

Universidade Federal do Rio de Janeiro  
Bacharelado em Ciência da Computação  
Avaliação e Desempenho

## Relatório

### Trabalho de Simulação

Bruno Hryniewicz - 115150099  
Eduardo da Silva Barbosa - 116150432  
Miguel Bicudo - 116033119  
Nickolas Gomes Pinto - 114148170  
Rafael Pais Cardoso - 116140788  
Tainá da Silva Lima - 116165607  
2019

# Sumário

<b>Introdução</b>	<b>3</b>
<b>Algoritmos e organização do trabalho</b>	<b>4</b>
<b>Soluções</b>	<b>5</b>
<b>Questão 3 (Filas Sem Prioridades)</b>	<b>5</b>
Descrição	5
Desafios e decisões	5
Interpretação do enunciado	5
Simulação	6
Cadeia de Markov	7
Implementação - Simulador	11
Implementação - Cadeia de Markov	12
Discussão	15
Interpretação dos resultados	16
Cenário 1	16
Cenário 2	19
Cenário 3	21
Cenário 4	23
<b>Questão 4 (Filas com prioridade e sem preempção)</b>	<b>25</b>
Descrição	25
Desafios e decisões	25
Simulação	25
Cadeia de Markov	25
Fórmula analítica	27
Implementação - Cadeia de Markov	28
Interpretação dos resultados	29
Cenário 2	29
Cenário 3	31
Cenário 4	33
<b>Questão 5 (Lei da conservação)</b>	<b>35</b>
Descrição	35
Desafios e decisões	35
Simulação	35
Fórmula analítica	36
Implementação	37
Discussão	37
Interpretação dos resultados	38
Item a	38

Cenário 2	38
Cenário 3	39
Item b	40
Cenário 2	40
Cenário 3	41
Item c	42
Cenário 2	42
Cenário 3	43
Questão 6 (Simulação para verificar os resultados analíticos)	45
Desafios e decisões	45
Implementação	45
Interpretação dos resultados	45
Item 1	45
Item 2	46
Item 3	46
Item 4	46
Item 5	46
Item 6	47
Item 7	47
Item 8	47
Item 9	48
Item 10	48
Item 11	49
<b>Conclusão</b>	<b>50</b>
<b>Referências bibliográficas</b>	<b>51</b>

# Introdução

Este documento tem por objetivo apresentar todo o processo de desenvolvimento do trabalho final da disciplina Avaliação e Desempenho, contemplando principalmente os obstáculos encontrados no nosso caminho, assim como todas as decisões e medidas tomadas para tentar superar esses obstáculos e o obter o melhor resultado possível.

O trabalho de simulação procura criar um ambiente prático para estimular a aplicação de vários (senão todos) conteúdos abordados durante a disciplina, assim como:

- Valores médios (Esperança);
- Desvio Padrão e Variância
- Intervalo de confiança
- Certas noções de probabilidade;
- Sistemas de filas;
- Lei de Little;
- PASTA;
- Lei da Conservação;
- Cadeia de Markov;
- Entre outros;

Essencialmente, para o desenvolvimento desse trabalho, nos foi proposto criar um simulador para um Sistema de Filas em diferentes cenários e situações a fim de comparar os resultados obtidos do nosso simulador com os resultados alcançados através da resolução teórica desses cenários (resultados analíticos).

De posse de todos os resultados obtidos, consideramos que uma parte essencial desse trabalho é a interpretação dos mesmo, pois de nada adianta simular um sistema, obter os resultados e não fazer a correlação desses dados com o conteúdo da disciplina, logo, assumimos que a ideia central do trabalho é justamente debater sobre os resultados encontrados, estabelecendo uma conexão com o conteúdo da disciplina.

É importante ressaltar que, esse documento não tem como objetivo apresentar por completo todos os objetos de código utilizados durante o desenvolvimento do trabalho. Os mesmos estarão em arquivos separados por questões e vão acompanhar o entregável deste relatório.

# Algoritmos e organização do trabalho

Para este trabalho foi utilizado como ambiente de desenvolvimento o Google Collaboratory, muito conhecido por ser uma boa ferramenta para realizar trabalhos em equipe. Além disso, todos os algoritmos criados para as questões foram feitos com a linguagem Python, bem como algumas bibliotecas extras, como NumPy, Matplotlib e PrettyTable.

As soluções foram organizadas de maneira que cada notebook guarda as respostas para cada questão, e, dentro dele, há subdivisões para os respectivos itens, caso estes existam.

Segue abaixo os links para o acesso aos algoritmos de cada questão:

- [Questão 3](#)
- [Questão 4](#)
- [Questão 5](#)
- [Questão 6](#)

Os mesmos links também são acessíveis através do clique em cima dos títulos de cada questão descrita no tópico “Soluções”.

# Soluções

## Questão 3 (Filas Sem Prioridades)

### Descrição

Nessa questão, nós temos uma fila única e sem prioridade, com dois fluxos Poisson de chegadas e dois serviços associados. Esse sistema é separado em quatro cenários, cada um com suas particularidades, alterando as taxas do fluxo de chegada, as taxas de atendimento de cada serviço, assim como a distribuição do tempo de serviço (exponencial, determinístico e etc).

De forma geral, identificamos que essa questão tinha por objetivo reforçar os conhecimentos sobre um sistema de fila única sem prioridade, levando em consideração todas as variações do mesmo (cenários), visto que, com a comparação dos diferentes métodos de resolução nós podemos verificar importantes características desse sistema (descritas a frente).

### Desafios e decisões

#### Interpretação do enunciado

Os primeiros desafios encontrados foram como interpretar cada cenário da Tabela 1, mais especificamente compreender os vários valores para a taxa de chegada  $\lambda_1$ , a quantidade de servidores (se seria somente um ou haveria mais de um) e suas disposições e como representar a cadeia de markov do cenário 2, 3 e 4.

Inicialmente pensamos que a taxa de chegada  $\lambda_1$  iria variar de um estado para o outro no Cenário 1, 2 e 3. No entanto, isso não faria sentido, uma vez que estamos trabalhando com cadeias infinitas. Assim, percebemos que estes cenários teriam diversas situações em que o  $\lambda_1$  assumiria o valor 0.05 ou 0.1 e assim sucessivamente, sendo acrescido 0.05, até chegar na situação em que assume o valor 0.9. Interpretando a Tabela 1 dessa forma, ficou mais fácil resolver a questão.

Cenário	tempo de serviço	$\lambda_1$	$\lambda_2$	serviço 1 parâmetros	serviço 2 parâmetros
1	exp. com taxa $\mu_i$	0.05, 0.1, 0.15, ..., 0.9	0	$\mu_1 = 1$	$\mu_2 = 0$
2	exp. com taxa $\mu_i$	0.05, 0.1, 0.15, ..., 0.6	0.2	$\mu_1 = 1$	$\mu_2 = 0.5$
3	determ., média $1/\mu_i$	0.05, 0.1, 0.15, ..., 0.6	0.2	$\mu_1 = 1$	$\mu_2 = 0.5$
4	unif., intervalo $[a, b]$	0.08	0.05	$[a_1 = 5, b_1 = 15]$	$[a_2 = 1, b_2 = 3]$

Tabela 1: Cenários a simular

## Simulação

Um primeiro desafio foi definir como ocorreria o tempo de simulação, se o mesmo se daria orientado a tempo real, iterações de loops ou orientado a evento. Como uma primeira tentativa, foi decidido a utilização de eventos para definir a quantidade de iterações que haveria, contudo, chegamos na conclusão que diferentes simulações possuíam diferentes tempos de execução e isso estava afetando os cálculos e fazendo com que os diversos cenários tivessem o mesmo total de clientes passando pelo sistema.

Uma segunda tentativa foi a utilização de uma linha do tempo, onde a mesma é incrementada com o instante do acontecimento de um evento. Tal solução apresentou melhores resultados e, desse modo, optamos por utilizá-la como a implementação principal.

Vale ressaltar que a primeira tentativa não possuía uma lista de eventos o que se mostrou um problema para o cenário 3 (Determinístico), uma vez que a cada iteração era escolhido o evento com menor tempo de chegada e os demais eram descartados. Deste modo, na segunda tentativa foi criada uma fila de eventos, tornando o código genérico para os diferentes cenários.

Por fim, houve dificuldade em aproximar os resultados da simulação aos resultados analítico e da cadeia de markov, pois para uma boa aproximação eram necessária mais de 10.000 iterações (Tempo de simulação) o que resultava em um maior tempo de execução do código, levando muito tempo para obter os resultados. Com o intuito de fins de testes foi utilizada 4.000 iterações e para o resultado final aumentamos o número de iterações.

## Cadeia de Markov

- Cenário 1

Inicialmente, construímos um código relativamente simples (descrito na seção Implementação - Cadeia de Markov), uma vez que só tínhamos clientes de um tipo no sistema, para a obtenção da matriz de transição de taxas para o cenário 1. Esse código se baseia em uma cadeia de nascimento e morte e seus estados representam a quantidade de pessoas totais no sistema (Figura 1).

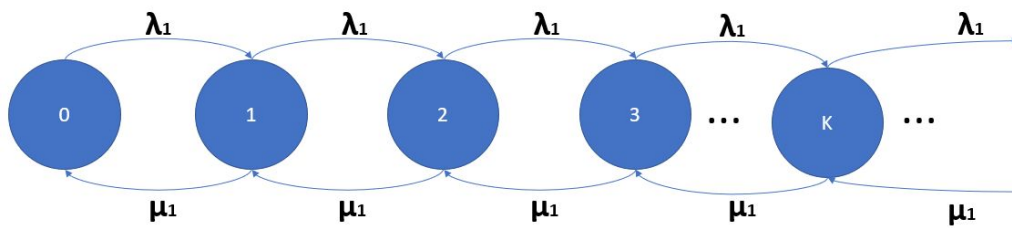


Figura 1: Cadeia de Markov em Tempo Contínuo. Cenário 1. Sendo  $\lambda_1 = 0,05, 0,1, 0,15, \dots, 0,9$ ;  $\mu_1 = 1$ . Os estados representam o número de pessoas no sistema.

No entanto, ainda tínhamos que determinar o ponto de corte para truncar a Cadeia de Markov. Assim, para o cenário 1, criamos uma Cadeia de Markov com 500 estados (Figura 2) e pelos resultados obtidos, constatamos que esse número de estados era suficiente, pois já tínhamos alcançado a convergência. Logo, aumentar o número de estados não iria acarretar em nenhuma mudança.

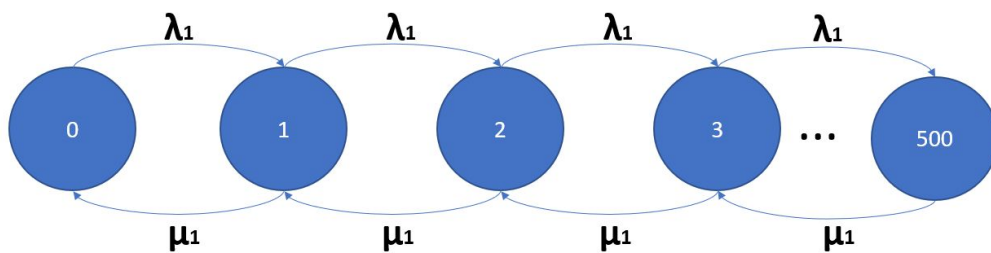


Figura 2: Cadeia de Markov em Tempo Contínuo truncada para 500 estados. Cenário 1. Sendo  $\lambda_1 = 0,05, 0,1, 0,15, \dots, 0,9$ ;  $\mu_1 = 1$ . Os estados representam o número de pessoas no sistema.



- Cenário 3

Para esse cenário, seria preciso trabalhar com uma Cadeia de Markov embutida. Como não tínhamos tempo e a Cadeia de Markov seria um bônus para o trabalho, optamos por não reproduzi-la.

- Cenário 2 e 4

Para os cenários 2 e 4, foi preciso modificar o código para obter a matriz de transição de taxas. Inicialmente, pensamos em montar uma Cadeia de Markov em que os estados representam o tipo de cliente no sistema e quantos clientes de cada tipo, conforme representado na Figura 3.

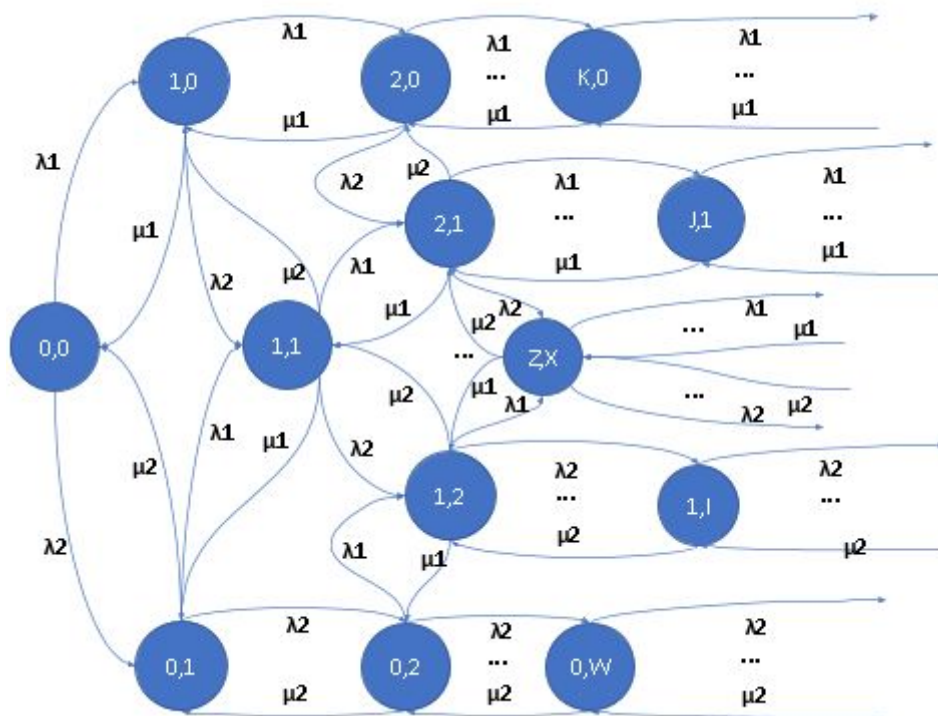


Figura 3: Cadeia de Markov em Tempo Contínuo. Cenário 2 e 4. Sendo  $\lambda_1 = 0,05, 0,1, 0,15, \dots, 0,9$ ;  $\mu_1 = 1$ ;  $\lambda_2 = 0,2$ ;  $\mu_2 = 0,5$  para o Cenário 2 e  $\lambda_1 = 0,08$ ;  $\mu_1 = 10$ ;  $\lambda_2 = 0,05$ ;  $\mu_2 = 2$  para o Cenário 4. Os estados representam o número de clientes de cada tipo no sistema (do lado esquerdo da vírgula são representados o número de clientes do tipo 1 e do lado direito o número de clientes do tipo 2).

O código construído, baseado na Cadeia de Markov da Figura 3, se mostrou ineficiente para gerar a matriz de taxas de transição e calcular os respectivos pis de cada estado, pois há uma demanda muito grande de processamento para realizar essas tarefas. Desse modo, estabelecemos um limite de estados que não demanda muito tempo para gerá-los e processá-los, e aplicamos uma abordagem alternativa para tentarmos nos aproximar da resposta exata, usando a matriz de probabilidade de transições.

Essa nova abordagem se resume em usar a matriz de taxa de transições e dividi-la pela soma das taxas de entrada e saída ( $\lambda_1 + \lambda_2 + u_1 + u_2$ ). Em seguida, somamos a nova matriz obtida com a matriz identidade. Desse modo, obtemos a matriz de probabilidade de transições e a elevamos a uma potência grande. A fórmula está representada abaixo:

$$P^n = ((Q/(\lambda + \mu)) + I)^n$$

Fórmula 1: Cálculo da matriz de probabilidade de transições, sendo P a matriz de probabilidades de transições, Q a matriz de taxas de transição,  $\lambda$  as taxas de entrada ( $\lambda_1 + \lambda_2$ ),  $\mu$  as taxas de saída ( $u_1 + u_2$ ), n uma potência grande e I a matriz identidade.

Os resultados obtidos dessa abordagem também não nos trouxeram uma melhora significativa. Chegamos a testá-la para o cenário 1 e a mesma funcionou como esperando. No entanto, o cenário 1 é um caso simples em que só existe um tipo de cliente. Como no cenário 2 e 4 estamos lidando com 2 clientes que apresentam taxas de entrada e serviços, é preciso levar em consideração a ordem com que os clientes chegam ao sistema e nós não estávamos levando isso em consideração ao montarmos a Cadeia de Markov.

Assim, precisamos levar a ordem dos clientes em consideração pelo seguinte motivo: vamos supor que cheguem 100 clientes com taxa de serviço 0.01 e depois 100 clientes com taxa de serviço 1. Os 100 primeiros que chegaram serão atendidos rapidamente, enquanto os outros serão atendidos mais lentamente. Analisando um determinado intervalo de tempo do sistema, podemos acabar obtendo um pequeno tamanho médio da fila de espera, pois o intervalo de tempo analisado pode cair justamente na situação em que temos só os clientes que são atendidos rapidamente. Se invertermos essa situação, fazendo com que cheguem 100 clientes com taxa de serviço 1 e depois 100 clientes com taxa de serviço 0.01, os 100 primeiros serão atendidos mais lentamente e ao pegar o mesmo intervalo de tempo do sistema, analisado anteriormente, podemos cair justamente na situação em que temos só os clientes que são atendidos lentamente e, assim, vamos acabar obtendo um maior tamanho médio da fila de espera.

Levando em consideração o que foi discutido acima, pensamos em construir uma nova representação da Cadeia de Markov, que pode ser vista na Figura 4. Cada estado representa a fila de clientes em ordem de chegada de cada cliente, em que o número 1 é o cliente do tipo 1 e o número 2 é o cliente do tipo 2. O cliente que aparece na extrema direita desse estado é o que está sendo servido, os demais estão na fila de espera. Assim, o estado 221 significa, por exemplo, que chegou um cliente do tipo 1 primeiro - e este está sendo servido - e depois dois clientes do tipo 2 chegaram e estão na fila de espera. O estado 0 significa que não há clientes na fila.

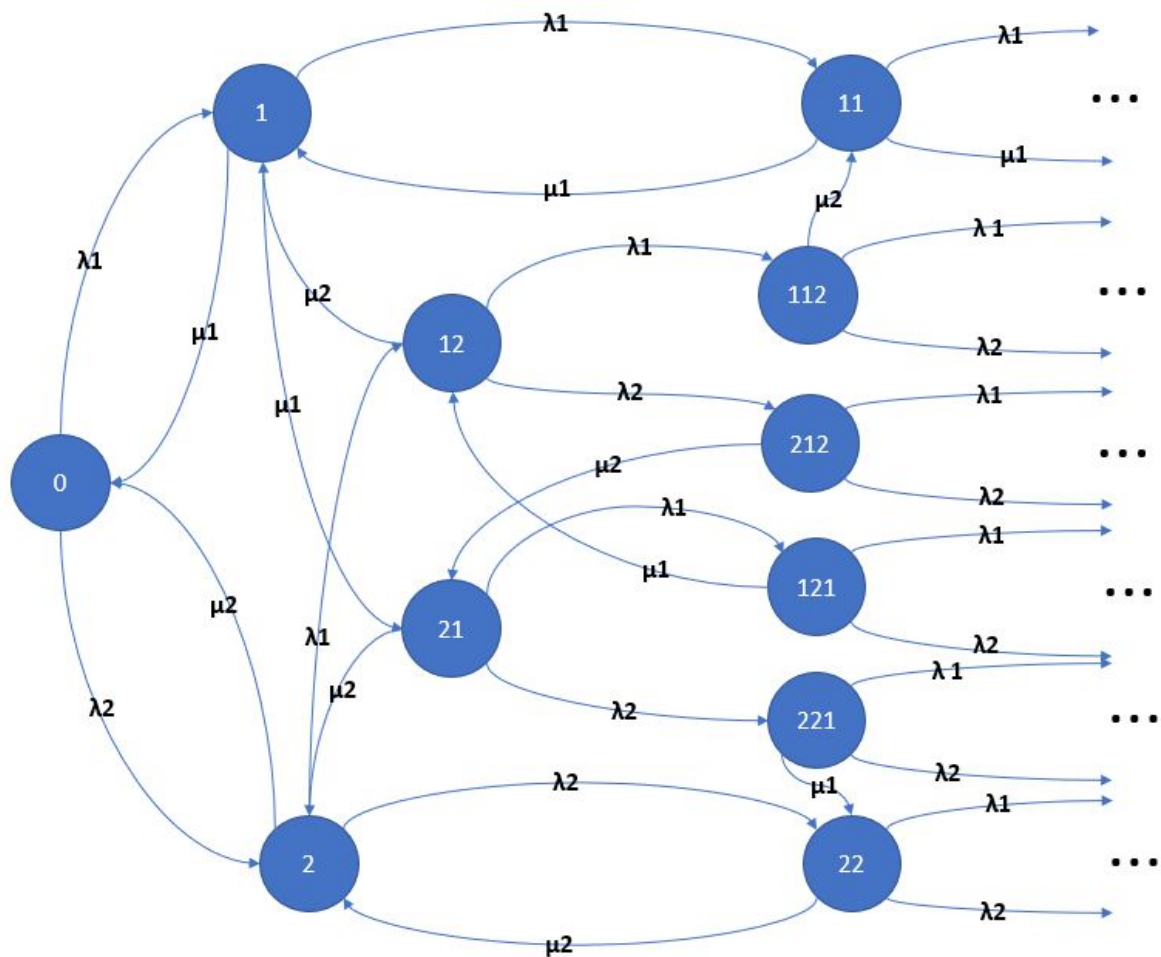


Figura 4: Cadeia de Markov em Tempo Contínuo. Cenário 2 e 4. Sendo  $\lambda_1 = 0,05, 0,1, 0,15, \dots, 0,9$ ;  $u_1 = 1$ ;  $\lambda_2 = 0,2$ ;  $u_2 = 0,5$  para o Cenário 2 e  $\lambda_1 = 0,08$ ;  $u_1 = 10$ ;  $\lambda_2 = 0,05$ ;  $u_2 = 2$  para o Cenário 4. O estado com número 0 significa que não há clientes na fila, os demais estados representam a fila em que o número 1 é o cliente do tipo 1 e o número 2 é o cliente do tipo 2. O cliente que está ao final da fila (extrema direita) representa o que está sendo servido.

Essa nova abordagem gerou resultados um pouco melhores. No entanto, continuamos com o problema para gerar um número suficiente de estados para alcançarmos valores próximos aos valores exatos. Tivemos que limitar o tamanho máximo que a fila poderia chegar para 10 clientes, sendo 1 deles o que está sendo servido e os demais aguardando na fila de espera, para que a execução não demandasse muito tempo. Com filas de até 10 clientes, são gerados 2047 estados que representam todas as possíveis combinações que os clientes podem estar distribuídos na fila. Apesar de termos um número considerado de estados, ainda não é suficiente para obtermos os valores mais próximos do real, uma vez que estamos analisando um sistema com poucos clientes.

No entanto, o perfil da curva gerado no cenário 2, por esta última abordagem, acompanha o perfil da simulação e o perfil da análise analítica, apesar da discrepância dos valores obtidos. Isso é mais perceptível para valores de  $\lambda_1$  acima

de 0.35. Chegamos a obter resultados para filas de até 11 e 12 clientes, mas com tempo de processamento muito demorados, e percebemos um aumento dos valores obtidos calculados. Assim, isso indica que se fosse possível aumentar a quantidade de estados gerados, conseguiríamos aproximar aos valores obtidos da simulação e no método analítico.

Também chegamos a converter a nossa matriz de taxas de transição para a matriz de probabilidade de transição e a elevamos a uma potência muito grande, conforme a Fórmula 1, mas não observamos melhoras nos resultados obtidos.

Uma outra alternativa seria encontrar um padrão que permitisse fazer a agregação de estados e, assim, montar cadeias maiores. No entanto, isso seria complexo de se fazer.

### Implementação - Simulador

Para essa seção, decidimos que a melhor representação seria pontuar as características mais importantes do nosso simulador e outras particularidades, tendo em vista que o funcionamento do mesmo é descrito parcialmente em outras seções deste documento assim como as decisões e escolhas tomadas pelo grupo, lembrando que o código completo ([implementação - questão 3](#)) acompanha o entregável final e o mesmo está comentado.

- O simulador é orientado a uma linha do tempo (variável de código: `linhaDoTempo`), que cresce de acordo com os instantes de ocorrência dos eventos, ou seja, a cada iteração um evento é tratado e a linha do tempo avança para o instante da ocorrência do evento.
- O nosso simulador inicialmente define um array de eventos (variável de código: `eventos`), onde o mesmo será mantido em ordem crescente durante todas as iterações. Além disso, o array de eventos só conterá no máximo 3 eventos, podendo ser composto de chegada 1, chegada 2 e serviço 1 ou chegada 1, chegada 2 ou serviço 2.
- Do ponto de vista das chegadas os eventos representam o instante na linha do tempo em que o cliente chega na fila de espera, enquanto do ponto de vista dos serviços os eventos representam o instante em que o cliente sairá do sistema, ou seja, após serem servidos.
- A simulação inicia realizando a amostragem de uma chegada do tipo 1 e do tipo 2 e ambas são armazenadas no array de eventos. A cada iteração o evento que ocorreu primeiro (tipo 1 ou tipo 2) é tratado e outra amostragem do evento de mesmo tipo é gerada a fim de representar uma nova chegada daquele tipo e é adicionada ao array de eventos.

- Além de um array de eventos é definido um array que representa uma fila de clientes, onde o mesmo armazenará tanto clientes do tipo 1 quanto clientes do tipo 2. Vale ressaltar que, a posição 0 da fila é utilizada para representar o servidor.
- Os eventos de serviços são gerados de acordo com o tipo do cliente que está no servidor (posição 0 a fila de clientes), deste modo o evento de serviço pode ser do tipo 1 ou do tipo 2.
- Cada simulação possui um tempo total de 4000, e a mesma é executada 1000 vezes, por fim, é realizado a média dos resultados de todas as simulações com seus devidos intervalos de confiança.
- Cada cenário possui um bloco de código onde a simulação é chamada com os parâmetros adequados.

### Implementação - Cadeia de Markov

Os códigos das implementações para as Cadeias de Markov estão no documento a seguir: [implementação - questão 3](#). Aqui faremos um breve resumo dessas implementações em cada cenário analisado.

- Cenário 1

Foi implementada uma Cadeia de Markov como um processo de nascimento e morte. Inicialmente, criamos uma matriz, chamada matriz, de tamanho 500x500 e preenchida com zeros. As linhas da matriz se referem aos respectivos estados e cada índice se refere aos demais estados em que podem ser alcançados por outro estado. Assim, a posição `matriz[0][1]`, por exemplo, é interpretada como sendo a taxa com que o estado 0 (linha 0) alcança o estado 1 (coluna 1). Após preencher a matriz com as respectivas taxas, a diagonal foi preenchida de maneira com que a soma dos elementos de uma linha sempre resulte em 0. Ou seja, a posição `matriz[i][i]` guarda a soma, com o sinal trocado, dos valores dos elementos que estão na mesma linha que esta posição em específico.

Feito isso, calculamos os  $\pi$ 's utilizando o passo a passo descrito na Seção 2 do trabalho, com a nossa matriz obtida.

Em seguida, calculamos  $E[Nq]$  utilizando a seguinte expressão:

$$E[Nq] = \sum_{k=1}^{\infty} (k-1)\pi_k$$

Fórmula 2: Cálculo da  $E[Nq]$

Para encontrarmos  $E[W]$ , aplicamos Little e utilizamos a  $E[Nq]$  calculada anteriormente e dividimos pela soma das taxas de entrada ( $\lambda_1 + \lambda_2$ ), como observado na Fórmula 2.

$$E[W] = \frac{E[Nq]}{\lambda}$$

Fórmula 3: Cálculo da  $E[W]$ , sendo  $\lambda$  a soma das taxas de entrada ( $\lambda_1 + \lambda_2$ )

- Cenário 2

- Cadeia de Markov da Figura 3

Nessa implementação, os estados da Cadeia de Markov representam o tipo de cliente no sistema e quantos clientes de cada tipo há no sistema. Além disso, não foi considerada a importância da ordem com que as pessoas chegam, como discutido anteriormente.

Como tivemos problemas de eficiência para gerar a matriz de taxas de transição e calcular os respectivos  $\pi$ 's de cada estado, tivemos que trabalhar com uma matriz de transição de taxas de tamanho 1600x1600. Essa matriz, apesar de parecer grande, ainda é pequena, pois o maior número de pessoas observado no sistema são 78. Assim, geramos todas as combinações possíveis de estados e atribuímos as respectivas taxas para os seus estados adjacentes.

Um exemplo de estado gerado é o 10/20, que corresponde a 10 clientes do tipo 1 e 20 clientes do tipo 2 no sistema. A partir desse estado, são gerados os estados 11/20 para representar uma chegada de um cliente do tipo 1, o estado 10/21 para representar uma chegada de um cliente do tipo 2, o estado 9/20 para representar uma saída do cliente do tipo 1, e o estado 10/19 para representar uma saída do cliente do tipo 2. Todas as taxas são associadas a esses estados e colocadas em um dicionário associado ao estado. Os estados serão as chaves e as taxas os valores desses estados no dicionário. Um outro dicionário mais externo irá associar o estado 10/20 (será a chave) com o dicionário dos seus estados adjacentes (será o valor).

Após gerar todos os estados e associar as respectivas taxas, será criada a matriz de transição de estados com todos esses valores obtidos. Com essa matriz, calculamos os  $\pi$ 's utilizando o passo a

passo descrito na Seção 2 do trabalho. Ou podemos usar a abordagem de transformá-la em uma matriz de probabilidade de transição (Fórmula 1), para tentar obter melhores resultados, e calcular os  $\pi$  obtidos com esta matriz. Em seguida, aplicamos a Fórmula 2 para calcular  $E[N_q]$  e a Fórmula 3 Para calcular  $E[W]$ .

- Cadeia de Markov da Figura 4

Nessa implementação, estamos levando em consideração a ordem com que os clientes chegam ao sistema.

Os estados Cadeia de Markov representam a fila de clientes em ordem de chegada de cada cliente, em que o número 1 é o cliente do tipo 1 e o número 2 é o cliente do tipo 2. O cliente que aparece na extrema direita desse estado é o que está sendo servido, os demais estão na fila de espera, como discutido anteriormente.

Estamos trabalhando com uma matriz de transição de taxas de tamanho  $2047 \times 2047$ . Apesar de ser grande, só são geradas no máximo 10 pessoas na fila (considerando a fila de 10 pessoas, teremos 9 na fila de espera e 1 sendo servida). Como a ordem importa, são geradas todas as combinações possíveis de tipos de clientes e suas posições na fila.

Ao gerar todos esses estados, que representam essas filas, associamos as respectivas taxas aos estados adjacentes e montamos a matriz de transição de taxas. Com essa matriz, calculamos os  $\pi$  utilizando o passo a passo descrito na Seção 2 do trabalho. Ou podemos usar a abordagem de transformá-la em uma matriz de probabilidade de transição (Fórmula 1), para tentar obter melhores resultados, e calcular os  $\pi$  obtidos com esta matriz. Em seguida, aplicamos a Fórmula 2 para calcular  $E[N_q]$  e a Fórmula 3 Para calcular  $E[W]$ .

Para maiores detalhes da implementação dessa cadeia, consulte o [notebook](#).

- Cenário 4

A implementação dessa Cadeia de Markov foi similar ao que foi descrito para as Cadeias de Markov do Cenário 2. No entanto, ajustes nos cálculos das taxas de serviços foram feitos, pois a distribuição do tempo de serviço é uniforme para esse cenário.

## Discussão

No nosso desenvolvimento inicial, construímos um simulador que pegava uma amostragem completamente nova a cada chegada, ou seja, a cada escolha de amostra (selecionada sempre a de menor tempo) as demais amostras eram descartadas, de modo que essa implementação não se mostrou correta para todos os cenários mas somente para os cenários 1 e 2, uma vez que os mesmos possuem uma distribuição de serviço do tipo exponencial e a propriedade de falta de memória pode ser utilizada.

Para solucionar isso, acrescentamos uma fila de eventos ao nosso simulador para armazenar cada chegada e cada serviço, ou seja, guarda os instantes de tempos das chegadas (para  $\lambda_1$  e  $\lambda_2$ ) e o instante de tempo do fim de serviços. Importante ressaltar que essa fila é ordenada em ordem crescente de instantes de tempo, isto é, o primeiro a chegar estará na posição mais à frente dessa fila e a posição 0 dessa fila representa o serviço.

Podemos observar também que, todos os nossos resultados ficaram extremamente próximos para o Cenário 1, sendo, na maior parte das vezes, compreendidos dentro do nosso intervalo de confiança, mostrando que a nossa implementação e as decisões tomadas durante esse processo foram boas escolhas para garantir uma boa aproximação do mesmo. Porém, vale ressaltar que isso nem sempre é válido para os outros cenários, mas somente para o Cenário 1. Nos demais cenários, no geral, são obtidos bons resultados para as resoluções analíticas e por simulação, porém, quando temos um número pequeno de iterações é possível obter uma diferença um pouco maior entre os diferentes métodos de resolução, de forma que simulações com muitas iterações são necessárias para uma aproximação mais precisa. No entanto, é necessário ressaltar que não conseguimos realizar simulações com um número expressivo de iterações, visto que em todas as tentativas a execução dava erro por limitação de recursos computacionais, erro esse que aconteceu com todos os membros do grupo.



Outro ponto importante é que num primeiro momento o nosso intervalo de confiança era calculado utilizando a função abaixo:

```
sp.stats.norm.interval(0.95, loc=media, scale=desvio)
```

Fórmula 4: Função que calcula o intervalo de confiança

Mas, ao calcular desse modo, era obtido um intervalo bem maior que os próprios resultados, inviabilizando o seu uso. Com isso, passamos a calcular o nosso intervalo de uma outra maneira (que pode ser vista abaixo), tornando o mesmo mais confiável.

$$\text{intervalo de confiança} = \text{média} \pm \left[ \frac{1.96 \times \sigma}{\sqrt{n}} \right]$$

Fórmula 5: Cálculo do intervalo de confiança

## Interpretação dos resultados

### Cenário 1

Neste primeiro cenário, podemos observar que os resultados gerados pela simulação são bem parecidos com aqueles calculados através de fórmulas analíticas e pelo método da Cadeia de Markov.

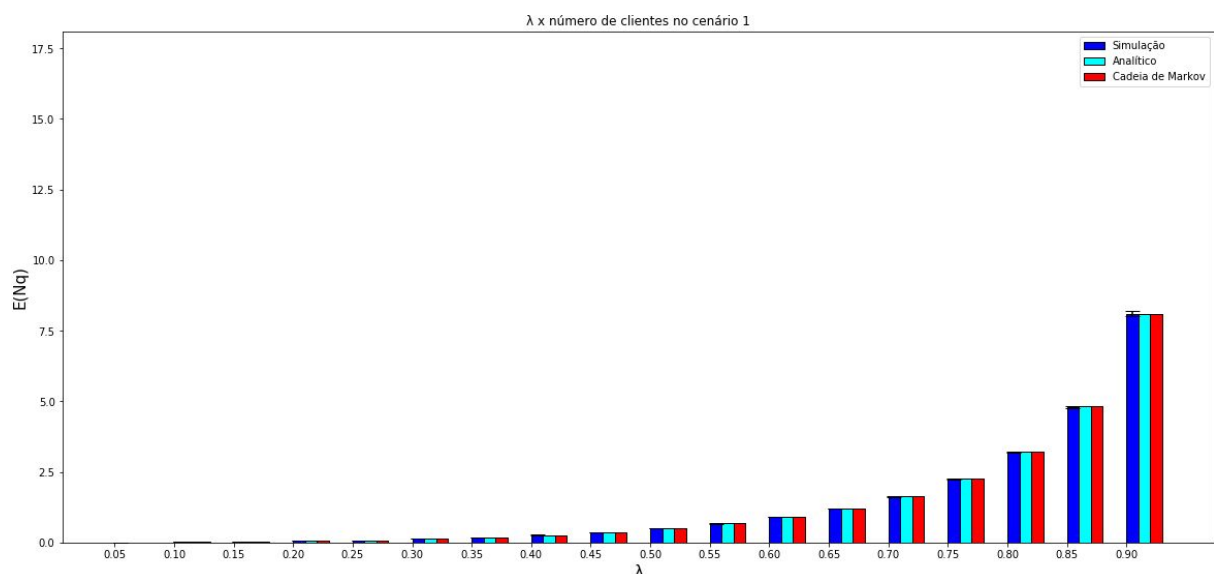


Figura 5: Resultados da E[Nq] obtidos via simulação (azul escuro), método analítico (azul claro) e Cadeia de Markov (vermelho). No eixo y temos a E[Nq] e no eixo x as taxas de entrada, variando entre 0.05 a 0.9.

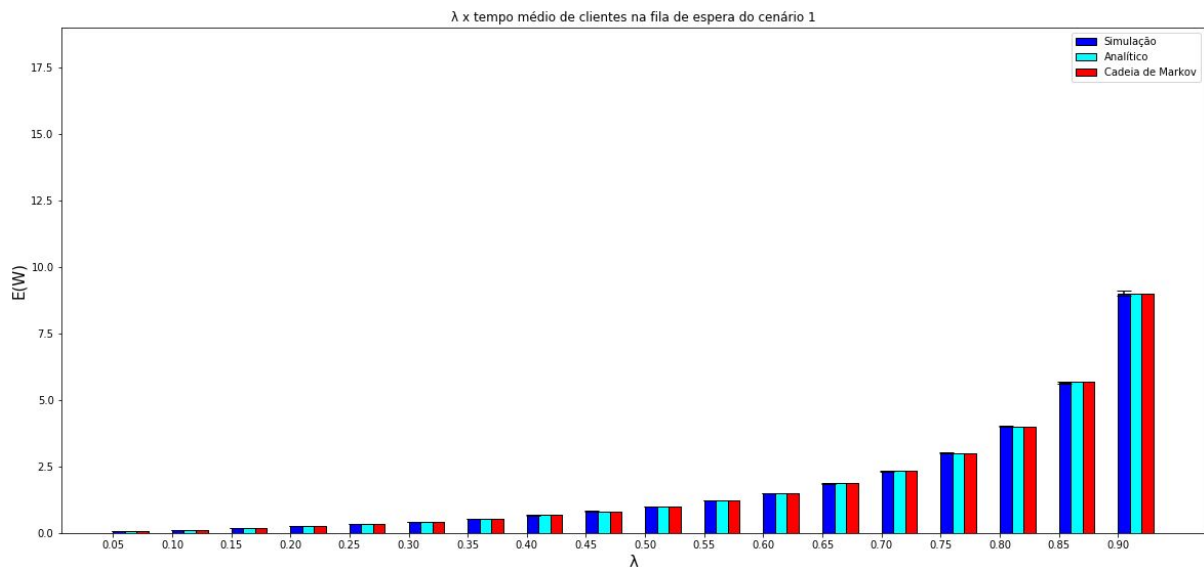


Figura 6: Resultados da  $E[W]$  obtidos via simulação (azul escuro), método analítico (azul claro) e Cadeia de Markov (vermelho). No eixo y temos a  $E[W]$  e no eixo x as taxas de entrada, variando entre 0.05 a 0.9.

Além disso, é perceptível que a curva gerada pelos valores da simulação é, de certa forma, coerente com o que ocorre na prática: dado um  $\lambda$  pequeno, o acúmulo de clientes na fila de espera se torna também pequeno, pois o tempo entre chegadas é grande; dado um  $\lambda$  grande, o tempo entre chegadas é pequeno, fazendo com que a cada instante exista um número mais significativo de clientes na fila de espera. Sendo assim, temos que  $E[N_q]$  cresce proporcionalmente ao  $\lambda$ .

$\lambda$	Simulação: $E[N_q]$ ; (Intervalo de confiança)	Analítico: $E[N_q]$	Cadeia de Markov: $E[N_q]$
0.05	0.00264; (0.0026, 0.00268)	0.00263	0.00263
0.1	0.01108; (0.01099, 0.01117)	0.01111	0.01111
0.15	0.02649; (0.02634, 0.02664)	0.02647	0.02647
0.2	0.04999; (0.04974, 0.05025)	0.05	0.05
0.25	0.08312; (0.08276, 0.08349)	0.08333	0.08333
0.3	0.12819; (0.12768, 0.1287)	0.12857	0.12857
0.35	0.18834; (0.18764, 0.18904)	0.18846	0.18846
0.4	0.26676; (0.26574, 0.26777)	0.26667	0.26667
0.45	0.36812; (0.36681, 0.36943)	0.36818	0.36818
0.5	0.49968; (0.49788, 0.50148)	0.5	0.5
0.55	0.67101; (0.66854, 0.67347)	0.67222	0.67223
0.6	0.89977; (0.8964, 0.90315)	0.9	0.9
0.65	1.20518; (1.20023, 1.21014)	1.20714	1.20714
0.7	1.62897; (1.62178, 1.63615)	1.63333	1.63334
0.75	2.24883; (2.23786, 2.25979)	2.25	2.25
0.8	3.20335; (3.18528, 3.22141)	3.2	3.2
0.85	4.79823; (4.7651, 4.83135)	4.81667	4.81669
0.9	8.10104; (8.01841, 8.18366)	8.1	8.1

Tabela 2: Resultados da  $E[N_q]$  obtidos via simulação, método analítico e Cadeia de Markov.

$\lambda$	Simulação: E[W]; (Intervalo de confiança)	Analítico: E[W]	Cadeia de Markov: E[W]
0.05	0.05281; (0.05203, 0.05359)	0.05263	0.05263
0.1	0.11079; (0.10991, 0.11168)	0.11111	0.11111
0.15	0.17662; (0.17562, 0.17762)	0.17647	0.17647
0.2	0.24997; (0.24872, 0.25123)	0.25	0.25
0.25	0.3325; (0.33104, 0.33395)	0.33333	0.33334
0.3	0.42729; (0.4256, 0.42899)	0.42857	0.42857
0.35	0.53812; (0.53612, 0.54012)	0.53846	0.53847
0.4	0.66689; (0.66435, 0.66943)	0.66667	0.66667
0.45	0.81804; (0.81514, 0.82095)	0.81818	0.81818
0.5	0.99936; (0.99576, 1.00297)	1.0	1.0
0.55	1.22001; (1.21552, 1.2245)	1.22222	1.22223
0.6	1.49962; (1.49399, 1.50525)	1.5	1.5
0.65	1.85413; (1.8465, 1.86176)	1.85714	1.85714
0.7	2.3271; (2.31684, 2.33736)	2.33333	2.33334
0.75	2.99844; (2.98382, 3.01306)	3.0	3.0
0.8	4.00418; (3.9816, 4.02676)	4.0	4.0
0.85	5.64497; (5.606, 5.68394)	5.66667	5.66669
0.9	9.00115; (8.90935, 9.09295)	9.0	9.0

Tabela 3: Resultados da E[W] obtidos via simulação, método analítico e Cadeia de Markov.

Além do E[Nq], outros parâmetros foram capturados, como E[W], E[T], etc. Eles podem ser consultados no [notebook](#).

## Cenário 2

Neste segundo cenário, podemos observar que a existência de uma segunda taxa de chegada ( $\lambda_2 = 0.2$ ), faz com que a quantidade média de pessoas na fila de espera (E[Nq]) e o tempo médio de espera na fila de espera (E[W]) subam consideravelmente. Ao analisarmos os valores de E[Nq] e E[W] para os dois cenários, podemos ver, nitidamente, essa diferença para cada valor que  $\lambda_1$  assume.

Vale ressaltar que neste cenário, a Cadeia de Markov não apresentou bons resultados para os  $\lambda_1$  acima de 0.35. Apesar de estarmos gerando 2047 estados, a maior fila que conseguimos gerar é de até 10 clientes, sendo, em uma fila com 10 clientes, 9 na fila de espera e 1 sendo servido, e, como a ordem importa, são geradas todas as combinações possíveis dos tipos de clientes e suas posições na fila. Por essa limitação do número máximo de pessoas no sistema, estamos obtendo resultados abaixo do esperado.

No entanto, podemos perceber que os resultados obtidos do E[Nq] e do E[W], da Cadeia de Markov, apresentam um perfil crescente à medida que aumentamos o valor do  $\lambda_1$ . Também fizemos testes com filas de até 11 e 12 pessoa, que demoraram mais de 10 minutos para rodarem, e percebemos um aumento dos valores de E[Nq] e do E[W] para os valores de  $\lambda_1$  acima de 0.35.

Assim, se conseguíssemos trabalhar com um número maior de clientes, muito provavelmente, alcançaríamos os valores obtidos na simulação e no método

analítico. Vale lembrar que também tentamos utilizar a matriz de probabilidade de transição, aplicando a Fórmula 1, mas não observamos melhora nos resultados obtidos.

Por fim, observando os resultados apresentados abaixo, foi possível notar que para o  $\lambda_1$  igual 0.6 tanto  $E[Nq]$  (Quantidade média de pessoas na fila de espera) quanto o  $E[W]$  tendem para infinito, tal fenômeno fica explícito ao olharmos o salto no gráfico e o salto dos resultados do  $\lambda_1$  igual a 0.55 para o  $\lambda_1$  igual a 0.6.

$\lambda$	Simulação: $E[Nq]$ ; (Intervalo de confiança)	Analítico: $E[Nq]$	Cadeia de Markov: $E[Nq]$
0.05	0.38491; (0.38301, 0.38682)	0.36818	0.38416
0.1	0.53826; (0.53567, 0.54086)	0.5	0.53189
0.15	0.73752; (0.734, 0.74104)	0.67222	0.7147
0.2	1.00075; (0.99578, 1.00571)	0.9	0.93748
0.25	1.35372; (1.34657, 1.36088)	1.20714	1.20407
0.3	1.83473; (1.82471, 1.84475)	1.63333	1.51632
0.35	2.52697; (2.51082, 2.54311)	2.25	1.87325
0.4	3.58288; (3.55824, 3.60751)	3.2	2.2704
0.45	5.3948; (5.34741, 5.44219)	4.81667	2.69974
0.5	9.14063; (9.03431, 9.24694)	8.1	3.15027
0.55	19.5709; (19.16959, 19.9722)	18.05	3.60924
0.6	97.78664; (94.17377, 101.39951)	998.001	4.06375

Tabela 3: Resultados da  $E[Nq]$  obtidos via simulação, método analítico e Cadeia de Markov.

$\lambda$	Simulação: $E[W]$ ; (Intervalo de confiança)	Analítico: $E[W]$	Cadeia de Markov: $E[W]$
0.05	1.53965; (1.53202, 1.54728)	1.47273	1.53663
0.1	1.79422; (1.78557, 1.80287)	1.66667	1.77297
0.15	2.10721; (2.09716, 2.11726)	1.92063	2.04199
0.2	2.50187; (2.48945, 2.51429)	2.25	2.34371
0.25	3.00828; (2.99238, 3.02417)	2.68254	2.67572
0.3	3.66946; (3.64942, 3.6895)	3.26667	3.03265
0.35	4.59448; (4.56513, 4.62384)	4.09091	3.40591
0.4	5.97146; (5.93041, 6.01252)	5.33333	3.784
0.45	8.29969; (8.22679, 8.3726)	7.41026	4.15344
0.5	13.05804; (12.90616, 13.20992)	11.57143	4.50038
0.55	26.09453; (25.55946, 26.6296)	24.06667	4.81232
0.6	122.2333; (117.71721, 126.74939)	1247.50125	5.07968

Tabela 4: Resultados da  $E[W]$  obtidos via simulação, método analítico e Cadeia de Markov.

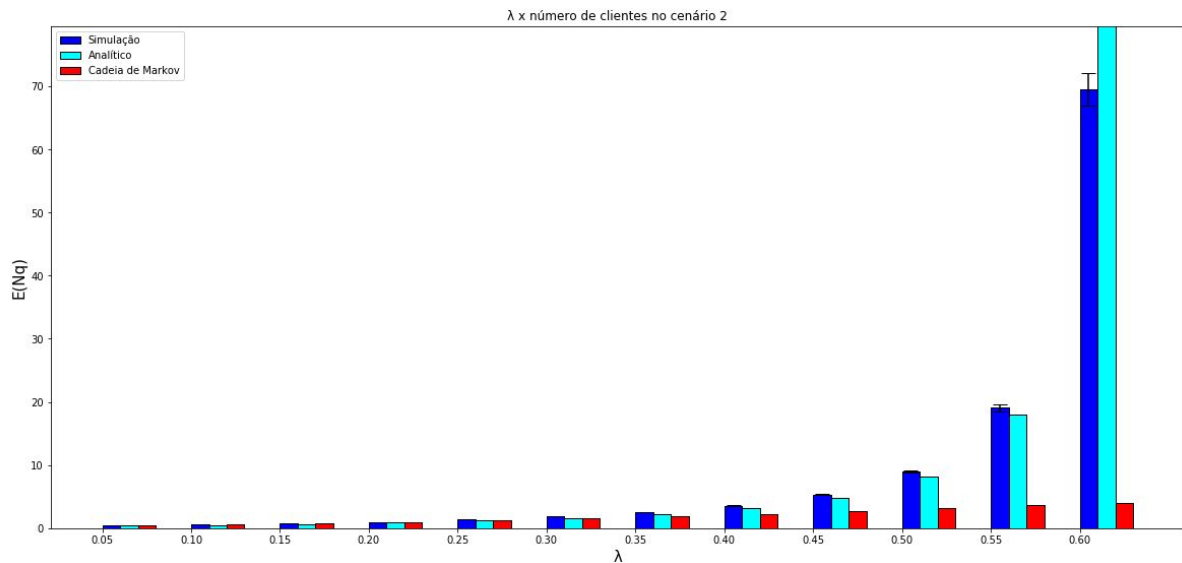


Figura 7: Resultados da  $E[Nq]$  obtidos via simulação (azul escuro), método analítico (azul claro) e Cadeia de Markov (vermelho). No eixo y temos a  $E[Nq]$  e no eixo x as taxas de entrada, variando entre 0.05 a 0.9.

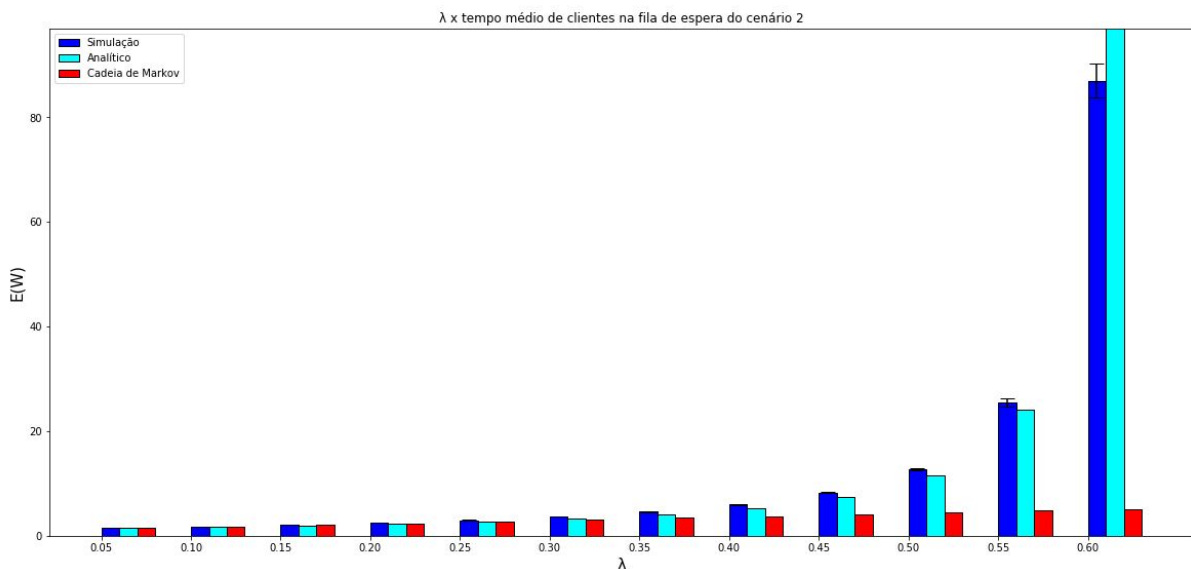


Figura 8: Resultados da  $E[Nq]$  obtidos via simulação (azul escuro), método analítico (azul claro) e Cadeia de Markov (vermelho). No eixo y temos a  $E[Nq]$  e no eixo x as taxas de entrada, variando entre 0.05 a 0.9.

### Cenário 3

Neste cenário é notável também a semelhança entre os valores gerados pelo simulador e pelos cálculos analíticos, com a exceção do caso em que  $\lambda = 0.6$ , pois é nele que ocorre o fenômeno em que  $E[Nq]$  tende a infinito.

Apesar de termos feito a abordagem da Cadeia de Markov para os outros cenários, neste se mostrou ser um desafio bastante complicado por fazer uso do conceito de cadeia de Markov embutida, assunto o qual não dominamos bem. Optamos por não fazê-la por motivos de falta de tempo associado ao fato de tal



parte do trabalho ser considerado ponto extra e, portanto, não temos tais resultados para que possamos compará-lo com os de simulação e analíticos.

$\lambda$	Simulação: $E[N_q]$ ; (Intervalo de confiança)	Analítico: $E[N_q]$
0.05	0.19339; (0.19268, 0.19409)	0.17045
0.1	0.27; (0.26901, 0.27099)	0.225
0.15	0.36938; (0.3681, 0.37065)	0.29944
0.2	0.4982; (0.49645, 0.49996)	0.4
0.25	0.6765; (0.67402, 0.67898)	0.53724
0.3	0.91629; (0.91268, 0.9199)	0.72917
0.35	1.26295; (1.25752, 1.26837)	1.00833
0.4	1.79473; (1.78578, 1.80367)	1.44
0.45	2.70745; (2.69107, 2.72383)	2.17652
0.5	4.52446; (4.48718, 4.56173)	3.675
0.55	9.94579; (9.79497, 10.09661)	8.22115
0.6	69.76665; (67.2253, 72.30799)	456.68571

Tabela 5: Resultados da  $E[N_q]$  obtidos via simulação e método analítico.

$\lambda$	Simulação: $E[W]$ ; (Intervalo de confiança)	Analítico: $E[W]$
0.05	0.77354; (0.7707, 0.77638)	0.68182
0.1	0.89999; (0.89669, 0.90329)	0.75
0.15	1.05536; (1.05171, 1.05901)	0.85556
0.2	1.24551; (1.24112, 1.2499)	1.0
0.25	1.50334; (1.49783, 1.50884)	1.19388
0.3	1.83258; (1.82537, 1.8398)	1.45833
0.35	2.29627; (2.28641, 2.30613)	1.83333
0.4	2.99121; (2.9763, 3.00612)	2.4
0.45	4.16531; (4.14011, 4.19051)	3.34848
0.5	6.46351; (6.41026, 6.51676)	5.25
0.55	13.26105; (13.05995, 13.46214)	10.96154
0.6	87.20831; (84.03163, 90.38498)	570.85714

Tabela 6: Resultados da  $E[W]$  obtidos via simulação e, método analítico.

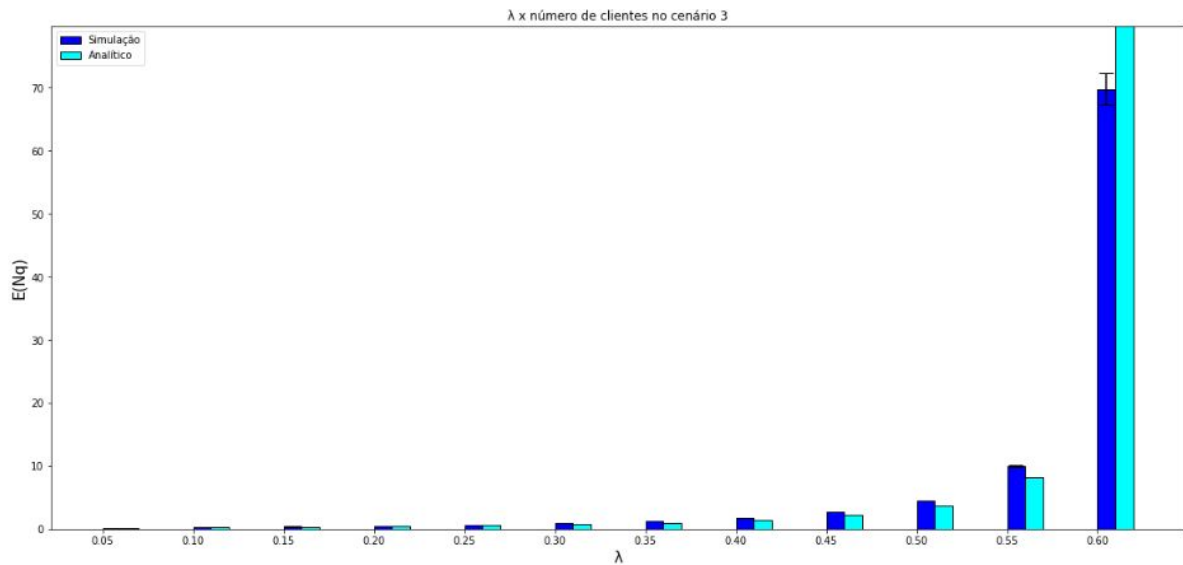


Figura 9: Resultados da  $E[Nq]$  obtidos via simulação (azul escuro) e método analítico (azul claro). No eixo y temos a  $E[Nq]$  e no eixo x as taxas de entrada, variando entre 0.05 a 0.9.

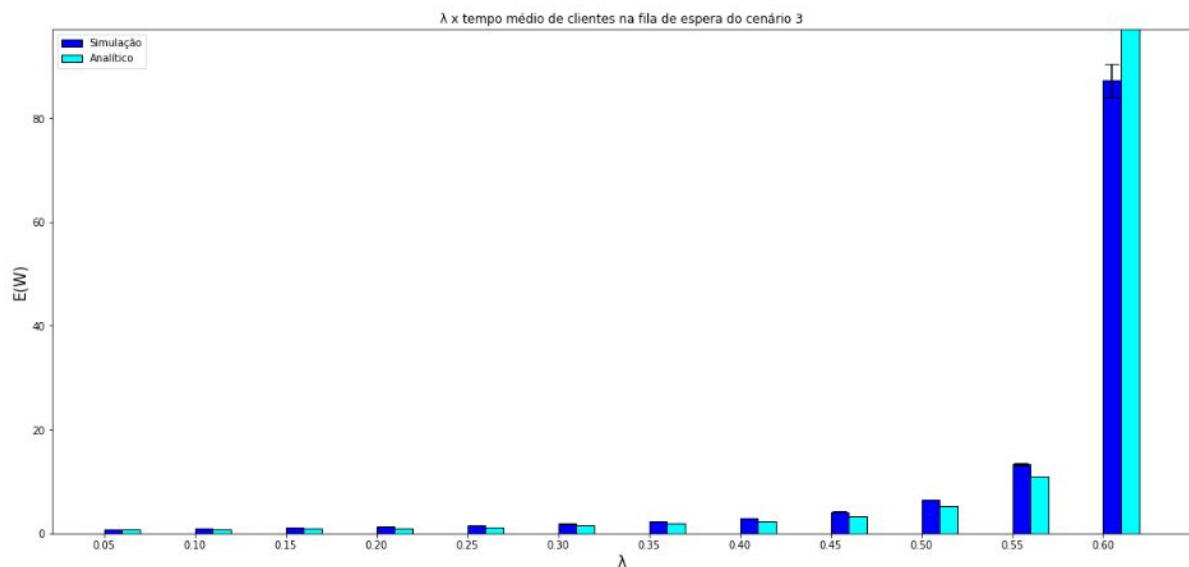


Figura 10: Resultados da  $E[W]$  obtidos via simulação (azul escuro) e método analítico (azul claro). No eixo y temos a  $E[W]$  e no eixo x as taxas de entrada, variando entre 0.05 a 0.9.

## Cenário 4

Neste último cenário, não obtivemos bons resultados, pois todos os métodos de resolução divergiram bastante. Acreditamos, que tal divergência para a Cadeia de Markov pode ser pelo fato da quantidade de estados não serem suficientes ou que as contas realizadas para as taxas de serviços estejam equivocadas ou que a própria Cadeia de Markov deveria ter sido representada de outra forma, contudo para a simulação e para a resolução analítica não descobrimos o motivo da divergência. Abaixo seguem os resultados encontrados.

$\lambda$	Simulação: $E[Nq]$ ; (Intervalo de confiança)	Analítico: $E[Nq]$	Cadeia de Markov: $E[Nq]$
0.08	5.54791; (5.43518, 5.66064)	1.73935	0.00179

Tabela 9: Resultados da  $E[Nq]$  obtidos via simulação, método analítico e Cadeia de Markov.

$\lambda$	Simulação: $E[W]$ ; (Intervalo de confiança)	Analítico: $E[W]$	Cadeia de Markov: $E[W]$
0.08	42.67625; (41.80909, 43.5434)	13.37963	0.01375

Tabela 10: Resultados da  $E[W]$  obtidos via simulação, método analítico e Cadeia de Markov.

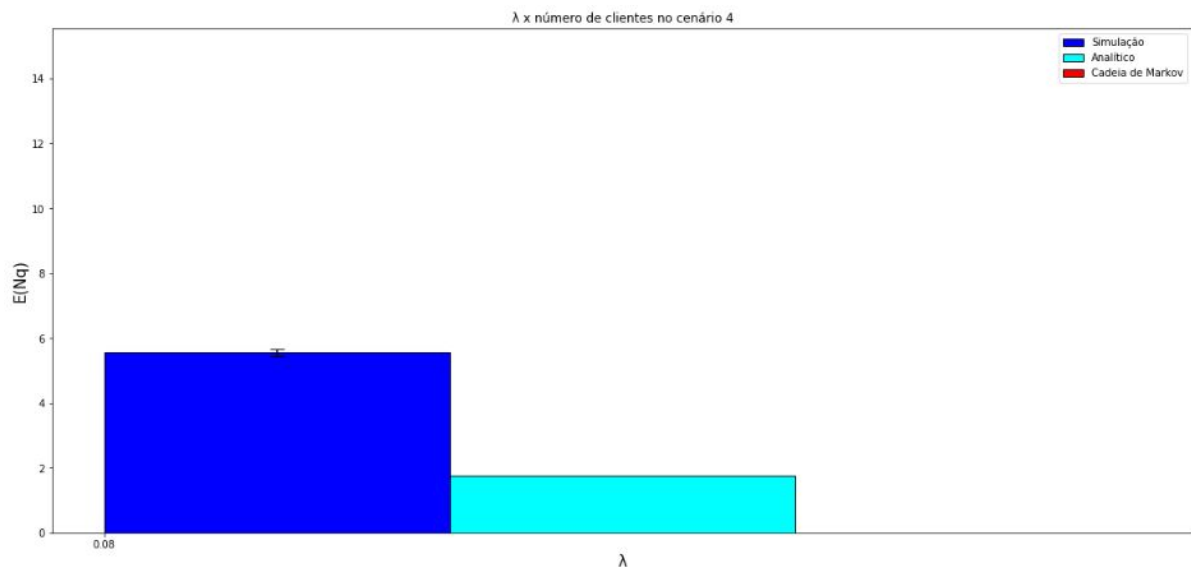


Figura 11: Resultados da  $E[Nq]$  obtidos via simulação (azul escuro), método analítico (azul claro) e Cadeia de Markov (vermelho). No eixo y temos a  $E[Nq]$  e no eixo x as taxas de entrada, variando entre 0.05 a 0.9.

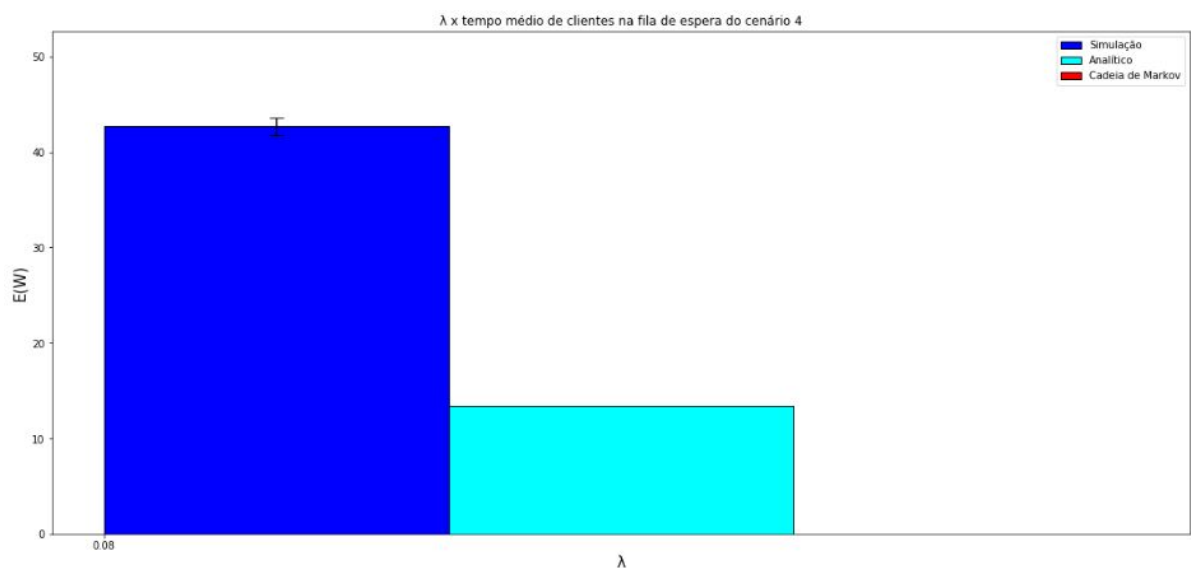


Figura 12: Resultados da  $E[Nq]$  obtidos via simulação (azul escuro), método analítico (azul claro) e Cadeia de Markov (vermelho). No eixo y temos a  $E[Nq]$  e no eixo x as taxas de entrada, variando entre 0.05 a 0.9.



## Questão 4 (Filas com prioridade e sem preempção)

### Descrição

Consiste de uma variação da questão anterior, onde todos os cenários são os mesmos, mas o nosso Sistema de Fila foi alterado de uma fila sem prioridade para uma fila com prioridade dos serviços da classe 1 sobre a classe 2, de forma que tivemos que realizar uma série de mudanças para o desenvolvimento do nosso simulador.

De forma geral, identificamos que essa questão tinha por objetivo reforçar os conhecimentos sobre um sistema de fila única com prioridade, levando em consideração todas as variações do mesmo (cenários), visto que, com a comparação dos diferentes métodos de resolução nós podemos verificar importantes características desse sistema (descritas a frente).

É importante ressaltar que essa questão é extremamente semelhante à questão anterior (Q3), mudando somente a política do sistema, desse modo muito do que foi falado na seção anterior se aplica nessa questão.

### Desafios e decisões

### Simulação

Com o objetivo de fazer a reutilização do código da questão 3 foi decidido manter apenas uma única fila de cliente, onde a mesma é mantida ordenada mantendo os clientes do tipo 1 na frente dos clientes do tipo 2. Deste modo, temos um sistema equivalente a um sistema com duas filas com diferentes prioridades.

### Cadeia de Markov

- Cenário 2 e 4

Para a construção da Cadeia de Markov, precisávamos levar em consideração que o cliente do tipo 1 tinha prioridade sobre o cliente do tipo 2 e a ordem dos clientes na fila não importa. Além disso, não há preempção. A cadeia está representada na Figura 13.

Feito isso, fizemos a implementação do código (descrito na seção Implementação - Cadeia de Markov) para obter a matriz de transição de taxas e depois fazer os cálculos para acharmos os  $\pi$ 's.

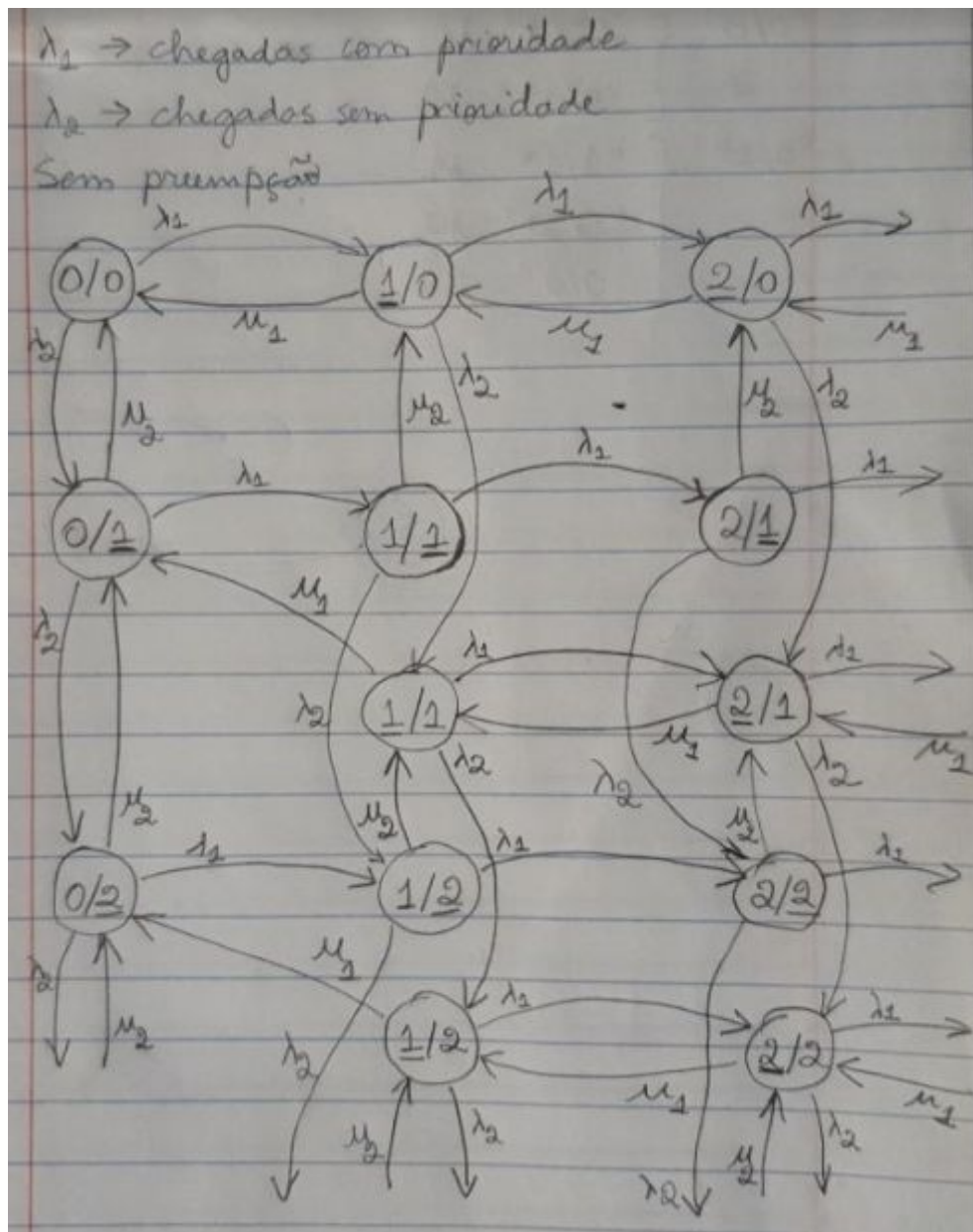


Figura 13: Cadeia de Markov em Tempo Contínuo. Sendo  $\lambda_1 = 0,05, 0,1, 0,15, \dots, 0,9$ ;  $\mu_1 = 1$ ;  $\lambda_2 = 0,2$ ;  $\mu_2 = 0,5$  para o Cenário 2 e  $\lambda_1 = 0,08$ ;  $\mu_1 = 10$ ;  $\lambda_2 = 0,05$ ;  $\mu_2 = 2$  para o Cenário 4. Os estados representam o número de clientes de cada tipo no sistema (do lado esquerdo são representados o número de clientes do tipo 1 e do lado direito). O tipo de cliente que está sendo servido se encontra sublinhado.

As dificuldades que encontramos foram em relação a como representar essa estrutura em código.

- Cenário 3

Para esse cenário, seria preciso trabalhar com uma Cadeia de Markov embutida. Como não tínhamos tempo e a Cadeia de Markov seria um bônus para o trabalho, optamos por não reproduzi-la.

#### Fórmula analítica

Uma dificuldade encontrada nesta questão foi resolvê-la utilizando somente fórmulas fechadas. Foi pensado de várias maneiras diferentes como se deveria fazer tais cálculos e, aparentemente, nenhum deles teve um resultado semelhantes aos feitos através da simulação e da cadeia de Markov. Por fim, chegamos à um consenso de como seria calculado o tempo médio de espera na fila de espera e o número médio de pessoas nessa fila, usando os resultados encontrados por J. Virtamos (autor) em suas notas de aula, visto através das fórmulas a seguir:

$$\bar{W} = \bar{W}_1 p_1 + \bar{W}_2 p_2$$

Fórmula 6: Cálculo do E(W) de maneira condicional

$$\bar{W}_1 = \frac{\bar{R}}{1 - \rho_1} \quad \bar{W}_2 = \frac{\bar{R}}{(1 - \rho_1)(1 - \rho_1 - \rho_2)}$$

Fórmula 7: E(W) para cada tipo de cliente

$$\bar{R} = \frac{1}{2}(\lambda_1 \bar{X}_1^2 + \lambda_2 \bar{X}_2^2)$$

Fórmula 8: Cálculo da vida residual

$$p_1 = \frac{\lambda_1}{\lambda_1 + \lambda_2} \quad p_2 = \frac{\lambda_2}{\lambda_1 + \lambda_2}$$

Fórmula 9: Probabilidade de ser cada tipo de cliente

Os códigos das implementações para as Cadeias de Markov estão no documento a seguir: [implementação - questão 4](#). Aqui faremos um breve resumo dessas implementações em cada cenário analisado.

- Cenário 2

Nessa implementação, os estados da Cadeia de Markov representam o tipo de cliente no sistema e quantos clientes de cada tipo há no sistema. O cliente do tipo 1 tem prioridade sobre o cliente do tipo 2 e não há preempção. O tipo de cliente que está sendo servido é marcado com um asterisco, no código. Para essa cadeia, não precisamos considerar a ordem em que os clientes chegam no sistema.

Como tivemos problemas de eficiência para gerar a matriz de taxas de transição e calcular os respectivos pis de cada estado, tivemos que trabalhar com uma matriz de transição de taxas de tamanho  $3121 \times 321$ . Essa matriz, apesar de parecer grande, ainda é pequena, pois o maior número de pessoas observado no sistema são 78. Assim, geramos todas as combinações possíveis de estados e atribuímos as respectivas taxas para os seus estados adjacentes. Para desenvolver o algoritmo da cadeia, também precisávamos levar em consideração alguns estados importantes como o  $1^*/1$  e o  $1/1^*$ , por exemplo, por conta da prioridade do cliente do tipo 1 sobre o cliente do tipo 2. O primeiro estado representa que um cliente do tipo 1 foi ser servido e chegou um cliente do tipo 2 (não havia, anteriormente, outro cliente do tipo 2 no sistema) ao sistema ou que o cliente do tipo 2 que estava sendo servido saiu (havia dois clientes do tipo 2 no sistema e um deles estava sendo servido e o outro na fila de espera junto com o cliente do tipo 1) e o cliente do tipo 1 passou a ser servido ou um cliente do tipo 1 que estava sendo servido saiu (havia dois cliente do tipo 1 no sistema, sendo 1 no servidor e o outro na fila de espera junto com o cliente do tipo 2) e o outro cliente do tipo 1 passou a ser servido, enquanto que o segundo estado ( $1/1^*$ ) representa o caso em que, como não havia cliente do tipo 1 no sistema, o cliente do tipo 2 foi servido e depois ocorreu uma chegada do cliente do tipo 1 no sistema. Neste último estado ( $1/1^*$ ), fica claro que o cliente do tipo 2 só será servido quando não existir cliente do tipo 1 no sistema.

Após gerar todos os estados e associar as respectivas taxas, será criada a matriz de transição de estados com todos esses valores

obtidos. Com essa matriz, calculamos os  $\pi$ 's utilizando o passo a passo descrito na Seção 2 do trabalho. Ou podemos usar a abordagem de transformá-la em uma matriz de probabilidade de transição (Fórmula 1), para tentar obter melhores resultados, e calcular os  $\pi$ 's obtidos com esta matriz. Em seguida, aplicamos a Fórmula 2 para calcular  $E[Nq]$  e a Fórmula 3 Para calcular  $E[W]$ .

- Cenário 4

A implementação dessa Cadeia de Markov foi similar ao que foi descrito para a Cadeias de Markov do Cenário 2. No entanto, ajustes nos cálculos das taxas de serviços foram feitos, pois a distribuição do tempo de serviço é uniforme para esse cenário.

## Interpretação dos resultados

### Cenário 2

Neste cenário, podemos observar que os resultados gerados pela simulação e pela Cadeia de Markov são bem próximos. Inclusive, a Cadeia de Markov se encontra rodeando os resultados encontrados pelo método analítico. Vale ressaltar, que da mesma forma que ocorre na questão 3 para o  $\lambda_1$  igual a 0.6 tanto  $E[Nq]$ , quanto  $E[W]$  tendem a infinito.

$\lambda$	Simulação: $E[Nq]$ ; (Intervalo de confiança)	Analítico: $E[Nq]$	Cadeia de Markov: $E[Nq]$
0.05	0.36656; (0.36322, 0.36989)	0.04354	0.3701
0.1	0.49734; (0.4926, 0.50209)	0.11111	0.5
0.15	0.66615; (0.66007, 0.67223)	0.2098	0.66438
0.2	0.86651; (0.85869, 0.87432)	0.35	0.875
0.25	1.1414; (1.13042, 1.15237)	0.54762	1.15
0.3	1.50235; (1.48742, 1.51727)	0.82857	1.51905
0.35	2.03154; (2.00898, 2.0541)	1.23846	2.03461
0.4	2.78736; (2.75408, 2.82065)	1.86667	2.79983
0.45	3.99857; (3.93401, 4.06314)	2.91818	4.04901
0.5	6.4193; (6.28612, 6.55249)	5.0	6.40769
0.55	12.31567; (11.89756, 12.73378)	11.12222	11.52693
0.6	31.39325; (30.29, 32.4965)	601.8	21.06448

Tabela 13: Resultados da  $E[Nq]$  obtidos via simulação, método analítico e Cadeia de Markov.

$\lambda$	Simulação: $E[W]$ ; (Intervalo de confiança)	Analítico: $E[W]$	Cadeia de Markov: $E[W]$
0.05	1.46622; (1.45288, 1.47957)	0.17416	1.48038
0.1	1.65781; (1.642, 1.67363)	0.37037	1.66667
0.15	1.90328; (1.88591, 1.92065)	0.59944	1.89823
0.2	2.16627; (2.14672, 2.18581)	0.875	2.1875
0.25	2.53643; (2.51205, 2.56082)	1.21693	2.55556
0.3	3.00469; (2.97485, 3.03454)	1.65714	3.03809
0.35	3.69371; (3.65269, 3.73472)	2.25175	3.69929
0.4	4.64561; (4.59013, 4.70109)	3.11111	4.66638
0.45	6.15165; (6.05233, 6.25098)	4.48951	6.22925
0.5	9.17044; (8.98017, 9.3607)	7.14286	9.15385
0.55	16.42089; (15.86342, 16.97837)	14.82963	15.36924
0.6	39.24156; (37.8625, 40.62063)	752.25	26.3306

Tabela 14: Resultados da  $E[W]$  obtidos via simulação, método analítico e Cadeia de Markov.

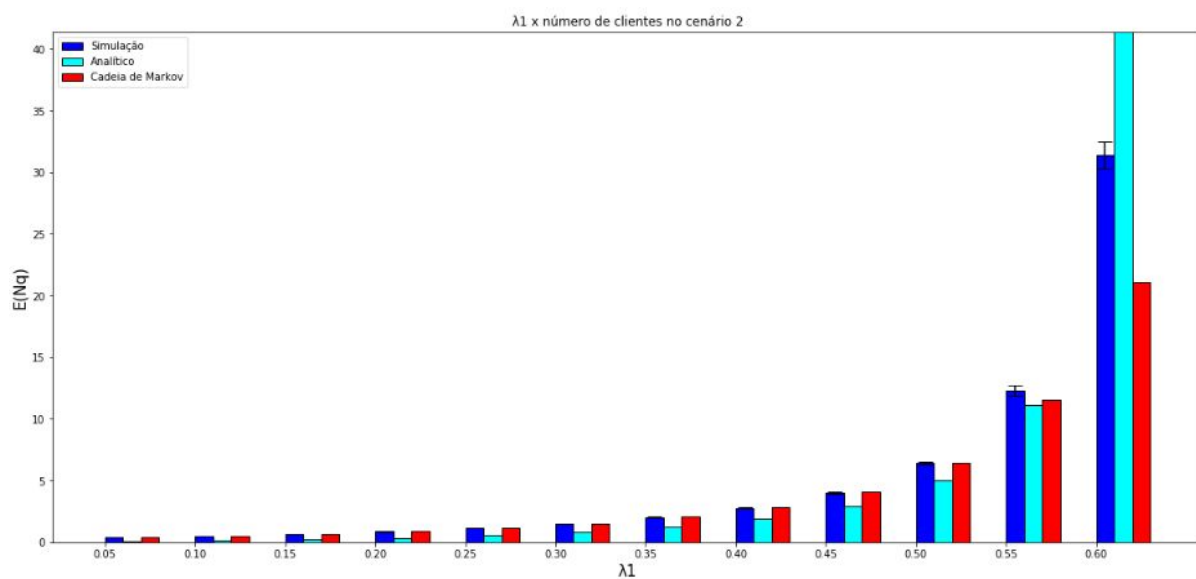


Figura 14: Resultados da  $E[Nq]$  obtidos via simulação (azul escuro), método analítico (azul claro) e Cadeia de Markov (vermelho). No eixo y temos a  $E[Nq]$  e no eixo x as taxas de entrada, variando entre 0.05 a 0.9.

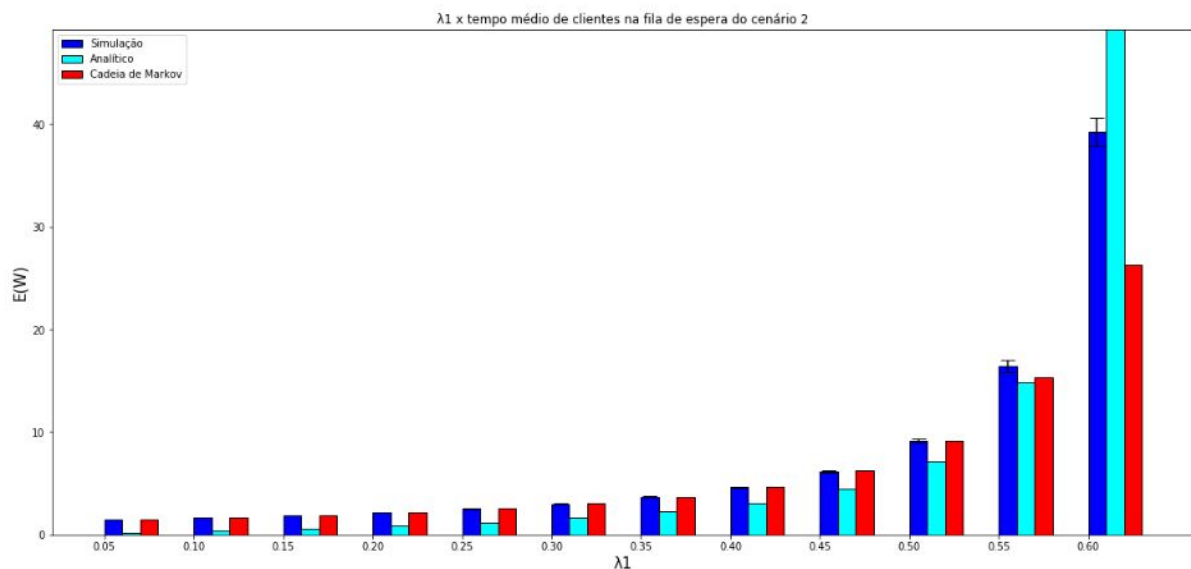




Figura 15: Resultados da  $E[W]$  obtidos via simulação (azul escuro), método analítico (azul claro) e Cadeia de Markov (vermelho). No eixo y temos a  $E[W]$  e no eixo x as taxas de entrada, variando entre 0.05 a 0.9.

### Cenário 3

Assim como na questão 3, não conseguimos fazer a abordagem da Cadeia de Markov, então não temos essa informação para compararmos as 3 frentes. Apesar disso, tivemos resultados relativamente interessantes: a diferença entre o que foi retornado da simulação e dos cálculos analíticos não é grande, mas ao mesmo tempo não é o ideal, visto que o valor analítico não está contido no intervalo de confiança calculado através da simulação.

Acreditamos que, assim como outros casos mencionados previamente, o valor de  $E[Nq]$  cresce tendendo ao infinito quando  $\lambda = 0.6$ , fazendo que o número médio de pessoas na fila de espera seja exorbitante para este  $\lambda$ .

$\lambda$	Simulação: $E[Nq]$ ; (Intervalo de confiança)	Analítico: $E[Nq]$
0.05	0.18466; (0.18329, 0.18603)	0.04354
0.1	0.25003; (0.24834, 0.25172)	0.11111
0.15	0.33007; (0.32795, 0.33219)	0.2098
0.2	0.43974; (0.43679, 0.44269)	0.35
0.25	0.576; (0.57216, 0.57983)	0.54762
0.3	0.76204; (0.75658, 0.7675)	0.82857
0.35	1.01539; (1.00702, 1.02377)	1.23846
0.4	1.39926; (1.38679, 1.41173)	1.86667
0.45	2.01914; (1.99681, 2.04147)	2.91818
0.5	3.21524; (3.16829, 3.26218)	5.0
0.55	6.43095; (6.26625, 6.59566)	11.12222
0.6	22.3273; (21.51473, 23.13988)	601.8

Tabela 15: Resultados da  $E[Nq]$  obtidos via simulação e método analítico.

$\lambda$	Simulação: $E[W]$ ; (Intervalo de confiança)	Analítico: $E[W]$
0.05	0.73864; (0.73315, 0.74413)	0.17416
0.1	0.83343; (0.82779, 0.83907)	0.37037
0.15	0.94306; (0.93699, 0.94912)	0.59944
0.2	1.09935; (1.09197, 1.10674)	0.875
0.25	1.27999; (1.27147, 1.28852)	1.21693
0.3	1.52408; (1.51317, 1.535)	1.65714
0.35	1.84617; (1.83094, 1.8614)	2.25175
0.4	2.3321; (2.31131, 2.35288)	3.11111
0.45	3.10637; (3.07202, 3.14072)	4.48951
0.5	4.59319; (4.52613, 4.66026)	7.14286
0.55	8.5746; (8.355, 8.79421)	14.82963
0.6	27.90913; (26.89341, 28.92485)	752.25

Tabela 16: Resultados da  $E[W]$  obtidos via simulação e método analítico.

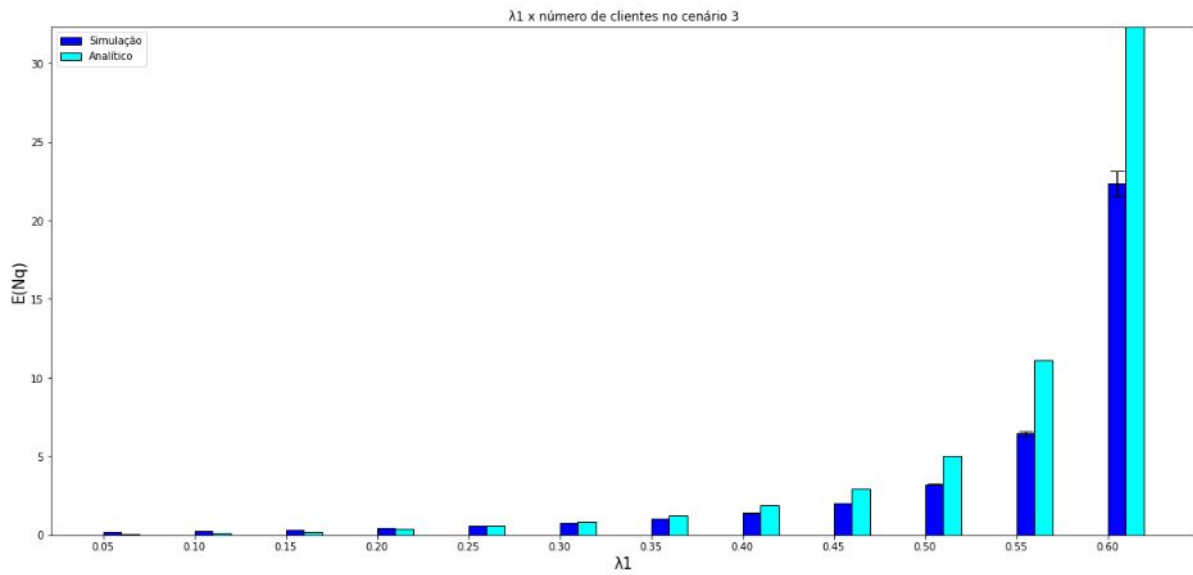


Figura 16: Resultados da  $E[Nq]$  obtidos via simulação (azul escuro e, método analítico (azul claro). No eixo y temos a  $E[Nq]$  e no eixo x as taxas de entrada, variando entre 0.05 a 0.9.

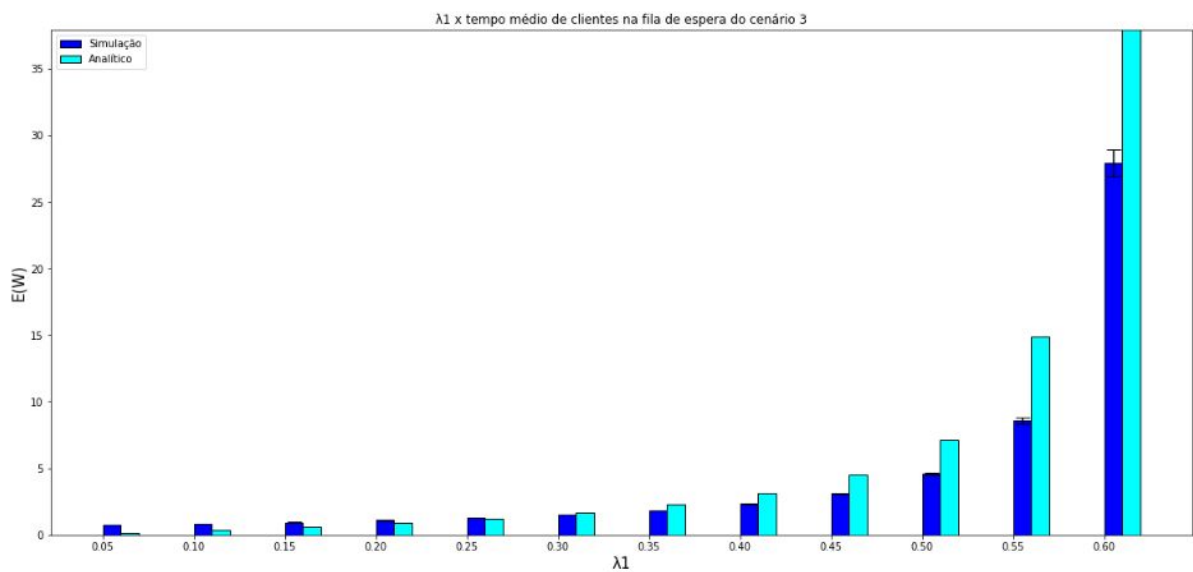


Figura 17: Resultados da  $E[W]$  obtidos via simulação (azul escuro) e método analítico (azul claro). No eixo y temos a  $E[W]$  e no eixo x as taxas de entrada, variando entre 0.05 a 0.9.



## Cenário 4

Assim como na questão 3, esse último cenário não apresentou bons resultados com os 3 métodos divergindo.

$\lambda$	Simulação: $E[Nq]$ ; (Intervalo de confiança)	Analítico: $E[Nq]$	Cadeia de Markov: $E[Nq]$
0.08	9.36877; (8.98503, 9.75252)	13.06933	0.00177

Tabela 17: Resultados da  $E[Nq]$  obtidos via simulação, método analítico e Cadeia de Markov.

$\lambda$	Simulação: $E[W]$ ; (Intervalo de confiança)	Analítico: $E[W]$	Cadeia de Markov: $E[W]$
0.08	72.06749; (69.11558, 75.01939)	100.53333	0.01358

Tabela 18: Resultados da  $E[W]$  obtidos via simulação, método analítico e Cadeia de Markov.

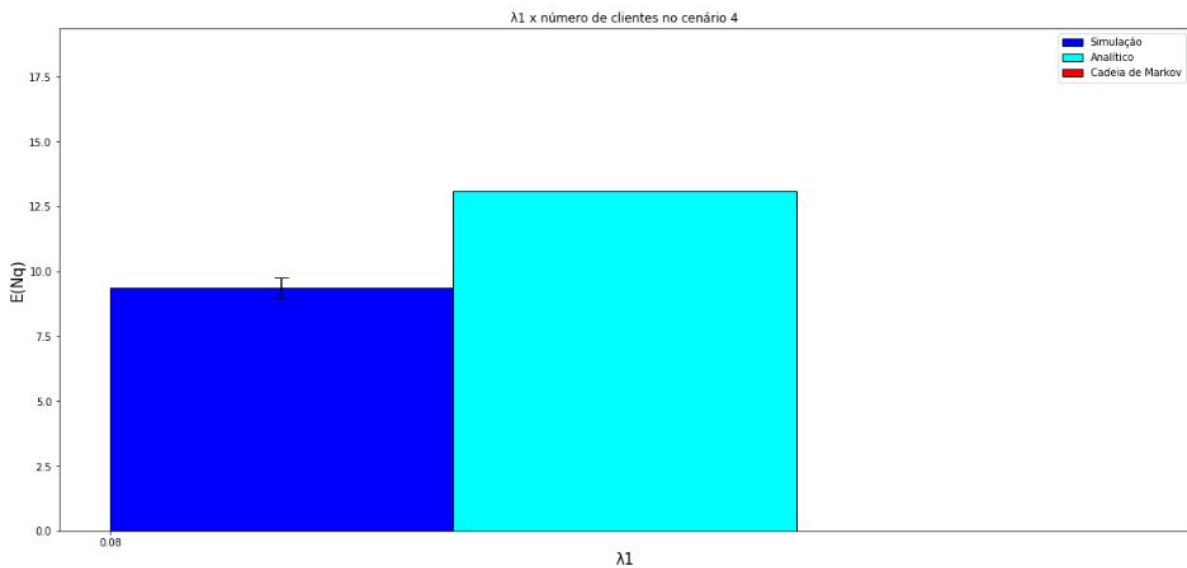


Figura 18: Resultados da  $E[Nq]$  obtidos via simulação (azul escuro), método analítico (azul claro) e Cadeia de Markov (vermelho). No eixo y temos a  $E[Nq]$  e no eixo x as taxas de entrada, variando entre 0.05 a 0.9.

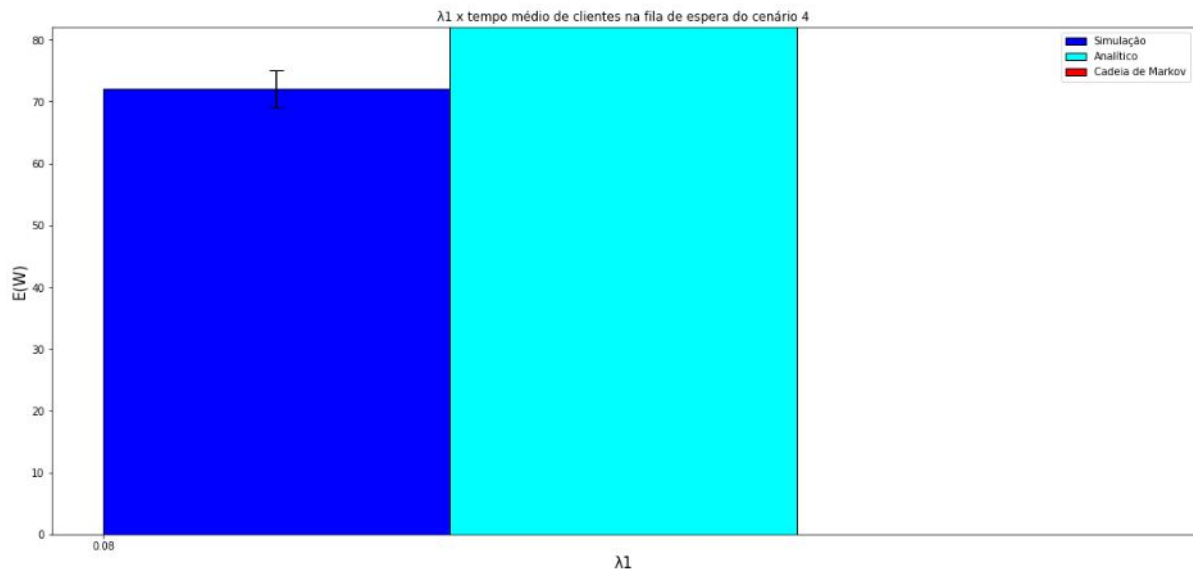


Figura 19: Resultados da  $E[W]$  obtidos via simulação (azul escuro), método analítico (azul claro) e Cadeia de Markov (vermelho). No eixo y temos a  $E[W]$  e no eixo x as taxas de entrada, variando entre 0.05 a 0.9.

## Questão 5 (Lei da conservação)

### Descrição

Temos um sistema de fila similar a questão 3, contudo a noção de fila única é aplicada apenas ao primeiro item, nos demais são utilizadas duas filas. Além disso, o objetivo central é calcular o trabalho pendente de um sistema para diferentes cenários (Fila única; Com preempção; Sem preempção) a fim de comparar nossos resultados com a Lei de Conservação.

Por fim, é requerido uma tabela de comparação para validar os cenários onde a equação abaixo é verdadeira.

$$E[U] = E[Nq_1]E[X_1] + E[Nq_2]E[X_2] + \rho E[X_r]$$

Fórmula 10: Cálculo do trabalho pendente

### Desafios e decisões

#### Simulação

Inicialmente tivemos a necessidade da obtenção dos valores  $E[Nq_1]$  e  $E[Nq_2]$  para o cálculo do trabalho pendente ( $E[U]$ ). Com o intuito de colher esses resultados, foi necessário alterar o código, passando a tratar os clientes de forma distinta na mesma fila (item a) ou criando filas específicas para cada tipo de cliente (item b).

Para a implementação do conceito de preempção é necessário o armazenamento do tempo de serviço residual do cliente interrompido. Deste modo, foi necessária a criação de uma classe “cliente” para associar o tempo residual de serviço, tal tempo residual de serviço só é preenchido quando um cliente é retirado a força do servidor.

Vale ressaltar, que inicialmente era desejado um código de simulação genérico para todos os itens, porém foi decidido pela criação de um código de simulação para cada item, como uma forma de otimizar o trabalho e garantir uma maior organização e corretude do algoritmo.

## Fórmula analítica

Inicialmente estávamos inseguros quanto ao cálculo do  $E[X_r]$  e do  $\rho$  para os diferentes cenários. No entanto, encontramos no material os seguintes cálculos:

$$E[X_r] = \frac{\rho_1 E[X_{r1}]}{\rho} + \frac{\rho_2 E[X_{r2}]}{\rho}$$

Fórmula 11: Cálculo da vida residual

$$\rho = \lambda_1 E[X_1] + \lambda_2 E[X_2]$$

Fórmula 12: Cálculo da utilização do sistema

Sendo  $E[X_{r1}]$  e  $E[X_{r2}]$  a vida residual do cliente do tipo 1 e do cliente do tipo 2, respectivamente;  $E[X_1]$  e  $E[X_2]$  o tempo médio de serviço do cliente do tipo 1 e do cliente do tipo 2, respectivamente; e  $\lambda_1$  e  $\lambda_2$  as taxas de chegadas do cliente do tipo 1 e do cliente do tipo 2.

Considerando o tempo de serviço com distribuição exponencial (Cenário 2), temos que  $E[X_1] = \frac{1}{\mu_1}$  e  $E[X_2] = \frac{1}{\mu_2}$  para o cliente do tipo 1 e para o cliente do tipo 2, respectivamente.  $E[X_{r1}] = \frac{1}{\mu_1}$  e  $E[X_{r2}] = \frac{1}{\mu_2}$  para o cliente do tipo 1 e para o cliente do tipo 2, respectivamente. Sendo  $\mu_1$  e  $\mu_2$  as taxas de serviço para o cliente do tipo 1 e para o cliente do tipo 2.

Considerando o tempo de serviço com distribuição determinística (Cenário 3), temos que  $E[X_1] = \frac{1}{\mu_1}$  e  $E[X_2] = \frac{1}{\mu_2}$  para o cliente do tipo 1 e para o cliente do tipo 2, respectivamente.  $E[X_{r1}] = \frac{1}{2\mu_1}$  e  $E[X_{r2}] = \frac{1}{2\mu_2}$  para o cliente do tipo 1 e para o cliente do tipo 2, respectivamente. Sendo  $\mu_1$  e  $\mu_2$  as taxas de serviço para o cliente do tipo 1 e para o cliente do tipo 2.

Todos esses parâmetros são utilizados para calcular o trabalho pendente em cada cenário. Utilizamos a seguinte fórmula do trabalho pendente (essa fórmula é válida para todas as políticas abordadas no item a, b e c da Questão 5):

$$E[U] = \frac{\rho E[X_r]}{(1-\rho)}$$

Fórmula 13: Cálculo do trabalho pendente

## Implementação

Para esse desenvolvimento utilizamos como código base o nosso simulador desenvolvido na questão 3, visto que esse item utiliza de cenários da questão 3 para provar a Lei de Conservação. Logo, fizemos poucas mudanças no nosso simulador base (listadas na seção Desafios).

Algumas mudanças:

- Criação de filas para cada tipo de cliente. (item b e c).
- Criação de uma classe cliente, para armazenar o tempo de serviço residual (item b).
- Adaptação para o retorno das médias de  $Nq1$ ,  $Nq2$ ,  $\rho1$ ,  $\rho2$ ,  $X1$  e  $X2$ .
- Possibilidade de preempção, quando ocorre a chegada de um cliente do tipo 1 e há um cliente do tipo 2 no servidor. (item b).

Vale ressaltar que para cada item, existe um simulador base, de modo que os diferentes cenários (cenário 2 e 3) chamam o mesmo simulador com os devidos parâmetros.

## Discussão

Num primeiro momento, tivemos uma certa dificuldade em interpretar o que estava sendo pedido na questão, pois no início do enunciado é dito para considerarmos o mesmo sistema da questão 3 (fila única e sem prioridade), porém, em seguida, é falado que temos uma fila para cada tipo de cliente e cenários preemptivos e não-preemptivos. Optamos por utilizar como base para o desenvolvimento desta questão um sistema com prioridade de classes para que pudéssemos utilizar o conceito da preempção e com duas filas (uma para cada classe).

Ao decorrer do nosso desenvolvimento, tivemos algumas dúvidas em como implementar a solução da simulação: se o melhor a fazer seria calcular essa solução utilizando somente o método de simulação, em que os parâmetros são totalmente retirados da simulação, ou de um modo híbrido, onde estes provêm da combinação dos resultados da simulação com as fórmulas fechadas e analíticas. Ao realizar os cálculos para ambas as frentes de raciocínio, puramente simulação e modo híbrido, foi observado que os resultados gerados estão bem distantes um do outro, o que nos causou uma grande estranheza visto que acreditamos que tais soluções deveriam ser próximas.

Após um debate, chegamos a conclusão de que o método híbrido seria de melhor valia, visto que por ele utilizar variáveis que são calculadas na própria simulação, acaba aproximando ainda mais todos os valores.

## Interpretação dos resultados

Analisando os diferentes cenários em cada item dessa questão, provamos que a fórmula 2 (trabalho pendente) da Questão 5 apresentou os mesmos valores para ambos os cenários dos itens 'a' e 'c'. No entanto, os cenários do item 'b' não bateram, o que é um resultado estranho, uma vez que essa fórmula é válida para todas as políticas tratadas nesses itens. Acreditamos que isso tenha acontecido por conta de alguma falha ou situação que deixamos de representar no simulador preemptivo.

Por fim, a fórmula 3 (trabalho pendente) da Questão 5 apresentou ser válida apenas para os cenários 2 e 3 do item a. No entanto, essa fórmula também deveria ser válida para o item 'c'. Assim, acreditamos ter cometido algum erro ao calcularmos essa fórmula para os cenários 2 e 3 do item c.

### Item a

#### Cenário 2

$\lambda_1$	$E(U) = (\rho \cdot X_r) / (1 - \rho)$ - Simulação	$E[Nq_1]E[X_1] + E[Nq_2]E[X_2] + \rho E[X_r]$ - Analítico
0.05	1.5455; (1.5418, 1.5492)	1.50335
0.1	1.80022; (1.79564, 1.8048)	1.71111
0.15	2.11201; (2.1054, 2.11861)	1.96993
0.2	2.49879; (2.48986, 2.50773)	2.3
0.25	3.01124; (2.99922, 3.02326)	2.73333
0.3	3.6747; (3.65578, 3.69363)	3.32381
0.35	4.6077; (4.58028, 4.63512)	4.16923
0.4	6.02527; (5.9789, 6.07165)	5.46667
0.45	8.49167; (8.3924, 8.59095)	7.67879
0.5	13.65314; (13.38578, 13.9205)	12.2
0.55	36.766; (26.682, 46.85)	26.02222
0.6	196.73235; (168.6316, 224.83311)	140.886

Tabela 19: Resultados da  $E[U]$  obtidos via simulação (à esquerda) usando a fórmula 2 da Questão 5 e via método analítico (à direita) usando a fórmula 3 da Questão 5.

E[Nq1]E[X1] + E[Nq2]E[X2] + ρE[Xr] - Simulação		
1.4492;	(1.44191,	1.4565)
1.62204;	(1.61301,	1.63108)
1.84629;	(1.83363,	1.85896)
2.14212;	(2.12647,	2.15777)
2.56165;	(2.54088,	2.58241)
3.13358;	(3.10232,	3.16483)
3.94021;	(3.89608,	3.98435)
5.21197;	(5.13934,	5.28461)
7.37799;	(7.24474,	7.51123)
11.48883;	(11.2252,	11.75245)
21.94112;	(21.1726,	22.70965)
54.93857;	(52.85991,	57.01724)

Tabela 20: Resultados da E[U] obtidos via simulação usando a fórmula 3 da Questão 5

### Cenário 3

λ1	E(U) = (ρ*Xr)/(1-ρ) - Simulação	E[Nq1]E[X1] + E[Nq2]E[X2] + ρE[Xr] - Analítico
0.05	0.77297; (0.77169, 0.77425)	0.75299
0.1	0.90022; (0.89855, 0.90189)	0.86111
0.15	1.05532; (1.05304, 1.0576)	0.9982
0.2	1.24941; (1.24632, 1.2525)	1.175
0.25	1.50352; (1.49919, 1.50785)	1.40833
0.3	1.83818; (1.83179, 1.84457)	1.72619
0.35	2.30321; (2.29323, 2.31319)	2.17885
0.4	3.00642; (2.99021, 3.02264)	2.86667
0.45	4.18859; (4.15641, 4.22077)	4.02348
0.5	6.56092; (6.48559, 6.63626)	6.35
0.55	15.26565; (14.26287, 16.26843)	13.34722
0.6	180.21696; (125.30355, 235.13038)	700.8993

Tabela 21: Resultados da E[U] obtidos via simulação (à esquerda) usando a fórmula 2 da Questão 5 e via método analítico (à direita) usando a fórmula 3 da Questão 5.

+-----+ E[Nq1]E[X1] + E[Nq2]E[X2] + ρE[Xr] - Simulação		
	0.71985; (0.7171, 0.7226)	
	0.78899; (0.78571, 0.79227)	
	0.88641; (0.88222, 0.8906)	
	1.01338; (1.0079, 1.01886)	
	1.19919; (1.19184, 1.20653)	
	1.4486; (1.43839, 1.4588)	
	1.81646; (1.80064, 1.83228)	
	2.37968; (2.35577, 2.40359)	
	3.40355; (3.35963, 3.44747)	
	5.42023; (5.32034, 5.52012)	
	11.48778; (11.11393, 11.86164)	
	37.32468; (35.93361, 38.71576)	
+-----+		

Tabela 22: Resultados da E[U] obtidos via simulação usando a fórmula 3 da Questão 5

Item b

Cenário 2

+-----+ λ1   E(U) = (ρ*Xr)/(1-ρ) - Simulação   E[Nq1]E[X1] + E[Nq2]E[X2] + ρE[Xr] - Analítico		
	0.05   2.58911; (2.56887, 2.60934)	1.54545
	0.1   2.96697; (2.94268, 2.99127)	1.8
	0.15   3.42748; (3.39527, 3.45968)	2.11111
	0.2   4.02718; (3.98561, 4.06874)	2.5
	0.25   4.8481; (4.78717, 4.90903)	3.0
	0.3   6.03075; (5.93911, 6.1224)	3.66667
	0.35   7.58084; (7.43335, 7.72832)	4.6
	0.4   10.36617; (10.03992, 10.69243)	6.0
	0.45   20.41876; (7.64037, 33.19715)	8.33333
	0.5   48.19511; (-58.08881, 154.47903)	13.0
	0.55   35.29654; (8.06394, 62.52913)	27.0
	0.6   -11.04733; (-59.1787, 37.08404)	142.065
+-----+		

Tabela 23: Resultados da E[U] obtidos via simulação (à esquerda) usando a fórmula 2 da Questão 5 e via método analítico (à direita) usando a fórmula 3 da Questão 5.



E[Nq1]E[X1] + E[Nq2]E[X2] + ρE[Xr] - Simulação		
0.90945;	(0.90862,	0.91028)
0.97135;	(0.97039,	0.97231)
1.03975;	(1.03858,	1.04093)
1.11776;	(1.11632,	1.11921)
1.20543;	(1.20342,	1.20744)
1.31267;	(1.31001,	1.31533)
1.43801;	(1.43442,	1.4416)
1.60071;	(1.59571,	1.60571)
1.80048;	(1.79349,	1.80748)
2.07136;	(2.06158,	2.08114)
2.42932;	(2.41509,	2.44354)
2.90687;	(2.88545,	2.9283)

Tabela 24: Resultados da E[U] obtidos via simulação usando a fórmula 3 da Questão 5

### Cenário 3

λ1	E(U) = (ρ*Xr)/(1-ρ) - Simulação	E[Nq1]E[X1] + E[Nq2]E[X2] + ρE[Xr] - Analítico
0.05	1.33688; (1.32865, 1.3451)	1.12045
0.1	1.71957; (1.70637, 1.73276)	1.35
0.15	2.33092; (2.30494, 2.35689)	1.63611
0.2	3.36191; (3.30247, 3.42136)	2.0
0.25	3.19221; (-0.85758, 7.242)	2.475
0.3	11.58757; (6.45408, 16.72106)	3.11667
0.35	-15.74072; (-73.82097, 42.33952)	4.025
0.4	-14.68336; (-39.50556, 10.13884)	5.4
0.45	315.52933; (-301.34854, 932.40721)	7.70833
0.5	-7.38063; (-8.71436, -6.04689)	12.35
0.55	-5.01275; (-5.12305, -4.90244)	26.325
0.6	-4.02693; (-4.09237, -3.9615)	141.3783

Tabela 25: Resultados da E[U] obtidos via simulação (à esquerda) usando a fórmula 2 da Questão 5 e via método analítico (à direita) usando a fórmula 3 da Questão 5.

E[Nq1]E[X1] + E[Nq2]E[X2] + ρE[Xr] - Simulação		
	0.44352;	(0.44331, 0.44373)
	0.47718;	(0.47689, 0.47746)
	0.51283;	(0.51244, 0.51321)
	0.5514;	(0.55089, 0.55191)
	0.59364;	(0.59297, 0.59431)
	0.64297;	(0.64205, 0.64389)
	0.69913;	(0.69789, 0.70036)
	0.76843;	(0.76677, 0.7701)
	0.85497;	(0.85275, 0.85718)
	0.96969;	(0.96637, 0.973)
	1.12517;	(1.12053, 1.12981)
	1.34374;	(1.33656, 1.35092)

Tabela 26: Resultados da E[U] obtidos via simulação usando a fórmula 3 da Questão 5

Item c

Cenário 2

λ1	E(U) = (ρ*Xr)/(1-ρ) - Simulação	E[Nq1]E[X1] + E[Nq2]E[X2] + ρE[Xr] - Analítico
0.05	1.5474; (1.54381, 1.55098)	1.50335
0.1	1.79917; (1.79447, 1.80387)	1.71111
0.15	2.11108; (2.10461, 2.11755)	1.96993
0.2	2.49895; (2.49027, 2.50762)	2.3
0.25	3.00928; (2.99698, 3.02158)	2.73333
0.3	3.68554; (3.66683, 3.70425)	3.32381
0.35	4.60685; (4.57856, 4.63513)	4.16923
0.4	6.0723; (6.02118, 6.12343)	5.46667
0.45	8.44712; (8.3506, 8.54364)	7.67879
0.5	13.83385; (13.45194, 14.21575)	12.2
0.55	31.16886; (28.48993, 33.8478)	26.02222
0.6	251.92625; (177.89335, 325.95915)	140.886

Tabela 27: Resultados da E[U] obtidos via simulação (à esquerda) usando a fórmula 2 da Questão 5 e via método analítico (à direita) usando a fórmula 3 da Questão 5.

E[Nq1]E[X1] + E[Nq2]E[X2] + ρE[Xr] - Simulação		
1.13154; (1.12608, 1.137)		
1.27894; (1.27178, 1.28609)		
1.47471; (1.46438, 1.48504)		
1.7294; (1.71589, 1.74292)		
2.07085; (2.05207, 2.08963)		
2.59039; (2.56134, 2.61945)		
3.2766; (3.23372, 3.31948)		
4.50802; (4.43342, 4.58262)		
6.418; (6.29296, 6.54304)		
10.74949; (10.45184, 11.04714)		
20.25571; (19.54801, 20.9634)		
54.01286; (51.9562, 56.06951)		

Tabela 28: Resultados da E[U] obtidos via simulação usando a fórmula 3 da Questão 5

### Cenário 3

λ1	E(U) = (ρ*Xr)/(1-ρ) - Simulação	E[Nq1]E[X1] + E[Nq2]E[X2] + ρE[Xr] - Analítico
0.05	0.7719; (0.77063, 0.77317)	1.07835
0.1	0.90036; (0.89859, 0.90212)	1.26111
0.15	1.05595; (1.05372, 1.05819)	1.49493
0.2	1.25148; (1.24832, 1.25464)	1.8
0.25	1.49949; (1.49531, 1.50366)	2.20833
0.3	1.83644; (1.83013, 1.84274)	2.77381
0.35	2.30106; (2.29139, 2.31073)	3.59423
0.4	3.00809; (2.99176, 3.02442)	4.86667
0.45	4.1775; (4.14621, 4.20879)	7.05379
0.5	6.67383; (6.5892, 6.75846)	11.55
0.55	14.80952; (14.10566, 15.51337)	25.34722
0.6	1257.95295; (-632.45261, 3148.35852)	140.1993

Tabela 29: Resultados da E[U] obtidos via simulação (à esquerda) usando a fórmula 2 da Questão 5 e via método analítico (à direita) usando a fórmula 3 da Questão 5.

+-----+		
E[Nq1]E[X1] + E[Nq2]E[X2] + $\rho$ E[Xr] - Simulação		
+-----+		
	0.50185; (0.50041, 0.50328)	
	0.55702; (0.55504, 0.559)	
	0.62753; (0.62471, 0.63035)	
	0.72442; (0.72039, 0.72844)	
	0.8507; (0.84518, 0.85621)	
	1.04264; (1.03392, 1.05137)	
	1.33969; (1.32567, 1.35371)	
	1.79819; (1.77616, 1.82021)	
	2.71858; (2.67659, 2.76056)	
	4.7431; (4.6356, 4.8506)	
	10.47854; (10.12836, 10.82872)	
	37.21524; (35.84286, 38.58762)	
+-----+		

Tabela 30: Resultados da E[U] obtidos via simulação usando a fórmula 3 da Questão 5

## Questão 6 (Simulação para verificar os resultados analíticos)

### Desafios e decisões

Para a análise destes resultados foi preciso incrementar o conjunto de variáveis analisadas do simulador usado anteriormente. Assim, foi reaproveitado quase todo simulador das questões anteriores.

Porém a fim de justificar os itens 10 e 11, foi criado um simulador simplificado para FCFS e LCFS a fim de gerar as distribuições pedidas.

Outro desafio foi propagar o intervalo de confiança a uma expressão matemática.

Para obter tal estimador, pegamos o resultado da função matemática aplicada às médias e diminuimos ela do valor da função aplicada no valor máximo do desvio padrão, assim chegando no valor para a distância positiva e negativa da média, estimando assim o novo desvio padrão.

Utilizamos  $\lambda_1 = 0.4$ ,  $\lambda_2 = 0.2$ ,  $\mu_1 = 1$ ,  $\mu_2 = 0.5$ .

### Implementação

Para os itens 10 e 11, foi implementado um simulador simples para assim validar os resultados. Para isso, são simuladas filas M/M/1 FCFS e LCFS e as únicas métricas de interesse são o tempo no sistema e a quantidade de pessoas no sistema.

Para isso, foi feito um simulador que sorteia se o próximo evento é uma chegada ou uma saída, e com esse resultado manipula uma fila e quem está sendo servido.

### Interpretação dos resultados

#### Item 1

$$E[W] = p_1 E[W_1] + p_2 E[W_2]$$

Resultado esperado:

Certo.  $E[W]$  é a média ponderada dos tempos de espera dos clientes do tipo 1 e 2.

Resultado Obtido:

$$E[W] = p_1 E[W_1] + p_2 E[W_2]$$

$$E[W] = 4.5717627439425215 \quad + - \quad 0.060341607787303124$$

$$p_1 E[W_1] + p_2 E[W_2] = 4.673689741703605 \quad + - \quad 0.0840723637705354$$

## Item 2

$$E[W_1] = \rho_1 E[X_{1r}] / (1 - \rho_1)$$

Resultado esperado:

Errado. Essa relação só vale para a fila virtual. Ela não vale para os clientes do tipo 1 em particular isoladamente.

Resultado Obtido:

$$E[W_1] = \rho_1 E[X_{1r}] / (1 - \rho_1)$$

$$E[W_1] = 1.9943608989705208 \quad + - \quad 0.010818169654673615$$

$$\rho_1 E[X_{1r}] / (1 - \rho_1) = 0.6644880737330358 \quad + - \quad 0.004295808876271012$$

## Item 3

$$E[W] = \rho E[X_r] / (1 - \rho)$$

Resultado esperado:

Errado. O tempo médio de espera dos clientes é diferente do serviço pendente encontrado por eles. Do lado direito da equação temos o serviço pendente, mas do lado esquerdo o tempo de espera na fila de espera, que deveria ser calculado usando o resultado do item 1.

Resultado Obtido:

$$E[W] = \rho E[X_r] / (1 - \rho)$$

$$E[W] = 4.574447550269498 \quad + - \quad 0.05791517058992681$$

$$\rho E[X_r] / (1 - \rho) = 5.9496313232841 \quad + - \quad 0.0715749858270609$$

## Item 4

$$E[U] = \rho E[X_r] / (1 - \rho)$$

Resultado esperado:

Certo. Agora sim, correto.

Resultado Obtido:

$$E[U] = \rho E[X_r] / (1 - \rho)$$

$$E[U] = 6.019475061097268 \quad + - \quad 0.08311620190259747$$

$$\rho E[X_r] / (1 - \rho) = 5.995562276556859 \quad + - \quad 0.07543966879652952$$

## Item 5

$$E[X] = (\rho_1/\rho)E[X_1] + (\rho_2/\rho)E[X_2]$$

Resultado esperado:

Errado. Os pesos deveriam ser  $p_1$  e  $p_2$  e não  $(\rho_1/\rho)$  e  $(\rho_2/\rho)$ .

Resultado Obtido:

$$\begin{aligned}E[X] &= (\rho_1/\rho)E[X_1] + (\rho_2/\rho)E[X_2] \\E[X] &= 1.3325144150605115 \quad + - \quad 0.0018927642793335388 \\p_1/p \ E[X_1] + p_2/p \ E[X_2] &= 1.4988487932754395 \quad + - \\&0.0012431882794490257\end{aligned}$$

Item 6

$$E[X_r] = (\rho_1/\rho)E[X_{r1}] + (\rho_2/\rho)E[X_{r2}]$$

Resultado esperado:

Certo.  $E[X_r]$  é calculado condicionado que o sistema esteja ocupado.

Resultado Obtido:

$$\begin{aligned}E[X_r] &= (\rho_1/\rho)E[X_{r1}] + (\rho_2/\rho)E[X_{r2}] \\E[X_r] &= 1.50222982014645 \quad + - \quad 0.004407165149585905 \\(p_1/p) \ E[X_{r1}] + (p_2/p) \ E[X_{r2}] &= 1.5009626823612248 \quad + - \\&0.006529172636894565\end{aligned}$$

Item 7

$$E[G] = E[X]/(1 - \rho)$$

Resultado esperado:

Certo. Essa relação é sempre válida para sistemas conservativos em trabalho.

Resultado Obtido:

$$\begin{aligned}E[G] &= E[X]/(1 - \rho) \\E[G] &= 6.637338027007191 \quad + - \quad 0.059598427492333524 \\E[X]/(1 - \rho) &= 6.613229703828735 \quad + - \quad 0.05824063174261074\end{aligned}$$

Item 8

$$E[U] = \rho_1 E[W_1] + \rho_2 E[W_2] + \rho E[X_r] = \lambda_1 E[X_1] E[W_1] + \lambda_2 E[X_2] E[W_2] + \rho E[X_r]$$

Resultado esperado:

Certo. Essa é a lei da conservação.

Resultado Obtido:

$$E[U] = \rho_1 E[W_1] + \rho_2 E[W_2] + \rho E[X_r] = \lambda_1 E[X_1] E[W_1] + \lambda_2 E[X_2] E[W_2] + \rho E[X_r]$$

$$E[U] = 6.019475061097268 \quad + - \quad 0.08311620190259747$$

$$\rho_1 E[W_1] + \rho_2 E[W_2] + \rho E[X_r] = 6.006639240781522 \quad + - \quad 0.09300317697874849$$

$$\lambda_1 E[X_1] E[W_1] + \lambda_2 E[X_2] E[W_2] + \rho E[X_r] = 6.005704962930188 \quad + - \quad 0.0786679029816666$$

#### Item 9

$$\rho_1 E[W_1] + \rho_2 E[W_2] + \rho E[X_r] = \rho_1 / \rho E[W_1] + \rho_2 / \rho E[W_2] = \rho E[X_r] / (1 - \rho) \Rightarrow \rho_1 E[W_1] + \rho_2 E[W_2] = \rho^2 E[X_r] / 1 - \rho$$

Resultado esperado:

Certo. Essa é a lei da conservação.

Resultado Obtido:

$$\rho_1 E[W_1] + \rho_2 E[W_2] + \rho E[X_r] = \rho_1 / \rho E[W_1] + \rho_2 / \rho E[W_2] = \rho E[X_r] / (1 - \rho) \Rightarrow \rho_1 E[W_1] + \rho_2 E[W_2] = \rho^2 E[X_r] / (1 - \rho)$$

$$\rho_1 E[W_1] + \rho_2 E[W_2] + \rho E[X_r] = 6.006639240781522 \quad + - \quad 0.09300317697874849$$

$$\rho_1 / \rho E[W_1] + \rho_2 / \rho E[W_2] = 6.009414651194662 \quad + - \quad 0.09733545577291913$$

$$\rho E[X_r] / (1 - \rho) = 5.995562276556859 \quad + - \quad 0.07543966879652952$$

$$\rho_1 E[W_1] + \rho_2 E[W_2] = 4.8053906166233515 \quad + - \quad 0.08717448425957652$$

$$\rho^2 E[X_r] / (1 - \rho) = 4.794313652398689 \quad + - \quad 0.06961097607735756$$

#### Item 10

A distribuição do número de clientes no sistema é invariante com relação ao fato de a política de serviço ser FCFS, LCFS sem interrupção, ou FCFS com diferentes classes de clientes, mais uma vez sem interrupção.

Resultado esperado:

Errado. O número médio de clientes é o mesmo segundo FCFS e LCFS sem interrupção, mas não em FCFS com 2 classes.



#### Resultado Obtido:

```
E[Nfcfs_1_classe] 1.1624720000000002 +- 0.07205076102895663  
E[Nlcfs] = 1.159182 +- 0.07184684471631832  
E[Nfcfs_2_classes] 4.05355440406667 +- 0.2512419049106407
```

#### Item 11

A distribuição do tempo dos clientes no sistema é invariante com relação ao fato de a política de serviço ser FCFS, LCFS sem interrupção, ou FCFS com diferentes classes de clientes, mais uma vez sem interrupção.

#### Resultado esperado:

Errado. O tempo dos clientes no sistema depende da política de serviço.

#### Resultado Obtido:

```
E[Tfcfs_1_classe] 1.6629573531930042 +- 0.10307116460117349  
E[Tlcfs] = 1.6634715909484916 +- 0.10310303738747088  
E[Tfcfs_2_classes] = 5.872503068765522 +- 0.06102704549636839
```

# Conclusão

Podemos concluir que o desenvolvimento deste trabalho reforçou muitos dos conhecimentos obtidos em sala de aula e ajudou a consolidar vários outros, pois fica evidente que encontramos diversos desafios no decorrer desse desenvolvimento e tivemos que tomar decisões “críticas” puramente baseadas nos nossos conhecimentos e no que achamos ser a melhor resposta para cada situação.

Ao construir os simuladores para cada questão, tivemos que verificar vários conceitos teóricos a fim de organizar a melhor forma de usá los, de modo que sentimos um pouco da dificuldade de estabelecer uma conexão forte e bem estruturada com os conhecimentos teóricos e a aplicação prática desses conceitos, já que essa foi uma experiência totalmente nova para a maior parte do grupo.

Quando nos deparamos com resultados inusitados ou dúvidas sobre a forma de como fazer os nossos simuladores, tivemos que organizar muito bem os nossos conceitos e pensar de forma clara sobre qual seria o objetivo do cenário em questão, de modo a levantar as melhores decisões para contornar o devido problema.

Em alguns casos não podemos afirmar com certeza absoluta se a decisão que tomamos foi a melhor e/ou mais eficiente para solucionar o problema, mas discutimos intensamente cada uma dessas decisões e todas elas foram um consenso geral do grupo, tomadas em cima de todo o conteúdo teórico disponível da disciplina e experiência dos membros.

As Cadeias de Markov também foram um desafio, principalmente quando estávamos lidando com dois tipos de clientes. Pensar em como montar a cadeia, o que cada estado deveria representar e conseguir visualizar todas as possibilidades envolvidas foi bastante agregador para consolidarmos o nosso conhecimento.

Torna-se evidente, portanto, que a tarefa de desenvolver esses simuladores foi bem desafiadora, pois em todos os casos tivemos que aplicar conhecimentos que abrangem toda a disciplina e diversas decisões tiveram que ser tomadas para garantir o que julgamos ser a melhor abordagem para cada problema. Ao nos depararmos com um novo obstáculo, supusemos muitos cenários e tivemos um grande aprendizado ao filtrar todas as possibilidades levantadas com o objetivo de garantir a correteza do nosso trabalho.

# Referências bibliográficas

Avaliação e Desempenho - Apostila. <https://dcc.ufrj.br/~sadoc/ad20161/apostila.pdf>. Último acesso em 22 Dez 2019

Avaliação e Desempenho - Apostila Extra. <https://dcc.ufrj.br/~sadoc/ad20161/apostilaextra.pdf>. Último acesso em 22 Dez 2019

Little's Result. <https://www.youtube.com/watch?v=p1sG7mm1lxo>. Último acesso em 22 Dez 2019

Simulation - Performance Evaluation. [https://www.youtube.com/watch?v=E1\\_Mne9nTus](https://www.youtube.com/watch?v=E1_Mne9nTus). Último acesso em 22 Dez 2019

Central Limit Theorem. [https://www.youtube.com/watch?v=zEwT\\_flpSBE](https://www.youtube.com/watch?v=zEwT_flpSBE). Último acesso em 22 Dez 2019

Inequalities Markov and Chebyshev, Law of Large Numbers. <https://www.youtube.com/watch?v=QZ0s8tx2SPo>. Último acesso em 22 Dez 2019

Residual life. <https://www.youtube.com/watch?v=rwAL4pYHqzo>. Último acesso em 22 Dez 2019

Busy period in Queuing Systems. <https://www.youtube.com/watch?v=smd7RSYtwSc>. Último acesso em 22 Dez 2019

M/G/1 Queue with Vacations. <https://www.youtube.com/watch?v=ULQ646sAn9U>. Último acesso em 22 Dez 2019

Queues: Average waiting time, stability and the embedded chain. <https://www.youtube.com/watch?v=9XVV3CBhqGA>. Último acesso em 22 Dez 2019

Queues: Average waiting time, stability and the embedded chain. <https://www.youtube.com/watch?v=9XVV3CBhqGA>. Último acesso em 22 Dez 2019

Queue Theory - Queues with priority. <https://www.youtube.com/watch?v=H8OLRhmlvEM>. Último acesso em 22 Dez 2019

PageRank and Markov Chains Discrete Time. <https://www.youtube.com/watch?v=lziPo9cnCSc>. Último acesso em 22 Dez 2019

Classification of Markov chains states. <https://www.youtube.com/watch?v=qfxTVf1myCg>. Último acesso em 22 Dez 2019

Birth and death process (CMTc). <https://www.youtube.com/watch?v=9cb4H4lXb4k>. Último acesso em 22 Dez 2019

CMTc - Continuous Time Markov Chains. [https://www.youtube.com/watch?v=-jU-\\_wfckOo](https://www.youtube.com/watch?v=-jU-_wfckOo). Último acesso em 22 Dez 2019

Notas de aula da disciplina “Queueing Theory”. J. Virtamo. <http://www.netlab.tkk.fi/opetus/s383143/kalvot/english.shtml>. Último acesso em 22 Dez 2019.