

Análise de Desempenho

Trabalho 1 - Relatório

Discente: José Flávio Lopes

Discente: Eduardo Guerreiro Rocha

1 - Introdução	3
2 - Validação Matemática	3
3 - Modificações no código do simulador	4
4 - Resultados finais e Gráficos	5
4.1 - Resultados finais	5
4.2 - Gráficos	6
4.2.1 - Ocupação de 80%	6
4.2.2 - Ocupação de 90%	8
4.2.3 - Ocupação de 95%	10
4.2.3 - Ocupação de 99%	12
4.3 - Considerações sobre os resultados	13
5 - Conclusão	14

1 - Introdução

Neste trabalho, nos foi proposto a modificação do código de um simulador de fila, de modo que dados do estado do simulador fossem coletados a cada 100 segundos. De três parâmetros utilizados pelo simulador, dois foram fixados, o que permitiu uma validação matemática sobre os resultados obtidos. A seguir, a forma como este relatório foi organizado: na seção 2, uma explicação sobre a validação matemática utilizada; na seção 3, uma explicação sobre as modificações feitas no código; na seção 4, a apresentação dos resultados finais e dos gráficos; por fim, na seção 5, uma conclusão.

2 - Validação Matemática

O simulador faz uso de três parâmetros, sendo eles: o tempo total de simulação (em segundos), o intervalo médio da chegada (em segundos) e o tempo médio de serviço (em segundos). Os dois primeiros parâmetros foram fixados, sendo que o tempo total de simulação é de 36000 segundos, e o intervalo médio de chegada de 5 por segundo, ou 0.2 segundos. Como estamos simulando um sistema de filas, uma medida interessante de se obter é quanto tempo o sistema fica ocupado (atendendo alguém). Ao saber que a cada 0.2 segundos ocorre uma chegada, podemos, de maneira intuitiva, perceber que se o tempo médio de serviço for, também, 0.2 segundos, o sistema ficará ocupado o tempo todo, pois ao terminar de atender alguém, outro terá chegado. Conseguimos expressar matematicamente essa medida ao dividir o tempo médio de serviço pelo tempo médio de chegada. No caso citado acima, teríamos $0.2 / 0.2 = 1$, que podemos interpretar como: 100% do tempo o sistema está ocupado. Portanto, temos que:

$$ocupação = \frac{\text{tempo médio de serviço}}{\text{tempo médio de chegada}}$$

Como consequência da fórmula acima, podemos encontrar qual o tempo médio de serviço dado um percentual de ocupação, pois já sabemos que o tempo médio de chegada é fixo em 0.2 segundos. Portanto, os resultados apresentados podem ser validados matematicamente a partir da fórmula apresentada.

Não nos ocorreu outra validação matemática.

3 - Modificações no código do simulador

A versão do código do simulador na qual iniciamos o trabalho tratava apenas de dois eventos, sendo a chegada e a saída. Como nos foi proposto coletar dados nos tempos exatos, a cada cem segundos, foi preciso implementar e tratar mais um evento, sendo efetivamente a coleta das seguintes medidas: ocupação, $E[N]$, $E[W]$ e o erro de Little. Para cada uma das medidas, um arquivo .txt foi criado, com o intuito de ser utilizado para a geração dos gráficos. A grosso modo, podemos dizer que apenas um bloco de código foi adicionado, sendo ele o que trata o novo evento do simulador. Abaixo, segue a imagem do trecho de código que foi incorporado:

```
tempo_decorrido = !fila ? minimo(chegada, tempo_coleta) : minimo(tempo_coleta, minimo(chegada, servico));

// coleta de dados
if (tempo_decorrido == tempo_coleta) {

    // ocupacao
    fprintf(arq_ocupacao, "%d %lf\n", tempo_coleta, soma_tempo_servico / maximo(tempo_coleta, servico));

    // en
    e_n_final = (e_n.soma_areas + (tempo_decorrido - e_n.tempo_anterior) * e_n.no_eventos) / tempo_decorrido;
    fprintf(arq_en, "%d %lf\n", tempo_coleta, e_n_final);

    // ew
    e_w_final = ( (e_w_chegada.soma_areas + (tempo_decorrido - e_w_chegada.tempo_anterior) * e_w_chegada.no_eventos) -
                  (e_w_saida.soma_areas + (tempo_decorrido - e_w_saida.tempo_anterior) * e_w_saida.no_eventos) ) /
                  e_w_chegada.no_eventos;
    fprintf(arq_ew, "%d %lf\n", tempo_coleta, e_w_final);

    // erro de little
    lambda = e_w_chegada.no_eventos / tempo_decorrido;
    fprintf(arq_erro_little, "%d %.20lf\n", tempo_coleta, fabs(e_n_final - lambda * e_w_final));

    tempo_coleta += 100;
```

4 - Resultados finais e Gráficos

Todos os resultados foram obtidos utilizando uma semente aleatória igual a 10000.

4.1 - Resultados finais

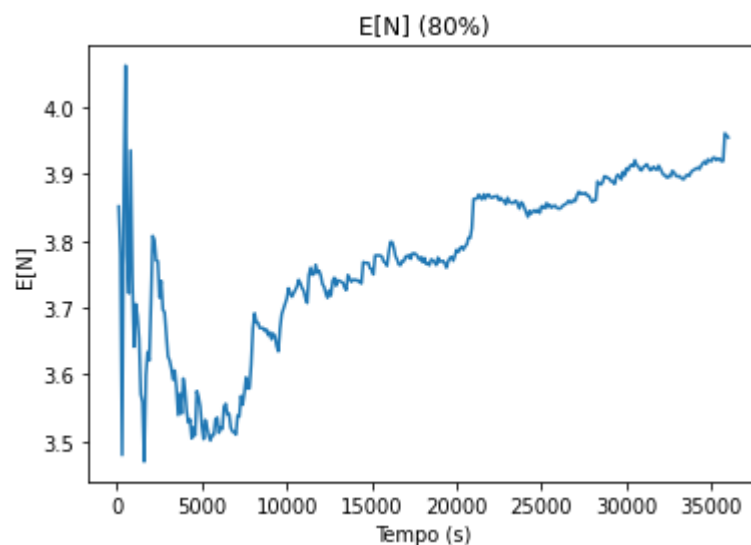
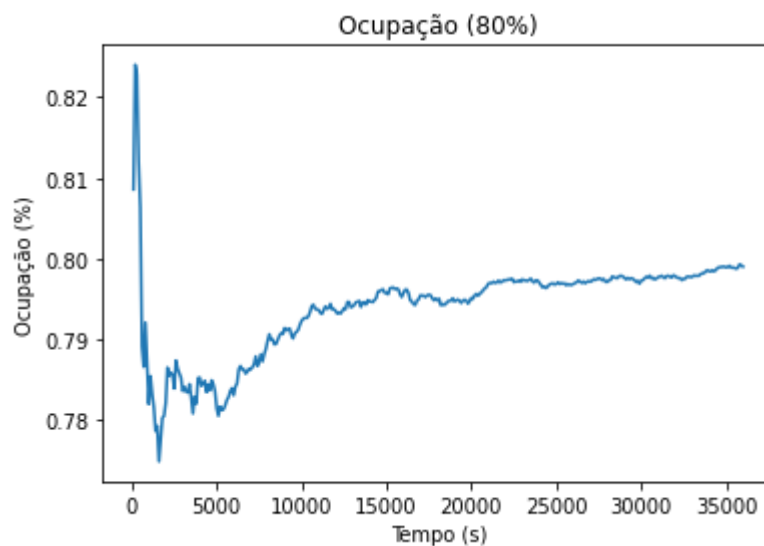
Tempo médio de serviço	Ocupação	E[W]	E[N]	Erro de Little
0.16	0.799039	0.792743	3.954661	0.0000000011 2027764843
0.18	0.896370	1.696817	8.451551	-0.000000000 11496226193
0.19	0.947630	3.486691	17.379988	0.0000000008 3020168518
0.198	0.992829	26.140279	130.670985	0.0000000004 8490278459

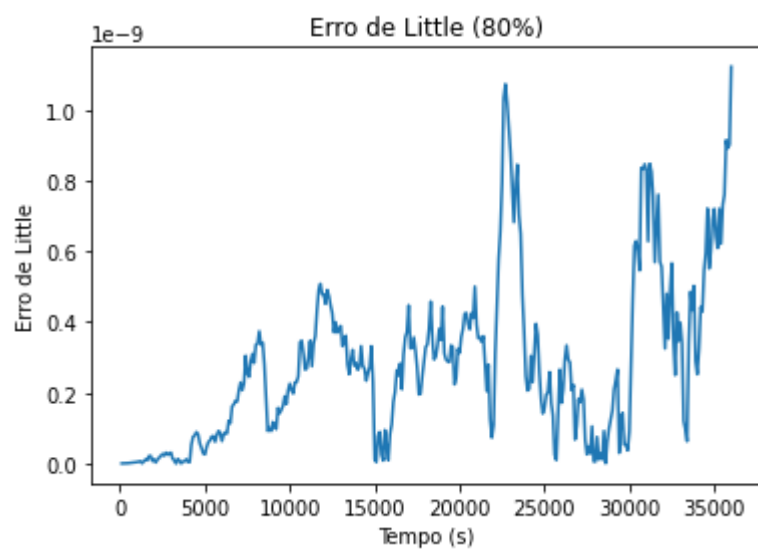
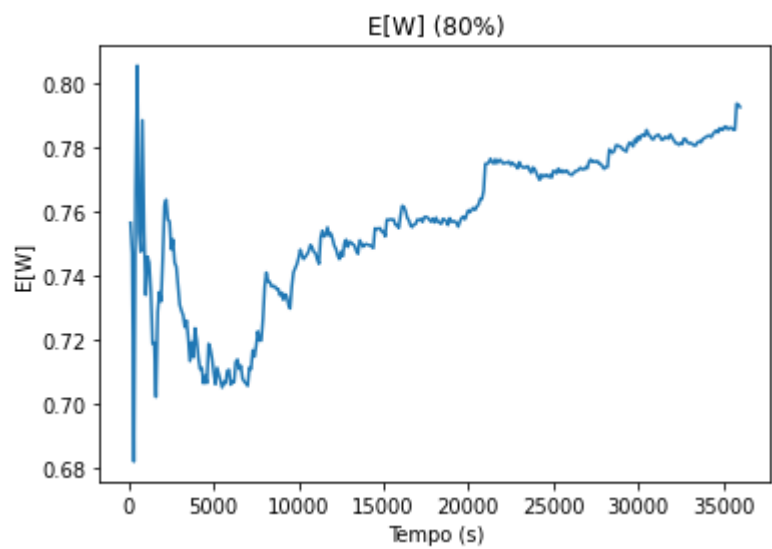
4.2 - Gráficos

4.2.1 - Ocupação de 80%

Utilizando da fórmula apresentada na seção 2, sabemos que o tempo médio de serviço para gerar uma ocupação de 80% é de 0.16 segundos.

Os gráficos abaixo mostram os dados gerados pelo simulador ao receber os seguintes parâmetros: tempo total de execução = 36000, tempo médio de chegada = 0.2, tempo médio de serviço = 0.16.

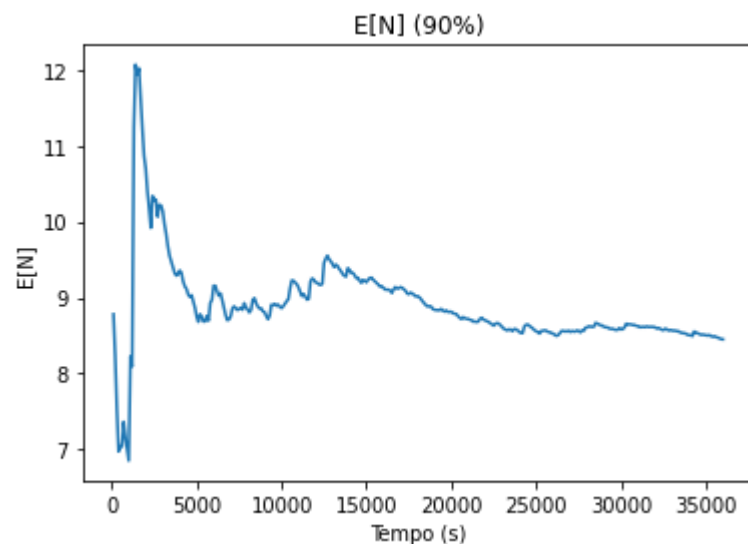
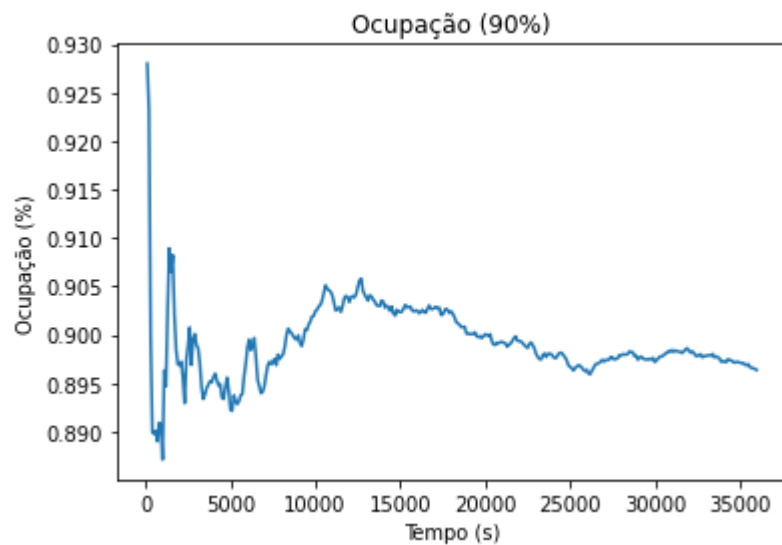


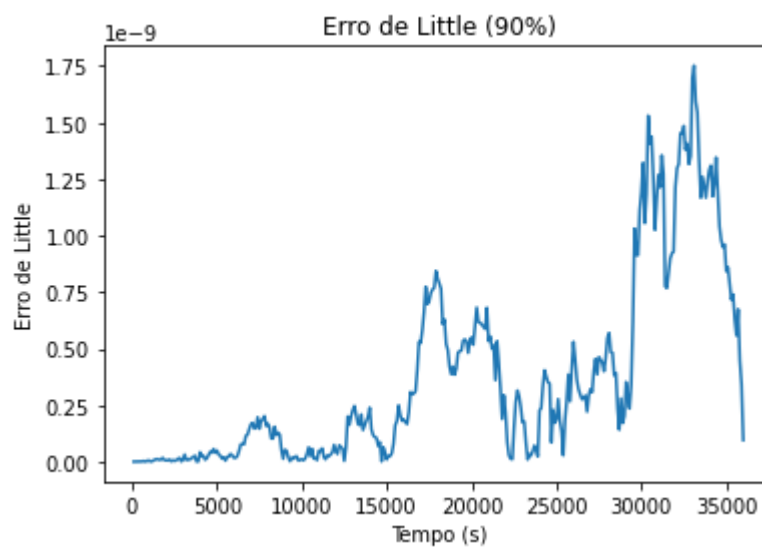
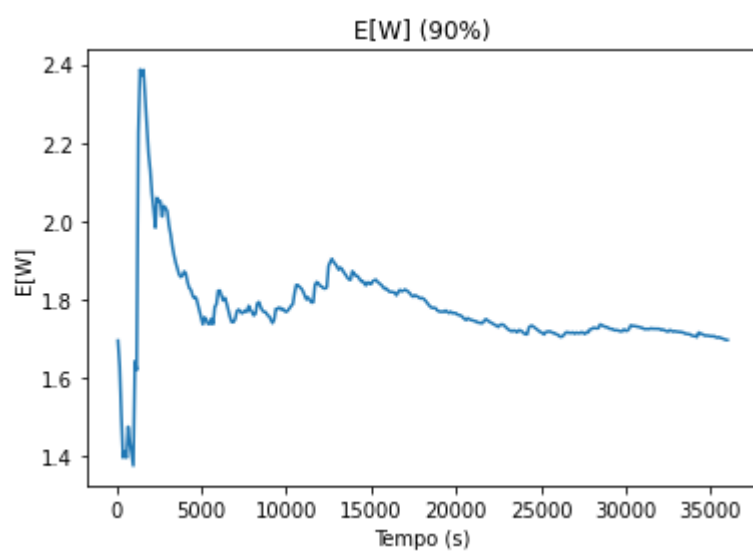


4.2.2 - Ocupação de 90%

Para haver uma ocupação de 90% do tempo, é preciso que o tempo médio de serviço seja de 0.18 segundos.

Os próximos gráficos foram gerados a partir dos dados computados pelo simulador ao receber os seguintes parâmetros: tempo total de execução = 36000, tempo médio de chegada = 0.2, tempo médio de serviço = 0.18.

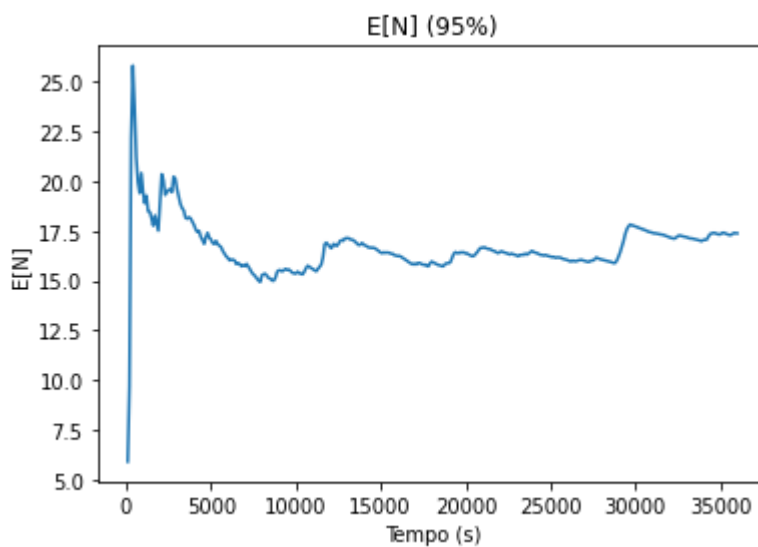
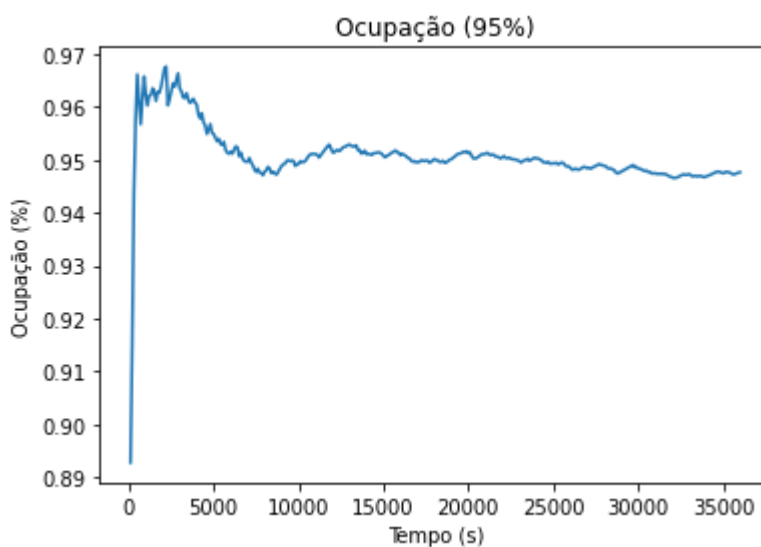


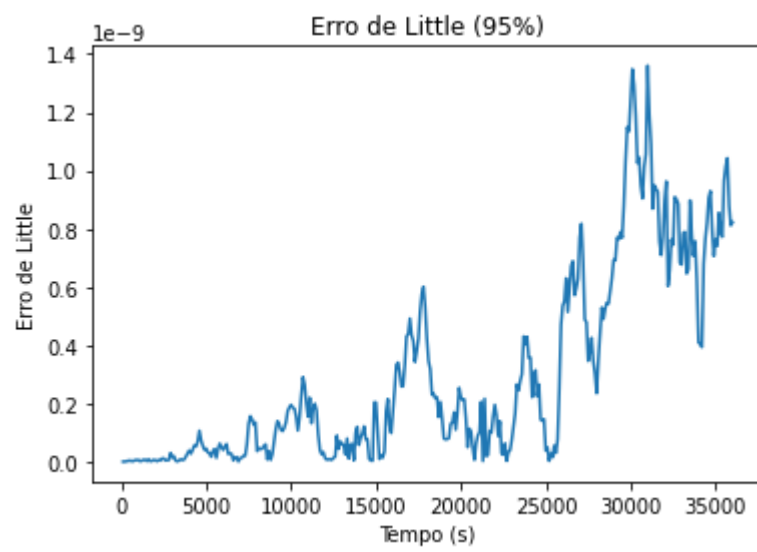
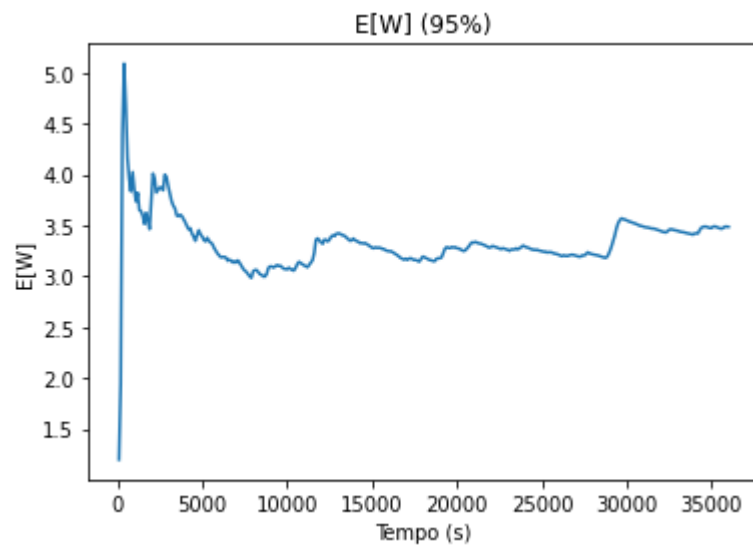


4.2.3 - Ocupação de 95%

Para uma ocupação de 95%, é necessário um tempo médio de serviço de 0.19.

Os gráficos a seguir foram coletados a partir da execução do simulador com os seguintes parâmetros: tempo total de execução = 36000, tempo médio de chegada = 0.2, tempo médio de serviço = 0.19.

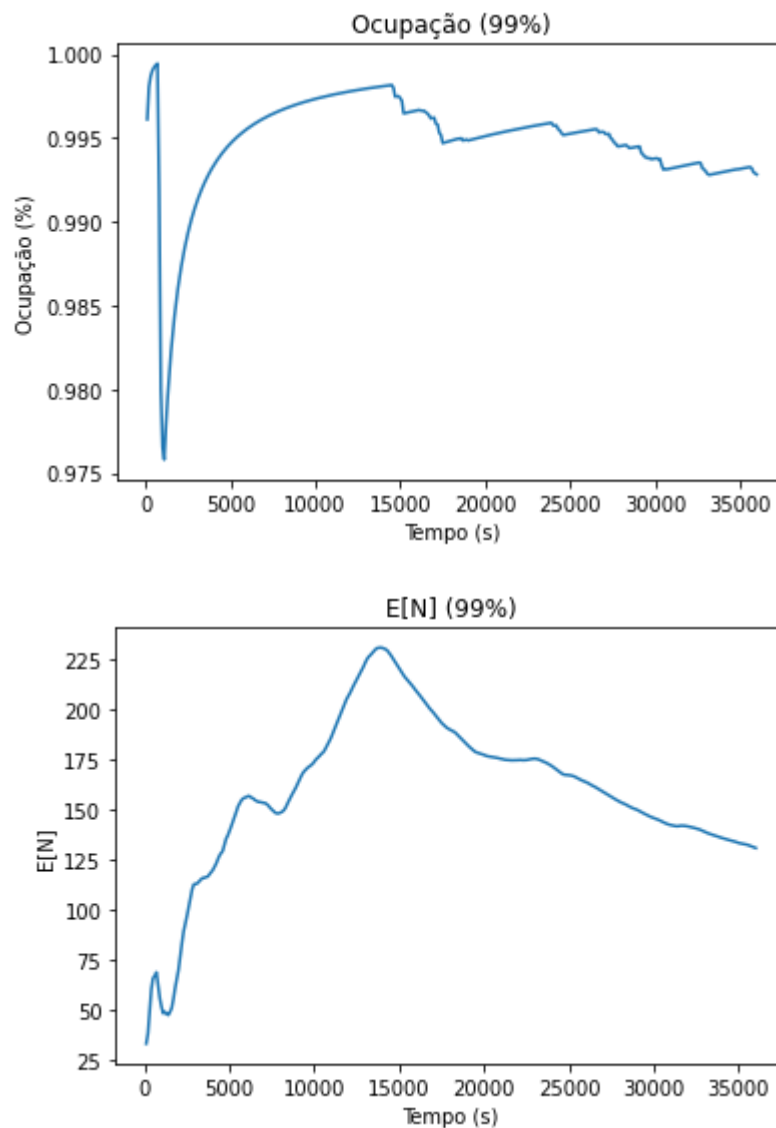


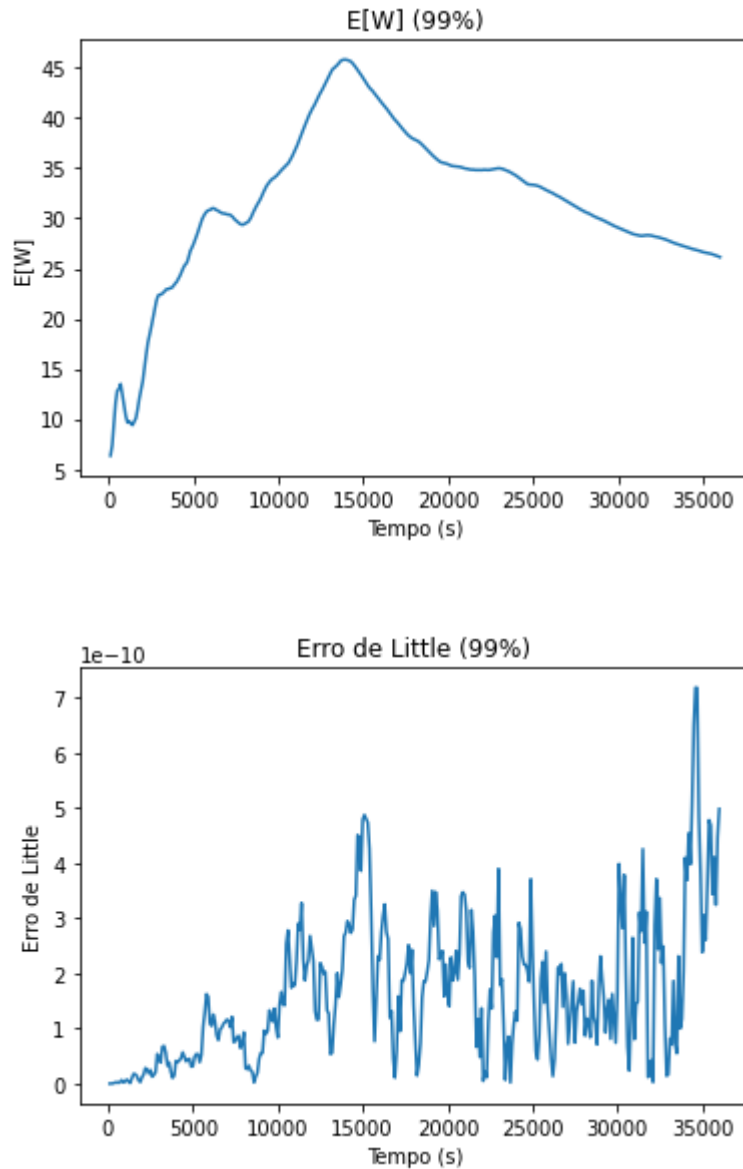


4.2.3 - Ocupação de 99%

Para uma ocupação de 99%, uma taxa média de serviço igual a 0.198 segundos é necessária.

Os gráficos a seguir foram gerados a partir de dados obtidos ao executar o simulador com os seguintes parâmetros: tempo total de execução = 36000, tempo médio de chegada = 0.2, tempo médio de serviço = 0.198.





4.3 - Considerações sobre os resultados

Enxergamos coerência nos dados da ocupação ao perceber que estão de acordo com a validação matemática descrita na seção 2.

Sobre os dados de $E[N]$ e de $E[W]$, notamos que eles crescem à medida que a ocupação aumenta, o que faz sentido, uma vez que o $E[W]$ é referente ao tempo médio que um elemento passa no sistema (ao aumentar o tempo médio de serviço, mais tempo um elemento passará na fila e sendo atendido, consequentemente ficará mais tempo no sistema; a taxa de ocupação também aumenta, logo, é coerente que $E[W]$ aumente quando a taxa de ocupação também aumentar);

também faz sentido o aumento em $E[N]$ quando aumenta a ocupação, pois, mais uma vez, a taxa de ocupação irá aumentar quando o tempo médio de serviço aumentar, e o aumento deste último ocasiona um maior tempo de espera na fila e um maior tempo sendo atendido, consequentemente aumentando o número de elementos no sistema.

5 - Conclusão

Em conclusão, acreditamos que a realização deste trabalho foi uma experiência benéfica para a consolidação do aprendizado, principalmente por ser uma continuação direta das aulas que foram ministradas, servindo para ampliar o entendimento do assunto que então vinha sendo abordado. Acerca das tarefas implementadas, ressaltamos a dificuldade inicial em relação a modificação do código para a coleta de dados. Embora agora o entendimento da solução esteja claro, muito tempo foi despendido tentando elaborar uma maneira de resolver o problema, o que também contribuiu para o ganho de conhecimento.