Simulador de rede de filas

**Bruno Carvalho da Costa**

# **Como executar?**

O programa foi feito em C# e, para executa-lo, basta executar o comando:

“./SimuladorDeRedeDeFilas <nome\_do\_arquivo\_de\_entrada.txt>”

Ou apenas

“./SimuladorDeRedeDeFilas”, passando o nome do arquivo posteriormente.

# Estrutura do arquivo

O arquivo inteiro está estruturado em TAGs, as quais tem sua própria configuração única. Para criar seu próprio arquivo, basta utilizar a TAG conforme definido abaixo. Deve-se seguir o padrão de “NOME DA TAG” e então sua definição nas linhas abaixo. Note que todos os tempos são *double*, enquanto a quantidade de servidores e capacidade são *integer*.

## TAMANHO\_DA\_FILA\_INFINITA

O tamanho (capacidade) definido para caso uma fila seja infinita.

## FILAS

Conforme o nome indica, aqui se definem as filas. Deve-se seguir o formato:

|  |
| --- |
| **<NOME DA FILA> <QUANTIDADE DE SERVIDORES>** *<CAPACIDADE> <CHEGADA MIN.> <CHEGADA MAX.>* **<ATENDIMENTO MIN> <ATENDIMENTO MAX>** |

É necessário definir no mínimo os campos em negrito, ou seja, somente a capacidade e os tempos de chegada são opcionais.

## PRIMEIRA\_CHEGADA

Define os tempos da primeira chegada em cada respectiva fila.

|  |
| --- |
| **<NOME DA FILA> <PRIMEIRA CHEGADA>** |

## REDE

Define a configuração da rede de filas, ou seja, como se comunicam as filas. Deve-se configurar como:

|  |
| --- |
| **<FILA DE ORIGEM> <FILA DE DESTINO> <PROBABILIDADE>** |

Note que a probabilidade deve estar entre 0.0 e 1.0, visto que a soma das probabilidades não deve passar de 1.0.

## NUMERO\_DE\_ALEATORIOS\_POR\_SEMENTE

Quantidade de números aleatórios para cada execução.

## NUMERO\_DE\_SEMENTES

A quantidade de sementes que serão geradas.

# Simulação

O simulador inicia por fazer o *parse* do arquivo de entrada e gerar a lista de filas e toda sua lógica de origem e destino. Após a configuração da hierarquia de rede, o simulador inicia sua execução, apresentando os resultados ao final.

Há três tipos de eventos na execução: chegada, saída e passagem.

É importante apresentar que cada evento acarreta numa mudança das propriedades da fila diferentemente.

A **chegada** realiza a seguinte operação:

|  |
| --- |
| Ao receber uma chegada, contabiliza o tempo de todas as filas e verifica se há espaço na fila atual.  **Se sim**, aumenta a quantidade de clientes na fila e verifica se há servidores e se deve acontecer um roteamento. Caso sim, agenda uma passagem para a outra fila, se não, agenda uma saída.  **Se não**, contabiliza uma perda para a fila.  Por fim, agenda outra chegada. |

A **saída** realiza uma operação mais simples:

|  |
| --- |
| Ao receber uma saída, contabiliza o tempo de todas as filas e decrementa o número de clientes na fila. Se ainda há clientes aguardando, agenda uma saída. |

Por fim, a **passagem** realiza algo como uma mescla entre a chegada e a saída:

|  |
| --- |
| Contabiliza o tempo das filas e decrementa a quantidade de clientes na fila de origem.  Se há clientes esperando na fila de origem, verifica se deve rotear.  Se sim, agenda uma passagem. Se não, agenda uma saída.  Se há espaço na fila de destino, incrementa o numero de clientes na mesma e faz a mesma validação anterior: verifica se deve rotear e, se sim, agenda uma passagem, se não, agenda uma saída. |