

# Trabalho 1

## INE5425 - Modelagem e Simulação

Bruno Marques do Nascimento\*

Gustavo Figueira Olegário †

Florianópolis, 26 de Outubro de 2017

### Introdução

Este relatório possui o objetivo de explicar o que foi desenvolvido para o Trabalho 1 da disciplina de Modelagem e Simulação, ministrada pelo professor Paulo Jose de Freitas Filho na Universidade Federal de Santa Catarina(UFSC), campus Florianópolis. O propósito do trabalho é implementar um programa que permita a modelagem e a simulação de um sistema com duas filas e dois servidores, no qual as entidades chegam ao sistema e entram em uma das filas para serem atendidas por um servidor.

Será abordado o funcionamento do programa, assim como as principais classes e funções responsáveis pela execução.

A linguagem de programação *C++* foi a escolhida para a implementação do trabalho e para o desenvolvimento da interface gráfica foi utilizado o framework *Qt*.

### 1 Estrutura da implementação

O problema foi estruturado em classes que definem os recursos utilizados, e uma classe de controle que será responsável pelo fluxo de execução da simulação e extração de dados. Abaixo temos as definições de cada classe e seus principais métodos e atributos.

- **Entity:**

Esta classe é responsável por representar as entidades que percorrem o sistema, seu código-fonte se encontra nos arquivos *entity.h* e *entity.cpp*.

- **Server:**

Esta classe é responsável por representar os servidores do sistema, que serão responsáveis pelo controle das filas de espera, computação das entidades, além de armazenar variáveis de controle referente às falhas e disponibilidade do servidor, seu código-fonte está presente em *server.h* e *server.cpp*.

---

\*brunomn95@gmail.com

†gustavo-olegario@hotmail.com

- **Event:**

Classe que representa os eventos que serão tratados pelo controlador, esses eventos são: chegada de uma nova entidade, chegada da entidade do servidor, evento de tempo entre falhas, evento de falha do servidor e saída de uma entidade do sistema, respectivamente representados pelo enum presente na classe:

```
enum Type{
    ENTITY_ARRIVAL,
    ARRIVAL_AT_SERVER,
    TEF_SERVER,
    TF_SERVER,
    ENTITY_DISPOSE
};
```

- **RandomFunctionsData:**

Esta classe é utilizada para a geração dos números aleatórios conforme definido pelo usuário, podendo ser: Constante, Normal, Uniforme, Triangular e Exponencial. Seu código-fonte encontra-se nos arquivos *random\_functions\_data.h* e *random\_functions\_data.cpp*.

- **SystemController:**

Esta é a classe mais importante do programa, ela é responsável por controlar o calendário de eventos, gerando eventos futuros e executando os eventos do tempo atual. É nela que são coletados os dados para futuras aferições, coleta dos parâmetros do usuário, etc. Segue abaixo um resumo das funções mais importantes.

- ***config\_initial\_state:***

Esta função é responsável por configurar o estado inicial da simulação, reinicia todas as variáveis necessárias, reinicia os servidores, limpa a lista de entidades, o calendário de evento, e lê os valores e funções definidos pelo usuário.

- ***create\_initial\_events:***

Cria os eventos responsáveis por continuarem gerando os próximos eventos até o final do tempo de simulação.

- ***process\_events:***

É a rotina que irá ler os eventos presentes no calendário e os executará conforme seu tipo.

- ***run:***

É o loop de execução da simulação, responsável por chamar a rotina de processamento de evento.

- ***refresh\_data:***

Função responsável por mostrar ao usuário as variáveis do sistema, durante e ao final de sua execução.