

Simulação Discreta - Sistema de Filas

Giovana Nogueira Oliveira,
Pedro Henrique Alves Barbosa

September 23, 2024

Abstract

Este relatório apresenta a análise de uma simulação discreta de um sistema de filas, que é um dos tópicos clássicos em teoria de filas e simulação. A simulação implementada visa analisar o desempenho do sistema sob diferentes taxas de ocupação, utilizando a fórmula de Little, além de calcular métricas importantes como o tempo médio de espera ($E[W]$), o número médio de clientes no sistema ($E[N]$), e a taxa de chegada (λ).

1 Descrição do Sistema

O sistema simulado é uma fila simples com chegadas de clientes distribuídas exponencialmente, o que caracteriza um sistema do tipo M/M/1 (chegadas de Poisson, tempos de serviço exponenciais, e uma única fila). O objetivo da simulação é avaliar o comportamento da fila sob diferentes níveis de ocupação, sendo eles 85%, 90%, 95% e 99%.

2 Fórmulas e Parâmetros

2.1 Variáveis de Entrada

- **Taxa média de chegada (λ):** 5.0 clientes por unidade de tempo.
- **Capacidade do servidor:** Calculada com base no nível de ocupação desejado, ou seja,

$$\text{capacidade} = \frac{\lambda}{\text{ocupação}}$$

- **Tempo de simulação:** 100.000 segundos.
- **Taxa de ocupação:** Foram testados quatro cenários de ocupação: 0.85, 0.90, 0.95 e 0.99.

2.2 Fórmula de Little

A fórmula de Little é uma das leis fundamentais da teoria das filas. Ela se aplica a sistemas estáveis e diz que:

$$E[N] = \lambda \cdot E[W] \tag{1}$$

Onde:

- $E[N]$ é o número médio de clientes no sistema.
- λ é a taxa média de chegada de clientes.
- $E[W]$ é o tempo médio que um cliente passa no sistema.

3 Estrutura do Código

3.1 Função *uniforme()*

Esta função gera um número aleatório em uma distribuição uniforme entre 0 e 1. O valor gerado é transformado para garantir que esteja sempre entre 0 e 1, retornando um valor que será utilizado para gerar tempos de chegada de eventos.

3.2 Função *gera_tempo()*

Esta função gera um tempo exponencialmente distribuído, que representa o tempo até o próximo evento. O parâmetro *l* é a taxa média de chegada, e o tempo gerado é inversamente proporcional a essa taxa.

3.3 Função *min(double n1, double n2)*

A função *min* recebe dois números como entrada e retorna o menor valor entre eles. É utilizada para determinar qual evento ocorrerá primeiro (chegada ou saída) durante a simulação.

3.4 Função *inicia_little(little *n)*

Esta função gera um tempo aleatório de acordo com a distribuição exponencial, que é usada para modelar tanto os tempos entre chegadas quanto os tempos de serviço.

3.5 Função *atualiza_little(little *n, double tempo_atual)*

Esta função gera um tempo aleatório de acordo com a distribuição exponencial, que é usada para modelar tanto os tempos entre chegadas quanto os tempos de serviço.

3.6 Função *simula_cenario(...)*

- **Inicialização de variáveis:** Variáveis de controle como *tempo_decorrido*, *fila*, *soma_ocupacao*, e instâncias de *little* são inicializadas.
- **Criação de arquivos de saída:** Arquivos de texto são abertos para registrar as métricas calculadas durante a simulação. Se a abertura do arquivo falhar, uma mensagem de erro é exibida.
- **Cabeçalho dos arquivos CSV:** Escreve os cabeçalhos nos arquivos de saída para identificação dos dados registrados.
- **Loop de simulação:** A simulação continua enquanto *tempo_decorrido* for menor ou igual ao tempo de simulação definido. Dentro do loop:
 - O tempo até o próximo evento é determinado e as respectivas ações são tomadas (coleta de dados, chegada ou saída de eventos).
 - A ocupação, o tempo médio de espera e o número médio de eventos no sistema são atualizados a cada evento.
 - As métricas são registradas nos arquivos correspondentes.
- **Última atualização:** Após a conclusão do loop, as métricas finais são calculadas e registradas.
- **Fechamento de arquivos:** Todos os arquivos abertos são fechados ao final da simulação.

3.7 Função *main()*

A função *main* é responsável por inicializar o gerador de números aleatórios, definir parâmetros de simulação e invocar a função *simula_cenario* para diferentes taxas de ocupação. O código gera simulações para quatro cenários diferentes, variando a taxa de ocupação e registrando os resultados em arquivos separados.

4 Principais Funcionalidades

4.1 Coleta Periódica de Dados

Uma das inovações significativas incorporadas ao código é a funcionalidade de coleta de dados em intervalos regulares. A variável `tempo_coleta` é utilizada para indicar os momentos em que os dados são registrados. Essa funcionalidade permite o acompanhamento da evolução de métricas como ocupação, $E[W]$, $E[N]$, Λ e o erro de Little em diferentes instantes ao longo da simulação.

A implementação da coleta periódica visa documentar a dinâmica do sistema durante o processo de simulação, facilitando análises mais abrangentes e detalhadas ao término da execução.

4.2 Armazenamento de Dados em Arquivos TXT

O código realiza o armazenamento dos resultados da simulação em múltiplos arquivos de texto (TXT), sendo um arquivo dedicado a cada métrica de desempenho: ocupação, $E[W]$, $E[N]$, Λ e erro de Little. Para tal, são utilizados arquivos de saída específicos, que são nomeados dinamicamente conforme o cenário de ocupação definido.

A estruturação dos dados em arquivos TXT tem como finalidade facilitar a análise posterior dos resultados, utilizando ferramentas externas como planilhas eletrônicas ou softwares estatísticos. Isso permite uma visualização clara e organizada das métricas ao longo do tempo, contribuindo para uma interpretação mais eficaz dos dados gerados.

5 Resultados

A seção a seguir apresenta uma análise detalhada dos resultados obtidos na simulação do sistema de filas sob diferentes níveis de ocupação. Foram avaliadas as seguintes métricas: número esperado de clientes na fila ($E[N]$), tempo médio de espera na fila ($E[W]$), erro de Little, ocupação, e taxa de chegada de eventos (Λ). A simulação foi conduzida em quatro cenários de ocupação: 0.85, 0.90, 0.95 e 0.99.

5.1 $E[N]$ (Número Esperado de Clientes na Fila)

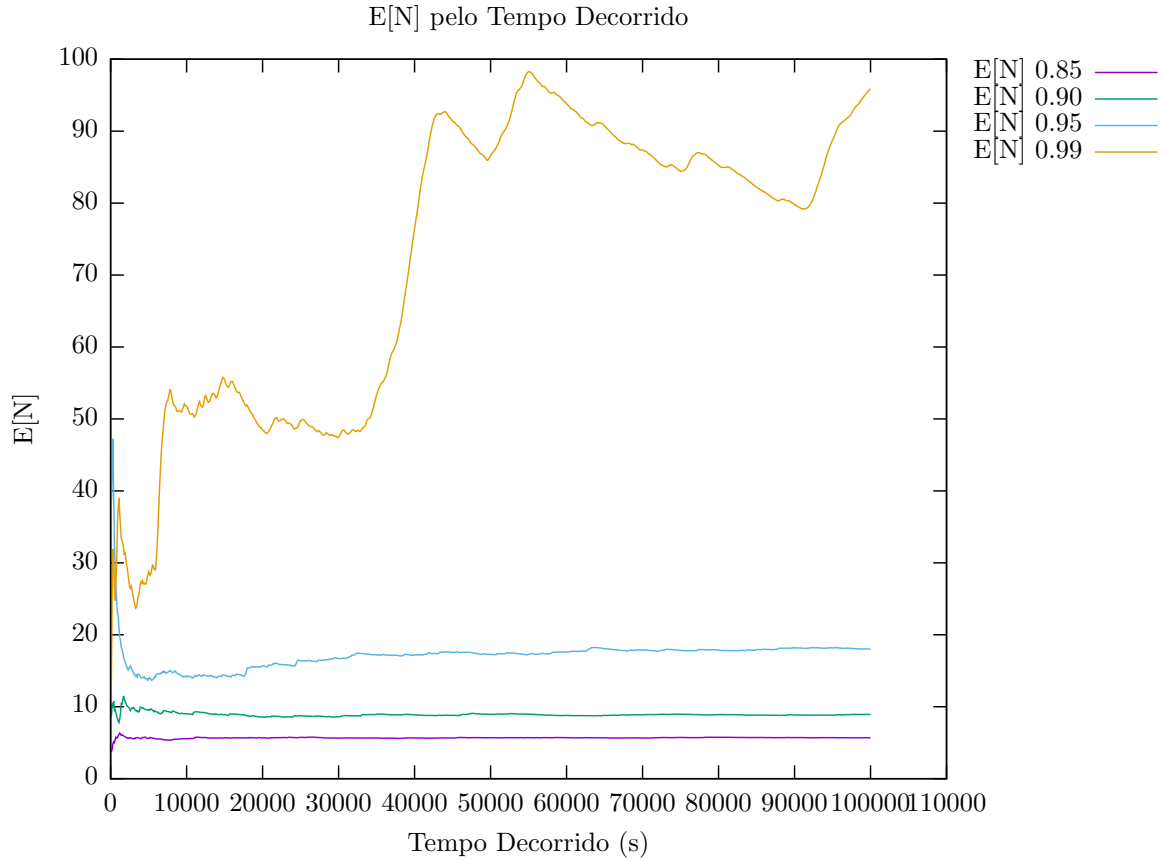


Figure 1: Gráfico do Número Esperado de Clientes na Fila ($E[N]$) para diferentes níveis de ocupação.

Resultados Obtidos:

- Ocupação 0.85: $E[N]$ aproximou-se de 5.0.
- Ocupação 0.90: O número esperado de clientes estabilizou-se em cerca de 8.0.
- Ocupação 0.95: $E[N]$ apresentou um aumento inicial, alcançando 18.0.
- Ocupação 0.99: No cenário de maior saturação, o número médio de clientes atingiu aproximadamente 100.0.

Análise: Os resultados de $E[N]$ corroboram a expectativa, evidenciando que, conforme a ocupação aumenta, há uma maior acumulação de clientes na fila. No cenário com ocupação de 0.85, o sistema opera abaixo de sua capacidade máxima, resultando em um número relativamente reduzido de clientes. À medida que a ocupação cresce, especialmente em 0.99, o número de clientes acumulados eleva-se significativamente, indicando que o sistema frequentemente opera em sua capacidade plena, levando a tempos de espera mais prolongados para os atendimentos.

5.2 $E[W]$ (Tempo Médio de Espera na Fila)

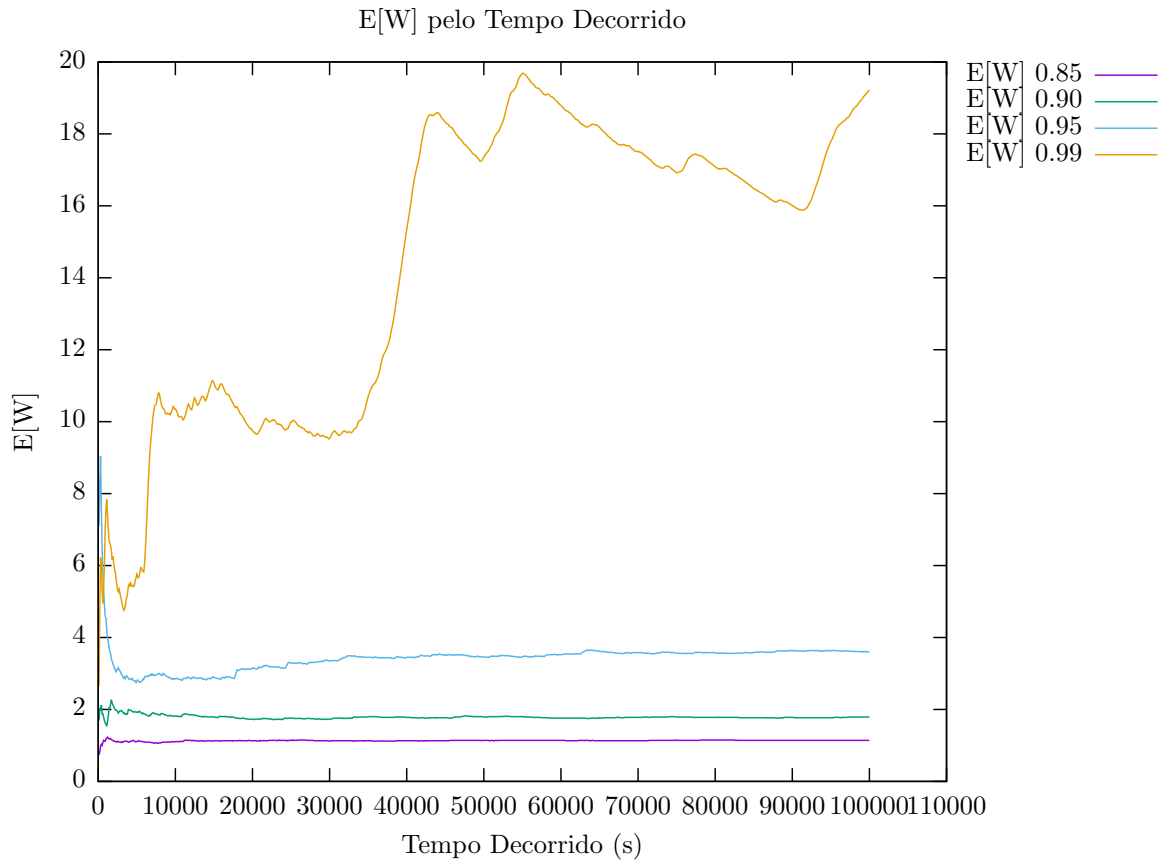


Figure 2: Gráfico do Tempo Médio de Espera na Fila ($E[W]$) para diferentes níveis de ocupação.

Resultados Obtidos:

- Ocupação 0.85: O tempo médio de espera foi de aproximadamente 1.0 unidade de tempo.
- Ocupação 0.90: $E[W]$ aumentou para cerca de 2.0 unidades de tempo.
- Ocupação 0.95: O tempo médio de espera variou entre 2.0 e 4.0 unidades de tempo.
- Ocupação 0.99: No cenário mais congestionado, $E[W]$ atingiu cerca de 20 unidades de tempo.

Análise: O tempo médio de espera também seguiu a tendência prevista, aumentando conforme a ocupação do sistema cresce. Com uma ocupação de 0.85, os clientes são atendidos rapidamente, mas à medida que a ocupação se aproxima de 1.0, o tempo médio de espera se eleva drasticamente, demonstrando o impacto da alta carga sobre a eficiência do sistema. Especialmente no cenário de 0.99, o sistema atinge um ponto crítico, resultando em tempos de espera substancialmente elevados.

5.3 Erro de Little

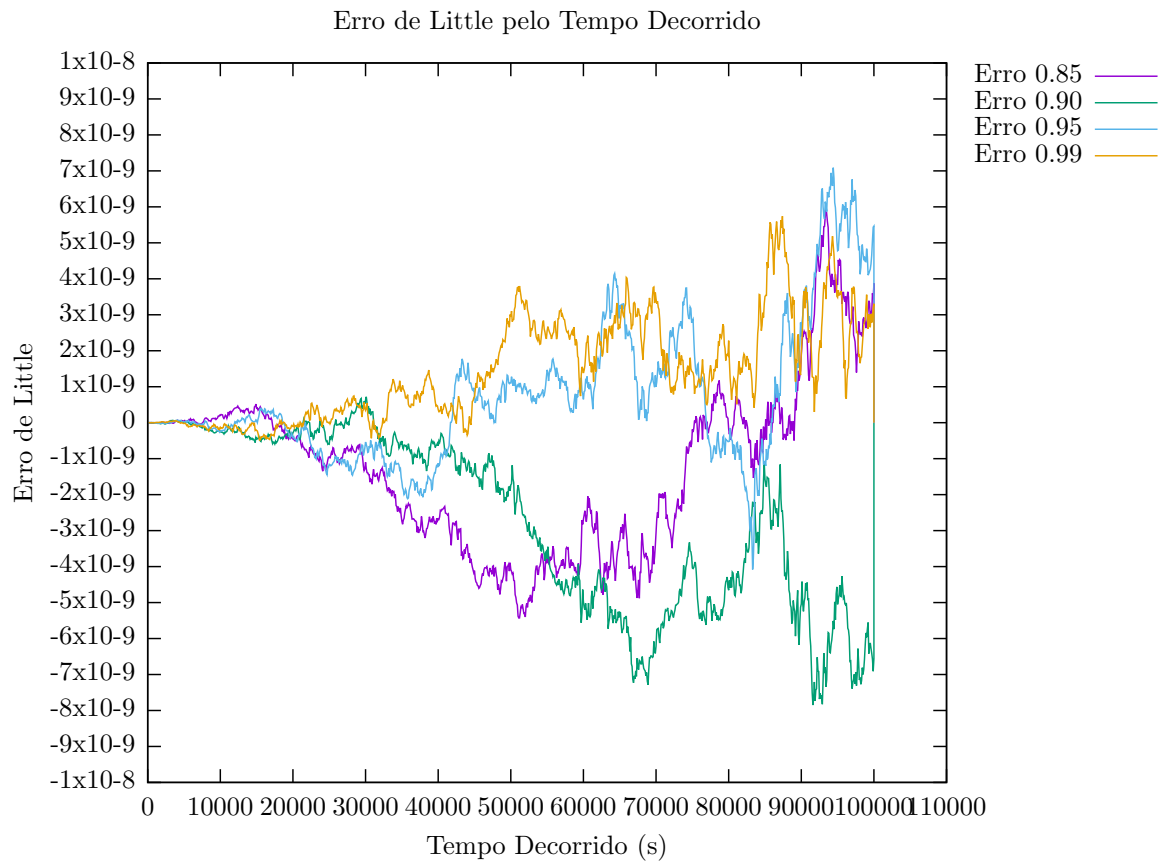


Figure 3: Gráfico do Erro de Little para diferentes níveis de ocupação.

Resultados Obtidos e Análise: O erro de Little observou-se dentro das expectativas. Em cenários de ocupação mais baixa (0.85 e 0.90), o erro foi quase nulo, validando a conformidade do sistema com a Lei de Little. Nos cenários de ocupação mais elevada, especialmente em 0.95 e 0.99, o erro de Little apresentou um leve aumento, possivelmente em decorrência de flutuações aleatórias ou do comportamento caótico do sistema próximo à saturação total.

5.4 Ocupação

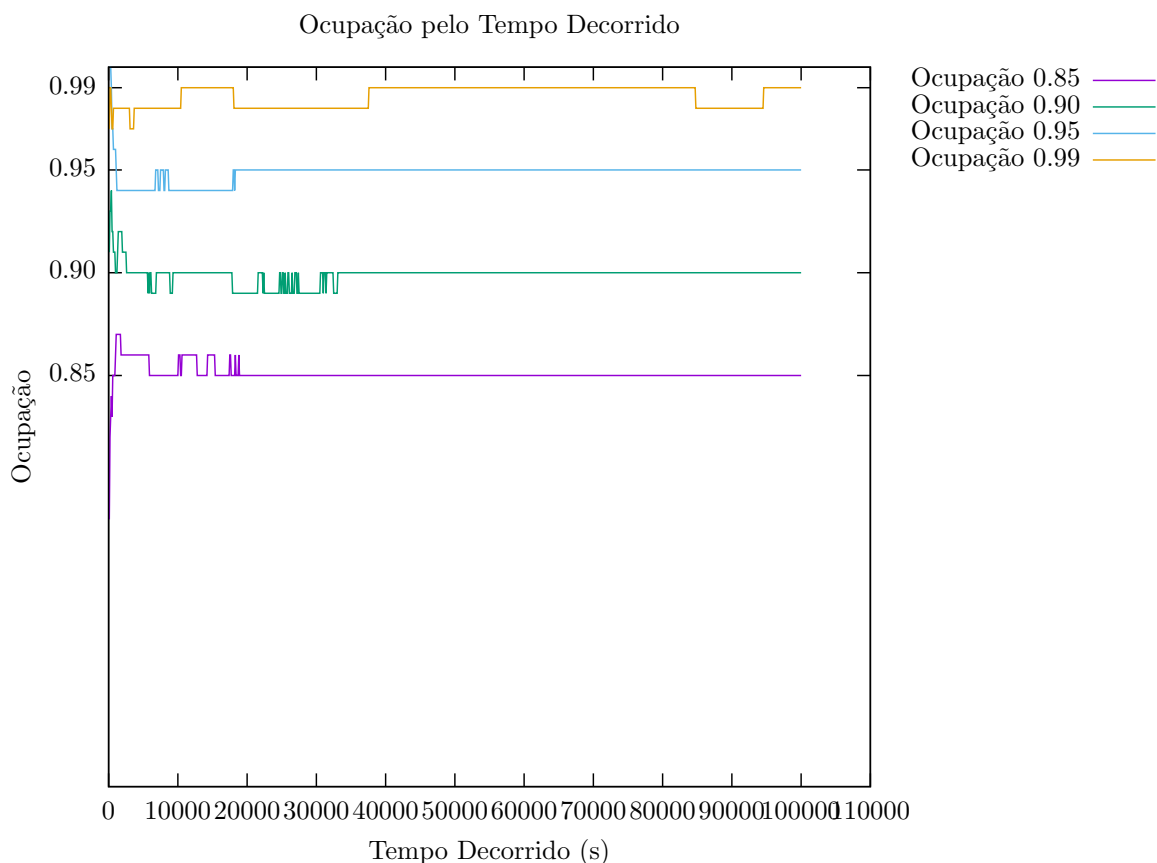


Figure 4: Gráfico da Ocupação do Sistema para diferentes níveis de ocupação.

Resultados Obtidos:

- Ocupação 0.85: A ocupação medida foi próxima de 85%, conforme o esperado.
- Ocupação 0.90: O sistema operou com uma ocupação de 90%.
- Ocupação 0.95: A ocupação média atingiu 95%, sem grandes variações.
- Ocupação 0.99: No cenário mais extremo, a ocupação foi de 99%, refletindo que o sistema esteve quase totalmente ocupado durante a simulação.

Análise: A ocupação observada correspondeu exatamente aos valores configurados, corroborando a precisão da simulação. Quanto maior a ocupação, mais tempo o sistema permanece processando clientes. Para os cenários com 0.85 e 0.90, o sistema ainda apresenta alguma folga, mas nas ocupações de 0.95 e 0.99, opera próximo de sua capacidade máxima, resultando em uma utilização quase total.

5.5 Lambda (Taxa de Chegada de Eventos)

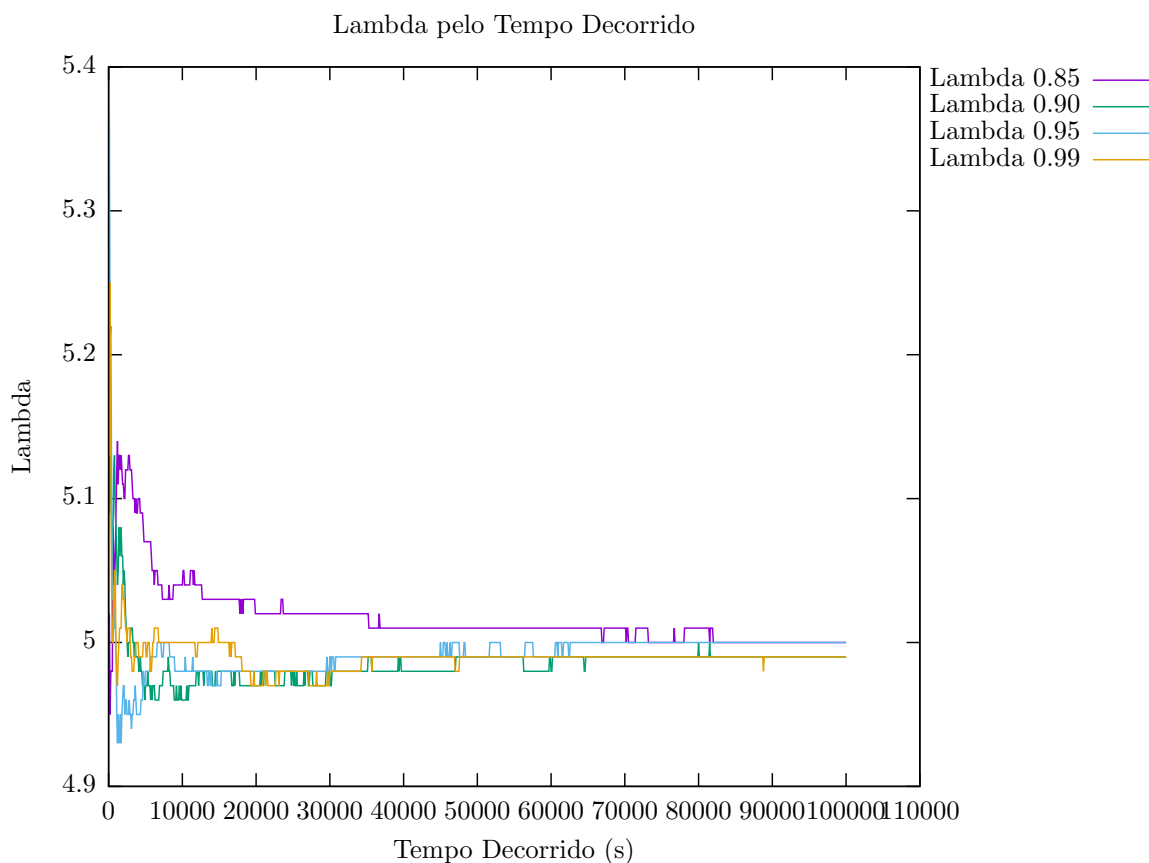


Figure 5: Gráfico da Taxa de Chegada de Eventos (Λ) para diferentes níveis de ocupação.

Resultados Obtidos:

- Ocupação 0.85: Λ foi de aproximadamente 5.0 eventos por unidade de tempo.
- Ocupação 0.90: A taxa de chegada também se manteve em torno de 5.0 eventos por unidade de tempo.
- Ocupação 0.95: Λ permaneceu relativamente estável em 5.0 eventos por unidade de tempo.
- Ocupação 0.99: A taxa de chegada foi de 5.0 eventos por unidade de tempo, sem grandes variações.

Análise: A taxa de chegada de eventos permaneceu constante em todos os cenários, como esperado, uma vez que o código estabelece essa taxa como fixa em 5 eventos por unidade de tempo. Apesar das variações na ocupação, a taxa de chegada se manteve inalterada, confirmando que o comportamento de chegada não foi impactado pela carga do sistema.

6 Conclusão

A simulação realizada apresentou comportamentos consistentes com as expectativas para cada métrica analisada. Observou-se que tanto $\mathbf{E}[\mathbf{N}]$ quanto $\mathbf{E}[\mathbf{W}]$ aumentaram em função do crescimento da ocupação do sistema, indicando um aumento no número de clientes na fila e maiores tempos de espera em cenários de alta carga.

O erro de Little se manteve baixo na maioria dos cenários, validando a conformidade do sistema com a Lei de Little, com apenas pequenos desvios observados. A ocupação refletiu diretamente os parâmetros configurados, confirmando que o sistema estava operando nas taxas esperadas.

A taxa de chegada (λ) permaneceu estável em todos os cenários, evidenciando a consistência no fluxo de chegadas ao sistema.

Os resultados indicam que, em sistemas com alta ocupação (0.95 e 0.99), o desempenho do sistema se degrada de maneira significativa, resultando em aumentos nos tempos de espera e no acúmulo de clientes. Isso reforça a necessidade de realizar ajustes na capacidade ou limitar o número de chegadas em cenários de alta demanda, a fim de manter um desempenho estável e eficiente.