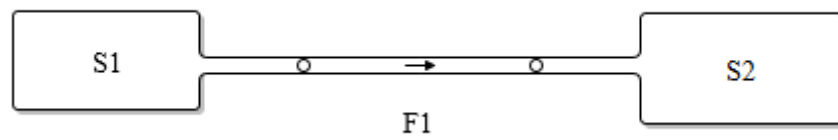
**Caso 4:** Um sistema com fluxo de saída



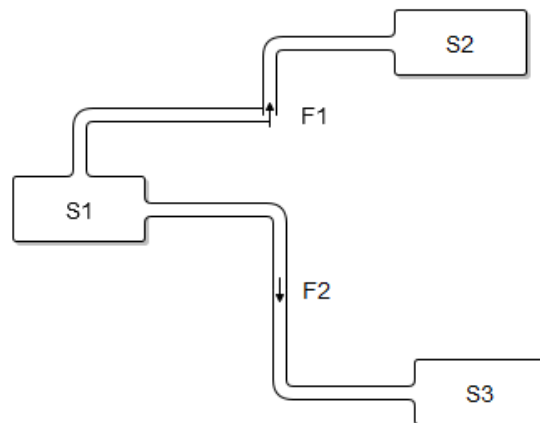
**Caso 5:** Um sistema com fluxo de entrada e de saída



**Caso 6:** Um sistema conectado a outra sistema por fluxo de entrada



**Caso 7:** Um sistema conectado a outros dois sistema por fluxo de saída



## Codificação dos casos de uso

Nessa parte é pensado como seria o uso da API através de códigos escritos pelo usuário. Para cada um dos casos de teste apresentados anteriormente foi imaginado como seria a implementação deles em código.

```
/*teste do Caso de uso 1
* Fluxo sem sistema
*/

Model m;
Flow f1("F1", NULL, NULL);
m.addFluxo(f1);
m.run();

/*teste do Caso de uso 2
* sistema entrada e saída
*/

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

/*teste do Caso de uso 3
* sistema com fluxo de entrada
*/

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

```
/*teste do Caso de uso 4
 * sistema com fluxo de saída
 */
```

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

```
/*teste do Caso de uso 5
 * sistema com fluxo de entrada e 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.run();
```

```
/*teste do Caso de uso 6
 * sistema conectado a outro sistema por fluxo de entrada
 */
```

```
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.run();
```

```
/*teste do Caso de uso 7
 * sistema conectado a outros dois sistemas por fluxo 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.run();
```

```
/*teste do Caso de uso Disponibilizado pelo professor  
* Sistema exponencial  
*/
```

```
Model m;  
System pop1("POP1", 100);  
System pop2("POP2", 0);  
Flow f1("F1", &pop1, &pop2);  
f1.setEquacao(0.01*pop1);  
m.add(pop1);  
m.add(pop2);  
m.add(f1);  
m.run();
```

```
/*teste do Caso de uso Disponibilizado pelo professor  
* Sistema lógico  
*/
```

```
Model m;  
System pop1("POP1", 100);  
System pop2("POP2", 10);  
Flow f1("F1", &pop1, &pop2);  
f1.setEquacao(0.01*pop2*(1-pop2/70));  
m.add(pop1);  
m.add(pop2);  
m.add(f1);  
m.run();
```

```
/*teste do Caso de uso Disponibilizado pelo professor
* Sistema Completo*/
Model m;
System q1("Q1", 100);
System q2("Q2", 0);
System q3("Q3", 100);
System q4("Q4", 0);
System q5("Q5", 0);
Flow f("F", &q1, &q2);
Flow g("G", &q1, &q3);
Flow r("R", &q2, &q5);
Flow t("T", &q2, &q3);
Flow u("U", &q3, &q4);
Flow v("V", &q4, &q1);
f.setEquacao(0.01*q1);
g.setEquacao(0.01*q1);
r.setEquacao(0.01*q2);
t.setEquacao(0.01*q2);
u.setEquacao(0.01*q3);
v.setEquacao(0.01*q4);
m.add(q1);
m.add(q2);
m.add(q3);
m.add(q4);
m.add(q5);
m.add(f);
m.add(g);
m.add(r);
m.add(t); m.add(u); m.add(v);
m.run();
```

## Diagrama UML

Por fim, foi criado o diagrama UML contendo os métodos, classes e relações da API de acordo com o que foi ministrado nas aulas da disciplina. Nesse diagrama é apresentado como a API deveria funcionar e tem como finalidade guiar o desenvolvimento do projeto.

