

Amanda de Souza Caetano  
Antônio Angelo Passini Júnior

# Trabalho de Programação

## Parte 2: Implementação do Modelo M/M/2

Professor Carlos Lopes

# 1. Instruções

## 1.1. Download e Execução

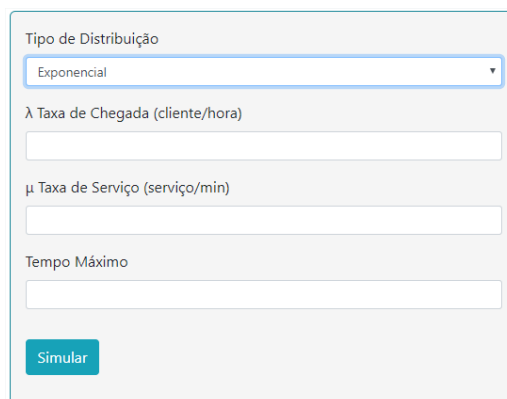
Primeiramente, para ter acesso ao programa desenvolvido, é necessário baixar e descompactar seus arquivos. Tais arquivos estão disponíveis no moodle e também no link [https://drive.google.com/open?id=1l3STQ4rd7g10n1tOgk1UB\\_x7fXHXhZJ9](https://drive.google.com/open?id=1l3STQ4rd7g10n1tOgk1UB_x7fXHXhZJ9).

Uma vez baixado e descompactado os arquivos, basta abrir o arquivo *index.html* em um navegador web<sup>1</sup>, como Google Chrome ou Mozilla Firefox.

## 1.1. Como Simular

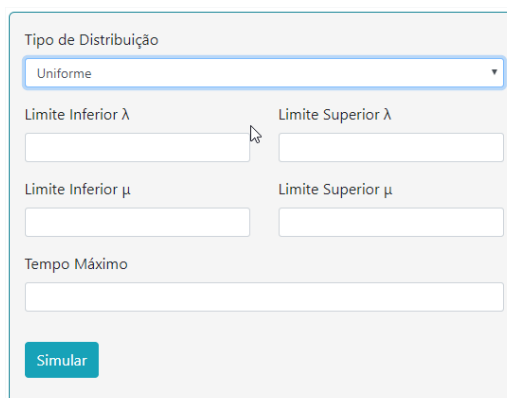
A tela principal<sup>2</sup> do programa contém somente um formulário para obtenção dos dados necessários para a simulação. O primeiro passo é escolher qual tipo de distribuição você pretende utilizar, sendo elas: exponencial, uniforme e discreta.

Uma vez selecionado o tipo, novos campos apareceram, de acordo com a escolha anterior, como mostra as figuras abaixo:



O formulário para distribuição discreta apresenta um menu suspenso no topo rotulado 'Tipo de Distribuição' com a opção 'Exponencial' selecionada. Abaixo, há três campos de entrada: 'λ Taxa de Chegada (cliente/hora)', 'μ Taxa de Serviço (serviço/min)' e 'Tempo Máximo'. Um botão azul com o texto 'Simular' está localizado na base do formulário.

Figura 1: Formulário para distribuição discreta



O formulário para distribuição uniforme apresenta um menu suspenso no topo rotulado 'Tipo de Distribuição' com a opção 'Uniforme' selecionada. Abaixo, há quatro campos de entrada organizados em duas colunas: 'Limite Inferior λ', 'Limite Superior λ', 'Limite Inferior μ' e 'Limite Superior μ'. Um campo de entrada 'Tempo Máximo' está na base. Um botão azul com o texto 'Simular' está na base do formulário.

Figura 2: Formulário para distribuição uniforme

<sup>1</sup> Foram testados somente os navegadores citados.

<sup>2</sup> Para melhor visualização dos dados, sugere-se uma resolução de no mínimo 700 pixels de largura.

Tipo de Distribuição

Discreta

λ Taxa de Chegada

Valor A

Valor B

Valor C

Prob A

Prob B

Prob C

μ Taxa de Serviço

Valor A

Valor B

Valor C

Prob A

Prob B

Prob C

Tempo Máximo

Simular

Figura 3: Formulário para distribuição discreta

Depois de preenchido o formulário<sup>3</sup>, basta clicar no botão **Simular**. Duas tabelas serão mostradas, uma contendo os dados da simulação e outro com análises de tais dados, como exemplificado a seguir.

Figura 4:

Cliente	Tempo Simulacao	Tempo de Serviço	Tempo Início	Tempo na Fila	Tempo Final	Operador que Atendeu
2	2	0	2	0	2	1
3	5	13	5	0	18	1
1	0	20	0	0	20	0
4	5	10	18	13	28	1
6	18	3	28	10	31	1
5	5	12	20	15	32	0
7	22	6	31	9	37	1

Tabela de Resultados

Descrição	Valor
Tempo Médio de Espera	6.71
Probabilidade de Esperar	0.57
Probabilidade de operador 1 livre	0.00
Probabilidade de operador 2 livre	0.14
Tempo medio de servico	9.14
Tempo medio despendido no sistema	5.29

Tabelas de resultados da simulação

É importante notar que para visualizar todas as linhas de uma tabela, pode ser necessário utilizar a barra de rolagem que fica na lateral direita da mesma.

Há também um botão no topo da página que permite fazer nova simulação.

3 Preencha o formulário corretamente e com atenção, uma vez que não há validação de dados no sistema. Usar como valores zero ou números negativos poderão acarretar em travamento do sistema.

## 2. Informações Técnicas

Para o desenvolvimento desse projeto foram usadas as linguagens JavaScript, HTML5 e CSS. Além disso, também foram usadas as bibliotecas Bootstrap, JQuery e Popper. Também houve o apoio das ferramentas Visual Studio Code, Google Chrome Developer Tools e GitHub.

O programa possui como programa principal “Simulacao.js” no qual o mesmo possui funções “inEvent()” e “outEvent()” responsáveis por fazer o controle de entrada e saída de clientes dentro do sistema. Como funções auxiliares tem-se “getInEvent()” e “getOutEvent()” responsáveis por fazer o controle de qual distribuição será usada.

No roteiro inicial são setadas todas as variáveis de controle além de criar uma lista de clientes no sistema e clientes que já foram atendidos e um operador para manter o controle do sistema. O programa entra em um while em que enquanto o tempo total for menor do que o desejado e não haver mais ninguém no sistema será feita a simulação.

Se o tempo de chegada previsto for menor que o tempo de saída previsto acontecerá um evento de chegada do cliente no servidor, caso contrário, será acontecerá um evento de saída do cliente no servidor.

Após acabar a simulação (programa sai do while) é gerada uma tabela com os dados de chegadas e saídas e uma tabela com os cálculos estatísticos. Os cálculos foram feitos seguindo as fórmulas propostas nos slides vistos em aula.

Os atores, “cliente” e “operador” são classes criadas para facilitar no controle e armazenamento de dados gerados pelo sistema.

Para gerar um número aleatório é utilizado o “GNA.js” no qual existe uma semente gerada ao criar o objeto, sendo essa semente o tempo em milissegundos do sistema e para gerar um número aleatório usamos a função:  $(\text{semente} = (a * \text{semente} \% m)) / m$

Para a geração das variáveis aleatórias é utilizado o “FGVA.js” possuindo como filho três distribuições: Exponencial; Uniforme; Discreta.

Uma diferença do MM1 para o MM2 é que agora temos uma lista de servidores onde os clientes podem ser atendidos de forma paralela, mudamos algumas funções para adequar o código e criamos algumas classes para facilitar na criação do mesmo.

A diferença é levada praticamente na mudança do servidor pois a mecânica da fila continua a mesma. Caso um cliente chegue no sistema e não possuir nenhum servidor livre ele entrará na fila, caso algum servidor esteja livre ele será ocupado pelo cliente, sendo que a prioridade de ocupação é sempre do primeiro servidor.