

Amanda de Souza Caetano
Antônio Angelo Passini Júnior

Trabalho de Programação

Parte 1: Implementação do Modelo M/M/1

Professor Carlos Lopes

1. Instruções

1.1. Download e Execução

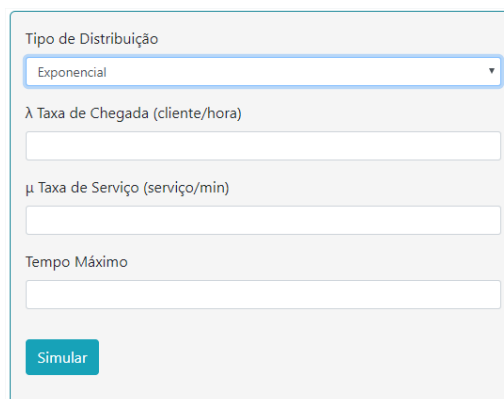
Primeiramente, para ter acesso ao programa desenvolvido, é necessário baixar e descompactar seus arquivos. Tais arquivos estão disponíveis no moodle e também no link https://drive.google.com/open?id=1l3STQ4rd7g10n1tOgk1UB_x7fXHXhZJ9.

Uma vez baixado e descompactado os arquivos, basta abrir o arquivo *index.html* em um navegador web¹, como Google Chrome ou Mozilla Firefox.

1.1. Como Simular

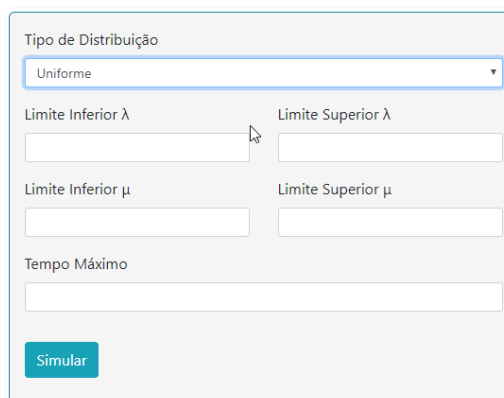
A tela principal² do programa contém somente um formulário para obtenção dos dados necessários para a simulação. O primeiro passo é escolher qual tipo de distribuição você pretende utilizar, sendo elas: exponencial, uniforme e discreta.

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



Formulário para distribuição discreta. O formulário possui um menu suspenso "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 "Simular" está localizado na base do formulário.

Figura 1: Formulário para distribuição discreta



Formulário para distribuição uniforme. O formulário possui um menu suspenso "Tipo de Distribuição" com a opção "Uniforme" selecionada. Abaixo, há quatro campos de entrada: "Limite Inferior λ", "Limite Superior λ", "Limite Inferior μ" e "Limite Superior μ". Um campo "Tempo Máximo" está localizado abaixo desses. Um botão "Simular" está na base.

Figura 2: Formulário para distribuição uniforme

¹ Foram testados somente os navegadores citados.

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

Tipo de Distribuição

Discreta

λ Taxa de Chegada			μ Taxa de Serviço		
Valor A	Valor B	Valor C	Valor A	Valor B	Valor C
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Prob A	Prob B	Prob C	Prob A	Prob B	Prob C
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Tempo Máximo

[Simular](#)

Figura 3: Formulário para distribuição discreta

Depois de preenchido o formulário³, 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.

Estado	Nº Cliente	Tempo Simulacao	Estado Servidor	Tamanho Fila	Agendamento Chegada	Agendamento Saída
CHEGADA	1	0	1	0	15	5
SAIDA	1	5	0	0	15	999999
CHEGADA	2	15	1	0	33	26
SAIDA	2	26	0	0	33	999999
CHEGADA	3	33	1	0	36	35
SAIDA	3	35	0	0	36	999999
CHEGADA	4	36	1	0	41	43

Descrição	Valor
Tempo Médio de Espera	15,71
Probabilidade de Esperar	0,43
Probabilidade de operador livre	0,09
Tempo medio de servico	24,57
Tempo medio despndido no sistema	27,14

Figura 4: 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: $(a * semente \% m) / m$

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