

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**NICOLAS CHARLES OLIVEIRA MEDEIRO**

**PREDIÇÃO DA DISPONIBILIDADE DE RECURSOS  
COMPUTACIONAIS A PARTIR DO USO DE MÉDIAS MÓVEIS EM  
SÉRIES TEMPORAIS**

**TRABALHO DE CONCLUSÃO DE CURSO**

**PONTA GROSSA**

**2021**

**NICOLAS CHARLES OLIVEIRA MEDEIRO**

**PREDIÇÃO DA DISPONIBILIDADE DE RECURSOS  
COMPUTACIONAIS A PARTIR DO USO DE MÉDIAS MÓVEIS EM  
SÉRIES TEMPORAIS**

Trabalho de Conclusão de Curso apresentada como requisito parcial à obtenção do título de Bacharel, em Ciência da Computação, do Departamento acadêmico de informática, da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Lourival Aparecido de Góis

**PONTA GROSSA**

**2021**



Ministério da Educação  
**Universidade Tecnológica Federal do Paraná**  
Câmpus Ponta Grossa  
Diretoria de Graduação e Educação Profissional  
Departamento Acadêmico de Informática  
Bacharelado em Ciência da Computação



---

## TERMO DE APROVAÇÃO

### PREDIÇÃO DA DISPONIBILIDADE DE RECURSOS COMPUTACIONAIS A PARTIR DO USO DE MÉDIAS MÓVEIS EM SÉRIES TEMPORAIS

Por

Nicolas Charles Oliveira Medeiro

Este Trabalho de Conclusão de Curso (TCC) foi apresentado em 19 de maio de 2021 como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

---

Prof. Dr. Lourival Aparecido de Góis  
Orientador

---

Prof. Dr. Augusto Foronda  
Membro titular

---

Prof. Dr. Erikson Freitas de Moraes  
Membro titular

---

Prof. MSc. Geraldo Ranthum  
Responsável pelo Trabalho de Conclusão  
de Curso

---

Prof<sup>a</sup> Dr<sup>a</sup> Mauren Louise Sguario  
Coordenadora

- A Folha de Aprovação assinada encontra-se arquivada na Secretaria Acadêmica -

Dedico este trabalho à minha família,  
pelos momentos de ausência.

## **AGRADECIMENTOS**

Certamente estes parágrafos não irão atender a todas as pessoas que fizeram parte dessa importante fase de minha vida. Portanto, desde já peço desculpas àquelas que não estão presentes entre essas palavras, mas elas podem estar certas que fazem parte do meu pensamento e de minha gratidão.

Agradeço ao meu orientador Prof. Dr. Lourival Aparecido de Góis, pela sabedoria com que me guiou nesta trajetória.

Aos meus colegas de sala.

Gostaria de deixar registrado também, o meu reconhecimento à minha família, pois acredito que sem o apoio deles seria muito difícil vencer esse desafio.

Enfim, a todos os que por algum motivo contribuíram para a realização desta pesquisa.

Os problemas são oportunidades para se mostrar o que sabe. (ELLINGTON, Duke)

## RESUMO

MEDEIRO, Nicolas Charles Oliveira. **Predição da disponibilidade de recursos computacionais a partir do uso de médias móveis em séries temporais**. 2021. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2021.

Este trabalho tem como objetivo geral a análise de séries temporais geradas através de coletas do percentual de uso de CPU em microcomputadores visando a identificação do comportamento de um equipamento. Para tanto, foi necessário, compreender as características destas séries temporais, e a partir delas, propiciar o estudo e a aplicação de métodos envolvendo médias móveis nos dados coletados a fim de se obter uma assinatura comportamental do equipamento monitorado. Esta assinatura comportamental foi obtida ao longo de análises sucessivas, comparada com o comportamento real do equipamento e com isto, identificar a margem de erro envolvida no processo. A aplicação destes métodos matemáticos possibilitou a modelagem de um sistema de gerenciamento de recursos computacionais (SGR) baseado na previsibilidade de seus índices de ocupação, permitindo e adequando o uso dos mesmos com a maior taxa de sucesso possível.

**Palavras-chave:** Predição. Série temporal. Recursos computacionais

## ABSTRACT

MEDEIRO, Nicolas Charles Oliveira. **Prediction of the availability of computational resources from the use of moving averages in time series**. 2021 Work of Conclusion Course (Graduation in Computer Science) - Federal Technology University - Paraná. Ponta Grossa, 2021.

This work has as general objective the analysis of time series generated through collections of the percentage of CPU usage in microcomputers aiming at the identification of the equipment behavior. For that, it was necessary to understand the characteristics of these time series, and from them, provide the study and the application of methods involving moving averages in the collected data in order to obtain a behavioral signature of the monitored equipment. This behavioral signature must, throughout successive analyzes, be compared with the actual behavior of the equipment and with this, identify the margin of error involved in the process. The application of these mathematical methods will enable the modeling of a computational resource management system (CRM) based on the predictability of their occupancy rates, allowing and adapting their use with the highest possible success rate.

**Keywords:** Prediction. Time series. Computational resources.



## LISTA DE ILUSTRAÇÕES

Figura 1 - Ilustração de um sistema distribuído .....	14
Figura 2 - Distribuição Normal.....	24
Figura 3 - Curva Normal Padronizada.....	25
Figura 4 - Tela principal <i>GNOME System Monitor</i> .....	27
Figura 5 - Tela de configuração <i>GNOME System Monitor</i> .....	27
Figura 6 - Tela inicial do <i>PERFMON</i> .....	28
Figura 7 - Configurando coletor de dados .....	28
Figura 8 - Contadores do <i>PERFMON</i> .....	29
Figura 9 - Tela inicial do Calc.....	31
Figura 10 - Tela inicial do <i>Excel</i> .....	32
Figura 11 - Tratamento dos dados .....	32
Figura 12 - Dados processados .....	33
Figura 13 - Tela inicial do R Studio .....	34
Figura 14 - Abertura Dataset.....	34
Figura 15 - Processamento do Dataset.....	35
Figura 16 - Exemplo de gráfico MMS vs Real .....	36
Figura 17 - Inserindo diretório .....	41
Figura 18 - <i>Struct</i> coletas .....	42
Figura 19 - Percorrendo arquivo de coletas .....	42
Figura 20 - Enviando dados para função MMS .....	42
Figura 21 - Função MMS.....	43
Figura 22 - Lendo arquivo de resultados.....	44
Figura 23 - Plotando comportamento real .....	44
Figura 24 - Dia 1: Comportamento real .....	45
Figura 25 - Plotando predição .....	45
Figura 26 - Dia 1: Predição.....	46
Figura 27 - Verificação de sazonalidade e tendência.....	46
Figura 28 - Dia 1: Análises .....	47
Figura 29 - Dia 1: Comparativo de séries.....	48
Figura 30 - Plotando assinatura comportamental.....	48
Figura 31 - Dia 1: Assinatura comportamental .....	49
Figura 32 - Dia 2: Comportamento real .....	50
Figura 33 - Dia 2: Predição.....	51
Figura 34 - Dia 2: Análises .....	51
Figura 35 - Dia 2: Comparativo de séries.....	52
Figura 36 - Dia 2: Assinatura comportamental .....	53
Figura 37 - Dia 3: Comportamento real .....	53

Figura 38 - Dia 3: Predição.....	54
Figura 39 - Dia 3: Análises .....	54
Figura 40 - Dia 3: Comparativo de séries .....	55
Figura 41 - Dia 3: Assinatura comportamental .....	56
Figura 42 - Dia 4: Comportamento real .....	56
Figura 43 - Dia 4: Predição.....	57
Figura 44 - Dia 4: Análises .....	57
Figura 45 - Dia 4: Assinatura comportamental .....	58
Figura 46 - Dia 5: Comportamento real .....	58
Figura 47 - Dia 5: Predição.....	59
Figura 48 - Dia 5: Análises .....	59
Figura 49 - Dia 5: Comparativo de séries .....	60
Figura 50 - Dia 5: Assinatura comportamental .....	61
Figura 51 - Dia 1: Comportamento real .....	62
Figura 52 - Dia 1: Predição.....	62
Figura 53 - Dia 1: Análises .....	63
Figura 54 - Dia 1: Comparativo de séries .....	64
Figura 55 - Dia 1: Assinatura comportamental .....	65
Figura 56 - Dia 2: Comportamento real .....	65
Figura 57 - Dia 2: Predição.....	66
Figura 58 - Dia 2: Análises .....	66
Figura 59 - Dia 2: Comparativo de séries .....	67
Figura 60 - Dia 2: Assinatura comportamental .....	68
Figura 61 - Dia 3: Comportamento real .....	68
Figura 62 - Dia 3: Predição.....	69
Figura 63 - Dia 3: Análises .....	69
Figura 64 - Dia 3: Comparativo de séries .....	70
Figura 65 - Dia 3: Assinatura comportamental .....	71
Figura 66 - Dia 4: Comportamento real .....	71
Figura 67 - Dia 4: Predição.....	72
Figura 68 - Dia 4: Análises .....	72
Figura 69 - Dia 4: Comparativo de séries .....	73
Figura 70 - Dia 4: Assinatura comportamental .....	74
Figura 71 - Dia 5: Comportamento real .....	74
Figura 72 - Dia 5: Predição.....	75
Figura 73 - Dia 5: Análises .....	75
Figura 74 - Dia 5: Comparativo de séries .....	76
Figura 75 - Dia 5: Assinatura comportamental .....	77

## LISTA DE TABELAS

Tabela 1 - Coletas de uso de CPU.....	41
Tabela 2 - Predição dos dados.....	43

## LISTA DE ABREVIATURAS E SIGLAS

### LISTA DE ABREVIATURAS

CRM	<i>Computational Resource Management System</i>
CSV	<i>Comma-separated values</i>
CPU	Unidade Central de Processamento
MCU	<i>Multipoint Control Unit</i>
MMS	Média Móvel Simples
SRG	Sistema de Gerenciamento de Recursos Computacionais
RAM	<i>Random Access Memory</i>
TXT	<i>Text</i>

### LISTA DE SIGLAS

IDE	<i>Integrated Development Environment</i>
LaCa	Laboratório de Computação Aplicada
PERFMON	Monitor de Performance
PG	Ponta Grossa
UTFPR	Universidade Tecnológica Federal do Paraná

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>13</b>
1.1 TRABALHOS RELACIONADOS .....	14
1.2 OBJETIVOS.....	15
1.2.1 Objetivo Geral.....	15
1.2.2 Objetivos Específicos.....	15
1.3 METODOLOGIA DO TRABALHO.....	16
<b>2 MÉTODOS PARA PREDIÇÃO EM SÉRIES TEMPORAIS</b> .....	<b>18</b>
2.1 PREDIÇÃO E SÉRIE TEMPORAL: UMA VISÃO GERAL.....	18
2.2 MÉTODOS PREDITIVOS .....	20
2.3 SUAVIZAÇÃO EXPONENCIAL: <i>HOLD-WINTERS</i> .....	20
2.4 SUAVIZAÇÃO LINEAR: MÉDIAS MÓVEIS .....	21
2.5 ANÁLISE DE SÉRIES TEMPORAIS.....	22
2.5.1 Verificação de sazonalidades .....	22
2.5.2 Verificação de convergência.....	23
2.5.2.1 Distribuição normal.....	24
<b>3 COLETA DE DADOS DE RECURSOS COMPUTACIONAIS</b> .....	<b>26</b>
3.1 DEFINIÇÃO DE FERRAMENTA DE COLETA.....	26
3.1.1 Comparação de ferramentas de coleta.....	26
3.2 TRATAMENTO DOS DADOS.....	30
3.2.1 Ferramentas para tratamento de dados.....	30
3.2.2 <i>Microsoft Excel</i> .....	31
3.2.3 <i>R Studio</i> .....	33
3.2.4 Código em C: Média Móveis Simples .....	36
<b>4 CENÁRIO DA COLETA DE DADOS</b> .....	<b>38</b>
4.1 CONJUNTO DE MICROCOMPUTADORES.....	38
4.2 METODOLOGIA DE COLETA .....	38
<b>5 RESULTADOS E DISCUSSÃO</b> .....	<b>40</b>
5.1 MÉDIAS MÓVEIS SIMPLES: JUSTIFICATIVA.....	40
5.2 MICROCOMPUTADOR 1: RESULTADOS .....	40
5.3 MICROCOMPUTADOR 2: RESULTADOS .....	61
5.4 SAZONALIDADE ESTOCÁSTICA NULA: EXPLICAÇÃO.....	77
<b>6 CONCLUSÃO</b> .....	<b>78</b>
6.2 PUBLICAÇÕES .....	78
<b>REFERÊNCIAS</b> .....	<b>80</b>
<b>ANEXO A - Código em R: <i>Script</i> para tratamento de séries temporais</b> .....	<b>81</b>
<b>ANEXO B - Código em C: Cálculo de MMS</b> .....	<b>83</b>

## 1 INTRODUÇÃO

Quando um usuário inicializa um microcomputador, acredita-se que o mesmo vai realizar alguma tarefa do seu interesse, como ler e-mails ou navegar na *web*. Aplicações como estas, corriqueiras do dia-a-dia, consomem poucos recursos computacionais, acarretando em uma subutilização e conseqüente desperdício da sua capacidade de processamento.

Várias abordagens têm sido estudadas visando minimizar este desperdício de potencial computacional, como pode ser constatado nos estudos de Góis (2009). Algumas abordagens exploram o compartilhamento deste equipamento entre vários usuários de acordo com uma escala pré-definida, outras se baseiam em processos de virtualização, e ainda, existem algumas que buscam a agregação destes equipamentos a ambientes que possam utilizá-los remotamente.

Estes ambientes são altamente escalonáveis e podem ser modelados segundo vários critérios que auxiliem nos processos de agregação, como são os casos envolvendo as plataformas P2P para transferência de arquivos entre usuários associados, dos clusters de equipamentos fracamente acoplados e das grades, citados por Tanenbaum e Wetherall (2011), as quais alcançam grandes quantidades de máquinas e com uma alta heterogeneidade.

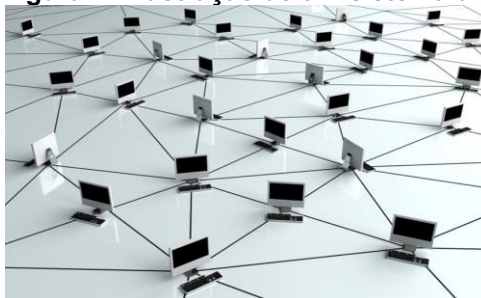
Um problema comum entre estes ambientes é a dificuldade na identificação destes equipamentos e o conhecimento de seus comportamentos no tempo. Entretanto, conhecer estes comportamentos seria sempre uma situação passada de seus índices de utilização e nunca sua posição atual. Alguns estudos visam o aproveitamento deste conhecimento organizando os dados geradores em séries temporais, as quais, mediante técnicas de análises podem suprir esta deficiência introduzindo conceito da previsibilidade, como é visto no trabalho de Silva et al. (2008). Prever o comportamento futuro é muito mais aproveitável para um sistema de gerenciamento de recursos do que seu comportamento passado.

Este trabalho explora o contexto de utilização de um equipamento remotamente, através do conhecimento prévio de seu comportamento e provendo subsídios para que se possam expandir este conhecimento para um futuro próximo. Partindo desta premissa, este trabalho irá apresentar um modelo de previsibilidade do comportamento futuro de recursos computacionais, afim de se otimizar o uso e diminuir o desperdício da capacidade computacional dos mesmos.

## 1.1 TRABALHOS RELACIONADOS

A utilização racional dos recursos computacionais é relevante nos dias atuais, principalmente em um contexto de sistemas distribuídos, que segundo Tanenbaum e Wetherall (2011) é um conjunto de computadores independentes entre si (e possivelmente diferentes), ligados por meio de uma rede de dados, que se apresentam aos usuários como um sistema único e coerente, como o exemplificado na Figura 1, que pode em muitos momentos apresentar equipamentos ociosos, ou seja, estarem realizando somente atividades básicas e com pouca participação nas tarefas do sistema distribuído.

**Figura 1 - Ilustração de um sistema distribuído**



**Fonte: Pplware (2015).**

Uma possível problemática onde a ociosidade de recursos computacionais poderia ser explorada é o trabalho de Roesler et al. (2018), que em breve síntese consiste no aprimoramento de um sistema MCU (*Multipoint Control Unit*) em nível de software, para diminuir o custo financeiro. Na hipótese de haver mais de um MCU disponível no sistema distribuído, é interessante possuir o percentual de uso desse recurso previamente, pois dessa forma é possível evitar que haja sobrecarga em um MCU e deixar outro em estado ocioso. O trabalho de Góis (2009) também trabalha com a possibilidade de se tentar prever eventos futuros.

A problemática de prever acontecimentos futuros também é explorada em outras áreas do conhecimento, como pode ser constatado nos estudos de Silva et al. (2008), em que se busca a previsão da temperatura média de Uberlândia – MG, por meio de séries temporais. O trabalho de Ehlers (2009) explica alguns modelos preditivos, como por exemplo, o Holt-Winters, um modelo de suavização exponencial dos dados.

Se estes estudos estiverem voltados para recursos computacionais, é possível que a partir da coleta dados de utilização da CPU de cada máquina em um sistema distribuído, gerar séries temporais (percentual de uso de CPU x intervalos pré-definidos) e descobrir como os recursos monitorados destes equipamentos se comportam no passado.

Entretanto, para um SGR, o mais importante é prever como estes comportamentos estarão em um futuro próximo. Partindo desta premissa, uma abordagem largamente aplicada em séries temporais é o uso de médias móveis. Segundo Ehlers (2009), um modelo matemático que realiza a predição, ou seja, a projeção do comportamento dos dados analisados de modo que se possa planejar e escalonar onde uma determinada tarefa será executada no futuro, escolhendo com sucesso uma máquina com capacidade de processamento para tal.

## 1.2 OBJETIVOS

Para a execução deste trabalho, tem-se um objetivo principal, que para atingi-lo se faz necessário delimitar tarefas intermediárias, como será visto logo abaixo.

### 1.2.1 Objetivo Geral

Este trabalho tem como objetivo principal viabilizar um modelo de previsão do comportamento de recursos computacionais através de análises qualitativas de séries temporais obtidas de equipamentos monitorados, buscando com isso, a definição de tendências, sazonalidades e anomalias com a aplicação das médias móveis na definição da assinatura comportamental destes equipamentos e, a comparação desta com o comportamento real dos mesmos.

### 1.2.2 Objetivos Específicos

Para cumprir com o objetivo proposto, ou seja, analisar o comportamento das séries temporais geradas pelas coletas do percentual de uso de CPU, foram seguidos alguns passos, como pode ser visto a seguir:



- Compreender o funcionamento de séries temporais;
- Verificar o comportamento de um equipamento, através da coleta do percentual de uso de CPU;
- Estudar médias móveis e suas variações;
- Aplicar médias móveis nas séries temporais para definição da assinatura comportamental;
- Analisar a assinatura comportamental em relação ao comportamento real.

### 1.3 METODOLOGIA DO TRABALHO

Para o início dos estudos, foi necessário realizar uma revisão sistemática da literatura acerca de séries temporais: conceitos elementares e uma visão geral sobre predição em séries históricas. De posse desse ferramental teórico, foi visto um comparativo de métodos preditivos e em que situação seu uso é recomendado.

Após o comparativo, foi aprofundado o entendimento sobre médias móveis e suas variantes, que foi utilizada para realizar a predição nas séries históricas, neste tópico será detalhado sobre seu funcionamento. Também está contido um pequeno exemplo de seu funcionamento.

Realiza a predição, foi necessário realizar a análise da série temporal gerada, por meio de dois parâmetros: sazonalidade e convergência. Para realizar estas duas análises, foram descritos modelos matemáticos e como seu uso deve ser feito nas séries temporais. Com isto visto, encerra-se o tópico de ferramentas de predição em séries temporais.

É necessário possuir alguma maneira de coletar o recurso computacional nos microcomputadores, portanto o próximo tópico será sobre as ferramentas de coletas nos principais sistemas operacionais do mercado. Foi feito um comparativo de funcionamento dentre as principais ferramentas: suas similaridades e passo a passo de uso.

Com os dados coletados, foi necessário tratá-los afim de conseguir aplicar o método preditivo escolhido, portanto apresentou-se algumas ferramentas de tratamento de dados e como deve ser utilizado; ocorreu a descrição de uso do

*Microsoft Excel*, que foi uma das ferramentas utilizada neste trabalho. Deste modo finalizando o tópico de coleta de dados de recurso computacional.

De posse de todas as ferramentas necessárias, foi necessário delimitar o cenário de coleta de dados, ou seja, especificar qual o conjunto de microcomputadores que serão coletados os dados. Foi apresentada a metodologia de coleta de dados, ou seja, como que se procedeu desde a execução da ferramenta de captura dos dados até o tratamento dos mesmos. Desta maneira finalizando o tópico sobre o cenário de coleta de dados.

Definida a metodologia de coleta, pode-se partir para os resultados e discussões, que foram apresentadas todas as coletas de dados dos microcomputadores, a predição realizada e as análises de convergência e sazonalidade, tudo demonstrado por meio de gráficos para facilitar a visualização do que está acontecendo.

## 2 MÉTODOS PARA PREDIÇÃO EM SÉRIES TEMPORAIS

O início dos estudos para a solução do problema apresentado no capítulo anterior parte de um levantamento sobre séries temporais e métodos preditivos. Para tanto, serão elencados os conceitos elementares de séries temporais e predição, um comparativo de alguns métodos matemáticos de predição e ferramentas de análises de séries temporais.

### 2.1 PREDIÇÃO E SÉRIE TEMPORAL: UMA VISÃO GERAL

Segundo Ehlers (2009), uma série temporal é uma coleção de observações feitas sequencialmente ao longo do tempo. É interessante atentar que observações vizinhas são dependentes e é um objetivo modelar e analisar essa dependência.

Há uma vasta área de aplicação de séries temporais, podendo ser citadas como exemplo, Economia (taxa mensal de empregos, produção em uma linha de montagem), Epidemiologia (Número diário de novos casos de COVID-19), Meteorologia (Índices de precipitação pluviométrica), entre muitos outros.

Uma série temporal pode, segundo Ehlers (2009), ser denominada contínua quando as observações são feitas continuamente no tempo. Sendo o conjunto definido por  $T = \{t: t_1 < t < t_2\}$  a série temporal é definida por  $\{X(t): t \in T\}$ . Uma série temporal é dita discreta quando as observações em intervalos equiespaçados. Sendo o conjunto  $T = \{t_1, \dots, t_n\}$ , onde  $t_n$  é valor da unidade temporal. Para fins de simplificação, pode-se escrever  $T = \{1, 2, \dots, n\}$ .

Os termos contidos em  $T$  não se referem a variável observada  $X$ , pois esta pode assumir valores discretos ou contínuos. Em diversos casos  $X$  pode ser discreta por definição, como por exemplo, o número de casos diários notificados de COVID-19, porém para análises estatísticas, podem ser tratadas como valores contínuos, caso não sejam valores pequenos.

Existem diversas finalidades para se construir uma série temporal, segundo Ehlers (2009), as principais finalidades são as seguintes:

- **Descrição:** Descrever propriedades da série, como por exemplo padrão de tendência, variações cíclicas ou sazonais, observações discrepantes, etc;

- **Explicação:** Usar a variação em uma série histórica para explicar a variação em outra série;
- **Predição:** Prever valores futuros com base em valores históricos. Deve-se assumir que o futuro envolve incertezas, previsões não são perfeitas. É necessário tentar mitigar os erros da predição;
- **Controle:** Os valores da série temporal, por exemplo, medem a precisão de um processo de manufatura e o objetivo é o controle deste processo.

Pelo propósito do trabalho aqui descrito, fica demonstrado que o principal objetivo de utilização de séries temporais é a predição de valores futuros para a definição da assinatura comportamental, que nada mais é que uma série temporal após a aplicação de métodos matemáticos afim de aprimorá-la, dos equipamentos que serão estudados.

Existem diversas abordagens de séries temporais, segundo Ehlers (2009), assim as descreve:

- **Técnicas descritivas:** Gráficos, identificações de padrões, etc;
- **Modelos probabilísticos:** Seleção, comparação e adequação de modelos e predição;
- **Análise espectral;**
- **Métodos não paramétricos:** Alisamento ou suavização;
- **Outras abordagens:** Modelos não lineares, séries multivaloradas, etc.

As abordagens descritivas e de modelos probabilísticos serão amplamente utilizadas neste trabalho, visto que a geração de gráficos é um importante método para visualização do que está ocorrendo e os modelos probabilísticos são as ferramentas para a geração da assinatura comportamental.

Dentro do contexto de séries temporais, há dois conceitos elementares: sazonalidade e tendência. O primeiro, segundo Ehlers (2009), é quando ocorre de um comportamento se repetir a cada determinado período de tempo, e o segundo é simplesmente a verificação da tendência de crescimento ou decréscimo dos valores observados ou previstos para um determinado período.

## 2.2 MÉTODOS PREDITIVOS

Na literatura há várias categorias para se realizar predição em séries temporais, e dentro dessas existem diversos métodos. Nos próximos dois tópicos, serão apresentados duas categorias e dois métodos: suavização exponencial e o método *Holt-Winters*, suavização linear com o método médias móveis.

### 2.3 SUAUIZAÇÃO EXPONENCIAL: *HOLT-WINTERS*

O modelo matemático de *Holt-Winters*, que segundo Ehlers (2009), pode ser generalizado para séries que contenham tendência e variação sazonal. Partindo de que ocorre observações mensais, e sejam  $(L_t, T_t, I_t)$  o nível, a tendência e o índice sazonal no tempo  $t$ , respectivamente. Então,  $T_t$  é o aumento ou redução esperada por mês no nível atual da série.

Supondo que no tempo  $t$  os termos  $(L_{t-1}, T_{t-1}, I_{t-1}), \dots, (L_1, T_1, I_1)$  sejam conhecidos. Portanto, após observar  $x_t$  os termos  $L_t, T_t, I_t$  são atualizados via alisamento exponencial. Se a variação sazonal for multiplicativa, ou seja, com valores observados que tendem a crescer ao longo do tempo, logo as equações de atualização na forma de recorrência são dadas por:

$$L_t = \alpha(x_t/I_{t-12}) + (1 - \alpha)(L_{t-1} + T_{t-1}) \quad (1)$$

$$T_t = \gamma(L_t - L_{t-1}) + (1 - \gamma)T_{t-1} \quad (2)$$

$$I_t = \delta(x_t/L_t) + (1 - \delta)I_{t-12} \quad (3)$$

E as previsões  $k$  períodos a frente é dada por:

$$\hat{x}_t(k) = L_t + kT_t + I_{t-12+k} \quad k = 1, 2, \dots \quad (4)$$

Os parâmetros de alisamento  $\alpha, \gamma, \delta$  para cada componente da série, que em geral são escolhidos no intervalo  $(0,1)$  e podem ser estimados utilizando a minimização da soma de quadrados dos erros de previsão, como descrito por Ehlers (2009).

## 2.4 SUAVIZAÇÃO LINEAR: MÉDIAS MÓVEIS

Um dos métodos mais elementares na literatura, médias móveis é a média de um determinado período de dados dispostos em uma série histórica, podendo estes dados possuir um peso para se calcular a média. Será detalhado abaixo sobre a Média Móvel Simples (MMS) Ehlers (2009).

Partindo de um simples exemplo: média de preços de um produto a cada 10 dias. Com o incremento das unidades temporais, que neste caso são os dias, novos dados vão substituindo os valores mais antigos, formando uma base amostral para elaboração da média que varia ao longo do tempo, essa variação chama-se janela móvel, e no exemplo citado são 10 dias. O comportamento destes preços pode ser verificado analisando-se a série MMS gerada pela equação 5.

$$MMS = \frac{\text{preço}_1 + \text{preço}_2 + \dots + \text{preço}_{10}}{10} \quad (5)$$

Seguindo a mesma lógica de raciocínio, seja um conjunto T definido com os valores:  $T = \{\text{valor}_1, \text{valor}_2, \text{valor}_3, \dots, \text{valor}_n\}$  portanto de tamanho  $n$ , o cálculo da MMS pode ser assim generalizado:

$$MMS = \frac{\text{valor}_1 + \text{valor}_2 + \dots + \text{valor}_n}{n} \quad (6)$$

A média móvel simples atribui peso igual para cada componente, ou seja, todos os dados possuem o mesmo peso na média. Tal método é eficaz para valores não discrepantes entre si, pois como se trata de média simples, há uma forte tendência de os valores calculados tenderem a média da janela móvel, desta forma não representando com precisão valores com desvio padrão elevado. Neste estudo será considerado que o comportamento dos equipamentos analisados segue uma tendência de utilização e conseqüentemente, com anomalias facilmente detectadas.

## 2.5 ANÁLISE DE SÉRIES TEMPORAIS

Após a geração da série temporal, é possível verificar algumas propriedades, para isso utilizando métodos matemáticos. Neste tópico serão abordados a verificação de sazonalidades e também de convergência, que compara a assinatura comportamental gerada pela predição com os valores que continuam sendo coletados, para ser possível verificar se o método está gerando um resultado convergente com a realidade.

### 2.5.1 Verificação de sazonalidades

Pode-se dizer que uma série temporal possui sazonalidade estocástica, segundo Ehlers (2009), quando a componente sazonal da série em estudo varia com o passar do tempo. Este procedimento pode ser usado quando acontece um padrão sazonal constante. Seja a série temporal denominada como  $Z_t$  e  $\hat{T}_t$  um estimador para a tendência calculado previamente, considera-se a seguinte série temporal:

$$Y_t = Z_t - \hat{T}_t \quad (7)$$

Considerando o caso em que se tem um padrão sazonal constante, utiliza-se  $Y_t$  para estimar  $S_t$ . Para fins de exemplificação, será considerado dados coletados mensalmente, tomando a média por mês.

$$\hat{y}_{.j} = \frac{1}{n_j} \sum_{i=1}^{n_j} Y_{ij} \quad j = 1, \dots, 12 \quad (8)$$

Em grande parte dos casos, a soma dos  $\hat{y}_{.j}$  não é zero, portanto, pode ser tomado como estimativa das constantes sazonais.

$$\hat{s}_j = \hat{y}_{.j} - \hat{y} \quad \text{onde, } \hat{y} = \frac{1}{12} \sum_{i=1}^{12} \hat{y}_{.j} \quad (9)$$

Portanto, a série original  $Z_t$  pode ser assim descrita, na forma aditiva:

$$Z_t = T_t + S_j + a_t \quad (10)$$

Sendo  $t = 12i + j; i = 0, 1, \dots, p-1; j = 1, 2, \dots, 12$ , onde  $p$  é o número de meses, por exemplo. Finalmente, a série temporal sem sazonalidades pode ser assim descrita:

$$z_t^* = z_t - \hat{s}_t \quad (11)$$

### 2.5.2 Verificação de convergência

Para verificar se a assinatura comportamental gerada pelo método preditivo está convergindo com os dados que estão acontecendo na realidade, pode-se utilizar o teste de *Wald-Wolfowitz*, com aplicação em suavizações lineares e exponenciais.

Segundo Strobel (1987), seja  $\{X_t, t = 1, \dots, N\}$  uma série temporal com  $N$  observações. Onde  $M$  sendo a mediana das  $N$  observações de  $X_t$ . Atribui-se a cada valor de  $X_t$  o símbolo "i" se ele for maior ou igual a  $M$ , e "j" caso ele for menor que  $M$ . Então, têm-se que  $N = ("N_i" \text{ pontos "i"}) + ("N_j" \text{ pontos "j"})$ . Desta forma, tem-se grupos de observações marcadas com "i" e grupos de observações marcadas com "j" ao longo da série temporal. O número total de grupos será a estatística do teste, isto é:  $K = \text{"número de grupos com símbolos iguais"}$ .

Considera-se as seguintes hipóteses:

$$\begin{cases} H_0 = \text{Não existe convergência;} \\ H_1 = \text{Existe convergência.} \end{cases}$$

Rejeita-se a hipótese nula  $H_0$  se houver um número pequeno de grupos com símbolos iguais, ou seja, se  $K$  for relativamente pequeno. Para valores de  $N_i$  e  $N_j$  superiores a 30, pode-se utilizar o Teorema Central do Limite e aproximar a distribuição de  $K$  por uma normal, esta distribuição de probabilidades será melhor explanada em outro tópico. O resultado após a aplicação deste método pode ser assim interpretado via teste de hipótese:



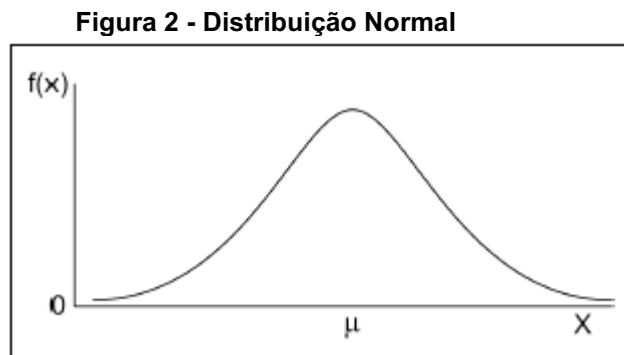
$\left\{ \begin{array}{l} \text{Se maior ou igual a 50\%} = \text{Há convergência;} \\ \text{Se menor que 50\%} = \text{Não há convergência.} \end{array} \right.$

### 2.5.2.1 Distribuição normal

Como o teste de convergência de *Wald-Wolfowitz* não consegue retornar uma resposta para um conjunto grande de pontos, a distribuição normal de probabilidades resolve este problema. Segundo Ehlres (2009),  $X$  tem distribuição de normal com parâmetros  $\mu$  e  $\sigma^2$ , sendo denotado por  $X \sim N(\mu, \sigma^2)$  se sua função de densidade é dada por:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad (12)$$

Sendo  $-\infty < \mu < \infty$  e  $\sigma^2 > 0$ . Seu gráfico tem a forma de um sino, como pode ser conferido na Figura 2:



Fonte: Autoria própria (2020).

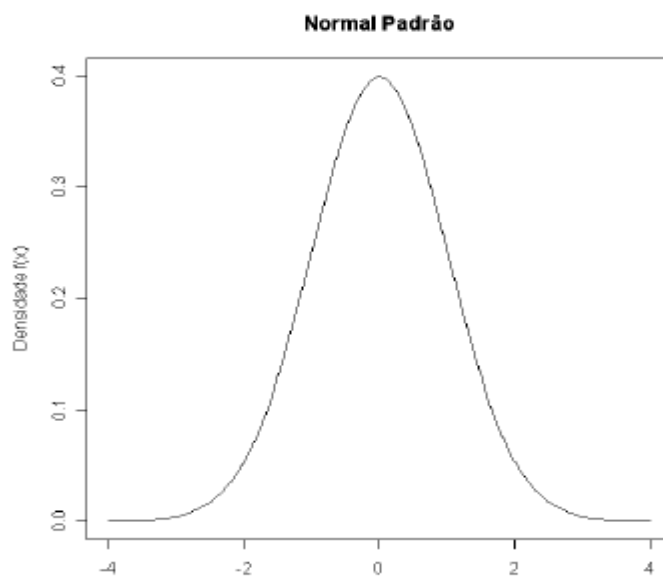
Quando  $\mu = 0$  e  $\sigma^2 = 1$  a distribuição de probabilidades é chamada de normal padrão. Deste modo, pode ser reduzida para uma variável  $Z$ , fazendo o seguinte cálculo:

$$Z = \frac{x-\mu}{\sigma} \quad (13)$$

Assim a distribuição possui média  $\mu = 0$  e desvio padrão  $\sigma = 1$ . Pelo fato de a distribuição ser simétrica em relação à média  $\mu = 0$  a área à direita é igual a área

à esquerda de  $\mu$ . Por ser uma distribuição muito usada, existem tabelas a qual encontram-se a resolução de suas integrais. Assim, a tabela fornece áreas acima de valores não negativos que vão desde 0,00 até 4,09. Na Figura 3 segue o gráfico da curva Normal padronizada.

**Figura 3 - Curva Normal Padronizada**



**Fonte: Autoria própria (2020).**

### 3 COLETA DE DADOS DE RECURSOS COMPUTACIONAIS

Para ser possível realizar a coleta de algum dado de recurso computacional, foi preciso utilizar algumas ferramentas que realizem esta tarefa. Neste capítulo estão demonstradas algumas dessas ferramentas de captura e como tratar os dados coletados em ferramentas de planilhas.

#### 3.1 DEFINIÇÃO DE FERRAMENTA DE COLETA

Um sistema operacional segundo Tanenbaum e Bos (2016) gera diversas informações sobre suas atividades, como por exemplo taxas de uso de memória RAM, consumo de energia elétrica, tempo de execução de aplicações, etc. E para gerenciar tais dados, se faz necessário uma ferramenta que controle e organize tais informações de modo que se possa obtê-las com facilidade.

Nos sistemas *Linux* e *Windows* há uma categoria de ferramentas chamadas de monitores de desempenho, sendo que no primeiro existem algumas ferramentas e aqui será demonstrada a *GNOME System Monitor* e no último há uma ferramenta nativa de monitoramento de desempenho, denominada *PERFMON*, que será apresentada logo abaixo.

##### 3.1.1 Comparação de ferramentas de coleta

Começando a demonstração pela aplicação da família *Linux*, *GNOME System Monitor* informa ao usuário da aplicação diversas informações sobre percentual de uso de CPU, taxa de uso de memória RAM, disco rígido, etc. A Figura 4 mostra seu *layout* principal.

**Figura 4 - Tela principal GNOME System Monitor**



Fonte: Autoria própria (2020).

Após a instalação da aplicação, pode-se configurar os parâmetros que serão observados. Na seguinte tela, contida na Figura 5, estão os dados que podem ser observados e algumas configurações disponíveis:

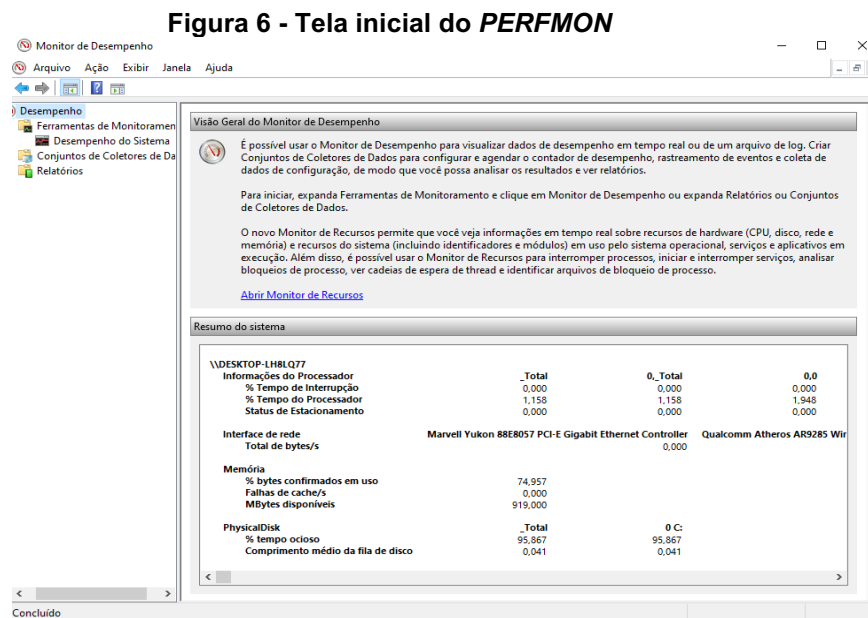
**Figura 5 - Tela de configuração GNOME System Monitor**



Fonte: Autoria própria (2020).

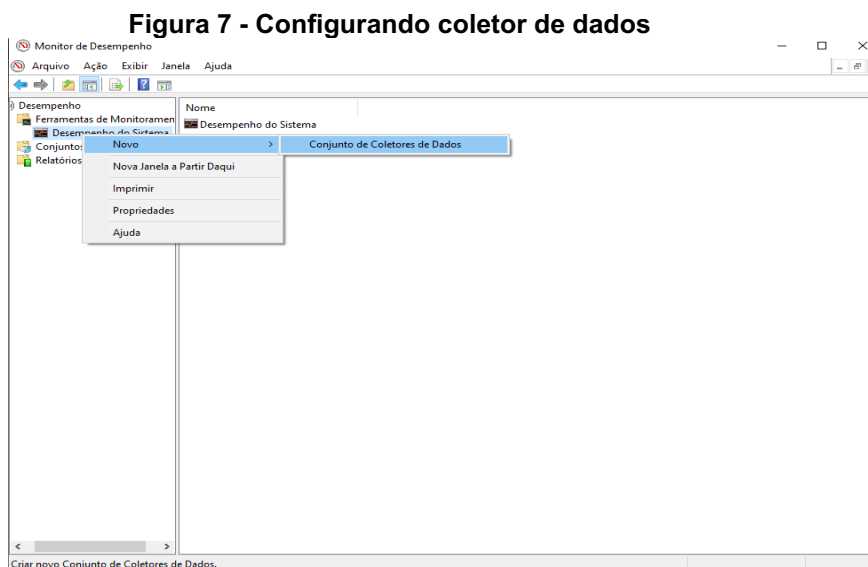
Na aba referente ao processador (CPU) é possível configurar para a aplicação salvar em um arquivo separado por vírgulas (.csv), deste modo sendo possível tratar os dados coletados em um editor de planilhas, como por exemplo o *Microsoft Excel*.

A ferramenta nativa do *Windows*, *PERMON*, possui um acervo de informações um pouco mais amplo que a ferramenta demonstrada anteriormente. A tela inicial da aplicação é demonstrada na Figura 6, onde se começa a configuração dos parâmetros que serão observados.



Fonte: Autoria própria (2020).

Para configurar os parâmetros de coleta, é necessário acessar em ferramentas de monitoramento, Desempenho do Sistema, Novo e finalmente em Conjunto de Coletores de dados, como pode ser observado na Figura 7:

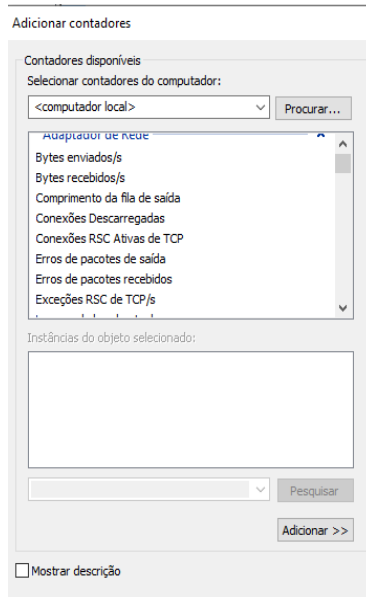


Fonte: Autoria própria (2020).

Uma opção interessante nesta aplicação é a possibilidade de gravar o *script* de configuração de coleta dos dados para ser executado em outro microcomputador. Essa funcionalidade facilita a execução do trabalho, pois assim só é necessário configurar os parâmetros uma única vez e executá-lo nas máquinas desejadas.

Após a geração do *script*, é momento de se selecionar os recursos que o monitor de desempenho irá verificar durante sua execução. O *PERFMON* traz uma vasta gama de possibilidades, como pode ser visto na Figura 8:

**Figura 8 - Contadores do *PERFMON***



**Fonte: Autoria própria (2020).**

Selecionando o parametro desejado para o trabalho, percentual de uso de CPU, a periodicidade das observações e o tempo total, o *script* está pronto para ser executado em qualquer máquina *Windows*.

Para este trabalho, como os microcomputadores analisados possuem como sistema operacional *Microsoft Windows*, a ferramenta *PERFMON* será utilizada. A possibilidade de se gerar um *script* para ser executado em qualquer máquina é uma funcionalidade a ser considerada, pois desta maneira o procedimento para captura dos dados será otimizado, visto que não é necessário sua configuração em todas as máquinas a cada vez que o monitor de desempenho é executado.

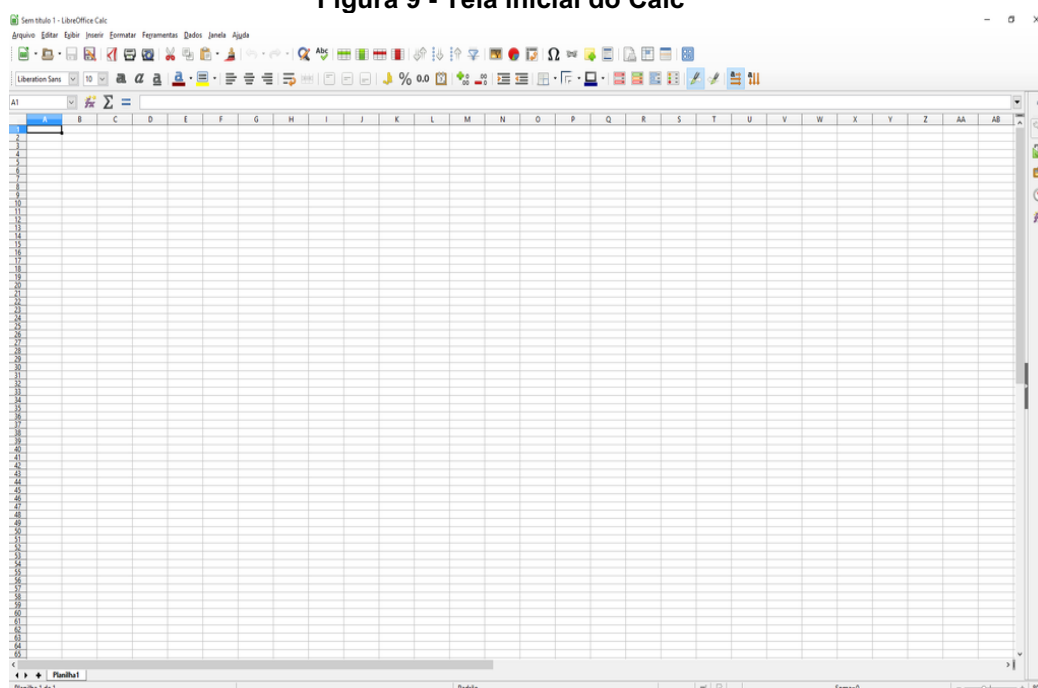
## 3.2 TRATAMENTO DOS DADOS

Com a execução diária do monitor de desempenho, foi gerado para cada dia um arquivo de extensão .csv contendo os dados coletados e mais algumas informações como nome do microcomputador e data da coleta dos dados, entre outras. Para efetivamente ser possível analisar os valores encontrados, é importante realizar uma espécie de processamento nos arquivos de coleta, tarefa que um editor de planilhas pode realizar.

### 3.2.1 Ferramentas para tratamento de dados

Existem várias ferramentas para tratamento de dados, em especial aqui será demonstrado duas categorias: editores de planilhas, com foco em duas ferramentas bem conhecidas, que são LibreOffice, com a sua ferramenta Calc e Microsoft Office com a sua ferramenta Excel, e também um *software* estatístico: *R Studio*. E findando a apresentação do ferramental necessário, será apresentado um código em linguagem de programação de computadores em C, que realiza a aplicação de MMS na série temporal coletada.

Começando por Calc, sua tela principal reúne praticamente todas suas funcionalidades, com uso bem intuitivo, como pode ser visto na Figura 9:

**Figura 9 - Tela inicial do Calc**

**Fonte: Autoria própria (2020).**

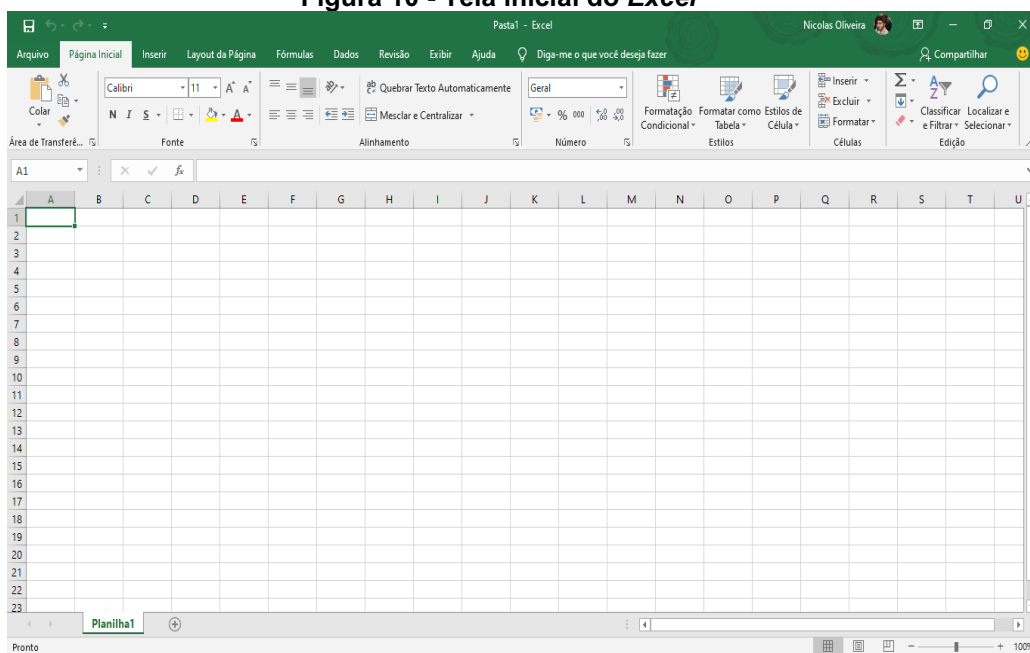
Um problema da ferramenta é por não possuir módulo de análise de dados, especificamente quanto a predição em séries temporais, que no Microsoft Office pode ser obtido baixando um módulo que contempla tal item. Dentro deste módulo também está contido o método de médias móveis, portanto a mesma ferramenta de tratamento dos dados também pode gerar a predição dos dados, de modo que será a escolhida para a continuação dos trabalhos.

### 3.2.2 *Microsoft Excel*

Diferente da ferramenta mencionada anteriormente, suas funcionalidades não estão todas disponíveis na tela principal, estão divididas em categorias de atividades, como ilustrado na Figura 10:



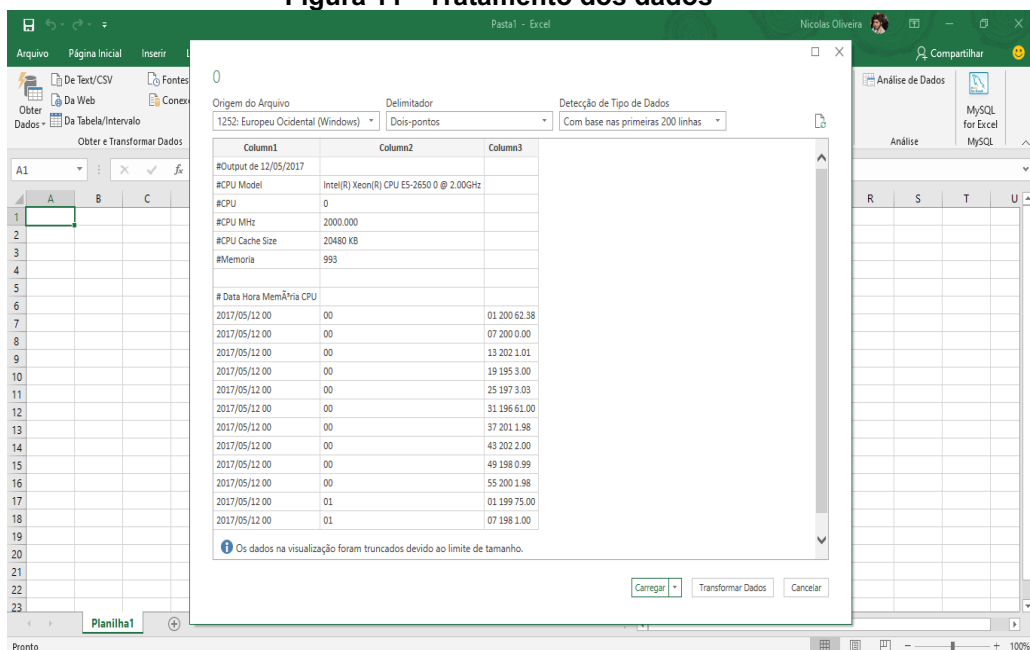
**Figura 10 - Tela inicial do Excel**



Fonte: Autoria própria (2020).

Para começar o tratamento dos dados coletados, é necessário ir até a aba Dados, Obter Dados, De Text/CSV. Nesta etapa é que será possível efetivamente tratar os dados, ou seja, deixá-los de modo que cada valor ocupe somente uma célula. A Figura 11 ilustra o tratamento dos dados.

**Figura 11 - Tratamento dos dados**



Fonte: Autoria própria (2020).

A Figura 12 mostra o estado final dos dados, que estão prontos para serem aplicados no método de predição escolhido para estudo.

**Figura 12 - Dados processados**

	Column1	Column2	Column3	Column4	Column5
2	#Output	de			
3	#CPU	Model:			
4	#CPU:	0			
5	#CPU	MHz:			
6	#CPU	Cache			
7	#Memoria:	993			
8					
9	#				
10	2017/05/12	00:00:01	0,02		
11	2017/05/12	00:00:07	0,02		
12	2017/05/12	00:00:13	2,02		
13	2017/05/12	00:00:19	1,95		
14	2017/05/12	00:00:25	1,97		
15	2017/05/12	00:00:31	1,96		
16	2017/05/12	00:00:37	2,01		
17	2017/05/12	00:00:43	2,02		
18	2017/05/12	00:00:49	1,98		
19	2017/05/12	00:00:55	0,02		
20	2017/05/12	00:01:01	1,99		
21	2017/05/12	00:01:07	1,98		
22	2017/05/12	00:01:13	2,02		
23	2017/05/12	00:01:19	1,92		

Fonte: Autoria própria (2020).

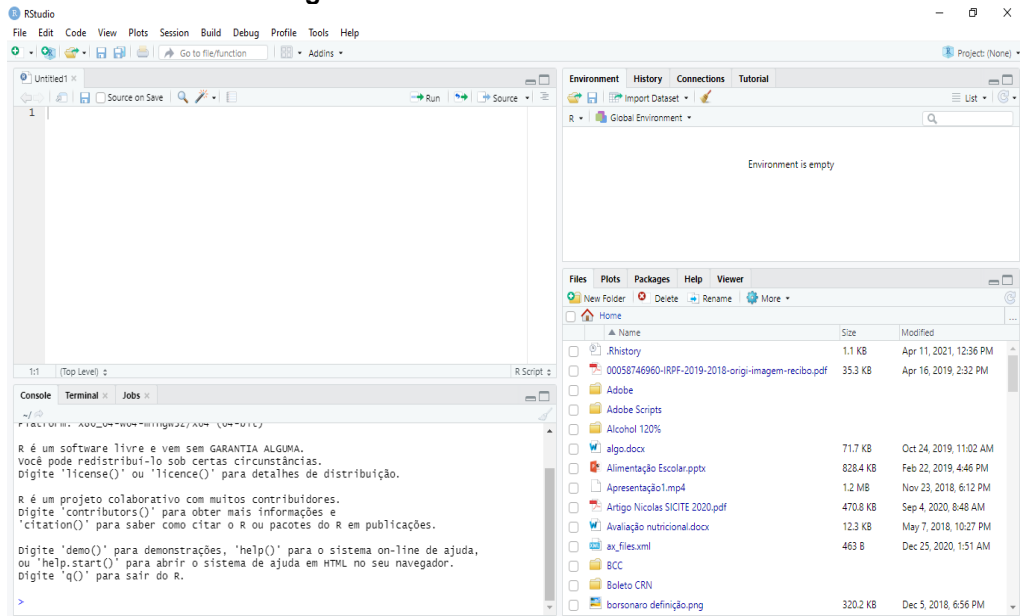
### 3.2.3 R Studio

Segundo Verzani (2011), *R Studio* é um *software* livre, mais especificamente uma IDE (*Integrated Development Environment*) da linguagem de programação R, que implementa funções e gráficos para cálculos estatísticos.

Este programa é compatível com diversas distribuições *Linux* e também com as plataformas *Windows*, sendo necessário instalar previamente o compilador e seus códigos-fonte; tal passo é necessário para a IDE ter seu funcionamento correto, pois ela somente disponibiliza a interface gráfica para implementação das funcionalidades presentes na linguagem de programação de computadores R.

A seguir, estará um breve passo a passo de como utilizar esta IDE, com ênfase no módulo que trata de séries temporais e predições. As imagens do software foram obtidas de uma instalação em um sistema *Windows*. A Figura 13 demonstra a tela inicial do sistema:

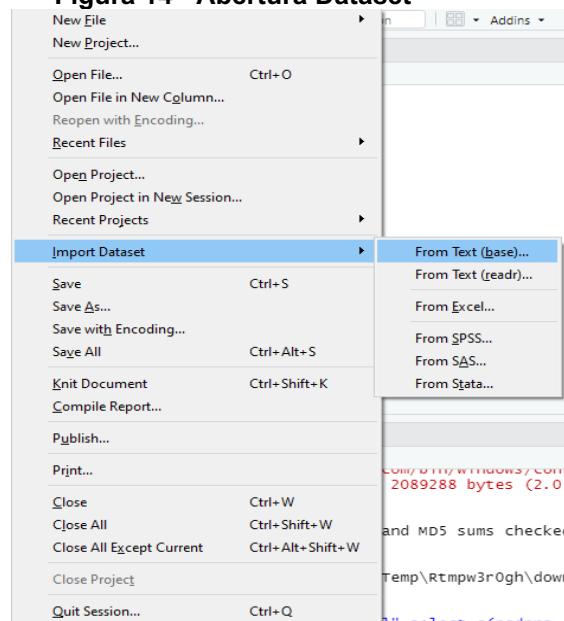
**Figura 13 - Tela inicial do R Studio**



**Fonte: Autoria própria (2020).**

A tela inicial reúne todas as funcionalidades, sendo necessário, para o problema proposto neste trabalho, a abertura do arquivo onde está contido os valores previstos através do método de Médias Móveis Simples. A Figura 14 ilustra os passos para a IDE processar os dados:

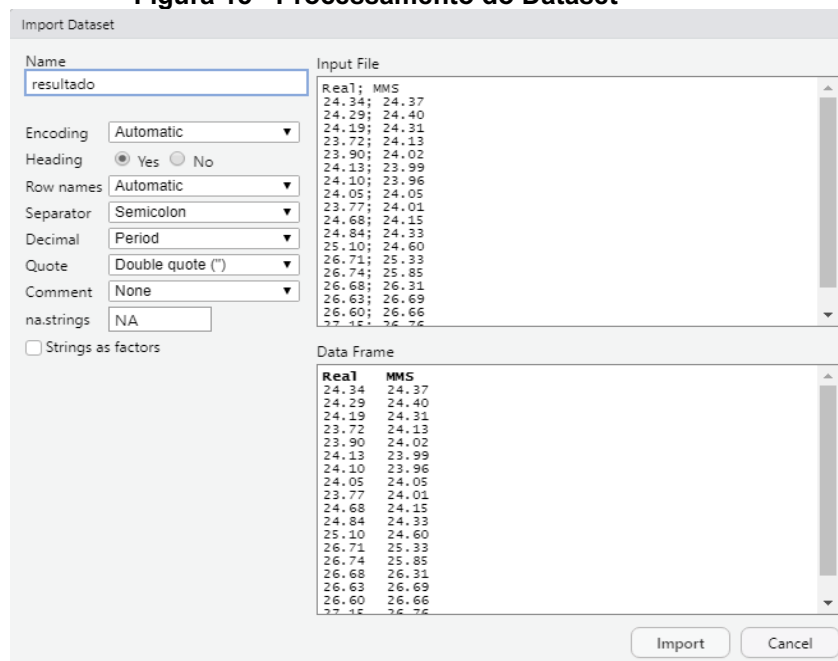
**Figura 14 - Abertura Dataset**



**Fonte: Autoria própria (2020).**

A IDE realiza o processamento dos dados e mostra na tela para verificação do usuário, inclusive realizando automaticamente a detecção de nomes das colunas, como é demonstrado na Figura 15:

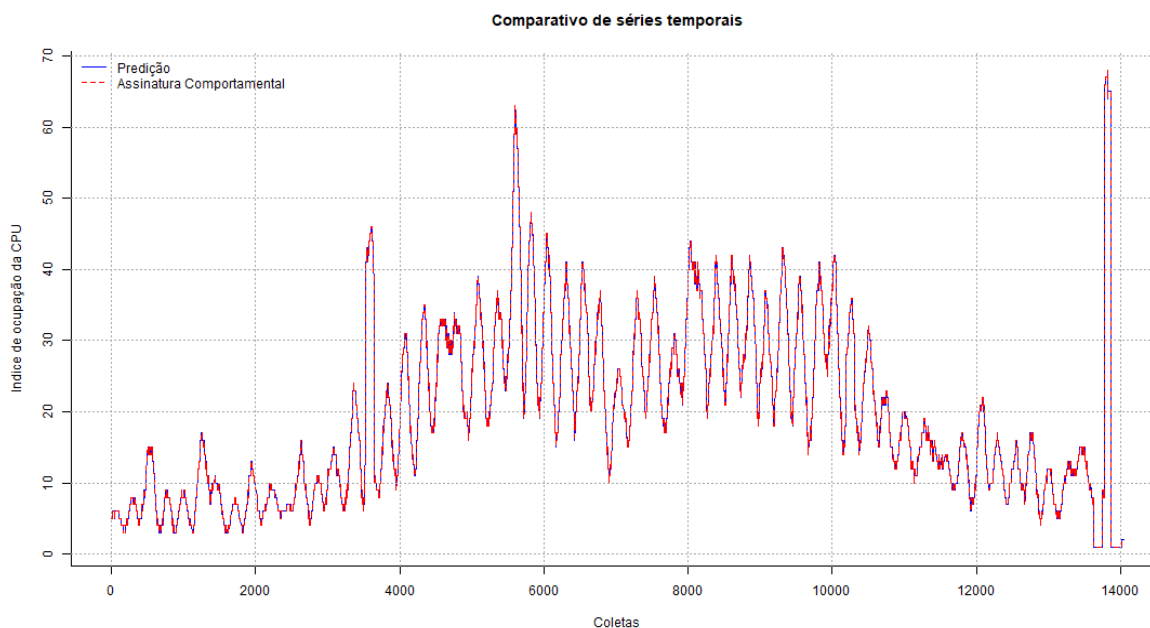
**Figura 15 - Processamento do Dataset**



**Fonte: Autoria própria (2020).**

Com os dados carregados, deve-se informar a IDE que os dados se tratam de uma série temporal. Este procedimento vai mudar a forma que os gráficos dos dados coletados serão plotados (por padrão, plota-se somente pontos no plano cartesiano), na Figura 16 é demonstrado o gráfico que é útil para este trabalho:

**Figura 16 - Exemplo de gráfico MMS vs Real**



**Fonte: Autoria própria (2020).**

O *R Studio*, segundo Verzani (2011), também dá suporte à geração de *scripts*, ou seja, é possível escrever um código em linguagem de programação de computadores R e salvá-lo. Com a possibilidade de criação de procedimentos, é uma forma de otimizar o processo, sendo somente necessário apontar a localização de cada arquivo que contém os dados coletados juntamente com a predição, para verificação de convergência e eliminação de possíveis sazonalidades.

Partindo dessa possibilidade de criação de *scripts*, desenvolveu-se um código em R, que tratará da análise de convergência, eliminação de possíveis sazonalidades e plotagem dos gráficos. Tal código está disponível no Anexo A deste trabalho; alguns trechos do mesmo serão explicados e detalhados durante a apresentação dos resultados do experimento.

### 3.2.4 Código em C: Média Móveis Simples

A linguagem de programação de computadores C, segundo Deitel e Deitel (2011), criada em 1972 por Dennis Ritchie, ainda é amplamente utilizada, por sua facilidade e também por ter servido de inspiração para outras linguagens, como por exemplo C++ e Java.

Para o cálculo da predição nas séries temporais, desenvolveu-se um código em linguagem C que recebe como entrada um arquivo no formato .csv contendo os percentuais de uso de CPU e como saída fornece um outro arquivo no mesmo formato, com os valores previamente coletados e também com a predição realizada. Tal código está presente no Anexo B deste trabalho; alguns trechos do mesmo serão explicados e detalhados durante a apresentação dos resultados do experimento.

## 4 CENÁRIO DA COLETA DE DADOS

Para a execução do trabalho, fez-se necessário dispor de um conjunto de microcomputadores para ser possível realizar a coleta do recurso computacional em estudo. Neste capítulo serão definidos o conjunto de microcomputadores e a metodologia das coletas.

### 4.1 CONJUNTO DE MICROCOMPUTADORES

Foram utilizados 2 microcomputadores do Laboratório de Computação Aplicada (LaCa), localizado na UTFPR Câmpus Ponta Grossa, com a seguinte especificação técnica:

- *Microsoft Windows 10 64bits;*
- *Placa Mãe HP Compaq 6005 Pro SFF PC;*
- *Processador QuadCore AMD Phenom II X4 B97;*
- *Memória 2x4GB;*
- *HD 500GB.*

Para ser possível a simulação de um sistema distribuído, se fez necessário a utilização de um equipamento concentrador de rede, mais especificamente um *switch* da marca Cisco, que estava disponível no LaCa. E a conexão entre os microcomputadores e o concentrador foi feita através de cabos GigaEthernet, que se encontravam no laboratório.

### 4.2 METODOLOGIA DE COLETA

Com o sistema operacional *Microsoft Windows* em funcionamento, iniciou-se a aplicação *PERFMON*, como já foi explanado no capítulo 3. Selecionou-se o recurso que se desejava monitoramento, neste caso o percentual de uso de CPU, intervalo de observações, que neste caso foram a cada 6 segundos, e o tempo total de observações, que foram 24 horas diárias, das 00:00 horas até as 23:59 horas.

Com os parâmetros definidos, restou-se aguardar a execução da aplicação, que ao final gerou um arquivo com extensão *.csv* contendo os percentuais de uso de CPU no intervalo definido. Este arquivo está pronto para ser tratado em um editor de

planilhas, por exemplo o *Microsoft Excel*, para que se pudesse efetuar a leitura correta dos dados coletados. Após este processo, o arquivo estava pronto para ser processado afim de gerar a série temporal e calcular a assinatura comportamental do equipamento.

O procedimento descrito acima contemplava a execução e coleta em um microcomputador e em um único dia, porém o processo é o mesmo para todo o sistema distribuído de máquinas analisadas, tais coletas foram realizadas pelo período de 5 dias uteis seguidos, ou seja, uma semana.

Para fins de simulação, utilizou-se de um *dataset* de coletas realizadas previamente, que estava a disposição no LaCa. Tal escolha foi tomada devido à falta de tempo de se realizar um levantamento de dados com uma quantidade relevante de coletas, que é imprescindível para ser possível aplicar em um método preditivo; porém todos os passos descritos anteriormente foram seguidos, não acarretando em nenhum prejuízo para o trabalho.



## 5 RESULTADOS E DISCUSSÃO

Nesta seção estão os resultados encontrados, com gráficos para melhor visualização, e a explicação dos motivos que levaram a escolha de Médias Móveis Simples como ferramenta de predição em séries temporais neste contexto.

### 5.1 MÉDIAS MÓVEIS SIMPLES: JUSTIFICATIVA

O princípio de funcionamento de Médias Móveis, independentemente da sua variação (simples ou ponderada), é de um indicador de tendência, segundo Morettim e Tolo (2006), ou seja, este método preditivo irá determinar se há crescimento ou decréscimo do item em estudo em algum determinado período de tempo.

No contexto de percentual de utilização de CPU, encontrar essa tendência é relevante, pois dessa forma é possível inferir duas informações: se o microcomputador está em estado ocioso (realizando tarefas irrelevantes para o sistema distribuído e conseqüentemente com baixo percentual de uso de CPU) e a tendência de uso de CPU dentro da janela móvel (se está aumentando ou diminuindo).

De posse dessas informações, já é possível tomar decisões quanto a escolha de uma máquina em um sistema distribuído, visto que, se está disponível a tendência de uso de recurso computacional conjuntamente com o informe de possível ociosidade, a escolha para alocar uma aplicação computacional no futuro se dará de uma maneira muito mais segura e precisa, visto que tal escolha estará embasada em uma tendência previamente calculada.

### 5.2 MICROCOMPUTADOR 1: RESULTADOS

Neste primeiro microcomputador, estão descritos passo a passo a construção dos resultados encontrados, de modo a levar o perfeito entendimento do trabalho. Realizada a coleta conforme descrito no Capítulo 4.2, os dados estão contidos em um arquivo de extensão .csv, porém, por facilitar a abertura do mesmo no código em C que foi desenvolvido, os mesmos foram convertidos em um arquivo de extensão .txt, em nada afetando sua estrutura e conteúdo.

Os dados estão dispostos em duas colunas, a primeira é o horário da captura do recurso computacional e a segunda é o percentual de uso de CPU. Na Tabela 1 estão as 9 primeiras linhas do arquivo:

**Tabela 1 - Coletas de uso de CPU**

<b>Hora</b>	<b>CPU</b>
00:00:01	28.86
00:00:07	20.25
00:00:13	15.67
00:00:19	3.99
00:00:25	25.00
00:00:31	18.73
00:00:37	0.75
00:00:43	28.57
00:00:49	20.30
...	...

**Fonte: Autoria própria (2021).**

Com este arquivo em mãos, é possível submetê-lo no programa desenvolvido para o cálculo da MMS do conjunto; para o início, foi necessário informar a localização do mesmo, podendo estar em qualquer diretório do sistema, o trecho de código na Figura 17 é o responsável pela captura do diretório:

**Figura 17 - Inserindo diretório**

```
...
printf("Digite o nome do arquivo de coletas: ");
scanf ("%s", &arquivo);
...
```

**Fonte: Autoria própria (2021).**

Após a leitura da localização do arquivo, o programa irá transferir os dados para um vetor de registros do tipo coleta, com a seguinte declaração contido na Figura 18:

**Figura 18 - Struct coletas**

```

...
typedef struct coleta{

char hora[15];
float coleta;

}coleta;
...

```

**Fonte: Autoria própria (2021).**

Para a transferência ser possível, é necessário um laço de repetição afim de percorrer todo o arquivo de coletas, que é realizado na Figura 19:

**Figura 19 - Percorrendo arquivo de coletas**

```

...
while (!feof(entrada)){

    result = fscanf(entrada, "%s          %f\n",&hora , &coleta);
    if (result) // Se foi possível ler
        printf("Linha %d -> Hora: %s CPU: %.2f\n",inc, hora, coleta);
        coletas[inc-1].coleta = coleta;
        strcpy(coletas[inc-1].hora, hora);
        inc++;

}
...

```

**Fonte: Autoria própria (2021).**

Nesta etapa é possível enviar o vetor contendo as coletas para a função que irá calcular a MMS, através de um laço de repetição e salvá-los em um arquivo, como pode ser observado na Figura 20:

**Figura 20 - Enviando dados para função MMS**

```

...
for (i=janelaMovel-1; i<14641; i++){

    printf("%.2f\n", mediaMovel(&coletas[i].coleta, janelaMovel));
    fprintf(saida, "%.2f; %.2f\n", coletas[i].coleta,
mediaMovel (&coletas[i].coleta, janelaMovel));

}
...

```

**Fonte: Autoria própria (2021).**

A função *mediaMovel()* acumula os valores e divide-os pela janela móvel definida pelo problema, que neste caso é equivalente a 600 observações, ou seja, 1 hora; estes parâmetros são executados conforme o seguinte código na Figura 21:

**Figura 21 - Função MMS**

```

...
float mediaMovel(float *p, int janelaMovel) {

int i;
float soma = 0;

for(i=janelaMovel; i>0; i--){
    soma += *p;
    p--;
}
return soma/janelaMovel;

}
...

```

**Fonte: Autoria própria (2021).**

O valor de 14641 presente no laço de repetição que submete os valores para a função *mediaMovel()* é equivalente ao número de coletas realizadas no intervalo de 24 horas, pois são realizadas a cada 6 segundos. O resultado final após a execução deste programa é um arquivo de extensão .txt que contém os valores coletados e a predição realizada; na tabela 2 estão as 9 primeiras linhas do arquivo gerado:

**Tabela 2 - Predição dos dados**

<b>Real</b>	<b>MMS</b>
28.68	4.15
36.55	4.10
8.75	4.01
2.99	4.03
25.19	4.08
14.00	4.06
16.83	4.07
24.94	4.11
26.02	4.33
...	...

**Fonte: Autoria própria (2021).**

Com a MMS calculada, foi possível partir para a análise dos dados, utilizando o código em R elaborado para este trabalho. Para tal, se fez necessário apontar o

diretório onde se encontra o arquivo e informar o nome do arquivo, a Figura 22 demonstra a execução dessa tarefa:

**Figura 22 - Lendo arquivo de resultados**

```
...
#lendo arquivo que contém coletas e predicao

setwd("C:\\Users\\nickv\\Documents\\BCC\\TCC2\\pcs\\testes")
resultado <- read.csv("resultado.csv", sep = ";", head = T)

#####
...
```

**Fonte: Autoria própria (2021).**

Com os dados carregados, foi possível plotar o gráfico com o comportamento real do recurso computacional, porém esta etapa merece **atenção**: pelo elevado número de pontos que foram plotados, são 14641 coletas, o gráfico ficou muito carregado, porém é possível observar o comportamento do recurso durante o dia de coleta.

Na Figura 23 está o trecho de código responsável pela geração do primeiro gráfico, comportamento real, e a sua plotagem, na Figura 24:

**Figura 23 - Plotando comportamento real**

```
...
#####

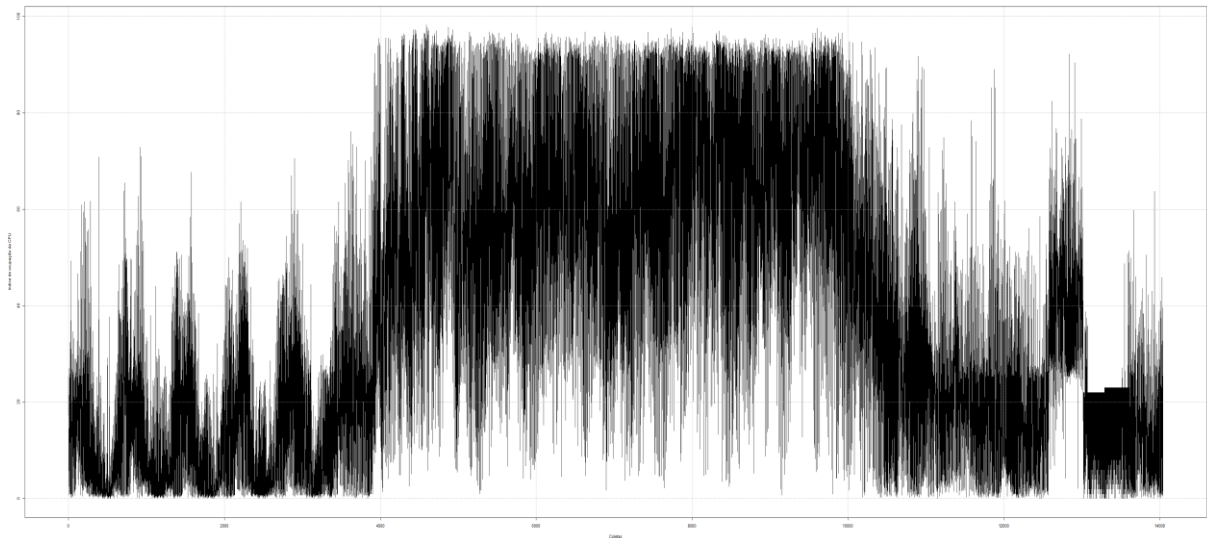
#plotando série original

dados <- ts(resultado, start=1, freq=1)

png(width = 4000, height = 1500, "sérieOriginal.png")
plot(dados[,1], main = "Série temporal histórica", ylab = "Índice
de ocupação da CPU", xlab = "Coletas")
grid(col='darkgrey', lwd=1)
dev.off()

#####
...
```

**Fonte: Autoria própria (2021).**

**Figura 24 - Dia 1: Comportamento real**

Fonte: Autoria própria (2021).

Como pode ser observado, há altos índices de uso de CPU durante uma faixa de horários, que coincidem com o horário comercial (08:00hr até 18:00hr). De posse desses dados, calculou-se a MMS no código em C, e poderá ser observado uma suavização dos dados. Para tanto o seguinte trecho em R, na Figura 25, deve ser executado para a geração do gráfico, o mesmo está na Figura 26:

**Figura 25 - Plotando predição**

```

...
#####

#plotando predição - sem eliminar sazonalidade

predicao <- ts(read.delim("resultado.csv", sep=";", head = T),
start=1, freq=1)
predicao <- predicao[,2]

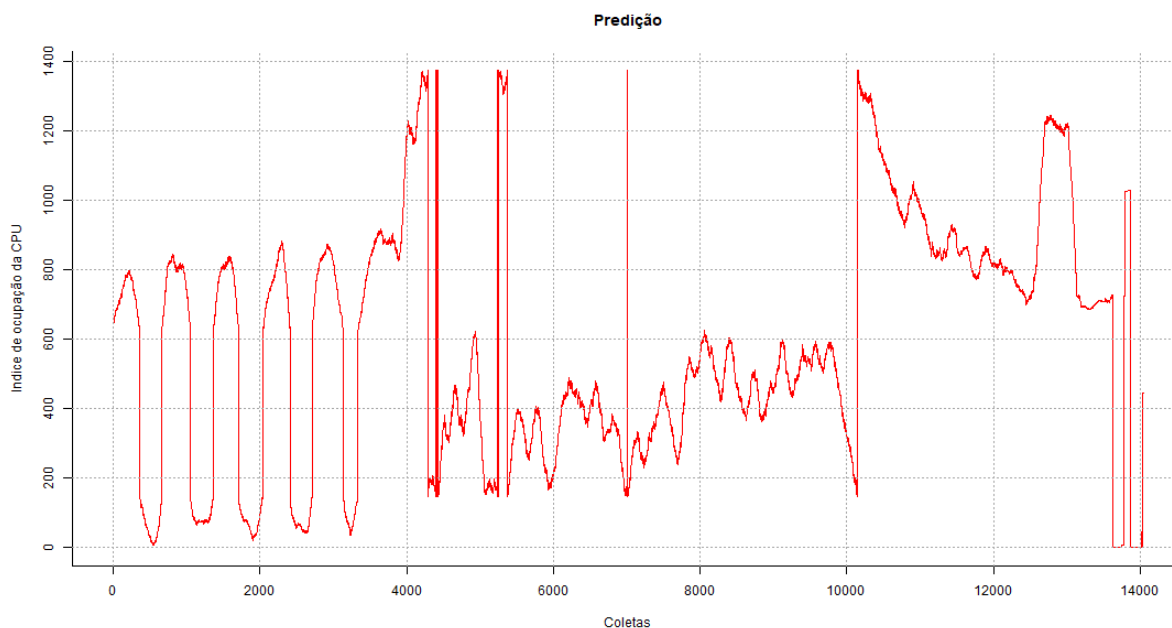
png(width = 1100, height = 600,"prediçãoI.png")
plot(predicao, main ='Predição',ylab='Índice de ocupação da CPU',
xlab = 'Coletas',bty='l',col='red',lty=1)
grid(col='darkgrey',lwd=1)
dev.off()

#####
...

```

Fonte: Autoria própria (2021).

**Figura 26 - Dia 1: Predição**



**Fonte: Autoria própria (2021).**

Neste gráfico foi possível perceber a suavização dos dados, e conseqüentemente a formação de uma **tendência** de utilização de uso do recurso computacional, ou seja, apesar de na Figura 24 haver diversos picos com percentual próximo de 100% de utilização, esse fato não se mantém constante, a MMS é capaz de suavizar tais dados e levá-los para valores próximos da média, demonstrando um comportamento mais factível de ser analisado futuramente por um SGR.

Para findar as análises da série temporal gerada, verificou-se a convergência dos valores encontrados via MMS e a eliminação de possíveis sazonalidades na predição, antes de efetivamente gerar a assinatura comportamental do equipamento. Na Figura 27 está o código que realiza as análises finais:

**Figura 27 - Verificação de sazonalidade e tendência**

```
...
#verificando possíveis sazonalidades, se sim: devem ser eliminadas
dados <- data.frame(predicao,dados[,1])
predicao.reg <- lm(predicao~dados[,1], data = dados)
summary(predicao.reg)
predicao.des <- ts(resid(predicao.reg),start = 1, freq=1)
predicao.hat <- ts(fitted(predicao.reg),start=1, freq=1)
png("componenteSazonal.png")
par(mfrow=c(1,2))
plot(main = 'Série dessazonalizada', xlab = 'Coletas', ylab =
'Índice de ocupação da CPU', predicao.des)
grid(col='darkgrey')
```

```

plot(main = 'Componente sazonal', xlab = 'Coletas', ylab = 'Indice
de ocupação da CPU', predicao.hat)
grid(col='darkgrey')
dev.off()
...

```

Fonte: Autoria própria (2021).

O trecho acima gerou algumas informações importantes para esse contexto: a apresentação da componente sazonal, se existir; e também a confirmação da convergência da série gerada. Na Figura 28 é exibido o que o comando `summary(predicao.reg)` retorna:

Figura 28 - Dia 1: Análises

```

...
Call: lm(formula = predicao ~ dados[, 1], data = dados)

Residuals:
Min       1Q   Median       3Q      Max
-2.270e-12 -2.800e-14 -1.900e-14 -7.000e-15  2.411e-10

Coefficients:
            Estimate Std. Error  t value Pr(>|t|)
(Intercept) -3.438e-12  3.394e-14 -1.013e+02  <2e-16 ***
dados[, 1]   1.000e+00  4.986e-17  2.005e+16  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.035e-12 on 14040 degrees of freedom
Multiple R-squared: 1, Adjusted R-squared: 1
F-statistic: 4.022e+32 on 1 and 14040 DF, p-value: < 2.2e-16
...

```

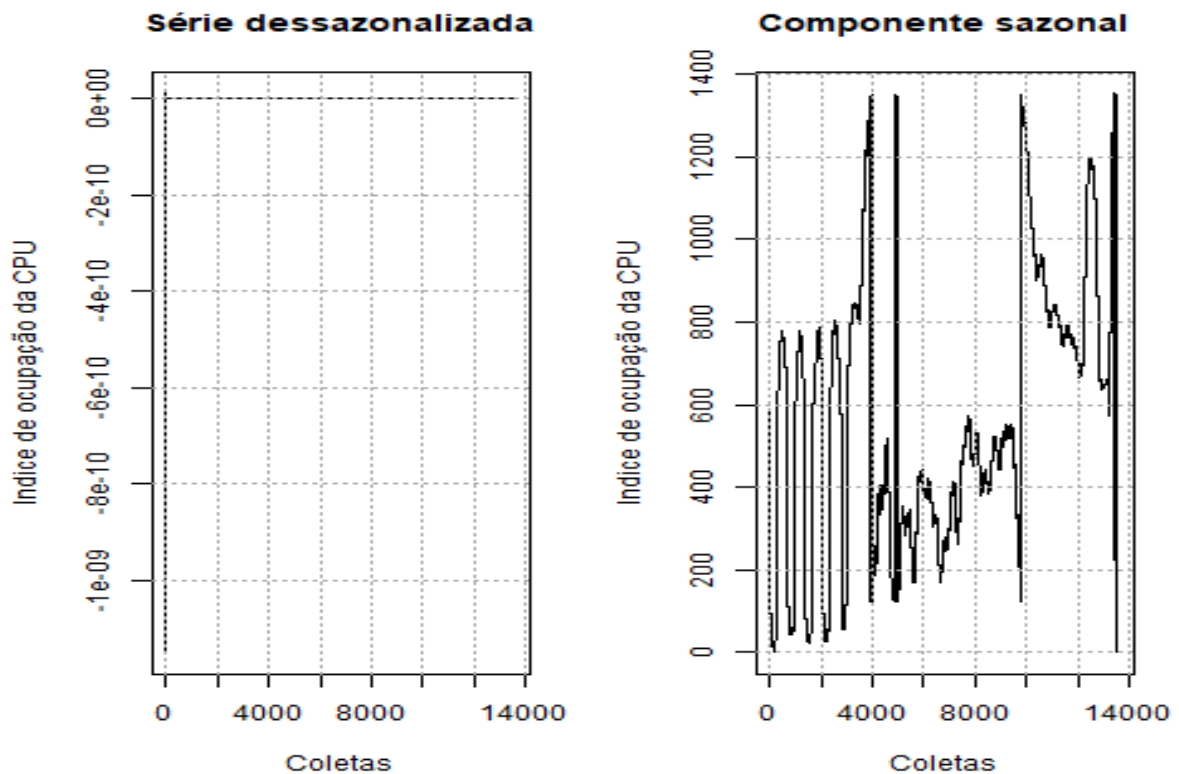
Fonte: Autoria própria (2021).

De todos os dados informados, o  $p\text{-value} < 2.2e-16$  reportou a margem de erro para verificar sazonalidades e também a convergência dos dados, visto que o comando `predicao.reg <- lm(predicao~dados[,1], data = dados)` compara os dados previstos com os dados calculados via MMS. No caso da verificação de convergência, este valor praticamente nulo indica convergência, segundo Morettim e Tolo (2006).

Já para a verificação de sazonalidade estocástica, segundo Morettim e Tolo (2006), é quando a componente sazonal da série varia com o tempo, se o teste executado no código elaborado em R retorna este valor próximo de zero, significa que esta componente sazonal pode existir, porém não é relevante para o contexto, como pode ser observado na Figura 29:



Figura 29 - Dia 1: Comparativo de séries



Fonte: Autoria própria (2021).

E com este gráfico ficou claro que a componente sazonal desta série é irrelevante, pois ao remover a possível sazonalidade, toda a série é excluída, segundo Morettim e Tolo (2006), se este resultado acontecer, a série temporal não possui componente sazonal relevante.

Por fim, agora foi possível definir a assinatura comportamental do equipamento no primeiro dia de coletas, Na Figura 30 é demonstrado o trecho do código em R desenvolvido para este trabalho que é responsável pela plotagem final:

Figura 30 - Plotando assinatura comportamental

```
...
predicao.desn <- predicao.des + mean(fitted(predicao.reg))
png(width = 1100, height = 600, "assinaturaComport.png")
par(mfrow=c(1,1))
plot(predicao,main = 'Comparativo de séries temporais'
,xlab='Coletas', ylab='Índice de ocupação da CPU',col='blue',bty='l')

par(new=TRUE)

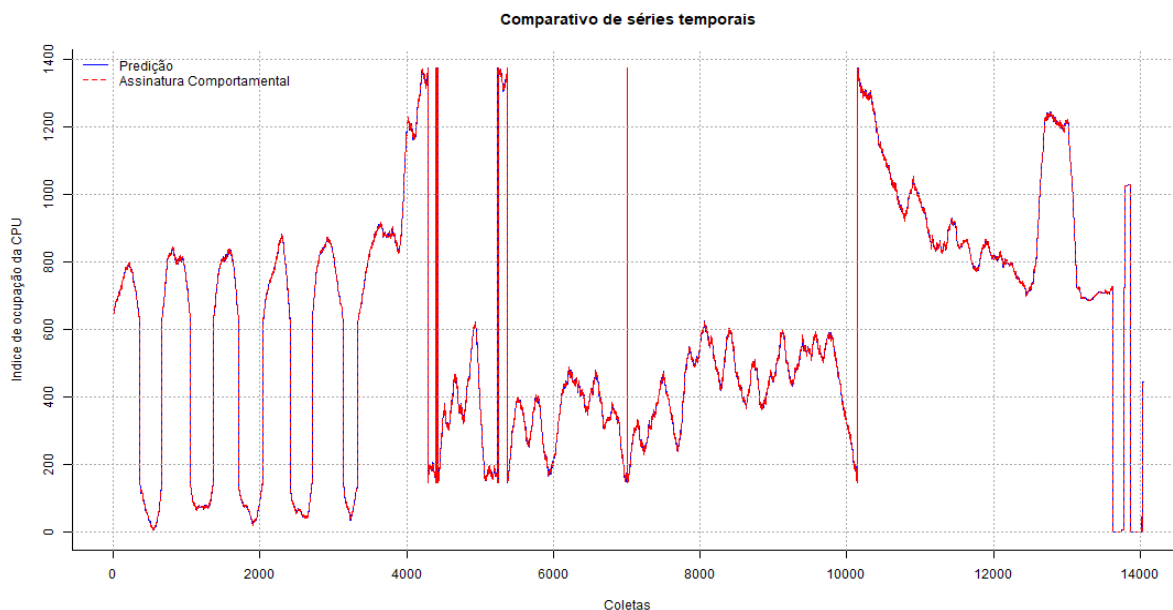
plot(predicao.desn,axes=F, ann=F,col='red',lty=2)
legend('topleft',c('Predição', 'Assinatura
Comportamental'),col=c('blue', 'red'), lty=1:2,bty='n')
grid(col='darkgrey')
```

```
dev.off()
...
```

Fonte: Autoria própria (2021).

O código acima gera o seguinte gráfico, conforme Figura 31:

Figura 31 - Dia 1: Assinatura comportamental

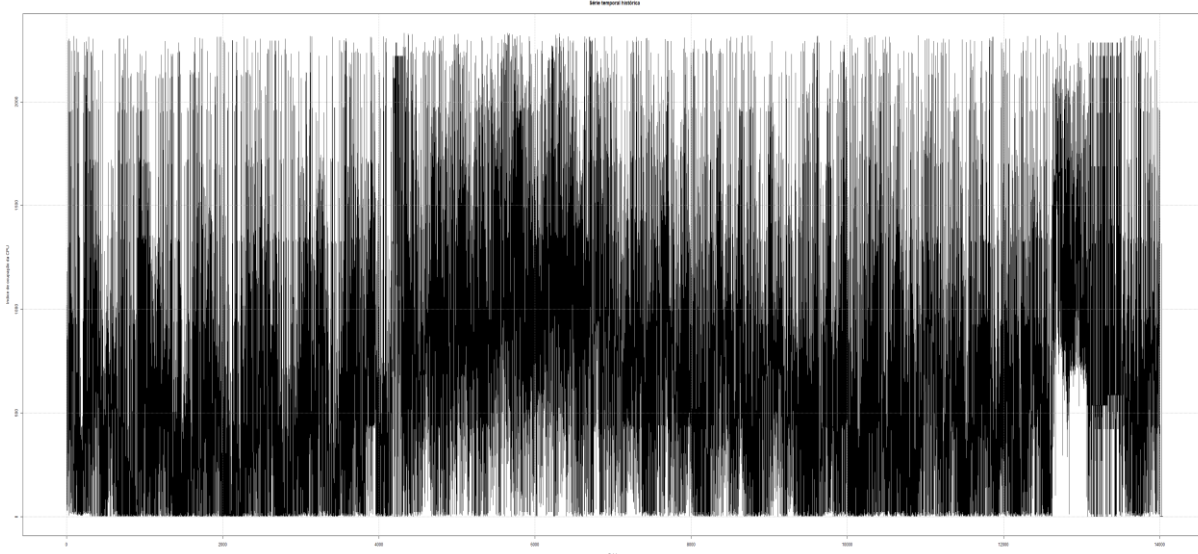


Fonte: Autoria própria (2021).

Pode-se perceber que a predição realizada pela MMS no código em C coincide com a assinatura comportamental gerada após análises no código em R, por uma questão simples e já debatida anteriormente: não há componente sazonal estocástica, portanto, a assinatura comportamental é igual a predição realizada.

Para os demais dias e máquinas, não foram exibidos os trechos de códigos e nem tabelas com os dados, justifica-se pois o procedimento é o mesmo para todo o conjunto; com relação as tabelas, exibiu-se somente para demonstrar como estão organizadas, já que sua visualização na íntegra neste trabalho é inviável, pois se tratava de um arquivo com 14641 linhas.

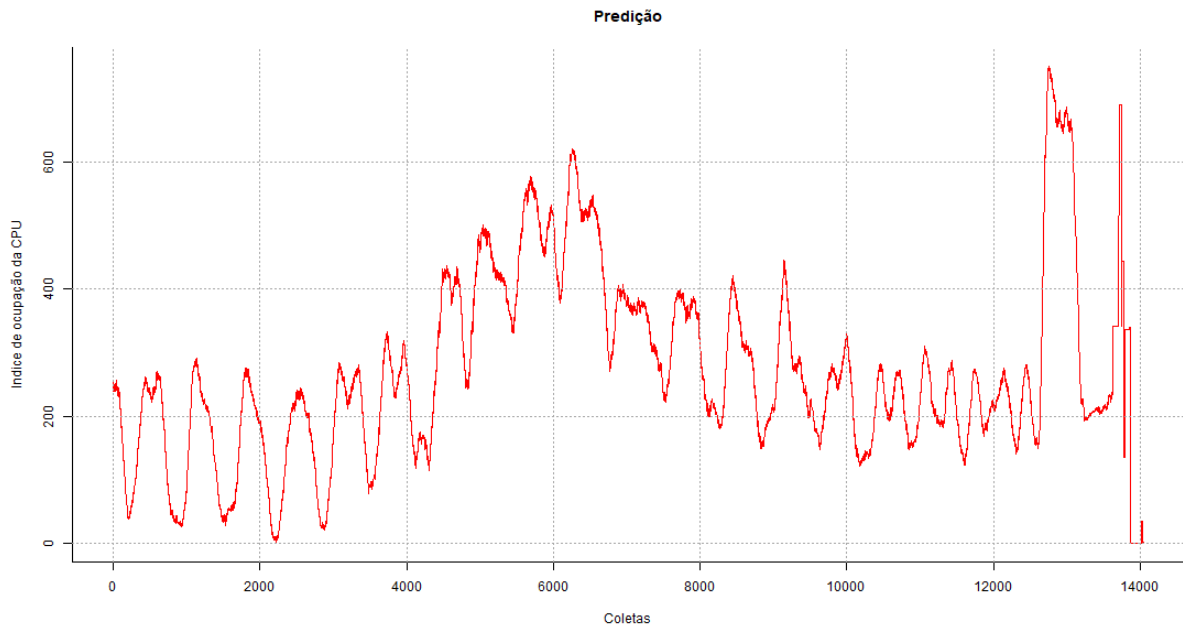
Agora serão exibidos os resultados encontrados considerando o **segundo dia** de coletas na máquina 1 deste sistema distribuído, observe na Figura 32 o comportamento real encontrado:

**Figura 32 - Dia 2: Comportamento real**

**Fonte: Autoria própria (2021).**

Mais uma vez ressalta-se que o gráfico do comportamento real da máquina está com a visualização carregada devido a quantidade de coletas, são mais de 14000. Percebeu-se que neste dia, houve uma variação muito grande do percentual de uso de CPU, porém com tendência ao centro, como pode ser observada na área mais escurecida do gráfico.

Com esses dados em mãos, foi possível ver a predição calculada e qual o seu comportamento, conforme a Figura 33:

**Figura 33 - Dia 2: Predição**

**Fonte: Autoria própria (2021).**

Foi possível verificar por meio deste gráfico que a MMS suavizou os dados para exatamente onde se concentravam a maioria das coletas, assim cumprindo com seu objetivo de indicar uma tendência dos dados. Executando-se as análises de convergência e sazonalidade, obteve-se o seguinte resultado, contido na Figura 34:

**Figura 34 - Dia 2: Análises**

```

...
Call:
lm(formula = predicacao ~ dados[, 1], data = dados)

Residuals:
    Min       1Q   Median       3Q      Max
-5.100e-12 -5.000e-14 -4.000e-14 -4.000e-14  6.037e-10

Coefficients:
            Estimate Std. Error  t value Pr(>|t|)
(Intercept)  9.517e-12  9.135e-14  1.042e+02 <2e-16 ***
dados[, 1]   1.000e+00  2.957e-16  3.382e+15 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

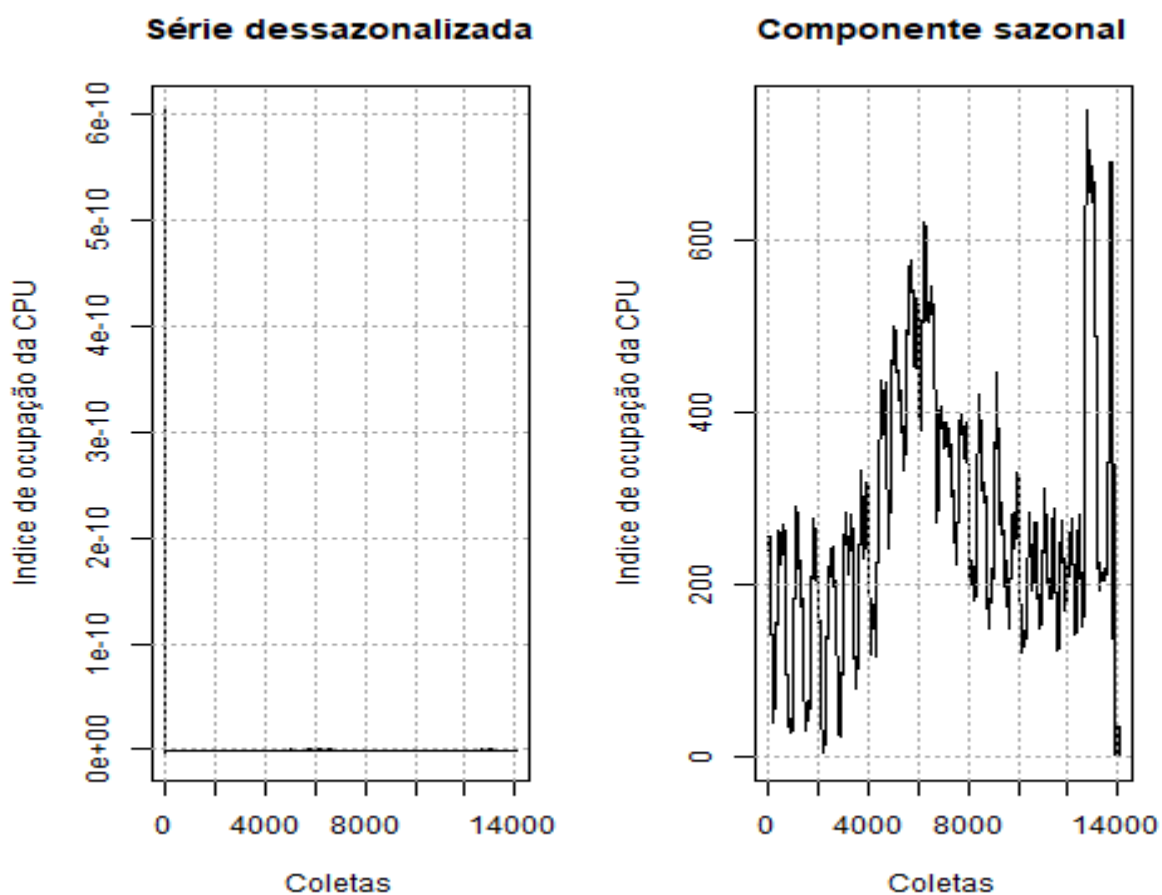
Residual standard error: 5.095e-12 on 14040 degrees of freedom
Multiple R-squared: 1, Adjusted R-squared: 1
F-statistic: 1.144e+31 on 1 and 14040 DF, p-value: < 2.2e-16
...

```

**Fonte: Autoria própria (2021).**

Seguindo a mesma lógica do dia anterior, não houve sazonalidade estocástica e houve convergência dos dados, o que pode ser visto no comparativo de séries contido na Figura 35:

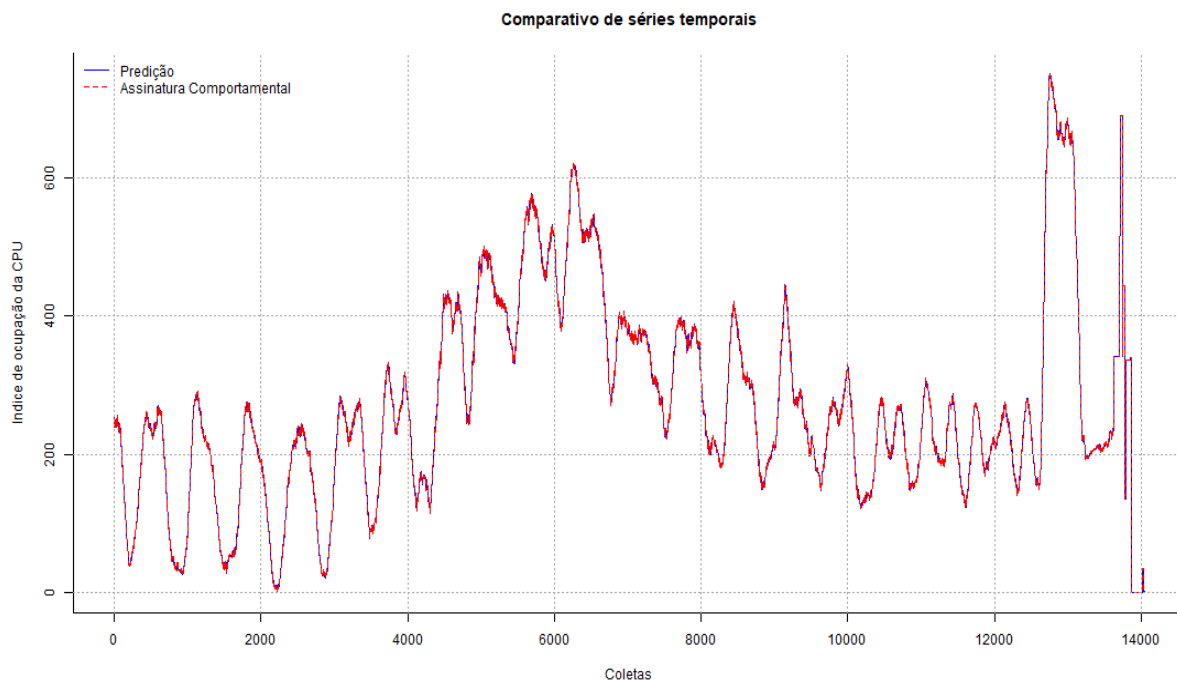
Figura 35 - Dia 2: Comparativo de séries



Fonte: Autoria própria (2021).

Como não houve componente sazonal e a predição calculada via MMS é equivalente a assinatura comportamental do equipamento, o resultado final deste dia é o que foi constatado na Figura 36:

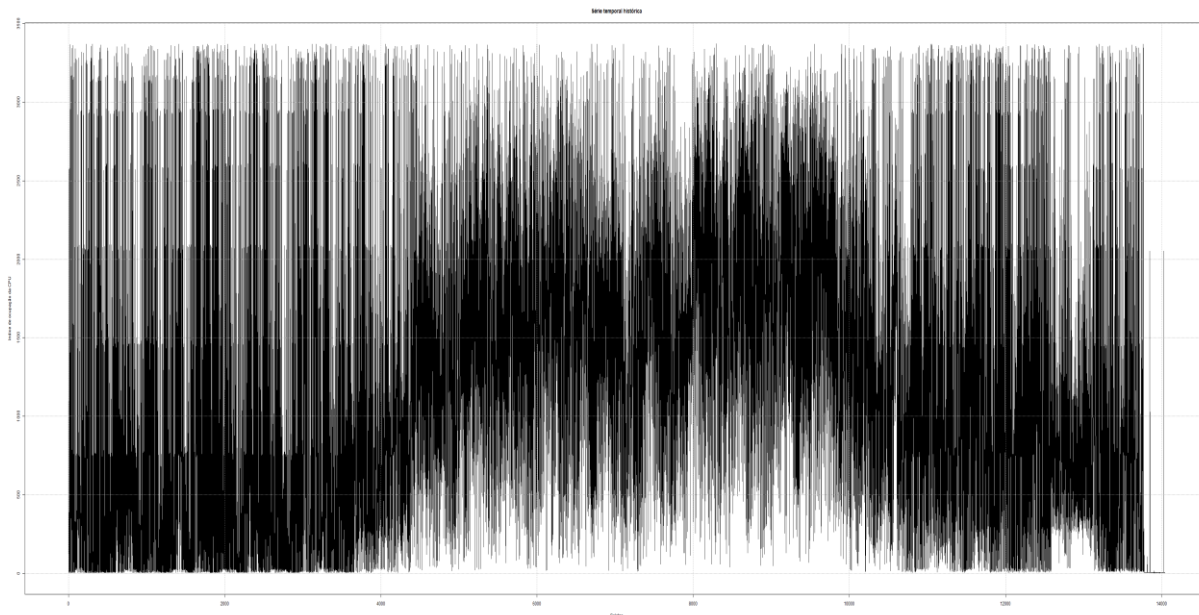
Figura 36 - Dia 2: Assinatura comportamental



Fonte: Autoria própria (2021).

Dessa forma definiu-se a assinatura comportamental do segundo dia na máquina 1. Agora será mostrado os resultados do **terceiro dia** desta mesma máquina, começando pelo seu comportamento real observado, contido na Figura 37:

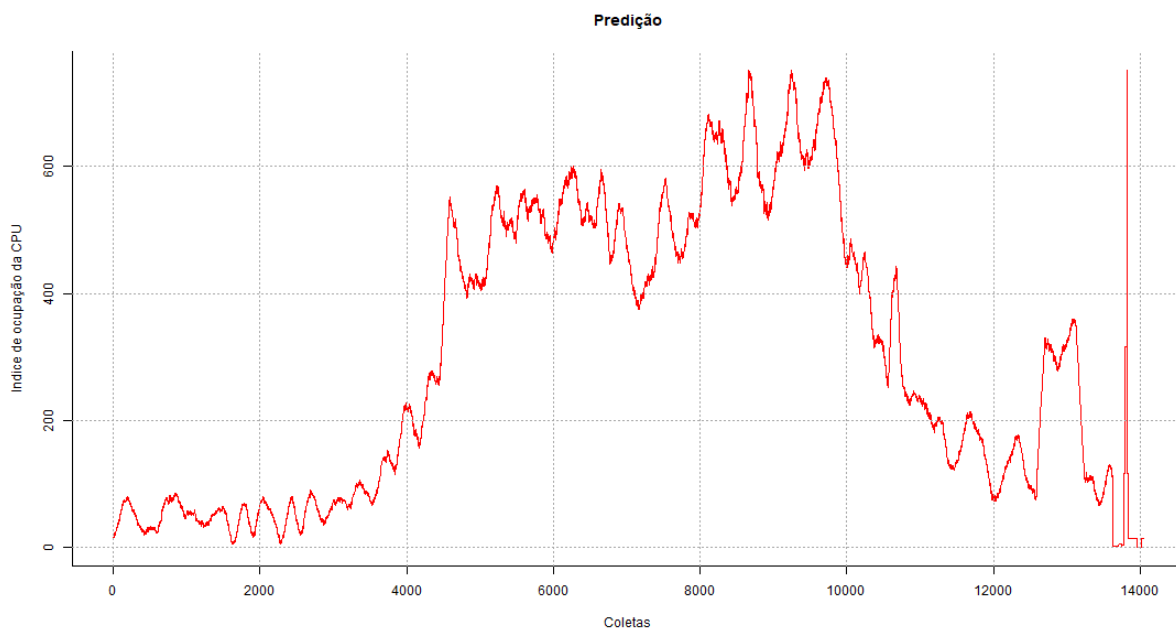
Figura 37 - Dia 3: Comportamento real



Fonte: Autoria própria (2021).

Pode-se observar neste terceiro dia de coletas que esta máquina experimentou uma *range* amplo de utilizações, com um índice mais elevado e constante durante o horário comercial (08hr até 18hr). Na Figura 38 está presente a suavização dos dados realizada via MMS:

**Figura 38 - Dia 3: Predição**



**Fonte: Autoria própria (2021).**

Com a predição realizada, foi possível analisá-la afim de se verificar convergência e sazonalidade, o que pode ser constatado na Figura 39:

**Figura 39 - Dia 3: Análises**

```

...
Call:
lm(formula = predicao ~ dados[, 1], data = dados)

Residuals:
    Min       1Q   Median       3Q      Max
-6.958e-11 -4.000e-15  5.000e-15  1.400e-14  6.640e-13

Coefficients:
            Estimate Std. Error  t value Pr(>|t|)
(Intercept) -7.982e-13  8.283e-15 -9.637e+01 <2e-16 ***
dados[, 1]   1.000e+00  2.219e-17  4.507e+16 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.095e-12 on 14040 degrees of freedom
Multiple R-squared: 1, Adjusted R-squared: 1
F-statistic: 1.144e+31 on 1 and 14040 DF, p-value: < 2.2e-16

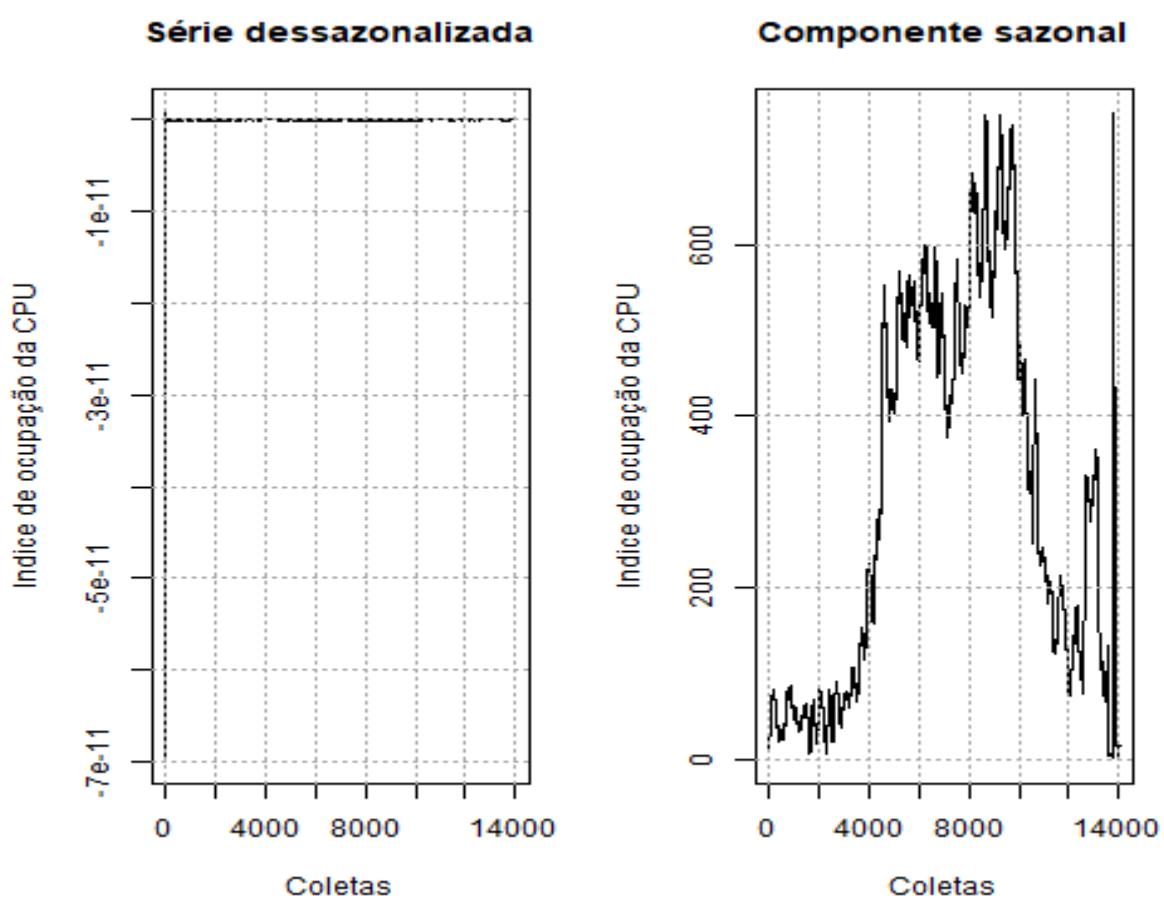
```

...

Fonte: Autoria própria (2021).

Do mesmo modo, não houve sazonalidade estocástica e houve convergência. E pelo terceiro dia seguido tais eventos foram observados, no fim dos experimentos foi discutido o motivo de tal fato estar ocorrendo. Na Figura 40 está o gráfico que certifica a inexistência de componente sazonal:

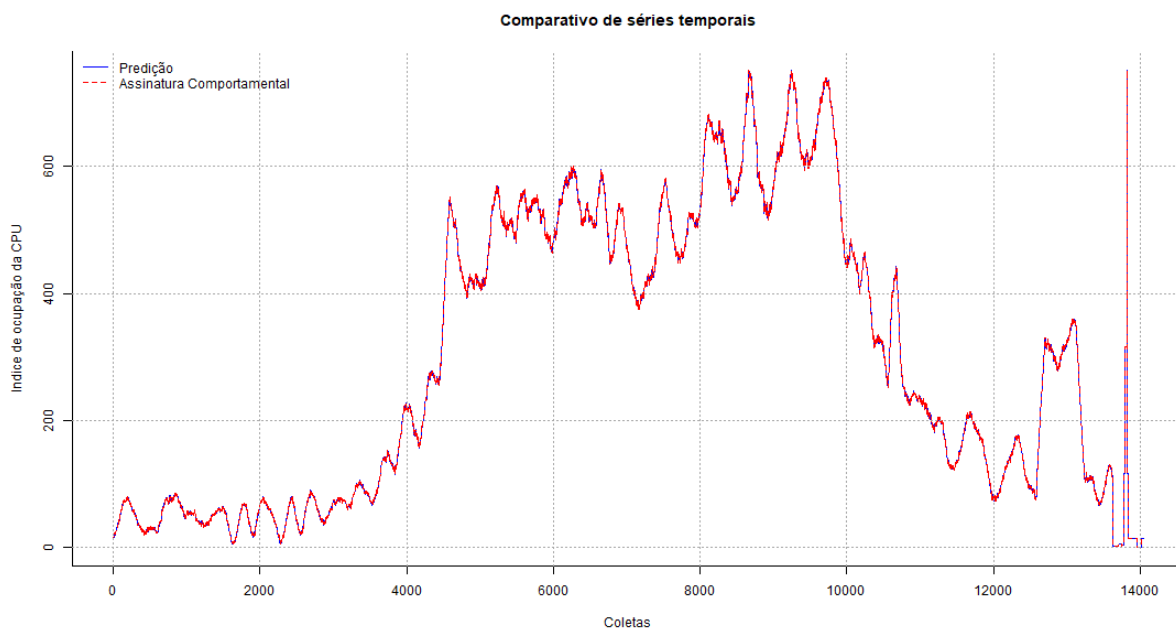
Figura 40 - Dia 3: Comparativo de séries



Fonte: Autoria própria (2021).

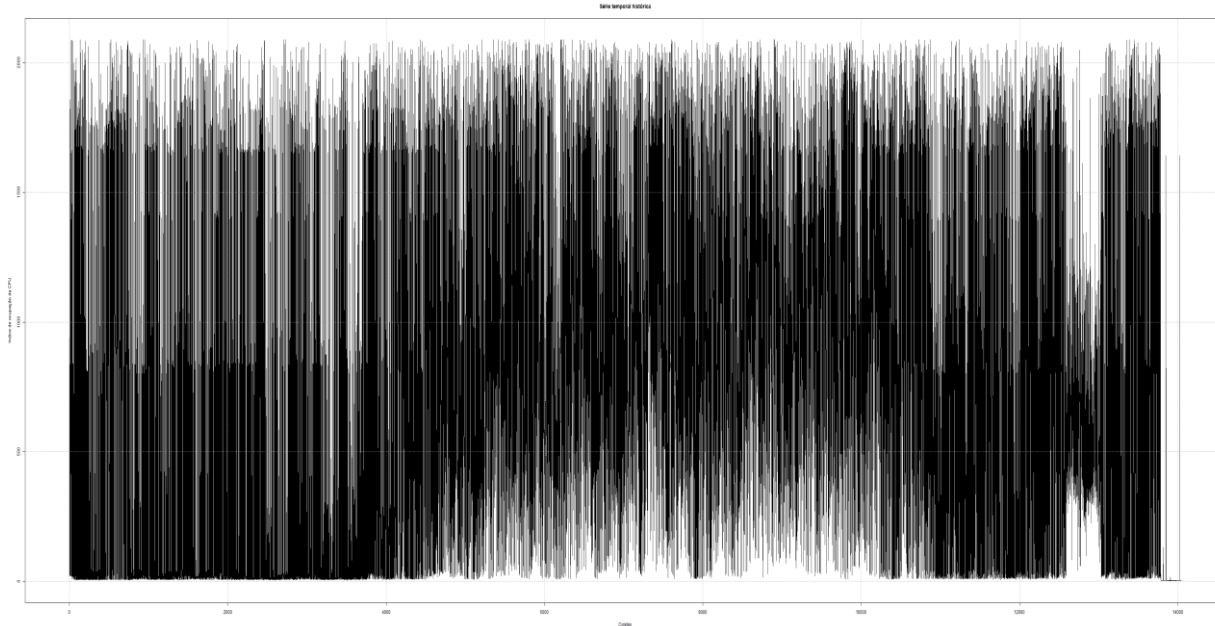
E por fim, na Figura 41, a assinatura comportamental da máquina 1 no terceiro dia foi a seguinte:



**Figura 41 - Dia 3: Assinatura comportamental**

Fonte: Autoria própria (2021).

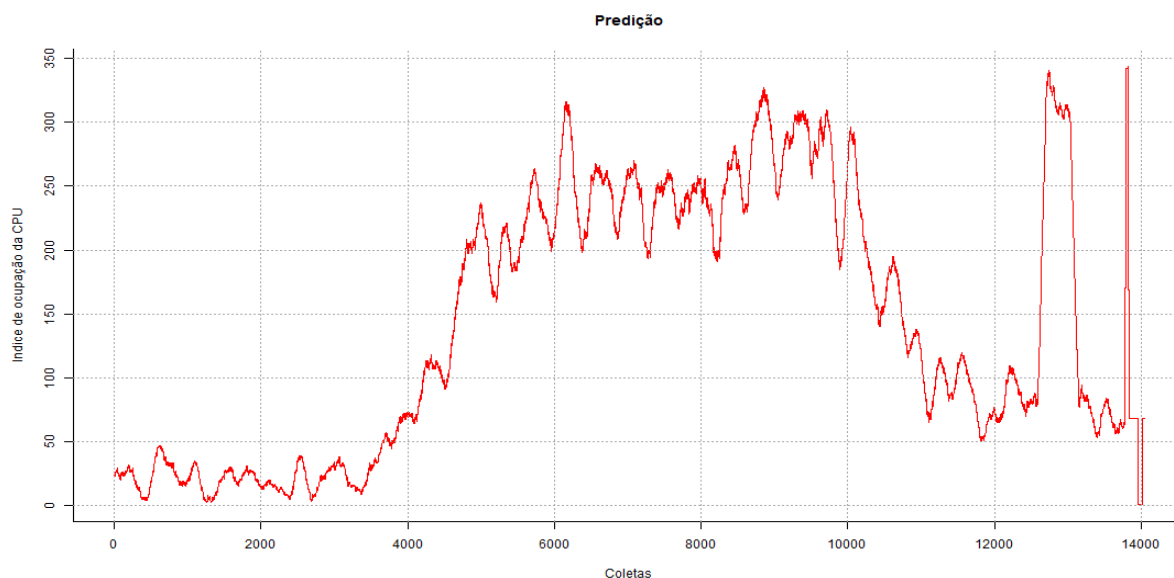
Como já visto nos demais dias, a assinatura comportamental convergiu com a previsão realizada. Por seqüência, serão vistos os resultados do **quarto dia** de coletas da máquina 1, inicializando pelo comportamento real capturado, localizado na Figura 42:

**Figura 42 - Dia 4: Comportamento real**

Fonte: Autoria própria (2021).

Para este dia, a máquina 1 apresentou muitos picos de ociosidade, visto que houve uma prevalência de uso da CPU em percentuais próximos de zero. A seguir, na Figura 43, os dados previstos utilizando MMS:

**Figura 43 - Dia 4: Predição**



**Fonte: Autoria própria (2021).**

Após a suavização realizada, foi visível que este método conseguiu encontrar uma **tendência** para os dados, informação relevante quando se pensa na construção de um Sistema de Gerenciamento de Recursos (SGR). Na Figura 44 estão as análises de convergência e sazonalidade:

**Figura 44 - Dia 4: Análises**

```

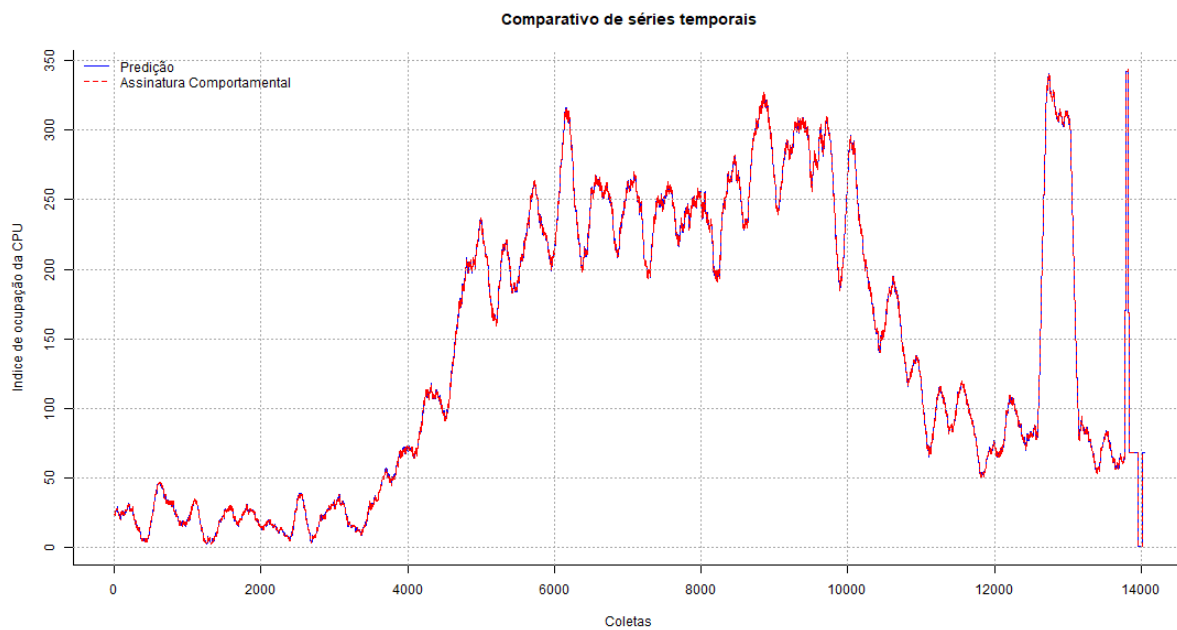
...
Call:
lm(formula = predicao ~ dados[, 1], data = dados)
Residuals:
    Min       1Q   Median       3Q      Max
-9.357e-11  1.000e-15  7.000e-15  1.300e-14  8.140e-13
Coefficients:
      Estimate Std. Error  t value Pr(>|t|)
(Intercept) -1.075e-12  1.142e-14 -9.408e+01 <2e-16 ***
dados[, 1]   1.000e+00  6.510e-17  1.536e+16 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 7.898e-13 on 14040 degrees of freedom
Multiple R-squared:  1,    Adjusted R-squared:  1
F-statistic: 2.36e+32 on 1 and 14040 DF,  p-value: < 2.2e-16
...

```

**Fonte: Autoria própria (2021).**

Como houve convergência e não houve sazonalidade estocástica, a assinatura comportamental no quarto dia será equivalente a predição já realizada, conforme é demonstrado na Figura 45:

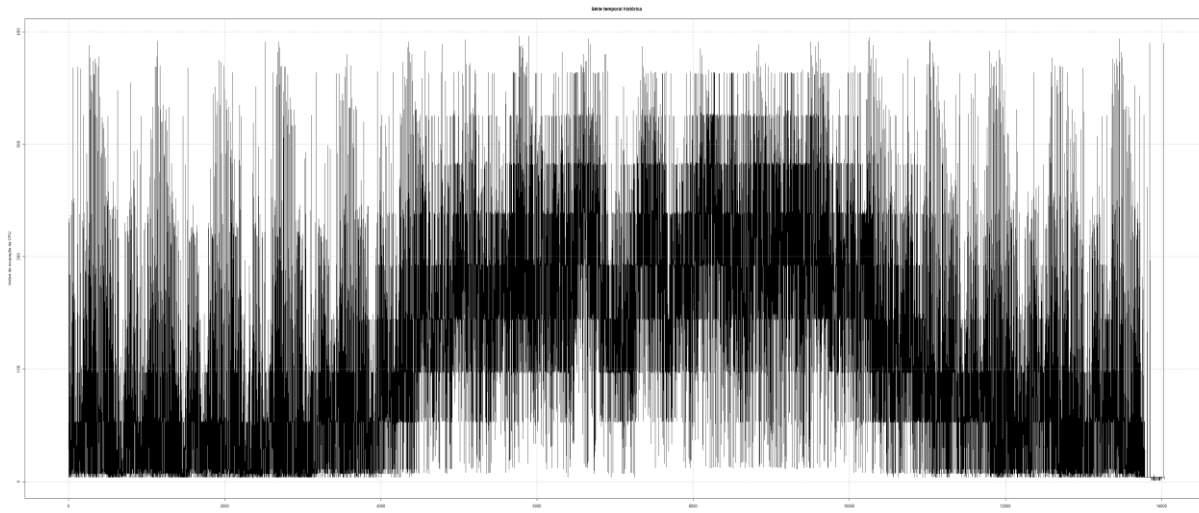
**Figura 45 - Dia 4: Assinatura comportamental**



**Fonte: Autoria própria (2021).**

E finalizando as observações nesta máquina, abaixo estão os resultados de coleta e predição no **quinto dia**, inicializando-se as demonstrações pelo gráfico de comportamento real do equipamento, demonstrado na Figura 46:

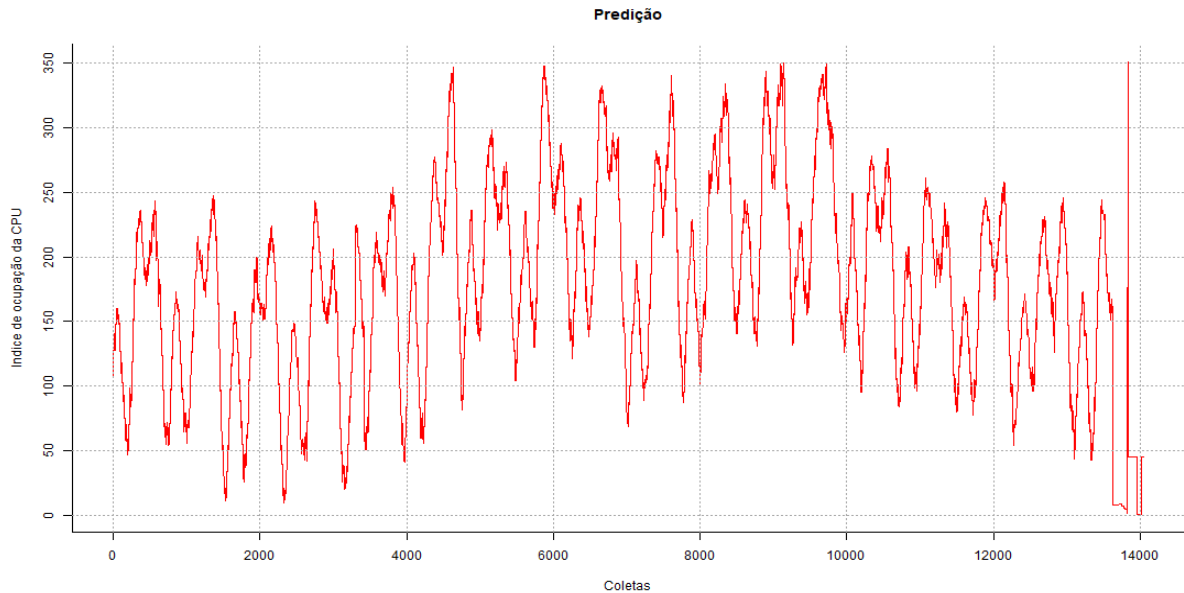
**Figura 46 - Dia 5: Comportamento real**



**Fonte: Autoria própria (2021).**

Neste dia de observações da máquina 1, houve uma predominância de observações abaixo de 50% de utilização da CPU, com um acréscimo de uso durante o horário comercial. A seguir, na Figura 47, a predição destes dados usando MMS:

**Figura 47 - Dia 5: Predição**



**Fonte: Autoria própria (2021).**

Para este dia de observações, não houve grandes concentrações de percentuais de uso de CPU, a MMS não realizou uma suavização que tende a linearidade, porém, ainda apontando uma **tendência**, contudo com um *range* de utilização maior se comparado com os demais dias. Na Figura 48, os testes de convergência e sazonalidade:

**Figura 48 - Dia 5: Análises**

```

...
Call:
lm(formula = predicacao ~ dados[, 1], data = dados)
Residuals:
    Min       1Q   Median       3Q      Max
-1.871e-12 -1.380e-15  2.700e-16  2.050e-15  3.245e-14
Coefficients:
      Estimate Std. Error  t value Pr(>|t|)
(Intercept) -3.070e-14  3.543e-16 -8.666e+01 <2e-16 ***
dados[, 1]   1.000e+00  1.828e-18  5.470e+17 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 1.645e-14 on 14040 degrees of freedom
Multiple R-squared: 1, Adjusted R-squared: 1
F-statistic: 2.992e+35 on 1 and 14040 DF, p-value: < 2.2e-16

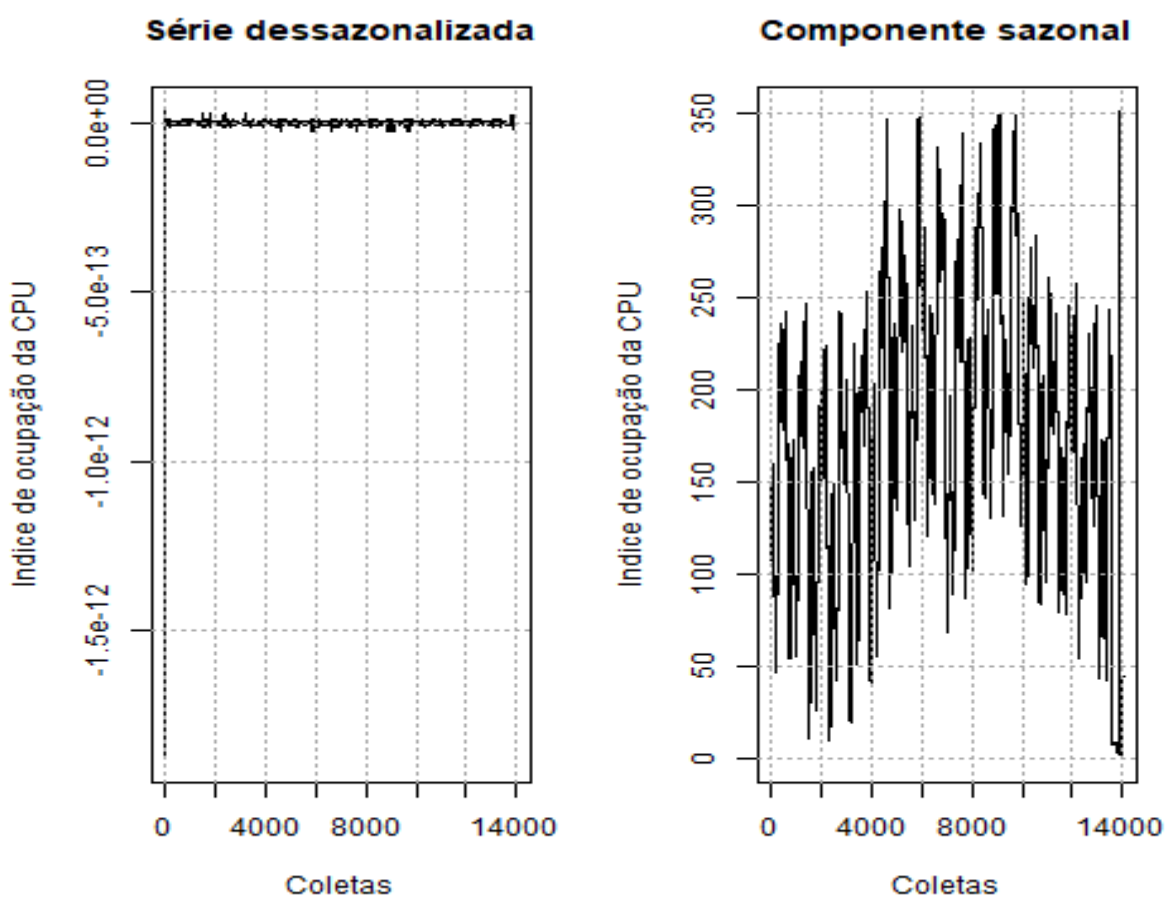
```

...

Fonte: Autoria própria (2021).

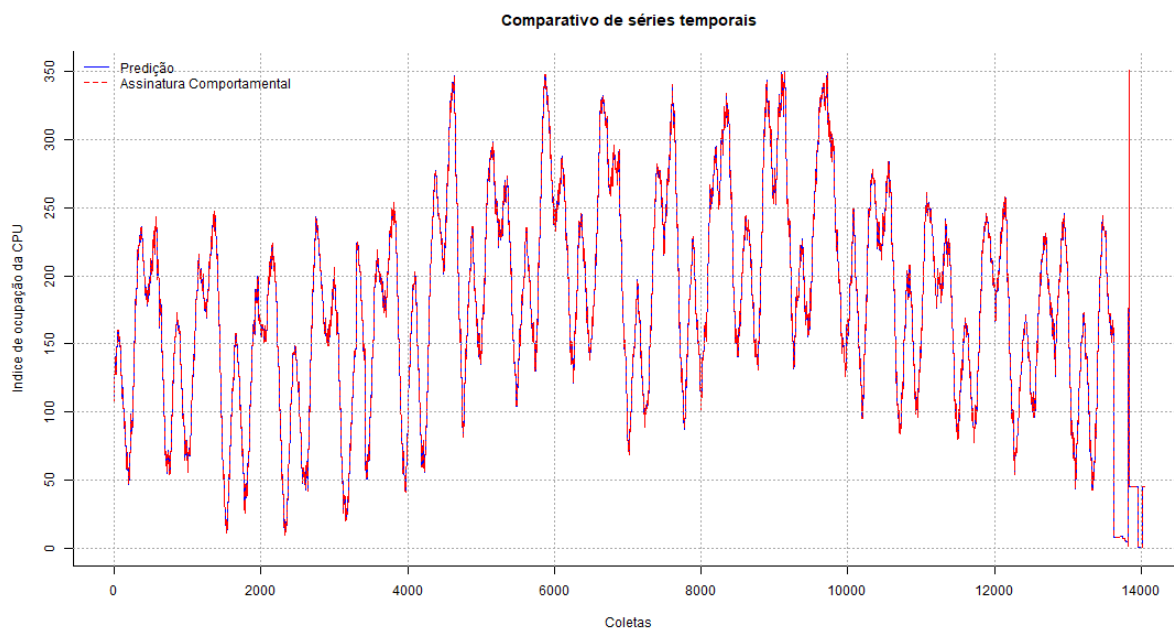
E como nos outros dias, houve convergência e não houve sazonalidade estocástica; na Figura 49, é demonstrado que a componente sazonal é igual a predição:

Figura 49 - Dia 5: Comparativo de séries



Fonte: Autoria própria (2021).

Nessa situação, onde a predição e assinatura comportamental se equivalem, é possível plotar o gráfico contendo o mesmo, como é apresentado na Figura 50:

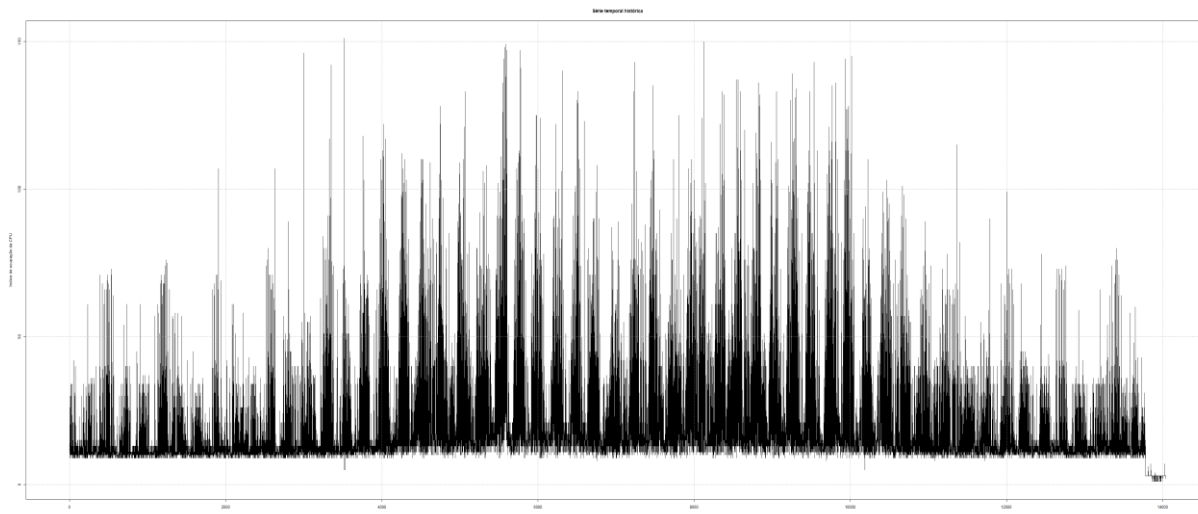
**Figura 50 - Dia 5: Assinatura comportamental**

**Fonte: Autoria própria (2021).**

Desse modo, findam-se as análises do comportamento de percentual de uso de CPU desta máquina, o próximo tópico contém o estudo e resultados encontrados com a máquina 2.

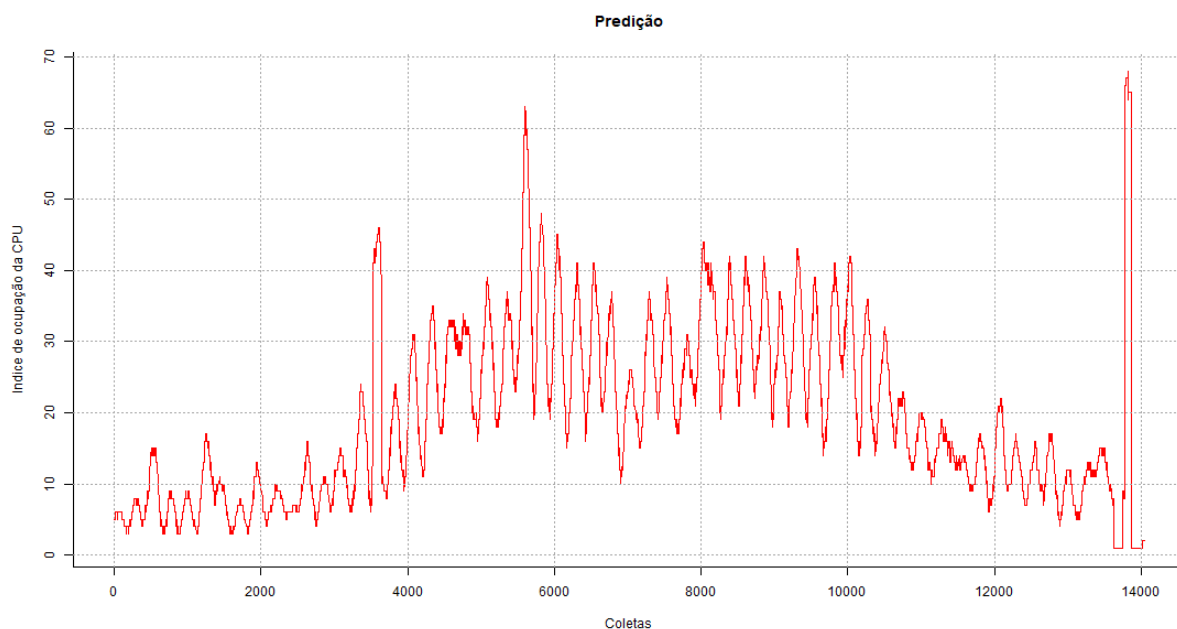
### 5.3 MICROCOMPUTADOR 2: RESULTADOS

Continuando com a apresentação dos resultados encontrados, neste tópico estão os dados coletados diariamente, pelo período de 5 dias corridos; como já consignado anteriormente, a máquina em questão é igual à que foi apresentada no tópico anterior. Iniciando pelo **primeiro dia** de coletas, na Figura 51, o comportamento observado:

**Figura 51 - Dia 1: Comportamento real**

Fonte: Autoria própria (2021).

Para este dia de coletas, a máquina passou um grande período de ociosidade, ou seja, executando tarefas irrelevantes para o sistema distribuído. Seguindo o *script*, deve-se gerar a predição dos dados coletados, obtendo-se o resultado da Figura 52:

**Figura 52 - Dia 1: Predição**

Fonte: Autoria própria (2021).

Com o gráfico gerado, foi possível observar que houve uma **tendência** de os dados estarem no meio da tabela, sugerindo que durante este dia não há uso intenso desta máquina. Na Figura 53, testes de convergência e sazonalidade estocástica:

**Figura 53 - Dia 1: Análises**

```

...
Call:
lm(formula = predicacao ~ dados[, 1], data = dados)

Residuals:
      Min       1Q   Median       3Q      Max
-8.558e-11  1.000e-15  7.000e-15  1.200e-14  7.070e-13

Coefficients:
      Estimate Std. Error  t value Pr(>|t|)
(Intercept) -1.182e-12  1.160e-14 -1.019e+02 <2e-16 ***
dados[, 1]   1.000e+00  5.041e-16  1.984e+15 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 7.223e-13 on 14040 degrees of freedom
Multiple R-squared: 1, Adjusted R-squared: 1
F-statistic: 3.936e+30 on 1 and 14040 DF, p-value: < 2.2e-16
...

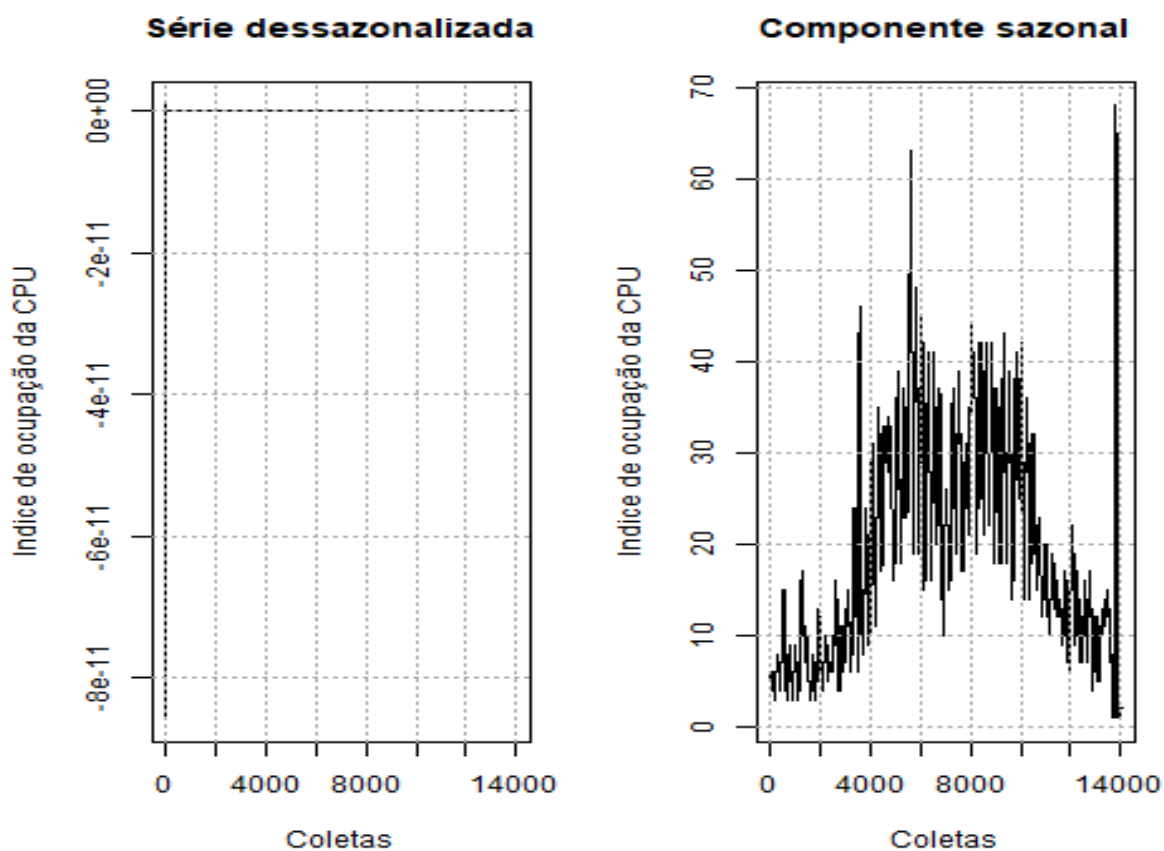
```

**Fonte: Autoria própria (2021).**

Como nas observações realizadas na outra máquina, houve convergência e não houve sazonalidade estocástica, portanto, a componente sazonal é nula e equivale a própria série temporal gerada, conforme Figura 54:

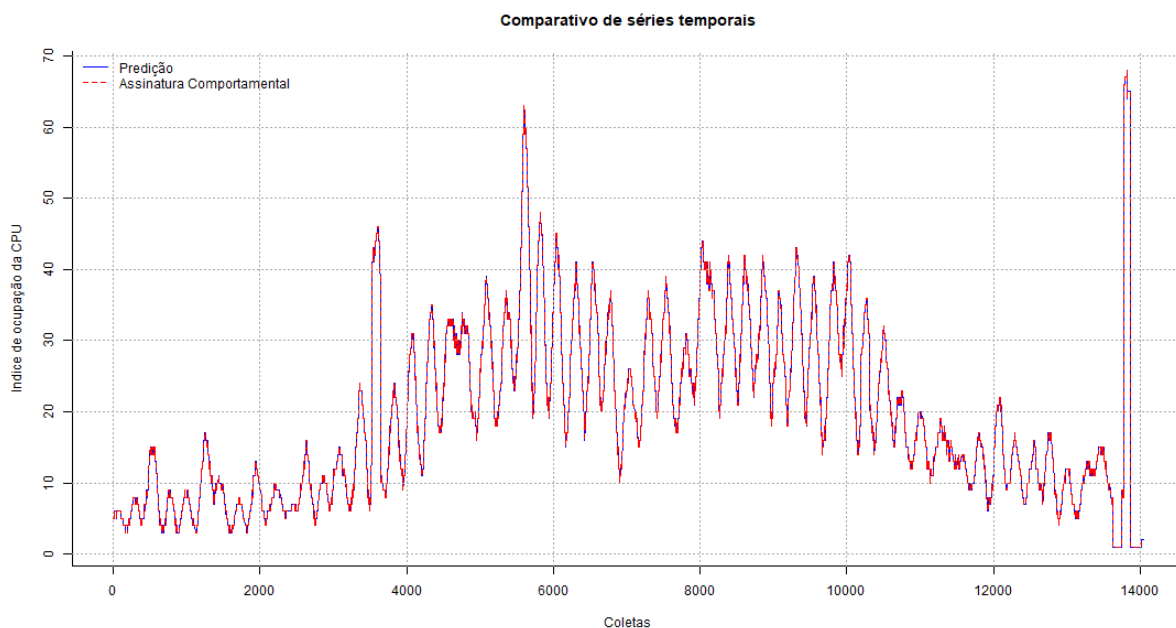


Figura 54 - Dia 1: Comparativo de séries



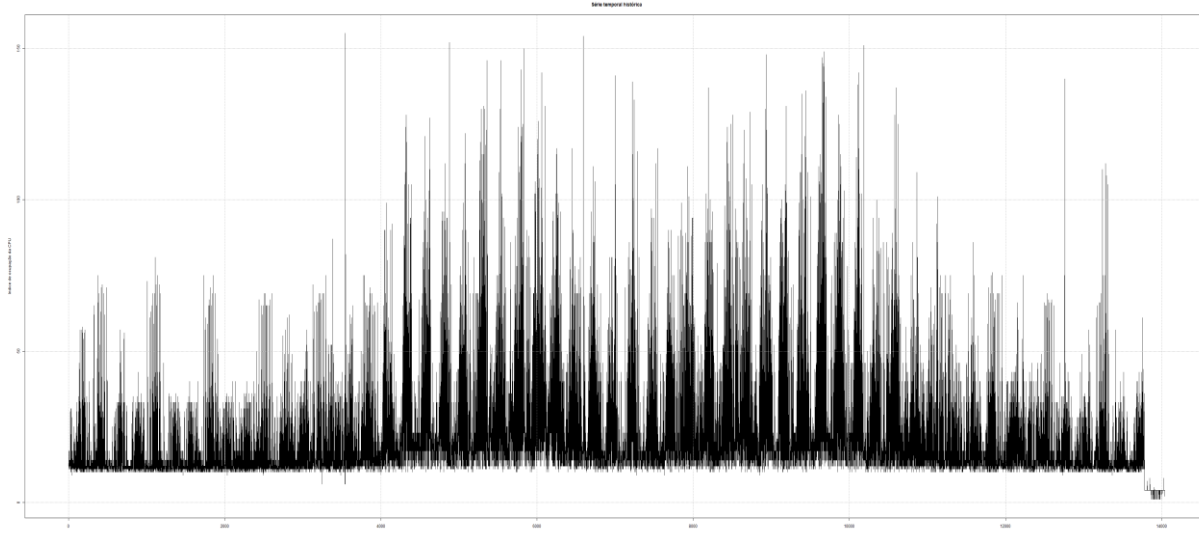
Fonte: Autoria própria (2021).

Portanto a predição realizada foi equivalente a assinatura comportamental do equipamento neste dia, conforme é exibido na Figura 55:

**Figura 55 - Dia 1: Assinatura comportamental**

Fonte: Autoria própria (2021).

A partir deste momento, serão apresentados os resultados decorrentes da captura do percentual de uso da CPU no **segundo dia**, na Figura 56 é apresentado o comportamento real do equipamento:

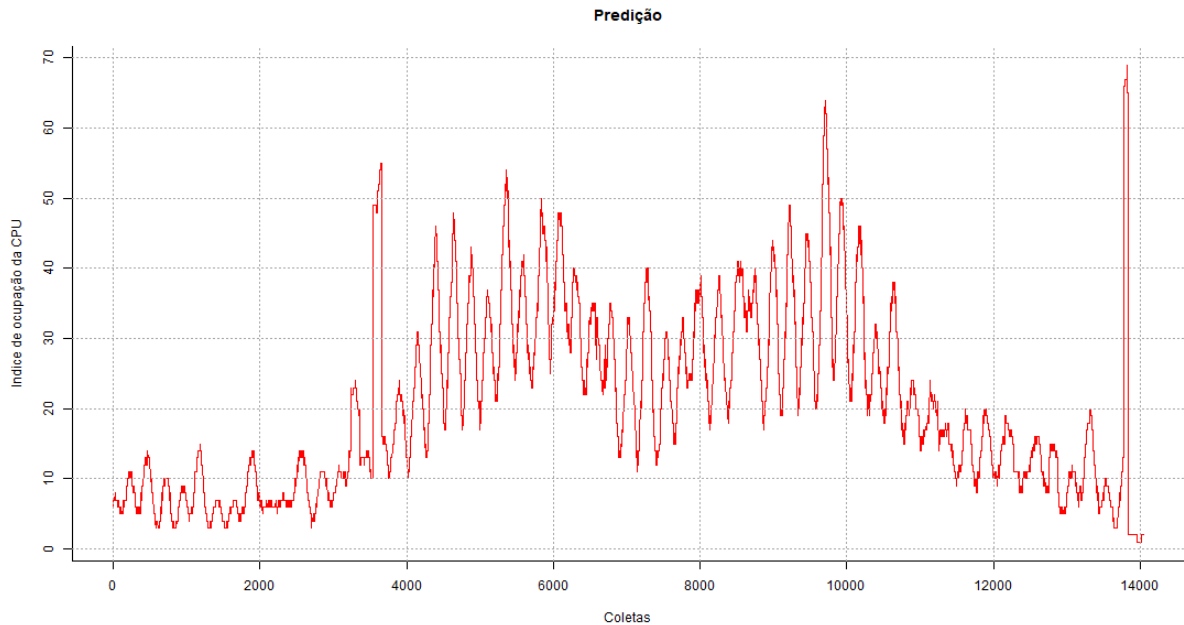
**Figura 56 - Dia 2: Comportamento real**

Fonte: Autoria própria (2021).

Neste dia, ocorreu um fato que não foi observado no outro equipamento: dois dias seguidos com observações semelhantes. Tal ocorrência pode ser benéfica no

futuro, para fins de transmissão da assinatura comportamental para o SGR, eliminando dados muito próximos/iguais. Na Figura 57, predição realizada via MMS:

**Figura 57 - Dia 2: Predição**



**Fonte: Autoria própria (2021).**

Novamente, o fato de os dados coletados serem próximos dos coletados no dia anterior, certamente deixariam a predição com forma e **tendência** muito próximos. Na Figura 58, estão os testes de convergência e sazonalidade estocástica:

**Figura 58 - Dia 2: Análises**

```

...
Call:
lm(formula = predicao ~ dados[, 1], data = dados)

Residuals:
    Min       1Q   Median       3Q      Max
-5.296e-13 -4.500e-16  2.300e-16  4.200e-16  3.694e-14

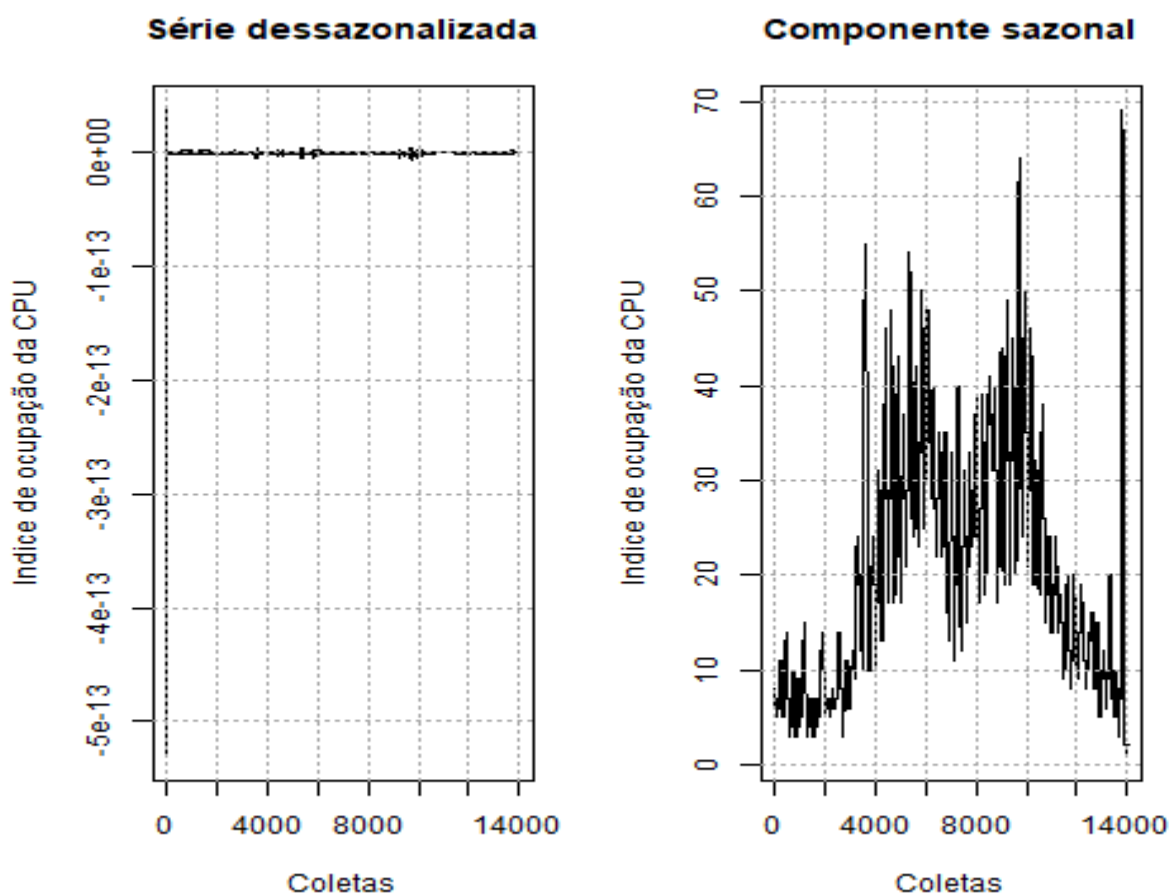
Coefficients:
      Estimate Std. Error  t value Pr(>|t|)
(Intercept) -3.838e-15  7.253e-17 -5.291e+01 <2e-16 ***
dados[, 1]   1.000e+00  2.965e-18  3.373e+17 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 4.556e-15 on 14040 degrees of freedom
Multiple R-squared:  1,    Adjusted R-squared:  1
F-statistic: 1.138e+35 on 1 and 14040 DF,  p-value: < 2.2e-16
...

```

**Fonte: Autoria própria (2021).**

Como não houve sazonalidade estocástica e existiu convergência dos dados, a predição realizada contém sazonalidade estocástica nula, conforme demonstra-se na Figura 59:

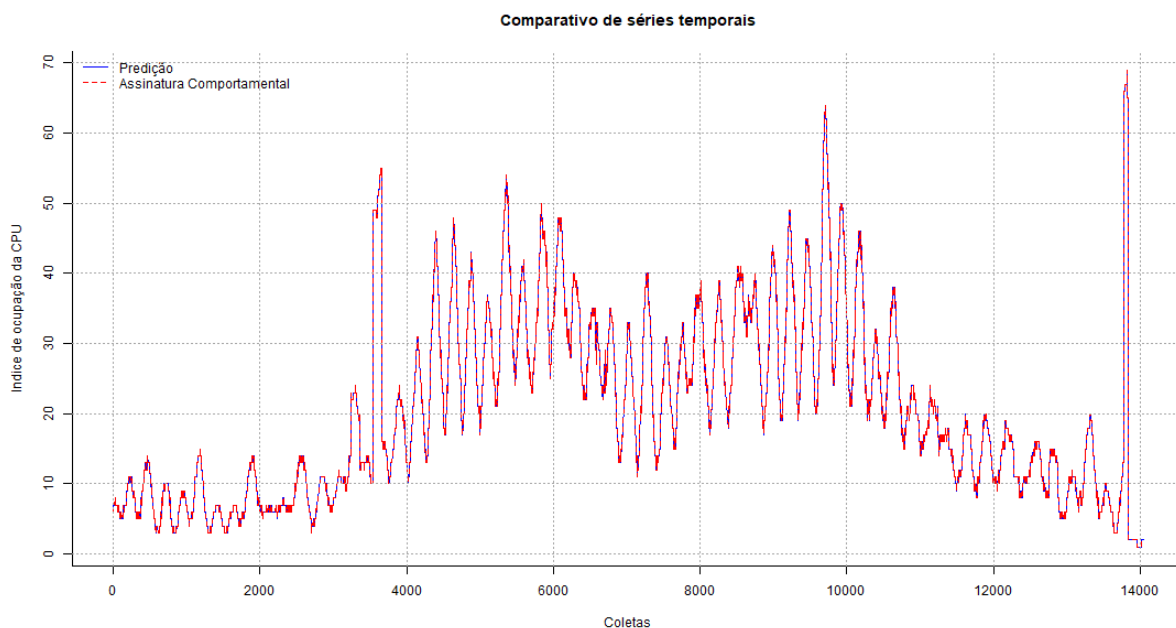
Figura 59 - Dia 2: Comparativo de séries



Fonte: Autoria própria (2021).

Deste modo, a assinatura comportamental deste dia pode ser assim representada, conforme Figura 60:

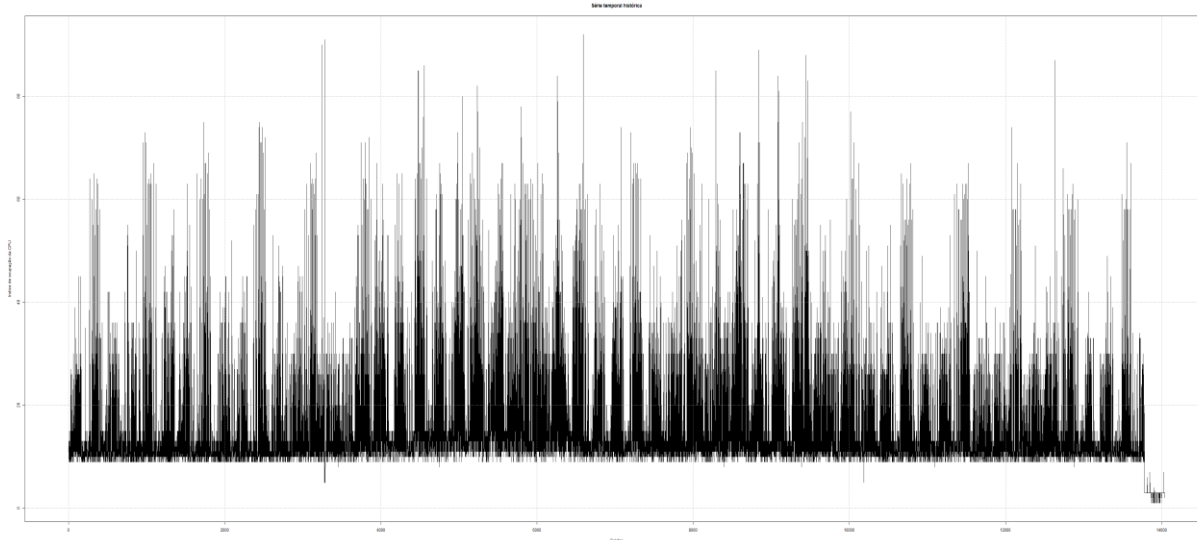
**Figura 60 - Dia 2: Assinatura comportamental**



Fonte: Autoria própria (2021).

Partindo para o **terceiro dia** de coletas, tem-se o seguinte comportamento real para este equipamento no dia, exibido na Figura 61:

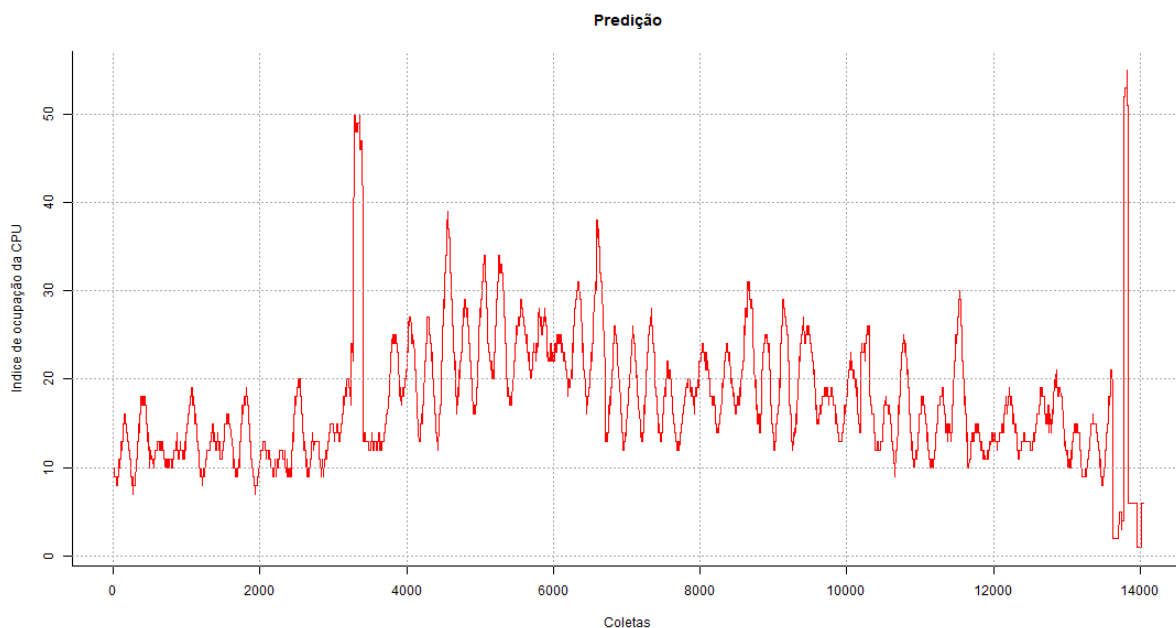
**Figura 61 - Dia 3: Comportamento real**



Fonte: Autoria própria (2021).

Neste dia, o microcomputador apresentou diversos picos de utilização, porém, a área mais escura do gráfico localiza-se abaixo, indicando possível ociosidade. A Figura 62 contém a previsão realizada por intermédio da MMS:

**Figura 62 - Dia 3: Predição**



**Fonte: Autoria própria (2021).**

Os picos de utilização foram eliminados nesta nova série temporal gerada, visto que são eventos raros e que não indicam a tendência de uso do equipamento. Na Figura 63, se apresentará os testes de convergência e sazonalidade estocástica:

**Figura 63 - Dia 3: Análises**

```

...
Call:
lm(formula = predicacao ~ dados[, 1], data = dados)

Residuals:
    Min       1Q   Median       3Q      Max
-3.471e-11  1.000e-15  3.000e-15  4.000e-15  2.770e-13

Coefficients:
            Estimate Std. Error  t value Pr(>|t|)
(Intercept) -7.368e-13  6.621e-15 -1.113e+02 <2e-16 ***
dados[, 1]   1.000e+00  3.503e-16  2.854e+15 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

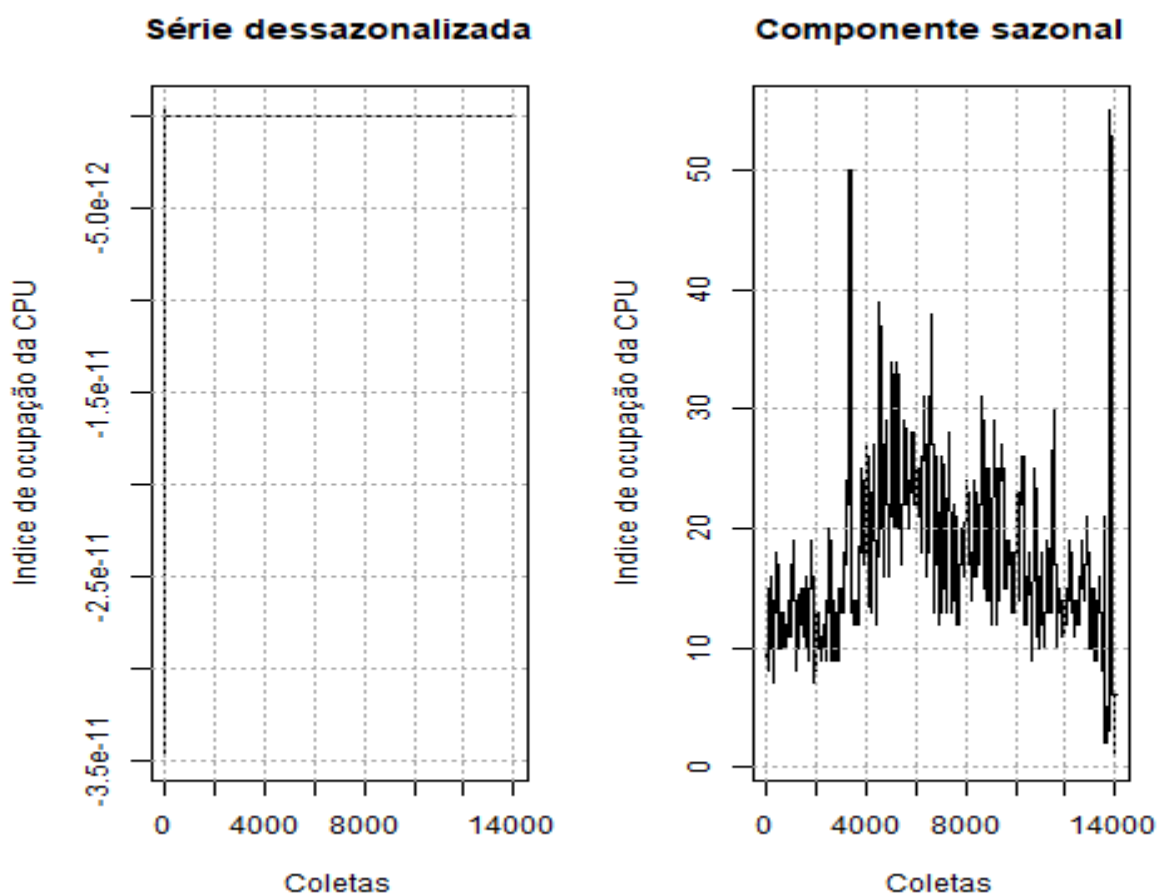
Residual standard error: 2.93e-13 on 14040 degrees of freedom
Multiple R-squared: 1, Adjusted R-squared: 1
F-statistic: 8.148e+30 on 1 and 14040 DF, p-value: < 2.2e-16
...

```

**Fonte: Autoria própria (2021).**

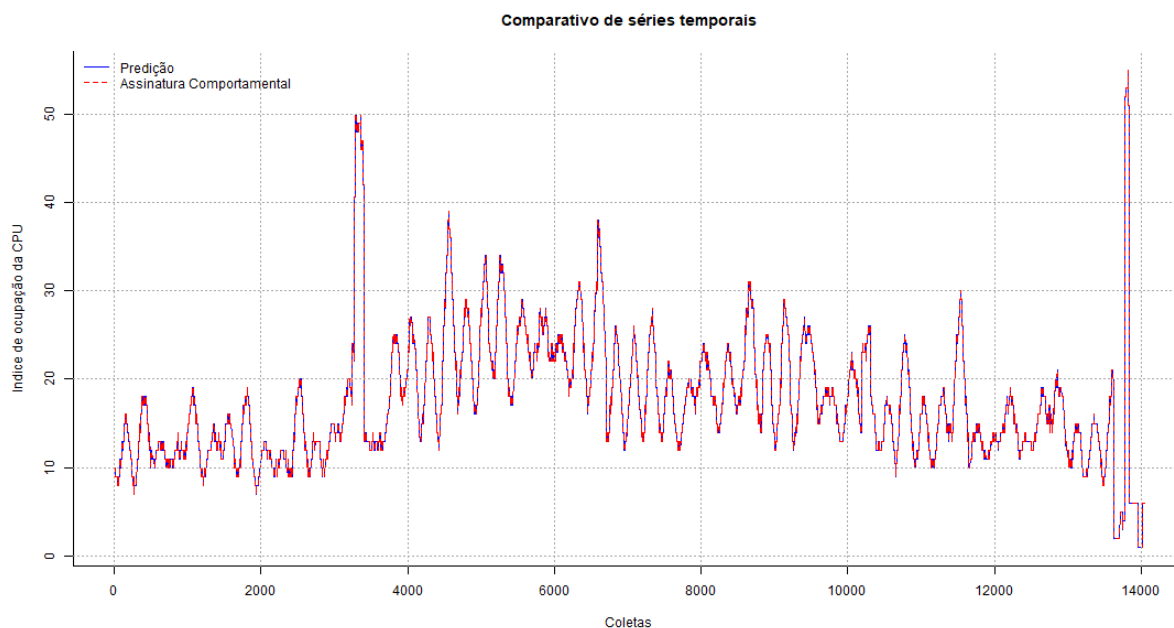
Como nos demais resultados já vistos, não houve sazonalidade estocástica e houve convergência dos dados, levando assim a uma componente sazonal nula, demonstrado na Figura 64:

Figura 64 - Dia 3: Comparativo de séries



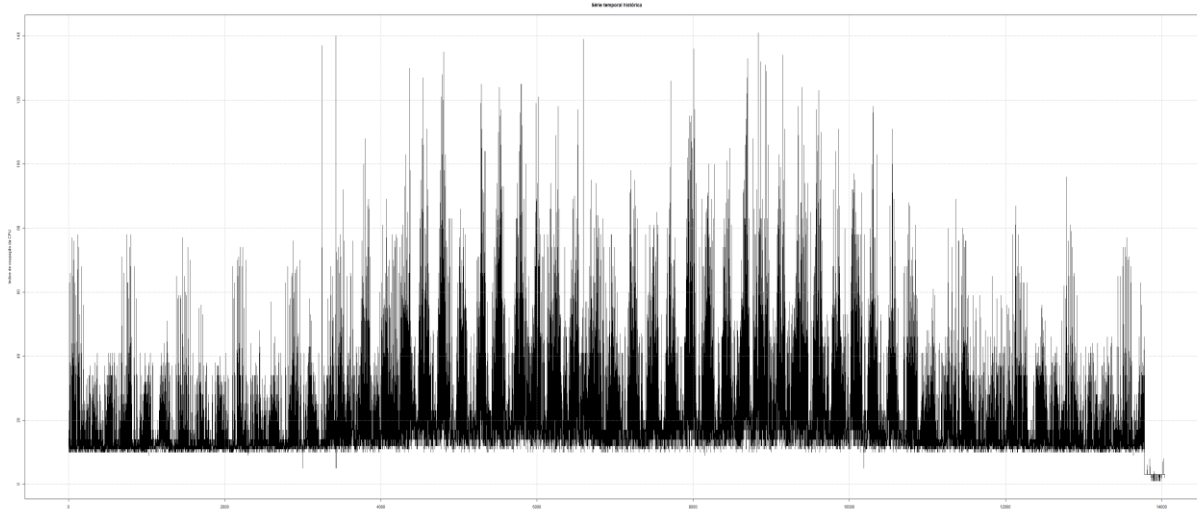
Fonte: Autoria própria (2021).

Deste modo, a assinatura comportamental pode ser assim apresentada no gráfico contido na Figura 65:

**Figura 65 - Dia 3: Assinatura comportamental**

Fonte: Autoria própria (2021).

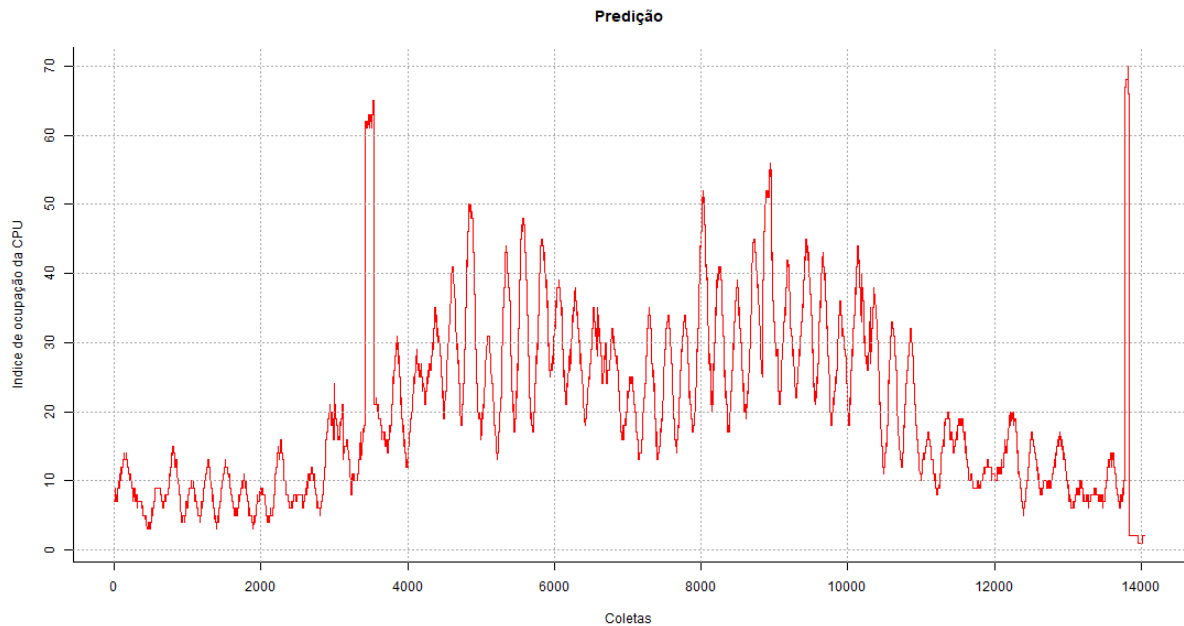
E no **quarto dia** de coletas, pode-se levantar os dados e a seguinte saída é observada na Figura 66:

**Figura 66 - Dia 4: Comportamento real**

Fonte: Autoria própria (2021).

Neste novo dia de coletas, pode-se perceber uma semelhança com o dia anterior, com indícios de uma tendência de ociosidade. Na figura 67, a previsão calculada pela MMS:



**Figura 67 - Dia 4: Predição**

**Fonte: Autoria própria (2021).**

Com índices de ocupação da CPU em torno de 30% a 50%, determinou-se uma tendência de utilização deste recurso computacional próximo destes percentuais. Na Figura 68, os testes de convergência e verificação de sazonalidade estocástica:

**Figura 68 - Dia 4: Análises**

```

...
Call:
lm(formula = predicao ~ dados[, 1], data = dados)

Residuals:
    Min       1Q   Median       3Q      Max
-1.088e-12 -1.800e-14 -1.100e-14 -3.000e-15  1.304e-10

Coefficients:
            Estimate Std. Error  t value Pr(>|t|)
(Intercept)  1.827e-12   1.786e-14  1.023e+02 <2e-16 ***
dados[, 1]   1.000e+00   7.489e-16  1.335e+15 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

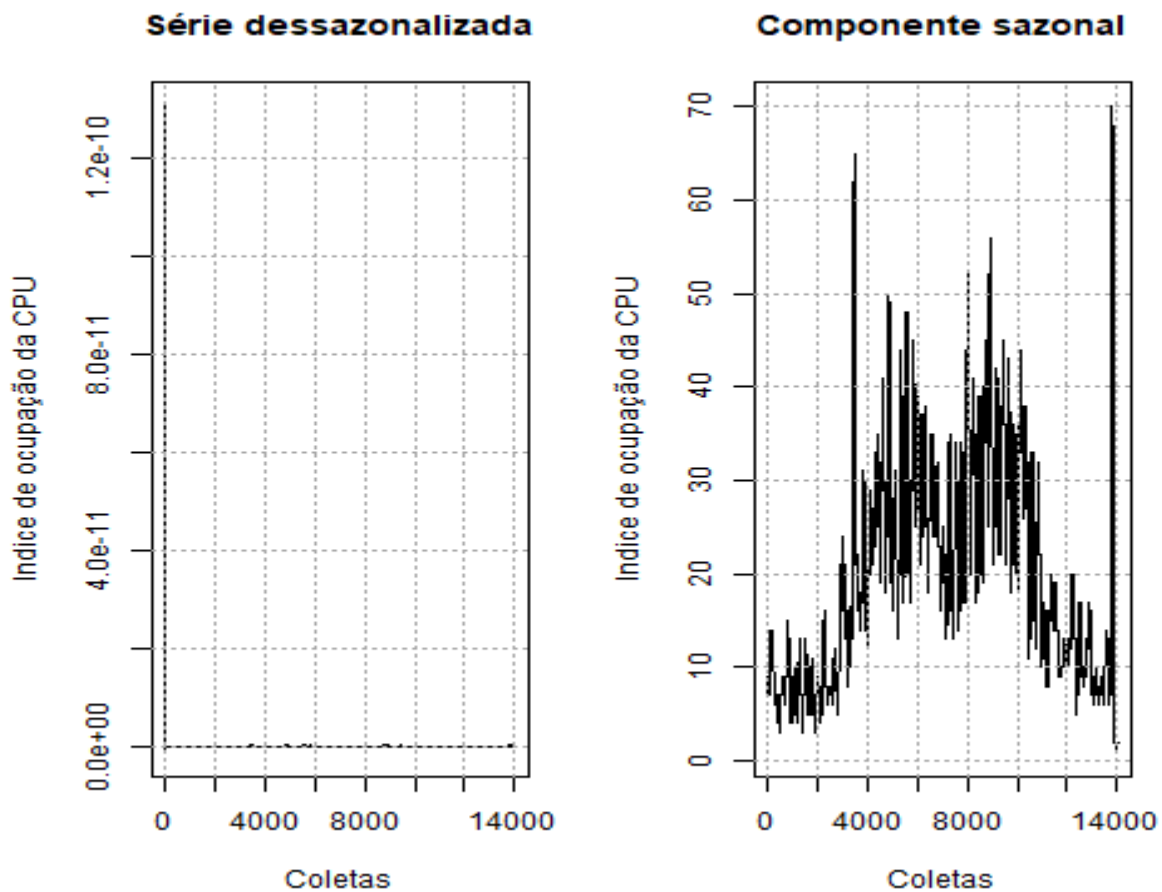
Residual standard error: 1.1e-12 on 14040 degrees of freedom
Multiple R-squared: 1, Adjusted R-squared: 1
F-statistic: 1.783e+30 on 1 and 14040 DF, p-value: < 2.2e-16
...

```

**Fonte: Autoria própria (2021).**

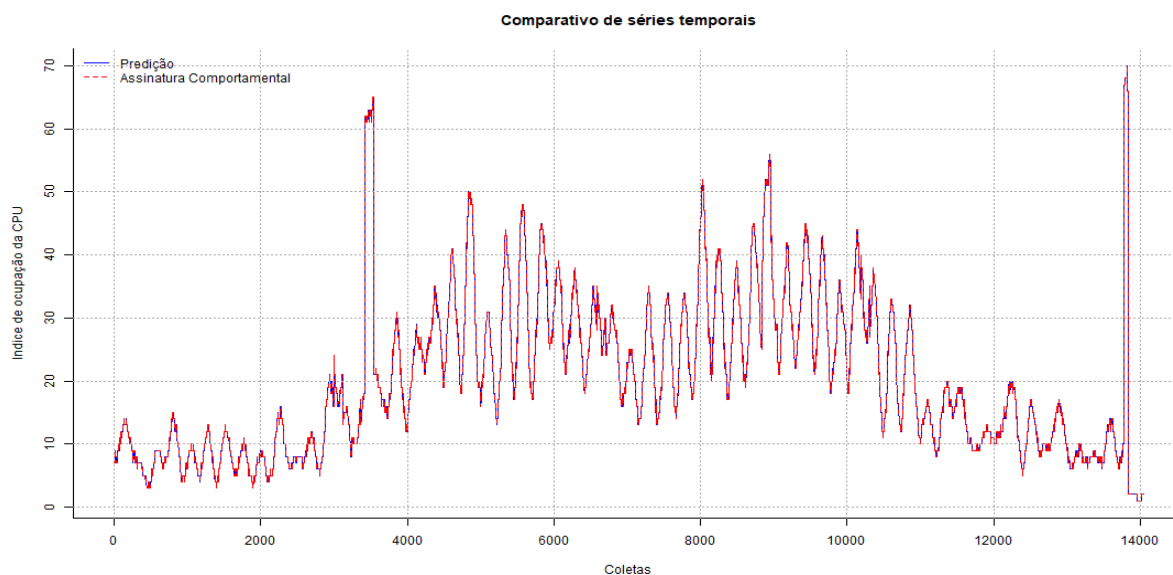
Com a convergência confirmada e a sazonalidade estocástica não encontrada, é possível demonstrar que a componente sazonal é nula, conforme exibido na figura 69:

Figura 69 - Dia 4: Comparativo de séries



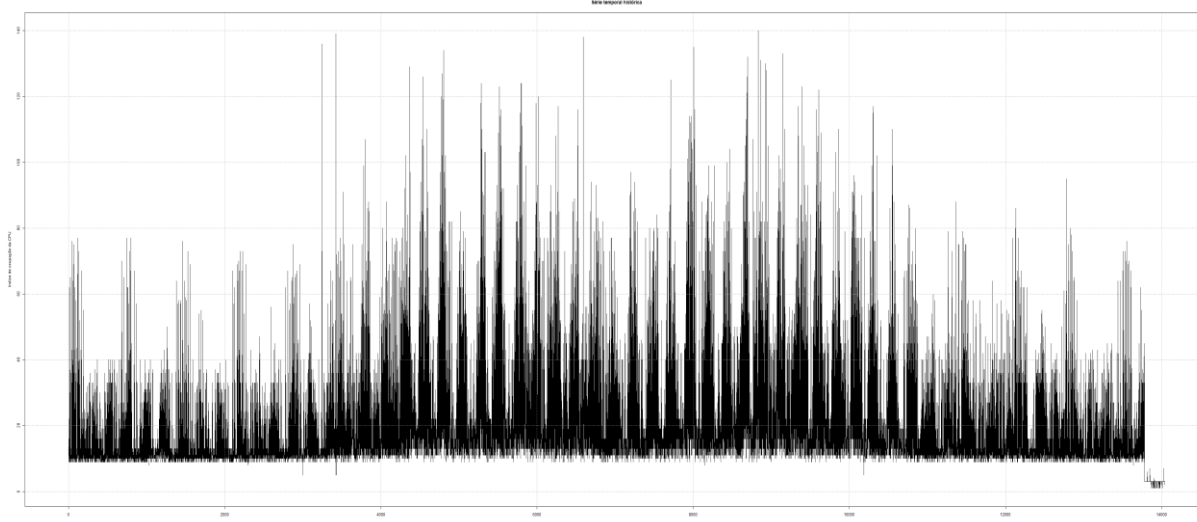
Fonte: Autoria própria (2021).

Por fim, a assinatura comportamental gerada para este dia fica da seguinte forma, demonstrada na Figura 70:

**Figura 70 - Dia 4: Assinatura comportamental**

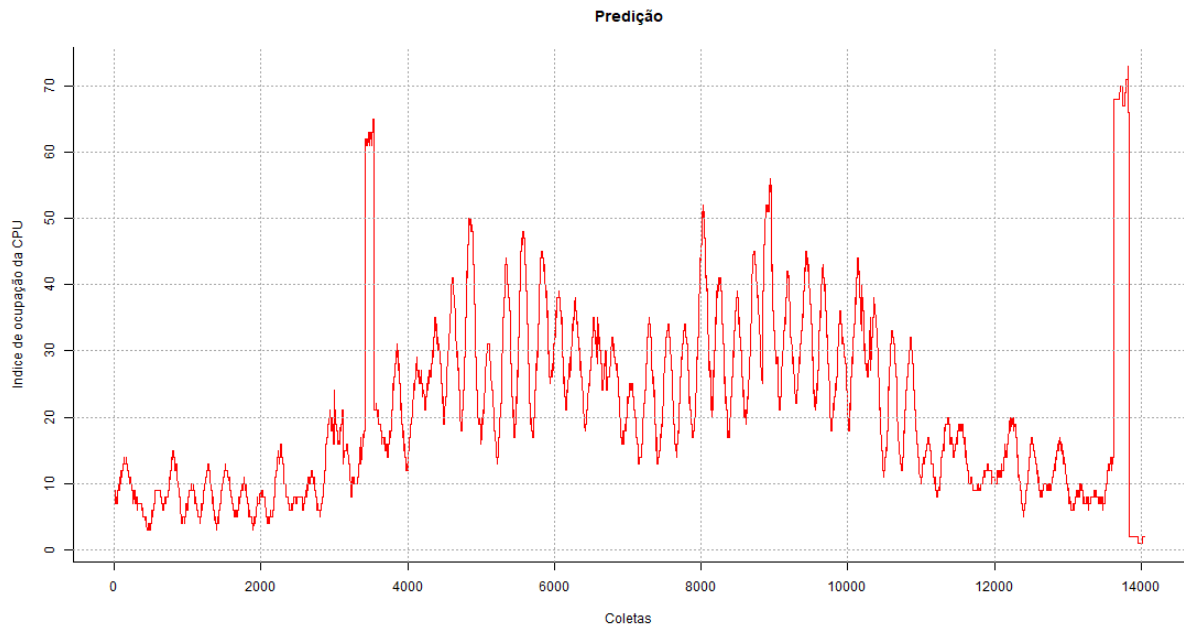
Fonte: Autoria própria (2021).

Findando as análises, será apresentado logo abaixo o **quinto dia**, e último, das coletas realizadas. Na Figura 71, o comportamento real do equipamento neste dia final de observações:

**Figura 71 - Dia 5: Comportamento real**

Fonte: Autoria própria (2021).

Percebeu-se um comportamento similar ao dia anterior, isso pode resultar em uma assinatura comportamental parecida com a análise anterior; na Figura 72, previsão via MMS:

**Figura 72 - Dia 5: Predição**

**Fonte: Autoria própria (2021).**

Uma tendência de uso da CPU abaixo de 50% por um longo período de tempo é uma informação interessante para um Sistema Gerenciador de Recursos (SGR), podendo ser uma máquina a ser escolhida para execução de alguma aplicação. Na Figura 73, estão as informações dos testes de convergência e sazonalidade estocástica:

**Figura 73 - Dia 5: Análises**

```

...
Call:
lm(formula = predicao ~ dados[, 1], data = dados)

Residuals:
    Min       1Q   Median       3Q      Max
-6.400e-13 -1.000e-14 -7.000e-15 -2.000e-15  7.536e-11

Coefficients:
            Estimate Std. Error  t value Pr(>|t|)
(Intercept) 1.013e-12  1.002e-14  1.011e+02 <2e-16 ***
dados[, 1]  1.000e+00  4.022e-16  2.486e+15 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

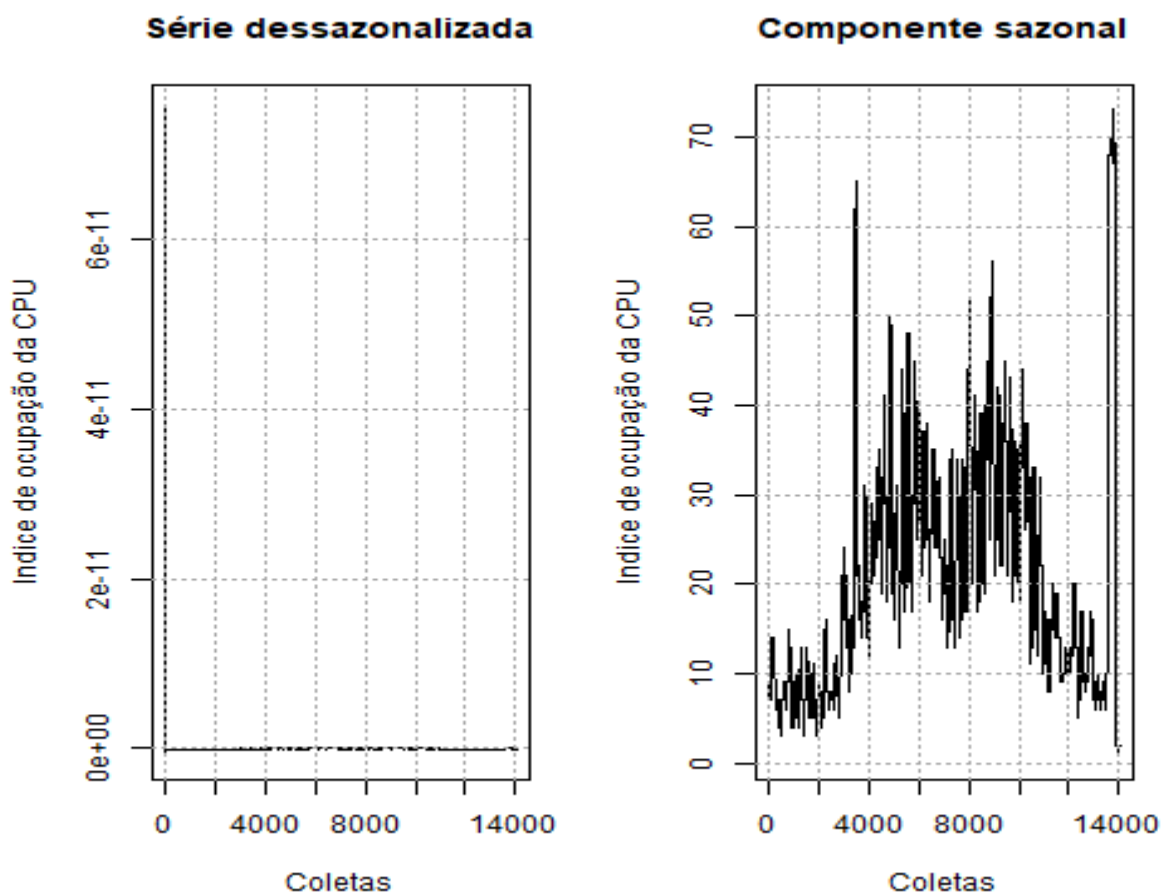
Residual standard error: 6.361e-13 on 14040 degrees of freedom
Multiple R-squared: 1, Adjusted R-squared: 1
F-statistic: 6.182e+30 on 1 and 14040 DF, p-value: < 2.2e-16
...

```

**Fonte: Autoria própria (2021).**

Não restando sazonalidade estocástica e havendo convergência, pode-se demonstrar a componente sazonal nula, como é constatado na Figura 74:

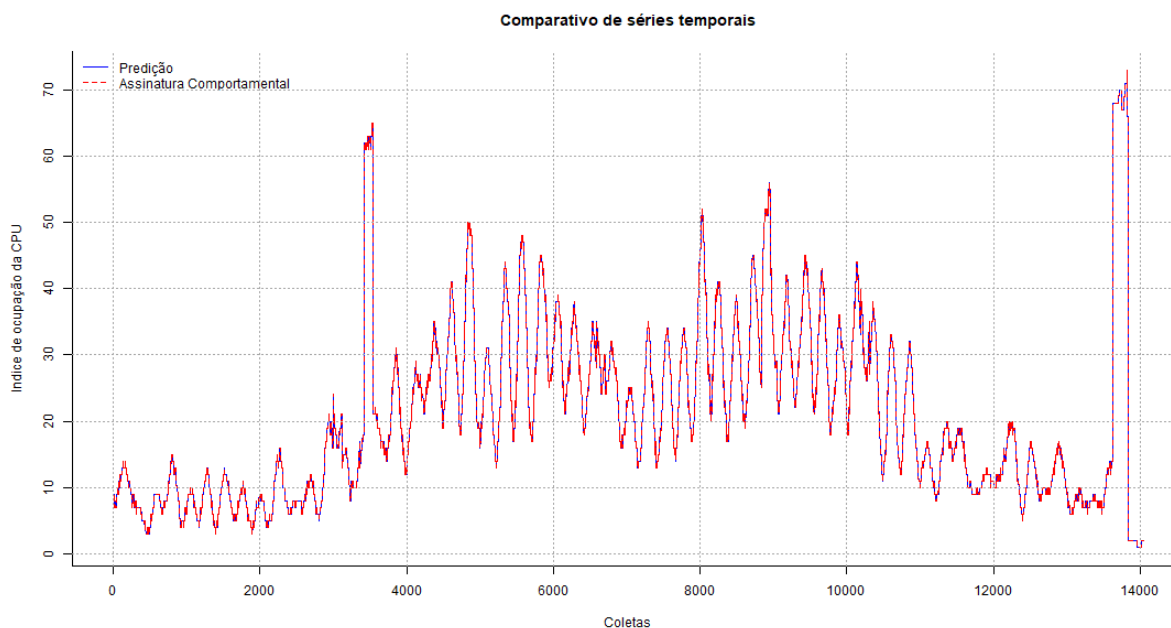
Figura 74 - Dia 5: Comparativo de séries



Fonte: Autoria própria (2021).

Com a componente sazonal zerada, a assinatura comportamental foi definida somente pela predição realizada, conforme é apresentado na Figura 75:

**Figura 75 - Dia 5: Assinatura comportamental**



**Fonte: Autoria própria (2021).**

Com este último gráfico, findou-se os experimentos para esse trabalho. No próximo tópico está explanado os motivos de todas as análises de sazonalidade estocástica serem nulas.

#### 5.4 SAZONALIDADE ESTOCÁSTICA NULA: EXPLICAÇÃO

O princípio da sazonalidade estocástica, segundo Ehlers (2009), é variação da mesma com o passar do tempo. Ocorre que nestes estudos, como há um intervalo muito pequeno entre as observações, de 6 segundos, não se gerou uma componente sazonal de tamanho significativo, ao menos com o comprimento da janela móvel, de modo que os cálculos realizados não conseguiram detectá-la.

Porém, não se pode afirmar que inexistem componentes sazonais. Visto que este cálculo envolve um processo estatístico, e, portanto, não há precisão absoluta no resultado. Erros são comuns em métodos probabilísticos.

Nada impede em que estudos futuros, se tente utilizar outra categoria de sazonalidade para poder captar estes eventos, visto que há um outro conjunto de sazonalidade que não foi contemplado neste trabalho.

## 6 CONCLUSÃO

Com a realização deste trabalho, pode-se perceber que foram obtidos com precisão os dados de uso da CPU com a ferramenta escolhida, e após a aplicação de MMS obteve-se a predição, que sua precisão testada e sua acurácia confirmada, demonstrando que os passos anteriores foram concluídos com êxito.

A definição da assinatura comportamental de um equipamento monitorado, permitirá a transferência desta informação para o módulo gerente do Sistema de Gerenciamento de Recursos (SGR), com uma quantidade de dados bem menor do que se este equipamento informasse seus índices de ocupação de CPU a cada 6 segundos, conforme definido neste trabalho. Apesar de não ser objeto deste estudo, fica evidenciado de que a comunicação dentro do SGR irá ocorrer com um overhead bem mais otimizado.

Considerando-se que o SGR terá armazenado em suas bases de dados, somente as assinaturas comportamentais de seus equipamentos gerenciados, e que estas assinaturas representam o conhecimento prévio de como estarão os índices de ocupação dos recursos computacionais em um futuro próximo, a escolha dos mais adequados para a execução de uma tarefa distribuída poderá ser feita com uma taxa de sucesso e com uma confiabilidade aceitável dentro de parâmetros definidos pelo SGR.

Este trabalho de conclusão de curso apresentou uma abordagem possível e aplicável em um SGR, a qual poderá ser exaustivamente testada e expandida através do prosseguimento dos estudos, visando a obtenção de resultados mais significativos, ficando como sugestão de trabalhos futuros o comparativo entre outras abordagens de predição que poderão ser estudadas levando em consideração os passos descritos neste trabalho.

### 6.2 PUBLICAÇÕES

O presente projeto teve como ponto de partida o artigo “Predição da disponibilidade de recursos computacionais a partir do uso de médias móveis em séries temporais” de Medeiro e Góis (2020), o qual foi apresentado no XXV Seminário de Iniciação Científica e Tecnológica da UTFPR. Neste aludido texto foi apresentada

uma abordagem similar ao deste, porém limitado a um único microcomputador e um dia de coletas de dados de recursos computacionais.

No presente trabalho de conclusão de curso, diferentemente do artigo citado no parágrafo anterior, houve a expansão do conjunto matemático empregado na resolução do problema proposto, e também a utilização de ferramentas mais elaboradas para o auxílio dos cálculos, como o desenvolvimento de programas em código R e C.



## REFERÊNCIAS

- EHLERS, Ricardo Sander. **Análise de Séries Temporais**. 5. ed. Curitiba: Universidade Federal do Paraná, 2009. 118 p.
- DEITEL, Paul J.; DEITEL, Harvey M.. **C: Como programar**. 6. ed. Sudbury: Pearson Brasil, 2011. 818 p.
- GÓIS, Lourival Aparecido de. **Estratégias para comercialização de recursos computacionais em desktop grids**. 2009. 125 f. Tese (Doutorado) - Curso de Engenharia Elétrica e Computação, Departamento de Telemática, Universidade Estadual de Campinas, Campinas, 2009.
- MEDEIRO, Nicolas Charles Oliveira; GÓIS, Lourival Aparecido de. **Predição da disponibilidade de recursos computacionais a partir do uso de médias móveis em séries temporais**. 2020. 8 f., Disponível em: <https://eventos.utfpr.edu.br//sicite/sicite2020/paper/view/7100>. Acesso em: 06 maio 2021.
- MORETTIM, Pedro Alberto; TOLOI, Celia Maria de Castro. **Análise de Séries Temporais**. 2. ed. São Paulo: Edgard Blucher, 2006. 538 p.
- PPLWARE. **Afinal o que é um sistema distribuído?** 2015. Disponível em: <https://pplware.sapo.pt/informacao/afinal-o-que-e-um-sistema-distribuido/>. Acesso em: 27 set. 2020.
- ROESLER, Valter; GASPARONI JÚNIOR, Mário; HUSEMANN, Ronaldo; COSTA FILHO, Roberto Irajá T. da; VALLE, Rafael. MCU em software distribuído em nuvem. **Webmedia'2018: Workshops e Pôsteres**, Salvador, p. 127-131, 12 ago. 2020.
- SILVA, Cidiney J.. **Predição de Séries Temporais no Contexto de Smart Grids**. 2016. 123 f. Tese (Doutorado) - Curso de Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Minas Gerais, Belo Horizonte, 2016.
- SILVA, Maria I. S.; GUIMARÃES, Ednaldo C.; TAVARES, Marcelo. Previsão da temperatura média mensal de Uberlândia, MG, com modelos de séries temporais. **Revista Brasileira de Engenharia Agrícola e Ambiental, Campina Grande**, v. 12, n. 5, p. 481-485, 30 jan. 2008.
- STROBEL, Orlando Maciel. Homogeneidade de séries temporais. **Revista Tecnologia e Humanismo, Curitiba**, v. 2, n. 3, p. 23-37, set. 1987.
- TANENBAUM, Andrew Stuart; BOS, Hebert. **Sistemas Operacionais Modernos**. 4. ed. Holanda: Pearson, 2016. 864 p.
- TANENBAUM, Andrew Stuart; WETHERALL, David J.. **Redes de Computadores**. 5. ed. Holanda: Pearson, 2011. 600 p.
- VERZANI, Jhon. **Getting Started with RStudio**. California: O'Reilly, 2011. 92 p.

**ANEXO A** - Código em R: *Script* para tratamento de séries temporais

```

#lendo arquivo que contém coletas e predição
setwd("C:\\Users\\nickv\\Documents\\BCC\\TCC2")
resultado <- read.csv("resultado.csv",sep = ";", head = T)
#####

#plotando série original
dados <- ts(resultado, start=1, freq=1)
plot(dados[,1], main = "Série temporal histórica", ylab = "Indice de
ocupação da CPU", xlab = "Coletas")
png("sérieOriginal.png")
plot(dados[,1], main = "Série temporal histórica", ylab = "Indice de
ocupação da CPU", xlab = "Coletas")
grid(col='darkgrey',lwd=2)
dev.off()

grid(col='darkgrey',lwd=2)
#####

#plotando predição - sem eliminar sazonalidade
predicao <- ts(read.delim("resultado.csv", sep= ";", head = T), start=1,
freq=1)
predicao <- predicao[,-1]

png("prediçãoI.png")
plot(predicao, main = 'Predição',ylab='Indice de ocupação da CPU', xlab =
'Coletas',bty='l',col='red',lty=1)
grid(col='darkgrey',lwd=2)
dev.off()
#####

#verificando possíveis sazonalidades, se sim: devem ser eliminadas
dados <- data.frame(predicao,dados[,1])
predicao.reg <- lm(predicao~dados[,1], data = dados)
summary(predicao.reg)

predicao.des <- ts(resid(predicao.reg),start = 1, freq=1)
predicao.hat <- ts(fitted(predicao.reg),start=1, freq=1)

png("componenteSazonal.png")
par(mfrow=c(1,2))
plot(main = 'Série dessazonalizada', xlab = 'Coletas', ylab = 'Indice de
ocupação da CPU', predicao.des)
plot(main = 'Componente sazonal', xlab = 'Coletas', ylab = 'Indice de
ocupação da CPU', predicao.hat)
dev.off()

predicao.desn <- predicao.des + mean(fitted(predicao.reg))
png("assinaturaComport.png")
par(mfrow=c(1,1))
plot(predicao,main = 'Comparativo de séries temporais' ,xlab='Coletas',
ylab='Indice de ocupação da CPU',col='blue',bty='l')

par(new=TRUE)

plot(dados[,1],axes=F, ann=F,col='red',lty=2)
legend('topleft',c('Predição', 'Assinatura Comportamental'),col=c('blue',
'red'), lty=1:2,bty='n')
grid(col='darkgrey')
dev.off()
#####

```

**ANEXO B - Código em C: Cálculo de MMS**

```

#include <assert.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef struct coleta{

    char hora[15];
    float coleta;

}coleta;

/*
 * Função que realiza o cálculo da média móvel da série temporal, assim
 gerando a assinatura comportamental do equipamento em estudo.
 */

float mediaMovel(float *p, int janelaMovel) {

    int i;
    float soma = 0;

    for(i=janelaMovel; i>0; i--){
        soma += *p;
        p--;
    }

    return soma/janelaMovel;

}

int main() {

    FILE *saida;
    FILE *entrada;
    coleta coletas[14641];

    char arquivo[50], hora[15];
    float coleta;
    int inc = 1, result, qtdeNos = 0;

    printf("Digite o nome do arquivo de coletas: ");
    scanf ("%s", &arquivo);

    saida = fopen ("resultado.csv", "w+");
    entrada = fopen (arquivo, "r+");

    while (!feof(entrada)){

        result = fscanf(entrada, "%s          %f\n",&hora , &coleta);

        if (result) // Se foi possível ler
            printf("Linha %d -> Hora: %s CPU: %.2f\n",inc ,hora
,coleta);

        coletas[inc-1].coleta = coleta;
        strcpy(coletas[inc-1].hora,hora);
        printf("no vetor: %s, %f\n", coletas[inc-1].hora,
coletas[inc-1].coleta);
        inc++;
    }
}

```

```
    }

    fclose(entrada);

    int i, janelaMovel;

    printf("Digite o tamanho da janela movel da MMS: ");
    scanf("%d", &janelaMovel);

    fprintf (saida, "Real; MMS\n");

    for (i=janelaMovel-1; i<14641; i++){

        printf("%.2f\n", mediaMovel(&coletas[i].coleta,
janelaMovel));
        fprintf(saida, "%.2f; %.2f\n", coletas[i].coleta,
mediaMovel(&coletas[i].coleta, janelaMovel));

    }

    fclose (saida);

    system ("pause");
    return 0;

}
```