

UNIVERSIDADE FEDERAL DO PARÁ
NÚCLEO DE DESENVOLVIMENTO AMAZÔNICO EM ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

JOSIANY BRITO GUIMARÃES

UMA PROPOSTA A SUSTENTABILIDADE DE DATA CENTERS BASEADA NO PARADIGMA
DEVOPS

Tucuruí-PA

Dezembro de 2025



Assinado com Assinatura Eletrônica (Lei 14.063/2020 | Regulamento 910/2014/EC)
Hash SHA256 do original: a5e25b5c0e00a01a802818ebefeb1bdbf74519fd101138a27bc4a35c89c97b2b
Link de validação: <https://valida.ae/f654bb1ae58562a2279c1a15723230185dde04e3e06bc90b1?sv>



Validador



UNIVERSIDADE FEDERAL DO PARÁ
NÚCLEO DE DESENVOLVIMENTO AMAZÔNICO EM ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

JOSIANY BRITO GUIMARÃES

**Uma proposta para a sustentabilidade de data centers baseada no
paradigma DevOps**

Tucuruí-PA
Dezembro de 2025



Assinado com Assinatura Eletrônica (Lei 14.063/2020 | Regulamento 910/2014/EC)
Hash SHA256 do original: a5e25b5c0e00a01a802818ebefeb1bdbf74519fd101138a27bc4a35c89c97b2b
Link de validação: <https://valida.ae/f654bb1ae58562a2279c1a15723230185dde04e3e06bc90b1?sv>



Validador



UNIVERSIDADE FEDERAL DO PARÁ
NÚCLEO DE DESENVOLVIMENTO AMAZÔNICO EM ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

JOSIANY BRITO GUIMARÃES

**Uma proposta para a sustentabilidade de data centers baseada no
paradigma DevOps**

Texto dissertativo apresentado ao Programa de Pós-Graduação em Computação Aplicada do Núcleo de Desenvolvimento Amazônico em Engenharia, da Universidade Federal do Pará, como requisito para a obtenção de créditos referentes à Defesa Final.

Orientador: Prof. Dr. Marcos Amaris Gonzalez

Tucuruí-PA
Dezembro de 2025



**Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD
Sistema de Bibliotecas da Universidade Federal do Pará
Gerada automaticamente pelo módulo Ficat, mediante os dados fornecidos pelo(a)
autor(a)**

B862p Brito Guimarães, Josiany.
Uma proposta para a sustentabilidade de data centers
baseada no paradigma DevOps / Josiany Brito Guimarães. —
2025.
96 f. : il. color.

Orientador(a): Prof. Dr. Marcos Amaris Gonzalez
Dissertação (Mestrado) - Universidade Federal do Pará,
Núcleo de Desenvolvimento Amazônico em Engenharia,
Mestrado Profissional em Computação Aplicada, Tucuruí,
2025.

1. Eficiência energética. 2. sustentabilidade. 3.
carbono. 4. data centers. 5. DevOps. I. Título.

CDD 001.642



Josiany Brito Guimarães

Uma proposta para a sustentabilidade de data centers baseada no paradigma DevOps

Texto dissertativo apresentado ao Programa de Pós-Graduação em Computação Aplicada do Núcleo de Desenvolvimento Amazônico em Engenharia, da Universidade Federal do Pará, como requisito para a obtenção de créditos referentes à Defesa Final.

Conceito: _____

Tucuruí-PA, 19 de dezembro de 2025.

BANCA EXAMINADORA


SIGNATÁRIO



Marcos Tulio Amaris Gonzalez
Data 09/02/2026 19:07
#c5443739060311f1800e42010a2b601f

Prof. Dr. Marcos Amaris Gonzalez
Orientador
PPCA/NDAE/UFPA


Documento assinado digitalmente



FABIO MANOEL FRANCA LOBATO
Data: 08/02/2026 06:01:44-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Fabio Lobato
Examinador externo
(PPGTPC/IEG/UFOPA)

Documento assinado digitalmente



OTAVIO NOURA TEIXEIRA
Data: 07/02/2026 20:17:30-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Otávio Teixeira
Examinador Interno
PPCA/NDAE/UFPA



Validador

*“Ao Rei eterno, Deus único,
imortal e invisível, seja honra e glória
pelos séculos dos séculos. Amém.”*

1 Timóteo 1:17



AGRADECIMENTOS

Primeiramente, registro minha gratidão a Deus, por conceder saúde, força, sabedoria e perseverança ao longo de toda esta jornada, sustentando-me nos momentos de dificuldade e renovando minhas forças para alcançar a conclusão deste trabalho.

Ao Núcleo de Desenvolvimento Amazônico em Engenharia (NDAE) e ao Programa de Pós-Graduação em Computação Aplicada (PPCA), expresso meus agradecimentos pelo suporte institucional, pela infraestrutura e pelas oportunidades acadêmicas que contribuíram de forma significativa para o desenvolvimento desta pesquisa.

Ao meu orientador, Prof. Dr. Marcos Tulio Amaris Gonzalez, manifesto reconhecimento pela orientação acadêmica, pelas contribuições técnicas e científicas e pelo acompanhamento ao longo do desenvolvimento deste trabalho.

Meu reconhecimento especial ao meu esposo Ageu de Lima Guimarães Neto, pelo apoio constante durante todo o período do mestrado. Sua paciência, compreensão e incentivo foram fundamentais para que eu conseguisse seguir firme mesmo nos momentos mais desafiadores dessa caminhada.

À minha mãe, Merian Matos, e à minha sogra, Ana Maria Borges, deixo registrada minha gratidão pelo apoio incondicional, pelo carinho, pela compreensão e pelo constante encorajamento. A presença, as palavras de incentivo e o suporte emocional de vocês foram essenciais para que eu tivesse forças e serenidade para concluir esta etapa importante da minha vida acadêmica.

Este trabalho é fruto não apenas de dedicação e esforço pessoal, mas também da fé, do apoio institucional e do amor daqueles que caminharam comigo ao longo dessa trajetória. Minha sincera gratidão.



RESUMO

Em resposta à crescente demanda por serviços computacionais e ao aumento da carga de trabalho em ambientes de tecnologia da informação, os *data centers* enfrentam desafios relevantes relacionados à eficiência energética e ao cumprimento de metas globais de sustentabilidade. A computação verde, ao promover o uso eficiente de recursos computacionais, apresenta-se como uma abordagem capaz de mitigar os impactos ambientais associados a essas infraestruturas. Neste contexto, esta dissertação investigou a aplicação do paradigma *DevOps*, com ênfase na automação de infraestrutura, como estratégia para promover a sustentabilidade operacional em *data centers*. Foram analisadas e comparadas duas abordagens para a execução de tarefas administrativas rotineiras de TI: a realização manual e a automatização, sendo esta última implementada por meio da ferramenta *Ansible*, utilizada como instância prática do paradigma adotado. O estudo compreendeu a criação do ambiente experimental, a implantação da automação e a avaliação de sua eficácia operacional. O consumo de energia elétrica foi mensurado por meio da plataforma *IMAMS*, enquanto as emissões de CO₂ foram estimadas com base no fator médio anual de emissão de 54,5 gCO₂/kWh. Os resultados indicaram que, no conjunto das tarefas avaliadas, a automação reduziu o tempo total de execução em aproximadamente 23,4% e o consumo de energia elétrica em cerca de 24,3% em comparação à execução manual, refletindo diretamente na diminuição das emissões estimadas de dióxido de carbono, que passaram de 22,72 gCO₂ no cenário manual para 17,19 gCO₂ no cenário automatizado. A partir dos achados, conclui-se que a adoção de práticas *DevOps* apoiadas por automação de infraestrutura contribui de forma mensurável para a eficiência operacional e para a redução de impactos ambientais em *data centers*, sendo a principal contribuição deste trabalho a proposição de uma abordagem experimental reproduzível para a avaliação integrada de tempo de execução, consumo energético e emissões de CO₂ associadas a tarefas de administração de sistemas, oferecendo subsídios técnicos tanto para pesquisadores quanto para gestores de infraestrutura na tomada de decisão sobre a adoção de automação como estratégia para a sustentabilidade em ambientes de TI.

Palavras-chave: Eficiência energética, sustentabilidade, carbono, *data centers*, *DevOps*.



ABSTRACT

In response to the growing demand for computing services and the increased workload in information technology environments, *data centers* face significant challenges related to energy efficiency and compliance with global sustainability goals. *Green computing*, by promoting the efficient use of computing resources, offers an approach to mitigate the environmental impacts associated with these infrastructures. In this context, this dissertation investigated the application of the *DevOps* paradigm, with an emphasis on infrastructure automation, as a strategy to promote operational sustainability in *data centers*. Two approaches to performing routine IT administrative tasks were analyzed and compared: manual execution and automated execution, the latter implemented using the *Ansible* tool, employed as a practical instance of the adopted paradigm. The study comprised the creation of the experimental environment, the implementation of automation, and the evaluation of its operational effectiveness. Electricity consumption was measured using the *IMAMS* platform, while CO₂ emissions were estimated based on the average annual emission factor of 54.5 gCO₂/kWh. The results indicated that, across all evaluated tasks, automation reduced total execution time by approximately 23.4% and electricity consumption by approximately 24.3% when compared to manual execution, directly reflecting a reduction in estimated carbon dioxide emissions, which decreased from 22.72 gCO₂ in the manual scenario to 17.19 gCO₂ in the automated scenario. Based on these findings, it is concluded that the adoption of *DevOps* practices supported by infrastructure automation contributes measurably to operational efficiency and to the reduction of environmental impacts in *data centers*. The main contribution of this work is the proposal of a reproducible experimental approach for the integrated evaluation of execution time, energy consumption, and CO₂ emissions associated with system administration tasks, providing technical support to both researchers and infrastructure managers in decision-making regarding the adoption of automation as a strategy for sustainability in IT environments.

Keywords: Energy efficiency, sustainability, carbon, *data centers*, *DevOps*.



LISTA DE ILUSTRAÇÕES

Figura 1 – Porcentagem dos resultados extraídos de cada plataforma de base de dados.	36
Figura 2 – Fluxograma das fases de mapeamento realizadas	38
Figura 3 – Representação dos cinco elementos essenciais para a sustentabilidade em data centers.	40
Figura 4 – Sistema de gerenciamento de energia.	42
Figura 5 – Exemplo ilustrativo do arquivo hosts/inventory (dados sensíveis suprimi- dos).	50
Figura 6 – Topologia resumida do ambiente (orquestrador e nós-alvo no hipervisor institucional).	51
Figura 7 – Retorno bem-sucedido do teste de conectividade (ansible all -m ping).	51
Figura 8 – Atualização lista de pacotes	52
Figura 9 – Evolução do consumo de energia elétrica no período de 25 de abril a 28 de junho de 2025.	52
Figura 10 – Comparativo Automação x Manual	59
Figura 11 – Comparativo Automação x Manual	63
Figura 12 – Comparativo Segurança	67
Figura 13 – Comparativo atualização OJS	71



LISTA DE TABELAS

Tabela 1 – Comparativo resumido entre ferramentas de automação IaC	24
Tabela 2 – Palavras-chave e seus sinônimos	36
Tabela 3 – Critérios de Inclusão (CI) e Exclusão (CE) utilizados na seleção de artigos para o estudo	37
Tabela 4 – Quantitativo de artigos encontrados, duplicados, excluídos e incluídos por base de dados	37
Tabela 5 – Lista de artigos incluídos na revisão sistemática, indicando os autores e as referências correspondentes	38
Tabela 6 – Questões de pesquisa e suas motivações	39
Tabela 7 – Tarefas organizadas por categoria no ambiente de servidor web	55
Tabela 8 – Leituras de consumo de energia elétrica no período do processo automatizado.	57
Tabela 9 – Leituras de consumo de energia elétrica no período do processo manual	57
Tabela 10 – Consumo estimado de energia elétrica nos períodos analisados	58
Tabela 11 – Leituras de consumo de energia elétrica no período do processo automatizado (LAMP)	60
Tabela 12 – Leituras de consumo de energia elétrica no período do processo manual	61
Tabela 13 – Consumo proporcional estimado de energia elétrica nos processos manual e automatizado (LAMP)	62
Tabela 14 – Consumo proporcional estimado de energia elétrica nos processos manual e automatizado (LAMP)	62
Tabela 15 – Leituras de consumo de energia elétrica no período do processo automatizado (Segurança)	64
Tabela 16 – Leituras de consumo de energia elétrica no período do processo manual (iptables + Fail2Ban)	65
Tabela 17 – Consumo proporcional estimado de energia elétrica nos processos manual e automatizado (Segurança)	66
Tabela 18 – Leituras de consumo de energia elétrica no período do processo automatizado (backup OJS)	69
Tabela 19 – Leituras de consumo de energia elétrica no período do processo manual (atualização do OJS 3.4.0-1)	69
Tabela 20 – Consumo proporcional estimado de energia elétrica nos processos manual e automatizado (Atualização do OJS)	70
Tabela 21 – Comparação entre a execução manual e a automatizada das tarefas com o fator de emissão de 2024.	73



LISTA DE ABREVIATURAS E SIGLAS

ACV	Avaliação do Ciclo de Vida (<i>Life Cycle Assessment</i>)
AWS	<i>Amazon Web Services</i>
CI/CD	Integração Contínua e Entrega Contínua (<i>Continuous Integration / Continuous Delivery</i>)
CO ₂	Dióxido de carbono
COP	Conferência das Partes
CPU	Unidade Central de Processamento
CPDs	Centros de Processamento de Dados
DevOps	<i>Development and Operations</i> (Desenvolvimento e Operações)
DSL	<i>Domain-Specific Language</i>
EDGAR	<i>Emissions Database for Global Atmospheric Research</i>
GCP	<i>Google Cloud Platform</i>
GEE	Gases de Efeito Estufa
GIDC	Gerenciamento de Infraestrutura de <i>Data Center</i>
GW	Gigawatt
HCI	<i>Hyperconverged Infrastructure</i>
HCL	<i>HashiCorp Configuration Language</i>
HD	Disco Rígido
HPC	<i>High Performance Computing</i>
IA	Inteligência Artificial
IaC	Infraestrutura como Código
IMAMS	<i>Intelligent Management and Monitoring System</i>
IPCC	Painel Intergovernamental sobre Mudanças Climáticas
IoT	<i>Internet of Things</i> (Internet das Coisas)
ITU	<i>International Telecommunication Union</i>



ML	<i>Machine Learning</i>
ONU	Organização das Nações Unidas
OJS	<i>Open Journal Systems</i>
RAET	<i>Reliable Asynchronous Event Transport</i>
REAd	Revista Eletrônica de Administração
SCI	<i>Software Carbon Intensity</i>
SSH	<i>Secure Shell</i>
TI	Tecnologia da Informação
tCO ₂ eq	Tonelada de dióxido de carbono equivalente



SUMÁRIO

1	INTRODUÇÃO	15
1.1	Justificativa	16
1.2	Questão da pesquisa	17
1.3	Objetivos	17
1.3.1	Objetivo Geral	17
1.3.2	Objetivos Específicos	17
1.4	Motivação	18
1.5	Estrutura da Dissertação	18
2	REFERENCIAL TEÓRICO	20
2.1	Computação Verde	20
2.2	Componentes de uma Infraestrutura de TI:	20
2.3	Pilha LAMP e Tecnologias Associadas	21
2.4	Ferramentas de Automação	22
2.5	Ferramentas de Automação IaC	24
2.5.1	Resumo técnico das ferramentas	24
2.5.2	Justificativa da escolha do <i>Ansible</i>	25
3	TRABALHOS RELACIONADOS	28
4	METODOLOGIA	31
4.1	Enquadramento metodológico	31
4.2	Revisão da literatura	31
4.3	Contexto do estudo	32
4.4	Procedimentos e coleta de dados	32
4.5	Planejamento dos ensaios	32
4.6	Tratamento e análise	32
4.7	Reprodutibilidade e registros	33
4.8	Limitações e escopo	33
5	REVISÃO BIBLIOGRÁFICA DE LITERATURA	34
5.1	Introdução da Revisão	34
5.2	Metodologia	35
5.2.1	Estratégias de busca	35
5.2.2	Seleção dos artigos:	36
5.2.3	Extração de Dados	37
5.2.4	Questionamentos Levantados	38
5.2.5	Análise dos Resultados	39
5.3	Conclusões da Revisão	48
6	SOLUÇÃO PROPOSTA	49
6.1	Visão geral e princípios	49



6.2	Cenário de infraestrutura	49
6.3	Arquitetura da solução DevOps	49
6.4	Ambiente de teste e validação inicial	50
6.5	Integração com o IMAMS	51
6.5.1	Estimativa das emissões: calculadora do IDESAM vs. cálculo manual	52
6.5.2	Compensação e mitigação.	53
6.6	Tarefas automatizadas no projeto	54
6.7	Reprodutibilidade e evidências	54
7	RESULTADOS E DISCUSSÃO	55
7.1	Automação de Backup e Restauração do Banco de Dados	55
7.1.1	Coleta de dados de consumo de energia do backup e transferência do banco de dados processo automático	56
7.1.2	Coleta de dados de consumo de energia do backup e transferência do banco de dados processo manual	57
7.1.3	Comparativo entre os Processos Manual e Automatizado de Backup do Banco de Dados	58
7.2	Automação para criação do Servidor Web	59
7.2.1	Coleta de dados de consumo de energia na instalação automática da pilha LAMP	60
7.2.2	Coleta de dados de consumo de energia na instalação manual da pilha LAMP	61
7.2.3	Comparativo manual e automatizado.	61
7.3	Automação de Regras de Segurança	63
7.3.1	Coleta de dados de consumo de energia das implantações automáticas de regras de segurança.	64
7.3.2	Coleta de dados de consumo de energia das implantações manuais das regras de segurança.	65
7.3.3	Comparativo manual e automatizado.	65
7.4	Automação da atualização do sistema OJS	66
7.4.1	Coleta de dados de consumo de energia da atualização do sistema OJS e da restauração dos arquivos do processo automático.	68
7.4.2	Coleta de dados de consumo de energia da atualização do sistema OJS e da restauração dos arquivos do processo manual.	69
7.4.3	Comparativo manual e automatizado.	70
7.5	Estimativa de emissões de CO₂ a partir do consumo de energia elétrica das tarefas	71
7.5.1	Metodologia de cálculo	71
7.5.2	Discussão comparativa entre os modelos <i>Green Algorithms</i> e <i>SimGrid/Batsim</i>	73
8	CONCLUSÃO	76
8.1	Trabalhos Futuros	76



REFERÊNCIAS	78
A APÊNDICE - SCRIPTS ANSIBLE PARA CONFIGURAÇÃO	83
A.1 Playbook de Transferência e Restauração do Banco de Dados OJS	83
A.2 Playbook Ansible para instalação da pilha LAMP	85
A.3 Playbook Ansible para configurar segurança com medição automatizada (iptables + fail2ban)	86
A.4 Playbook de Backup, atualização e restauração do sistema Open Journal System	89
B APÊNDICE – LOG DE EXECUÇÃO DAS CONFIGURAÇÕES	93
B.1 Atualização de lista de repositórios	93
B.2 Log do Ansible Ping	93
B.3 Log de Transferência e Restauração do Banco de Dados OJS	94
B.4 Log de execução da instalação da pilha LAMP	95
B.5 Log de configuração de segurança com medição automatizada (iptables + fail2ban)	96
B.6 Log de e Backup, atualização e restauração do sistema Open Journal System	97



1 INTRODUÇÃO

A crescente demanda por dados e serviços computacionais tem impulsionado a expansão de *data centers*, resultando em um aumento no consumo de energia associado a essas infraestruturas. No contexto brasileiro, dados do Ministério de Minas e Energia indicam que, até maio de 2024, havia 22 projetos de *data centers* aguardando aprovação, distribuídos por diferentes Estados, o que evidencia o crescimento acelerado desse setor ([Ministério de Minas e Energia, 2025](#)).

As projeções apontam que, até 2037, a demanda adicional de energia elétrica requerida por essas infraestruturas poderá atingir aproximadamente 2,5 GW, valor comparável ao consumo energético de cidades de médio a grande porte. Para fins de contextualização, o consumo médio mensal de energia elétrica por residência no Brasil é de cerca de 152,2 kWh, com variações conforme a região climática e o perfil dos equipamentos utilizados ([LABEEE, 2025](#)), o que permite dimensionar a magnitude do impacto energético associado à expansão dos *data centers*.

Diante desse cenário, observa-se a necessidade de discutir não apenas a ampliação da capacidade computacional, mas também a sustentabilidade operacional dessas infraestruturas. A adoção de práticas que promovam maior eficiência energética e a redução de impactos ambientais devem estar alinhadas às metas globais de desenvolvimento sustentável, como os Objetivos de Desenvolvimento Sustentável (ODS) da Organização das Nações Unidas, em especial os ODS 9 e 13 ([Organização das Nações Unidas, 2015](#)), que enfatizam a inovação tecnológica sustentável e o enfrentamento das mudanças climáticas.

No Brasil, cerca de 84% da população utiliza a internet, acessando-a por diversos meios, como redes móveis. Conforme o relatório da *International Telecommunication Union (ITU)* ([International Telecommunication Union, 2025](#)), a tendência global é de crescimento contínuo, prevendo-se que 68% da população mundial estará *on-line* em 2024. O relatório também aponta que a taxa de crescimento anual do uso da internet aumentou de 2,7% para 3,4% entre 2022 e 2023, indicando uma demanda crescente por essas infraestruturas. Esse cenário, por sua vez, resulta em aumento do consumo de energia e das emissões de gases de efeito estufa.

Eventos globais relacionados às mudanças climáticas têm sido realizados ao longo das últimas décadas, destacando-se as Conferências das Partes (COP), promovidas pela Organização das Nações Unidas no âmbito da Convenção-Quadro das Nações Unidas sobre Mudança do Clima (UNFCCC). Esses eventos reúnem representantes de cerca de 200 países, com o objetivo de discutir e estabelecer compromissos voltados à mitigação das emissões de gases de efeito estufa e ao enfrentamento das mudanças climáticas ([United Nations Framework Convention on Climate Change, 2024](#)).

Segundo relatórios recentes do WRI Brasil ([WRI Brasil, 2025](#)), as emissões globais de carbono atingiram recordes históricos, superando inclusive os níveis observados durante a



Revolução Industrial. Os dados indicam que, entre 1850 e 2022, as emissões aumentaram cerca de 182 vezes. Observa-se ainda que, em 2020, em decorrência da pandemia de Covid-19, houve uma redução temporária de cerca de 1,5 bilhão de toneladas métricas de emissões; entretanto, essa queda não se sustentou, e já em 2021 os níveis de emissões voltaram a crescer, superando aqueles registrados em 2019.

Uma das soluções viáveis para reduzir as emissões de CO₂ na área de tecnologia é a automação de tarefas diárias por meio de ferramentas voltadas à infraestrutura de TI, como as propostas pela abordagem conhecida como *DevOps*. Conforme definido por Patrick Debois (DEBOIS, 2008), o termo resulta da fusão de "*Software Development*" (desenvolvimento de software) com "*Information Technology Operations*" (operações de TI).

Segundo o próprio Debois (DEBOIS, 2008), tradicionalmente, as atividades de desenvolvimento, infraestrutura e operações eram desempenhadas por grupos distintos dentro das organizações, que trabalhavam de forma isolada e se integravam apenas nas fases finais dos projetos. Esse modelo frequentemente resultava em conflitos organizacionais, atrasos na entrega de soluções e divergências entre os ambientes utilizados pelos desenvolvedores e os configurados pelas equipes de infraestrutura, o que ocasionava retrabalho. Esses ciclos repetitivos de correção e processos ineficientes contribuíam para o uso excessivo de recursos computacionais, como memória e CPU, impactando negativamente a eficiência operacional e, conseqüentemente, o consumo de energia.

Portanto, ferramentas associadas à abordagem *DevOps* podem ser utilizadas para promover a eficiência energética em organizações de diferentes portes, ao possibilitar a automação de tarefas de gerenciamento de infraestrutura, como a configuração de servidores, o provisionamento de máquinas virtuais e o monitoramento do consumo de recursos computacionais. Estudos recentes indicam que a integração de práticas de automação e sustentabilidade no contexto de *DevOps* contribui para a otimização do uso de recursos e para a redução de impactos ambientais em ambientes de TI (ONOJA; ONYENZE; AKINTOYE, 2024).

1.1 Justificativa

O crescimento da demanda por serviços digitais tem impulsionado a expansão da infraestrutura computacional e intensificado o consumo de energia associado aos *data centers*. No contexto brasileiro, dados recentes do Ministério de Minas e Energia indicam que o número de solicitações de conexão de *data centers* à rede elétrica ultrapassou 50 solicitações em 2025, evidenciando a pressão crescente dessas infraestruturas sobre o sistema energético nacional (Ministério de Minas e Energia, 2025).

Paralelamente, observa-se um aumento do interesse científico em abordagens voltadas à sustentabilidade da Tecnologia da Informação. Um estudo bibliométrico baseado na base de dados Web of Science aponta que, no Brasil, a maioria das pesquisas sobre *Green IT* tem como



cenário de aplicação as Instituições de Ensino Superior (IES), representando aproximadamente 67% dos estudos analisados (SILVA et al., 2018). Esse resultado evidencia a relevância das universidades como ambientes estratégicos para a adoção e a avaliação de práticas sustentáveis na infraestrutura de TI.

1.2 Questão da pesquisa

Ferramentas e práticas de automação de infraestrutura apresentam grande potencial para otimizar o gerenciamento de recursos computacionais e para contribuir para a eficiência energética. No entanto, a literatura acadêmica ainda carece de estudos empíricos conclusivos que avaliem o impacto da adoção de práticas de *DevOps* na redução do consumo de energia e das emissões de CO₂ em ambientes de TI, especialmente em *data centers* de instituições públicas de ensino superior.

Diante desse contexto, a presente pesquisa busca responder à seguinte questão de pesquisa:

A adoção de práticas de DevOps para a automação de infraestrutura pode contribuir para a redução do consumo de energia e das emissões de CO₂ em data centers de instituições de ensino superior?

1.3 Objetivos

1.3.1 Objetivo Geral

Investigar a implementação de práticas de eficiência energética em data centers acadêmicos, por meio do uso da ferramenta de automação Ansible, avaliando a viabilidade dessa solução de código aberto para otimizar o consumo de energia e promover práticas sustentáveis, sem comprometer a qualidade dos serviços oferecidos.

1.3.2 Objetivos Específicos

- Conduzir uma revisão bibliográfica da literatura sobre práticas de sustentabilidade e eficiência energética em *data centers*, a fim de identificar estratégias e abordagens adotadas no cenário atual;
- Implementar a automação de tarefas administrativas por meio do Ansible, com foco na instalação e atualização de serviços como a pilha LAMP, o OJS e ferramentas de segurança;
- Realizar medições experimentais de tempo de execução, desempenho e consumo de energia durante os processos manuais e automatizados;



- Estimar as emissões de CO₂ associadas aos processos avaliados, a partir do consumo energético medido e de fatores de emissão adequados;
- Comparar os resultados obtidos entre os métodos manual e automatizado, avaliando o impacto da automação na eficiência energética, no tempo de execução das tarefas e nas emissões de CO₂.

1.4 Motivação

A crescente modernização dos processos acadêmicos e administrativos impõe novos desafios às universidades públicas brasileiras, que precisam conciliar eficiência operacional, restrições orçamentárias e compromisso com a sustentabilidade. No contexto de uma universidade no norte do Brasil, a equipe técnica responsável pela infraestrutura de Tecnologia da Informação é reduzida, o que torna o gerenciamento manual de servidores, serviços e sistemas um processo oneroso, sujeito a retrabalho e com elevado consumo de tempo e energia.

Essa realidade reforça a necessidade de adotar soluções que promovam a automação, a padronização e a racionalização de recursos, reduzindo o esforço humano e o impacto ambiental associado às operações de TI. Nesse cenário, as práticas de Infraestrutura como Código (IaC) e DevOps configuram-se como alternativas promissoras para a sustentabilidade computacional, ao proporcionarem maior controle sobre o ciclo de vida dos sistemas, otimizarem o uso de recursos e viabilizarem o monitoramento do consumo de energia.

No entanto, tais práticas, por si sós, não asseguram a plena sustentabilidade. Elas precisam ser acompanhadas de políticas institucionais de eficiência energética, de capacitação técnica e da incorporação de métricas ambientais aos processos de gestão tecnológica. Assim, a adoção de metodologias automatizadas em uma instituição pública representa não apenas um avanço em direção à modernização, mas também um passo concreto rumo a uma cultura de sustentabilidade digital alinhada aos princípios da Agenda 2030 da Organização das Nações Unidas (ONU), em especial aos Objetivos de Desenvolvimento Sustentável (ODS) 7 e 13, que tratam de “energia acessível e limpa” e de “ação contra a mudança global do clima” (UNIDAS, 2015).

Diante desse contexto e em consonância com os objetivos estabelecidos, este trabalho tem como motivação compreender, de forma empírica e mensurável, como a automação de infraestrutura pode contribuir para a redução do consumo energético e das emissões de CO₂ em ambientes reais de *data centers*. Adicionalmente, busca-se evidenciar o potencial da automação como instrumento estratégico para promover a eficiência operacional e a responsabilidade ambiental no contexto da administração pública universitária.

1.5 Estrutura da Dissertação

Este trabalho está organizado da seguinte forma:



- Capítulo 1: Introdução, onde é apresentado o contexto do trabalho e a justificativa para sua realização;
- Capítulo 2: referencial teórico, que discute os conceitos fundamentais relacionados aos temas abordados ao longo do texto;
- Capítulo 3: apresenta os trabalhos relacionados, destacando pesquisas que abordam a automação de infraestrutura, eficiência energética e sustentabilidade em data centers;
- Capítulo 5: revisão bibliográfica da literatura, na qual são analisadas as discussões sobre eficiência energética, estratégias utilizadas, impacto ambiental e a necessidade de adotar práticas que promovam *data centers* sustentáveis;
- Capítulo 4: metodologia, que descreve os passos seguidos para a condução da pesquisa e a elaboração da solução proposta;
- Capítulo 6: solução proposta, apresentando o desenvolvimento da solução, os testes realizados e os resultados iniciais;
- Capítulo 7: Resultado, onde foi realizada a implantação da automação e realizados processos com seu uso;
- Capítulo 8: neste capítulo, são relatadas as considerações finais do trabalho, suas implicações e possíveis trabalhos futuros;
- Por fim, são apresentadas as Referências, que listam todas as fontes utilizadas para fundamentar a pesquisa.



2 REFERENCIAL TEÓRICO

A expansão dos serviços de Tecnologia da Informação (TI) resulta em um consumo crescente de recursos computacionais, o que acarreta desafios tanto ambientais quanto operacionais. Para mitigar esses impactos, diversas estratégias têm sido desenvolvidas, com destaque para a Computação Verde, que busca reduzir a pegada ecológica dos Centros de Processamento de Dados (CPDs), e para as ferramentas de automação, que aprimoram a eficiência energética e a gestão da infraestrutura de TI.

Este capítulo apresenta um panorama teórico dos principais conceitos e práticas relacionados à Computação Verde, abordando estratégias para otimizar a infraestrutura de TI e a importância da automação na redução do consumo energético. Além disso, são discutidos componentes essenciais de um ambiente computacional, bem como as ferramentas de automação e orquestração utilizadas para garantir maior eficiência na administração de sistemas complexos.

2.1 Computação Verde

A computação verde abrange uma série de práticas que visam reduzir o impacto ambiental dos *data centers*. Segundo (MARCO, 2024), foi identificada uma série de estratégias que podem ser utilizadas para promover a sustentabilidade, como otimização do fluxo de ar, melhoramento da disposição dos equipamentos, gerenciamento do armazenamento de dados e virtualização dos servidores; essas estratégias ajudam a reduzir o consumo de energia. Segundo Uddin, Mueen e Rahman, Azizah Abdul (UDDIN; RAHMAN, 2012), TI Verde se configura como uma alternativa para promover a sustentabilidade e a economicidade no setor de TI. Ela se baseia em pilares como virtualização, terceirização de Computação em Nuvem, reciclagem, aquisição e gerenciamento eficiente de energia. Essas ações citadas anteriormente têm como principal objetivo reduzir o (CO₂) um gás incolor, inodoro e insípido, formado por dois átomos de oxigênio e um átomo de carbono. É um dos gases de efeito estufa que contribuem para o aquecimento global. O CO₂ é produzido naturalmente pela respiração e pela decomposição de organismos vivos, mas também é produzido por atividades humanas, como a queima de combustíveis fósseis (National Oceanic and Atmospheric Administration (NOAA), 2022).

2.2 Componentes de uma Infraestrutura de TI:

Para compreender melhor os fatores que impulsionam o consumo de energia, é fundamental discutir a infraestrutura de (TI). Segundo (GENG, 2014) ele diz que, geralmente, os principais componentes de uma infraestrutura estão concentrados em *data centers*, que são instalações físicas destinadas a abrigar dispositivos de computação. Entre esses dispositivos, destacam-se os servidores, que podem ser físicos ou virtuais, e os sistemas de armazenamento, também



conhecidos como equipamentos de *storage*, responsáveis pelo armazenamento de dados. Além disso, outros componentes essenciais incluem os *hardwares* de rede, como roteadores, *switches*, *firewalls* e pontos de acesso *Wi-Fi*, que garantem a conectividade e a segurança da comunicação entre os dispositivos.

Segundo autor (PEDRONI; COELHO, 2023), essas instalações são projetadas para garantir o funcionamento ininterrupto dos serviços de Tecnologia da Informação e contêm um sistema de resfriamento para manter os equipamentos em temperatura adequada, sendo essenciais para empresas que dependem de tecnologia para seu funcionamento.

Segundo (CHANOPAS; KRAIRIT; KHANG, 2006; BYRD, 2000), para assegurar uma operação ininterrupta, fontes de energia redundantes são empregadas; isso inclui a energia fornecida pela concessionária da cidade, aliada a um banco de baterias ou a um gerador. Dessa maneira, no caso de uma interrupção na energia principal, o banco de baterias ou o gerador entra em ação, assegurando a continuidade dos serviços prestados.

Existe também outro tipo de centro de processamento de dados, chamado de provedor de serviços em nuvem. Essa tecnologia ajuda empresas que não têm interesse ou não dispõem de recursos para ter sua própria estrutura física a utilizar serviços diversos. Uma das mais conhecidas é a *Amazon Web Services* ^[1]. Segundo o livro (VERAS, 2013), trata-se de uma plataforma de serviços de computação em nuvem da empresa *Amazon*. Ela oferece diversos serviços, como infraestrutura global, incluindo computação, armazenamento, rede, gerenciamento de banco de dados e gerenciamento da infraestrutura. Essas soluções podem ser utilizadas por empresas de todos os portes. Existem também outras plataformas, como *Azure da Microsoft* e *Google Cloud*, que funcionam de forma semelhante.

2.3 Pilha LAMP e Tecnologias Associadas

Um servidor web é responsável por disponibilizar sites e aplicações na internet, gerenciando as requisições dos navegadores dos usuários e entregando o conteúdo solicitado de forma adequada. Para viabilizar esse funcionamento, é necessária a integração de diferentes componentes de software que atuam de forma complementar na infraestrutura do servidor.

Nesse contexto, a configuração do ambiente de hospedagem adotado neste trabalho baseia-se na pilha LAMP, composta pelo sistema operacional Linux, pelo servidor *web Apache*, pelo gerenciador de banco de dados *MariaDB* e pela linguagem de programação *PHP*. Essa combinação é amplamente utilizada em ambiente web por sua estabilidade, flexibilidade e adoção consolidada tanto em contextos acadêmicos quanto corporativos.

O *Apache* é um servidor HTTP de código aberto amplamente empregado, responsável por interpretar as requisições HTTP recebidas e entregar os arquivos necessários à exibição correta

¹ <https://aws.amazon.com/pt/>



do conteúdo dos sites aos usuários (FIELDING; KAISER, 1997). Sua ampla adoção decorre do suporte a múltiplas plataformas, da extensibilidade por meio de módulos e da integração eficiente com linguagens de programação voltadas ao desenvolvimento web.

Entre essas linguagens, destaca-se o **PHP**, uma linguagem de programação de código aberto amplamente utilizada no desenvolvimento de aplicações *web* dinâmicas. Conforme apresentado por (NIEDERAUER, 2017), sua compatibilidade com a maioria dos servidores HTTP, incluindo o Apache, contribui significativamente para sua popularidade, permitindo a construção de sistemas interativos e orientados a banco de dados.

O armazenamento e o gerenciamento das informações são realizados por meio do sistema de gerenciamento de banco de dados **MariaDB**. Desenvolvido como alternativa ao MySQL após sua aquisição pela Oracle, o MariaDB mantém compatibilidade com este último, diferenciando-se por ser mantido como software livre de código aberto e por apresentar melhorias de desempenho e segurança (BARTHOLOMEW, 2012). Essa característica o torna uma escolha frequente em ambientes que demandam confiabilidade e aderência aos princípios do software livre.

Além dos componentes centrais da pilha LAMP, a segurança do ambiente de servidor é reforçada por meio do uso de ferramentas complementares. O **Fail2Ban** é uma solução amplamente utilizada em servidores Linux para mitigar acessos indevidos, monitorando arquivos de log em busca de tentativas repetidas de autenticação mal-sucedidas, como conexões via SSH. Ao identificar padrões suspeitos, a ferramenta aplica regras temporárias no *firewall*, bloqueando endereços IP por um período previamente definido (Fail2Ban Contributors, 2024).

O **iptables**, por sua vez, atua como um mecanismo de filtragem de pacotes, permitindo ou bloqueando o tráfego de entrada e de saída da rede com base em regras previamente configuradas (Red Hat, Inc., 2013). Seu uso permite um controle granular do tráfego de rede, o que é fundamental para a proteção da infraestrutura contra acessos não autorizados.

Por fim, a adoção dessas tecnologias também se justifica diante de ameaças recorrentes a servidores expostos à internet. Conforme destacado por (ALBALAWI et al., 2022), o *website defacement* constitui um vetor de ataque significativo, comprometendo a integridade visual e funcional de sistemas online. A literatura aponta a importância do monitoramento contínuo e da aplicação de mecanismos de segurança para reduzir a superfície de ataque e preservar a confiabilidade dos serviços disponibilizados.

2.4 Ferramentas de Automação

Conforme os autores do artigo intitulado “Melhores Práticas para TI verde” (REDDY et al., 2018), a implantação de ferramentas de automação é destacada como uma das boas práticas a serem adotadas. Eles descrevem como as ferramentas são necessárias para automatizar tarefas, como provisionamento, configuração, aplicação de *patches*, gerenciamento de versões e garantia



de conformidade. No estudo realizado pelos autores, observou-se que a maioria dos *data centers* adota ferramentas de automação para otimizar processos, reduzir as taxas de erro e aprimorar o desempenho.

Segundo a fabricante Red Hat (REDHAT, 2018), a **automação de infraestrutura de TI** reduz a intervenção humana e torna a administração de componentes de armazenamento, sistema operacional, rede, *software* e *hardware* mais eficiente. Eles executam tarefas automaticamente; assim, reduzem o tempo de execução e mitigam a possibilidade de erros humanos, simplificando a administração de sistemas complexos.

Com o surgimento desse novo olhar sobre a forma de se trabalhar, foi criado o método de *DevOps*, uma abordagem de colaboração entre desenvolvimento e operações que visa aprimorar a entrega de *software* por meio de automação e integração contínua. Métodos ágeis no desenvolvimento de *software* enfatizam flexibilidade, colaboração e entrega contínua de valor ao cliente (REDHAT, 2022; BECK et al., 2001). Essas abordagens promovem equipes ágeis, responsivas e permitem adaptação às mudanças e entregas rápidas e de qualidade.

Infraestrutura como Código

De acordo com (HASHICORP, 2025b) a IaC representa uma metodologia no gerenciamento de infraestrutura, empregando princípios de desenvolvimento de software para configurar e provisionar recursos de TI. Esta abordagem permite definir com precisão e eficiência a configuração de elementos como servidores, armazenamento e redes por meio de código. Sendo possível criar, modificar e gerenciar a infraestrutura de forma segura, consistente e repetível.

As vantagens da IaC são essenciais para ambientes de TI. Em primeiro lugar, destaca-se a segurança, que reduz significativamente a incidência de erros humanos no gerenciamento da infraestrutura, garantindo um ambiente mais confiável e estável. Outro benefício é a padronização das configurações, que assegura consistência em todos os ambientes, do desenvolvimento à produção, minimizando discrepâncias que poderiam causar falhas operacionais.

Outro aspecto relevante é a repetibilidade. A automação das tarefas de gerenciamento de infraestrutura resulta em economia de tempo e recursos; assim, as equipes podem concentrar seus esforços em atividades de maior valor estratégico. Tal eficiência operacional impulsiona a capacidade de resposta da organização às demandas do negócio, elevando sua competitividade e agilidade.

O uso de IaC não apenas simplifica a gestão dos recursos de TI, mas também impulsiona os pilares da segurança, consistência e eficiência em toda a organização, configurando-se como uma ferramenta indispensável na era da transformação digital.



2.5 Ferramentas de Automação IaC

A automação da infraestrutura, por meio do paradigma de Infraestrutura como Código (IaC), tornou-se indispensável na gestão moderna de ambientes computacionais. Diversas ferramentas têm-se sobressaído por possibilitarem a definição de configurações e o provisionamento de forma automatizada e versionável. Na Tabela 1 são apresentadas as ferramentas e é realizado um comparativo de algumas características.

Tabela 1 – Comparativo resumido entre ferramentas de automação IaC

Ferramenta	Tipo	Agente	Linguagem	Apren.	Estado	Uso principal
Ansible	Decl.	Não	YAML	Baixa	Não	Config. de sistemas e apps
Puppet	Decl.	Sim	DSL (Ruby)	Média	Sim	Ambientes padronizados
Chef	Imper.	Sim	Ruby	Alta	Sim	Lógicas complexas
Terraform	Decl.	Não	HCL	Média	Sim	Infraestrutura (cloud)
SaltStack	Híbrido	Opc.	YAML + Jinja	Média	Sim	Escalabilidade massiva

Fonte: Elaborado pela autora com base em (ÖZDOĞAN; CERAN; ÜSTÜNDAĞ, 2023), (PALMA; NUCCI; TAMBURRI, 2020) e na documentação oficial das ferramentas.

2.5.1 Resumo técnico das ferramentas

A ferramenta Ansible emprega uma abordagem declarativa, distinguindo-se pela simplicidade e pela ausência de agentes nos nós gerenciados. Sua operação baseia-se no uso de arquivos *YAML*, denominados *playbooks*, para a descrição de configurações que são executadas remotamente via SSH. É amplamente utilizada em contextos que demandam automação leve e rápida. Cabe ressaltar que (PALMA; NUCCI; TAMBURRI, 2020) propuseram a biblioteca *Ansible Metrics*. Esta ferramenta consiste em um conjunto de métricas estáticas desenvolvidas para avaliar a qualidade de *scripts*.

O *Puppet*, segundo (SHAMBAUGH; WEISS; GUHA, 2016), é uma ferramenta declarativa baseada em agentes, amplamente empregada em ambientes corporativos. A plataforma utiliza uma linguagem declarativa própria, fundamentada em *Ruby*, para descrever os estados desejados do sistema, aplicando as configurações de forma automatizada. A ferramenta é particularmente útil para o gerenciamento de infraestruturas de grande porte, assegurando configurações padronizadas, controle de estado robusto e elevada escalabilidade.

Segundo a documentação oficial, o Chef (*Progress Chef*, 2025) é uma ferramenta de gerenciamento de configuração para automatizar a infraestrutura. Desenvolvido em *Ruby*, define o estado desejado dos recursos do sistema (tais como pacotes, serviços e arquivos) por meio

de receitas, que são *scripts* declarativos elaborados em uma DSL própria. Essas receitas são estruturadas em livros de receitas, que contêm configurações organizadas e reutilizáveis. O Chef opera em dois modos distintos: *chef-client/server*, em que os nós se conectam a um servidor central, e *chef-solo*, destinado à execução local e independente.

Segundo a (HASHICORP, 2025a), o Terraform, desenvolvido pela HashiCorp, é utilizado no provisionamento de infraestrutura em ambientes de computação em nuvem. Sua capacidade de automação e escalabilidade é otimizada em plataformas como AWS, Microsoft Azure, GCP, Oracle Cloud, IBM Cloud, Alibaba Cloud, bem como em serviços como Cloudflare, GitHub, Kubernetes e Datadog, devido ao suporte nativo e às APIs.

A ferramenta utiliza HCL, uma linguagem declarativa que permite definir a infraestrutura de forma reproduzível, segura e versionável.

Em contraste, o uso do Terraform em infraestruturas locais, apesar de viável com provedores como VMware vSphere, Libvirt, Proxmox ou VirtualBox, é mais limitado. Isso se deve à dependência de suporte específico e à necessidade de integração com outras ferramentas, como Ansible ou Chef, para a configuração detalhada de sistemas operacionais. Assim, embora possa ser empregado em ambientes *on-premise*, não oferece as mesmas vantagens de automação e escalabilidade que em ambientes de nuvem.

Por fim, segundo a documentação oficial (Salt Project, 2025), o SaltStack é uma ferramenta de automação, execução remota e orquestração projetada para gerenciar sistemas em larga escala. Desenvolvido em Python, o *framework* é predominantemente declarativo, utilizando arquivos de definição em YAML com suporte a templates do Jinja2, embora também permita instruções imperativas para tarefas específicas.

Sua arquitetura cliente-servidor é composta por um *master* que controla múltiplos *minions*, garantindo uma comunicação eficiente. Essa comunicação é estabelecida por meio de ZeroMQ, com suporte a RAET e comunicação via SSH (*salt-ssh*).

2.5.2 Justificativa da escolha do Ansible

Neste trabalho, serão enfatizadas as ferramentas utilizadas para automação; será apresentado o Ansible, um *software* que permite automatizar tarefas repetitivas do dia a dia, como configuração, implantação e gerenciamento de infraestrutura; ele foi desenvolvido por Michael DeHaan e, atualmente, a detentora dos direitos é a empresa Red Hat.

A escolha do Ansible como ferramenta de automação de infraestrutura é amplamente justificada pela literatura científica recente e pela vasta adoção em ambientes profissionais que destaca sua simplicidade, facilidade de uso e arquitetura sem agente.

Simplicidade e curva de aprendizado: ao contrário de ferramentas como *Puppet* e *Chef*, que demandam conhecimento de linguagens específicas (Ruby ou DSLs), o *Ansible* emprega



arquivos YAML, uma linguagem declarativa, simples e legível. (PALMA; NUCCI; TAMBURRI, 2020) evidenciam que essa simplicidade é um dos principais fatores responsáveis por sua popularidade entre profissionais com diferentes níveis de experiência.

Arquitetura sem agente: diferentemente de *Puppet e Chef*, que exigem a instalação de agentes nos nós gerenciados, o *Ansible* opera via SSH, eliminando a necessidade de instalar software adicional nas máquinas remotas. Isso resulta em menor complexidade de implantação e em uma superfície de ataque reduzida, promovendo ambientes mais seguros, segundo (RAHMAN; MAHDAVI-HEZAVEH; WILLIAMS, 2019).

Ampla adoção no cenário profissional: Um estudo recente de (CARREIRA et al., 2025) analisou mais de 59 mil postagens em fóruns (*Stack Overflow e Reddit*) e realizou entrevistas com usuários de *Ansible*, confirmando seu uso generalizado e identificando desafios reais, como a depuração e a estruturação de código. Essa presença consolidada no mercado técnico-profissional demonstra a maturidade e relevância prática da ferramenta.

Flexibilidade e integração: Embora o *Terraform* seja mais adequado para o provisionamento de infraestrutura na nuvem, o *Ansible* se destaca na configuração de sistemas, na orquestração de serviços e na implantação de aplicações. Em ambientes modernos, ambas as ferramentas são frequentemente complementares: o *Terraform* realiza o provisionamento e o *Ansible* a pós-configuração.

Suporte à qualidade de código: A biblioteca *AnsibleMetrics*, desenvolvida por (PALMA; NUCCI; TAMBURRI, 2020), fortalece o ecossistema do *Ansible* ao permitir a medição da qualidade de *scripts* por meio de métricas estáticas, uma funcionalidade ainda rara em outras ferramentas *IaC*.

Com base nos estudos de (PALMA; NUCCI; TAMBURRI, 2020), (RAHMAN; MAHDAVI-HEZAVEH; WILLIAMS, 2019) e (CARREIRA et al., 2025), conclui-se que o *Ansible* é uma escolha tecnicamente sólida e cientificamente embasada para projetos de automação de infraestrutura que demandam:

- Facilidade de adoção;
- Baixa complexidade de implantação;
- Documentação acessível e comunidade ativa;
- Suporte a métricas de qualidade de código;
- Aplicação eficiente na automação de sistemas, serviços e aplicações.

Este capítulo abordou os fundamentos da abordagem (*IaC*), bem como uma análise comparativa das principais ferramentas de automação disponíveis, com ênfase em suas características

técnicas e aplicabilidade em distintos contextos. A partir dessa análise, justificou-se a seleção do *Ansible* como ferramenta central deste estudo, devido à sua simplicidade, arquitetura sem agente, linguagem acessível e ampla adoção tanto em ambientes profissionais quanto acadêmicos.

Essa fundamentação teórica estabelece as bases para a etapa prática da pesquisa. No capítulo subsequente, serão apresentados alguns trabalhos relacionados que abordam a automação de infraestrutura, a eficiência energética e a sustentabilidade em data centers.



3 TRABALHOS RELACIONADOS

A literatura recente tem explorado o papel das metodologias *DevOps* e da automação de infraestrutura no contexto da sustentabilidade computacional, com foco na redução do consumo de energia e das emissões de carbono em ambientes de TI. Entre os trabalhos de maior relevância, destacam-se iniciativas que buscam integrar práticas ambientais ao ciclo de desenvolvimento e de operação de software, bem como mensurar a eficiência e o impacto energético das soluções automatizadas.

O artigo “*Green DevOps: A Strategic Framework for Sustainable Software Development*” (AILANE; RUBNER; RAUSCH, 2025) propõe a integração de práticas de sustentabilidade em todas as fases do ciclo *DevOps*, com base em princípios da ACV. O *framework* define diretrizes para as etapas de planejamento, design, implementação e operação, destacando métricas como SCI e consumo energético por função. No estudo de caso realizado na Siemens, o modelo foi aplicado para incorporar métricas ambientais às *pipelines* de CI/CD, promovendo a conscientização e a padronização de práticas sustentáveis. Entre as discussões, os autores apontam que linguagens como *C*, *Go* e *Rust* apresentam maior eficiência energética, recomendando seu uso em tarefas críticas de processamento. O trabalho reforça a necessidade de automatizar a coleta de métricas e integrar indicadores ambientais diretamente às rotinas *DevOps*. As principais contribuições do estudo concentram-se em três eixos: diretrizes de sustentabilidade adaptadas a cada fase do ciclo *DevOps*, um estudo de caso detalhado em ambiente corporativo e a identificação de desafios em aberto, como a falta de métricas padronizadas e de dados de energia em tempo real. Os resultados demonstraram que a incorporação de métricas ambientais nas rotinas automatizadas promove maior conscientização e otimização de recursos, embora persistam limitações práticas quanto à mensuração automatizada de consumo energético e à ausência de padronização global de indicadores.

Em linha semelhante, o artigo “*Sustainable DevOps: Minimizing the Carbon Footprint of Banking Data Centers*” (MUPPALA, 2025) examina o impacto ambiental de centros de dados do setor financeiro e propõe práticas de automação sustentável. O estudo emprega uma abordagem de métodos mistos combinando entrevistas, estudos de caso e métricas de consumo pré e pós-implementação, para avaliar estratégias como containerização, microsserviços, automação com *Jenkins* e *Ansible*, monitoramento energético e migração para nuvens alimentadas por fontes renováveis. Os autores apontam que o provisionamento excessivo de recursos, as elevadas demandas de refrigeração e as políticas de redundância são fatores que ampliam a pegada de carbono dos *data centers*, e sugerem que a automação e a otimização de cargas de trabalho podem contribuir significativamente para mitigar tais impactos.

Em uma perspectiva mais voltada à mensuração da eficiência de automações, o artigo “*AnsibleMetrics: A Python Library for Measuring Infrastructure-as-Code Blueprints in Ansible*” (PALMA; NUCCI; TAMBURRI, 2020) apresenta uma ferramenta híbrida, desenvolvida em



Python, que pode ser utilizada tanto na linha de comando quanto como módulo importável. A proposta visa extrair métricas estáticas de scripts *Ansible*, permitindo avaliar a complexidade, a manutenibilidade e a qualidade estrutural de configurações de IaC. O *AnsibleMetrics* gera relatórios automáticos com indicadores, como o número de *plays*, tarefas, variáveis e entropia textual, o que auxilia na análise quantitativa de automações e na identificação de padrões que impactam o desempenho. Embora não seja voltado especificamente à análise energética, o trabalho fornece uma base objetiva para avaliar a eficiência e as boas práticas em scripts de infraestrutura, podendo ser integrado a *pipelines DevOps* e a estudos sobre sustentabilidade computacional.

Nos últimos anos, têm surgido pesquisas voltadas à predição de consumo energético em infraestruturas de alto desempenho, utilizando técnicas de aprendizado de máquina. O estudo de (ZHANG; LIU, 2022) propõe um modelo de previsão de consumo total de energia em HPC, integrando variáveis como temperatura, carga de CPU e uso de memória para estimar a eficiência global dos *data centers*. Da mesma forma, (CHATLATANAGULCHAI; CHANTRAPORNCHAI, 2024) apresentam um estudo de caso em uma universidade tailandesa, no qual foram coletados e analisados dados reais de consumo de energia para subsidiar a modelagem preditiva e otimizar a operação do *data center*. Esses trabalhos evidenciam o potencial da inteligência artificial para aprimorar a eficiência energética e reduzir as emissões em operações contínuas, embora ainda sejam escassas as pesquisas que aplicam tais técnicas em contextos públicos e educacionais.

No contexto brasileiro, destaca-se o estudo de (LUNARDI; SIMÕES; FRIO, 2014), intitulado “TI Verde: uma análise dos principais benefícios e práticas utilizadas pelas organizações”, publicado na REAd da UFRGS. O trabalho realiza uma análise exploratória-descritiva das práticas sustentáveis de Tecnologia da Informação em organizações brasileiras, com base em aproximadamente duzentas fontes publicadas entre 2006 e 2011, abrangendo desde artigos científicos até reportagens técnicas. Mais de 60% das referências são nacionais, envolvendo empresas como Ambev, USP, Alog e Itaotec, além de veículos especializados como *Computerworld*, *Info Corporate* e *ITWeb Brasil*. As práticas identificadas incluem virtualização de servidores, descarte e reaproveitamento de equipamentos, uso racional de energia e promoção de uma cultura organizacional voltada à sustentabilidade. Entre os benefícios apontados, destacam-se a economia de energia, a racionalização de insumos e a redução de resíduos eletrônicos. Apesar dos avanços, os autores destacam que a literatura acadêmica brasileira ainda é incipiente quanto à mensuração de impactos ambientais em processos de TI, o que converge com as análises de (BROOKS; WANG; SARKER, 2010) e de (WATSON; BOUDREAU; CHEN, 2010), que enfatizam a necessidade de abordagens mais técnicas e mensuráveis para a sustentabilidade computacional.

Apesar de sua relevância histórica, esses trabalhos refletem um período inicial das discussões sobre sustentabilidade em Tecnologia da Informação, anterior à consolidação de práticas como *DevOps*, *Infrastructure as Code* (IaC) e automação em larga escala. Estudos mais recentes têm buscado suprir essas limitações ao incorporar métricas operacionais, medições



de consumo energético e avaliações práticas em ambientes reais, evidenciando a necessidade de abordagens experimentais que integrem automação, eficiência energética e sustentabilidade computacional.

Complementarmente, (ARRUDA et al., 2025) conduziram a revisão sistemática, na qual analisaram o papel das tecnologias da Indústria 4.0 como IoT, Big Data, *Machine Learning* e Inteligência Artificial na promoção de processos sustentáveis em ambientes corporativos. O estudo identificou que tais tecnologias permitem o monitoramento e o controle automatizado de processos, favorecendo a eficiência energética e a redução de resíduos. Entretanto, os autores apontam que a maioria das pesquisas ainda apresenta caráter conceitual, sem experimentação empírica nem mensuração de emissões ou de consumo energético em ambientes institucionais. Essa constatação reforça a carência de estudos práticos que associem automação, eficiência e sustentabilidade em contextos públicos, lacuna que o presente trabalho busca suprir.

Assim, ao considerar estudos nacionais e internacionais, observa-se que a literatura tem avançado na formulação de *frameworks* conceituais e na incorporação gradual de métricas ambientais às práticas de *DevOps* e *TI Verde*, mas ainda carece de experimentos aplicados em ambientes reais. Nesse contexto, o presente estudo propõe-se a contribuir por meio de uma abordagem prática voltada à automação sustentável de infraestrutura em uma universidade pública, utilizando princípios de *DevOps* e de IaC para apoiar a eficiência energética e a redução de emissões.

Dessa forma, este capítulo evidencia que, embora a literatura apresente avanços significativos na proposição de *frameworks* conceituais e na incorporação gradual de métricas ambientais às práticas de *DevOps* e *TI Verde*, ainda persiste uma lacuna quanto a estudos práticos aplicados a ambientes públicos e educacionais. O presente trabalho insere-se nesse contexto ao propor e avaliar uma abordagem prática de automação sustentável de infraestrutura em uma universidade pública, contribuindo para a consolidação de métodos mensuráveis e replicáveis voltados à eficiência energética e à redução de emissões.

Com isso, este capítulo consolida o panorama teórico que sustenta a pesquisa, apresentando as bases conceituais e empíricas que orientam o desenvolvimento do modelo proposto. No capítulo seguinte, apresenta-se a *Revisão Bibliográfica de Literatura*, que descreve o protocolo de busca, os critérios de seleção e o mapeamento conceitual das produções científicas que fundamentam as decisões metodológicas e experimentais deste trabalho.



4 METODOLOGIA

Este capítulo apresenta o percurso metodológico adotado para investigar, em ambiente real de uma universidade pública, como a automação de rotinas de infraestrutura (com ênfase em práticas de DevOps e em Infraestrutura como Código (IaC)) pode reduzir o consumo de energia, as emissões de CO₂ e o tempo de operação em um data center. A revisão da literatura, descrita no Capítulo 5, fundamentou as escolhas do desenho do estudo e a seleção das métricas analisadas.

4.1 Enquadramento metodológico

A pesquisa situa-se no **paradigma positivista**, com foco na mensuração objetiva e na verificação empírica. Quanto à natureza, trata-se de um estudo aplicado, voltado a resolver um problema concreto de eficiência energética na operação de data center. Em termos de objetivo, assume caráter **descritivo-avaliativo**: descreve procedimentos, coleta dados e avalia os efeitos da automação sobre indicadores operacionais e ambientais.

O delineamento é quase experimental. As mesmas rotinas foram executadas de duas formas (*manual e automatizada*) no ambiente institucional, sob janelas de operação controladas e com procedimentos padronizados. Não houve isolamento completo de todas as variáveis de um data center em funcionamento contínuo; por isso, empregaram-se controles e repetição de rodadas para mitigar interferências e reforçar a validade dos achados.

A abordagem é quantitativa. As variáveis principais são: tempo total de execução das tarefas; energia elétrica consumida nas janelas de interesse; e emissões estimadas de CO₂ a partir do consumo de energia elétrica medido. Os dados foram registrados, organizados e comparados entre os dois modos de execução.

4.2 Revisão da literatura

A revisão da literatura teve como propósito identificar práticas, estratégias e evidências científicas relacionadas à eficiência energética, à sustentabilidade e à adoção de tecnologias sustentáveis em *Data Centers*. O levantamento bibliográfico concentrou-se em compreender como abordagens baseadas em automação e em Infraestrutura como Código (IaC) podem contribuir para o uso racional de energia e para a redução da pegada de carbono em ambientes computacionais.

O processo de seleção foi orientado por três questões norteadoras: (i) identificar estratégias para melhoria da eficiência energética em *Data Centers*; (ii) compreender o impacto ambiental dessas infraestruturas; e (iii) analisar a necessidade e os benefícios da adoção de modelos de *data centers* verdes (Tabela 6). Essas questões direcionaram a análise crítica dos estudos e auxiliaram na identificação de lacunas e de oportunidades de pesquisa.



Os resultados dessa revisão forneceram a base conceitual para a formulação da metodologia e para a discussão dos resultados apresentados nos capítulos seguintes, reforçando a relevância de iniciativas que promovam a eficiência energética e a sustentabilidade em ambientes computacionais.

4.3 Contexto do estudo

O estudo foi conduzido em um *data center* do tipo contêiner, com climatização de precisão, alimentação elétrica autônoma e rede corporativa. Para representar a operação cotidiana com o menor viés possível, utilizou-se um arranjo de teste composto por: (i) um servidor de orquestração com Linux e Ansible; (ii) nós virtuais-alvo com sistemas estáveis e perfis de carga conhecidos; e (iii) o sistema institucional de monitoramento elétrico do contêiner, que registra grandezas em intervalos.

4.4 Procedimentos e coleta de dados

Selecionaram-se rotinas recorrentes na operação diária (atualizações, reinícios controlados de serviços, limpeza e rotação de registros e manutenção do sistema de periódicos). Cada rotina foi descrita em um roteiro claro e, em seguida, encapsulada em um *playbook* equivalente.

Em todas as execuções, registrou-se o tempo (início e término) e o consumo de energia na janela correspondente, por meio do subsistema de medição do contêiner (IMAMS). Esse sistema fornece o acumulado de energia por intervalo, associado temporalmente a cada execução. Leituras auxiliares de uso do processador e da memória foram realizadas apenas para verificar a estabilidade do ambiente, sem integrar à análise quantitativa.

4.5 Planejamento dos ensaios

As execuções foram programadas em períodos de menor interferência operacional, mantendo-se janelas comparáveis entre os modos manual e automatizado. Em rodadas sucessivas, alternou-se a ordem de execução para reduzir os efeitos transitórios (aquecimento do *hardware* e flutuações momentâneas de carga). Ocorrências externas potencialmente influentes foram registradas e, quando necessário, levaram ao descarte justificado da rodada.

4.6 Tratamento e análise

A energia consumida por rodada foi obtida pela variação do acumulado fornecido pelo IMAMS entre o início e o fim da janela. As emissões de CO₂ foram estimadas por conversão



direta dos kWh medidos, utilizando o fator médio de emissão do Sistema Interligado Nacional adotado nesta dissertação.

A comparação entre modos considerou, por tarefa, o tempo total e a energia consumida, com síntese por meio de medidas simples de tendência central e dispersão. A coerência dos resultados foi verificada pela estabilidade das leituras ao longo das repetições e pela compatibilidade entre as reduções de tempo e de energia, quando presentes.

4.7 Reprodutibilidade e registros

Para permitir repetição do estudo no mesmo ambiente, foram preservados: os *playbooks* efetivamente utilizados, arquivos de inventário, roteiros de execução (“passo a passo”), parâmetros operacionais, registros temporais de início e término e as saídas geradas. Dados sensíveis (chaves, endereços e credenciais) foram removidos das cópias e mantidos sob controle institucional. Planilhas com medições e scripts simples de cálculo (tempo, energia e conversão em CO₂) acompanham a documentação desta pesquisa.

4.8 Limitações e escopo

Duas limitações são assumidas: (i) o sistema de medição elétrica observa o contêiner em sua totalidade, e não cada nó isoladamente; para mitigar essa limitação, adotaram-se janelas exclusivas, repetição e controle de concorrência; (ii) as conclusões referem-se especificamente ao contexto institucional analisado. Ainda assim, os procedimentos descritos são gerais e podem ser adaptados a outros ambientes que possuam monitoramento elétrico equivalente.

Parte da revisão linguística e da adequação textual deste trabalho contou com o apoio da ferramenta *ChatGPT* (OPENAI, 2025), desenvolvida pela OpenAI. Seu uso teve caráter exclusivamente auxiliar, restrito à reformulação de trechos para aprimorar a clareza, a coesão e a correção gramatical, sem interferir na elaboração conceitual, metodológica ou analítica do conteúdo.

Com a metodologia delineada, o Capítulo 6 apresenta a solução proposta como um *projeto DevOps sustentável*, descrevendo o fluxo operacional, a organização dos *playbooks* e a integração com o sistema de medição que sustenta a comparação entre a execução manual e a automatizada.



5 REVISÃO BIBLIOGRÁFICA DE LITERATURA

5.1 Introdução da Revisão

A era atual é marcada pelo surgimento de novas tecnologias, como a IA, que transformam a sociedade em geral e são incorporadas aos mais diversos setores. E o aumento do uso da internet, aliado à ampla dependência dos *smartphones* como computadores de mão, levou a um crescimento exponencial no tráfego de dados. Esse fenômeno impulsionou a demanda por centros de processamento de dados operando 24 horas por dia, o que gerou a necessidade de discutir a eficiência energética em um contexto global.

Segundo (LYKOU; MENTZELIOTI; GRITZALIS, 2018), pesquisas indicam que os centros de processamento de dados são responsáveis por cerca de 2% do consumo global de energia elétrica e por aproximadamente 1% das emissões globais de gases de efeito estufa, evidenciando seu impacto ambiental. Ademais, o elevado consumo de água para o resfriamento dos equipamentos, bem como a utilização de minerais e metais na construção e operação dessas infraestruturas, agrava os efeitos ambientais associados à sua atividade.

A esfera computacional, que inclui infraestruturas de processamento de dados, laboratórios de informática, *desktops*, *notebooks*, sensores e equipamentos de IoT, também contribui para as emissões de carbono por meio do consumo de energia. Grandes instalações de processamento de dados demandam quantidades significativas de energia para operar continuamente, o que torna necessário promover estudos voltados à utilização eficiente dessa energia.

Segundo (IIZUKA; PEÇANHA,), a importância de eventos globais, como a COP, promovidos pelas Nações Unidas. Esses encontros reúnem nações para colaborar em estratégias que respondam às crescentes demandas energéticas, reafirmando o compromisso com um futuro sustentável e enfrentando as complexidades do cenário energético atual.

O Brasil, como país em desenvolvimento, desempenha um papel significativo neste cenário. Com uma das maiores florestas do mundo e uma vasta rede de recursos hídricos, o país está em uma posição vantajosa para aproveitar fontes de energia limpa, como a hidrelétrica. Recentemente, há um esforço concentrado para diversificar o portfólio energético do país, investindo em fontes como a energia eólica e solar, que se alinham às condições climáticas favoráveis.

Segundo a Agência Nacional de Energia Elétrica (ANEEL), o Brasil teve um crescimento exponencial de 5,1 gigawatts (GW) na capacidade de geração de energia em 2023, contribuindo para um total de 193,9 GW disponíveis, (ELÉTRICA, 2023). Entraram em operação 160 novas usinas, o que reflete um avanço significativo. No entanto, no contexto da eficiência energética (CAMPOS, 2022), o Brasil ainda avança de forma moderada, ocupando a 20ª posição em um ranking de 25 países, conforme relatório de 2022 sobre indicadores de eficiência energética.



Este artigo realiza uma análise sobre a eficiência energética e sustentabilidade em infraestruturas de centros de processamento de dados. São exploradas melhores práticas baseadas em diversas fontes, avaliando criticamente o impacto ambiental dessas instalações. Também são investigadas razões para a adoção de práticas sustentáveis, conhecidas como TI Verde.

5.2 Metodologia

Em termos metodológicos, a pesquisa caracteriza-se como bibliográfica, de natureza qualitativa, fundamentada na análise de artigos científicos que abordam estratégias para eficiência energética em *data centers*, os impactos ambientais dessas infraestruturas e a adoção de práticas sustentáveis. Os trabalhos analisados foram selecionados com base em critérios de relevância, atualidade e contribuição para a compreensão do tema.

A pesquisa buscou explorar e sistematizar as principais estratégias propostas na literatura para a redução do consumo de energia, a avaliação dos impactos ambientais dos *data centers* e a justificativa da necessidade de abordagens sustentáveis. Dessa forma, a revisão bibliográfica possibilita uma visão ampla e fundamentada sobre o tema, destacando desafios, tendências e recomendações para a adoção de soluções mais eficientes e ambientalmente responsáveis.

5.2.1 Estratégias de busca

Nesta revisão bibliográfica, foi utilizada a ferramenta online *Parsifal*¹ para a filtragem e extração de dados, a fim de elaborar o protocolo de revisão, o que possibilitou a coleta de dados provenientes de três bases de dados distintas: *Google Scholar*, *Scopus* e Periódicos da Capes. Esses dados foram extraídos por meio de uma busca realizada com as strings relacionadas na Tabela 2, restringindo ao período de 2018 a 2023, totalizando um intervalo de cinco anos.

String de busca utilizada para nas bases Capes e Google Scholar:

String de busca utilizada nas bases Capes e Google Scholar:

(“*energy efficiency*” OR “*sustainability*”) AND (“*data centers*”) AND (“*carbon*”) AND (“*best practices*”)

String de busca utilizada na base Scopus:

(TITLE-ABS-KEY(“*energy efficiency*”) OR TITLE-ABS-KEY(“*sustainability*”)) AND TITLE-ABS-KEY(“*data centers*”) AND TITLE-ABS-KEY(“*carbon*”) AND TITLE-ABS-KEY(“*best practices*”)

As palavras-chave empregadas foram:

¹ <https://parsif.al/about/>: é uma ferramenta online desenvolvida para apoiar pesquisadores na realização de revisões.

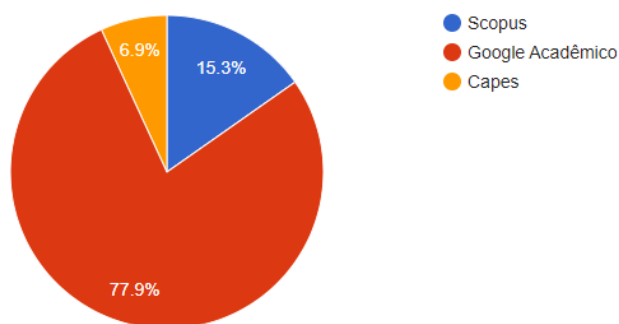


Tabela 2 – Palavras-chave e seus sinônimos

Palavra-chave	Sinônimo
Eficiência Energética	<i>Energy Efficiency</i>
Sustentabilidade	<i>Sustainability</i>
Centro de Armazenamento de Dados	<i>Data Centers</i>
Carbono	<i>Carbon</i>
Melhores Práticas	<i>Best Practices</i>

Fonte: Elaborado pela autora.

Para extração dos dados do *Google Scholar* foi utilizado o *software Publish or Perish*². A Figura 1 ilustra a porcentagem dos resultados extraídos de cada plataforma.

Figura 1 – Porcentagem dos resultados extraídos de cada plataforma de base de dados.

Fonte: Extraído do Parsifal

5.2.2 Seleção dos artigos:

Após as buscas, os trabalhos encontrados foram baixados e enviados ao Parsifal, onde foi possível catalogar os resultados. Em seguida, verificou-se a existência de artigos duplicados e aplicaram-se filtros com base nos critérios de inclusão (CI) e exclusão (CE) para determinar quais trabalhos estão alinhados ao tema abordado. Posteriormente, realizou-se a leitura completa para identificar quais artigos respondiam aos questionamentos levantados.

Foram aplicados os seguintes critérios de exclusão(CE) e inclusão(CI) relacionados na Tabela 3:

² <https://harzing.com/resources/publish-or-perish/>: é um software de desktop gratuito que extrai dados do Google Scholar e de outras plataformas para ajudar pesquisadores.

Tabela 3 – Critérios de Inclusão (CI) e Exclusão (CE) utilizados na seleção de artigos para o estudo

Critérios de Inclusão (CI)	
CI1	Estudos que abordem eficiência energética e práticas sustentáveis
CI2	Publicações que estejam disponíveis integralmente
CI3	Publicações dos últimos 5 anos
Critérios de Exclusão (CE)	
CE1	Estudos que discutem TI verde sem data center e energia
CE2	Estudos que tratam de tecnologias obsoletas
CE3	Publicações que não possuem o texto completo disponível
CE4	Trabalhos que não apresentam dados técnicos, estudos de caso ou validação científica

Fonte: Elaborado pela autora.

5.2.3 Extração de Dados

Na realização da busca, obteve-se o retorno de 131 artigos catalogados na ferramenta online *Parsifal*. Destes, 9 eram duplicados, sendo excluídos, enquanto 10 foram selecionados para serem utilizados na revisão sistemática, após a aplicação dos critérios de CI e CE, conforme apresentado na Tabela 4 abaixo:

Tabela 4 – Quantitativo de artigos encontrados, duplicados, excluídos e incluídos por base de dados

Base de dados	Saídas	Duplicadas	Excluídas	Incluídas
Scopus	20	4	16	0
Capes	9	1	6	1
Google Acadêmico	102	3	90	9
Total	131	9	112	10

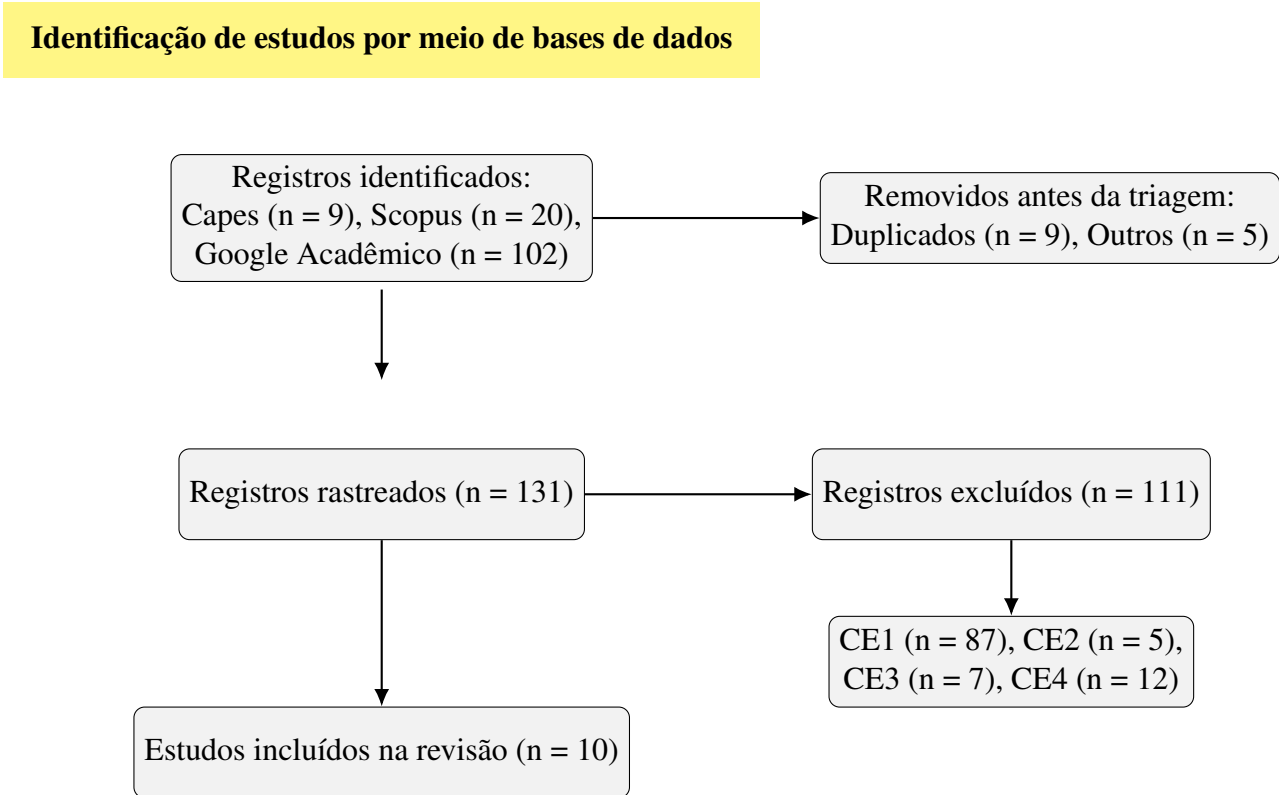
Fonte: Elaborado pela autora.

Na etapa de identificação, conforme ilustra a Figura 2, o fluxograma apresenta o processo completo de identificação, triagem, exclusão e inclusão de registros durante a revisão sistemática, detalhando a quantidade de registros em cada etapa e os critérios aplicados para exclusão e inclusão. Os registros foram categorizados da seguinte forma: na base da Capes, foram identificados 9 registros; na base Scopus, 20 registros; e no Google Acadêmico, 102 registros. Destes, foram removidos 9 duplicados e 5 registros fora do período estipulado.

Durante a fase de triagem, 131 registros foram submetidos a uma análise mais detalhada. Desses, 111 foram excluídos conforme os critérios de exclusão apresentados na Tabela 3. No critério CE1, foram excluídos 87 artigos; no CE2, 5 artigos foram descartados; no CE3, 7 artigos não foram considerados; e, finalmente, no CE4, 12 artigos foram removidos. Como resultado, 10 artigos foram selecionados e incorporados à revisão bibliográfica abaixo; na Tabela 5 estão apresentados os artigos utilizados. Todo o processo foi realizado na ferramenta *on-line Parsifal*.



Figura 2 – Fluxograma das fases de mapeamento realizadas



Fonte: Elaborado pela autora.

Tabela 5 – Lista de artigos incluídos na revisão sistemática, indicando os autores e as referências correspondentes

ID	Trabalhos Utilizados
A1	(LYKOU; MENTZELIOTI; GRITZALIS, 2018)
A2	(DUMITRESCU et al., 2018)
A3	(JNR; MAJID; ROMLI, 2018)
A4	(KOTOWSKI; OKO et al., 2018)
A5	(BASMADJIAN, 2019)
A6	(MASANET; LEI, 2020)
A7	(PATTERSON et al., 2022)
A8	(MUPPALA, 2025)
A9	(XU; BUYYA, 2019)
A10	(HEIKKINEN, 2023)

Fonte: Elaborado pela autora.

5.2.4 Questionamentos Levantados

Os questionamentos apresentados na Tabela 6 visam facilitar a compreensão do assunto abordado. Quando se trata de eficiência energética, geralmente são encontradas abordagens em diversos setores, e o setor computacional deve estar incluído nas metas mundiais de redução de dióxido de carbono.



Tabela 6 – Questões de pesquisa e suas motivações

Questão de Pesquisa	Motivação
Q1: Estratégias – Quais são as estratégias que podem ser adotadas para eficiência energética em <i>Data Centers</i> ?	A resposta é motivada pela necessidade de abordar a importância da eficiência energética em data centers e pela busca por estratégias eficazes para alcançá-la.
Q2: Impacto – Qual o impacto ambiental dos <i>Data Centers</i> ?	A resposta à pergunta revela a importância de compreender e mitigar os efeitos negativos que essas infraestruturas podem ter no meio ambiente.
Q3: Necessidade – Qual a necessidade de um <i>data center</i> verde?	A resposta à pergunta leva à compreensão abrangente de por que é indispensável adotar práticas sustentáveis na construção e na operação de data centers.

Fonte: Elaborado pela autora.

Isso se deve ao fato de que equipamentos, *data centers*, laboratórios e infraestrutura em geral demandam significativamente energia elétrica. Por exemplo, esses centros requerem energia 24 horas por dia. Portanto, os seguintes questionamentos são apresentados para aprimorar a compreensão do tema:

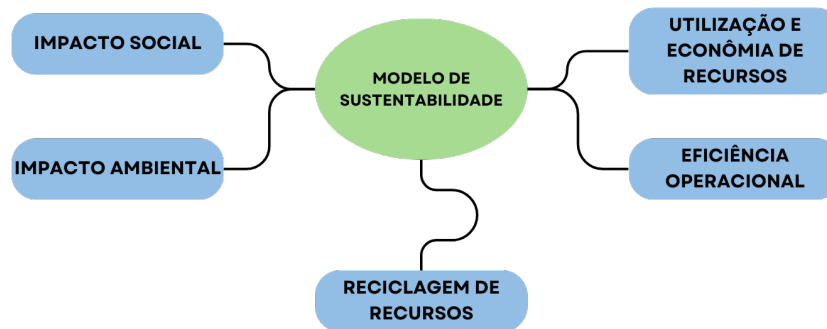
5.2.5 Análise dos Resultados

Q1- Quais as estratégias que podem ser adotadas para eficiência energética em *Data Centers*? Estudos de caso a seguir demonstram, com base empírica, a efetividade das práticas propostas para aprimorar a eficiência energética em data centers.

Por meio de 5 elementos : Eficiência energética e operacional, energia verde, reciclagem e impacto social.

Para os autores G Lykou, D Mentzelioti e D Gritzalis, apresentam uma proposta que visa avaliar a viabilidade de tornar os data centers mais sustentáveis por meio de cinco principais elementos, cada um constituído por uma variedade de fatores que influenciam essa sustentabilidade (LYKOU; MENTZELIOTI; GRITZALIS, 2018). Esses elementos-chave incluem eficiência energética, eficiência operacional, adoção de fontes de energia renováveis, reciclagem de recursos e impacto social. Eles são influenciados por diversos fatores, como o consumo e reaproveitamento de energia proveniente de fontes renováveis, a utilização de materiais reciclados e a gestão adequada dos recursos. A Figura 3 ilustra esses cinco elementos essenciais à eficiência energética, eficiência operacional, adoção de energia renovável, reciclagem de recursos e impacto social, detalhando como cada fator contribui para a avaliação abrangente da sustentabilidade dos *data centers*.



Figura 3 – Representação dos cinco elementos essenciais para a sustentabilidade em data centers.

Fonte: (LYKOU; MENTZELIOTI; GRITZALIS, 2018)

Os fatores foram analisados e pontuados em 5 elementos, resultando em uma avaliação abrangente da sustentabilidade. Seguem abaixo os fatores que influenciaram os elementos:

1. Impacto Ambiental: ao considerar o impacto ambiental, propõe-se o uso de energia verde e a adoção de políticas de reciclagem;
2. Utilização de recursos e economicidade: avalia a eficiência operacional com o uso eficiente de recursos como energia, água e materiais, bem como prezando pela economicidade dos custos;
3. Eficiência operacional: avalia a eficiência das operações, incluindo equipamentos eficientes, segurança, continuidade do negócio, gestão de manutenção e a gestão de energia;
4. Reciclagem de recursos: a capacidade de reciclar recursos, como resíduos de água, e uma política de reciclagem eficaz;
5. Impacto social: análise do impacto social, responsabilidade social corporativa, saúde e segurança operacional dos funcionários, bem como iniciativas e incentivos ao uso de transportes verdes, como o estímulo ao desenvolvimento de ônibus elétricos, o uso de veículos elétricos e/ou a construção de ciclovias.

Por meio de adoção de práticas de *Green DevOps*

De acordo com Muppala (MUPPALA, 2025), a adoção de práticas sustentáveis de DevOps em centros de dados bancários resultou em reduções significativas no consumo de energia. Entre as estratégias avaliadas, a adoção de microsserviços e containerização reduziu o uso mensal de energia de 1.200 kWh para 800 kWh (uma economia de 33,33%), e a automação de processos alcançou uma redução semelhante. O monitoramento avançado com ferramentas de análise em tempo real reduziu o consumo de 1.500 kWh para 900 kWh (uma economia de 40%).

A migração para serviços em nuvem alimentados por energia renovável resultou na maior economia, reduzindo o consumo de 1.500 kWh para 800 kWh (46,67%) e as emissões de CO₂ de 1.000 kg para 400 kg.

Esses resultados estão associados às características técnicas de cada solução: *contêineres* e microsserviços promovem a alocação sob demanda e reduzem a ociosidade dos servidores; ferramentas de automação, como Jenkins e Ansible, desativam dinamicamente recursos inativos em períodos de baixa demanda; o monitoramento contínuo com análise preditiva permite o desligamento inteligente de equipamentos subutilizados; e a computação em nuvem com fontes renováveis aproveita uma infraestrutura eficiente baseada em energia solar e eólica.

A pesquisa empregou uma abordagem metodológica mista, combinando a análise de métricas energéticas com dados qualitativos coletados por meio de questionários aplicados a profissionais de TI, equipes de DevOps e gerentes de data centers.

Os instrumentos tinham como objetivo avaliar as percepções sobre a eficácia das práticas adotadas, os obstáculos enfrentados e o impacto na eficiência operacional. Embora o número exato de respondentes não tenha sido especificado, os dados contribuíram para validar a eficácia das intervenções no contexto prático do setor bancário.

Assim, o estudo de caso apresentado por Muppala et al. (MUPPALA, 2025) demonstra como práticas sustentáveis de DevOps podem ser estrategicamente implementadas para melhorar a eficiência energética em ambientes computacionais críticos, como data centers bancários.

Por meio de otimização de energia:

Em um estudo de caso conduzido por Basmadjian *et al.* (BASMADJIAN, 2019), realizado em um *data center* da HPE localizado em Milão, foi implementado um sistema de otimização energética baseado em predição de carga, orquestração e controle inteligente de recursos. A abordagem incluiu ações como o desligamento de servidores virtuais ociosos e a migração de cargas de trabalho entre servidores ou entre *data centers*. Os resultados foram expressivos: a economia de energia em picos chegou a **50%**, enquanto a **redução total de consumo atingiu 25%**, evidenciando a efetividade da abordagem para melhorar a sustentabilidade em *data centers*. A Figura 4 ilustra o esquema da arquitetura utilizada no experimento:

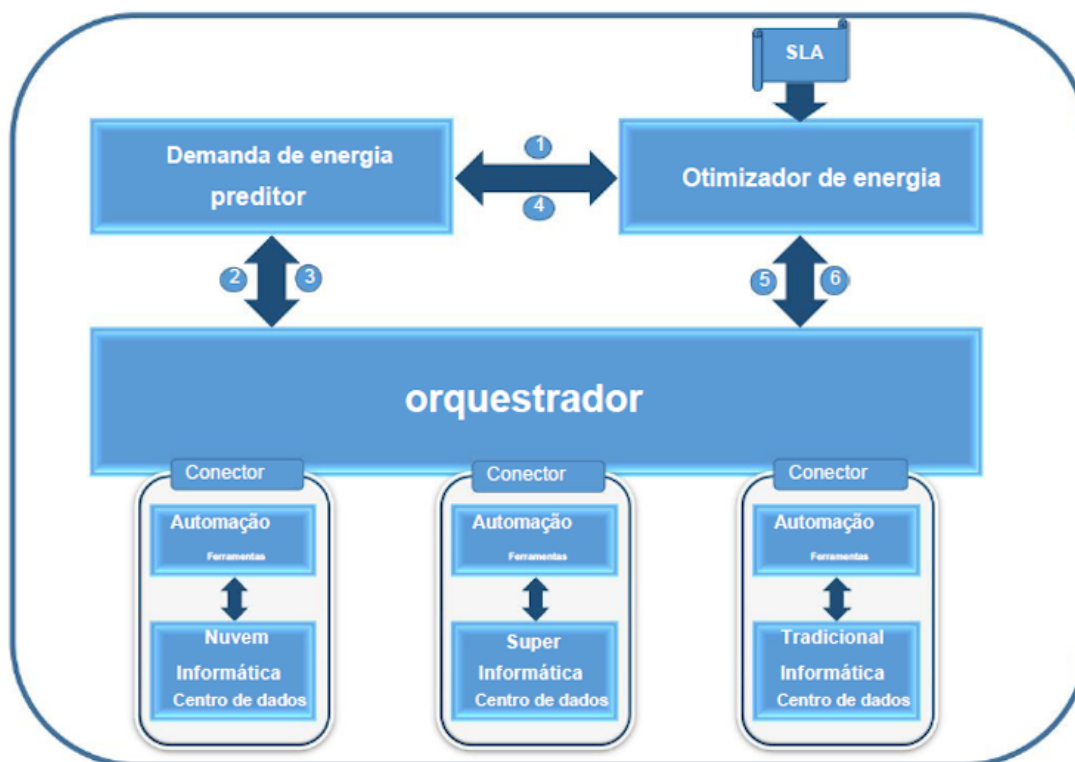
Otimizador: é um componente de *software* que usa as previsões para tomar as decisões. ex: ligar e desligar servidores, migrar máquinas virtuais de um servidor hospedeiro para outro ou alterar o uso de recursos como o uso da CPU e memória;

Preditor: é um *software* que usa modelos matemáticos para estimar a demanda de energia;

Orquestrador: o sistema implementa as decisões do otimizador e também pode coordenar, monitorar e relatar informações ao otimizador. Pode ser um sistema autônomo ou integrar-se a um sistema de gerenciamento.



Figura 4 – Sistema de gerenciamento de energia.



Fonte: (BASMADJIAN, 2019)

No esquema da Figura 4, o otimizador precisa da previsão da demanda de energia que será necessária. Então, o otimizador consulta o preditor de demanda de energia para obter previsões. O preditor obtém as informações dinamicamente, as combina e calcula as previsões. Em seguida, o otimizador recebe essas informações para tomar as decisões, enviadas para o orquestrador e por ele são executadas. O sistema é utilizado para reduzir o consumo de energia dentro do *data center* e como ele colabora para ajustar a carga de energia conforme a demanda, maximizando a eficiência energética e minimizando o desperdício.

Por meio de uso de contêineres:

Segundo os autores Minxian *et al.* (XU; BUYYA, 2019), o uso de técnicas baseadas em contêineres, como o *brownout*, é uma alternativa para melhorar a eficiência energética em *data centers* em nuvem. Essa abordagem permite a desativação dinâmica de componentes opcionais de aplicações durante picos de demanda, reduzindo o consumo de energia sem comprometer significativamente a qualidade do serviço.

Os experimentos foram conduzidos em ambiente real, utilizando a infraestrutura do Grid'5000, com servidores físicos monitorados por APIs de consumo de energia. O sistema proposto demonstrou capacidade de economizar entre 10% e 40% de energia em relação aos algoritmos de base, como o NPA e o HS, mantendo os níveis aceitáveis de tempo médio de resposta e violações de SLA.

Conjunto de medidas:

No artigo (DUMITRESCU et al., 2018), os autores propõem estratégias para aprimorar a eficiência energética em data centers, adotando uma abordagem abrangente que engloba diversas áreas. Essas estratégias visam não apenas reduzir o consumo de energia, mas também otimizar a gestão operacional. Algumas das recomendações incluem:

- Otimização de *Hardware* e Gestão de Ar: recomenda-se buscar eficiência energética por meio da atualização de *hardware*, bem como aprimorar a gestão do fluxo de ar e otimizar os sistemas de resfriamento. Isso pode envolver soluções de resfriamento de alta eficiência, como economizadores, *free-cooling* ou reaproveitamento de calor;
- Controle de Temperatura: monitorar e controlar a temperatura é fundamental. Recomenda-se ajustar os pontos de temperatura em direção aos limites superiores recomendados para alcançar o equilíbrio entre a eficiência energética e a manutenção das condições ideais para os equipamentos.
- Consolidação e Virtualização: transformar servidores físicos em servidores virtuais através da consolidação é outra medida recomendada. Isso não apenas economiza energia, mas também otimiza o uso dos recursos de *hardware*.
- Gestão de Energia: melhorar o gerenciamento de energia dos servidores evita o desperdício de energia durante períodos de ociosidade. A integração de sistemas com o *software* GIDC auxilia no monitoramento e no controle eficientes.
- Desativação Seletiva: desligar componentes de infraestrutura não utilizados e ativar *backups* somente quando necessário contribuem para reduzir o consumo energético.

O autor ilustra essas medidas (DUMITRESCU et al., 2018) por meio de um estudo de caso de um data center de médio porte localizado em Bucareste, em operação desde 2005. Nesse caso, a otimização envolveu a implementação das estratégias mencionadas acima, resultando em resultados positivos na redução do consumo de energia e na melhoria da eficiência operacional.

Além das práticas discutidas anteriormente, é fundamental adotar métricas padronizadas. Os indicadores comumente usados incluem a Eficácia no Uso de Energia (PUE), que mede a relação entre a energia total usada e a energia efetivamente utilizada pelos equipamentos de TI; a Eficácia no Uso de Água (WUE), que mede o volume de água usado por kWh de energia fornecida aos dispositivos de TI; e a Eficácia no Uso de Carbono (CUE), que associa as emissões de CO₂ ao consumo de energia dos sistemas de TI. Esses três indicadores constituem o “triângulo da sustentabilidade” dos data centers, oferecendo uma base sólida para o monitoramento contínuo e melhorias na eficiência energética, hídrica e de carbono (VERTIV, 2023; LYKOU; MENTZELIOTI; GRITZALIS, 2018).



Além disso, várias medidas práticas devem ser consideradas. Lykou et al. sugerem atualizar os sistemas de refrigeração e substituir os equipamentos por hardware mais eficiente em termos energéticos, enquanto Dumitrescu et al. enfatizam a importância do controle de temperatura, da consolidação de servidores, da virtualização e de sistemas de gerenciamento integrados. Prasanna et al. (PRASANNA; SINGH, 2023) recomendam a adoção de arquiteturas baseadas em contêineres e a implementação de técnicas de *brownout* para ajustar os recursos de computação de acordo com a demanda de forma dinâmica. Quando alinhadas às métricas de sustentabilidade padronizadas, essas práticas oferecem uma abordagem sólida e mensurável para a criação de data centers mais ecológicos e eficientes.

Q2- Qual o impacto ambiental dos Data Centers?

Emissão de dióxido de carbono:

Estima-se que os data centers sejam responsáveis por aproximadamente 2% do consumo global de eletricidade e cerca de 1% das emissões de GEE relacionadas à energia (LYKOU; MENTZELIOTI; GRITZALIS, 2018), (International Energy Agency, 2023). A Agência Internacional de Energia (AIE) relata que os data centers, juntamente com as redes de transmissão de dados, contribuem com cerca de 1% das emissões globais de GEE relacionadas à energia, o que equivale a aproximadamente 330 milhões de toneladas de CO₂ equivalente (MtCO₂e) em 2020. No mesmo ano, o consumo de eletricidade dessas infraestruturas variou de 240 a 340 TWh, o que corresponde a 1 a 1,3% da demanda global de eletricidade (International Energy Agency, 2023).

Embora essa porcentagem possa parecer pequena em comparação com setores como o transporte ou a agricultura, ela se torna cada vez mais significativa à luz da crescente digitalização da economia, da centralização da computação em infraestruturas de hiperescala e da demanda por serviços digitais, inteligência artificial e computação em nuvem. Além disso, o impacto ambiental varia segundo as fontes de energia utilizadas, com maiores emissões ocorrendo em regiões que continuam a depender fortemente de combustíveis fósseis.

Portanto, esse valor de 1% não apenas reflete a pegada ambiental atual dos data centers, mas também ressalta a importância estratégica do setor nas políticas de mitigação climática, particularmente aquelas alinhadas aos objetivos do Acordo de Paris. O estudo (LYKOU; MENTZELIOTI; GRITZALIS, 2018) ressalta que os centros de dados consomem aproximadamente 2% do suprimento global de energia elétrica, uma parcela significativa. Além disso, eles contribuem com 1% das emissões globais de gases de efeito estufa, destacando assim seu papel nos desafios ambientais atuais. Outro aspecto ambiental importante é o consumo de água.

Alguns *data centers* utilizam água para a refrigeração dos sistemas. Adicionalmente, é relevante mencionar o uso de minerais e metais essenciais em sua construção e operação, o que intensifica o impacto ambiental associado a essas infraestruturas. A extração desses minerais para a fabricação de equipamentos provoca a degradação dos ecossistemas, agravando ainda



mais as preocupações ambientais.

Para diminuir os impactos e colaborar para serem alcançadas as metas estabelecidas no Acordo de Paris, neste artigo, eles propuseram as melhores práticas para aumentar a eficiência energética; por exemplo, em servidores, os autores propõem hardware mais eficiente, como processadores que trabalhem de forma simultânea.

Alto consumo de energia

O artigo *How much energy do data centers really use* destaca que a escassez de dados públicos sobre o consumo energético e as emissões de carbono dos *data centers* dificulta estimativas precisas do impacto ambiental do setor (MASANET; LEI, 2020). Apenas algumas grandes empresas, como *Google, Apple, Switch e Facebook*, divulgam essas informações, o que reforça a ausência de transparência generalizada.

Os autores ressaltam que essa falta de dados, especialmente sobre localização e a intensidade de carbono da matriz elétrica utilizada, impede a quantificação adequada das emissões. Também comparam a magnitude das emissões dos *data centers* à da indústria da aviação e alertam para o papel crescente de tecnologias emergentes — como inteligência artificial e 5G — no aumento do tráfego de dados e, conseqüentemente, no consumo energético.

Como estratégias para melhorar a transparência, o artigo propõe o desenvolvimento e o compartilhamento de dados confiáveis sobre infraestrutura e uso de energia, o aprimoramento das capacidades de modelagem por meio de abordagens preditivas e colaborativas, além da criação de métodos voltados à análise de tendências tecnológicas emergentes (MASANET; LEI, 2020).

Emissão de dióxido no treinamento de aprendizagem

O artigo em questão, publicado pela renomada empresa de tecnologia Google, aborda o consumo de energia no treinamento de Aprendizado de Máquina (PATTERSON et al., 2022). As emissões de dióxido de carbono nesse processo são classificadas como operacionais e de ciclo de vida. As emissões operacionais referem-se ao custo da energia para operar o *hardware*, incluindo despesas gerais, enquanto as emissões de ciclo de vida ocorrem durante a fabricação dos componentes.

O artigo destaca principalmente as emissões operacionais, mencionando estudos que identificaram uma superestimação do custo e da pegada de carbono no treinamento de aprendizado de máquina. Isso se deve à falta de acesso a informações corretas, à extrapolação de dados e à não consideração das melhorias em *hardware e software*. É essencial obter números mais precisos para avaliar os desafios relacionados às mudanças climáticas adequadamente.

O artigo propõe melhores práticas, como a seleção de arquiteturas de modelos eficientes e o uso de abordagens híbridas otimizadas para o treinamento de Aprendizado de Máquina e de computação em nuvem. Essas práticas podem reduzir significativamente o consumo de energia e



as emissões de carbono, contribuindo para a redução da pegada de carbono ao longo do tempo.

Q3- Qual a necessidade de um *data center* verde?

Redução do uso de energia e o impacto ambiental

Segundo (JNR; MAJID; ROMLI, 2018) as indústrias de TI são as maiores usuárias de *data centers*, o uso da TI verde se faz necessário pelo grande consumo de energia, e se torna prática constante o uso de *data center* verde para que seja possível a redução do uso de energia e do impacto ambiental que o consumo dela causa: liberação de dióxido de carbono na atmosfera. Eles propõem algumas práticas a serem adotadas, como servidores e sistemas de refrigeração mais eficientes, o uso de fontes de energias diversas como a eólica e a solar, com a otimização de energia com o uso de ferramentas de monitoração, desligamento de servidores ociosos e reciclagem dos materiais.

Neste artigo, os autores realizam uma revisão da literatura e explicam cinco estratégias do ciclo de vida da Computação Verde em *data centers*: projeto verde, produção verde, compras verdes, operações ecológicas e descarte verde. Cada fase do ciclo aborda o uso de fontes de energia renováveis, virtualização de servidores, consolidação de recursos, monitoramento do desempenho, otimização do resfriamento, reciclagem de equipamentos e gerenciamento de resíduos, para reduzir as emissões de CO₂.

O questionário aplicado pelos autores Jnr *et al.* (JNR; MAJID; ROMLI, 2018) foi elaborado com base na natureza descritiva do estudo e nos procedimentos de Hair *et al.* (HAIR *et al.*, 2009). Antes da aplicação, três especialistas em computação verde validaram o instrumento, ajustado conforme suas recomendações. A coleta de dados envolveu 133 profissionais de TI de indústrias da Malásia com *data centers* internos. A confiabilidade foi verificada por meio do alfa de Cronbach, com valor de 0,986. As respostas analisaram cinco estratégias de ciclo de vida que avaliam a aplicação prática da computação verde em *data centers*, com os resultados tratados por meio de estatísticas descritivas. Tais resultados reforçam a ideia de que a prática de design verde deve ser aplicada nos *data centers* do setor de TI.

Alto consumo de energia e pouca utilização:

Segundo os autores Kotowski, Jerzy e Jacek, o aumento do uso da internet e a consequente demanda por *data centers* (KOTOWSKI; OKO *et al.*, 2018). Com o crescimento anual da internet acima de 10%, as demandas por esses Centros de Processamento de Dados aumentaram em 20%. No entanto, esse aumento de uso também gera preocupações quanto aos custos, especialmente devido aos sistemas de armazenamento de energia caros.

A eficiência energética se destaca como um ponto fundamental. Grandes *data centers* frequentemente operam abaixo de 10% de sua capacidade, o que resulta em gastos energéticos excessivos. Essa ineficiência também está ligada às emissões de carbono, o que torna a adoção de práticas sustentáveis fundamental.



Os autores sugerem a adoção estratégica da Computação em Nuvem Verde como solução. Isso permitiria o compartilhamento otimizado de recursos, promovendo eficiência energética e automação da infraestrutura. Em resumo, otimizar o uso de data centers, adotar práticas sustentáveis e explorar a computação em nuvem são passos cruciais para lidar com a crescente demanda, reduzir custos e minimizar os impactos ambientais.

Sustentabilidade e Economicidade:

No artigo elaborado por Heikkinen e Daan destaca-se a relevância da adoção de abordagens sustentáveis em *data centers*, referidos como *Data Centers Verdes* (HEIKKINEN, 2023). Os autores enfatizam não somente a importância da implementação de energias renováveis, mas também abordam questões críticas relacionadas às emissões de gases de efeito estufa, conservação de água, gerenciamento de resíduos, uso responsável da terra, preservação dos ecossistemas e da biodiversidade, todos essenciais para construir um futuro sustentável.

O contexto moderno, caracterizado pelo aumento do uso de dispositivos móveis e tecnologias computacionais, resulta em um significativo incremento no consumo de energia e água, acompanhado de um aumento das emissões de dióxido de carbono em cada data center implantado. Nesse sentido, torna-se imperativo a adoção de abordagens mais ecológicas nos data centers, de modo a contribuir efetivamente para as metas de redução das emissões de CO₂.

Além das considerações ambientais, o aspecto econômico também se destaca. Pesquisas demonstram claramente os benefícios substanciais que as empresas obtêm ao adotar práticas sustentáveis. Portanto, há uma interseção proveitosa entre a sustentabilidade e o benefício financeiro.

Um fator determinante para a construção de data centers verdes é a demanda exponencial por armazenamento de dados. Nesse contexto, a eficiência energética é essencial para atender às necessidades crescentes e reduzir o impacto ambiental.

Assim sendo, *data centers* ecológicos desempenham um papel fundamental na sociedade contemporânea. No entanto, é imprescindível que essas iniciativas não apenas se alinhem à redução do impacto ambiental, mas também gerem eficiência nos custos operacionais. Para atender tanto à demanda de armazenamento de dados quanto à adoção de energias renováveis, essas práticas contribuem significativamente para o meio ambiente e também para a saúde financeira das organizações.

Em síntese, ao analisarmos os estudos sobre eficiência energética e sustentabilidade em *data centers*, percebemos que ainda há importantes soluções a serem exploradas, tais como o uso de ferramentas de automação, *DevOps* e Inteligência Artificial, que não receberam a devida atenção. Existem poucos estudos sobre esses temas. A maioria das pesquisas até o momento tem se concentrado em soluções como o resfriamento da sala dos servidores, a redução do consumo de energia por meio de sistemas de gerenciamento de energia e a reciclagem de materiais. No entanto, é importante direcionar o foco para a possibilidade de alcançar eficiência por meio do



uso de ferramentas de automação.

5.3 Conclusões da Revisão

Nesta análise sistemática, observou-se que há inúmeros motivos para a adoção de *data centers*, que desempenham um papel sustentável no meio ambiente. Para alcançar a eficiência, diversas práticas podem ser implementadas, tais como a containerização, o aprimoramento na gestão do resfriamento, a consolidação de servidores, além do descarte e da reciclagem adequados dos materiais e resíduos gerados. Como destacado, a adoção de sustentabilidade é de extrema importância, pois não apenas auxilia na redução do consumo energético, mas também contribui para a economia de recursos financeiros e desempenha seu papel na sociedade, tornando-a mais conectada sem agredir o meio ambiente.

Esta análise sistemática investigou estratégias para melhorar a eficiência energética, com foco na adoção de práticas sustentáveis. A análise dos dados coletados revelou que a implementação de técnicas, como o uso de *DevOps*, a otimização de energia, a gestão de energia e outros elementos que contribuem para a redução do consumo de energia, resultou na otimização dos recursos, confirmando que essas práticas podem reduzir significativamente o consumo de energia e aumentar a sustentabilidade operacional.

Os resultados obtidos têm importantes implicações para a indústria de tecnologia da informação, indicando que a adoção de práticas verdes e de tecnologias emergentes, como a virtualização e a inteligência artificial, pode não apenas reduzir os custos operacionais, mas também minimizar o impacto ambiental. Por exemplo, a utilização de fontes de energia renováveis e a reciclagem de recursos são estratégias eficazes que podem ser amplamente implementadas.

Apesar das contribuições significativas, este estudo apresenta algumas limitações. Primeiramente, a análise foi limitada a artigos em inglês por ter um conteúdo mais abrangente na língua inglesa e verificamos também que em alguns artigos foi citada a dificuldade em mensurar a emissão dos Gases do Efeito Estufa (GEE) e outra limitação observada foi estudos relacionados ao uso de *DevOps* para redução do consumo de energia. Essas limitações sugerem a necessidade de estudos futuros que explorem o uso de *DevOps*, especialmente em infraestruturas de grande porte.

Em conclusão, este estudo oferece uma perspectiva sobre a gestão sustentável de *data centers* e abre caminho para futuras pesquisas que podem continuar a explorar estratégias de eficiência energética. A aplicação dos resultados pode beneficiar empresas de TI e gestores de centros de processamento de dados, contribuindo para avanços relacionados a sustentabilidade tecnológica.



6 SOLUÇÃO PROPOSTA

Este capítulo apresenta a solução proposta como um *projeto DevOps sustentável* para o ambiente institucional. O objetivo é organizar o fluxo de trabalho, os artefatos de automação e a integração com a medição elétrica, de modo a permitir, no Capítulo 7, a comparação objetiva entre a execução manual e a automatizada quanto a tempo e energia, com a consequente estimativa das emissões de CO₂.

6.1 Visão geral e princípios

A proposta baseia-se em três princípios: (i) padronização das rotinas operacionais em roteiros e *playbooks* equivalentes; (ii) observabilidade do consumo elétrico por meio de janelas temporais alinhadas às execuções; e (iii) segurança operacional por meio de janelas de mudança, validações e retorno seguro (*rollback*) quando necessário. Com isso, a automação não apenas reduz tempo e retrabalho, mas também fornece um trilho verificável para associar ações de operação a efeitos medidos de energia.

6.2 Cenário de infraestrutura

O ambiente computacional encontra-se em *data center* do tipo contêiner, com climatização de precisão, alimentação ininterrupta e rede corporativa. Há dois *clusters* de virtualização: um de produção (hipervisor VMware ESXi 7.x) e um legado (VMware ESXi 6.7.x). Em conjunto, a instituição opera 134 máquinas virtuais, que hospedam serviços acadêmicos e administrativos (aplicações web, bancos de dados, autenticação, monitoramento e segurança).

Topologia e pools: Os recursos estão organizados em pools lógicos de produção, homologação e desenvolvimento, com limites e reservas de CPU e de memória definidos pelo time de infra. As camadas de rede separam o tráfego de usuários, o gerenciamento e o armazenamento, reduzindo a interferência entre os planos de dados e de controle.

Sistemas operacionais e versões: O parque virtual é heterogêneo: há instâncias com Debian, Ubuntu, CentOS, Rocky Linux, AlmaLinux e Microsoft Windows, em diferentes versões suportadas pela instituição. No escopo desta dissertação, não foi realizada a contagem exata por família/versão; quando necessário, a análise se refere apenas às máquinas diretamente envolvidas nos ensaios.

6.3 Arquitetura da solução DevOps

A solução organiza-se em quatro camadas articuladas:



(a) Orquestração e automação: Um nó de orquestração, com Linux e *Ansible*, executa os *playbooks* contra os nós-alvo. O inventário define grupos e variáveis; cada rotina operacional possui um roteiro humano (propósito, pré-condições, passos e critérios de sucesso) e um *playbook* correspondente com os mesmos passos, garantindo equivalência entre os modos manual e automatizado.

(b) Integração com a medição elétrica: O subsistema IMAMS, acoplado ao contêiner, fornece leituras periódicas de energia. As janelas de execução (início/fim) são registradas para associar, por meio da variação do acumulado, o consumo de cada rodada.

(c) Trilhas de auditoria: Cada execução registra carimbos de tempo, saídas de validação e contexto de ambiente (sistema operacional do nó-alvo, hipervisor e parâmetros relevantes). Esses registros permitem repetir e auditar o procedimento.

(d) Operação segura: As execuções ocorrem em janelas de menor interferência, com checagens de pré- e pós-condições e, quando aplicável, plano de retorno, reduzindo o risco à produção.

6.4 Ambiente de teste e validação inicial

Para validar o encadeamento proposto, utilizou-se um ambiente representativo no hipervisor institucional: um servidor de orquestração (Debian 12) e dois nós-alvo virtuais, *debian01* (Debian 12) e *rock01* (Rocky Linux 9) conforme mostra Figura 6. O arquivo *hosts/inventory* mapeia nomes lógicos e parâmetros de conexão conforme mostra exemplo na Figura 5.

Figura 5 – Exemplo ilustrativo do arquivo *hosts/inventory* (dados sensíveis suprimidos).

```
web1.example.com
web2.example.com
db.example.com

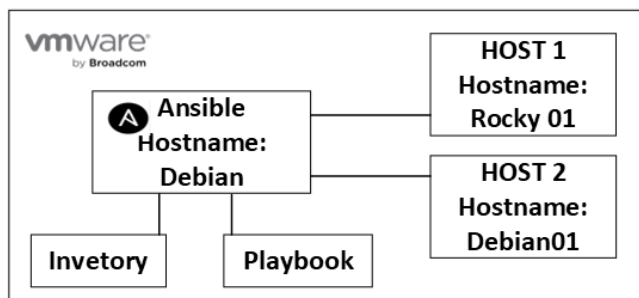
[webservers]
web1.example.com
web2.example.com

[dbservers]
db.example.com

[datacenter:children]
webservers
dbservers

[datacenter:vars]
ansible_connection=local
```

Fonte: (ANSIBLE, 2023).

Figura 6 – Topologia resumida do ambiente (orquestrador e nós-alvo no hipervisor institucional).

Fonte: Do autor.

O primeiro ensaio consistiu em verificar a conectividade e a autorização por SSH entre o orquestrador e os nós, conforme mostra a Figura 7

```
ansible all -m ping
```

Listing 6.1 – Teste de conectividade com os nós-alvo**Figura 7 – Retorno bem-sucedido do teste de conectividade (ansible all -m ping).**

```
root@debian:/etc/ansible# ansible -m ping all
debian01 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
rocky01 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
```

Fonte: Do autor.

Com a conectividade validada, executou-se uma rotina simples e controlada de atualização de índices de pacotes do sistema (equivalente ao procedimento manual), confirmando a idempotência e a aplicação consistente das mudanças nas duas máquinas conforme pode ser observado na Figura 8. Esses testes asseguram que a automação reproduz fielmente o roteiro humano, condição necessária para a comparação a ser realizada no Capítulo 7.

6.5 Integração com o IMAMS

O IMAMS registra tensão, corrente, potência e energia acumulada do contêiner em intervalos. Para mensurar o consumo associado a cada execução, as janelas de início e término são alinhadas às leituras do IMAMS. Como o instrumento observa o contêiner como um todo, a proposta mitiga o ruído por meio de três medidas complementares: (i) escolha de janelas

Figura 8 – Atualização lista de pacotes

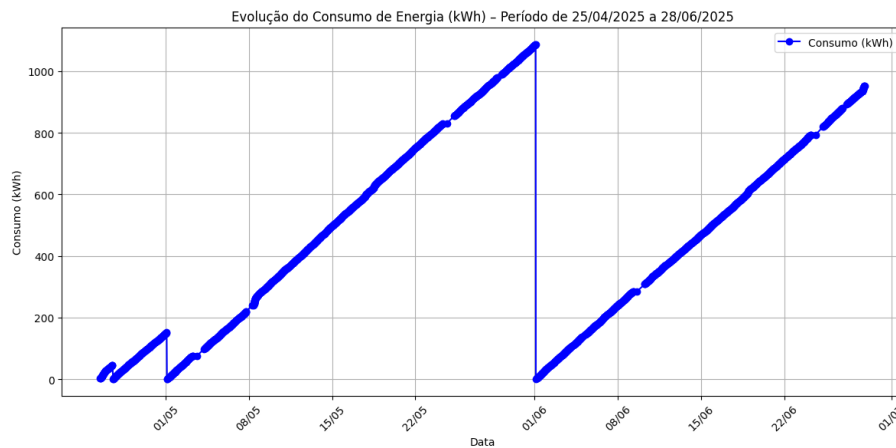
```

root@debian:/etc/ansible/inventory# ansible-playbook -i hosts /etc/ansible/playbooks/1_UpdateLinux.yml
PLAY [Update Rocky Linux Servers - Starting Deploy] *****
TASK [Gathering Facts] *****
ok: [rocky01]
TASK [DNF | Update System] *****
changed: [rocky01]
PLAY [Update Debian Linux Servers - Starting Deploy] *****
TASK [Gathering Facts] *****
ok: [debian01]
TASK [Debian Like | Update System] *****
changed: [debian01]
PLAY RECAP *****
  debian01 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
  rocky01  : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
    
```

Fonte: Do autor.

de menor interferência; (ii) repetição de rodadas sob condições equivalentes; e (iii) descarte justificado de rodadas com anomalias operacionais detectadas em registros auxiliares.

Figura 9 – Evolução do consumo de energia elétrica no período de 25 de abril a 28 de junho de 2025.



Fonte: Elaboração própria a partir de dados do sistema IMAMS (2025).

A figura 9 fornece a visão geral utilizada como referência institucional de consumo, permitindo identificar tendências e apoiar o planejamento das janelas de ensaio.

Esta representação identifica padrões de uso, avalia a eficiência energética dos processos e embasa ações de otimização. Ressalta-se que os valores apresentados são acumulativos, indicando o crescimento do consumo ao longo do tempo.

6.5.1 Estimativa das emissões: calculadora do IDESAM vs. cálculo manual

No período de abril a junho de 2025, o consumo elétrico totalizado foi de **2.476,38 kWh**. Para usar a calculadora do IDESAM (que solicita um valor *mensal*), dividimos esse total por três meses:

$$\bar{E}_{\text{mês}} = \frac{2.476,38 \text{ kWh}}{3} \approx 825,46 \text{ kWh/mês.}$$

Inserindo $\bar{E}_{\text{mês}}$ na calculadora do IDESAM (Instituto de Conservação e Desenvolvimento Sustentável da Amazônia, 2024) (modo “consumo de energia”), obteve-se uma emissão de



$\approx 0,042 \text{ t CO}_2/\text{ms}$. Projetando para três meses, isso resulta em $\approx 0,126 \text{ t CO}_2$ no período.

Em paralelo, realizamos o cálculo *manual* utilizando o fator médio de emissão adotado nesta dissertação para 2024, de $54,5 \text{ g CO}_2/\text{kWh} = 0,0545 \text{ kg CO}_2/\text{kWh}$:

$$E_{\text{período}} \times \text{fator} = 2.476,38 \text{ kWh} \times 0,0545 \frac{\text{kg CO}_2}{\text{kWh}} \approx 134,96 \text{ kg CO}_2 = 0,135 \text{ t CO}_2.$$

Em termos mensais, o mesmo fator aplicado ao consumo médio dá:

$$\bar{E}_{\text{mês}} \times \text{fator} = 825,46 \text{ kWh} \times 0,0545 \frac{\text{kg CO}_2}{\text{kWh}} \approx 44,99 \text{ kg CO}_2 = 0,045 \text{ t CO}_2/\text{ms}.$$

A pequena diferença entre os resultados ($\approx 0,042 \text{ t}$ vs. $\approx 0,045 \text{ t}$ por mês) é esperada e decorre de arredondamentos e/ou do fator efetivamente utilizado pela calculadora em seu backend (por exemplo, atualização anual do fator médio ou ajustes regionais). Nesta dissertação, mantemos o fator explícito de $54,5 \text{ g CO}_2/\text{kWh}$ (ano-base 2024) para garantir a transparência e a reprodutibilidade dos cálculos.

6.5.2 Compensação e mitigação.

Além de orientar ações de eficiência energética e a adoção de fontes renováveis (como a geração fotovoltaica), as emissões estimadas podem ser objeto de *compensação* por meio de re-florestamento. Na calculadora do IDESAM, o montante anual de aproximadamente $0,505 \text{ tCO}_2$ indica a necessidade de **1 árvore/ano** para neutralização indicativa. Adotando a mesma razão, proporcional ao período analisado nesta seção (três meses, $\approx 0,126 \text{ tCO}_2$), obtém-se a recomendação prática de **três árvores** no total (*equivalente a uma árvore por mês*).

Ressalta-se que essa indicação tem caráter *operacional* e depende de premissas simplificadoras da própria calculadora (espécie, bioma e horizonte de sequestro). Ainda assim, fornece um referencial transparente de compensação a ser considerado em conjunto com as medidas estruturais de redução no consumo e com iniciativas de energia limpa.

O modelo matemático adotado para a estimativa das emissões de dióxido de carbono (CO_2) foi desenvolvido com base nas diretrizes do *Green Algorithms* (LANNELONGUE; GREALEY; INOUE, 2021), que estabelecem uma relação direta entre o consumo energético e o fator de emissão de carbono da matriz elétrica, conforme a ABNT NBR ISO 14067:2015. Essa escolha fundamenta-se na ampla aceitação do modelo na literatura científica e em sua adequação para medições empíricas em ambientes computacionais reais, como o caso em questão. Além disso, sua estrutura conceitual foi comparada ao modelo incremental proposto por Saraiva et al. (SARAIVA et al., 2025), que, embora voltado à simulação de sistemas distribuídos, oferece uma perspectiva complementar e de validação conceitual para os resultados obtidos nesta dissertação.



6.6 Tarefas automatizadas no projeto

As rotinas selecionadas refletem o cotidiano do time de operações e têm impacto transversal: atualização de índices e pacotes do sistema; reinício controlado de serviços de aplicação e de banco de dados; limpeza e rotação de arquivos de registro; e tarefas de manutenção do sistema de periódicos. Cada rotina mantém paridade entre o roteiro humano e o *playbook*, permitindo medir diferenças de tempo e de consumo de energia exclusivamente atribuíveis ao modo de execução.

6.7 Reprodutibilidade e evidências

Embora não tenha sido utilizado um sistema formal de versionamento, foram preservados os artefatos necessários à repetição dos ensaios: *playbooks* efetivamente utilizados, inventário, roteiros de execução, parâmetros adotados, marcações temporais e saídas das execuções. Os anexos reúnem cópias desses materiais, com remoção de dados sensíveis, permitindo que outro integrante da equipe repita os procedimentos nas mesmas condições do experimento original. As evidências consolidadas também foram arquivadas no repositório ([GitHub](#)), conforme a referência indicada nos anexos.

Com a arquitetura definida, o encadeamento operacional estabilizado e a integração com o IMAMS estabelecida, o próximo capítulo apresenta as execuções controladas e a análise comparativa entre os modos manual e automatizado, reportando tempos, consumos de energia por janela e as emissões estimadas de CO₂ associadas.



7 RESULTADOS E DISCUSSÃO

Este capítulo detalha os resultados da implementação da automação no ambiente de produção. Foram abordados as tarefas automatizadas, os recursos empregados e os dados de consumo de energia apurados. O processo iniciou-se com a análise das atividades críticas e rotineiras da infraestrutura, visando identificar as mais adequadas à automação. A partir desse levantamento, organizou-se uma lista de tarefas, segmentada em quatro categorias principais: banco de dados, servidor web, segurança do servidor e atualização do sistema OJS. A Tabela 7 apresenta uma visão geral dessas categorias e das ações automatizadas correspondentes, que serão detalhadas nas subseções seguintes.

Além dos resultados experimentais, os artefatos produzidos ao longo desta pesquisa, incluindo os *playbooks Ansible*, arquivos de configuração e scripts auxiliares utilizados na automação das tarefas avaliadas, foram disponibilizados em um repositório público no GitHub, de modo a garantir transparência, reprodutibilidade e acesso aberto aos materiais desenvolvidos. O repositório pode ser acessado em: <https://github.com/josianybritto/ansible/tree/main>.

Tabela 7 – Tarefas organizadas por categoria no ambiente de servidor web

Banco de Dados	Servidor Web	Segurança do Servidor	Sistema OJS
Backup	Instalação do banco de dados	Regras no IPTABLES	Backup
Restauração	Instalação do PHP e módulos Instalação do Apache	Instalação do Fail2Ban	Atualização Restauração

Fonte: Elaborado pela autora.

Para a coleta de dados referentes ao consumo de energia, utilizou-se o sistema proprietário *IMAMS (Integrated Monitoring and Management System)*, cedido pela MDC Indústria, empresa responsável pela fabricação do contêiner. Esta solução é utilizada para o monitoramento remoto da infraestrutura física do *data center*, incluindo climatização, fornecimento de energia, sensores ambientais, segurança e conectividade de rede. O *IMAMS* possibilitou a obtenção de informações confiáveis para avaliar o impacto da automação no consumo de energia da infraestrutura.

7.1 Automação de Backup e Restauração do Banco de Dados

Conforme apresentado na Coluna 1 da Tabela 7, foi realizada a automação das tarefas de **backup e restauração** de bancos de dados. Estas atividades são essenciais para garantir a integridade e a segurança das informações, além de facilitar a migração entre servidores.

Na prática, a automação desse fluxo possibilita a cópia e a transferência seguras de dados entre servidores, do de origem para o de destino. Isso minimiza o risco de perda e agiliza o tempo



de resposta em situações de troca ou atualização do ambiente de hospedagem. Um exemplo disso, no contexto analisado, foi a migração do sistema operacional da máquina de produção do sistema OJS, do Ubuntu 22 para o Debian 12.

Para fins comparativos, foram realizados dois procedimentos distintos: um automatizado e outro manual. O processo automatizado foi conduzido com a ferramenta *Ansible*, por meio da execução de um *playbook*, conforme apresentado no Apêndice [A.1](#). As tarefas contempladas nesse fluxo foram:

1. Verificação da existência do usuário do banco de dados na máquina de destino, com criação do mesmo caso inexistente;
2. Execução do *dump* (backup) da base de dados;
3. Substituição da *collation* `utf8mb4_0900_ai_ci` por `utf8mb4_general_ci`, necessária devido à diferença de versões do MariaDB entre a origem e o destino;
4. Transferência do arquivo de *dump* para a máquina de destino;
5. Restauração do banco de dados no servidor de destino;
6. Exclusão dos arquivos temporários de *dump* em ambas as máquinas.

Ressalta-se que a alteração da *collation* foi necessária para permitir a restauração bem-sucedida do banco de dados do sistema OJS.

Para efeito de medição e comparação, o mesmo processo foi reproduzido manualmente por meio de um *script shell*, permitindo a análise do tempo e do desempenho entre os dois métodos, tema aprofundado nas seções subsequentes. O resultado da execução do *playbook* pode ser observado no Log de execução apresentado no Apêndice [B.3](#)

7.1.1 Coleta de dados de consumo de energia do backup e transferência do banco de dados processo automático

Durante a execução automatizada do processo de backup e restauração do banco de dados do sistema OJS, coletaram-se dados de consumo energético. A medição foi realizada no intervalo de 09:44 a 11:01, conforme apresentado na Tabela [8](#).

A coleta de energia durante esse período de execução não foi registrada segundo o segundo, o que inviabilizou o cálculo direto do consumo específico do processo. Por esse motivo, foi necessário realizar uma estimativa proporcional com base no tempo.

O consumo total de energia no período das 09:44:18 às 11:01:19 foi de 1,01 kWh, ao longo de 77 minutos e 1 segundo, totalizando 4621 segundos. Como o processo de cópia e



Tabela 8 – Leituras de consumo de energia elétrica no período do processo automatizado.

Período	Início	Fim	Consumo (kWh)
Antes	09:44:18	10:21:18	10,06
Depois	10:21:48	11:01:19	11,07
Total	09:44:18 → 11:01:19	Duração: 1h17min01s	1,01

Fonte: Elaborado pela autora.

restauração levou 107 segundos, foi necessário estimar o consumo proporcional. Aplicando a regra de três, estimou-se um consumo de aproximadamente 0,0234 kWh.

Dessa forma, aplicou-se uma regra de três para estimar o consumo proporcional de energia referente apenas ao processo automatizado, conforme descrito a seguir:

A energia proporcional E consumida durante o processo automatizado é estimada por:

$$E = \left(\frac{t_{\text{execução}}}{T_{\text{intervalo}}} \right) \times E_{\text{total}} = \left(\frac{107}{4621} \right) \times 1,01 \quad (7.1)$$

Efetuada os cálculos:

$$E \approx 0,0234 \text{ kWh} = 0,0234 \text{ kWh} \quad (7.2)$$

7.1.2 Coleta de dados de consumo de energia do backup e transferência do banco de dados processo manual

Para a execução manual da tarefa de backup e restauração do banco de dados, realizou-se também a medição do consumo de energia elétrica. A atividade foi executada com o auxílio de um script que facilitou o controle do tempo total do procedimento. Os dados registrados estão apresentados na Tabela 9.

Tabela 9 – Leituras de consumo de energia elétrica no período do processo manual

Período	Início	Fim	Consumo (kWh)
Antes	17:27:48	17:56:48	21,94
Depois	17:57:18	18:36:48	22,95
Total	17:27:48 → 18:36:48	Duração: 69 minutos	1,01

Fonte: Elaborado pela autora.

O consumo total de energia registrado entre 17:27:48 e 18:36:48 foi de 1,01 kWh, ao longo de 69 minutos (4140 segundos). O tempo cronometrado para a execução manual da cópia e da restauração do banco de dados foi de 106 segundos. Para estimar o consumo proporcional referente a esse intervalo específico, aplicou-se a seguinte fórmula:



$$E = \left(\frac{t_{\text{execução}}}{T_{\text{intervalo}}} \right) \times E_{\text{total}} = \left(\frac{106}{4140} \right) \times 1,01 \quad (7.3)$$

Efetuada os cálculos:

$$E \approx 0,0258 \text{ kWh} = 0,0258 \text{ kWh} \quad (7.4)$$

Dessa forma, estima-se que o processo manual de backup e restauração do banco de dados consumiu aproximadamente **0,026 kWh** (ou **26 Wh**) de energia elétrica, entre os horários de 17:36:34 e 17:38:20.

7.1.3 Comparativo entre os Processos Manual e Automatizado de Backup do Banco de Dados

Esta subseção apresenta a comparação entre os procedimentos manuais e automatizados de backup e restauração do banco de dados do sistema OJS. Ambos os testes foram realizados em condições controladas, utilizando os mesmos dados e o mesmo ambiente computacional.

Conforme mostrado na Tabela 10, os dois processos apresentaram tempos muito próximos, com duração de aproximadamente 1 minuto e 47 segundos. O consumo estimado de energia, entretanto, foi ligeiramente inferior no processo automatizado, que consumiu 0,023 kWh, enquanto o manual consumiu 0,026 kWh.

Tabela 10 – Consumo estimado de energia elétrica nos períodos analisados

Tipo	Período	Duração	Consumo estimado (kWh)
Automação	10:04–10:06	1min47s	0,023
Manual	17:36–17:38	1min46s	0,026

Fonte: Elaborado pela autora.

A economia percentual do processo automatizado em relação ao processo manual é calculada por:

$$\text{Economia (\%)} = \left(\frac{E_{\text{manual}} - E_{\text{auto}}}{E_{\text{manual}}} \right) \times 100 = \left(\frac{0,026 - 0,023}{0,026} \right) \times 100 \quad (7.5)$$

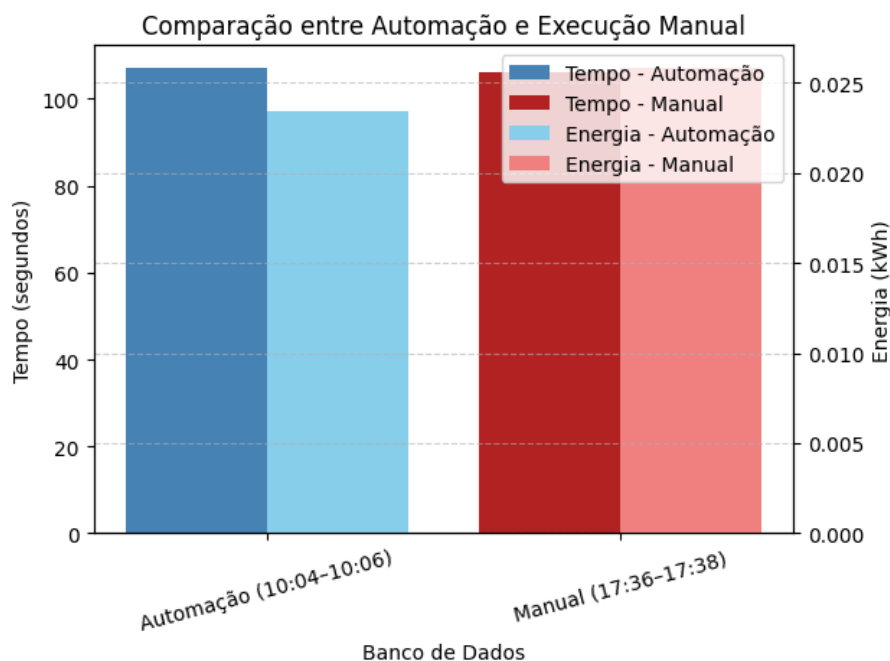
$$\text{Economia (\%)} \approx \left(\frac{0,003}{0,026} \right) \times 100 \approx 11,54\% \quad (7.6)$$

Assim, estima-se uma economia de aproximadamente **11,54%** no consumo de energia elétrica com o uso da automação. Embora o tempo de execução dos dois métodos tenha sido

praticamente o mesmo, a redução no consumo energético reforça a importância da automação como prática sustentável e eficiente, especialmente quando aplicada de forma recorrente em ambientes computacionais de maior escala.

A Figura 10 ilustra essa comparação entre os métodos.

Figura 10 – Comparativo Automação x Manual



Fonte: Elaborado pela autora.

7.2 Automação para criação do Servidor Web

Esta seção apresenta os resultados da automação do processo de criação de um ambiente de servidor web, correspondente à Coluna 2 da Tabela 7. O ambiente automatizado baseia-se na pilha LAMP, composta pelos serviços Apache, MariaDB e PHP, conforme descrito no Capítulo 2.

A automação foi realizada por meio de um *playbook Ansible*, apresentado no Apêndice A.2, responsável pela instalação, configuração e inicialização dos serviços necessários ao funcionamento do servidor web.

A automação foi realizada por meio de um *playbook Ansible* descrito no Apêndice A.2. Esse *playbook* foi estruturado com as seguintes tarefas:

- Instalação do serviço `apache2`;
- Instalação dos pacotes `mariadb-server` e `mariadb-client`;
- Instalação do `php8.1` e suas respectivas dependências.



O processo equivalente foi conduzido manualmente por meio de um script em shell, com o objetivo de facilitar a medição do tempo de execução. O resultado da execução do playbook pode ser observado no Log de Execução apresentado no Apêndice [B.4](#).

7.2.1 Coleta de dados de consumo de energia na instalação automática da pilha LAMP

A coleta dos dados de consumo de energia do ambiente ocorreu durante a execução automatizada do processo de instalação da pilha LAMP (Linux, Apache, MariaDB e PHP). A medição foi realizada entre 07:37:24 e 09:01:54, conforme detalhado na Tabela [II](#).

O consumo total de energia no período das 07:39:24 às 09:01:54 foi de 1,00 kWh, ao longo de 74 minutos, totalizando 4140 segundos o intervalo de leitura de energia. O processo de instalação da pilha LAMP, de forma automática, levou 43,68 segundos e, em 4440 segundos, foram consumidos 1,00 kWh. Então, em 43,68 segundos foram consumidos x kWh, conforme podemos observar na Tabela [II](#).

Tabela 11 – Leituras de consumo de energia elétrica no período do processo automatizado (LAMP)

Período	Início	Fim	Consumo (kWh)
Antes	07:39:24	08:23:24	7,04
Depois	08:23:54	09:01:54	8,04
Total	07:39:24 → 09:01:54	Duração: 74 minutos	1,00

Fonte: Elaborado pela autora.

Então aplicamos a seguinte regra de três para estimar o consumo proporcional, onde:

A energia proporcional E consumida durante o processo automatizado é estimada por regra de três:

$$E = \left(\frac{t_{\text{execução}}}{T_{\text{intervalo}}} \right) \times E_{\text{total}} = \left(\frac{43,68}{4440} \right) \times 1,01 \quad (7.7)$$

Efetuando os cálculos:

$$E \approx 0,0098 \text{ kWh} = 0,0098 \text{ kWh} \quad (7.8)$$

Assim, o consumo estimado de energia entre 07:39:24 e 09:38:20 foi de aproximadamente 0,0098 kWh de energia elétrica.

Portanto, estima-se que a instalação automática da pilha **LAMP** tenha consumido aproximadamente.



7.2.2 Coleta de dados de consumo de energia na instalação manual da pilha LAMP

No experimento da instalação manual da pilha LAMP (Linux, Apache, MariaDB e PHP), realizou-se a coleta de energia durante o período de 08:58:48 às 10:12:48 conforme mostra a Tabela 12.

Tabela 12 – Leituras de consumo de energia elétrica no período do processo manual

Período	Início	Fim	Consumo (kWh)
Antes	08:58:48	09:33:48	13,07
Depois	09:34:18	10:12:48	14,07
Total	08:58:48 → 10:12:48	Duração: 74 minutos	1,00

Fonte: Elaborado pela autora.

O consumo total de energia no período das 08:58:48 às 10:12:48 foi de 1,00 kWh, ao longo de 74 minutos, totalizando 4140 segundos o intervalo de leitura de energia. O processo de instalação da pilha LAMP levou 56,38 segundos, ou seja, se em 4140 segundos foram consumidos 1,00 kWh, então em 56,38 segundos foram consumidos x kWh, conforme podemos observar na Tabela 12.

Então aplicamos a seguinte regra de três para estimar o consumo proporcional, onde:

A energia proporcional E consumida durante o processo manual é estimada por regra de três:

$$E = \left(\frac{t_{\text{execução}}}{T_{\text{intervalo}}} \right) \times E_{\text{total}} = \left(\frac{56,38}{4140} \right) \times 1,01 \quad (7.9)$$

Efetuando os cálculos:

$$E \approx 0,0137 \text{ kWh} = 0,0137 \text{ kWh} \quad (7.10)$$

A instalação manual da pilha **LAMP** consumiu aproximadamente 0,0137 kWh de energia elétrica.

7.2.3 Comparativo manual e automatizado.

Conforme mostrado na Tabela 14, apresenta-se o comparativo entre o processo manual e o automatizado de execução da instalação da pilha LAMP, com os tempos de execução e os consumos de energia estimados. Esses processos foram executados de forma semelhante, com a instalação dos mesmos pacotes e das mesmas versões no mesmo ambiente.



Tabela 13 – Consumo proporcional estimado de energia elétrica nos processos manual e automatizado (LAMP)

Tipo	Início–Fim	Duração	Consumo proporcional (kWh)
Automação	08:47:56– 08:48:40	43,68 s	0,0098
Manual	09:17:34– 09:18:30	56,38 s	0,0137

Fonte: Elaborado pela autora.

A automação realizada via playbook do Ansible teve um tempo de execução de 43,68 segundos, enquanto o processo manual demorou cerca de 56,38 segundos para concluir um processo semelhante. A redução de tempo foi de cerca de 22,5% em relação ao método de instalação mais usual.

Tabela 14 – Consumo proporcional estimado de energia elétrica nos processos manual e automatizado (LAMP)

Tipo	Início–Fim	Duração	Consumo proporcional (kWh)
Automação	08:47:56– 08:48:40	43,68 s	0,0098
Manual	09:17:34– 09:18:30	56,38 s	0,0137

Fonte: Elaborado pela autora..

A economia percentual do processo automatizado em relação ao processo manual é calculada por:

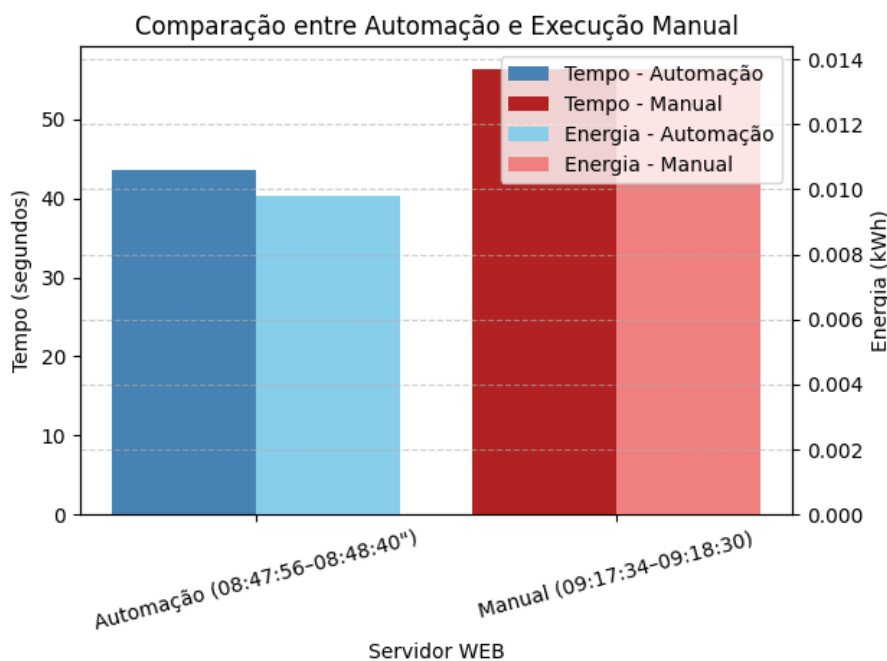
$$\text{Economia (\%)} = \left(\frac{0,0137 - 0,0098}{0,0137} \right) \times 100 \quad (7.11)$$

$$\text{Economia (\%)} \approx \left(\frac{0,0039}{0,0137} \right) \times 100 \approx 28,47\% \quad (7.12)$$

A diferença entre as execuções com relação à economia de energia pode ser estimada em cerca de 28,47% em favor da automação. Se for considerada a recorrência dos processos executados em ambientes de larga escala, como data centers de grande porte e servidores institucionais, a automação é uma boa prática que pode ser adotada para promover um ambiente mais sustentável e eficiente em termos de recursos computacionais.

A Figura 11 apresenta a comparação entre os métodos automáticos e manuais de instalação da pilha LAMP, considerando o tempo de execução e o consumo estimado de energia elétrica. Verificou-se que a automação apresenta menor tempo de execução e consumo energético proporcionalmente menores, o que comprova ser uma prática eficiente e sustentável na gestão de TI.

Figura 11 – Comparativo Automação x Manual



Fonte: Elaborado pela autora.

7.3 Automação de Regras de Segurança

No que se refere ao item da coluna 3 da Tabela 7, que trata da segurança do servidor, destaca-se a importância crítica dessa etapa no contexto de sistemas web expostos à internet. Tais sistemas são alvos constantes de ataques, especialmente aqueles que exploram vulnerabilidades para execução remota de código, com o objetivo de hospedar páginas maliciosas em *websites* legítimos previamente comprometidos, prática conhecida como *defacement*.

Para mitigar esses riscos, é fundamental que as regras de segurança sejam aplicadas desde a criação do servidor que hospedará os sistemas. No caso específico do servidor web que hospedará o sistema OJS, será adotada a configuração do *firewall iptables*.

Além disso, será implementada a ferramenta Fail2Ban, voltada à prevenção de ataques de força bruta e de negação de serviço distribuída (DDoS).

A execução automatizada da configuração de segurança no servidor foi realizada por meio do *playbook* apresentado no Apêndice A.3, que aplica as principais regras de proteção



utilizando os recursos do *firewall iptables* em conjunto com a ferramenta *Fail2Ban*. A estrutura do *playbook* abrange desde a criação de um diretório para armazenamento de métricas até a configuração completa das regras de segurança e dos serviços de proteção.

Inicialmente, o *playbook* registra o horário local de início da execução e, ao final, também o de término, permitindo calcular a duração total da tarefa. Durante a execução, o tempo é capturado com precisão pelo comando `time`, que armazena as informações de desempenho em um arquivo específico.

A parte central do processo compreende a instalação dos pacotes *iptables*, *iptables-persistent* e *fail2ban*, seguida da definição de regras explícitas para controle de acesso às portas SSH, HTTP, HTTPS, MySQL, e protocolos ICMP. Tais regras asseguram que apenas redes autorizadas tenham acesso aos serviços essenciais, bloqueando as conexões externas não previstas.

Posteriormente, o script configura o *Fail2Ban* para proteger os serviços SSH e apache, ajustando os parâmetros de monitoramento e o limite de tentativas falhas. O serviço é então habilitado e reiniciado, concluindo a automação da política de segurança.

O registro completo das etapas realizadas, incluindo os horários de início e término e a aplicação das regras, pode ser observado no log de execução apresentado na Listagem [B.5](#), que comprova a efetividade da tarefa e serve de base para a análise da execução automatizada.

7.3.1 Coleta de dados de consumo de energia das implantações automáticas de regras de segurança.

No experimento de implantação das regras de segurança de forma automática e instalação do *Fail2band* e criação da *jail* do SSH e do Apache, realizou-se a coleta dos dados de energia durante o período que foi de 16:40:26 às 17:57:26, conforme mostra a Tabela [15](#)

Tabela 15 – Leituras de consumo de energia elétrica no período do processo automatizado (Segurança)

Período	Início	Fim	Consumo (kWh)
Antes	16:40:26	17:23:56	24,18
Depois	17:24:26	17:57:26	25,19
Total	16:40:26 → 17:57:26	Duração: 77 minutos	1,01

Fonte: Elaborado pela autora.

O consumo total de energia no período das 16:40:26 às 17:57:26 foi de 1,00 kWh, ao longo de 77 minutos, totalizando 4.620 segundos o intervalo de leitura de energia. O processo de implantação das regras de segurança, instalação do *fail2ban* e criação das *jails* de forma automática levou 3,29 segundos; ou seja, se em 4.620 foram consumidos 1,00 kWh, então em 3,29 segundos foram consumidos x kWh, conforme podemos observar na Tabela [15](#).



A energia proporcional E consumida durante o processo automatizado é estimada por regra de três:

$$E = \left(\frac{t_{\text{execução}}}{T_{\text{intervalo}}} \right) \times E_{\text{total}} = \left(\frac{3,29}{4620} \right) \times 1,01 \quad (7.13)$$

Efetuando os cálculos:

$$E \approx 0,000719 \text{ kWh} = 0,000719 \text{ kWh} \quad (7.14)$$

Portanto, a instalação do *iptables* e do *Fail2Ban*, e a aplicação das regras de *firewall* consumiram aproximadamente **0.000719 kWh** de energia elétrica.

7.3.2 Coleta de dados de consumo de energia das implantações manuais das regras de segurança.

Tabela 16 – Leituras de consumo de energia elétrica no período do processo manual (iptables + Fail2Ban)

Período	Início	Fim	Consumo (kWh)
Antes	12:52:56	13:24:26	18,14
Depois	13:24:56	14:03:26	19,14
Total	12:52:56 → 14:03:26	Duração: 70 minutos	1,00

Fonte: Elaborado pela autora.

A energia proporcional E consumida durante o processo manual é estimada por regra de três:

$$E = \left(\frac{t_{\text{execução}}}{T_{\text{intervalo}}} \right) \times E_{\text{total}} = \left(\frac{4,28}{4200} \right) \times 1,00 \quad (7.15)$$

Efetuando os cálculos:

$$E \approx 0,00102 \text{ kWh} = 0,00102 \text{ kWh} \quad (7.16)$$

7.3.3 Comparativo manual e automatizado.

Conforme mostrado na Tabela [17](#), é apresentado o comparativo entre o processo manual e o automatizado da implantação das regras de segurança e instalação do *fail2ban* e criação das *jails*, mostrando o tempo de execução e o consumo de energia estimado. Esses processos foram executados de forma semelhante e no mesmo ambiente.



A automação realizada via playbook do Ansible teve um tempo de execução de 3,29 segundos, e o processo manual demorou cerca de 4,29 segundos para concluir o processo semelhante.

Tabela 17 – Consumo proporcional estimado de energia elétrica nos processos manual e automatizado (Segurança)

Tipo	Início–Fim	Duração	Consumo proporcional (kWh)
Automação	17:24:26–17:24:29	3,29 s	0,000719
Manual	13:24:56–13:25:00	4,28 s	0,00102

Fonte: Elaborado pela autora.

A economia percentual do processo automatizado em relação ao processo manual é calculada por:

$$\text{Economia (\%)} = \left(\frac{E_{\text{manual}} - E_{\text{auto}}}{E_{\text{manual}}} \right) \times 100 = \left(\frac{0,00102 - 0,000719}{0,00102} \right) \times 100 \quad (7.17)$$

$$\text{Economia (\%)} \approx \left(\frac{0,000301}{0,00102} \right) \times 100 \approx 29,51 \% \quad (7.18)$$

A diferença entre as execuções, com relação à economia de energia, pode ser estimada em cerca de 29,51% em favor da automação. Ainda que o consumo absoluto seja baixo, quando se consideram cenários com múltiplas execuções em servidores institucionais ou em ambientes com alta frequência de provisionamento de regras de segurança, a automação representa uma prática vantajosa do ponto de vista energético e operacional.

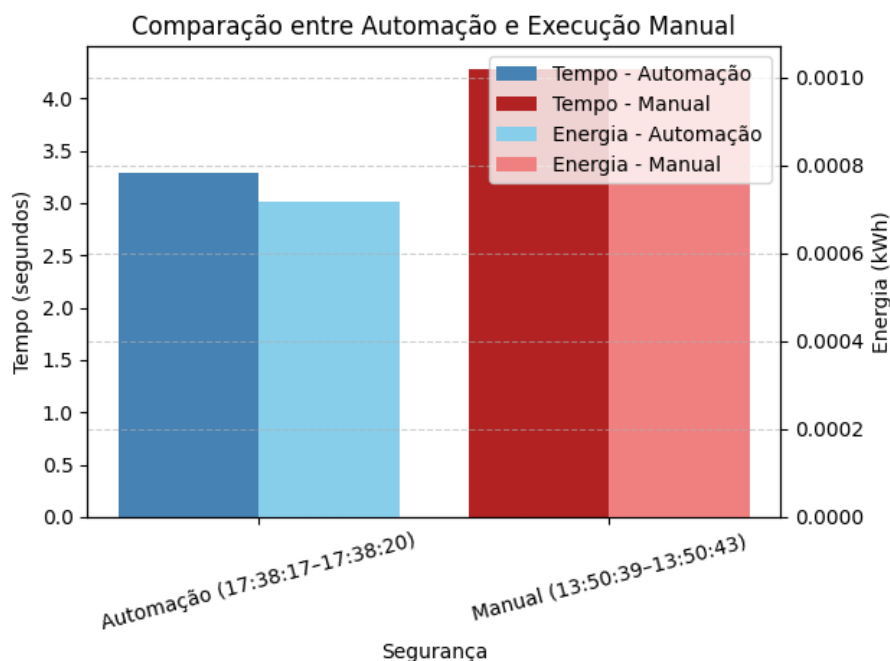
A Figura 12 apresenta a comparação entre os métodos automáticos e manuais da aplicação das regras de segurança, considerando tanto o tempo de execução quanto o consumo proporcional de energia elétrica. Verifica-se que a automação proporciona menor tempo de execução e menor consumo energético, reforçando seu papel como prática eficiente e sustentável na gestão da infraestrutura de TI.

7.4 Automação da atualização do sistema OJS

A última coluna da Tabela 7 OJS é um sistema web de código aberto utilizado para a publicação de periódicos; ele facilita o uso para editores e autores, permitindo a revisão dos textos de forma simples e intuitiva. Visando assegurar a padronização, a correção e a rastreabilidade do processo de atualização do software OJS, foi desenvolvido um script automatizado com a ferramenta Ansible. Tal script, detalhado no Apêndice A.4, orquestra uma sequência de tarefas



Figura 12 – Comparativo Segurança



Fonte: Elaborado pela autora.

complexas entre um computador central, um servidor de dados (ajaes) e um servidor de operação do OJS (hajaes). O procedimento foi estruturado em etapas bem definidas, descritas a seguir.

A atualização automatizada do sistema OJS para a versão 3.4.0-2 foi conduzida por meio do playbook apresentado no Apêndice **A.4**. Esse procedimento é dividido em cinco fases principais, organizadas logicamente para garantir a integridade dos dados, a preparação adequada do ambiente de destino, a restauração dos arquivos persistentes e a finalização correta da configuração do sistema.

Na **Fase I**, realiza-se o *backup* dos dados persistentes diretamente do servidor de origem. Essa operação é iniciada com o registro do tempo de execução e utiliza o comando `tar` via SSH, transmitindo os dados diretamente ao nó de controle, sem criar arquivos intermediários no servidor remoto. São preservados os diretórios `public/`, `ojs-files/` (com a exclusão do diretório `usageStats`, por ser volumoso e regenerável) e o arquivo `config.inc.php`, que contém as configurações críticas do sistema.

Na **Fase II**, o ambiente de destino é preparado para receber a nova versão do OJS. O pacote oficial é baixado diretamente do site do *Public Knowledge Project*, descompactado e movido para o diretório de instalação. A estrutura anterior do sistema é removida, garantindo uma instalação limpa e livre de conflitos com versões anteriores.

A **Fase III** trata da transferência do arquivo de *backup* para o servidor de destino. A operação é realizada com o comando `scp` a partir do nó de controle, executada apenas uma vez, mesmo que o *playbook* seja aplicado a múltiplos servidores. Isso assegura a consistência e a

eficiência na transferência de dados.

Na **Fase IV**, com o *backup* posicionado no destino, inicia-se o processo de restauração. É criado um diretório temporário para a extração dos arquivos, evitando a substituição direta dos diretórios de produção. A cópia é realizada com o comando `rsync`, utilizando as opções `-a` e `-delete`, garantindo a preservação de permissões, e propriedade, além de uma sincronização precisa. Por fim, o arquivo `config.inc.php` é restaurado, concluindo a configuração da instância.

A **Fase V** é responsável pela conclusão da atualização. Nela, são ajustadas as permissões de arquivos e diretórios, garantindo que o usuário do servidor web (`www-data`) tenha controle total sobre os dados e os arquivos de configuração. Em seguida, o *script* de atualização do OJS (`php tools/upgrade.php upgrade`) é executado, aplicando as modificações necessárias no banco de dados para a compatibilidade com a nova versão.

Ao término do processo, o bloco `post_tasks` limpa os arquivos temporários e registra o tempo total de execução. Essa abordagem automatizada permite uma implantação segura, repetível e monitorável, de acordo com as boas práticas da IaC. O log completo da execução, contendo os registros de tempo e a sequência de tarefas realizadas, encontra-se no Apêndice **B.6**.

7.4.1 Coleta de dados de consumo de energia da atualização do sistema OJS e da restauração dos arquivos do processo automático.

No experimento de atualização do sistema OJS, onde foi realizada a atualização da versão 3.4.0.1 para a versão 3.4.0.2 e a restauração dos arquivos para o diretório web, realizou-se a coleta dos dados de energia durante o período de 22:27:07 às 23:00:37 e a instalação aconteceu no período de 22:42:05 às 22:55:21, o que corresponde ao tempo de 796 segundos (13 minutos e 16 segundos).

A energia proporcional E consumida durante o processo automatizado é estimada por regra de três:

$$E = \left(\frac{t_{\text{execução}}}{T_{\text{intervalo}}} \right) \times E_{\text{total}} = \left(\frac{796}{4320} \right) \times 1,53 \quad (7.19)$$

Efetuando os cálculos:

$$E \approx 0,1843 \times 1,53 \approx 0,282 \text{ kWh} \quad (7.20)$$

Portanto, a transferência dos arquivos de uma máquina para outra, a atualização da versão e a restauração dos arquivos do sistema OJS consumiram aproximadamente 0.282 kWh de energia elétrica.



Tabela 18 – Leituras de consumo de energia elétrica no período do processo automatizado (backup OJS)

Período	Início	Fim	Consumo (kWh)
Antes	21:48:37	22:01:07	32,00
Depois	22:27:07	23:00:37	33,53
Total	21:48:37 → 23:00:37	Duração: 72 minutos	1,53

Fonte: Elaborado pela autora.

7.4.2 Coleta de dados de consumo de energia da atualização do sistema OJS e da restauração dos arquivos do processo manual.

No experimento de atualização do sistema OJS de forma manual, onde foi realizada a atualização da versão 3.4.0.0 para a versão 3.4.0.1 e a restauração dos arquivos para o diretório web, realizou-se a coleta dos dados de energia durante o período de 16:48:35 às 17:36:35 e a instalação aconteceu no período de 16:54:16 às 17:13:28, o que corresponde ao tempo de 1074 segundos (17 minutos e 54 segundos).

A energia proporcional E consumida durante o processo manual é estimada por regra de três:

$$E = \left(\frac{t_{\text{execução}}}{T_{\text{intervalo}}} \right) \times E_{\text{total}} = \left(\frac{1074}{2880} \right) \times 1,01 \quad (7.21)$$

Efetuando os cálculos:

$$E \approx 0,3729 \times 1,01 \approx 0,376 \text{ kWh} \quad (7.22)$$

Portanto, a transferência dos arquivos de uma máquina para outra, a atualização da versão e a restauração dos arquivos do sistema OJS consumiram aproximadamente 0.376 kWh de energia elétrica.

Tabela 19 – Leituras de consumo de energia elétrica no período do processo manual (atualização do OJS 3.4.0-1)

Período	Início	Fim	Consumo (kWh)
Antes	16:06:35	16:48:35	23,43
Depois	16:49:05	17:36:35	24,44
Total	16:48:35 → 17:36:35	Duração: 48 minutos	1,01

Fonte: Elaborado pela autora.



7.4.3 Comparativo manual e automatizado.

Conforme mostrado na Tabela 20, é apresentado o comparativo entre o processo manual e o automatizado da transferência dos arquivos, atualização da versão do sistema OJS e restauração dos arquivos persistentes, mostrando o tempo de execução e o consumo proporcional de energia elétrica. Esses processos foram executados de forma semelhante, com os mesmos arquivos, comandos e ambiente de execução.

A automação realizada via playbook do Ansible teve um tempo de execução de 13 minutos e 16 segundos (ou 796 segundos), enquanto o processo manual demorou aproximadamente 17 minutos e 54 segundos (ou 1074 segundos) para concluir o procedimento equivalente. A redução de tempo foi de aproximadamente 25,87% em favor da automação.

Tabela 20 – Consumo proporcional estimado de energia elétrica nos processos manual e automatizado (Atualização do OJS)

Tipo	Início-Fim	Duração	Consumo proporcional (kWh)
Automação	22:42:05–22:55:21	13 min 16 s	0,282
Manual	16:54:16–17:13:28	17 min 54 s	0,376

Fonte: Elaborado pela autora.

A economia percentual do processo automatizado em relação ao processo manual é calculada por:

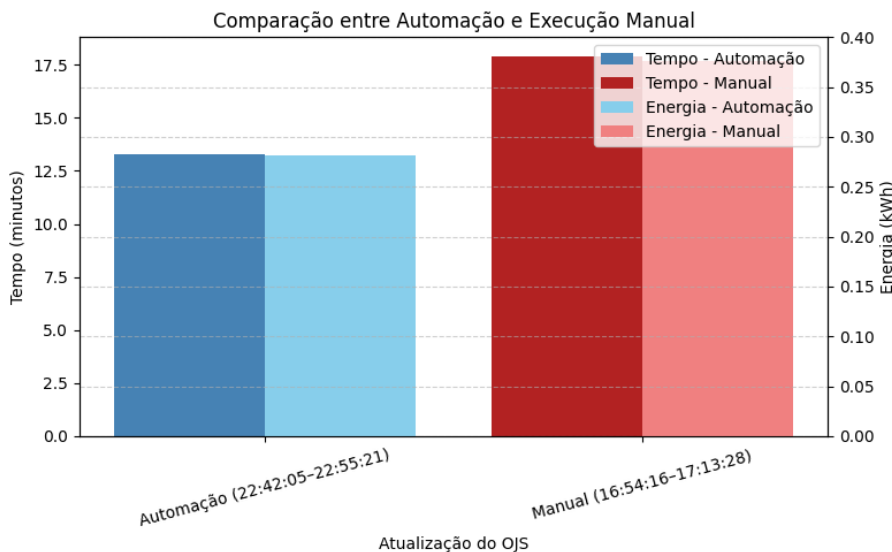
$$\text{Economia (\%)} = \left(\frac{E_{\text{manual}} - E_{\text{auto}}}{E_{\text{manual}}} \right) \times 100 = \left(\frac{0,376 - 0,282}{0,376} \right) \times 100 \quad (7.23)$$

$$\text{Economia (\%)} \approx \left(\frac{0,094}{0,376} \right) \times 100 \approx 25,0\% \quad (7.24)$$

A diferença entre os métodos em relação ao consumo energético evidencia uma economia estimada de 25% com o uso da automação. Essa redução torna-se relevante principalmente quando se considerada a repetição da tarefa em ambientes institucionais, como data centers e servidores de múltiplas revistas. A automação, além de reduzir tempo e esforço operacional, mostra-se eficiente também do ponto de vista energético.

A Figura 13 apresenta graficamente a comparação entre os métodos automatizados e manual para a atualização do OJS, destacando as diferenças de tempo de execução e consumo estimado de energia. Observa-se que o método automatizado apresenta vantagens expressivas em ambos os aspectos, consolidando-se como uma abordagem sustentável e eficiente na manutenção de sistemas de publicação científica.

Figura 13 – Comparativo atualização OJS



Fonte: Elaborado pela autora.

7.5 Estimativa de emissões de CO₂ a partir do consumo de energia elétrica das tarefas

Após a apresentação dos resultados referentes ao consumo energético, esta seção aborda a estimativa das emissões de dióxido de carbono (CO₂) associadas às execuções analisadas, bem como a fundamentação teórica dos modelos empregados nessa estimativa. O cálculo das emissões foi realizado conforme as diretrizes do modelo *Green Algorithms* (LANNELONGUE; GREALEY; INOUYE, 2021), que se baseia na relação direta entre a energia consumida e o fator de emissão da matriz elétrica, conforme a norma ABNT NBR ISO 14067:2015.

Na subseção seguinte, apresenta-se uma análise comparativa entre o modelo *Green Algorithms* e o modelo *SimGrid/Batsim* (SARAIVA et al., 2025), destacando suas diferenças conceituais, metodológicas e de aplicabilidade no contexto deste estudo.

7.5.1 Metodologia de cálculo

A medição das emissões de dióxido de carbono (CO₂) associadas ao consumo de energia elétrica foi realizada com base na norma ABNT NBR ISO 14067:2015 (Associação Brasileira de Normas Técnicas, 2015), que estabelece princípios, requisitos e diretrizes para a quantificação da pegada de carbono de produtos, considerando todo o seu ciclo de vida. De acordo com essa norma, as emissões indiretas decorrentes do consumo de eletricidade devem ser calculadas pela multiplicação da quantidade de energia utilizada pelo fator de emissão correspondente da matriz energética do país ou região.

O cálculo das emissões de dióxido de carbono (CO₂) neste estudo seguiu as diretri-



zes propostas por (LANNELONGUE; GREALEY; INOUYE, 2021), amplamente utilizadas em avaliações computacionais de sustentabilidade. Segundo os autores, as emissões de carbono associadas ao consumo de energia elétrica (Escopo 2) podem ser estimadas conforme a Equação 7.25:

$$E_{CO_2} = E \times FE_{CO_2} \quad (7.25)$$

onde:

- E_{CO_2} representa as emissões de carbono (em gCO₂);
- E é o consumo de energia elétrica (em kWh);
- FE_{CO_2} é o fator de emissão de CO₂ (em gCO₂/kWh).

No Brasil, a responsabilidade pela publicação dos fatores de emissão de CO₂ é do **Ministério da Ciência, Tecnologia e Inovação (MCTI)**, por intermédio do **Sistema de Registro Nacional de Emissões (SIRENE)**. Segundo (Ministério da Ciência, Tecnologia e Inovação, 2024a), os fatores refletem o perfil da matriz energética do **Sistema Interligado Nacional (SIN)**, predominantemente composta por fontes renováveis.

Com base nos dados oficiais divulgados pelo MCTI, o fator médio de emissão para o ano de 2024 foi de:

$$54,5 \text{ gCO}_2/\text{kWh}$$

A partir desse valor, as emissões de carbono associadas ao consumo energético total (E) de cada tarefa automatizada foram obtidas pela Equação 7.25, na qual o fator de emissão FE_{CO_2} corresponde ao perfil da matriz elétrica brasileira. Assim, a fórmula prática utilizada neste trabalho foi expressa da seguinte forma:

$$\text{Emissão de CO}_2 \text{ (g)} = \text{Energia (kWh)} \times 54,5 \text{ gCO}_2/\text{kWh} \quad (7.26)$$

O consumo energético de cada tarefa foi obtido do sistema *Integrated Monitoring and Management System (IMAMS)*, com registros médios a cada 30 minutos. Como o sistema não fornece dados em segundos, o consumo energético (E) de cada tarefa foi estimado proporcionalmente ao tempo de execução, conforme a Equação 7.27:

$$E = \left(\frac{t_{\text{execução}}}{t_{\text{intervalo}}} \right) \times E_{\text{total}} \quad (7.27)$$

onde:

- $t_{\text{execução}}$ é o tempo de execução da tarefa (em segundos);
- $t_{\text{intervalo}}$ é o intervalo total de medição (em segundos);
- E_{total} é o consumo de energia total registrado no intervalo (em kWh).

Substituindo os valores obtidos em $E_{CO_2} = E \times FE_{CO_2}$, foi possível estimar as emissões associadas a cada cenário (execuções manuais e automatizadas). Dessa forma, o percentual de economia de emissões foi calculado a partir da diferença entre os dois modos de execução, refletindo o impacto direto da automação sobre a sustentabilidade energética das operações.

Esta metodologia permite mensurar, de forma proporcional, as emissões indiretas associadas às tarefas computacionais analisadas, tanto na execução manual quanto na automatizada. Os valores obtidos são apresentados na Tabela 21.

Tabela 21 – Comparação entre a execução manual e a automatizada das tarefas com o fator de emissão de 2024.

Tarefa	T-Manual (s)	Energia-M (kWh)	T-Auto (s)	Energia-A (kWh)	Economia (%)	CO ₂ Manual (g)	CO ₂ Auto (g)
Servidor Web	56,38	0,0137	43,68	0,0098	28,47	0,75	0,53
Banco de Dados	106,00	0,0260	107,00	0,0230	11,54	1,42	1,25
Regras de Segurança	4,28	0,00102	3,29	0,000719	29,51	0,06	0,04
Atualização do Sistema OJS	1074,00	0,3760	796,00	0,2820	25,00	20,49	15,37
Total	1240,66	0,41672	949,97	0,31552	24,28	22,72	17,19

Fator de emissão adotado: 54,5 gCO₂/kWh (MCTI/SIRENE, 2024).

Fonte: Elaborado pela autora, com base em dados do MCTI (2024).

Com base nos dados apresentados na Tabela 21, verifica-se que a execução manual das tarefas resultou em uma emissão total de **22,72 g de CO₂**, enquanto a execução automatizada gerou **17,19 g de CO₂**. Tais valores foram calculados multiplicando-se a energia consumida em cada abordagem pelo fator médio de emissão de **54,5 gCO₂/kWh**, referente ao ano-base de 2024, conforme disponibilizado pelo MCTI (Ministério da Ciência, Tecnologia e Inovação, 2024b).

Os resultados alcançados evidenciam que a implementação de automação, além de aprimorar a eficiência do tempo de execução, concorre para a diminuição do consumo energético e, por conseguinte, das emissões de CO₂. A análise comparativa reitera o papel da automação como uma abordagem sustentável na administração de uma infraestrutura computacional

7.5.2 Discussão comparativa entre os modelos *Green Algorithms* e SimGrid/Batsim

O presente estudo adotou como referência metodológica o modelo proposto por Lannelongue, Grealey e Inouye, denominado *Green Algorithms*, amplamente reconhecido por estabelecer uma estrutura padronizada e de ampla aplicabilidade para estimar a pegada de carbono de processos computacionais (LANNELONGUE; GREALEY; INOUE, 2021). Tal modelo fundamenta-se na quantificação das emissões indiretas de dióxido de carbono (CO₂)

associadas ao consumo de energia elétrica, em consonância com os princípios estabelecidos na norma ABNT NBR ISO 14067:2015, por meio da combinação entre o consumo de energia e o fator de emissão da matriz elétrica considerada.

De forma simplificada, o cálculo das emissões de CO₂ associado ao consumo de energia elétrica neste trabalho segue a relação proposta por (LANNELONGUE; GREALEY; INOUE, 2021), expressa na Equação 7.28:

$$E_{CO_2} = E \times FE_{CO_2}, \quad (7.28)$$

em que E representa a energia consumida (em kWh) e FE_{CO_2} corresponde ao fator de emissão de CO₂ (em gCO₂e/kWh) associado à matriz energética adotada. Essa formulação permite estimar as emissões de Escopo 2 a partir de parâmetros observáveis (tempo de execução, recursos computacionais e contexto energético), preservando a compatibilidade com avaliações de ciclo de vida e com relatórios de sustentabilidade baseados em fatores de emissão médios.

Por outro lado, o modelo apresentado por Saraiva et al. no trabalho *Estimating CO₂ emissions of distributed applications and platforms with SimGrid/Batsim* (SARAIVA et al., 2025) adota uma abordagem incremental e simulacional, implementada no ecossistema SimGrid/Batsim, com foco na estimativa contínua das emissões de CO₂ associadas ao consumo de energia de *hosts* computacionais em ambientes distribuídos. Nesse modelo, o consumo de energia em um intervalo de tempo é calculado a partir da potência instantânea P do *host* e do intervalo $[t_0, t_1]$, conforme a Equação 7.29:

$$E_{\text{step}} = P \times (t_1 - t_0), \quad (7.29)$$

sendo posteriormente convertido para kWh pela Equação 7.30:

$$E_{\text{step}}^{\text{kWh}} = \frac{E_{\text{step}}}{3,6 \times 10^6}, \quad (7.30)$$

e utilizado para o cálculo incremental das emissões de carbono no intervalo, de acordo com a Equação 7.31:

$$C_{\text{step}} = E_{\text{step}}^{\text{kWh}} \times CI_{\text{step}}, \quad (7.31)$$

Em que CI_{step} é a intensidade de carbono da matriz elétrica (em gCO₂e/kWh) no intervalo considerado. As emissões acumuladas até o instante t_1 são então atualizadas conforme a Equação 7.32:

$$C_{\text{total}}(t_1) = C_{\text{total}}(t_0) + C_{\text{step}}. \quad (7.32)$$



A comparação entre os dois modelos evidencia diferenças relevantes de escopo e de granularidade. O modelo *Green Algorithms* (LANNELONGUE; GREALEY; INOUYE, 2021) foi concebido para fornecer uma estimativa padronizada e de ampla aplicação da pegada de carbono de tarefas computacionais, a partir de parâmetros agregados (tempo de execução, número de núcleos, memória, eficiência do data center e intensidade de carbono), o que favorece a reprodutibilidade e a comparação entre estudos. Já o modelo de Saraiva et al. (SARAIVA et al., 2025) prioriza uma representação detalhada e dinâmica do comportamento energético de *hosts* em simulações de larga escala, atualizando o consumo de energia e as emissões a cada evento relevante (inicialização, mudança de estado de potência, início e término de tarefas).

No contexto desta dissertação, a adoção do modelo baseado em *Green Algorithms* mostrou-se mais adequada, uma vez que o objetivo central é estimar e comparar as emissões de CO₂ associadas a execuções reais (manuais e automatizadas) em um ambiente computacional específico, utilizando dados empíricos de tempo e energia agregados. Assim, o modelo de Lannelongue et al. (LANNELONGUE; GREALEY; INOUYE, 2021) oferece uma estrutura conceitual compatível com a ABNT NBR ISO 14067:2015 e suficientemente flexível para integrar fatores de emissão representativos da matriz elétrica brasileira, enquanto o modelo de Saraiva et al. (SARAIVA et al., 2025) se mostra especialmente útil em cenários de simulação detalhada de plataformas distribuídas.

Dessa forma, pode-se afirmar que os dois modelos são complementares: o *Green Algorithms* fornece uma base metodológica normativa e generalista para quantificação da pegada de carbono de computações, ao passo que o modelo SimGrid/Batsim fornece um arcabouço incremental e orientado a eventos para estudos de estratégias de escalonamento e gestão de recursos em sistemas distribuídos. A escolha pelo primeiro, neste trabalho, decorre tanto da natureza empírica dos experimentos quanto da necessidade de alinhamento às normas de pegada de carbono de produtos e serviços.



8 CONCLUSÃO

O presente estudo adotou práticas de *DevOps* para a automação da instalação e configuração de ambientes computacionais, com foco no sistema OJS. A metodologia empregada alinhou-se aos princípios da IaC, visando otimizar a padronização, a eficiência operacional e a sustentabilidade ambiental.

A análise comparativa entre os modos de operação manual e automatizado evidenciou que a adoção de práticas de *DevOps* promove uma contribuição para a otimização do tempo de execução das tarefas e para a mitigação do consumo de energia elétrica, impactando, por conseguinte, na redução das emissões de CO₂. A estimativa baseada no fator médio de emissão nacional de 2024 permitiu quantificar esses impactos de forma confiável, resultando em uma redução média de 24,3% nas emissões de carbono no processo automatizado.

Os resultados obtidos reforçam que a automação, orientada pelo paradigma *DevOps*, não apenas otimiza os processos, mas também proporciona benefícios ambientais e econômicos. Além disso, essa abordagem proporciona flexibilidade, escalabilidade e reprodutibilidade, elementos essenciais para a gestão eficiente de ambientes computacionais modernos.

Além disso, a integração dessa abordagem em fluxos de trabalho mais amplos é simplificada, o que possibilita aprimorar as tarefas automatizadas por meio de ações como a execução de rotinas administrativas recorrentes, a coleta de métricas, o envio de notificações a sistemas de monitoramento e o registro de logs e resultados em repositórios externos. Essa flexibilidade operacional ressalta que a adoção de práticas de *DevOps* pode constituir uma estratégia estratégica e complementar para a governança e a sustentabilidade de ambientes computacionais.

Conclui-se, assim, que a automação orientada por práticas de *DevOps* representa uma estratégia eficaz, viável e ambientalmente sustentável para ambientes de TI, sendo recomendada para organizações, tanto públicas quanto privadas, que buscam otimizar seus recursos e reduzir seu impacto ambiental.

8.1 Trabalhos Futuros

Este trabalho apresenta resultados promissores e abre perspectivas para aprimoramentos e novas implementações, entre as quais destacam-se:

- **Automação verde e sustentabilidade:** integrar práticas de TI Verde em *playbooks* de automação, criando rotinas que, além de configurar sistemas, otimizem automaticamente o consumo de energia.
- **Avaliação de impacto social:** analisar como a prática *DevOps* influencia o desempenho da equipe de TI, considerando a redução de tarefas repetitivas, a curva de aprendizado e a



realocação de esforços para atividades estratégicas.

- **Integração com ferramentas de orquestração e entrega contínua:** expandir a automação para além da configuração de servidores, incorporando soluções que atuam em diferentes fases do ciclo de vida da infraestrutura e aplicações:
 - *Release:* *Jenkins* e *Rundeck*, para automação de *pipelines* de integração e entrega contínua.
 - *Operate:* *Rancher* e *Kubernetes*, para orquestração e gerenciamento de contêineres.
 - *Configure:* *Ansible*, consolidando a gestão de configuração de sistemas e serviços.

Essa integração possibilitaria avaliar a eficiência energética em *pipelines DevOps* completos, do *release* à operação, ampliando o nível de automação estudado.



REFERÊNCIAS

- AILANE, M. T.; RUBNER, C.; RAUSCH, A. Green devops: A strategic framework for sustainable software development. In: **Computer Sciences & Mathematics Forum**. [S.l.]: MDPI, 2025. v. 10, p. 5. Article 5.
- ALBALAWI, M. et al. Website defacement detection and monitoring methods: A review. **Electronics**, MDPI, v. 11, n. 21, p. 3573, 2022.
- ANSIBLE. **Ansible Documentation**. 2023. <<https://docs.ansible.com/>>.
- ARRUDA, P. L. L. de et al. Práticas da ti verde 4.0 que impactam a sustentabilidade 4.0: uma revisão sistemática da literatura. **Revista de Gestão e Secretariado**, v. 16, n. 5, p. e4862–e4862, 2025.
- Associação Brasileira de Normas Técnicas. **ABNT NBR ISO 14067: Gases de efeito estufa — Pegada de carbono de produtos — Requisitos e diretrizes para quantificação e comunicação**. Rio de Janeiro, 2015. Disponível para consulta no Catálogo da ABNT.
- BARTHOLOMEW, D. Mariadb vs. mysql. **Dostopano**, v. 7, n. 10, p. 2014, 2012.
- BASMADJIAN, R. Flexibility-based energy and demand management in data centers: a case study for cloud computing. **Energies**, MDPI, v. 12, n. 17, p. 3301, 2019.
- BECK, K. et al. **Manifesto for agile software development**. [S.l.]: Snowbird, UT, 2001. <<https://agilemanifesto.org/>>.
- BROOKS, S.; WANG, X.; SARKER, S. Unpacking green it: A review of the existing literature. 2010.
- BYRD, D. E. T. T. A. Measuring the flexibility of information technology infrastructure: Exploratory analysis of a construct. **Journal of management information systems**, Taylor & Francis, v. 17, n. 1, p. 167–208, 2000.
- CAMPOS, L. **Eficiência Energética**. 2022. <<https://ecoa.org.br/eficiencia-energetica/>> Acesso em: 25 jul. 2023.
- CARREIRA, C. et al. From "worse is better" to better: Lessons from a mixed methods study of ansible's challenges. **arXiv preprint arXiv:2504.08678**, 2025.
- CHANOPAS, A.; KRAIRIT, D.; KHANG, D. B. Managing information technology infrastructure: a new flexibility framework. **Management Research News**, Emerald Group Publishing Limited, v. 29, n. 10, p. 632–651, 2006.
- CHATLATANAGULCHAI, W.; CHANTRAPORNCHAI, C. Energy consumption data collection: case study on data center in a thai university. **Energy Informatics**, Springer, v. 7, n. 1, p. 26, 2024.
- DEBOIS, P. Agile infrastructure and operations: How infra-gile are you? In: IEEE. **Agile 2008 Conference**. [S.l.], 2008. p. 202–207.
- DUMITRESCU, C. et al. Methods for reducing energy consumption, optimization in operational data centers. In: IEEE. **2018 International Conference and Exposition on Electrical And Power Engineering (EPE)**. [S.l.], 2018. p. 0483–0486.



ELÉTRICA, A. N. de E. **Primeiro semestre de 2023 tem crescimento de 5,1 GW na matriz elétrica brasileira**. 2023.

<https://www.gov.br/aneel/pt-br/assuntos/noticias/2023/primeiro-semester-de-2023-tem>
Acesso em: 25/07/2023.

Fail2Ban Contributors. **Fail2Ban Official Documentation**. 2024. https://www.fail2ban.org/wiki/index.php/Main_Page. Acesso em: 9 jul. 2025.

FIELDING, R.; KAISER, G. The apache http server project. **IEEE Internet Computing**, v. 1, n. 4, p. 88–90, 1997.

GENG, H. **Data center handbook**. [S.l.]: John Wiley & Sons, 2014.

HAIR, J. F. et al. **Análise multivariada de dados**. [S.l.]: Bookman Editora, 2009.

HASHICORP. **Terraform: What is Terraform?** 2025. Documentação oficial. Disponível em: <https://developer.hashicorp.com/terraform>.

HASHICORP. **What is Infrastructure as Code: What Is It? Why Is It Important?** 2025. Website. Disponível em: <https://www.hashicorp.com/en/resources/what-is-infrastructure-as-code>. Acesso em: 20 set. 2025. Disponível em: <https://www.hashicorp.com/en/resources/what-is-infrastructure-as-code>.

HEIKKINEN, D. The need for green data centers in modern society: Technology, economy and environmental sustainability. **mpra**, 2023.

IIZUKA, E. S.; PEÇANHA, R. S. Revista de gestão ambiental e sustentabilidade–geas.

Instituto de Conservação e Desenvolvimento Sustentável da Amazônia. **Calculadora de Pegada de Carbono**. 2024. <https://idesam.org/calculadora/>. Acesso em: 7 jul. 2025.

International Energy Agency. **Data centres and data transmission networks**. 2023. Online: IEA. Available at: <https://www.iea.org/energy-system/buildings/data-centres-and-data-transmission-networks>. Accessed: 15 Jul. 2025.

International Telecommunication Union. **ITU DataHub - Brazil - Mobile-cellular subscriptions**. 2025. Acesso em: 27 jan. 2025. Disponível em: <https://datahub.itu.int/data/?e=BRA&c=701&i=11624>.

JNR, B. A.; MAJID, M. A.; ROMLI, A. A descriptive study towards green computing practice application for data centers in it based industries. In: EDP SCIENCES. **MATEC web of conferences**. [S.l.], 2018. v. 150, p. 05048.

KOTOWSKI, J.; OKO, J. et al. Green computing and energy storage systems. In: EDP SCIENCES. **E3S Web of Conferences**. [S.l.], 2018. v. 44.

(LABEEE), U. F. d. S. C. Laboratório de E. E. **Usos Finais de Energia Elétrica no Setor Residencial**. 2025. Acesso em: 29 de jan.2026. Disponível em: <https://labeee.ufsc.br/pt-br/node/480>.

LANNELONGUE, L.; GREALEY, J.; INOUYE, M. Green algorithms: quantifying the carbon footprint of computation. **Advanced science**, Wiley Online Library, v. 8, n. 12, p. 2100707, 2021.



LUNARDI, G. L.; SIMÕES, R.; FRIO, R. S. Ti verde: Uma análise dos principais benefícios e práticas utilizadas pelas organizações. **REAd. Revista Eletrônica de Administração (Porto Alegre)**, SciELO Brasil, v. 20, p. 1–30, 2014.

LYKOU, G.; MENTZELIOTI, D.; GRITZALIS, D. A new methodology toward effectively assessing data center sustainability. **Computers e Security**, Elsevier Advanced Technology, v. 76, p. 327–340, 7 2018. ISSN 0167-4048.

MARCO, K. Soluções de TI e eficiência energética para data centers. In: EDITORA CIENTÍFICA DIGITAL. **Tecnologia, Comunicações e Gestão da Informação: Contribuições para o Desenvolvimento de Novos Conhecimentos VOL. 2**. [S.l.], 2024. v. 2, p. 185–197.

MASANET, E.; LEI, N. How much energy do data centers really use. **Aspen Global Change Institute**, 2020.

Ministério da Ciência, Tecnologia e Inovação. **Fator de emissão de CO₂ na geração de energia elétrica no Brasil em 2023 é o menor em 12 anos**. 2024. Notícia no portal Gov.br pelo SIRENE. Publicado em 22 fev. 2024.

Ministério da Ciência, Tecnologia e Inovação. **Fator médio de emissão de CO₂ para inventários corporativos – Sistema Interligado Nacional (SIN)**. 2024. <<https://www.gov.br/mcti/pt-br/acompanhe-o-mcti/cgcl/paginas/fator-medio-inventarios-corporativos>>. Acesso em: jul. 2025.

Ministério de Minas e Energia. **MME identifica crescimento acelerado na demanda de energia elétrica para projetos de data centers**. 2025. Acesso em: 13 fev. 2025. Disponível em: <<https://www.gov.br/mme/pt-br/assuntos/noticias/mme-identifica-crescimento-acelerado-na-demanda-de-energia-eletrica-para-projetos-de-data-centers>>.

MUPPALA, P. K. Sustainable devops: Minimizing the carbon footprint of banking data centers. **International Journal of Science and Research Archive**, International Journal of Science and Research Archive, v. 14, n. 1, p. 1780–1793, 2025.

National Oceanic and Atmospheric Administration (NOAA). **Carbon Toolkit - Basics**. 2022. <https://gml.noaa.gov/education/carbon_toolkit/basics.html>. Acesso em: 24 jul. 2023.

NIEDERAUER, J. **PHP para quem conhece PHP**. [S.l.]: Novatec Editora, 2017.

ONOJA, M. O.; ONYENZE, C. C.; AKINTOYE, A. A. Devops and sustainable software engineering: Bridging speed, reliability, and environmental responsibility. **International Journal of Technology Management & Humanities**, 2024.

OPENAI. **ChatGPT: ferramenta de inteligência artificial**. 2025. Disponível em: <<https://chat.openai.com/>>. Acesso em: 6 dez. 2025.

Organização das Nações Unidas. **Transformando Nosso Mundo: A Agenda 2030 para o Desenvolvimento Sustentável**. 2015. Acesso em: 4 fev. 2025. Disponível em: <<https://sdgs.un.org/goals>>.

PALMA, S. D.; NUCCI, D. D.; TAMBURRI, D. A. Ansiblemetrics: A python library for measuring infrastructure-as-code blueprints in ansible. **SoftwareX**, Elsevier, v. 12, p. 100633, 2020.

- PATTERSON, D. et al. The carbon footprint of machine learning training will plateau, then shrink. **Computer**, IEEE, v. 55, n. 7, p. 18–28, 2022.
- PEDRONI, A. Z.; COELHO, B. N. Estudo de requisitos para datacenters e salas industriais para sistemas de tecnologia e automação. **Refas-Revista Fatec Zona Sul**, v. 10, n. 1, p. 1–24, 2023.
- PRASANNA, T.; SINGH, K. P. Improving energy efficiency in cloud data centers through an brownout software system strategy based on containers. **eurchembull.com**, 2023.
- Progress Chef. **Chef Overview**. 2025. Documentação oficial. Disponível em: https://docs.chef.io/chef_overview/.
- RAHMAN, A.; MAHDAVI-HEZAVEH, R.; WILLIAMS, L. A systematic mapping study of infrastructure as code research. **Information and Software Technology**, Elsevier, v. 108, p. 65–77, 2019.
- Red Hat, Inc. **Red Hat Enterprise Linux 6: Guia de Segurança — IPTables**. Red Hat, Inc., 2013. Disponível em: https://docs.redhat.com/pt-br/documentation/red_hat_enterprise_linux/6/html/security_guide/sect-security_guide-iptables. Acesso em: 21 abr. 2025. Disponível em: https://docs.redhat.com/pt-br/documentation/red_hat_enterprise_linux/6/html/security_guide/sect-security_guide-iptables.
- REDDY, V. D. et al. Best practices for sustainable datacenters. **IT Professional**, v. 20, n. 5, p. 57–67, 2018.
- REDHAT. **O que é a automação de TI?** 2018. <https://www.redhat.com/pt-br/topics/automation/whats-it-automation>. Acessado em 28/12/2023.
- REDHAT. **Introdução ao DevOps**. 2022. <https://www.redhat.com/pt-br/topics/devops#:~:text=DevOps%20%C3%A9%20uma%20abordagem%20de,r%C3%A1pidas%20e%20de%20alta%20qualidade.> Acesso em: 10 jun. 2023.
- Salt Project. **Salt Project Documentation: Overview**. 2025. Documentação oficial. Disponível em: <https://docs.saltproject.io/en/latest/>.
- SARAIWA, G. et al. Estimating CO₂ emissions of distributed applications and platforms with simgrid/batsim. **arXiv preprint arXiv:2508.13693**, 2025.
- SHAMBAUGH, R.; WEISS, A.; GUHA, A. Rehearsal: A configuration verification tool for puppet. In: **Proceedings of the 37th ACM SIGPLAN Conference on Programming Language Design and Implementation**. [S.l.: s.n.], 2016. p. 416–430.
- SILVA, L. C. S. d. et al. Uma análise de publicações brasileiras sobre green it na base de dados web of science. In: UNIVERSIDADE FEDERAL DE GOIÁS, REGIONAL CATALÃO. **Anais do II Simpósio de Engenharia de Produção (SIENPRO)**. Catalão, GO, Brasil, 2018. Acesso em: dd mmm. aaaa. Disponível em: <https://sienpro.catalao.ufg.br/p/25912-anais-do-ii-sienpro-2018-issn-2594-410x>.
- UDDIN, M.; RAHMAN, A. A. Energy efficiency and low carbon enabler green it framework for data centers considering green metrics. **Renewable and Sustainable Energy Reviews**, Elsevier, v. 16, n. 6, p. 4078–4094, 2012.
- UNIDAS, O. das N. **Agenda 2030 para o Desenvolvimento Sustentável**. 2015. Disponível em: <https://brasil.un.org/pt-br/sdgs>. Acesso em: 20 out. 2025.

United Nations Framework Convention on Climate Change. **Conference of the Parties (COP)**. 2024. Acesso em: 29 de jan. 2026. Disponível em: <https://unfccc.int/process/bodies/supreme-bodies/conference-of-the-parties-cop>.

VERAS, M. **Arquitetura de Nuvem (AWS): Amazon Web Services**. [S.l.]: Brasport, 2013.

VERTIV. **Words to go: 6 key data center sustainability metrics**.

2023. Disponível em: <https://www.techtarget.com/searchdatacenter/feature/Words-to-go-6-key-data-center-sustainability-metrics>. Acesso em: 15 jul.

2025. Disponível em: <https://www.techtarget.com/searchdatacenter/feature/Words-to-go-6-key-data-center-sustainability-metrics>.

WATSON, R. T.; BOUDREAU, M.-C.; CHEN, A. J. Information systems and environmentally sustainable development: energy informatics and new directions for the is community. **MIS quarterly**, JSTOR, p. 23–38, 2010.

WRI Brasil. **Os países que mais emitiram gases de efeito estufa**. WRI Brasil, 2025.

Acessado em: 03 fev. 2025. Disponível em: <https://www.wribrasil.org.br/noticias/os-paises-que-mais-emitiram-gases-de-efeito-estufa>.

XU, M.; BUYYA, R. Brownoutcon: A software system based on brownout and containers for energy-efficient cloud computing. **Journal of Systems and Software**, Elsevier, v. 155, p. 91–103, 2019.

ZHANG, Y.; LIU, J. Prediction of overall energy consumption of data centers in different locations. **Sensors**, v. 22, n. 10, 2022. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/22/10/3704>.

ÖZDOĞAN, E.; CERAN, O.; ÜSTÜNDAĞ, M. T. Systematic analysis of infrastructure as code technologies. **Gazi University Journal of Science, Part A: Engineering and Innovation**, v. 10, n. 4, p. 452–471, 2023.



A APÊNDICE - SCRIPTS ANSIBLE PARA CONFIGURAÇÃO

A.1 *Playbook* de Transferência e Restauração do Banco de Dados OJS

```

\label{lst:playbook-bd}
---
- name: Transferencia de Banco de Dados OJS
  hosts: hajaes
  become: true
  vars_files:
    - vars.yml
  tasks:
    - name: Verificar se o usuario portalseer ja existe
      shell: |
        sudo mysql -NBe "SELECT EXISTS(SELECT 1 FROM mysql.user WHERE user = '{{
          db_user }}')"
      register: usuario_existe_raw
      changed_when: false

    - name: Definir variavel booleana com base no resultado da consulta
      set_fact:
        usuario_existe: "{{ usuario_existe_raw.stdout == '1' }}"

    - name: Criar usuario do banco se nao existir
      shell: |
        sudo mysql -e "CREATE USER '{{ db_user }}'@'localhost' IDENTIFIED BY '{{
          db_pass }}'; GRANT ALL PRIVILEGES ON *.* TO '{{ db_user
          }}'@'localhost'; FLUSH PRIVILEGES;"
      when: not usuario_existe

- name: Dump e transferencia com medicao
  hosts: localhost
  gather_facts: false
  vars_files:
    - vars.yml
  vars:
    dump_path: /home/ansible/dump_db.sql
    log_file: /home/ansible/copia_banco.log
    time_file: /home/ansible/copia_banco_time.log
    perf_file: /home/ansible/copia_banco_perf.log
    remote_path: /home/ansible/dump_db.sql
  tasks:
    - name: Carregar informacoes do host ajaes

```



```

set_fact:
  ajaes_host: "{{ hostvars['ajaes'] }}"

- name: Carregar informacoes do host hajaes
  set_fact:
    hajaes_host: "{{ hostvars['hajaes'] }}"

- name: Registrar horario de inicio da copia
  shell: 'date "+[INFO] Inicio da copia do banco: %Y-%m-%d %H:%M:%S" >> "{{
    log_file }}"'

- name: Executar dump e transferencia com perf e time
  shell: >
    echo "[INFO] Iniciando copia com perf e time..." | tee -a "{{ log_file
    }}"
    perf stat -o "{{ perf_file }}" -- /usr/bin/time -v -o "{{ time_file }}"
    bash -c '
    ssh -p {{ ajaes_host.ansible_port }} {{ ajaes_host.ansible_user }}@{{
    ajaes_host.ansible_host }} "
    mysqldump --quick --single-transaction -u{{ db_user }} -p"{{ db_pass
    }}\" {{ db_name }} > {{ dump_path }} &&
    sed -i \"s/utf8mb4_0900_ai_ci/utf8mb4_general_ci/g\" {{ dump_path }}
    &&
    chmod 644 {{ dump_path }}"
    &&
    scp -P {{ ajaes_host.ansible_port }} {{ ajaes_host.ansible_user }}@{{
    ajaes_host.ansible_host }}:{{ dump_path }} {{ dump_path }} &&
    scp {{ dump_path }} {{ hajaes_host.ansible_user }}@{{
    hajaes_host.ansible_host }}:{{ remote_path }}'

args:
  executable: /bin/bash

- name: Registrar horario de fim da copia
  shell: 'date "+[INFO] Fim da copia do banco: %Y-%m-%d %H:%M:%S" >> "{{
    log_file }}"'

- name: Restaurar dump no hajaes com medicao
  hosts: localhost
  gather_facts: false
  vars_files:
    - vars.yml
  vars:
    restore_log: /home/ansible/restauracao_banco.log
    restore_time: /home/ansible/restauracao_time.log
    restore_perf: /home/ansible/restauracao_perf.log
    remote_path: /home/ansible/dump_db.sql
  tasks:

```



```

- name: Carregar informacoes do host hajaes
  set_fact:
    hajaes_host: "{{ hostvars['hajaes'] }}"

- name: Registrar horario de inicio da restauracao
  shell: 'date "+[INFO] Inicio da restauracao: %Y-%m-%d %H:%M:%S" >> "{{
    restore_log }}"'

- name: Executar restauracao com perf e time via SSH
  shell: >
    echo "[INFO] Iniciando restauracao com perf e time..." | tee -a "{{
      restore_log }}";
    perf stat -o "{{ restore_perf }}" -- /usr/bin/time -v -o "{{ restore_time
      }}" \
    ssh {{ hajaes_host.ansible_user }}@{{ hajaes_host.ansible_host }} "
      sudo mysql -e 'CREATE DATABASE IF NOT EXISTS {{ db_name }} CHARACTER
        SET utf8mb4 COLLATE utf8mb4_general_ci;' &&
      sudo mysql {{ db_name }} < {{ remote_path }}"
  args:
    executable: /bin/bash

- name: Registrar horario de fim da restauracao
  shell: 'date "+[INFO] Fim da restauracao: %Y-%m-%d %H:%M:%S" >> "{{
    restore_log }}"'

- name: Remover dump da maquina origem (ajaes)
  hosts: ajaes
  become: true
  tasks:
    - name: Apagar dump em ajaes
      file:
        path: /home/ansible/dump_db.sql
        state: absent

```

Listing A.1 – Playbook Ansible para transferência e restauração do banco de dados OJS

A.2 *Playbook Ansible* para instalação da pilha LAMP

```

---
- name: Instalar a pilha LAMP com medicao de tempo e desempenho
  hosts: hajaes
  become: yes
  vars:
    metricas_dir: "/root/lamp_aut"
    timestamp: "{{ lookup('pipe', 'date +%Y-%m-%d_%H-%M-%S') }}"
    perf_file: "{{ metricas_dir }}/metricas_{{ timestamp }}.perf"

```



```

time_file: "{{ metricas_dir }}/metricas_{{ timestamp }}.txt"

pre_tasks:
  - name: Criar diretorio para salvar metricas
    ansible.builtin.file:
      path: "{{ metricas_dir }}"
      state: directory
      mode: '0755'

  - name: Marcar horario local de inicio
    ansible.builtin.shell: date '+%Y-%m-%d %H:%M:%S'
    register: inicio_local

  - name: Exibir horario local de inicio
    ansible.builtin.debug:
      msg: "Inicio da instalacao (horario local): {{ inicio_local.stdout }}"

tasks:
  - name: Instalar pacotes LAMP com perf e time
    ansible.builtin.shell: >
      echo "# started on $(date -d '{{ inicio_local.stdout }}' '+%a %b %d
        %H:%M:%S %Y')" > {{ perf_file }} && \
      /usr/bin/time -v -o {{ time_file }} \
      perf stat -a -r 1 -o - \
      apt install -y apache2 mariadb-server mariadb-client \
      php8.1 php8.1-cli php8.1-common php8.1-mysql \
      php8.1-xml php8.1-mbstring php8.1-curl php8.1-zip \
      php8.1-soap php8.1-intl php8.1-gd libapache2-mod-php8.1 \
      >> {{ perf_file }} 2>&1
    args:
      executable: /bin/bash

post_tasks:
  - name: Marcar horario local de fim
    ansible.builtin.shell: date '+%Y-%m-%d %H:%M:%S'
    register: fim_local

  - name: Exibir horario local de fim
    ansible.builtin.debug:
      msg: "Fim da instalacao (horario local): {{ fim_local.stdout }}"

```

Listing A.2 – Playbook Ansible para instalação da pilha LAMP

A.3 Playbook Ansible para configurar segurança com medição automatizada (iptables + fail2ban)



```
---
- name: Configurar segurança com medicao automatizada (iptables + fail2ban)
  hosts: hajaes
  become: true
  gather_facts: true

  vars:
    metricas_dir: "/root/seg-aut"
    timestamp: "{{ lookup('pipe', 'date +%Y-%m-%d_%H-%M-%S') }}"
    perf_file: "{{ metricas_dir }}/metricas_{{ timestamp }}.perf"
    time_file: "{{ metricas_dir }}/metricas_{{ timestamp }}.txt"
    log_file: "{{ metricas_dir }}/regras_aplicadas_{{ timestamp }}.txt"

  tasks:

    - name: Criar diretorio para salvar as metricas
      file:
        path: "{{ metricas_dir }}"
        state: directory
        mode: '0755'

    - name: Marcar horario local de inicio
      shell: date "+%Y-%m-%d %H:%M:%S"
      register: hora_inicio

    - name: Exibir horario local de inicio
      debug:
        msg: "Inicio da configuracao (horario local): {{ hora_inicio.stdout }}"

    - name: Escrever cabecalho inicial no arquivo .perf
      copy:
        dest: "{{ perf_file }}"
        content: |
          # started on {{ hora_inicio.stdout }}

          Performance counter stats for firewall setup:

    - name: Executar medicao com perf e time
      shell: |
        /usr/bin/time -v -o "{{ time_file }}" \
        perf stat -a -r 1 -o - bash -c '
          apt update
          apt install -y iptables iptables-persistent fail2ban

          iptables -I INPUT 1 -s 192.0.101.2 -j ACCEPT
          iptables -F
```



```
iptables -X

iptables -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
iptables -A INPUT -p tcp --dport 22 -s 192.0.100.0/24 -j ACCEPT
iptables -A INPUT -p tcp --dport 22 -s 192.0.101.0/24 -j ACCEPT
iptables -A INPUT -p tcp --dport 22 -s 192.0.0.281 -j ACCEPT
iptables -A INPUT -p tcp --dport 22 -j DROP
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -p tcp --dport 443 -j ACCEPT
iptables -A INPUT -p icmp --icmp-type echo-request -s 192.0.100.0/24 -j
ACCEPT
iptables -A INPUT -p icmp --icmp-type echo-request -s 192.0.101.0/24 -j
ACCEPT
iptables -A INPUT -p tcp --syn -j LOG --log-prefix "IPTABLES TCP SCAN: "
iptables -A INPUT -p udp -j LOG --log-prefix "IPTABLES UDP SCAN: "
iptables -A INPUT -p icmp -j LOG --log-prefix "IPTABLES ICMP SCAN: "
iptables -A INPUT -p tcp --dport 3306 -s 192.0.100.0/24 -j ACCEPT
iptables -A INPUT -p tcp --dport 3306 -s 192.0.101.0/24 -j ACCEPT
iptables -A INPUT -p tcp --dport 3306 -j DROP
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT

iptables-save > /etc/iptables/rules.v4
systemctl restart netfilter-persistent

echo "[sshd]" > /etc/fail2ban/jail.local
echo "enabled = true" >> /etc/fail2ban/jail.local
echo "port = ssh" >> /etc/fail2ban/jail.local
echo "filter = sshd" >> /etc/fail2ban/jail.local
echo "logpath = /var/log/auth.log" >> /etc/fail2ban/jail.local
echo "maxretry = 3" >> /etc/fail2ban/jail.local

echo "[apache]" >> /etc/fail2ban/jail.local
echo "enabled = true" >> /etc/fail2ban/jail.local
echo "port = http,https" >> /etc/fail2ban/jail.local
echo "filter = apache-auth" >> /etc/fail2ban/jail.local
echo "logpath = /var/log/apache2/error.log" >> /etc/fail2ban/jail.local
echo "maxretry = 3" >> /etc/fail2ban/jail.local

systemctl enable fail2ban
systemctl restart fail2ban

iptables -L -n -v > {{ log_file }}
' >> "{{ perf_file }}" 2>&1

- name: Marcar horario local de fim
```



```

shell: date "+%Y-%m-%d %H:%M:%S"
register: hora_fim

- name: Exibir horario local de fim
  debug:
    msg: "Fim da configuracao (horario local): {{ hora_fim.stdout }}"

- name: Adicionar linha de final da execucao ao .perf
  lineinfile:
    path: "{{ perf_file }}"
    line: "# finished on {{ hora_fim.stdout }}"
    insertafter: EOF

```

Listing A.3 – Playbook Ansible para configurar segurança com medição automatizada (iptables + fail2ban)

A.4 Playbook de Backup, atualização e restauração do sistema *Open Journal System*

```

---
- name: "Atualizao Completa do OJS (v3.4.0-2)"
  hosts: hajaes
  become: true
  gather_facts: false

  vars:
    # --- Configuraes ---
    versao_ojs: "3.4.0-2"
    destino_ojs: "/var/www/html"
    diretorio_arquivos_ojs: "/usr/local/ojs-files"
    usuario_web: "www-data"

    # --- Host de Origem ---
    host_origem: "ajaes"
    usuario_ssh_origem: "root"
    porta_ssh_origem: 2738

    # --- Nomes Fixos e Previsveis ---
    dir_temp_remoto: "/var/tmp"
    dir_temp_local: "/tmp"
    nome_arquivo_backup: "persistentes_v3.4.tgz"
    caminho_backup_local: "{{ dir_temp_local }}/{{ nome_arquivo_backup }}"
    dir_restore_remoto: "{{ dir_temp_remoto }}/restore_manual"

  pre_tasks:
    - name: "Registrar tempo de inicio"

```



```
run_once: true
delegate_to: localhost
block:
  - name: "Coletar tempo (epoch) do controller"
    command: date +%s
    register: start_epoch_cmd
    changed_when: false
  - name: "Definir tempo de inicio"
    set_fact:
      start_epoch: "{{ start_epoch_cmd.stdout }}"
- name: "Exibir Horario de Incio"
  run_once: true
  delegate_to: localhost
  debug:
    msg: "Inicio do Playbook: {{ lookup('pipe', 'date +%Y-%m-%d %H:%M:%S') }}"

tasks:
- name: "FASE I: Backup dos Dados Persistentes"
  delegate_to: localhost
  run_once: true
  shell: |
    set -o pipefail
    sudo ssh -p {{ porta_ssh_origem }} {{ usuario_ssh_origem }}@{{
      host_origem }} \
    "export TMPDIR={{ dir_temp_remoto }} && sudo tar czf - -C /var/www/html
      public -C /usr/local --exclude='ojs-files/usageStats' ojs-files -C
      /var/www/html config.inc.php" \
    > {{ caminho_backup_local }}
  args:
    executable: /bin/bash

- name: "FASE II: Preparao do Ambiente de Destino"
  shell: |
    cd {{ dir_temp_remoto }}
    wget -q https://pkp.sfu.ca/ojs/download/ojs-{{ versao_ojs }}.tar.gz
    sudo rm -rf {{ destino_ojs }}
    sudo tar -xzf ojs-{{ versao_ojs }}.tar.gz -C /var/www
    sudo mv /var/www/ojs-{{ versao_ojs }} {{ destino_ojs }}
  args:
    executable: /bin/bash

- name: "FASE III: Transferncia do Backup"
  delegate_to: localhost
  run_once: true
  command: >
```

```

sudo scp {{ caminho_backup_local }} ansible@{{ inventory_hostname }}:{{
    dir_temp_remoto }}/

- name: "FASE IV: Restauração dos Arquivos"
  shell: |
    sudo rm -rf {{ dir_restore_remoto }} && sudo mkdir {{ dir_restore_remoto
    }}
    sudo tar xzf {{ dir_temp_remoto }}/{{ nome_arquivo_backup }} -C {{
    dir_restore_remoto }}/
    sudo rsync -a --delete {{ dir_restore_remoto }}/public/ {{ destino_ojs
    }}/public/
    sudo rsync -a --delete {{ dir_restore_remoto }}/ojs-files/ {{
    diretorio_arquivos_ojs }}/
    sudo cp {{ dir_restore_remoto }}/config.inc.php {{ destino_ojs
    }}/config.inc.php
  args:
    executable: /bin/bash

- name: "Garantir a configuração de e-mail correta"
  blockinfile:
    path: "{{ destino_ojs }}/config.inc.php"
    marker: "; {mark} ANSIBLE MANAGED EMAIL CONFIG"
    insertafter: '\[email\]'
    block: |
      default = sendmail
      sendmail_path = "/usr/sbin/sendmail -bs"

- name: "FASE V: Finalizar a Atualização"
  block:
    - name: "Ajustar permissões e limpar cache"
      shell: |
        sudo chown -R {{ usuario_web }}:{{ usuario_web }} {{ destino_ojs }}
        sudo chown -R {{ usuario_web }}:{{ usuario_web }} {{
        diretorio_arquivos_ojs }}
        sudo rm -rf {{ destino_ojs }}/cache/*.php {{ destino_ojs
        }}/cache/t_compile/*.tpl.php
      args:
        executable: /bin/bash
    - name: "Executar o script de upgrade do OJS"
      command: "sudo -u {{ usuario_web }} php {{ destino_ojs
      }}/tools/upgrade.php upgrade"

post_tasks:
  - name: "Limpeza Final"
    block:
      - name: "Remover arquivos temporários do host de destino"
        file: { path: "{{ item }}", state: absent }

```



```
    loop:
      - "{{ dir_restore_remoto }}"
      - "{{ dir_temp_remoto }}/ojs-{{ versao_ojs }}.tar.gz"
      - "{{ dir_temp_remoto }}/{{ nome_arquivo_backup }}"
    - name: "Remover arquivos temporrios do n de controle"
      delegate_to: localhost
      run_once: true
      file: { path: "{{ caminho_backup_local }}", state: absent }

- name: "Finalizar e registrar tempo total"
  run_once: true
  delegate_to: localhost
  block:
    - name: "Registrar tempo de fim"
      command: date +%s
      register: end_epoch_cmd
      changed_when: false
    - name: "Calcular durao total"
      set_fact:
        total_duration: "{{ end_epoch_cmd.stdout | int - start_epoch | int }}"
    - name: "Exibir tempo final"
      debug:
        msg: ["Fim da execucao: {{ lookup('pipe', 'date +%Y-%m-%d
          %H:%M:%S\\')} }", "Duracao total: {{ total_duration }} segundos."]
```

Listing A.4 – Playbook de Backup, Atualização e restauração do sistema *Open Journal System*

B APÊNDICE – LOG DE EXECUÇÃO DAS CONFIGURAÇÕES

B.1 Atualização de lista de repositórios

```

root@debian:/etc/ansible/inventory# ansible-playbook -i hosts
/etc/ansible/playbooks/1_UpdateLinux.yml

PLAY [Update Rocky Linux Servers - Starting Deploy] *****

TASK [Gathering Facts] *****
ok: [rocky01]

TASK [DNF | Update System] *****
changed: [rocky01]

PLAY [Update Debian Linux Servers - Starting Deploy] *****

TASK [Gathering Facts] *****
ok: [debian01]

TASK [Debian Like | Update System] *****
changed: [debian01]

PLAY RECAP *****
debian01 : ok=2 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
rocky01  : ok=2 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

```

Listing B.1 – Log de execução da atualização lista de repositórios

B.2 Log do Ansible Ping

```

root@debian:/etc/ansible# ansible -m ping all
debian01 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}

rocky01 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },

```



```
"changed": false,  
"ping": "pong"  
}
```

Listing B.2 – Log de execução de ping

B.3 Log de Transferência e Restauração do Banco de Dados OJS

```
PLAY [Garantir que o usuario do banco exista em hajaes] *****  
  
TASK [Gathering Facts] *****  
[WARNING]: Platform linux on host hajaes is using the discovered Python  
interpreter at /usr/bin/python3.11, but future installation  
of another Python interpreter could change the meaning of that path. See  
https://docs.ansible.com/ansible-  
core/2.17/reference_appendices/interpreter_discovery.html for more information.  
ok: [hajaes]  
  
TASK [Verificar se o usuario do banco ja existe] *****  
ok: [hajaes]  
  
TASK [Definir variavel booleana com base no resultado da consulta] *****  
ok: [hajaes]  
  
TASK [Criar usuario do banco se nao existir] *****  
skipping: [hajaes]  
  
PLAY [Dump e transferencia com medicao] *****  
  
TASK [Carregar informacoes do host ajaes] *****  
ok: [localhost]  
  
TASK [Carregar informacoes do host hajaes] *****  
ok: [localhost]  
  
TASK [Registrar horario de inicio da copia] *****  
changed: [localhost]  
  
TASK [Executar dump e transferencia com perf e time] *****  
changed: [localhost]  
  
TASK [Registrar horario de fim da copia] *****  
changed: [localhost]
```



```

PLAY [Restaurar dump no hajaes com medicaao] *****

TASK [Carregar informacoes do host hajaes] *****
ok: [localhost]

TASK [Registrar horario de inicio da restauracao] *****
changed: [localhost]

TASK [Executar restauracao com perf e time via SSH] *****
changed: [localhost]

TASK [Registrar horario de fim da restauracao] *****
changed: [localhost]

PLAY [Remover dump da maquina origem (ajaes)] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host ajaes is using the discovered Python
interpreter at /usr/bin/python3.10, but future installation of
another Python interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-
core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [ajaes]

TASK [Apagar dump em ajaes] *****
changed: [ajaes]

PLAY RECAP *****
ajaes : ok=2 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
hajaes : ok=3 changed=0 unreachable=0 failed=0 skipped=1 rescued=0 ignored=0
localhost : ok=9 changed=6 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

```

Listing B.3 – Log de Transferência e Restauração do Banco de Dados OJS

B.4 Log de execução da instalação da pilha LAMP

```

PLAY [Instalar a pilha LAMP com medicaao de tempo e desempenho] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host hajaes is using the discovered Python
interpreter at /usr/bin/python3.11, but future installation of another Python
interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-core/2.17/reference_appendices/
interpreter_discovery.html for more information.

```



```

ok: [hajaes]

TASK [Criar diretorio para salvar metricas] *****
ok: [hajaes]

TASK [Marcar horario local de inicio] *****
changed: [hajaes]

TASK [Exibir horario local de inicio] *****
ok: [hajaes] => {
  "msg": "Inicio da instalacao (horario local): 2025-06-10 08:47:56"
}

TASK [Instalar pacotes LAMP com perf e time] *****
changed: [hajaes]

TASK [Marcar horario local de fim] *****
changed: [hajaes]

TASK [Exibir horario local de fim] *****
ok: [hajaes] => {
  "msg": "Fim da instalacao (horario local): 2025-06-10 08:48:40"
}

PLAY RECAP *****
hajaes : ok=7 changed=3 unreachable=0 failed=0
        skipped=0 rescued=0 ignored=0

```

Listing B.4 – Log de execução da instalação da pilha LAMP

B.5 Log de configuração de segurança com medição automatizada (iptables + fail2ban)

```

PLAY [Configurar seguranca com medicao automatizada (iptables + fail2ban)] ***

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host hajaes is using the discovered Python
interpreter at /usr/bin/python3.11, but future installation of another Python
interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-
core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [hajaes]

TASK [Criar diretorio para salvar as metricas] *****
ok: [hajaes]

```



```

TASK [Marcar horario local de inicio] *****
changed: [hajaes]

TASK [Exibir horario local de inicio] *****
ok: [hajaes] => {
  "msg": "Inicio da configuracao (horario local): 2025-06-11 17:38:17"
}

TASK [Escrever cabecalho inicial no arquivo .perf] *****
changed: [hajaes]

TASK [Executar medicao com perf e time] *****
changed: [hajaes]

TASK [Marcar horario local de fim] *****
changed: [hajaes]

TASK [Exibir horario local de fim] *****
ok: [hajaes] => {
  "msg": "Fim da configuracao (horario local): 2025-06-11 17:38:22"
}

TASK [Adicionar linha de final da execucao ao .perf] *****
changed: [hajaes]

PLAY RECAP *****
hajaes : ok=9 changed=5 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

```

Listing B.5 – Log de configuração de segurança com medição automatizada (iptables + fail2ban)

B.6 Log de e Backup, atualização e restauração do sistema Open Journal System

```

sudo ansible-playbook ojs-3.4.0-2.yml

PLAY [Atualizacao Completa do OJS (v3.4.0-2 - Metodo Automatizado)] *****

TASK [Coletar tempo (epoch) do controller] *****
ok: [hajaes -> localhost]

TASK [Definir registro tempo de incio] *****
ok: [hajaes -> localhost]

TASK [Exibir Horario de Inicio] *****
ok: [hajaes -> localhost] => {

```



```
"msg": "Inicio do Playbook: 2025-06-20 22:42:05"
}

TASK [FASE I: Backup dos Dados Persistentes] *****
changed: [ajaes -> localhost]

TASK [FASE II: Preparacao do Ambiente de Destino] *****
[WARNING]: Platform linux on host hajaes is using the discovered Python
interpreter at /usr/bin/python3.11, but future installation of another Python
interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-
core/2.17/reference_appendices/interpreter_discovery.html for more information.
changed: [hajaes]

TASK [FASE III: Transferencia do Backup] *****
changed: [hajaes -> localhost]

TASK [FASE IV: Restauracao dos Arquivos] *****
changed: [hajaes]

TASK [Garantir a configuracao de e-mail correta] *****
changed: [hajaes]

TASK [Ajustar permissoes e limpar cache] *****
changed: [hajaes]

TASK [Executar o script de upgrade do OJS] *****
changed: [hajaes]

TASK [Remover arquivos temporarios do host de destino] *****
changed: [hajaes] => (item=/var/tmp/restore_manual)
changed: [hajaes] => (item=/var/tmp/ojs-3.4.0-2.tar.gz)
changed: [hajaes] => (item=/var/tmp/persistentes_3.4.tgz)

TASK [Remover arquivos temporarios do no de controle] *****
changed: [hajaes -> localhost]

TASK [Registrar tempo de fim] *****
ok: [hajaes -> localhost]

TASK [Calcular durao total] *****
ok: [hajaes -> localhost]

TASK [Exibir tempo final] *****
ok: [hajaes -> localhost] => {
  "msg": [
    "Fim da execucao: 2025-06-20 22:55:21",
```



```
"Durao total: 796 segundos."
]
}

PLAY RECAP *****
hajaes : ok=15 changed=9 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

Listing B.6 – Log de Backup, atualização e restauração do sistema Open Journal System

