



UNIVERSIDADE FEDERAL DO RIO GRANDE - FURG
CENTRO DE CIÊNCIAS COMPUTACIONAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO
CURSO DE MESTRADO EM ENGENHARIA DE COMPUTAÇÃO

CUSTO ENERGÉTICO EM ANÁLISES DE BIOINFORMÁTICA

FABRÍCIO LUCAS LOPES

Dissertação de Mestrado apresentado ao Programa de Pós-Graduação em Computação da Universidade Federal do Rio Grande - FURG, para a obtenção do grau de Mestre em Engenharia de Computação

Orientador:
Coorientadora:

Prof. Dr. Adriano Velasque Werhli
Prof. Dr^a. Karina dos Santos Machado

RIO GRANDE - 2026

L864c

Lopes, Fabrício Lucas

Custo energético em análises de bioinformática / Fabrício Lucas Lopes. -
2026.
113f.

Dissertação (Mestrado) – Universidade Federal do Rio Grande –
Programa de Pós-Graduação em Computação, 2026.

Orientador: Dr. Adriano Velasque Werhli

Coorientadora: Dr^a. Karina dos Santos Machado

1. Computação. 2. Bioinformática. 3. Consumo de energia. 4. Simulação
computacional. 5. Computação Verde. 6. IoT. Objetivos para
desenvolvimento sustentável. I. Werhli, Adriano Velasque. II. Machado,
Karina dos Santos. III. Título.

CDU 004

CUSTO ENERGÉTICO EM ANÁLISES DE BIOINFORMÁTICA

FABRÍCIO LUCAS LOPES

Banca Examinadora

Prof. Dr. Adriano Velasque Verhli

Prof. Dr. Vinicius Menezes de Oliveira

Prof. Dr. Marilton Sanchotene de Aguiar

*Dedico este trabalho a todos os professores e cientistas que trabalham
na busca para que a humanidade tenha a Luz.*

AGRADECIMENTOS

Quero agradecer a minha esposa Adriana por sempre ter me encorajado a buscar a excelência e a superar meus próprios limites. Por ser meu porto seguro durante todo o processo de elaboração deste trabalho. Seu amor e comprometimento com nossa relação foram um estímulo na minha dedicação ao trabalho. Aos nossos filhos, Pedro e Raíssa, presentes de Deus em nossas vidas, motivo de conquistas dos objetivos traçados. Aos meus pais que me ensinaram a importância da disciplina, do esforço e da dedicação, e que me apoiaram em todas as escolhas que fiz durante minha jornada, e seu exemplo de vida, é minha inspiração e motivação para buscar sempre o melhor. Por fim, gostaria de agradecer ao Corpo Docente do Centro de Ciências Computacionais da Universidade Federal do Rio Grande, por criar e incentivar um ambiente Técnico-Colaborativo, e a todos os colegas do Programa de Pós-Graduação do C3.

"A Ciência é uma equação diferencial. Religião é a condição de contorno!"
ALAN TURING

RESUMO

LOPES, Fabricio Lucas. CUSTO ENERGÉTICO EM ANÁLISES DE BIOINFORMÁTICA. 2026. 113 f. Dissertação de Mestrado – Esta dissertação investiga o custo operacional relativo ao consumo de energia elétrica em infraestruturas computacionais (CPUs e GPUs) durante simulações de Bioinformática. O trabalho fundamenta-se no conceito de "Computação Verde", buscando quantificar o consumo de energia (em kWh ou kJ) por volume de dado processado (Mb ou Gb). A pesquisa justifica-se pela crescente demanda por poder computacional em simulações biomoleculares e pela necessidade de alinhar a prática científica aos Objetivos de Desenvolvimento Sustentável (ODS) da Agenda 2030 da ONU. A evolução dos microchips e o desenvolvimento das GPUs proporcionaram que a simulação dos comportamentos que ocorrem na natureza, como a estrutura de proteínas e simulações de macromoléculas por dinâmica molecular. As simulações computacionais demandam altos custos de equipamentos e de horas de programadores, e esses custos podem ser determinados através dos valores econômicos gastos com o custeio de equipamentos e do material humano utilizado. O objetivo geral do trabalho 'quantificar o consumo de energia elétrica no sequenciamento de genomas, identificando as principais etapas de consumo energético, dando subsídios para definir estratégias para a sua redução. A justificativa para a pesquisa inclui a relevância das simulações biomoleculares, que são intensivas em termos computacionais e exigem recursos significativos, impactando o meio ambiente, alinhando o presente trabalho com o conceito de "*Computação Verde*". A metodologia experimental envolveu o desenvolvimento de um sistema embarcado de monitoramento em tempo real, composto por uma placa Arduino Uno e um sensor PZEM-004T, validado em equipamentos domésticos antes da aplicação laboratorial. Foram realizados experimentos de sequenciamento genético com tecnologia de nanoporos (Oxford Nanopore Technology) e simulações de genomas completos de organismos procariontes, como *Haemophilus Haemolyticus*, *Staphylococcus Aureus* e *Stenotrophomonas Maltophilia*. Para o desenvolvimento do trabalho, valeu-se de uma busca por fontes acadêmicas, que possibilitaram a definição da pergunta de pesquisa, busca nas bases de dados bibliográficas, seleção dos estudos incluídos na busca e análise das evidências científicas relevantes sobre o consumo de energia elétrica em simulações computacionais em bioinformática, no entanto tal procura mostrou uma lacuna importante no tema. A aplicação do Método de Pareto revelou que a etapa de Montagem (software Flye) é o maior gargalo de no consumo de energia em todos os organismos analisados, representando, junto ao Basecalling, a maior parte do impacto no gasto energético. Apesar dos resultados promissores, o estudo apresenta limitações quanto à diversidade de arquiteturas de hardware testadas e à exclusão de variáveis ambientais externas que podem influenciar o consumo térmico. Como trabalhos futuros, propõe-se a expansão da análise para ambientes de computação em nuvem e o desenvolvimento de modelos preditivos baseados em aprendizado de máquina para estimar o custo energético antes da execução dos pipelines. O trabalho conclui que a eficiência na bioinformática contemporânea depende da convergência entre o monitoramento granular de hardware e a reengenharia algorítmica, estabelecendo a taxa de eficiência por megabyte (kJ/Mb) como uma métrica essencial para uma ciência sustentável.

Palavras-chave: Bioinformática, Consumo de energia, Simulação Computacional, Computação Verde, IoT, Objetivos para Desenvolvimento Sustentável.

ABSTRACT

LOPES, Fabricio Lucas. ENERGY COST IN BIOINFORMATICS ANALYSIS. 2026. 113 p. Master's Dissertation – Graduate Program in Computing. Federal University of Rio Grande - FURG, Rio Grande. This dissertation investigates the operational cost related to electricity consumption in computational infrastructures (CPUs and GPUs) during bioinformatics simulations. The work is based on the concept of "Green Computing," seeking to quantify energy consumption (in kWh or kJ) per volume of data processed (Mb or Gb). The research is justified by the growing demand for computational power in biomolecular simulations and the need to align scientific practice with the Sustainable Development Goals (SDGs) of the UN's 2030 Agenda. The evolution of microchips and the development of GPUs have enabled the simulation of behaviors that occur in nature, such as protein structure and macromolecule simulations through molecular dynamics. Computational simulations demand high equipment and programmer hour costs, and these costs can be determined through the economic values spent on equipment and human resources. The overall objective of this work is to quantify the electrical energy consumption in genome sequencing, identifying the main energy-consuming steps and providing support for defining strategies to reduce it. The rationale for the research includes the relevance of biomolecular simulations, which are computationally intensive and require significant resources, impacting the environment, aligning this work with the concept of "Green Computing". The experimental methodology involved the development of an embedded real-time monitoring system, composed of an Arduino Uno board and a PZEM-004T sensor, validated on household equipment before laboratory application. Genetic sequencing experiments were performed using nanopore technology (Oxford Nanopore Technology) and simulations of complete genomes of prokaryotic organisms, such as *Haemophilus Haemolyticus*, *Staphylococcus Aureus*, and *Stenotrophomonas Maltophilia*. For the development of this work, a search for academic sources was conducted, which allowed for the definition of the research question, a search in bibliographic databases, selection of studies included in the search, and analysis of relevant scientific evidence on electrical energy consumption in computational simulations in bioinformatics. However, this search revealed a significant gap in the field. The application of the Pareto Method revealed that the Assembly stage (Flye software) is the biggest bottleneck in energy consumption in all organisms analyzed, representing, along with Basecalling, the largest part of the impact on energy expenditure. Despite the promising results, the study presents limitations regarding the diversity of hardware architectures tested and the exclusion of external environmental variables that can influence thermal consumption. As future work, it is proposed to expand the analysis to cloud computing environments and to develop predictive models based on machine learning to estimate the energy cost before the execution of the pipelines. The study concludes that efficiency in contemporary bioinformatics depends on the convergence between granular hardware monitoring and algorithmic re-engineering, establishing the efficiency rate per megabyte (kJ/Mb) as an essential metric for sustainable science.

Keywords: Bioinformatics, Energy Consumption, Computer Simulation, Green Computing, IoT, Sustainable Development Goals.

LISTA DE FIGURAS

Figura 1 – 17 Objetivos para o Desenvolvimento Sustentável	23
Figura 2 -- Diagrama Arduino Uno x Sensor PZEM-004T	44
Figura 3 -- Protótipo Arduino Uno x Sensor PZEM-004T	54
Figura 4 – Pegada de Carbono – CPU	68
Figura 5 – Uso de Memória – CPU	69
Figura 6 – Pegada de CO ₂ por local – CPU	69
Figura 7 – Pegada de Carbono – GPU	66
Figura 8 – Uso de memória- GPU	70
Figura 9 – Pegada de CO ₂ por local – GPU	70
Figura 10 – Pegada de Carbono - Haemophilus Haemolyticus	77
Figura 11 – Uso de Memória - Haemophilus Haemolyticus	77
Figura 12 – Pegada de CO ₂ por Local - Haemophilus Haemolyticus	78
Figura 13 – Pegada de Carbono - Staphylococcus Aureus	80
Figura 14 – Uso de Memória - Staphylococcus Aureus	81
Figura 15 – Pegada de CO ₂ por Local - Staphylococcus Aureus	81
Figura 16 – Pegada de Carbono - Stenotrophomonas Maltophilia	83
Figura 17 – Uso de Memória - Stenotrophomonas Maltophilia	84
Figura 18 – Pegada de CO ₂ por Local - Stenotrophomonas Maltophilia	84
Figura 19 – Taxa de Eficiência Energética	87
Gráfico 1 - Laptop Samsung Intel(R) Core(TM)	58
Gráfico 2 - Chaleira Elétrica - Cadence Infusion - 127V 1200W	59
Gráfico 3 - Refrigerador Frostfree Eletrolux - DF52X - 127V 1200W	59
Gráfico 4 - Consumo de Energia Oxford Nanopore Technology (CPU)	64
Gráfico 5 - Consumo de Energia Oxford Nanopore Technology (CPU)	64
Gráfico 6 - Consumo de Energia Oxford Nanopore Technology (GPU)	65
Gráfico 7 - Consumo de Energia Oxford Nanopore Technology (GPU)	65
Gráfico 8 - Comparativo do Consumo de Energia CPU x GPU	66
Gráfico 9 - Comparativo do Consumo de Energia CPU x GPU	67'
Gráfico 10 – Energia Consumida – Ambiente Computacional	67
Gráfico 11 – Eficiência Energética - Haemophilus Haemolyticus	76
Gráfico 12 – Eficiência Energética - Staphylococcus Aureus	79
Gráfico 13 – Eficiência Energética - Stenotrophomonas Maltophilia	82
Gráfico 14 – Comparativo por Etapa de Processamento	86

LISTA DE FIGURAS

Tabela 1 -- Laptop Samsung Intel(R) Core(TM)	58
Tabela 2 -- Chaleira Elétrica - Cadence Infusion - 127V 1200W	58
Tabela 3 -- Refrigerador Frostfree Eletrolux - DF52X - 127V 1200W	59
Tabela 4 - Consumo de Energia Oxford Nanopore Technology (CPU)	64
Tabela 5 - Consumo de Energia Oxford Nanopore Technology (GPU)	65
Tabela 6 – Comparativo do Consumo de Energia CPU x GPU	66
Tabela 7 – Energia Consumida – Ambiente Computacional	67
Tabela 8 – Organismo Haemophilus Haemolyticus	73
Tabela 9 – Organismo Staphylococcus Aureus	73
Tabela 10 – Organismo Stenotrophomonas Maltophilia	74
Tabela 11 – Resumo Simulação - Haemophilus Haemolyticus	75
Tabela 12 – Resumo Simulação - Staphylococcus Aureus	78
Tabela 13 – Resumo Simulação - Stenotrophomonas Maltophilia	82
Tabela 14 – Comparativo em kJ/Mb	86

LISTA DE ABREVIATURAS E SIGLAS

A	Ampere (Unidade de corrente elétrica SI)
A/D	Analógico/Digital (conversor)
ABNT	Associação Brasileira de Normas Técnicas
AC	Corrente Alternada (<i>Alternating Current</i>)
API	Interface de Programação de Aplicações (<i>Application Programming Interface</i>)
BUSCO	<i>Benchmarking Universal Single-Copy Orthologs</i>
C3	Centro de Ciências Computacionais
CO_{2e}	Dióxido de Carbono emitido (ou Kg CO _{2e} / g CO _{2e})
CI	Circuito Integrado
Comb-Lab	Laboratório de Bioinformática
CPD	Centro de Processamento de Dados
CPS	Sistemas Ciber-Físicos (<i>Cyber-Physical Systems</i>)
CPU	Unidade Central de Processamento (<i>Central Processing Unit</i>)
CUDA	<i>Compute Unified Device Architecture</i>
DC	Corrente Contínua (<i>Direct Current</i>)
DNA	Ácido Desoxirribonucleico
FASTQ	Formato de arquivo para sequências biológicas e pontuações de qualidade
FP	Fator de Potência
FURG	Universidade Federal do Rio Grande
GEE	Gases de Efeito Estufa
GPU	Unidade de Processamento Gráfico (<i>Graphic Processing Unit</i>)
GWAS	Estudo de Associação Genômica (<i>Genome-Wide Association Study</i>)
HAC	Alta Acurácia (High Accuracy)
HPC	Computação de Alto Desempenho (<i>High Performance Computing</i>)
I/O	Entrada/Saída (Input/Output)
IA	Inteligência Artificial
IDE Environment)	Ambiente de Desenvolvimento Integrado (Integrated Development
IEA	Agência Internacional de Energia
IEEE	Institute of Electrical and Electronics Engineers

LISTA DE ABREVIATURAS E SIGLAS

IoT	Internet das Coisas (<i>Internet Of Things</i>)
IPEA	Instituto de Pesquisa e Economia Aplicada
k	Quilo (10^3 , unidades de grandeza SI)
kJ	Quilojoules (unidade de consumo de energia, SI)
kWh	Quilowatt hora (unidade de consumo de energia, SI)
J	Joule (unidade de energia consumida, SI)
LTS	Suporte de Longo Prazo (<i>Long Term Support</i>)
Mb	Megabytes
NGS	Sequenciamento de Nova Geração (<i>Next Generation Sequencing</i>)
ODS	Objetivos para o Desenvolvimento Sustentável
ONU	Organização das Nações Unidas
PUE	Eficácia do Uso de Energia (<i>Power Usage Effectiveness</i>)
QUAST	<i>Quality Assessment Tool for Genome Assemblies</i>
RAPL <i>Limit</i>)	Interface de Limite de Potência Média de Execução (<i>Running Average Power</i>
RNA	Ácido Ribonucleico
RNN	Redes Neurais Recorrentes
SBC	Sociedade Brasileira de Computação
TDP	Energia de Projeto Térmico (<i>Thermal Design Power</i>)
TI	Tecnologia da Informação
TTL	Transistor-Transistor Logic (Níveis Lógicos de Tensão)
USB	Universal Serial Bus
V	Volt (unidade de tensão elétrica, SI)
VRAM	Memória de Vídeo (Video RAM)
W	Watt (unidade de potência elétrica, SI)

SUMÁRIO

1	INTRODUÇÃO	14
1.1	O Tema da Pesquisa e Sua Importâncias.....	14
1.2	Objetivo Geral e Específicos.....	17
1.3	Hipóteses	18
2	FUNDAMENTAÇÃO TEÓRICA	22
2.1	Bioinformática e Simulações Computacionais	26
2.2	Consumo Energético e Computação Verde	31
2.3	Sustentabilidade e os Objetivos de Desenvolvimento Sustentável (ODS)	31
2.4	Desafios Energéticos em Infraestruturas Computacionais	34
3	REVISÃO DA LITERATURA.....	37
4	METODOLOGIA EXPERIMENTAL.....	42
4.1	Desenvolvimento do Sistema de Monitoramento.....	42
4.2	Arquitetura do Protótipo com Arduino e Sensor PZEM-004T.....	53
4.3	Procedimento de Medição e Coleta de Dados	57
4.4	Medições em Equipamentos Domésticos	57
4.5	Análise de dados – Método de Pareto	60
5	RESULTADOS E ANÁLISE COMPARATIVA.....	63
5.1	Experimentos com Sequenciadores Oxford Nanopore	63
5.2	Comparativo de Consumo: CPU vs. GPU	66
5.3	Quantificação do consumo do sistema operacional	67
5.4	Pegada de Carbono e Avaliação com Green Algorithms	68
5.5	Simulação de Organismos Procariontes.....	70
5.5.1	Simulação de Organismo Haemophilus Haemolyticus.....	75
5.5.2	Simulação de Organismo Staphylococcus Aureus.....	78
5.5.3	Simulação de Organismo Stenotrophomonas Maltophilia.....	81
5.5	Discussão dos Resultados.....	85
6	CONCLUSÃO E TRABALHOS FUTUROS.....	89
	REFERÊNCIAS	95
	ANEXO 1	99
	ANEXO 2.....	102
	ANEXO 3.....	110
	ANEXO 4	111

1 INTRODUÇÃO

O presente trabalho visa analisar o consumo de energia elétrica na infraestrutura computacional utilizada em simulações computacionais de Bioinformática. Os custos de consumo de energia nosso objetivo não é quantificar os custos econômicos, pois estes envolvem aspectos de contrato de fornecimento com as concessionárias de energia e seus clientes/consumidores, por exemplo instituições de ensino e pesquisa, ou empresas de processamento de dados. Este trabalho tem por intuito de quantificar o consumo de energia elétrica, em kWh (ou kJ), por dado processado, em Gb (ou Mb).

Além de identificar o quanto estas simulações podem estar em alinhamento com o conceito de Computação Verde, e verificar o quanto o presente trabalho alinha-se com o desenvolvimento sustentável, e a utilização de métodos mais eficientes de análise computacional. Este conceito engloba práticas e tecnologias que visam minimizar o impacto ambiental da computação, incluindo a redução do consumo energético e a otimização da eficiência dos recursos computacionais e seus algoritmos, não estar atento a esta prática, torna a prática das pesquisas genômicas, um prática não sustentável.

1.1 O TEMA DA PESQUISA E SUA IMPORTÂNCIA

A rápida evolução dos microchips, impulsionando o lançamento de Unidades Centrais de Processamento (CPU) mais velozes e robustas, resultou no avanço das Unidades de Processamento Gráfico (GPU). Essa transformação, detalhada por Stallings em Arquitetura e organização de computadores, Stallings (2017), exemplifica a revolução na capacidade computacional e a subsequente evolução dos sistemas de processamento de informações, elementos cruciais para a simulação de fenômenos naturais complexos.

Dentro desse contexto, a biologia molecular utiliza essa capacidade computacional como base para progressos significativos, como o sequenciamento de ácidos nucleicos (RNA e DNA) de microrganismos. Essas análises, essenciais para a compreensão do funcionamento desses sistemas biológicos, são abordadas na obra Bioinformática: da biologia à flexibilidade molecular, de Bruno César Feltes (2015). Nela, Feltes demonstra como a modelagem computacional possibilita a elucidação da complexa arquitetura e da dinâmica molecular, impulsionando o conhecimento em áreas como a genômica e a transcriptômica.

No entanto, a execução de simulações computacionais em bioinformática impõe desafios consideráveis à infraestrutura, principalmente em relação aos custos de aquisição e operação de Centros de Processamento de Dados (CPD), bem como à necessidade de profissionais especializados em desenvolvimento de software. A quantificação desses custos, que abrange o investimento em equipamentos de alto desempenho e em capital humano especializado, mostra-se fundamental para o planejamento e a gestão eficiente na pesquisa científica.

Nesse cenário, emerge uma preocupação urgente, alinhada à agenda global de desenvolvimento sustentável da Organização das Nações Unidas (ONU) em sua Agenda 2030 (2018). A crescente demanda por poder computacional para simulações biomoleculares levanta questionamentos críticos sobre o consumo energético dessas infraestruturas.

Em sintonia com os princípios da "Computação Verde" (Grealey, 2022), torna-se imperativo investigar o consumo de energia em CPD e GPU dedicados à bioinformática, bem como o custo energético específico de simulações de sequenciamento genético ou modelagem molecular. Além disso, questiona-se se a energia utilizada provém de fontes renováveis, mitigando o impacto ambiental, ou de fontes fósseis, contribuindo para as emissões de Gases de Efeito Estufa (GEE).

Neste contexto, o artigo "Pegada de Carbono", explora o conceito associado às simulações em bioinformática e destaca a importância de otimizar os recursos para reduzir os impactos gerados. O autor compara a eficiência energética de diferentes CPD e plataformas de computação, incluindo a computação em nuvem, e discute a influência da localização desses centros na pegada de carbono.

É crucial ressaltar que a pegada de carbono também é impactada pela paralelização, alocação de memória e pela utilização de ferramentas de bioinformática como AutoDock Vina e rDock, assim como pela eficiência de grandes centros de processamento em nuvem, como Google Cloud, Microsoft Azure e Amazon Web Services.

O site www.green-algorithms.org, (www.green-algorithms.org) (2025), ilustra a crescente conscientização sobre a pegada de carbono da computação científica. Ele permite comparar a emissão de dióxido de carbono equivalente (KgCO₂e) entre diferentes abordagens algorítmicas e arquiteturas de hardware (CPU vs. GPU) no processamento de dados genômicos. Essa perspectiva reforça a urgência na adoção de "Algoritmos Verdes" e na otimização do uso de recursos computacionais.

Diante desse panorama, o presente trabalho assume a relevância de analisar o consumo de energia elétrica inerente aos processos de sequenciamento genético e modelagem de genomas. O objetivo principal é identificar os fatores primários que regulam esse consumo, as metodologias usadas para sua mensuração e, de forma crucial, as estratégias inovadoras para sua redução. Nesse sentido, serão investigadas técnicas e abordagens propostas e implementadas para minimizar a demanda energética no sequenciamento genético e como o impacto dessa otimização se manifesta em diversas áreas, desde a engenharia e a física até a biologia.

A obra de Feltes (2015), “Bioinformática: da biologia à flexibilidade molecular”, oferece insights valiosos sobre as abordagens computacionais mais eficientes para a análise de dados biológicos. Nela, a bioinformática é apresentada como uma área interdisciplinar que une biologia, ciência da computação e estatística para a análise e interpretação de dados biológicos, desempenhando um papel crucial na modelagem de estruturas biomoleculares essenciais, como proteínas e ácidos nucleicos.

As simulações computacionais permitem a investigação de processos moleculares complexos, como interações proteína-ligante, dobramento proteico e docagem molecular, contribuindo significativamente para a compreensão de mecanismos de doenças e o desenvolvimento de novas terapias farmacológicas. Dada a natureza intrinsecamente intensiva do uso de recursos computacionais em simulações biomoleculares, frequentemente executadas em clusters de processamento (CPD) e explorando a capacidade paralela das GPU, o custo energético é significativo.

Alinhar as práticas de pesquisa com os Objetivos de Desenvolvimento Sustentável da ONU, na Agenda 2030, exige a priorização de fontes de energia renovável e a implementação de práticas computacionais sustentáveis. A ausência de uma quantificação sistemática do consumo de energia nos CPD dedicados à bioinformática, que pode representar uma parte substancial dos custos totais de pesquisa, impede uma avaliação macroeconômica precisa do impacto energético da infraestrutura de simulações computacionais.

Essa lacuna dificulta a tomada de decisões estratégicas e a promoção de uma ciência mais sustentável, em consonância com os princípios da Agenda 2030. A análise proposta neste trabalho busca preencher essa lacuna, oferecendo uma perspectiva mais clara sobre os desafios energéticos e as oportunidades de otimização no campo da bioinformática computacional, também no que tange trabalhos científicos na utilização de quantificação granular de consumo de energia nas simulações genômicas.

1.2 OBJETIVO GERAL E ESPECÍFICOS

Este estudo se dedica à análise do consumo de energia elétrica em processos de sequenciamento e modelagem de genomas. Para isso, buscamos quantificar o consumo de energia elétrica no sequenciamento de genomas, identificando as principais etapas de consumo energético, dando subsídios para definir estratégias para a sua redução.

A pesquisa tem como objetivos específicos:

- Identificar sistemas embarcados para medição do consumo de energia;
- Analisar quais as fases de montagem do genoma demandam o maior consumo de energia;
- Compreender e determinar qual o maior impacto no consumo de energia elétrica neste tipo de simulação computacional.

A evolução da bioinformática no último século foi impulsionada pelos avanços da Biologia Molecular, desde estudos de eletroforese de proteínas até a elucidação da estrutura da dupla hélice do DNA e o sequenciamento da insulina bovina. Esses progressos culminaram no uso intenso de simulações computacionais, incluindo o emprego de GPUs, que possibilitaram análises mais sofisticadas e precisas. Essa trajetória histórica foi documentada por Jawad (2006) em seu artigo “*The Era of Bioinformatics*”, que traça um panorama desde experimentos de laboratório até as simulações computacionais modernas.

As simulações biomoleculares, por sua natureza intensiva em termos computacionais, exigem recursos significativos, como clusters de processamento (CPU) e GPU, resultando em um alto custo energético, especialmente com o crescimento da demanda por pesquisa em bioinformática. Além disso, o elevado consumo de energia elétrica pode gerar impactos ambientais relevantes.

Nesse contexto, torna-se essencial alinhar as práticas de pesquisa aos Objetivos de Desenvolvimento Sustentável da ONU (2018), promovendo o uso de fontes de energia renovável e a adoção de práticas sustentáveis em simulações.

Apartir desse cenário, buscamos fontes na literatura que nos oferecessem subsídios para:

- Avaliar, de forma abrangente, o consumo de energia elétrica em simulações biomoleculares;
- Identificar práticas computacionais eficientes e sustentáveis no contexto da bioinformática;
- Comparar diferentes abordagens e tecnologias em relação ao seu consumo energético;
- Propor soluções concretas para otimizar o uso de recursos energéticos em simulações biomoleculares.

A definição de objetivos para este trabalho, estabelece a necessidade imperativa de uma bioinformática mais consciente, na qual a precisão biológica não ignore o custo operacional de sua execução. No entanto, para que a transição de um modelo de pesquisa intensivo para um paradigma de 'Computação Verde' seja efetiva, torna-se necessário confrontar a realidade operacional das infraestruturas de processamento de alto desempenho (HPC).

A eficiência energética em bioinformática não é vista apenas como uma métrica de software, mas como uma necessidade emergente da interação entre algoritmos de alinhamento e a arquitetura do hardware subjacente. Nesse sentido, a análise das variáveis que regem o consumo em Centros de Processamento de Dados (CPD) permite fundamentar as suposições técnicas que guiam este estudo. A seguir, apresentam-se as hipóteses que estruturam a viabilidade de uma quantificação granular, essencial para buscar uma transição para a sustentabilidade na simulação genômica.

1.3 HIPÓTESES

O consumo energético em Centros de Processamento de Dados (CPD) constitui um fator determinante nos custos operacionais de pesquisas e simulações computacionais, impactando diretamente a viabilidade econômica e a sustentabilidade dessas atividades. Dada a crescente demanda por processamento de alto desempenho, o gasto energético desses centros tem se tornado um aspecto cada vez mais crítico, tanto do ponto de vista financeiro quanto ambiental. Segundo o trabalho de Volpini, (2018)[36], o consumo de energia em ambientes de processamento de dados massivos

(big data) em nuvem, destacando a relevância do tema devido ao impacto energético dos datacenters, que representam cerca de 2% do consumo global de energia.

O estudo investiga como fatores como virtualização e paralelização afetam o consumo de energia, utilizando uma solução de monitoramento chamada Seshat para coletar métricas de desempenho e consumo energético. O estudo conclui que a escolha da configuração ideal para o processamento de dados massivos em nuvem não é trivial e sugere a realização de novos testes com diferentes algoritmos, “virtualizadores” e formas de armazenamento para explorar soluções mais eficientes em termos de consumo energético e desempenho.

A ferramenta Seshat é uma arquitetura de monitoramento escalável projetada especificamente para ambientes em nuvem, com foco particular em infraestruturas virtualizadas de processamento de dados massivos (big data). Cuja sua finalidade, é de coletar métricas de desempenho e consumo energético dos recursos computacionais em tempo real.

Permite a quantificação do consumo de energia em um ambiente dinâmico de nuvem, que é crucial para gerenciar custos e implementar políticas de eficiência energética. Coleta dados que permitem correlacionar o desempenho de aplicações com o consumo de energia, ajudando a identificar a configuração ideal para o balanceamento entre desempenho e sustentabilidade. A arquitetura do Seshat é projetada para ser elástica, escalável e resiliente, sendo capaz de lidar com a complexidade e o volume de dados gerados em sistemas de nuvem, mesmo com o comportamento dinâmico de recursos virtualizados.

A ausência de uma quantificação precisa do consumo energético nesses ambientes impede a formulação de uma estimativa macroeconômica abrangente, dificultando a implementação de políticas de eficiência energética e sustentabilidade. Sem dados concretos, instituições de pesquisa e empresas de tecnologia enfrentam desafios na otimização de seus recursos computacionais, limitando a adoção de estratégias para redução do impacto ambiental e dos custos operacionais.

Para mitigar esse problema, é essencial investir em metodologias e ferramentas que permitam a análise detalhada do consumo energético nos CPDs, considerando fatores como hardware utilizado, algoritmos empregados e padrões de uso dos recursos computacionais.

Além disso, iniciativas voltadas para a incorporação de tecnologias mais eficientes, como servidores de baixo consumo e sistemas de resfriamento otimizados, podem contribuir significativamente para a redução do gasto energético sem comprometer o desempenho das operações. Uma abordagem multidisciplinar envolvendo especialistas em ciência da computação, engenharia elétrica e sustentabilidade pode fornecer insights valiosos sobre medidas que conciliem eficiência computacional e responsabilidade ambiental.

Dessa forma, o estudo aprofundado do consumo energético nos CPD não apenas favorece uma melhor gestão de custos, mas também alinha os avanços tecnológicos às metas globais de redução da pegada de carbono e uso consciente de energia.

Este trabalho, portanto, investiga o alto custo de consumo de energia elétrica em infraestruturas computacionais (CPD, CPU e GPU) utilizadas para simulações de bioinformática, como sequenciamento e modelagem de genomas. Diante da crescente demanda computacional e da importância de alinhar a pesquisa científica com metas de sustentabilidade, como a Agenda 2030 da ONU e os princípios da "Computação Verde", o estudo se propõe a analisar os fatores que influenciam esse gasto energético.

Com objetivo de monitorar o consumo de energia elétrica através de um sistema embarcado para a medição do consumo e através deste passo estar alinhado com o princípio dos "Algoritmos Verdes" e a "Green Computing", otimizando o uso de recursos e reduzindo a pegada de carbono associada a essas atividades, preenchendo a lacuna na quantificação sistemática desse impacto e promovendo uma ciência mais sustentável. Contribuindo para consolidação de Políticas Públicas para mitigar o consumo de energia elétrica indo ao encontro do postulado por Flora (2013), visando alinhamento com a Política Nacional de Eficiência Energética.

A crescente demanda por processamento de alto desempenho em áreas como a bioinformática exige uma transformação urgente nas práticas de gestão energética dos CPD. Conforme destacado, a falta de quantificação precisa do consumo energético é uma barreira para a implementação de políticas de eficiência e sustentabilidade. A mitigação eficaz desse problema reside em uma abordagem multidisciplinar que combina monitoramento avançado, otimização de hardware e a adoção de "Algoritmos Verdes".

A lacuna na quantificação precisa do consumo energético pode ser preenchida pelo investimento em novas metodologias de monitoramento. A integração de Sistemas Embarcados e da Internet das Coisas (IoT) é crucial. Sensores de baixo custo e alto desempenho podem ser instalados em nível de rack, servidor, e até mesmo componente

(CPU/GPU) para coletar métricas de consumo energético e desempenho em tempo real. Isso permite uma granularidade de dados muito superior aos métodos tradicionais, facilitando a identificação de hotspots de ineficiência.

Ferramentas de análise de dados, como o Seshat, aliadas aos dados da IoT, podem mapear a relação direta entre fatores como virtualização, paralelização e o consumo de energia. Essa análise detalhada é o alicerce para a tomada de decisões informadas e a otimização dinâmica de recursos computacionais.

A infraestrutura física é um vetor primário de consumo energético, demandando a incorporação de tecnologias mais eficientes, como a adoção de servidores energeticamente eficientes (como os que utilizam arquiteturas otimizadas e processadores de baixo consumo por tarefa - low-power), é fundamental. A escolha de hardware deve ser baseada não apenas na capacidade de processamento, mas também na métrica de desempenho por watt.

O resfriamento de sistemas computacionais pode representar até 40% do consumo total de um CPD. Estratégias como a utilização de corredores quentes/frios, refrigeração líquida (liquid cooling) e a exploração do clima externo (free cooling) reduzem drasticamente a dependência de sistemas de ar-condicionado tradicionais, contribuindo significativamente para a redução do PUE (Power Usage Effectiveness).

Assim a eficiência não é apenas uma questão de hardware; ela também reside no software e nas metodologias de processamento. A pesquisa e implementação de "Algoritmos Verdes" visam a otimização da eficiência do código para que as tarefas computacionais sejam concluídas com o menor número de ciclos de processamento possível e, conseqüentemente, com o mínimo consumo de energia. Isso inclui o refinamento de algoritmos de scheduling, gerenciamento de memória e otimização de I/O (Entrada/Saída), especialmente em simulações intensivas como as de bioinformática.

A computação em nuvem oferece a oportunidade de consolidar cargas de trabalho e alocar recursos de forma dinâmica (ligando e desligando clusters sob demanda). Esta abordagem, potencializada pela virtualização, permite que os recursos de computação sejam utilizados de forma mais eficiente, reduzindo o desperdício de energia em períodos de baixa demanda.

Ao integrar o monitoramento granular da IoT, o uso de hardware eficiente e a aplicação de "Algoritmos Verdes", as instituições de pesquisa e empresas de tecnologia podem alinhar os avanços tecnológicos com os princípios da "Computação Verde" e as metas de sustentabilidade globais, como a Agenda 2030 da ONU, promovendo uma ciência mais sustentável e economicamente viável.

Nossa hipótese baseia-se no monitoramento granular correlacionado ao ajuste dinâmico de parâmetros de paralelização e virtualização em algoritmos de alinhamento de sequências genômicas pode permitir a redução no consumo energético por giga-base processado, sem comprometer a acurácia biológica ou exceder o tempo limite de execução (deadline) das simulações.

Este estudo estabelece pretende fornecer subsídios de que a eficiência na bioinformática contemporânea depende da convergência entre o monitoramento granular de hardware e a otimização algorítmica. Todavia, para que tal investigação transcenda o domínio puramente técnico, e adquira relevância institucional, torna-se imperativo fundamentar essas proposições nas diretrizes globais de desenvolvimento sustentável.

A transição de um modelo computacional intensivo para uma infraestrutura de 'Computação Verde' não constitui apenas um desafio de engenharia, mas um compromisso ético alinhado às metas internacionais de preservação ambiental e eficiência energética. Nesse sentido, a seção a seguir detalha o embasamento teórico que sustenta esta pesquisa, conectando as demandas de processamento de alto desempenho aos Objetivos de Desenvolvimento Sustentável (ODS) da Agenda 2030, estabelecendo o nexó necessário entre a inovação tecnológica e a responsabilidade socioambiental.

2 FUNDAMENTAÇÃO TEÓRICA

Esta pesquisa alinha suas investigações e simulações computacionais com os princípios delineados no relatório "ODS – Metas Nacionais dos Objetivos de Desenvolvimento Sustentável" ONU (2018), que articula a agenda 2030 para o desenvolvimento sustentável no contexto brasileiro. Os 17 Objetivos para o Desenvolvimento Sustentável (ODS) são enfatizados na figura abaixo:

Figura 1 – 17 Objetivos para o Desenvolvimento Sustentável



Fonte: ODS – IPEA em 2018

A relação entre Computação Verde, Bioinformática, IoT e Consumo de Energia com o ODS 3 – Assegurar a Vida Saudável e Promover o Bem-estar de Todos, em Todas as Idades é bidirecional e crítica. Onde a Bioinformática e a IoT apoiam diretamente o ODS 3 ao acelerar a pesquisa em saúde e melhorar a assistência médica. A Computação Verde mitiga o impacto negativo do alto consumo de energia e das emissões de carbono geradas por essas tecnologias, garantindo que o progresso na saúde seja sustentável e não comprometa o meio ambiente (o que afeta indiretamente a saúde humana).

A Bioinformática é essencial para atingir diversas metas do ODS 3. Nas simulações de genomas e a modelagem molecular de vírus e bactérias (como as que demandam alto processamento em CPDs) aceleram a descoberta de novos medicamentos, vacinas e o diagnóstico rápido de doenças transmissíveis e negligenciadas. A análise de dados massivos (big data) genômicos e clínicos permite o desenvolvimento da medicina personalizada, que oferece tratamentos mais eficazes e com menos efeitos colaterais.

A IoT, dispositivos vestíveis e sensores ambientais monitoram a saúde de pacientes remotamente, permitindo a manutenção preventiva, o diagnóstico precoce e a gestão de doenças crônicas, ampliando o acesso à saúde de qualidade. Sensores de IoT monitoram a qualidade do ar, da água e do solo, fornecendo dados em tempo real. Isso permite que órgãos de saúde pública tomem medidas rápidas para reduzir a exposição da população a produtos químicos perigosos e poluentes, que são causas diretas de muitas doenças.

O avanço da Bioinformática e da IoT exige um volume colossal de processamento, elevando o consumo de energia dos CPDs (que já representam cerca de 2% do consumo global, com alguns especialistas projetando um aumento significativo). Esse alto consumo gera emissões de gases de efeito estufa e poluição que, paradoxalmente, prejudicam a saúde humana e o bem-estar – indo contra o espírito do ODS 3.

A relação entre tecnologia e sustentabilidade energética no contexto do ODS 7 (Energia Acessível e Limpa), destacando o papel dual da tecnologia. O processamento intensivo (HPC) para Bioinformática e as redes de IoT impulsionam o alto consumo de energia nos CPD. Se essa energia for cara e não renovável, ela ameaça a acessibilidade e a segurança energética. A Computação Verde (com servidores, refrigeração e Algoritmos Verdes) promove a eficiência energética nos CPD, reduzindo a demanda e aliviando a pressão sobre a rede elétrica.

A IoT é essencial para a criação de Redes Inteligentes (Smart Grids), que otimizam a distribuição, equilibram a carga e gerenciam as fontes renováveis, melhorando a eficiência em toda a rede. A Bioinformática acelera a pesquisa e otimização de organismos para a produção de biocombustíveis e outras fontes de energia renovável. Em essência, a Computação Verde e a IoT são cruciais para transformar a tecnologia de um grande consumidor de energia para um otimizador de eficiência energética fundamental, apoiando o acesso à energia limpa e sustentável para todos.

A relação entre Computação Verde, Bioinformática, IoT e Consumo de Energia com o ODS 9 – Indústria, Inovação e Infraestrutura é extremamente forte, pois essas tecnologias representam tanto a infraestrutura digital moderna quanto os principais motores da inovação e da industrialização sustentável. O ODS 9 busca desenvolver infraestruturas de qualidade, confiáveis e resilientes. IoT e Resiliência de Infraestrutura: A IoT é fundamental para a gestão de infraestruturas físicas (transporte, utilidades, cidades inteligentes). Sensores monitoram o desempenho, preveem falhas e otimizam a manutenção em tempo real, tornando essas infraestruturas mais resilientes e eficientes. Os Centros de Processamento de Dados (CPD) são a espinha dorsal da infraestrutura digital. A Computação Verde garante a resiliência dessa infraestrutura.

O monitoramento de consumo e desempenho (via IoT/Sistemas Embarcados) permite a identificação precoce de sobrecargas e ineficiências. A eficiência energética (Algoritmos Verdes e refrigeração otimizada) reduz o risco de superaquecimento e falhas,

aumentando a confiabilidade e o tempo de atividade (uptime), essenciais para a infraestrutura moderna. A Bioinformática e o uso de HPC (Computação de Alto Desempenho) são cruciais para o aumento da pesquisa científica e o fomento à inovação. Ao processar dados massivos (genômicos, proteômicos), impulsiona inovações em saúde, novos materiais e agricultura. O baixo custo energético proporcionado pela Computação Verde torna o acesso ao HPC mais acessível e inclusivo, permitindo que mais instituições, independentemente do seu poder financeiro, participem da P&D,

A IoT promove a inovação ao coletar dados que permitem novas formas de gestão e otimização, como o desenvolvimento de Smart Grids e fábricas inteligentes. O ODS 9 exige a atualização da indústria para torná-la sustentável, aumentando a eficiência no uso de recursos. A Computação Verde e a IoT são ferramentas primárias para cumprir essa meta. IoT na Indústria 4.0: Sensores IoT monitoram o consumo de energia e materiais nas fábricas, permitindo o ajuste fino dos processos e a maximização da eficiência de recursos. O alto consumo energético dos CPD (que suportam a Indústria 4.0) deve ser mitigado. A Computação Verde garante que a infraestrutura digital que sustenta a industrialização seja ecologicamente responsável, reduzindo a pegada de carbono e o desperdício de energia. A Bioinformática contribui para o desenvolvimento de processos industriais mais limpos, como a biorremediação e a produção de materiais via biotecnologia.

A relação entre Computação Verde, Bioinformática, IoT e Consumo de Energia com o ODS 13, estabelece a conexão entre tecnologia e ação climática onde o alto consumo energético dos CPD (para Bioinformática e IoT) é uma importante fonte de emissões de GEE. A Computação Verde vem como uma medida urgente para mitigação, utilizando Algoritmos Verdes, hardware eficiente e monitoramento preciso para reduzir a pegada de carbono da infraestrutura digital. A utilização de IoT é a principal ferramenta de monitoramento climático em tempo real, fornecendo dados cruciais para modelos preditivos e Sistemas de Alerta Precoce contra eventos extremos. A Bioinformática utiliza HPC para desenvolver culturas agrícolas mais resistentes a condições climáticas extremas e modelar a adaptação de ecossistemas, apoiando a adaptação e a pesquisa.

2.1- Bioinformática e Simulações Computacionais

A vertiginosa evolução dos microchips, culminando na sofisticação das Unidades de Processamento Gráfico (GPUs), conforme detalhado por Stallings (2017), revolucionou a capacidade de simular fenômenos naturais complexos. Na seara da biologia molecular, essa capacidade computacional tem sido fundamental para avanços significativos, como o sequenciamento de ácidos nucleicos (RNA e DNA) de microrganismos.

Essas análises, cruciais para a compreensão do funcionamento intrínseco desses sistemas biológicos, são minuciosamente exploradas na obra seminal de Bruno César Feltes, "Bioinformática: da biologia à flexibilidade molecular". Feltes (2015) elucida como a modelagem computacional permite desvendar a intrincada arquitetura e dinâmica molecular, impulsionando o conhecimento em áreas como a genômica e a transcriptômica.

O campo do sequenciamento de nova geração (NGS) tem testemunhado avanços significativos, com novas tecnologias e metodologias a surgir continuamente. A literatura recente destaca a rápida evolução e adoção generalizada do NGS na pesquisa biológica de Satam (2023). Estes avanços impulsionam novas tendências na análise genômica, facilitando estudos cada vez mais complexos e em larga escala.

À medida que as plataformas de sequenciamento produzem volumes cada vez maiores de dados, cresce a importância de uma análise de dados eficiente e robusta. Este aspecto é fundamental para extrair informações significativas da saída bruta do sequenciamento e garantir a precisão das aplicações a jusante. A compreensão do contexto histórico e da progressão das tecnologias de sequenciamento oferece perspectiva sobre as capacidades atuais. A inovação contínua nos métodos de sequenciamento assenta numa base de investigação prévia e de marcos tecnológicos.

O sequenciamento por nanopore representa uma abordagem transformadora, oferecendo vantagens únicas no comprimento das leituras e na aquisição de dados em tempo real. As estratégias bioinformáticas associadas aos dados de nanopore, bem como a gama crescente de aplicações, estão bem documentadas em estudos recentes, como nos relata Wang (2021).

Inovações em ferramentas de basecalling baseadas em redes neurais melhoraram notavelmente o desempenho e a precisão do sequenciamento Oxford Nanopore. Análises comparativas destas ferramentas demonstram avanços na

conversão de dados de sinal bruto em sequências de nucleotídeos, algo essencial para a ampla adoção em ambientes de investigação e clínicos mencionados por Wick (2019).

O processo de basecalling é intrínseco ao funcionamento de sequenciadores, como os da Oxford Nanopore Technologies. Nestes sistemas, a molécula de DNA é forçada a passar por um poro de dimensão nanométrica (nanoporo). A passagem de cada base nitrogenada altera a corrente elétrica que atravessa o poro de forma distinta, gerando um sinal elétrico único para Adenina (A), Timina (T), Citosina (C) e Guanina (G).

O basecalling é executado por um software que interpreta essas variações de sinal elétrico. Por meio de algoritmos complexos, frequentemente baseados em redes neurais e aprendizagem profunda, o software associa cada padrão de sinal elétrico a uma base específica, traduzindo os dados de corrente elétrica em uma sequência genética (por exemplo, "ACGTGCAG...").

A relevância do basecalling para o consumo energético reside na sua elevada intensidade computacional. A tradução dos sinais elétricos em sequências de bases exige um poder de processamento considerável, tornando essa etapa um dos principais gargalos no processamento de dados em genômica.

Para otimizar o processo e alcançar alta precisão, o basecalling frequentemente emprega o uso de unidades de processamento gráfico (GPUs), que são conhecidas por seu alto consumo de energia. Adicionalmente, a capacidade de realizar o basecalling em tempo real, enquanto o sequenciamento ocorre, representa um avanço tecnológico, mas demanda um fluxo contínuo de recursos computacionais, impactando diretamente o consumo energético.

A pesquisa em bioinformática tem se dedicado à otimização do basecalling, visando o desenvolvimento de algoritmos mais eficientes e a utilização de hardware especializado. O objetivo central é reduzir o tempo de processamento e, por consequência, o consumo de energia, promovendo a sustentabilidade e acessibilidade da análise genômica.

A Oxford Nanopore Technologies disponibiliza ferramentas de software que gerenciam o processo de sequenciamento e basecalling. Entre as mais conhecidas estão:

- **MinKNOW:** Este é o software principal que controla o dispositivo de sequenciamento. Ele gerencia o fluxo da amostra, a aquisição de dados e, dependendo da configuração de hardware, também pode realizar o basecalling em tempo real. O MinKNOW serve como a interface principal

de onde o usuário inicia, monitora e gerencia o experimento de sequenciamento.

- Guppy: Por um longo período, o Guppy foi o software padrão de basecalling da Oxford Nanopore. Sendo um kit de ferramentas de processamento de dados, ele converte os sinais brutos em sequências de bases. Embora tenha sido um avanço significativo, atualmente é considerado uma ferramenta legada, pois a Oxford Nanopore recomenda o uso do Dorado, seu mais novo basecaller, que promete maior precisão e eficiência. O Guppy é executado via linha de comando, proporcionando flexibilidade para “bioinformaticistas”.

O processo de basecalling é notavelmente intensivo em termos de computação, principalmente devido a duas razões, o volume massivo de dados e os Algoritmos complexos de aprendizado profundo.

Com relação ao volume massivo de dados, cada molécula de DNA que passa pelo nanoporo gera milhares de amostras de sinais elétricos por segundo. Um único experimento de sequenciamento pode produzir centenas de gigabytes de dados brutos que exigem processamento. O software de basecalling precisa lidar com esse fluxo contínuo e volumoso de dados em tempo real.

Já valendo-se de algoritmos complexos de aprendizado profundo, para converter os sinais elétricos em bases com alta precisão, as ferramentas de basecalling empregam modelos de redes neurais recorrentes (RNN) e redes neurais convolucionais (CNN). Esses modelos de aprendizado de máquina são complexos e demandam um poder de processamento massivo para realizar as inferências necessárias a partir de cada sinal. Eles são treinados com grandes volumes de dados para reconhecer padrões sutis, e a execução desses modelos para cada base da sequência consome muitos recursos da CPU e, principalmente, da GPU.

Para superar esse gargalo computacional, as plataformas de sequenciamento modernas, como as da Oxford Nanopore, frequentemente integram Unidades de Processamento Gráfico (GPUs) diretamente no hardware do sequenciador. As GPUs são projetadas para processar tarefas paralelas de forma eficiente, o que as torna ideais para a natureza do basecalling, permitindo a tradução dos dados em tempo real, um recurso crucial para diagnósticos e vigilância genômica.

Na pesquisa de bioinformática é de fundamental importância entender o que é o “Oxford Nanopore Technologies (ONT)”, o qual passou por uma transição tecnológica do

“Guppy”, hoje considerado como um produto “Legacy” (Legado), enquanto o atual o “Dorado” é a ferramenta atual para basecalling em produção atual. Comparando as versões, Guppy, onde sua versão final é a de v6.5.7 é amplamente citada em artigos até 2024, e suporta arquivos tipo .fast5. Já o Dorado, em produção, com sua versão v1.x, substitui o anterior, com arquivos gerados tipo .pod5, integrando modelos de metilação e duplex.

O Dorado simplificou a interface de utilização em comparação com o Guppy, automatizando a seleção de modelos com base nos metadados dos arquivos analisados. Com relação a acurácia, ambos utilizam três níveis, O Fast (com menos carga computacional e menos acurácia), o HAC (High Accuracy, com equilíbrio entre a carga computacional e acurácia, sendo ideal para a maioria das aplicações) e SUP (Super Accuracy, com máxima acurácia e alta carga computacional, exigindo GPUs potentes para o processamento.

A entrada de metadados no Guppy é através de arquivos no formato .fast5, o Dorado também pode ler este tipo de arquivo, no entanto faz uso preferencial de arquivos tipo .pod5 para leitura de dados em paralelo, otimizando o processamento, tornando o Dorado significativamente mais rápido que o Guppy em uma configuração de hardware equivalente. Para o processamento utilizando o Guppy, era necessário o uso de modelos específicos para cada parametrização ou ferramentas externas (Remora). Já no Dorado o processamento é mais simples via flags simples como a modified-bases. Modelos de processamento mais recentes do Dorado (v5.2.0) alcançam acurácia em uma única leitura superior a 99%, muito superior que as versões finais do Guppy.

Entretanto, a sofisticação dessas simulações computacionais em bioinformática impõe demandas substanciais em termos de infraestrutura, notadamente no que concerne aos custos de aquisição e operação de Centros de Processamento de Dados (CPDs) e na expertise de profissionais de desenvolvimento de software. A quantificação precisa desses custos, abrangendo tanto o investimento em equipamentos de alta performance quanto o capital humano especializado, é essencial para o planejamento e a gestão eficiente da pesquisa científica.

O presente trabalho, portanto, assume a relevância de analisar o consumo de energia elétrica inerente aos processos de sequenciamento genético e modelagem de genomas. O objetivo central reside em identificar os fatores primários que modulam esse consumo, as metodologias empregadas para sua mensuração e, crucialmente, as estratégias inovadoras para sua redução.

Nesse sentido, investigando quais as técnicas e abordagens têm sido propostas e implementadas para minimizar a demanda energética no sequenciamento genético, e como o impacto dessa otimização se manifesta em diversas áreas de aplicação, desde a engenharia e a física até a própria biologia. A obra de Feltes (2015) pode fornecer insights valiosos sobre as abordagens computacionais mais eficientes para a análise de dados biológicos.

A bioinformática, como disciplina interdisciplinar que integra a biologia, a ciência da computação e a estatística para a análise e interpretação de dados biológicos, desempenha um papel catalisador na modelagem de estruturas biomoleculares essenciais, como proteínas e ácidos nucleicos. As simulações computacionais possibilitam a investigação de processos moleculares complexos – interações proteína-ligante, dobramento proteico, docagem molecular – contribuindo de forma indelével para a compreensão de mecanismos de doenças e o desenvolvimento de novas terapias farmacológicas.

Contudo, a natureza intrinsecamente intensiva em recursos computacionais das simulações biomoleculares, frequentemente executadas em clusters de processamento (CPDs) e utilizando a capacidade paralela das GPUs, implica um custo energético significativo. Dada a crescente demanda por pesquisa em bioinformática, essa consideração energética transcende o mero aspecto econômico, adquirindo uma dimensão ambiental crucial. Alinhar as práticas de pesquisa com a Agenda 2030, exige a priorização de fontes de energia renovável e a implementação de práticas computacionais sustentáveis.

A parcela do consumo de energia nos CPDs dedicados à bioinformática pode representar uma fração considerável dos custos totais de pesquisa. A ausência de uma quantificação sistemática desse consumo impede uma avaliação macroeconômica precisa do impacto energético da infraestrutura de simulações computacionais, um aspecto fundamental para a tomada de decisões estratégicas e para a promoção de uma ciência mais sustentável, em consonância com os princípios delineados na Agenda 2030. A análise proposta neste trabalho busca mitigar essa lacuna, oferecendo uma perspectiva mais clara sobre os desafios energéticos e as oportunidades de otimização no campo da bioinformática computacional.

2.2- Consumo Energético e Computação Verde

Como mencionado anteriormente, a trajetória da evolução tecnológica, marcada pela miniaturização de componentes eletrônicos e o advento de Unidades de Processamento Gráfico (GPUs) de alta performance, revolucionou a capacidade de modelar e simular processos biológicos complexos em escala computacional. Contudo, este avanço impõe uma responsabilidade intrínseca de alinhamento com os princípios do consumo consciente de energia elétrica e a agenda dos Objetivos de Desenvolvimento Sustentável (ODS) da ONU.

O presente trabalho dedica-se à investigação e à análise aprofundada do consumo energético em simulações computacionais, reconhecendo-as como um pilar fundamental da bioinformática moderna. O objetivo é identificar os vetores primários que modulam esse consumo, bem como examinar as metodologias de quantificação e as estratégias inovadoras para sua mitigação.

2.3- Sustentabilidade e os Objetivos de Desenvolvimento Sustentável (ODS)

A investigação do consumo energético em simulações computacionais aplicadas à bioinformática é essencial para otimizar a eficiência dos processos, reduzir custos operacionais e promover a sustentabilidade, reforçando a necessidade de uma abordagem consciente e estratégica para o uso responsável dos recursos computacionais.

A evolução da eletrônica, impulsionada pela miniaturização de componentes, levou à invenção dos microchips e, posteriormente, ao desenvolvimento de Unidades de Processamento Gráfico (GPUs) de alto desempenho. Essa revolução tecnológica aprimorou significativamente a modelagem computacional de fenômenos biológicos complexos, permitindo avanços na elucidação da estrutura tridimensional de proteínas e na simulação de interações moleculares com RNA e DNA (docking), conforme demonstrado nos trabalhos de Feltes (2015) e Stallings (2017). No entanto, à medida que a demanda por poder computacional cresce exponencialmente, torna-se imprescindível uma análise crítica do impacto energético dos Centros de Processamento de Dados (CPDs) e das GPUs.

No artigo "A Brief History of Bioinformatics" Gauthier (2019), apresenta uma revisão histórica sobre o desenvolvimento da bioinformática, destacando os principais

avanços e desafios enfrentados ao longo de mais de 50 anos. A bioinformática começou na década de 1960, com a aplicação de métodos computacionais na análise de sequências de proteínas, antes mesmo de o DNA ser sequenciado. Margaret Dayhoff, considerada a pioneira da bioinformática, desenvolveu o primeiro software para montagem de sequências de proteínas e criou o primeiro banco de dados de sequências biológicas.

Na década de 1970, houve uma mudança de paradigma para a análise de DNA, impulsionada pela decodificação do código genético e pelo desenvolvimento de métodos de sequenciamento, como o método de Sanger. Paralelamente, avanços em biologia molecular e ciência da computação permitiram a análise de genomas completos, culminando na era da genômica nos anos 1990, com o sequenciamento do genoma humano.

Mas foi a partir dos anos 2000, o advento do sequenciamento de segunda geração (NGS) trouxe um aumento exponencial de dados biológicos, criando desafios relacionados ao gerenciamento de Big Data e à reprodutibilidade dos resultados. A bioinformática também se beneficiou do crescimento da internet, que facilitou o compartilhamento de dados e ferramentas, além de promover a colaboração científica global.

Nos últimos anos, a bioinformática tem enfrentado desafios como a integração de grandes volumes de dados, a necessidade de ferramentas mais acessíveis e a inclusão da disciplina nos currículos acadêmicos. Além disso, subdisciplinas como biologia sintética e modelagem de células inteiras estão emergindo, impulsionadas pela complementaridade entre biologia e ciência da computação. Nesse contexto, a questão central reside na magnitude do consumo energético associado a essas tecnologias e nas possíveis estratégias de otimização.

A quantificação precisa do consumo energético em Data Centers, particularmente em Unidades Centrais de Processamento (CPUs) e GPUs, é um aspecto-chave dentro da abordagem conhecida como "computação verde" (green computing). Esse conceito busca equilibrar alta performance computacional com práticas sustentáveis, minimizando o impacto ambiental e maximizando a eficiência energética.

A computação desempenha um papel fundamental ao impulsionar a sustentabilidade por meio de princípios como abertura, interoperabilidade, confidencialidade, integridade, autenticidade, disponibilidade e controle de qualidade dos dados. Esses princípios não apenas promovem inovação e transparência, mas também

fortalecem a confiança, garantindo a rastreabilidade e a eficiência no uso dos dados ambientais.

Pensando na sustentabilidade e no desempenho computacional dos sistemas de dados a SBC, propôs a formulação de uma Política Nacional com foco na sustentabilidade, o manifesto, “Tecnologias Digitais para o Meio Ambiente: Manifesto SBC”, SBC (2022), estabelece parâmetro para estas metas.

A aplicação de diversas tecnologias computacionais é de suma importância para beneficiar o meio ambiente. Entre elas, destacam-se os sistemas ciberfísicos (CPS), a Internet das Coisas (IoT), a computação de alto desempenho (HPC) e a inteligência artificial (IA), todas fundamentais para o desenvolvimento sistemas computacionais mais eficientes, possibilitando otimização do uso de recursos naturais. A computação em nuvem e a IA ampliam ainda mais as capacidades de análise e previsão dessas soluções tecnológicas.

Entre as ações propostas, está a formulação de uma Política Nacional voltada à sustentabilidade digital, que prevê o estabelecimento de indicadores regionais, o incentivo ao financiamento estratégico, a promoção da colaboração global e a criação de um Instituto Nacional de Ciência e Tecnologia focado em computação para o meio ambiente. Estes pontos são resumidos por Aguiar, AGUIAR (2025), em seu artigo “A Brazilian Perspective on Computing for the Planet’s Sustainability”.

Para aprofundar essa discussão, torna-se fundamental examinar os principais fatores que influenciam o consumo energético em simulações computacionais, conforme abordado por Berezovskaya (2020). Entre os aspectos determinantes, destacam-se: A eficiência algorítmica, que pode reduzir a complexidade computacional dos cálculos, minimizando a demanda por ciclos de processamento, a otimização do hardware, incluindo a escolha de arquiteturas energeticamente eficientes, como chips especializados para cálculos científicos, a adoção de fontes de energia renováveis, fundamental para mitigar o impacto ambiental gerado pelo funcionamento contínuo de CPDs.

Dessa forma, investigar e implementar estratégias voltadas à redução do consumo energético em bioinformática não apenas favorece a sustentabilidade, mas também fortalece a capacidade das pesquisas computacionais de se manterem viáveis a longo prazo. A intersecção entre inovação tecnológica e eficiência energética é, portanto, um campo de estudo indispensável para a evolução da ciência computacional e suas aplicações no estudo da vida.

Nesse contexto, a presente investigação estruturará sua busca e análise em torno dos seguintes eixos temáticos:

- **Metodologias de Medição:** Investigar as abordagens metodológicas utilizadas na quantificação do consumo de energia elétrica em simulações computacionais, abrangendo a análise de ferramentas de monitoramento em tempo real, técnicas de avaliação retrospectiva e métricas de eficiência energética, avaliando a eficácia de ferramentas de software como PowerTOP com o intuito de realizar comparações com as medições físicas.
- **Estratégias de Redução:** Explorar estratégias e técnicas literárias para a mitigação do consumo energético em ambientes de Computação de Alto Desempenho (HPC). A abordagem contempla o desenvolvimento de algoritmos eficientes, a otimização de sistemas de arrefecimento e a integração de fontes de energia renovável à infraestrutura..
- **Impacto Ambiental:** Avaliar as implicações ambientais do dispêndio energético em simulações computacionais intensivas. A análise centra-se na aplicação dos princípios de Green Computing como estratégia para fomentar a sustentabilidade, visando especificamente a mitigação das emissões de gases de efeito estufa (GEE) e a consequente redução da pegada de carbono.
- **Práticas Sustentáveis:** Identificar e analisar práticas de otimização energética aplicáveis a simulações biomoleculares e outras demandas computacionais intensivas, priorizando soluções que conciliem eficiência operacional e sustentabilidade ambiental.

2.4 - Desafios Energéticos em Infraestruturas Computacionais

A crescente demanda por energia elétrica, impulsionada pelo desenvolvimento tecnológico e o aumento da população global, torna a gestão eficiente do consumo uma prioridade. Para tanto, a quantificação precisa e em tempo real do uso de energia surge como uma ferramenta indispensável, empoderando consumidores e indústrias a tomarem decisões mais conscientes.

A evolução da tecnologia, especialmente a proliferação de microcontroladores de baixo custo e a expansão da Internet das Coisas (IoT), tem democratizado o acesso a

soluções inovadoras para essa medição, conforme evidenciado pelos trabalhos de J. Breda et al. (2019), R. W. Fransiska et al. (2013) e K. Luechaphonthara e V. A (2019).

A evolução da computação contemporânea consolidou a coexistência de duas arquiteturas de processamento distintas e complementares: a Unidade Central de Processamento (CPU) e a Unidade de Processamento Gráfico (GPU). Segundo Stallings (2018), a divergência fundamental entre ambas não reside apenas na contagem de núcleos, mas na filosofia de projeto voltada para diferentes classes de problemas computacionais.

A CPU é caracterizada por ser um processador de uso geral, otimizado para a redução da latência de execução de threads individuais. Sua arquitetura é robusta em termos de lógica de controle e memória cache. Grande parte da área de silício de uma CPU é dedicada a mecanismos sofisticados, como predição de desvios (branch prediction), execução fora de ordem e unidades de controle complexas, visando manter o fluxo de instruções sequenciais o mais contínuo possível. Assim, a CPU é ideal para tarefas que exigem tomadas de decisão rápidas e manipulação de fluxos de execução variados.

Em contrapartida, a GPU foi concebida sob o paradigma da computação massivamente paralela, focada no aumento do throughput (vazão) de dados. Ao contrário da CPU, a GPU dedica a maior parte de seus transistores ao processamento de dados propriamente dito (unidades lógicas e aritméticas - ALUs) em detrimento da lógica de controle e dos grandes níveis de cache. Stallings destaca que as GPUs operam predominantemente sob o modelo SIMD (Single Instruction, Multiple Data), onde uma única instrução é aplicada simultaneamente a um vasto conjunto de dados. Em arquiteturas modernas, como a Fermi citada pelo autor, essa organização se dá por meio de Multiprocessadores de Streaming (MS), que gerenciam milhares de threads ativas para esconder a latência de acesso à memória.

No que tange ao desempenho computacional, a distinção é quantificada pela capacidade de operações de ponto flutuante (FLOPs). Enquanto a CPU apresenta um crescimento de desempenho linear e limitado pela complexidade do paralelismo de nível de instrução (ILP), a GPU exhibe um crescimento exponencial em sua capacidade de cálculo. Isso ocorre porque o desempenho da GPU é escalonado pela adição de mais núcleos de processamento simples, permitindo que ela execute tarefas matemáticas intensivas com uma eficiência de ordem de magnitude superior à da CPU.

Contudo, Stallings ressalta que o alto desempenho da GPU é condicionado à natureza do algoritmo. Em um modelo de computação heterogênea (GPGPU), a CPU assume o papel de "host", encarregada da lógica de controle e das tarefas seriais, enquanto a GPU atua como um acelerador ou "device", processando os núcleos computacionais (kernels) que apresentam alto grau de paralelismo de dados. Portanto, a superioridade de desempenho de uma sobre a outra não é absoluta, mas dependente da carga de trabalho: a CPU domina na complexidade lógica e baixa latência, enquanto a GPU é a ferramenta de escolha para o processamento de grandes volumes de dados que podem ser decompostos paralelamente.

Tradicionalmente, a medição de consumo elétrico era realizada por medidores eletromecânicos, oferecendo apenas dados agregados e com atraso. A introdução de microcontroladores como o Arduino, entretanto, revolucionou essa realidade. Fransiska et al. (2013) demonstraram o potencial do Arduino Uno em conjunto com o software LabVIEW para a medição de potência elétrica, provando a viabilidade de sistemas de baixo custo para monitoramento. Esse trabalho seminal abriu caminho para o desenvolvimento de soluções mais acessíveis e flexíveis, superando as limitações dos sistemas convencionais e possibilitando a coleta de dados mais granulares.

Avançando nessa linha, Breda et al. (2019) apresentaram o desenvolvimento de um medidor de consumo de energia elétrica em tempo real, utilizando medidores de corrente não invasivos e a plataforma Arduino. A abordagem não invasiva é particularmente vantajosa, pois elimina a necessidade de intervenções complexas na instalação elétrica, tornando a implementação mais segura e prática para o usuário comum. A capacidade de monitoramento em tempo real fornecida por essas soluções é crucial, pois permite que os consumidores identifiquem picos de consumo, detectem aparelhos ineficientes e ajustem seus hábitos para otimizar o uso de energia.

A inovação na quantificação do consumo de energia, apresentado na integração desses sistemas com a Internet das Coisas (IoT) proposto por Luechaphonthara e V. A (2019)], onde exploraram a aplicação de sistemas baseados em IoT para o monitoramento do consumo de energia elétrica em eletrodomésticos. Essa interconexão permite que os dados de consumo sejam acessados remotamente, em qualquer lugar e a qualquer momento, por meio de dispositivos móveis ou plataformas web. Além de simples visualização, a IoT abre portas para a análise inteligente desses dados, possibilitando a identificação de padrões de consumo, a previsão de gastos e até mesmo o controle automatizado de dispositivos para reduzir o desperdício.

A quantificação do consumo de energia elétrica, impulsionada por essas inovações, transcende a mera coleta de números. Ela capacita indivíduos e organizações a assumirem um papel ativo na gestão de sua pegada energética, resultando não apenas em economia financeira, mas também em um impacto ambiental significativamente reduzido. A disseminação de tecnologias acessíveis e a facilidade de integração com a IoT apontam para um futuro em que a ineficiência energética será cada vez mais difícil de ser ignorada, e a gestão inteligente da energia se tornará uma norma, e não uma exceção.

Uma vez estabelecidos os pilares conceituais os quais vinculam a computação de alto desempenho aos objetivos de sustentabilidade da Agenda 2030, torna-se necessário investigar como a comunidade científica tem endereçado, na prática, o binômio desempenho-eficiência. A compreensão da bioinformática como uma disciplina catalisadora de avanços na saúde, conforme discutido na fundamentação, exige no momento atual um mergulho sistemático na “Computação Verde”. Reconhecer que além importância dos algoritmos de basecalling ou da infraestrutura de GPUs, é necessário mapear o consumo de energia nos CPD, com evidências técnicas e mapear melhoria e otimização dos algoritmos que sustentam e ordenam as simulações genômicas, mostram a necessidade de novos modelos de monitoramento. Dessa forma, a seção seguinte dedica-se a uma revisão criteriosa da literatura, buscando nas bases de dados globais os fundamentos práticos e as estratégias de mitigação que validam a construção de uma bioinformática tecnicamente robusta e economicamente sustentável.

3 REVISÃO DA LITERATURA

A metodologia para a análise foi a de uma busca na literatura, que investigou o panorama da eficiência energética em simulações computacionais, o qual é um aspecto de crescente e relevante no contexto da bioinformática e de outras disciplinas que fazem o uso intensivo de computação. A estratégia de busca empregada abrangeu bases de dados científicas de reconhecida importância, nomeadamente IEEE Xplore, Google Scholar e ScienceDirect. A seleção dessas plataformas justifica-se pela sua abrangência e indexação de um vasto espectro de publicações científicas e técnicas.

A formulação da estratégia de busca foi meticulosamente elaborada, empregando um conjunto de palavras-chave interconectadas e representativas no âmbito investigado:

'energy consumption in computational simulations', 'electricity usage in computer simulations', 'energy efficiency in computational modeling', 'power consumption measurement in simulations', 'reducing energy consumption in simulations' e, reconhecendo a interface com práticas computacionais sustentáveis, 'green computing and bioinformatics'.

Com o propósito de consolidar o estado da arte sobre o custo energético na bioinformática, a pesquisa explorou as bases IEEE Xplore, Google Acadêmico e ScienceDirect. A escolha dessas plataformas foi estratégica: o Google Acadêmico oferece abrangência, indexando uma vasta literatura (teses, livros e anais) de múltiplas editoras; já o IEEE Xplore e o ScienceDirect conferem especificidade técnica, cobrindo publicações cruciais em engenharia, computação de alto desempenho e sistemas de energia.

A estratégia de busca abrangeu múltiplas dimensões do problema: a mensuração direta ('power consumption measurement in simulations', 'electricity usage in computer simulations'), a análise geral de consumo ('energy consumption in computational simulations') e as metodologias de eficiência ('energy efficiency in computational modeling', 'reducing energy consumption in simulations'). Por fim, alinhando-se à sustentabilidade, integrou-se o descritor 'green computing and bioinformatics' para conectar as simulações às práticas de TI Verde.

A aplicação da estratégia de busca nas referidas plataformas resultou na identificação dos seguintes achados:

No Google Scholar

- Termo de busca: “energy consumption in computational simulations and/or green computing”;
- Termo de Busca: “electricity usage in computer simulations and/or green computing” ;
- Termo de busca: “energy efficiency in computational modeling and/or green computing”;
- Termo de Busca: “power consumption measurement in simulations and/or green computing”;
- Termo de busca: “reducing energy consumption in simulations and/or green computing”;
- Termo de busca “green computing and bioinformatics”;

- Termo de busca “carbon footprint”;
- Termo de busca “carbon footprint on bioinformatics”;
- Termo de busca “Consumption of power in computer simulations”;
- Termo de busca “economic quantification of electrical energy consumption in computer simulations”;
- Termo de busca “ consumption power in computer simulations on bioinformatics”;
- Termo de busca “economic quantification of electrical energy consumption in computer simulations on bioinformatics”.

No IEEEExplore

- Termo de busca: “energy consumption in computational simulations, green computing”;
- Termo de Busca: “electricity usage in computer simulations, green computing”;
- Termo de busca: “energy efficiency in computational modeling, green computing”;
- Termo de Busca: “power consumption measurement in simulations, green computing”;
- Termo de busca: “reducing energy consumption in simulations, green computing”;
- Termo de busca “green computing and bioinformatics”;
- Termo de busca “carbon footprint”;
- Termo de busca “carbon footprint on bioinformatics”;
- Termo de busca “Consumption of power in computer simulations”;
- Termo de busca “economic quantification of electrical energy consumption in computer simulations”;
- Termo de busca “ consumption power in computer simulations on bioinformatics”;
- Termo de busca “economic quantification of electrical energy consumption in computer simulations on bioinformatics”.

A seleção de artigos científicos para esta revisão da literatura, foi centrada no consumo de energia elétrica em simulações computacionais de bioinformática, e conduzida seguindo os critérios estabelecidos por R. S. Wazlawick em Metodologia de Pesquisa para Ciência da Computação, Wazlawick (2009).

Inicialmente, priorizou-se a relevância temática, selecionando estudos que explorassem diretamente a interseção entre consumo energético e aplicações

bioinformáticas, como análise de genomas, modelagem molecular e descoberta de fármacos. Em seguida, avaliou-se a qualidade e a credibilidade das fontes, dando preferência a periódicos e conferências de alto impacto, revisados por pares, para garantir a robustez e a confiabilidade dos dados e metodologias apresentados.

Além disso, a atualidade das publicações foi considerada essencial, focando em estudos recentes a fim de captar tendências e avanços tecnológicos na área. Por fim, a pertinência metodológica guiou a seleção de artigos que apresentassem abordagens claras e replicáveis para medição, otimização e análise do consumo energético, possibilitando uma compreensão aprofundada das soluções propostas e dos desafios ainda existentes no campo.

Este estudo realizou uma revisão da literatura para investigar a eficiência energética em simulações computacionais, com foco na área da bioinformática. A pesquisa foi conduzida nas bases de dados IEEE Xplore, Google Scholar e ScienceDirect, utilizando um conjunto detalhado de palavras-chave sobre consumo de energia, eficiência, medição, redução e "computação verde", o que resultou em uma vasta quantidade inicial de artigos. Subsequentemente, foi aplicado um rigoroso processo de seleção baseado em critérios de relevância temática, qualidade da fonte, atualidade da publicação e pertinência metodológica, a fim de refinar os achados e garantir a robustez da análise sobre os desafios e soluções no campo.

A análise dos resultados obtidos nas bases Google Scholar e IEEE Xplore revela uma dicotomia clara entre o volume de discussão teórica e a disponibilidade de soluções técnicas específicas, desenhando um "funil de especificidade" que justifica a relevância da presente pesquisa.

Observa-se uma disparidade quantitativa significativa entre as plataformas. O Google Scholar, por indexar literatura cinzenta e citar termos em qualquer parte do texto, apresenta números na ordem de milhões (para "carbon footprint"). Isso indica que a sustentabilidade é um tema onipresente no discurso acadêmico global. Em contrapartida, o IEEE Xplore, focado em engenharia e ciências exatas, apresenta números drasticamente menores para os mesmos termos (para "carbon footprint"). Isso sugere que, embora muito se fale sobre o impacto ambiental, há proporcionalmente menos trabalhos focados na engenharia da solução e na implementação técnica de medidas de mitigação em hardware e software.

Os dados apontam que as linhas de pesquisa voltadas para Estratégias de Mitigação estão bem estabelecidas. Termos como "reducing energy consumption" e

"energy efficiency" retornaram centenas de milhares de resultados no Scholar e mantiveram uma liderança relativa no IEEE (475 e 440 resultados, respectivamente). A comunidade científica já reconhece o problema e foca fortemente em como gastar menos (eficiência algorítmica, hardware verde), consolidando o campo do Green Computing como uma área madura.

A busca por "green computing and bioinformatics" no Google Scholar retorna expressivos resultados, indicando que biólogos e cientistas de dados discutem sustentabilidade.

Porém, no IEEE Xplore, esse número cai para apenas 59 artigos. Isso evidencia que, embora a bioinformática seja uma grande consumidora de recursos, há uma carência de estudos técnicos de engenharia (hardware/software) dedicados especificamente a otimizar este nicho. A interseção entre Engenharia Elétrica/Computação e Bioinformática sob a ótica da energia ainda é pouco explorada tecnicamente.

O achado mais contundente desta revisão reside na escassez de trabalhos sobre a mensuração prática e a valoração econômica do consumo. Enquanto teorias sobre "reduzir consumo" abundam, a medição prática ("consumption of power in computer simulations on bioinformatics") retornou apenas 17 resultados no Scholar e 11 no IEEE.

O cenário torna-se ainda mais deserto quando se insere a variável financeira. O termo "economic quantification of electrical energy consumption in computer simulations on bioinformatics" retornou apenas 2 resultados no Scholar e zero (0) resultados no IEEE.

A literatura atual apresenta uma vasta cobertura sobre a necessidade teórica de sustentabilidade (Green Computing) e estratégias gerais de eficiência. No entanto, existe uma lacuna severa no que tange à instrumentação prática e à análise de custo-benefício financeiro em bioinformática. A ausência de registros no IEEE sobre a quantificação econômica em bioinformática valida, a originalidade desta pesquisa, que propõe não apenas estimar, mas medir e precificar o custo energético de simulações, preenchendo um espaço vazio entre a ciência da computação teórica e a engenharia aplicada.

A análise da literatura confirma a existência de uma lacuna crítica na quantificação prática e econômica do gasto energético em bioinformática, e a carência de estudos de engenharia aplicada. Para preencher tal lacuna, a seção a seguir detalha a Metodologia Experimental, descrevendo o desenvolvimento de um sistema de monitoramento customizado baseado em hardware livre e sensores de alta precisão. Esta etapa é

fundamental para converter os conceitos teóricos de 'Computação Verde' em dados empíricos mensuráveis, permitindo que a investigação avance para a fase de coleta, validação em equipamentos de perfis distintos e, por fim, a aplicação de ferramentas de priorização estatística, como o Método de Pareto, na identificação dos gargalos energéticos do sequenciamento genômico.

4 METODOLOGIA EXPERIMENTAL

4.1 Desenvolvimento do Sistema de Monitoramento

A análise aponta para fatores críticos que governam o consumo de energia elétrica em simulações bioinformáticas, incluindo a eficiência intrínseca dos algoritmos implementados, a otimização da arquitetura de hardware subjacente e a crucial consideração da proveniência energética, com ênfase na adoção de fontes renováveis. A mensuração deste consumo abrange um espectro de abordagens, desde ferramentas de monitoramento em tempo real, capazes de fornecer granularidade temporal fina, até análises retrospectivas, que oferecem uma visão longitudinal do comportamento energético.

Outras estratégias para a redução do consumo energético em simulações bioinformáticas englobam além da implementação de algoritmos computacionalmente mais eficientes, a adoção de sistemas de refrigeração de alta performance e baixo consumo, sendo este também um paradigma energético sustentável. O impacto ambiental decorrente do consumo energético em simulações computacionais é inegavelmente significativo, e a internalização de práticas de 'computação verde' emerge como um imperativo para a promoção da sustentabilidade e a minimização da pegada de carbono inerente às atividades de pesquisa na área de bioinformática. A otimização do uso de recursos energéticos em simulações biomoleculares, através da adoção de práticas eficientes e sustentáveis, não é apenas uma questão de responsabilidade ambiental, mas também um alinhamento estratégico com os ODS da ONU, que preconizam a utilização responsável de recursos e a mitigação das mudanças climáticas.

Estudos progressos, como as investigações sobre 'IoT based application for monitoring electricity power consumption in home appliances', Luechaphonthara, (2019) e 'Low-Cost Energy-Efficient Smart Monitoring System Using Open-Source Microcontrollers', Jameel (2016), ilustram o potencial de sistemas embarcados e da

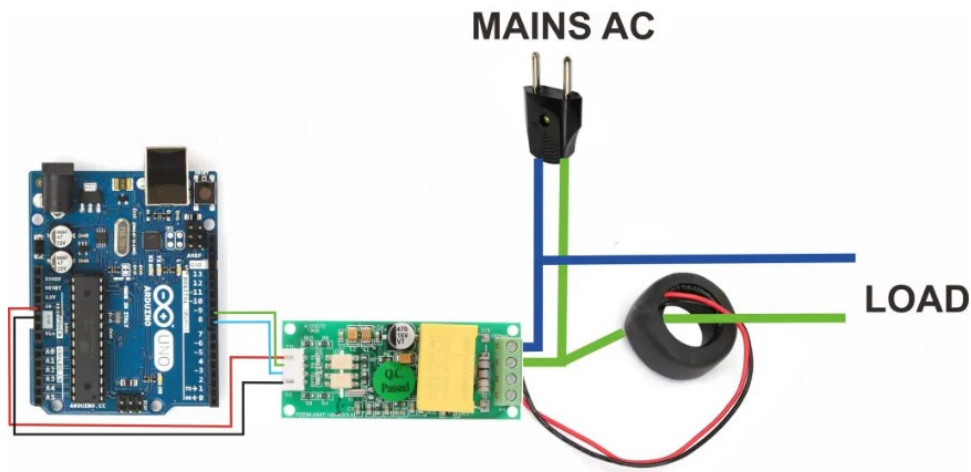
Internet das Coisas (IoT) para o monitoramento preciso do consumo energético em contextos diversos. Tais sistemas, fundamentados na utilização de microcontroladores e sensores inteligentes, possibilitam a coleta e transmissão de dados de consumo em tempo real, capacitando os usuários a refinarem seus padrões de utilização e a contribuir ativamente para a conservação de energia.

A consolidação e análise de dados provenientes de estudos anteriores relevantes para a bioinformática, relatórios de agências como a Agência Internacional de Energia (IEA) e dados empíricos coletados sobre o consumo energético em simulações computacionais de processos biológicos, revelam uma tendência inequívoca de aumento significativo no consumo de energia elétrica. Este incremento é diretamente impulsionado pela crescente demanda por poder de processamento computacional de alto desempenho, essencial para a resolução de problemas cada vez mais complexos na fronteira da pesquisa bioinformática.

A análise detalhada dos dados corrobora este aumento expressivo no consumo energético em simulações computacionais aplicadas à bioinformática, reafirmando a influência da crescente necessidade por poder computacional avançado. Adicionalmente, identificamos os fatores primários que modulam este consumo, reiterando a importância da eficiência algorítmica, da otimização do hardware e da escolha estratégica de fontes de energia renováveis. A implementação de sistemas de monitoramento inteligente emerge como uma ferramenta promissora para o ajuste dinâmico dos padrões de uso, oferecendo um caminho potencial para a significativa redução do consumo energético neste campo de atuação.

Desenvolveu-se um algoritmo para executar o monitoramento utilizando uma placa de ARDUINO UNO e um sensor PZEMT-004T, os detalhes do mesmo estão descritos no Anexo 1. A utilização da placa de Arduino Uno integrado ao sensor de monitoramento de energia PZEM-004T, segue os seguintes passos descritos e a programação em linguagem C++ estão no Anexo 2 deste trabalho. A arquitetura escolhida com base na revisão bibliográfica executada consiste em uma placa de ARDUINO UNO acoplada em um sensor do tipo PZEMT-004T.

Figura 2 – Diagrama Arduino Uno x Sensor PZEM-004T



Fonte: Miliohns.com

A utilização da placa de Arduino Uno integrado ao sensor de monitoramento de energia PZEM-004T, segue os seguintes passos:

1. Inclusão de Biblioteca:

- ✓ Conceito: A comunicação com o sensor PZEM-004T ocorre através da interface serial TTL (níveis lógicos de tensão). O Arduino Uno possui uma porta serial principal que geralmente é utilizada para a comunicação com o computador (via USB).
- ✓ Implementação: Para utilizar a comunicação serial com o sensor em pinos digitais do Arduino, emprega-se a biblioteca SoftwareSerial. Esta biblioteca permite emular uma porta serial em qualquer par de pinos digitais do Arduino.
- ✓ Código: `#include <SoftwareSerial.h>` - Esta linha inclui a biblioteca SoftwareSerial no projeto, tornando suas funcionalidades disponíveis.

2. Definição de Pinos Seriais:

- ✓ Conceito: É necessário definir quais pinos digitais do Arduino serão utilizados para a comunicação serial com o PZEM-004T. Os pinos de transmissão (TX) de um dispositivo devem ser conectados aos pinos de recepção (RX) do outro, e vice-versa.
- ✓ Implementação:

- `const int rxPin = 2;` - Define o pino digital 2 do Arduino como o pino de recepção (RX) dos dados enviados pelo PZEM-004T. Este pino será conectado ao pino TX do PZEM-004T.
 - `const int txPin = 3;` - Define o pino digital 3 do Arduino como o pino de transmissão (TX) dos dados para o PZEM-004T. Este pino será conectado ao pino RX do PZEM-004T.
- ✓ Importância: A correta conexão dos pinos RX e TX é crucial para o estabelecimento da comunicação serial. Uma conexão invertida impedirá a troca de dados entre o Arduino e o sensor.

3. Criação do Objeto SoftwareSerial:

- ✓ Conceito: Para utilizar as funcionalidades da biblioteca SoftwareSerial, é necessário criar um objeto dessa classe. Este objeto representará a porta serial virtual que será utilizada para se comunicar com o PZEM-004T.
- ✓ Implementação: `SoftwareSerial pzemSerial(rxPin, txPin);` - Esta linha cria um objeto chamado `pzemSerial`, associando-o aos pinos digitais 2 (RX) e 3 (TX) definidos anteriormente. Todas as operações de comunicação com o PZEM-004T serão realizadas através deste objeto.

4. Endereço do Dispositivo:

- ✓ Conceito: O PZEM-004T se comunica utilizando um protocolo que permite que múltiplos dispositivos sejam conectados a um mesmo barramento serial. Cada dispositivo possui um endereço único para identificação.
- ✓ Implementação: `byte pzemAddress = 0x01;` - Esta linha declara uma variável do tipo `byte` chamada `pzemAddress` e a inicializa com o valor `0x01`. Este é o endereço padrão de fábrica da maioria dos módulos PZEM-004T.
- ✓ Consideração para Múltiplos Módulos: Caso haja a necessidade de conectar vários sensores PZEM-004T a um único Arduino, o endereço de cada módulo precisará ser alterado (através de comandos específicos enviados ao módulo) para evitar conflitos na comunicação.

5. Variáveis para Dados:

- ✓ Conceito: Para armazenar os diferentes parâmetros de energia lidos do PZEM-004T, é necessário declarar variáveis com o tipo de dado apropriado para representar valores numéricos com casas decimais.
- ✓ Implementação: São declaradas variáveis do tipo float (ponto flutuante) para cada parâmetro:
 - `tensaoRedeLida` (Tensão em Volts)
 - `correnteLida` (Corrente em Amperes)
 - `potenciaLida` (Potência em Watts)
 - `energiaLida` (Energia consumida em kWh)
 - `frequenciaLida` (Frequência da rede em Hz)
 - `fatorPotenciaLido` (Fator de potência)

6. Função `setup()`:

- ✓ Conceito: A função `setup()` é executada apenas uma vez quando o Arduino é ligado ou resetado. Ela é utilizada para realizar configurações iniciais.
- ✓ Implementação:
 - `Serial.begin(9600);` - Inicializa a comunicação serial principal (via USB) com o computador a uma taxa de transmissão de 9600 bits por segundo (baud). Isso permite que o Arduino envie informações para o monitor serial do IDE.
 - `Serial.println("Iniciando leitura do PZEM-004T...");` - Exibe uma mensagem no monitor serial indicando o início do processo de leitura do sensor.
 - `pzemSerial.begin(9600);` - Inicializa a comunicação serial com o sensor PZEM-004T através da porta serial virtual (`pzemSerial`) na mesma taxa de baud de 9600. Esta é a taxa de comunicação padrão do PZEM-004T.

- `delay(100);` - Introduz uma pequena pausa de 100 milissegundos para garantir que os dispositivos tenham tempo suficiente para inicializar corretamente antes de iniciar a comunicação.

7. Função `loop()`:

- ✓ Conceito: A função `loop()` é executada continuamente após a função `setup()` ter sido concluída. Ela contém o código principal que será repetido indefinidamente.
- ✓ Implementação:
 - `if (requestPZEMData()) { ... } else { ... }` - Chama a função `requestPZEMData()` para solicitar os dados do sensor. A estrutura condicional `if-else` verifica o valor de retorno desta função:
 - Se `requestPZEMData()` retornar `true`, significa que os dados foram lidos com sucesso, e o bloco de código dentro do `if` é executado para exibir os valores lidos no monitor serial.
 - Se `requestPZEMData()` retornar `false`, indica que ocorreu um erro durante a leitura, e o bloco de código dentro do `else` é executado para exibir uma mensagem de erro.
 - `Serial.print(...)` e `Serial.println(...)` - Dentro do bloco `if`, estas funções são utilizadas para enviar os valores de tensão, corrente, potência, energia, frequência e fator de potência para o monitor serial do Arduino IDE, permitindo que o usuário visualize os dados lidos.
 - `delay(1000);` - Introduz uma pausa de 1000 milissegundos (1 segundo) antes da próxima iteração do `loop`, controlando a frequência com que os dados são lidos do sensor.

8. Função `requestPZEMData()`:

- ✓ Conceito: Esta função é responsável por enviar a solicitação de dados ao sensor PZEM-004T e receber a resposta contendo as informações de energia.
- ✓ Implementação:

- Frame de Requisição:

- ❖ `byte requestFrame[] = {pzemAddress, 0x04, 0x00, 0x00, 0x00, 0x0A, 0x21, 0xCB};` - Define um array de bytes chamado `requestFrame`. Este array contém o comando específico para solicitar os dados de leitura do PZEM-004T (função de leitura com código 0x04).
- ❖ Os bytes individuais têm o seguinte significado:
- ❖ `pzemAddress`: Endereço do dispositivo PZEM-004T (definido como 0x01).
- ❖ 0x04: Código da função de leitura (Read Input Registers).
- ❖ 0x00, 0x00: Endereço inicial do registro a ser lido (neste caso, o primeiro registro de dados).
- ❖ 0x00, 0x0A: Número de registros a serem lidos (10 registros contendo todos os parâmetros de energia).
- ❖ 0x21, 0xCB: CRC (Cyclic Redundancy Check) de 16 bits para garantir a integridade do frame de requisição. O CRC é calculado com base nos bytes anteriores do frame.

- Buffer de Resposta:

- ❖ `byte responseFrame[25];` - Declara um array de bytes chamado `responseFrame` com tamanho 25. Este array será utilizado para armazenar os bytes de resposta enviados pelo PZEM-004T. O tamanho da resposta para esta solicitação é de 25 bytes, conforme a documentação do sensor.

- Timeout:

- ❖ `unsigned int timeout = 1000;` - Define um tempo limite de 1000 milissegundos (1 segundo) para aguardar a resposta do sensor.
- ❖ `unsigned long startTime = millis();` - Armazena o tempo em que a solicitação foi enviada.
- ❖ `int bytesReceived = 0;` - Inicializa um contador para o número de bytes recebidos.

- ❖ O loop `while (millis() - startTime < timeout && bytesReceived < 25)` aguarda a chegada dos bytes de resposta até que o tempo limite seja atingido ou todos os 25 bytes esperados sejam recebidos.
- ❖ `if (pzemSerial.available()) { responseFrame[bytesReceived++] = pzemSerial.read(); }` - Dentro do loop, verifica se há dados disponíveis na porta serial virtual (pzemSerial). Se houver, um byte é lido e armazenado no responseFrame, e o contador bytesReceived é incrementado.
- Envio da Requisição:
 - ❖ `pzemSerial.write(requestFrame, sizeof(requestFrame));` - Envia o array requestFrame (contendo o comando de leitura) para o PZEM-004T através da porta serial virtual.
- Verificação da Resposta:
 - ❖ `if (bytesReceived == 25 && responseFrame[0] == pzemAddress)` - Após o período de espera, esta condição verifica se:
 - ❖ O número de bytes recebidos (bytesReceived) é igual ao tamanho esperado da resposta (25).
 - ❖ O primeiro byte da resposta (responseFrame[0]) corresponde ao endereço do dispositivo solicitado (pzemAddress), garantindo que a resposta veio do sensor correto.
- Extração dos Dados:

Se a resposta for válida, os bytes relevantes do responseFrame são extraídos e combinados para formar os valores dos parâmetros de energia. As fórmulas de conversão (multiplicação e divisão por diferentes potências de 10) são específicas do protocolo de comunicação do PZEM-004T e estão detalhadas em sua documentação técnica

Alinhada aos Objetivos de Desenvolvimento Sustentável (ODS) da ONU e fundamentada em uma Revisão da Literatura, esta pesquisa enfrenta o desafio crescente do consumo energético em simulações de bioinformática, exacerbado pela evolução de hardwares como CPUs e GPUs. No contexto da 'Computação Verde', o estudo investiga

metodologias de medição, estratégias de mitigação — incluindo algoritmos eficientes e energias renováveis — e o impacto da pegada de carbono. Em caráter prático, desenvolveu-se um sistema embarcado utilizando uma placa de Arduino Uno e um sensor PZEM-004T para o monitoramento em tempo real, quantificado o consumo de energia elétrica na simulação.

Por meio da construção de um modelo de avaliação do consumo de energia elétrica em CPDs e GPUs, no Laboratório de Bioinformática, Combi-Lab, do Centro de Ciências Computacionais, C3, da Universidade Federal do Rio Grande, FURG, considerando o processo de sequenciamento genético, com a tecnologia de nanoporos, com base em indicadores quantitativos do consumo de energia.

O modelo será desenvolvido por software de quantificação de consumo e desempenho, utilizando dados de pesquisas de simulações de sequenciamento genômico de proteínas no Combi-Lab. A partir do modelo poderá ser avaliado a importância da quantificação do consumo de energia em centro de processamento de dados, e seu impacto no consumo de energia elétrica.

A bioinformática, conforme delineado por Jawdat (2006), consolidou-se como um campo interdisciplinar essencial, unindo ciência da computação, estatística e biologia molecular para interpretar o crescente volume de dados biológicos. Essa disciplina transformou a pesquisa em diversas áreas, desde a descoberta de fármacos até a compreensão da evolução genômica. No entanto, a crescente sofisticação das análises bioinformáticas e sua ampla escala impõem desafios computacionais significativos, levantando questões cruciais sobre seu impacto ambiental.

Nesse contexto, o estudo de Grealey (2022) sobre a "pegada de carbono da bioinformática" apresenta uma análise fundamental. A pesquisa quantifica as emissões de gases de efeito estufa associadas às atividades bioinformáticas, demonstrando que o processamento e o armazenamento de extensos conjuntos de dados genômicos e proteômicos contribuem substancialmente para a pegada de carbono da pesquisa científica. Elementos como o consumo energético dos data centers, a eficiência dos algoritmos empregados e as práticas de computação em nuvem desempenham papéis determinantes nesse cenário.

A análise conduzida por Grealey (2022) ressalta a necessidade urgente de práticas bioinformáticas mais sustentáveis. Entre as soluções propostas, destacam-se o desenvolvimento de algoritmos energeticamente mais eficientes, a otimização do uso de

recursos em nuvem para minimizar impactos ambientais e a adoção de políticas de armazenamento de dados mais conscientes do ponto de vista energético.

Em síntese, embora a bioinformática, como previsto por Jawdat (2006), permaneça uma ferramenta indispensável para o avanço da biologia molecular e áreas afins, é imperativo reconhecer e mitigar seu impacto ambiental. O estudo de Grealey (2022) fornece uma base essencial para a implementação de estratégias que promovam uma bioinformática mais verde e sustentável, garantindo que os avanços científicos ocorram sem comprometer o meio ambiente.

Na bioinformática, onde a trajetória histórica é magistralmente narrada por Gauthier (2019), que redefiniu a pesquisa biológica colabora com ferramentas computacionais e analíticas indispensáveis para lidar com a explosão de dados gerados pelas tecnologias ômicas. Desde as primeiras tentativas de sequenciamento de proteínas e ácidos nucleicos até o surgimento de bancos de dados genômicos e proteômicos abrangentes, a bioinformática evoluiu de um campo incipiente para uma disciplina central na biologia moderna. Gauthier (2019) destacam como marcos fundamentais—como o desenvolvimento de algoritmos de alinhamento de sequências, ferramentas de análise filogenética e plataformas de anotação genômica—impulsionaram descobertas significativas na compreensão da vida em nível molecular. Essa trajetória ilustra o papel da bioinformática como um catalisador de avanços, permitindo a interpretação de dados complexos e a formulação de hipóteses testáveis em uma escala inédita.

Entretanto, a análise contemporânea de Grealey (2022) evidencia um efeito colateral preocupante dessa expansão: a crescente pegada de carbono da bioinformática, resultado do aumento da demanda por poder computacional e armazenamento de dados necessário para análises de larga escala. À medida que a bioinformática se torna cada vez mais integrada a diversas frentes da pesquisa biológica—desde a genômica de populações até a identificação de novos alvos terapêuticos—o volume de dados a ser processado e armazenado cresce exponencialmente. Gauthier (2019) já apontavam a escalada da complexidade dos dados como um dos desafios futuros da bioinformática. A pesquisa de Grealey (2022) quantifica o impacto ambiental dessa tendência e alerta para a necessidade de tornar a sustentabilidade um princípio essencial no desenvolvimento e na aplicação de metodologias bioinformáticas.

Para assegurar que o avanço científico impulsionado pela bioinformática seja sustentável a longo prazo, torna-se imprescindível adotar práticas computacionais mais eficientes, otimizar o consumo de energia em data centers e implementar políticas de armazenamento de dados com menor impacto ambiental. Isso pode envolver o desenvolvimento de algoritmos mais rápidos e menos intensivos em recursos, a transição para infraestruturas de computação em nuvem movidas por energia renovável e a formulação de estratégias de arquivamento de dados que reduzam o consumo energético a longo prazo.

Assim, a evolução da bioinformática deve incorporar uma crescente responsabilidade ecológica, equilibrando a necessidade de inovação científica com a urgência de minimizar seu impacto ambiental. A trajetória dessa disciplina, conforme delineada por Gauthier (2019), evidencia sua capacidade de adaptação e inovação; agora, esse mesmo potencial deve ser direcionado para a construção de um futuro da bioinformática que seja não apenas tecnologicamente avançado, mas também ambientalmente sustentável.

A intensidade energética do hardware, a localização dos centros de dados (e a fonte de energia que os alimenta) e a ineficiência de alguns algoritmos são fatores-chave que contribuem para essa pegada. Em muitos casos, os pesquisadores optam por soluções computacionais baseadas na nuvem, que, embora ofereçam flexibilidade e escalabilidade, podem operar em centros de dados com matrizes energéticas predominantemente baseadas em combustíveis fósseis. Além disso, a otimização algorítmica muitas vezes não é priorizada em detrimento da velocidade de desenvolvimento ou da facilidade de uso, resultando em códigos que consomem mais recursos do que o necessário.

Para conciliar o ímpeto da inovação bioinformática com a responsabilidade ambiental é fundamental a conscientização. A comunidade bioinformática deve ser elucidada sobre o impacto de suas escolhas computacionais. Iniciativas como "Green Algorithms" fornecem ferramentas que visam estimar a pegada de carbono de diferentes processos, capacitando os pesquisadores a tomada de decisões mais fundamentadas. Também urge que a otimização de códigos e algoritmos. Desenvolvedores devem ser incentivados a criar softwares mais eficientes em termos energéticos, explorando paralelização, uso inteligente de memória e estruturas de dados otimizadas.

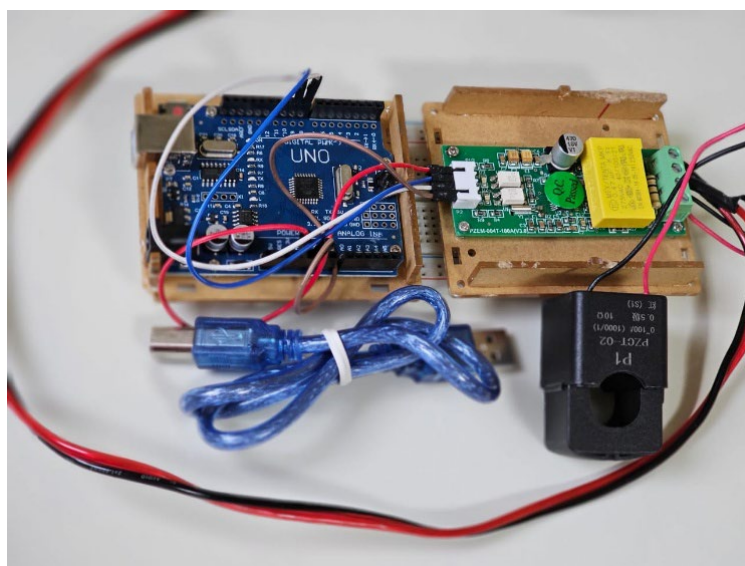
A escolha da infraestrutura computacional é um fator determinante. Sempre que possível, priorizando-se o uso de centros de dados que operem com fontes de energia

renovável ou que implementem práticas de resfriamento e gestão de energia eficientes. A transição para uma bioinformática mais verde não é um obstáculo, mas sim uma oportunidade para inovar e construir um futuro mais sustentável para a ciência.

O monitoramento do consumo energético dos processos computacionais mostra-se uma premissa no que tange a verificar ‘full time’ o gasto energético em simulações de bioinformática. Para esta quantificação os dados podem ser obtidos através de sensor acoplados a um sistema embarcado, que envia estes dados para armazenamento e quantificação e/ou valendo-se software de código aberto que monitoram o consumo energético do esforço computacional o qual a CPU e GPU são submetidos.

Desenvolveu-se um sistema de quantificação de energia valendo-se da utilização de um sensor PZEM-004t acoplado a uma placa de Arduino Uno R3, tomando por base o que discorreram em seus trabalhos Luechaphonthara e V. A (2019) e Jameel (2016). A arquitetura escolhida com base na revisão bibliográfica executada consiste em uma placa de ARDUINO UNO acoplada em um sensor do tipo PZEMT-004T.

Figura 3 – Protótipo Arduino Uno x Sensor PZEM-004T



Fonte: O autor

4.2 Arquitetura do Protótipo com Arduino e Sensor PZEM-004T

No atual cenário de crescente demanda por eficiência energética e gestão consciente de recursos, o desenvolvimento de sistemas de monitoramento de consumo de energia elétrica de baixo custo, precisão e tempo real assume papel de protagonismo.

Nesse contexto, a utilização de plataformas de prototipagem eletrônica, como o Arduino, acopladas a sensores especializados, emerge como uma solução tecnológica de grande relevância acadêmica, prática e funcional. Com base nos trabalhos de J. Breda, (2019), e em uma análise aprofundada das tecnologias envolvidas, o desenvolvimento de um sistema embarcado para o monitoramento do consumo de energia elétrica, utilizando-se uma placa de Arduino UNO em conjunto com o sensor PZEM-004T, mostra-se como uma alternativa viável e de baixo custo.

A escolha da plataforma Arduino UNO como unidade central de processamento em projetos de monitoramento de energia não é fortuita. Sua arquitetura de hardware aberta, ambiente de desenvolvimento integrado (IDE) intuitivo e a vasta gama de bibliotecas disponíveis simplificam e aceleram o ciclo de desenvolvimento. Conforme apontado em diversos estudos, a flexibilidade do Arduino permite a fácil integração com uma variedade de sensores e módulos de comunicação, tornando-o um componente ideal para a prototipagem e implementação de sistemas embarcados customizados e de baixo custo.

O sensor PZEM-004T, por sua vez, apresenta-se como um componente robusto e altamente especializado para a medição de grandezas elétricas em corrente alternada (AC). Sua principal vantagem reside na capacidade de mensurar, com notável precisão, múltiplos parâmetros essenciais para uma análise completa do consumo energético. Entre as grandezas aferidas pelo PZEM-004T, destacam-se:

- Tensão (V): Fornece o valor da diferença de potencial na rede elétrica.
- Corrente (A): Mede a intensidade da corrente elétrica que atravessa o condutor, sendo este um sensor do tipo não invasivo (utiliza um transformador de corrente), o que garante segurança e facilidade na instalação.
- Potência Ativa (W): Indica a potência real consumida pela carga, fundamental para o cálculo do consumo de energia.
- Energia Ativa Acumulada (kWh): Totaliza o consumo de energia ao longo do tempo, valor base para a tarifação pelas concessionárias de energia.

A combinação do sistema composto pelo Arduino UNO e o sensor PZEM-004T resulta em um sistema de monitoramento com um balanço ideal entre custo, precisão e funcionalidade. Enquanto o PZEM-004T se encarrega da complexa tarefa de medição das grandezas elétricas com alta fidelidade e de forma segura, o Arduino UNO atua como o cérebro do sistema, processando os dados recebidos do sensor, realizando cálculos

e, potencialmente, exibindo as informações em um display, armazenando-as em um cartão de memória ou transmitindo-as para uma plataforma online para visualização e análise remota.

O desenvolvimento de um sistema embarcado com as características mencionadas, fundamentado nos princípios explorados por J. Breda, (2019) em seu trabalho sobre medidores não invasivos, oferece um leque de vantagens significativas:

- I. Baixo Custo e Acessibilidade: A utilização de componentes de hardware de baixo custo, como o Arduino UNO e o PZEM-004T, viabiliza a implementação de sistemas de monitoramento a uma fração do preço de equipamentos comerciais equivalentes, democratizando o acesso a essa tecnologia tanto para fins acadêmicos e de pesquisa, quanto para aplicações residenciais e em pequenos comércios.
- II. Monitoramento em Tempo Real e Tomada de Decisão: A capacidade de coletar e processar dados em tempo real permite aos usuários uma compreensão instantânea de seus perfis de consumo. Essa informação é vital para a identificação de equipamentos ineficientes, a detecção de picos de consumo e a implementação de estratégias de economia de energia de forma ágil e informada.
- III. Ser não Invasivo e Segurança: O uso de um transformador de corrente não invasivo, como o que acompanha o sensor PZEM-004T, elimina a necessidade de intervenções diretas e potencialmente perigosas na fiação elétrica para a instalação do medidor. Isso não apenas simplifica o processo, mas também eleva a segurança do sistema.
- IV. Flexibilidade e Escalabilidade: A natureza modular do sistema permite a sua fácil adaptação e expansão. É possível, por exemplo, adicionar módulos de comunicação sem fio (Wi-Fi, Bluetooth) ao Arduino UNO para enviar os dados para a nuvem, criar painéis de visualização online ou integrar o sistema a plataformas de automação residencial, ampliando exponencialmente suas funcionalidades.
- V. Ferramenta Pedagógica e de Pesquisa: A concepção e implementação de um sistema de monitoramento de energia com Arduino e PZEM-004T constituem uma excelente ferramenta de aprendizado prático em áreas como eletrônica, programação de sistemas embarcados, e eficiência energética. Para pesquisadores, a plataforma oferece um meio acessível e customizável para a coleta de dados detalhados sobre o consumo de energia em diferentes cenários.

Por esta razão de união da versatilidade e do baixo custo do Arduino UNO, com a precisão e a segurança do sensor PZEM-004T configura uma solução tecnológica de elevado potencial para o desenvolvimento de sistemas embarcados de monitoramento de consumo de energia elétrica. Inspirado e validado por J. Breda, (2019), este tipo de sistema não apenas se alinha com as crescentes demandas por sustentabilidade e uso racional da energia, mas também fomenta a inovação e o conhecimento aplicado, representando um avanço significativo na área de medição e gerenciamento energético. A sua implementação capacita os usuários com informações valiosas, promovendo uma cultura de consumo consciente e abrindo portas para futuras pesquisas e desenvolvimentos no campo da automação, eficiência energética e da Computação Verde. No Anexo 1, está a estrutura do Algoritmo pensado para desenvolvimento do software para controle do Protótipo e no Anexo 2, a programação de controle em C++ do Arduino UNO.

Para a calibração e validação das medições de corrente e tensão do sistema embarcado (composto pelo microcontrolador Arduino Uno e o módulo sensor PZEM-004T), utilizou-se um alicate amperímetro digital de referência, modelo ET-3200A (Minipa). O equipamento é classificado com categoria de segurança CAT II 1000V, operando com um conversor A/D de rampa dupla e display LCD de 3½ dígitos (contagem máxima de 2000). As especificações técnicas relevantes para os ensaios realizados estão detalhadas abaixo:

- Capacidade de Medição:
 - Corrente AC: Escalas de 20A a 1000A (frequência de 50Hz-60Hz).
 - Tensão: DC (200mV a 1000V) e AC (200V a 750V).
 - Resistência: 200Ω a 2MΩ.
- Parâmetros de Precisão: Apresenta precisão básica de $\pm 2,0\%$, garantindo a confiabilidade necessária para a comparação dos dados lidos pelo sensor PZEM-004T.
- Aspectos Físicos e Funcionais: O dispositivo possui abertura de garra de 57mm, permitindo o acoplamento em diversos diâmetros de condutores, e função *Peak Hold*, utilizada para o registro de valores de pico durante as medições.

Nota Acadêmica: A escolha deste equipamento justifica-se pela sua ampla aceitação em ambientes laboratoriais e industriais, fornecendo a base comparativa para a verificação do erro percentual do sistema desenvolvido na dissertação, as escalas utilizadas foram as de 200V AC e 20A AC, que são as mais próximas da realidade do PZEM-004T, garantindo menor incerteza na leitura.

4.3 Procedimento de Medição e Coleta de Dados

A metodologia para avaliação do consumo energético foi estabelecida em duas etapas. Primeiramente, realizou-se a validação do sistema de medição (Arduino Uno/PZEM-004T) por meio da aferição de equipamentos com cargas elétricas distintas e conhecidas: um laptop (mista), uma chaleira elétrica (resistiva) e um refrigerador (indutiva). Os resultados, alinhados ao perfil teórico esperado e detalhados no Capítulo 5, confirmaram a precisão do sistema. Subsequentemente, o aparato validado foi empregado no objetivo central do estudo: quantificar e comparar o consumo energético de sequenciadores da Oxford Nanopore Technology durante o processamento de um mesmo experimento, utilizando alternadamente a CPU e a GPU para traçar o perfil de consumo de cada unidade. Para garantir a quantificação do consumo de energia apenas da simulação genômica, quantificou-se o consumo energético do ambiente computacional quando está apenas operando seu sistema operacional.

4.4 Medições em Equipamentos Domésticos

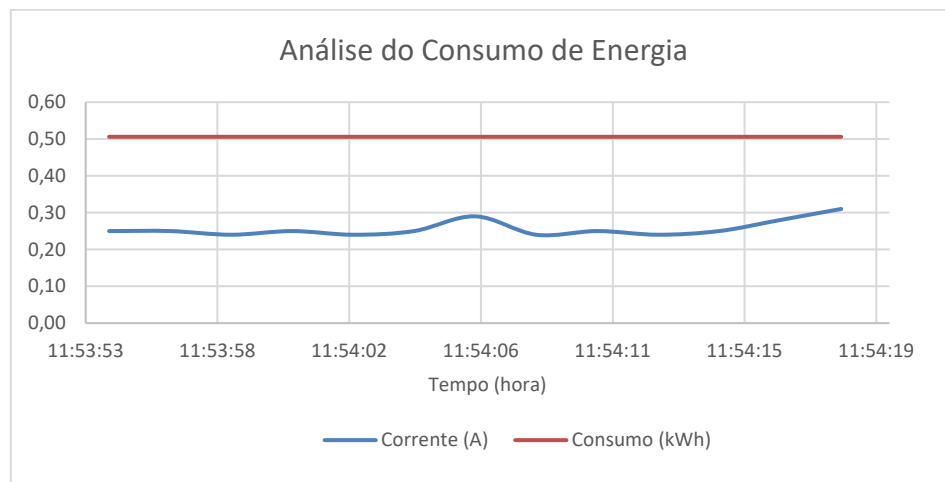
Após a construção do protótipo para a validação da programação e verificação de seu funcionamento foram executadas medições consumo de energia elétrica em três equipamentos domésticos, onde seus consumos já foram monitorados pelo INMETRO, são eles: um Laptop Samsung Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.11 GHz, uma Chaleira Elétrica - Cadence Infusion - 127V 1200W e um Refrigerador Frostfree Eletrolux - DF52X - 127V 1200W. Os dados de consumo foram obtidos no dia 09 de maio de 2025, com uma voltagem de 127V e uma frequência de rede de 60Hz, e são expostos abaixo pelas tabelas e gráficos.

Tabela 1 - Laptop Samsung Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.11 GHz

Consumo de Energia de um Laptop - Samsung Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.11 GHz						
Hora	Voltagem (V)	Corrente (A)	Potência (W)	Consumo (kWh)	Frequência (Hz)	FP
11:53:54	127,80	0,25	17,70	0,506	59,90	0,56
11:53:56	127,80	0,25	17,80	0,506	60,00	0,56
11:53:58	127,70	0,24	17,50	0,506	60,00	0,56
11:54:00	127,60	0,25	17,70	0,506	59,90	0,56
11:54:02	127,60	0,24	17,50	0,506	60,00	0,57
11:54:04	127,60	0,25	17,60	0,506	59,90	0,56
11:54:06	127,60	0,29	20,80	0,506	60,00	0,56
11:54:08	127,60	0,24	17,50	0,506	60,00	0,57
11:54:10	127,70	0,25	18,10	0,506	60,00	0,56
11:54:12	127,60	0,24	17,40	0,506	59,90	0,57
11:54:14	127,70	0,25	17,70	0,506	59,90	0,56
11:54:16	127,70	0,28	20,00	0,506	60,00	0,56
11:54:18	127,70	0,31	21,80	0,506	60,00	0,56

Fonte: O autor

Gráfico 1 - Laptop Samsung Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.11 GHz



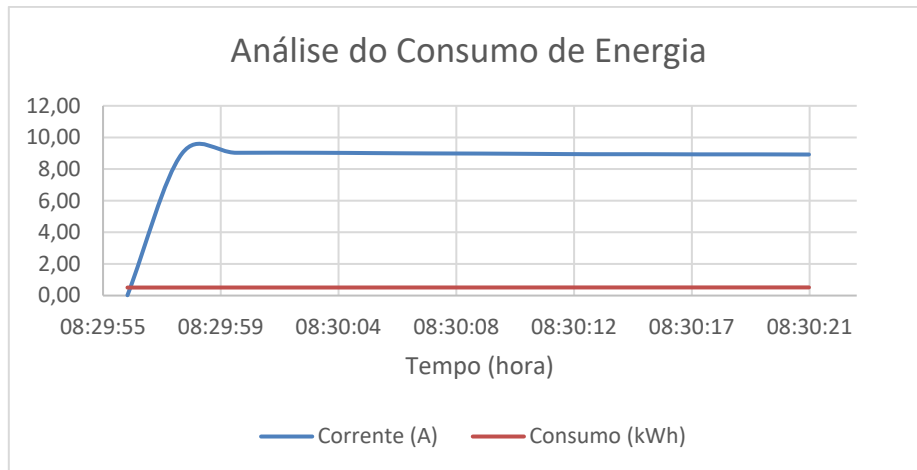
Fonte: O autor

Tabela 2 - Chaleira Elétrica - Cadence Infusion - 127V 1200W

Consumo de Energia de uma Chaleira Elétrica - Cadence Infusion - 127V 1200W						
Hora	Voltagem (V)	Corrente (A)	Potência (W)	Consumo (kWh)	Frequência (Hz)	FP
08:29:56	126,00	0,02	0,00	0,507	60,00	0,00
08:29:58	123,20	8,98	1073,12	0,508	60,00	0,97
08:30:00	123,30	9,03	1112,70	0,509	60,00	1,00
08:30:03	123,40	9,03	1113,60	0,509	60,00	1,00
08:30:05	123,50	9,01	1112,30	0,510	60,00	1,00
08:30:07	123,40	8,99	1109,20	0,511	60,00	1,00
08:30:09	123,40	8,98	1106,90	0,511	60,00	1,00
08:30:11	123,20	8,96	1103,50	0,512	60,00	1,00
08:30:13	123,20	8,94	1101,60	0,513	60,00	1,00
08:30:15	123,10	8,94	1100,10	0,513	60,00	1,00
08:30:17	123,20	8,93	1100,10	0,514	60,00	1,00
08:30:19	123,10	8,93	1099,30	0,514	60,00	1,00
08:30:21	123,10	8,92	1098,30	0,515	60,00	1,00

Fonte: O autor

Gráfico 2 - Chaleira Elétrica - Cadence Infusion - 127V 1200W



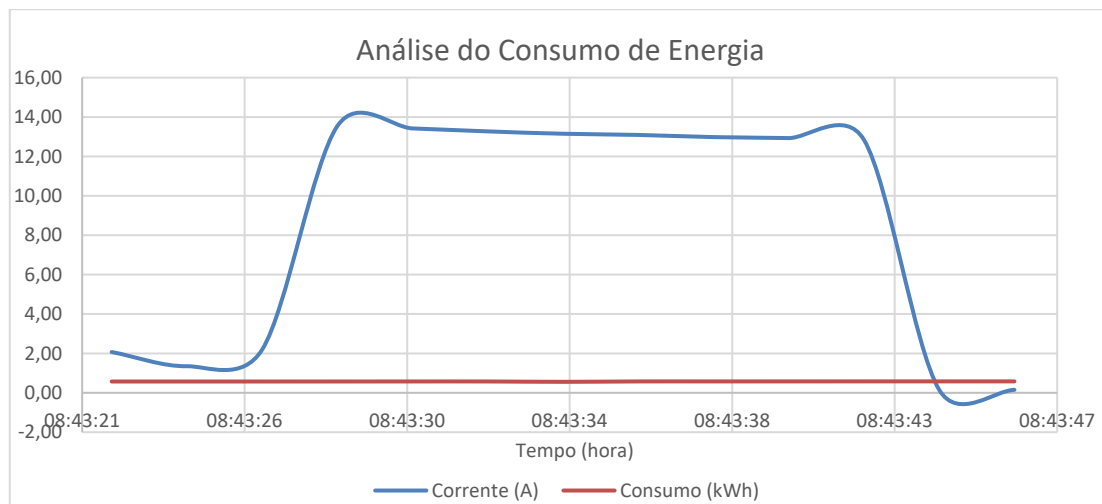
Fonte: O autor

Tabela 3 - Refrigerador Frostfree Eletrolux - DF52X - 127V 1200W

Consumo de Energia de um Refrigerador Frostfree Eletrolux - DF52X - 127V 1200W						
Hora	Voltagem (V)	Corrente (A)	Potência (W)	Consumo (kWh)	Frequência (Hz)	FP
08:43:22	124,30	2,07	163,30	0,574	54,70	0,64
08:43:24	124,20	1,35	158,30	0,574	59,90	0,94
08:43:26	124,00	2,19	271,00	0,574	60,00	1,00
08:43:28	121,60	13,54	1266,30	0,575	59,90	0,77
08:43:30	121,80	13,42	1255,00	0,576	59,90	0,77
08:43:32	121,70	13,27	1245,30	0,576	60,00	0,77
08:43:34	121,70	13,15	1239,30	0,557	59,90	0,77
08:43:36	121,80	13,09	1235,70	0,578	59,90	0,78
08:43:38	121,70	12,98	1229,30	0,578	60,00	0,78
08:43:40	121,70	12,93	1226,80	0,579	60,00	0,78
08:43:42	121,80	12,84	1223,20	0,580	59,90	0,78
08:43:44	124,40	0,15	12,60	0,580	60,00	0,68
08:43:46	124,40	0,15	12,60	0,580	60,00	0,68

Fonte: O autor

Gráfico 3 - Refrigerador Frostfree Eletrolux - DF52X - 127V 1200W



Fonte: O autor

4.5 Análise de dados – Método de Pareto

O Diagrama de Pareto é uma das sete ferramentas clássicas da qualidade, amplamente utilizada na gestão e na melhoria de processos para separar os problemas vitais dos triviais. Ele se baseia no "Princípio de Pareto", Araújo (2012), também conhecido como a Regra 80/20, que sugere que aproximadamente 80% dos efeitos advêm de apenas 20% das causas. Abaixo, detalhamos o funcionamento, a estrutura e a aplicação dessa ferramenta essencial para a tomada de decisão assertiva.

O diagrama, nomeado pelo consultor de qualidade Joseph Juran em homenagem ao economista italiano Vilfredo Pareto, Pareto observou que 80% da riqueza da Itália pertencia a apenas 20% da população do século XIX (c. 1896-1897). No contexto da gestão, essa lógica é aplicada para identificar que a maioria dos erros, perdas ou atrasos em uma empresa costuma estar concentrada em poucas categorias de causas, por exemplo 80% dos problemas gargalho logístico de um processo fabril tem como causa 20% de um tipo de processo.

Visualmente, o Diagrama de Pareto é um gráfico de colunas que organiza as frequências das ocorrências da maior para a menor, permitindo a visualização da prioridade. Ele é composto por três elementos principais:

- Eixo Horizontal (X): Apresenta as categorias ou causas que estão sendo analisadas (ex: tipos de defeitos, nomes de produtos, causas de atraso).
- Eixo Vertical Esquerdo (Y1): Indica a frequência absoluta (número de ocorrências) ou o custo de cada categoria.
- Eixo Vertical Direito (Y2) e Curva de Pareto: Representa a porcentagem acumulada das ocorrências, ajudando a identificar o ponto onde se atinge o patamar de 80%.

Para construir uma análise de Pareto eficaz, segue-se um roteiro estruturado:

1. Determinar qual problema será analisado (ex: "Por que os clientes estão reclamando?").
2. Reunir informações quantitativas sobre as ocorrências em um determinado período.
3. Agrupar os dados por tipo e ordená-los em ordem decrescente de frequência.
4. Somar as frequências progressivamente para entender o impacto combinado das

causas.

5. Desenhar as barras e a linha de porcentagem acumulada.

Assim como a análise de Pareto é uma ferramenta de priorização, ao invés de uma equipe dispersar energia tentando resolver todos os problemas de uma vez (o que é ineficiente e caro), ela foca nos "poucos vitais" como:

- Identificar quais itens (Curva A) representam o maior valor financeiro.
- Focar nos defeitos que mais geram refugo ou devoluções.
- Identificar quais componentes de uma máquina são responsáveis pela maior parte do tempo de parada (down time).

A utilização do Diagrama de Pareto não resolve o problema por si só, mas funciona como uma bússola para a identificação e gestão de um processo. Ao evidenciar onde o esforço de melhoria terá o maior retorno e eficácia, ele otimiza recursos e acelera a obtenção de resultados significativos dentro de qualquer organização.

Já no contexto da bioinformática, onde o processamento de grandes volumes de dados (como sequenciamento genômico ou modelagem molecular) exige alto desempenho computacional, o Diagrama de Pareto é uma ferramenta estratégica para a sustentabilidade e eficiência energética.

A aplicação consiste em identificar quais elementos da infraestrutura ou do fluxo de trabalho são os "vilões" do consumo, permitindo uma otimização direcionada. O primeiro passo é categorizar o que consome energia no ambiente de bioinformática. Podemos dividir as causas em três grandes grupos:

- Hardware (Infraestrutura): CPU, GPU, Memória RAM, Sistemas de Arrefecimento (Cooling) e Unidades de Armazenamento (Storage).
- Algoritmos/Softwares: Ferramentas específicas (ex: BLAST, BWA, GATK) ou etapas do pipeline (alinhamento, indexação, variant calling).
- Gestão de Dados: Transferência de dados via rede, compressão e descompressão de arquivos (FASTQ, BAM).

Para a coleta dos dados para alimentar o diagrama, é necessário medir o consumo real. Utilizamos o medidor de consumo confeccionado para a mesma, onde a telemetria do hardware fornece dados de consumo de energia por tarefa. Ao rodar um pipeline de

bioinformática podemos então, medir o consumo de energia gasto por cada ferramenta individualmente. Ao plotar o Diagrama de Pareto do consumo energético, é comum observar que:

- 20% das ferramentas de software (geralmente aquelas que realizam alinhamento de sequências ou montagem de novo) são responsáveis por 80% do consumo total de energia do pipeline.
- Ou, em termos de hardware, que o estado de espera (idle), estado operacional em que um dispositivo (como CPU, computador ou servidor) está ligado e funcional, mas não está processando ativamente tarefas ou dados de usuário, consomem a maior parte da energia, e não o processamento em si.

Uma vez identificados os "poucos vitais" através do gráfico, as ações de engenharia tornam-se mais eficientes:

- Causa Identificada (20%) - Ação de Otimização Recomendada;
- Algoritmo Específico - Substituição por uma versão otimizada para GPU ou paralelização mais eficiente.

A conclusão da fase experimental e a subsequente validação do protótipo em equipamentos de carga mista, resistiva e indutiva as quais confirmam a confiabilidade do sistema Arduino/PZEM-004T como instrumento de telemetria energética. Através da precisão do hardware assegurada, torna-se possível aplicar esta metodologia para a medição do consumo de energia no processamento de dados genômicos reais, estabelecendo uma razão de consumo de energia (kW.h) por tamanho do arquivo processado((Mb)..

A utilização deste sistema embarcado (Arduino UNO – Sensor PZEM-004T) permite a aquisição de resultados confiáveis permitindo comparar o desempenho de diferentes arquiteturas computacionais (CPU e GPU) e/ou simulações genômicas, sob a ótica da 'Computação Verde', quantificando o impacto de cada etapa do pipeline — do basecalling à montagem — no consumo elétrico total. A seção a seguir detalha essas medições, correlacionando o esforço computacional exigido por diferentes organismos à sua pegada de carbono correspondente, estabelecendo o nexos entre a eficiência algorítmica e a sustentabilidade na bioinformática moderna.

5 RESULTADOS E ANÁLISE COMPARATIVA

5.1 Experimentos com Sequenciadores Oxford Nanopore

Fazendo uma ponte os estudos de Luechaphonthara e V. A (2019), para a utilização de sistemas baseados em IoT para o monitoramento do consumo de energia elétrica com o exposto por Grealey (2022) com a evidência preocupante da crescente pegada de carbono da bioinformática, implementou-se a verificação do consumo de energia gasto com a montagem de genomas de eucariotos a partir de dados gerados por sequenciadores da Oxford Nanopore Technology, no trabalho acadêmico de Lucas Silveira Veiga, do PPGComp do C3, da FURG. A simulação foi executada em um desktop com as seguintes características em seu ambiente computacional de configurações de hardware e software:

- Sistema Operacional: Linux versão Ubuntu 22.04 LTS;
- Processador (CPU): Intel Core i7-13700K, com 16 núcleos físicos e 24 threads. O clock máximo de operação é de 5.4 GHz;
- Memória RAM: 62 GB de memória RAM;
- Placa de Vídeo (GPU): NVIDIA GeForce GTX 1050 Ti. A versão do driver NVIDIA utilizada é a 535.247.01, com suporte à versão CUDA 12.2.

O processamento paralelo das simulações foi conduzido em uma unidade de processamento gráfico (GPU) NVIDIA GeForce GTX 1050 Ti. Baseada na arquitetura Pascal (chip GP107), a placa dispõe de 768 núcleos CUDA e 4 GB de memória GDDR5, operando com uma interface de 128-bits e largura de banda de 112 GB/s. O dispositivo apresenta um desempenho de ponto flutuante de precisão simples (FP32) estimado em 2,1 TFLOPS.

Para o gerenciamento de recursos e interface com as bibliotecas de bioinformática, utilizou-se o driver NVIDIA versão 535.247.01 em ambiente Linux. Esta versão do driver provê suporte ao CUDA Toolkit 12.2, garantindo compatibilidade com algoritmos modernos de aceleração, apesar da arquitetura de hardware legada. No contexto da eficiência energética, a GTX 1050 Ti destaca-se pelo seu baixo consumo projetado (TDP) de 75W, dispensando conectores de alimentação externa na maioria das implementações, o que a torna um objeto de estudo relevante para cenários de Green Computing de baixo custo, a GPU utilizada é um equipamento que por sua simplicidade pode limitar uma análise comparativa moderna de alto desempenho.

O sequenciamento genético, um dos pilares de estudo da área da bioinformática esplanada nas obras de Gauthier (2019), Stallings (2017), Feltres (2015) e Jawdat (2006), foram executadas duas simulações de duas proteínas de um micro-organismo eucarionte. A primeira simulação utilizou-se apenas a capacidade computacional da CPU e foram obtidos os dados da tabela abaixo:

Tabela 4 - Consumo de Energia em Sequenciadores da Oxford Nanopore Technology (CPU)

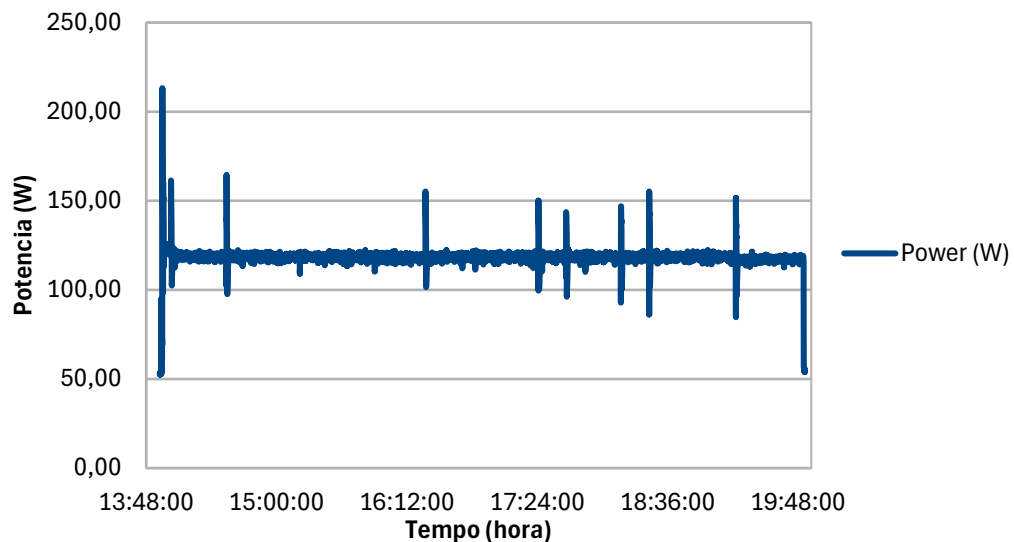
Time	Voltage (V)	Current (A)	Power (W)	Energy (kWh)
05:56:09	120,72*	1,07*	118,27*	0,935*

Fonte: O autor

*valores médios

Gráfico 4 - Consumo de Energia em Sequenciadores da Oxford Nanopore Technology (CPU)

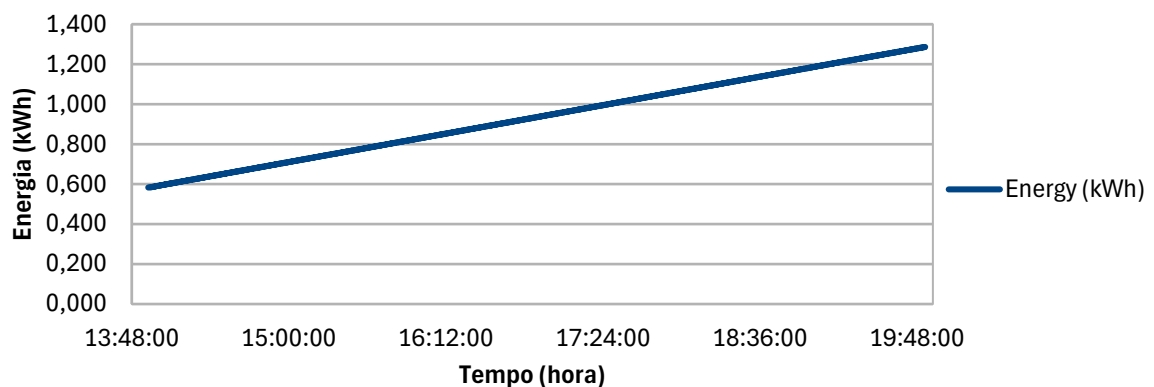
Análise do Consumo de Energia - CPU Sequenciadores da Oxford Nanopore Technology
- Power (W)



Fonte: O autor

Gráfico 5 - Consumo de Energia em Sequenciadores da Oxford Nanopore Technology (CPU)

Análise do Consumo de Energia - CPU Sequenciadores da Oxford Nanopore Technology - Energy (kWh)



Fonte: O autor

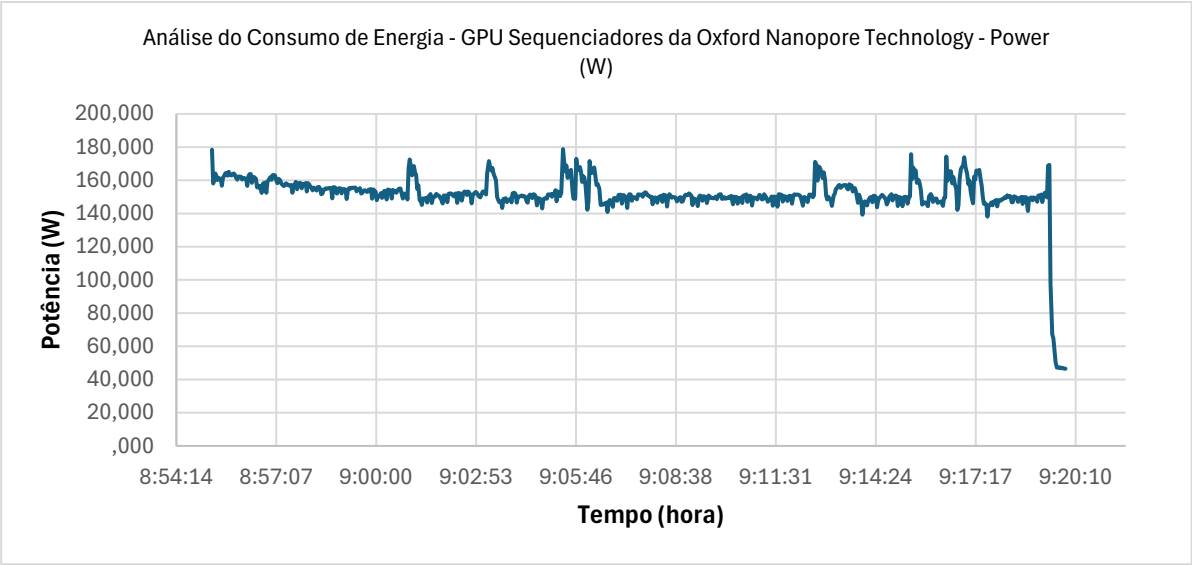
Tabela 5 - Consumo de Energia em Sequenciadores da Oxford Nanopore Technology (GPU)

Time	Voltage (V)	Current (A)	Power (W)	Energy (kWh)
0:24:36	122,93*	1,32*	151,21*	1,328*

Fonte: O autor

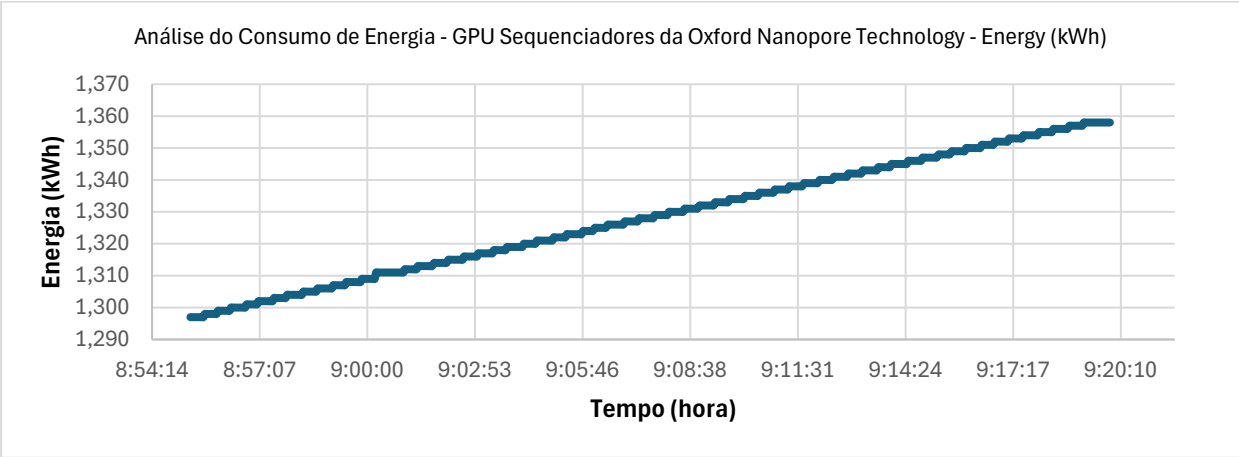
*valores médios

Gráfico 6 - Consumo de Energia em Sequenciadores da Oxford Nanopore Technology (GPU)



Fonte: O autor

Gráfico 7 - Consumo de Energia em Sequenciadores da Oxford Nanopore Technology (GPU)



Fonte: O autor

Para fins de delimitação do escopo experimental, a análise restringiu-se ao consumo energético direto dos componentes de processamento (CPU e GPU). Não foram contabilizados no cálculo final os gastos associados à infraestrutura predial, como

sistemas de climatização (ar-condicionado) e iluminação. Adicionalmente, o estudo quantificou a carga residual de processos do sistema operacional executados em segundo plano (background processes), e não foi considerado eventuais ineficiências decorrentes de flutuações de tensão ou ruídos harmônicos provenientes da rede elétrica pública.

5.2 Comparativo de Consumo: CPU vs. GPU

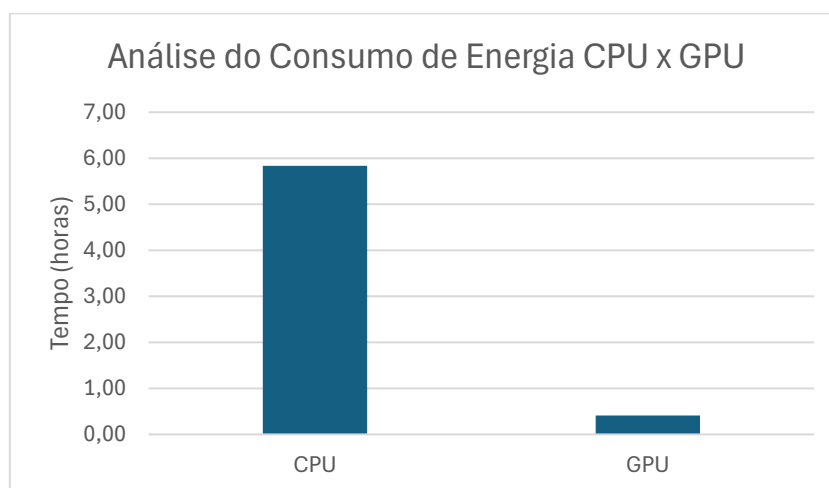
Na simulação dos 64Gb do Desktop estão disponíveis 62 Gb, vale ressaltar que devido limitações computacionais foi utilizado um sub dataset para realizar na etapa de basecalling, tradução de dados brutos de sequenciamento da Tecnologia Oxford Nanopore, em informações biológicas. Traçando um comparativo entre as simulações obtivemos os seguintes dados:

Tabela 6 – Comparativo do Consumo de Energia CPU x GPU

	CPU	GPU
Tempo da Simulação (H)	5:50:24	0:24:36
Energia Consumida (kJ)	19641,37	1960,33

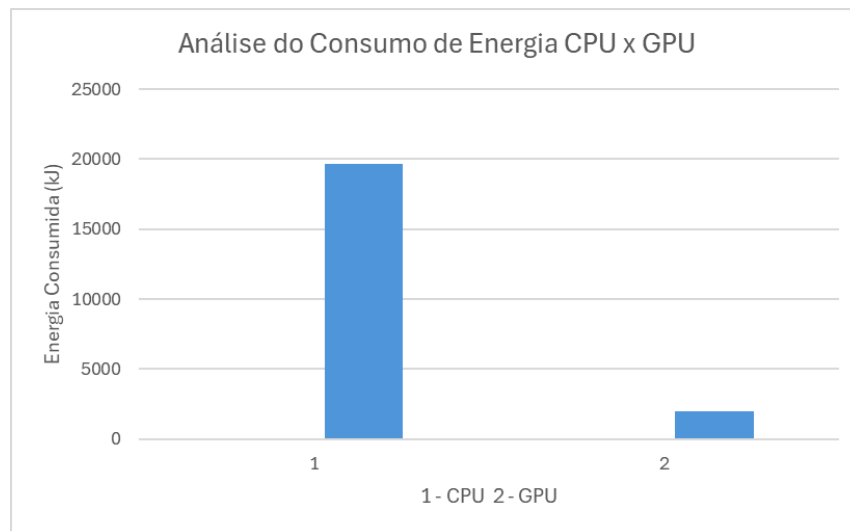
Fonte: O autor

Gráfico 8 - Comparativo do Consumo de Energia CPU x GPU



Fonte: O autor

Gráfico 9 - Comparativo do Consumo de Energia CPU x GPU



Fonte: O autor

5.3 Quantificação do consumo do sistema operacional:

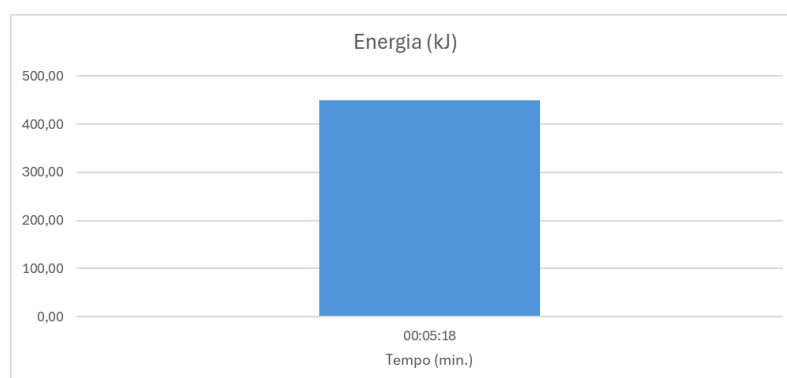
Para a obtenção de dados de consumo energético apenas da simulação computacional, quantificamos o consumo energético do ambiente computacional apenas com o sistema operacional rodando e obtivemos os dados expostos abaixo, que demonstram comparativamente o alto consumo energético quando a simulação é posta em marcha.

Tabela 7 – Energia Consumida – Ambiente Computacional

Consumo - CPU sistema operacional	
Tempo total(mim)	Energia (kJ)
00:05:18	448,75

Fonte: O autor

Gráfico 10 – Energia Consumida – Ambiente Computacional



Fonte: O autor

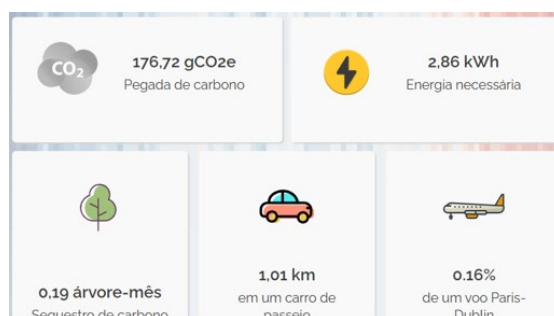
5.4 Pegada de Carbono e Avaliação com Green Algorithms:

Com os dados de consumo obtidos das simulações, foi realizada uma análise comparativa utilizando a plataforma Green Algorithms (www.green-algorithms.org). Esta ferramenta permite quantificar o impacto ambiental, fornecendo os dados de consumo de CPU e GPU e relacionando-os diretamente com a emissão de quilogramas de CO₂ equivalente (Kg de CO₂e), em seu artigo "The Carbon Footprint of Bioinformatics", Grealey (2022), quantifica as emissões de gases de efeito estufa (GEEs) associadas a diversas ferramentas e metodologias, como estudos de associação genômica (GWAS), sequenciamento de RNA, montagem de genomas, metagenômica, filogenética e simulações moleculares. Utilizando o modelo Green Algorithms (www.green-algorithms.org)(2025), o estudo revela variações significativas na pegada de carbono, mesmo entre softwares com desempenho similar.

Os resultados demonstram que a escolha de ferramentas e a configuração dos recursos computacionais são determinantes para o impacto ambiental. Por exemplo, a paralelização de tarefas, embora eficaz na redução do tempo de execução, pode paradoxalmente aumentar as emissões totais de GEEs. Da mesma forma, o uso de Unidades de Processamento Gráfico (GPUs), apesar de sua rapidez, nem sempre representa uma redução na pegada de carbono em comparação com as Unidades Centrais de Processamento (CPUs) tradicionais. Fatores como a eficiência energética do centro de dados — que pode reduzir as emissões em até 34% — e a localização geográfica da computação, privilegiando países com baixa intensidade de carbono (CI), são cruciais.

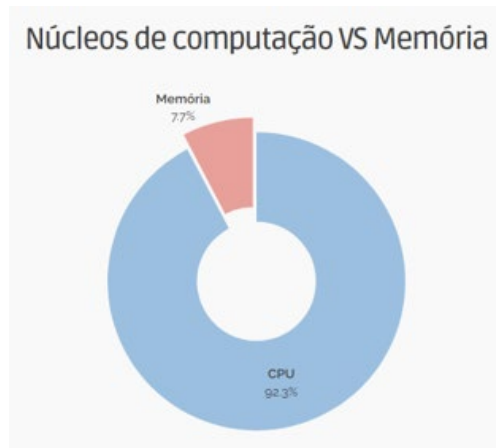
Dados da CPU:

Figura 4 – Pegada de Carbono - CPU



Fonte: www.green-algorithms.org

Figura 5 – Uso de memória- CPU



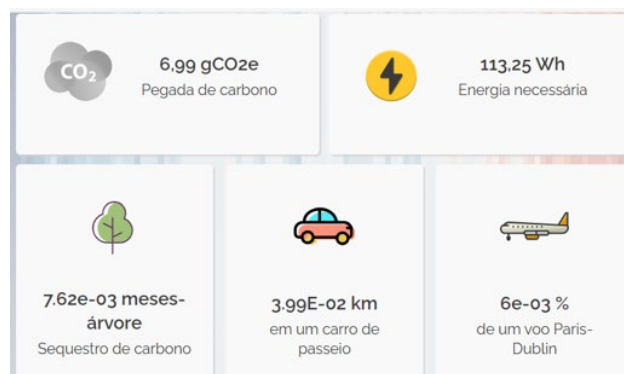
Fonte: www.green-algorithms.org

Figura 6 – Pegada de CO₂ por local - CPU

Fonte: www.green-algorithms.org

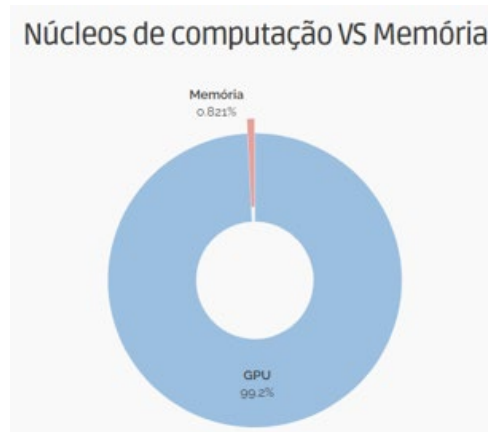
Dados da GPU:

Figura 7 – Pegada de Carbono - GPU

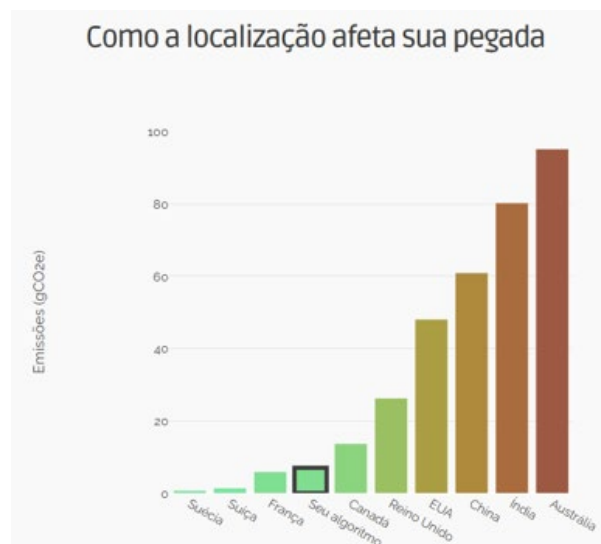


Fonte: www.green-algorithms.org

Figura 8 – Uso de memória- GPU



Fonte: www.green-algorithms.org

Figura 9 – Pegada de CO₂ por local - GPU

Fonte: www.green-algorithms.org

5.5 Simulação de Organismos Procariontes

Dando continuidade ao desenvolvimento metodológico para a quantificação do consumo energético em bioinformática, esta seção detalha os procedimentos adotados para a coleta de dados de simulação dos genomas completos de *Haemophilus Haemolyticus*, *Staphylococcus Aureus* e *Stenotrophomonas Maltophilia*. O propósito central reside na aplicação do modelo de monitoramento fundamentado na integração Arduino Uno e sensor PZEM-004T para a mensuração do custo energético real no

processamento genômico desses patógenos. Complementarmente, utilizou-se o software *PowerTop* para o monitoramento intrínseco do sistema operacional na estação de trabalho onde as simulações foram processadas.

A seleção dos organismos justifica-se pela diversidade de suas complexidades genômicas e pela relevância no cenário clínico. Os espécimes *Haemophilus haemolyticus* e *Staphylococcus aureus* representam genomas bacterianos procariontes, com ênfase na identificação de perfis de resistência por meio do sequenciamento por nanoporos. Já o *Stenotrophomonas Maltophilia* destaca-se por sua alta plasticidade genômica, o que demanda um esforço computacional superior durante as fases críticas de *basecalling* e montagem. Estes organismos são oriundos da Dissertação de Mestrado do Acadêmico Adriano Terra do Combi-Lab.

As simulações foram conduzidas no Laboratório de Bioinformática (Combi-Lab) do Centro de Ciências Computacionais (C3) da FURG, sob os seguintes parâmetros técnicos:

- **Processamento de Alto Desempenho:** Priorizou-se o uso da GPU NVIDIA GeForce GTX 1050 Ti (arquitetura Pascal, 768 núcleos CUDA). Esta escolha fundamentou-se em resultados experimentais prévios que indicaram uma redução de aproximadamente dez vezes no consumo energético (0,54 kWh via GPU em comparação a 5,46 kWh via CPU) para cargas de trabalho equivalentes.
- **Instrumentação e Monitoramento:** A coleta contínua (*full time*) de grandezas elétricas, como tensão (V), corrente (A) e potência ativa (W), foi realizada pelo protótipo desenvolvido com Arduino e PZEM-004T.
- **Tradução Genômica (*Basecalling*):** Empregou-se o software *Dorado* para a conversão de sinais elétricos brutos, provenientes dos dispositivos de nanoporos, em sequências lineares de nucleotídeos (A, T, C, G).

Para cada organismo, os dados brutos de consumo foram processados e submetidos às análises de eficiência e impacto ambiental:

- **Eficiência Energética:** Determinação do índice de consumo de eletricidade expresso em quilowatt-hora por Gigabyte (kWh/Gb) processado.
- **Pegada de Carbono:** Estimativa das emissões de dióxido de carbono equivalente (gCO₂e) por meio da plataforma *Green Algorithms*. Esta análise correlaciona o

tempo de execução algorítmica com a intensidade de carbono da matriz energética local.

- Alinhamento aos ODS: Avaliação do impacto da otimização computacional genômica frente aos Objetivos de Desenvolvimento Sustentável da Agenda 2030 da ONU, especificamente o ODS 3 (Saúde e Bem-Estar) e o ODS 9 (Indústria, Inovação e Infraestrutura).

As configurações das etapas do processamento foram as que seguem:

- A etapa inicial, basecalling e conversão de sinais com o Dorado, consistiu na conversão dos sinais brutos de corrente elétrica, armazenados em arquivos de formato POD5, em sequências de nucleotídeos (reads). Para tanto, utilizou-se o software Dorado (versão v4.1.0), operando com o subcomando basecaller. O modelo preditivo aplicado foi o `dna_r10.4.1_e8.2_400bps_fast@v4.1.0`, selecionado por sua compatibilidade com a química de poro R10.4.1. A execução priorizou os parâmetros padrão da ferramenta para garantir a integridade da conversão, mantendo o registro de tempo (time) ativado para fins de rastreabilidade do processamento.
- Após a geração das sequências no formato FASTQ, realizou-se o controle de qualidade por meio do software NanoPlot. Esta etapa é fundamental para a avaliação estatística das reads provenientes do basecalling. Foram extraídas métricas críticas como o N50, o comprimento médio das sequências e a distribuição dos scores de qualidade (Phred score). Tais dados permitiram uma análise diagnóstica da biblioteca antes das etapas subsequentes de filtragem e montagem.
- Na sequência, visando otimizar a montagem e reduzir o ruído computacional, aplicou-se o software Filtlong para o refinamento das reads. Os parâmetros foram ajustados para remover sequências menores que 1000 pares de bases (`--min_length 1000`) e com qualidade média inferior a 7 (`--min_mean_q 7`). Estabeleceu-se um alvo de 80 Mb de dados (`--target_bases 80.000.000`). Essa estratégia assegura uma cobertura de aproximadamente 20-40x para genomas bacterianos (estimados entre 2-5 Mb), provendo redundância suficiente para uma

montagem robusta com reads longas, enquanto minimiza o custo de memória e tempo de processamento.

- A montagem do genoma foi executada pelo algoritmo Flye, utilizando as reads longas filtradas como entrada através do parâmetro --nano-raw. O tamanho do genoma foi estimado em 2.0 Mb (--genome-size 2.0m) para orientar o algoritmo na resolução de repetições e na organização dos contigs. O processo foi paralelizado com o uso de 20 threads para acelerar a execução, organizando-se os resultados em diretórios específicos para análise posterior.
- Na etapa final de refinamento, utilizou-se o software Medaka para realizar o polimento (polishing) do consenso gerado pelo Flye. O objetivo foi a correção de erros residuais de inserção e deleção (indels), típicos da tecnologia de nanoporos. O polimento foi conduzido utilizando o modelo r1041_e82_400bps_fast_g632, garantindo estrita compatibilidade com o modelo de basecalling inicialmente empregado, resultando em uma sequência genômica final de alta acurácia.

As tabelas a seguir contêm os dados dos softwares utilizados na simulação, bem como o tamanho dos arquivos processados:

Tabela 8 – Organismo Haemophilus Haemolyticus

Organismo	Etapa	Software	Tamanho Arquivo(Mb)
Haemophilus Haemolyticus	Base calling	Dorado Fast	178,40
Haemophilus Haemolyticus	Controle da qualidade das reads	Nanoplot	5,90
Haemophilus Haemolyticus	Filtragem das reads	Filtlong	160,60
Haemophilus Haemolyticus	Montagem	Flye	34,60
Haemophilus Haemolyticus	Controle de Qualidade da Montagem	QUAST	4,60
Haemophilus Haemolyticus	Controle de Qualidade da Montagem	BUSCO	9,10
Haemophilus Haemolyticus	Polimento	Medaka	185,60
		Total	578,80

Fonte: Acadêmico de Mestrado Adriano Terra

Tabela 9 – Organismo Staphylococcus Aureus

Organismo	Etapa	Software	Tamanho Arquivo(Mb)
Staphylococcus Aureus	Base calling	Dorado Fast	484,00
Staphylococcus Aureus	Controle da qualidade das reads	Nanoplot	5,80
Staphylococcus Aureus	Filtragem das reads	Filtlong	435,60
Staphylococcus Aureus	Montagem	Flye	49,40
Staphylococcus Aureus	Controle de Qualidade da Montagem	QUAST	2,50
Staphylococcus Aureus	Controle de Qualidade da Montagem	BUSCO	11,20
Staphylococcus Aureus	Polimento	Medaka	355,70
		Total	1344,20

Fonte: Acadêmico de Mestrado Adriano Terra

Tabela 10 – Organismo *Stenotrophomonas Maltophilia*

Organismo	Etapa	Software	Tamanho Arquivo(Mb)
<i>Stenotrophomonas Maltophilia</i>	Base calling	Dorado Fast	991,10
<i>Stenotrophomonas Maltophilia</i>	Controle da qualidade das reads	Nanoplot	5,90
<i>Stenotrophomonas Maltophilia</i>	Filtragem das reads	Filtlong	892,00
<i>Stenotrophomonas Maltophilia</i>	Montagem	Flye	81,30
<i>Stenotrophomonas Maltophilia</i>	Controle de Qualidade da Montagem	QUAST	2,80
<i>Stenotrophomonas Maltophilia</i>	Controle de Qualidade da Montagem	BUSCO	17,80
<i>Stenotrophomonas Maltophilia</i>	Polimento	Medaka	803,40
		Total	2794,30

Fonte: Acadêmico de Mestrado Adriano Terra

O processo de sequenciamento genético em bioinformática, especificamente voltado para a análise de dados de nanoporos como os utilizados nos estudos do Combi-Lab, seguem um fluxo de trabalho rigoroso para transformar sinais elétricos em genomas confiáveis. Abaixo, descreve-se as etapas conforme os padrões técnicos consolidados da área de estudo:

- **Basecalling:** É o estágio inicial onde sinais elétricos brutos, gerados pela passagem do DNA através de um nanoporo, são traduzidos em sequências de nucleotídeos (A, T, C, G). Este processo é altamente intensivo computacionalmente e frequentemente utiliza algoritmos de redes neurais e aceleração por GPU para garantir precisão e velocidade.
- **Controle de Qualidade das Reads:** Após a tradução, as sequências brutas (reads) são avaliadas para verificar métricas como a pontuação de qualidade Phred (Q-score), que indica a probabilidade de erro em cada base chamada.
- **Filtragem das Reads:** Nesta etapa, removem-se as sequências que não atingem os critérios mínimos de comprimento ou qualidade, além de eliminar adaptadores técnicos e possíveis contaminantes, garantindo que apenas dados confiáveis sigam para a montagem.
- **Montagem:** As reads filtradas são sobrepostas e organizadas por algoritmos específicos para reconstruir a sequência contínua do genoma original (contigs). Em organismos com alta plasticidade genômica, como o *Stenotrophomonas maltophilia*, esta etapa exige estratégias computacionais mais robustas.

- Controle de Qualidade da Montagem (QUAST e BUSCO):
 - QUAST: Avalia a continuidade da montagem através de métricas estatísticas como o N50, o número de contigs e o tamanho total do genoma reconstruído.
 - BUSCO: Verifica a completitude biológica do genoma ao procurar por genes ortólogos universais que deveriam estar presentes de forma única no grupo taxonômico do organismo.
- Polimento: É o refinamento final da montagem, onde as reads originais (ou sequências de alta precisão) são mapeadas de volta ao genoma montado para corrigir erros residuais de base, como inserções ou deleções (indels), melhorando a acurácia da sequência final.

Optou-se pela utilização do quilojoule (kJ) como unidade de quantificação do consumo energético em detrimento do quilowatt-hora (kWh). Tal escolha justifica-se pela escala de magnitude dos arquivos de organismos procariotos, cujas simulações processam volumes de dados reduzidos, tornando o kJ uma unidade de medida mais sensível e adequada para representar as flutuações de consumo observadas.

5.5.1. Simulação de Organismo *Haemophilus Haemolyticus*

A simulação do genoma do organismo *Haemophilus Haemolyticus* foi executada com as etapas de Basecalling, Controle da Qualidade das Reads, Filtragem das Reads, Montagem, Controle de Qualidade da Montagem (QUALI e BUSCO) e Polimento, abaixo mostramos a tabela de resumo dos dados coletados na simulação computacional.

Tabela 11 – Resumo Simulação *Haemophilus Haemolyticus*

Etapa	Haemophilus Haemolyticus						
	Tamanho Arquivo(Mb)	Tempo da Simulação (s)	Energia Consumida na Simulação (kJ)	Energia Consumida na Simulação (J)	Energia Consumida na Simulação (kW.h)	Taxa Eficiência (KJ / Mb)	Taxa Eficiência (kWh / Mb)
Basecalling	178,40	300	4,638	4638,369	0,083333	0,0260	0,000467
Ctr Qual Reads	5,90	7	0,108	108,470	0,001944	0,0184	0,000330
Filtragem das reads	160,60	7	0,108	108,469	0,001944	0,0007	0,000012
Montagem	34,60	132	2,046	2045,803	0,036667	0,0591	0,001060
Ctr Qual Quasti	4,60	6	0,093	93,005	0,001667	0,0202	0,000362
Ctr Qual. Busco	9,10	12	0,186	186,036	0,003333	0,0204	0,000366
Polimento	185,60	46	0,713	713,239	0,012778	0,0038	0,000069

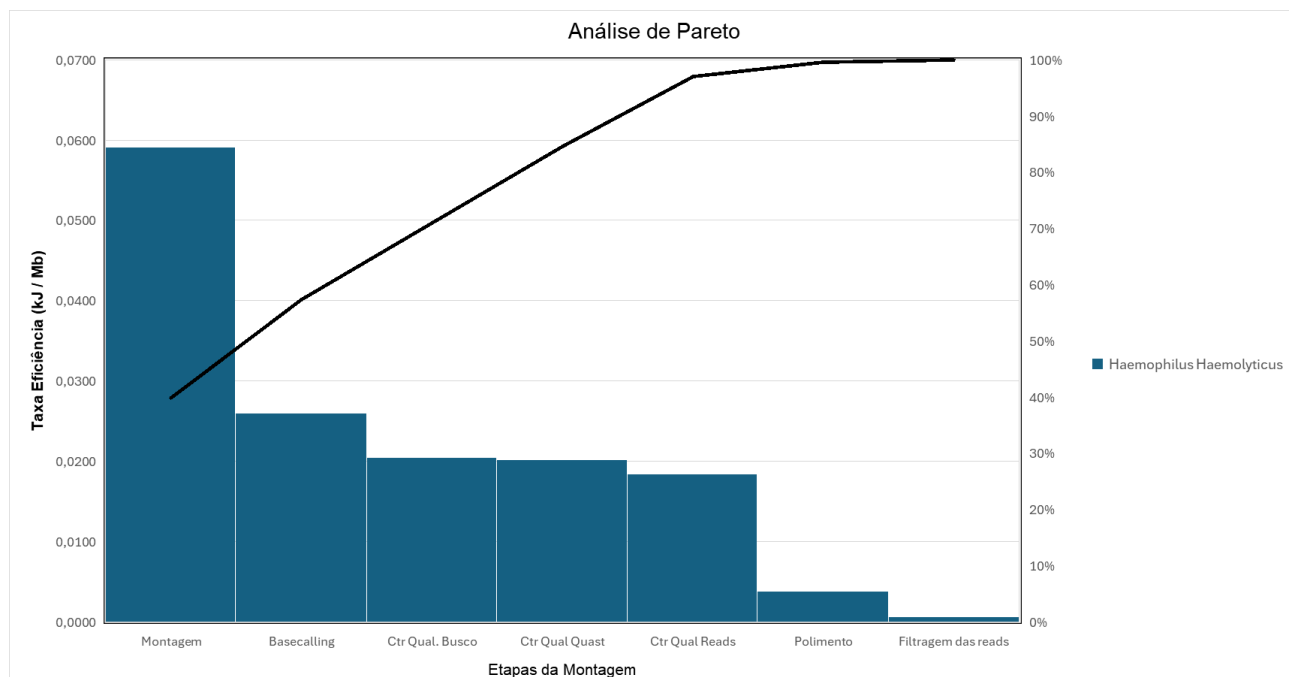
Fonte – O autor

Nota

*Ctr Qual. Reads = Controle de Qualidade da Montagem das Reads

*Ctr Qual. Quast = Controle de Qualidade da Montagem - Quast

*Ctr Qual. Busco = Controle de Qualidade da Montagem – Busco

Gráfico 11 – Eficiência Energética - *Haemophilus Haemolyticus*

Fonte – O autor

A análise do Diagrama de Pareto aplicado aos dados experimentais do processamento da bactéria *Haemophilus Haemolyticus* permite identificar os gargalos críticos para a otimização da eficiência energética. Tal abordagem fundamenta-se na premissa de que a infraestrutura computacional utilizada na bioinformática possui uma pegada de carbono significativa, frequentemente subestimada em comparação com outras indústrias.

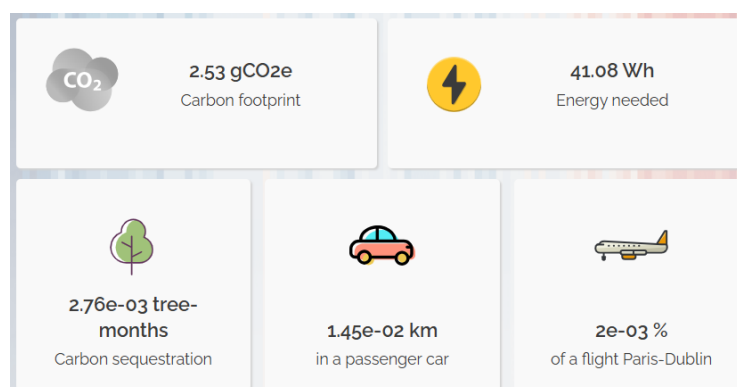
Observa-se que a etapa de Montagem (Software Flye) e os processos de Controle de Qualidade, somados ao *Basecalling*, representam aproximadamente 85% do impacto total na taxa de eficiência (kWh/Mb). Embora o *Basecalling* (Dorado Fast) registre o maior consumo absoluto de energia (4.638,369 J ou 4,38 kJ) e o maior tempo de execução (300s), a Montagem demonstra ser o processo menos eficiente proporcionalmente. Este fenômeno ocorre pois, apesar de consumir menos energia total, o volume de dados processados na Montagem é significativamente menor, resultando em uma taxa de 0,0591 kJ/Mb.

Em consonância com as observações de Grealey et al. em "*The Carbon Footprint of Bioinformatics*", ressalta-se que a escolha de algoritmos e a eficiência do software são determinantes para a sustentabilidade da disciplina. No presente estudo, etapas como o

Polimento (Medaka) e a Filtragem das reads (Filtlong) situam-se na zona dos "muitos triviais" do diagrama. A filtragem, especificamente, apresenta a melhor taxa de eficiência (0,000012 kJ/Mb), indicando que esforços de otimização focados nestas fases finais teriam um retorno marginal no que tange à redução da pegada de carbono total do pipeline.

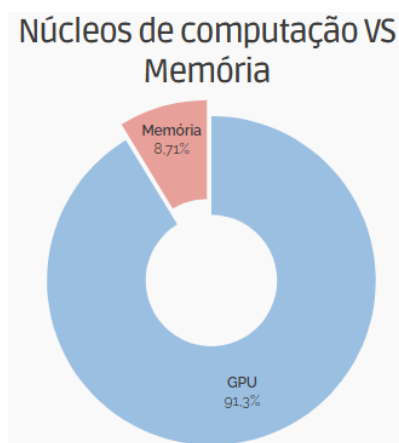
Podemos observar com a análise de Pareto que a quantificação da eficiência energética para as etapas da montagem do referido genoma foi de 0,0260 kJ/Mb para o basecalling, 0,0184 kJ/Mb para o Controle de Qualidade das Reads, 0,0007 kJ/Mb para a Filtragem das Reads, 0,0591 kJ/Mb para a Montagem(mais ineficiente), 0,0202 kJ/Mb para o Controle de Qualidade (Quast), 0,0204 kJ/Mb para o Controle de Qualidade(Busco) e 0,0038 kJ/Mb para o Polimento.

Figura 10 – Pegada de Carbono - Haemophilus Haemolyticus



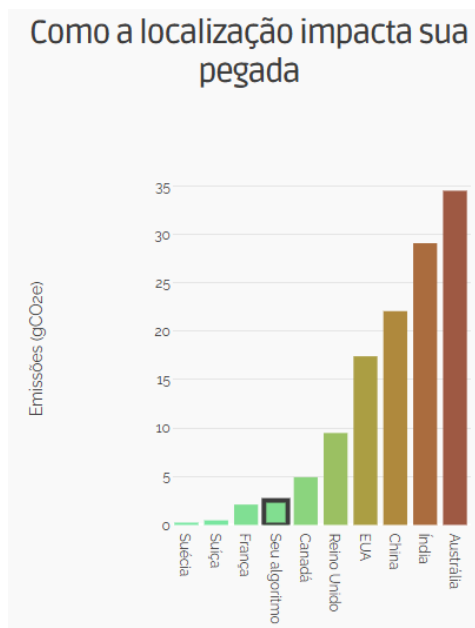
Fonte: www.green-algorithms.org

Figura 11 – Uso de memória - Haemophilus Haemolyticus



Fonte: www.green-algorithms.org

Figura 12 – Pegada de CO₂ por local - Haemophilus Haemolyticus



Fonte: www.green-algorithms.org

5.5.2 Simulação de Organismo *Staphylococcus Aureus*

A simulação do genoma do organismo *Staphylococcus Aureus* foi executada com as etapas de Basecalling, Controle da Qualidade das Reads, Filtragem das Reads, Montagem, Controle de Qualidade da Montagem (QUALI e BUSCO) e Polimento, abaixo mostramos a tabela de resumo dos dados coletados na simulação computacional.

Tabela 12 – Resumo Simulação *Staphylococcus Aureus*

Etapa	Staphylococcus Aureus						
	Tamanho Arquivo(Mb)	Tempo da Simulação(s)	Energia Consumida na Simulação (kJ)	Energia Consumida na Simulação (J)	Energia Consumida na Simulação (kW.h)	Taxa Eficiência (kJ / Mb)	Taxa Eficiência (kWh / Mb)
Basecalling	484,00	499	7,719	7718,612	55,685376	0,0159	0,115052
Ctr Qual Reads	5,80	10	0,155	155,070	55,825333	0,0267	9,625057
Filtragem das reads	435,60	6	0,093	93,045	55,827000	0,0002	0,128161
Montagem	49,40	571	8,856	8856,012	55,834752	0,1793	1,130258
Ctr Qual Quast	2,50	4	0,062	62,046	55,841667	0,0248	22,336667
Ctr Qual. Busco	11,20	7	0,109	108,586	55,844000	0,0097	4,986071
Polimento	355,70	59	0,915	915,272	55,847103	0,0026	0,157006

Fonte – O autor

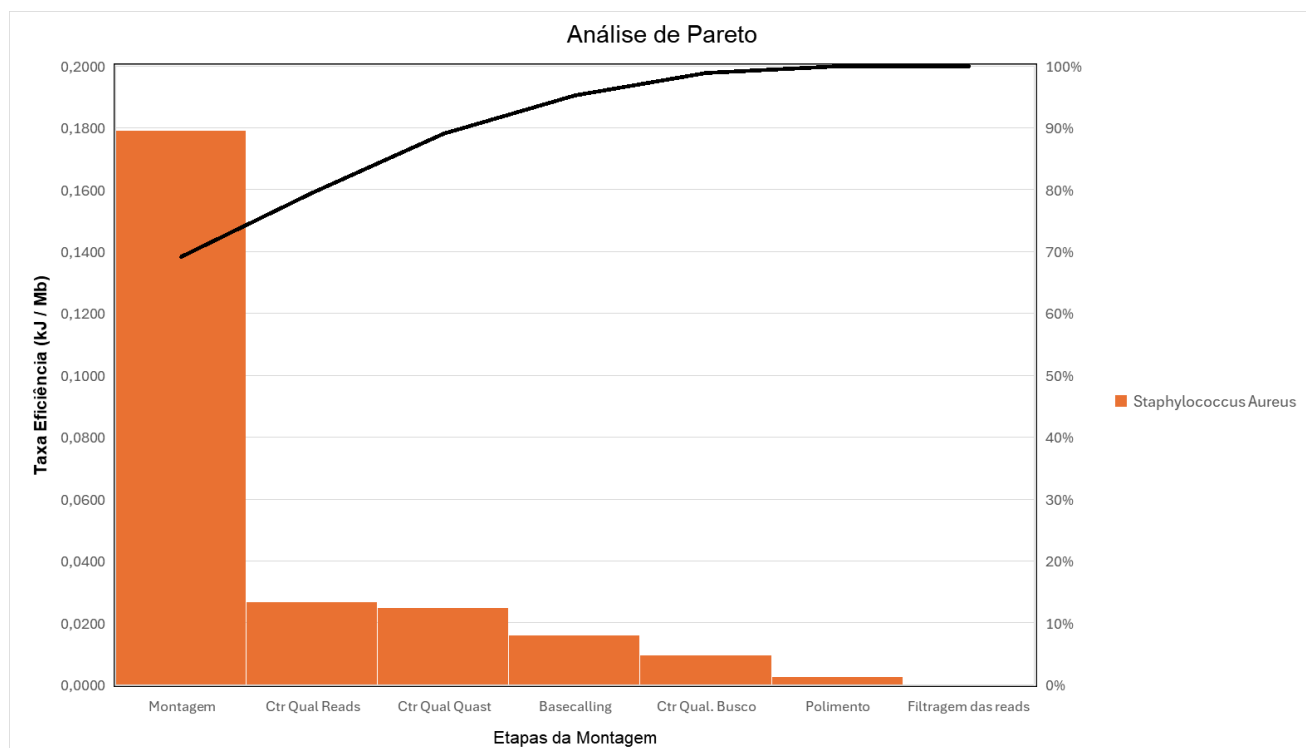
Nota

*Ctr Qual. Reads = Controle de Qualidade da Montagem das Reads

*Ctr Qual. Quast = Controle de Qualidade da Montagem - Quast

*Ctr Qual. Busco = Controle de Qualidade da Montagem – Busco

Gráfico 12 – Eficiência Energética - *Staphylococcus Aureus*



Fonte – O autor

A análise do Diagrama de Pareto aplicado aos dados experimentais do processamento da bactéria *Staphylococcus Aureus* permite identificar os gargalos críticos para a otimização da eficiência energética. Tal abordagem fundamenta-se na premissa de que a infraestrutura computacional utilizada na bioinformática possui uma pegada de carbono significativa, frequentemente subestimada em comparação com outras indústrias.

Observa-se que a etapa de Montagem (Software Flye) e os processos de Controle de Qualidade, somados ao *Basecalling*, representam aproximadamente 85% do impacto total na taxa de eficiência (kWh/Mb). Embora o *Basecalling* (Dorado Fast) registre o maior consumo absoluto de energia (7.718,612 J ou 7,719 kJ) e o maior tempo de execução (499s), a Montagem demonstra ser o processo menos eficiente proporcionalmente. Este fenômeno ocorre pois, apesar de consumir menos energia total, o volume de dados processados na Montagem é significativamente menor, resultando em uma taxa de 0,1793 kJ/Mb.

Em consonância com as observações de Grealey et al. em "*The Carbon Footprint of Bioinformatics*", ressalta-se que a escolha de algoritmos e a eficiência do software são determinantes para a sustentabilidade da disciplina. No presente estudo, etapas como o Polimento (Medaka) e a Filtragem das reads (Filtlong) situam-se na zona dos "muitos

triviais" do diagrama. A filtragem, especificamente, apresenta a melhor taxa de eficiência (0,0002 kJ/Mb), indicando que esforços de otimização focados nestas fases finais teriam um retorno marginal no que tange à redução da pegada de carbono total do pipeline.

Ao observar a linha de porcentagem acumulada no Diagrama de Pareto, nota-se que as três primeiras etapas (Montagem, Basecalling e Controle de Qual. Busco) são responsáveis por aproximadamente 75% da ineficiência acumulada do sistema. Basecalling (Dorado Fast), apresentou um maior consumo absoluto de energia (7,719 kJ) e o maior tempo de simulação (499s). No entanto, por processar um volume de dados muito superior (484,00 Mb), sua taxa de eficiência (0,0159 kJ/Mb) é mais eficiente que a da Montagem.

Controle de Qualidade, Busco/Quast, apresentaram taxas de eficiência situando-se em um patamar intermediário de impacto. As etapas de Polimento (Medaka) e Filtragem das reads (Filtlong) compõem a cauda do gráfico de Pareto. A Filtragem das reads demonstrou ser o processo mais eficiente do pipeline, com uma taxa de apenas 0,0026⁵ kJ/Mb. Mesmo lidando com um volume expressivo de dados (435,60 Mb). Do ponto de vista da sustentabilidade computacional, investimentos em otimização nestas etapas podem trazer retornos insignificantes para o conceito de pegada de carbono total do experimento.

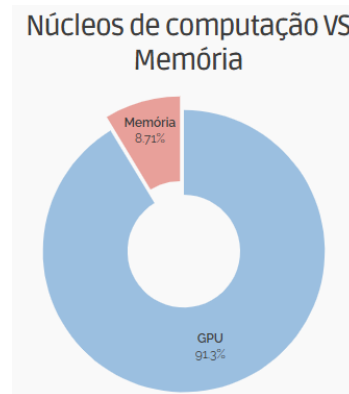
Assim a quantificação da eficiência energética para as etapas da montagem do referido genoma foi de 0,0159 kJ/Mb para o basecalling, 0,0267 kJ/Mb para o Controle de Qualidade das Reads, 0,0002 kJ/Mb para a Filtragem das Reads, 0,1793 kJ/Mb para a Montagem, 0,0247 kJ/Mb para o Controle de Qualidade (Quast), 0,0097 kJ/Mb para o Controle de Qualidade(Busco) e 0,0026 kJ/Mb para o Polimento.

Figura 13 – Pegada de Carbono - Staphylococcus Aureus



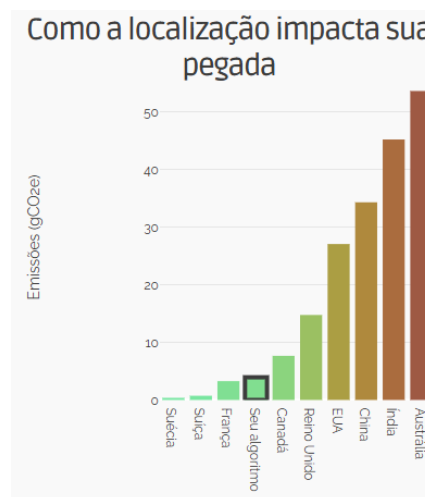
Fonte: www.green-algorithms.org

Figura 14 – Uso de memória - Staphylococcus Aureus



Fonte: www.green-algorithms.org

Figura 15 – Pegada de CO₂ por local - - Staphylococcus Aureus



Fonte: www.green-algorithms.org

5.5.3 Simulação de Organismo Stenotrophomonas Maltophilia

A simulação do genoma do organismo *Stenotrophomonas Maltophilia* foi executada com as etapas de Basecalling, Controle da Qualidade das Reads, Filtragem das Reads, Montagem, Controle de Qualidade da Montagem (QUALI e BUSCO) e Polimento, abaixo mostramos a tabela de resumo dos dados coletados na simulação computacional.

Tabela 13 – Resumo Simulação *Stenotrophomonas Maltophilia*

Stenotrophomonas Maltophilia							
Etapas	Tamanho Arquivo(Mb)	Tempo da Simulação(s)	Energia Consumida na Simulação (kJ)	Energia Consumida na Simulação (J)	Energia Consumida na Simulação (kW.h)	Taxa Eficiência (kJ / Mb)	Taxa Eficiência (kWh / Mb)
Basecalling	991,10	1175	18,189	18188,878	55,728	0,0184	0,0562
Ctrl Qual Reads	5,90	16	0,248	248,233	55,852	0,0421	9,4665
Filtragem das reads	892,00	16	0,248	248,233	55,852	0,0003	0,0626
Montagem	81,30	406	6,300	6300,220	55,864	0,0775	0,6871
Ctrl Qual Quast	2,80	5	0,078	77,601	55,873	0,0277	19,9546
Ctrl Qual. Busco	17,80	12	0,186	186,247	55,874	0,0105	3,1390
Polimento	803,40	149	2,313	2312,762	55,879	0,0029	0,0696

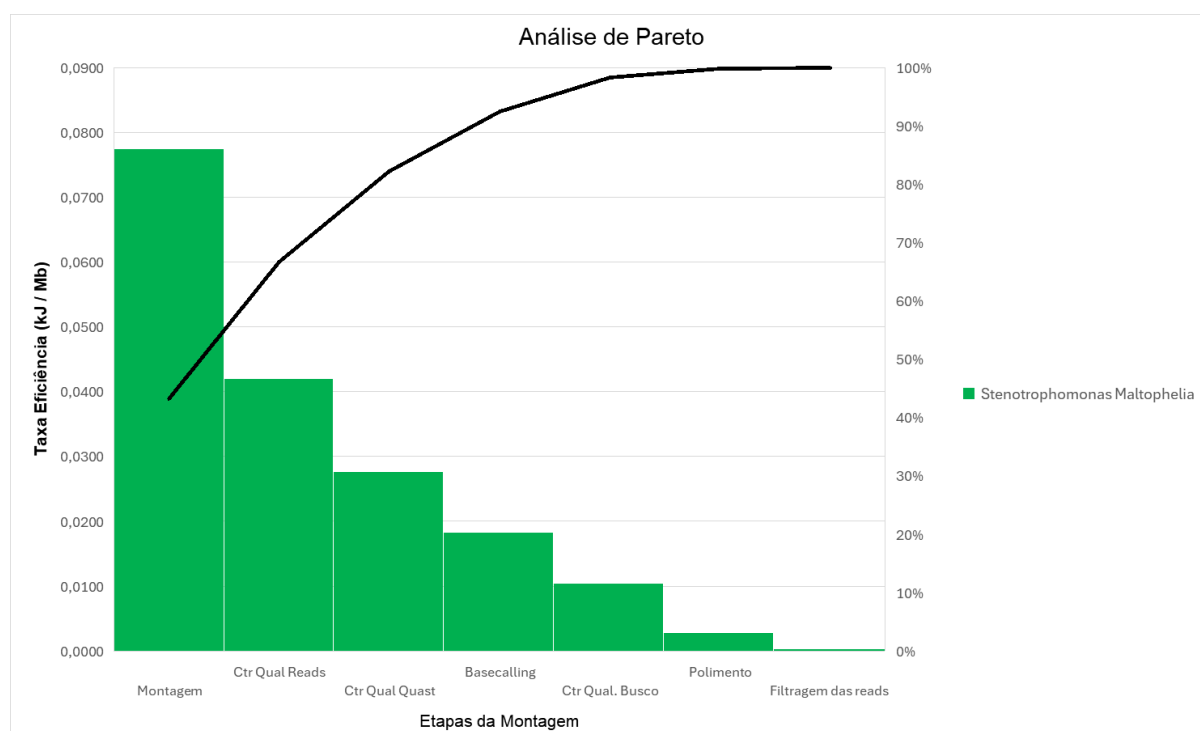
Fonte – O autor

Nota

*Ctrl Qual. Reads = Controle de Qualidade da Montagem das Reads

*Ctrl Qual. Quast = Controle de Qualidade da Montagem - Quast

*Ctrl Qual. Busco = Controle de Qualidade da Montagem – Busco

Gráfico 13 – Eficiência Energética - *Stenotrophomonas Maltophilia*

Fonte – O autor

O processamento da *Stenotrophomonas Maltophilia* envolveu o maior volume de dados entre as amostras analisadas (2794,30 Mb), resultando em um consumo total de 27,562 kJ. A aplicação do Diagrama de Pareto sobre a Taxa de Eficiência permite isolar as etapas que exigem atenção prioritária sob a ótica da "Bioinformática Verde".

Assim como observado nas simulações anteriores, a Montagem (Software Flye) apresenta-se como o maior gargalo de eficiência energética. Nesta etapa obtivemos um

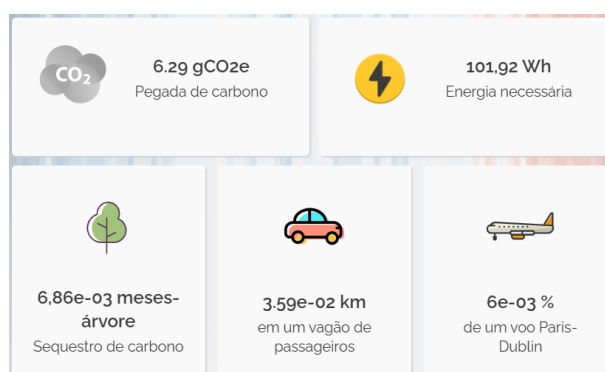
resultado de 0,0775 kJ/Mb, a mais alta do pipeline. Embora o tempo de simulação (406s) e a energia consumida (6,30 kJ) sejam menores que os do Basecalling, o custo por megabyte processado é significativamente superior, colocando-a no topo dos "pontos vitais" de Pareto.

O Basecalling (Dorado Fast) demonstrou o maior consumo absoluto de energia (18,189 kJ) e o maior tempo de execução (1175s). Apesar da alta demanda energética, a taxa de eficiência permanece em 0,0184 kJ/Mb. Isso indica que, embora o *Basecalling* seja o processo que mais "pesa" em termos de consumo, ele é mais eficiente que a montagem na conversão de energia em dados processados.

A curva de porcentagem acumulada (linha preta) mostra que a Montagem e o Basecalling sozinhos respondem por aproximadamente 60% do impacto de ineficiência total. Se somarmos as etapas de controle de qualidade (BUSCO e QUAST), chegamos a quase 90% do custo de eficiência. Em contrapartida, etapas como Polimento e Filtragem das reads (com taxas de 0,0029 e 0,0003 respectivamente) representam a "cauda longa" do gráfico, onde o potencial de otimização energética é desprezível.

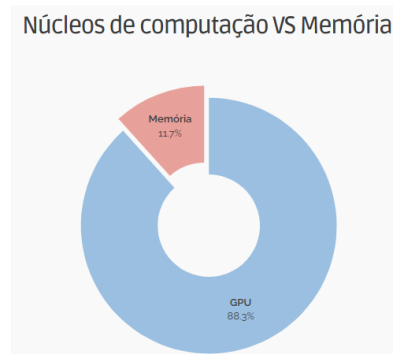
Assim a quantificação da eficiência energética para as etapas da montagem do referido genoma foi de 0,0184 kJ/Mb para o basecalling, 0,0421 kJ/Mb para o Controle de Qualidade das Reads, 0,0003 kJ/Mb para a Filtragem das Reads, 0,0775 kJ/Mb para a Montagem, 0,0277 kJ/Mb para o Controle de Qualidade (Quast), 0,0105 kJ/Mb para o Controle de Qualidade(Busco) e 0,0029 kJ/Mb para o Polimento.

Figura 16 – Pegada de Carbono - *Stenotrophomonas Maltophilia*



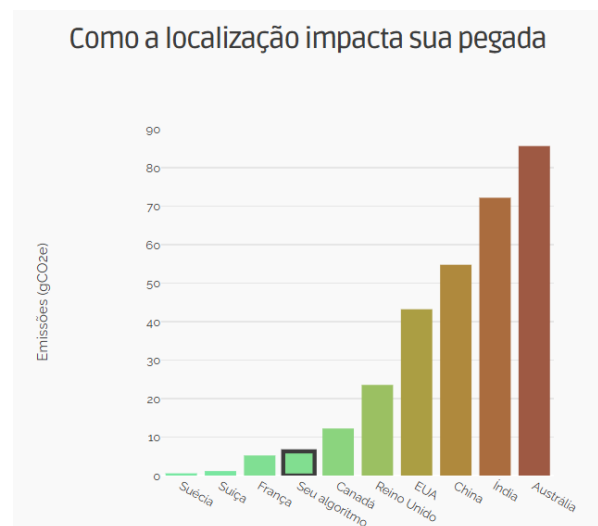
Fonte: www.green-algorithms.org

Figura 17 – Uso de memória - Stenotrophomonas Maltophelia



Fonte: www.green-algorithms.org

Figura 18 – Pegada de CO₂ por local - Stenotrophomonas Maltophelia



Fonte: www.green-algorithms.org

Além do monitoramento do consumo de energia em tempo real valendo-se do uso de sistema Arduino UNO / sensor PZEM-004T, utilizamos o software PowerTop para medição do consumo de energia, o qual não se mostrou satisfatório em termos de quantificação em tempo real da demanda de energia visto que sua métrica de demonstração do consumo de energia se dá através do percentual de utilização do hardware, o que em nossa para a nossa análise não tem o retorno esperado.

5.6 Discussão dos Resultados

A análise temporal demonstrou que o processamento utilizando a CPU consumiu aproximadamente 5 horas e 50 minutos (5:50:24). Em contrapartida, a GPU executou a mesma tarefa em apenas 24 minutos e 36 segundos (0:24:36). Esta diferença de desempenho refletiu-se diretamente no consumo de energia: o processamento via CPU consumiu um total de 5,46 kWh, enquanto o uso da GPU demandou apenas 0,54 kWh, representando um consumo cerca de dez vezes menor para a mesma carga de trabalho.

A avaliação do impacto ambiental foi complementada com a plataforma Green Algorithms, que quantifica a pegada de carbono de processos computacionais, o processamento com CPU resultou em uma pegada de carbono de 176,72 gCO₂e. A energia consumida foi de 2,86 kWh, o que correspondeu a 1,01 km percorrido por um carro e 0,16% de um voo entre Paris e Dublin. O uso da GPU, por sua vez, gerou uma pegada de carbono notavelmente inferior, de 6,99 gCO₂e. A energia necessária foi de 113,25 Wh. O impacto ambiental do uso da GPU foi comparado a 3,99E-02 km rodados em um carro e 6e-03% de um voo Paris-Dublin.

A análise do uso de recursos computacionais revelou que, no processamento via CPU, 77% da utilização foi destinada à memória. Já com a GPU, a utilização dos núcleos de computação alcançou 98,2%. A pesquisa em bioinformática tem destacado a crescente pegada de carbono da área, e a escolha de ferramentas e a configuração dos recursos computacionais são fatores cruciais para o impacto ambiental.

Embora a paralelização possa reduzir o tempo de execução, ela pode, paradoxalmente, aumentar as emissões totais de gases de efeito estufa. Do mesmo modo, a velocidade das GPUs não garante, por si só, uma redução na pegada de carbono em comparação com as CPUs tradicionais, o que ressalta a importância da eficiência energética dos centros de dados e da localização geográfica da computação, que deve privilegiar regiões com baixa intensidade de carbono.

A análise dos dados experimentais referentes ao processamento genômico das bactérias *Haemophilus haemolyticus*, *Staphylococcus aureus* e *Stenotrophomonas maltophilia* revela uma disparidade significativa na distribuição do consumo energético ao longo dos pipelines de bioinformática. Ao aplicar o Diagrama de Pareto sobre a métrica de Taxa de Eficiência (kWh/Mb), torna-se possível identificar os "poucos vitais" — etapas que, embora não necessariamente apresentem o maior consumo absoluto,

demonstram a menor eficiência na conversão de recursos computacionais em dados processados.

Historicamente, a bioinformática priorizou a acurácia biológica e o tempo de execução Gauthier et al., (2019). No entanto, conforme argumentado por Grealey et al. (2020), a pegada de carbono desta disciplina é substancial e frequentemente subestimada. A infraestrutura de Computação de Alto Desempenho (HPC), essencial para a era da bioinformática, Jawdat (2021), exige uma transição para modelos de "TI Sustentável", onde a otimização algorítmica é o principal vetor de mitigação ambiental.

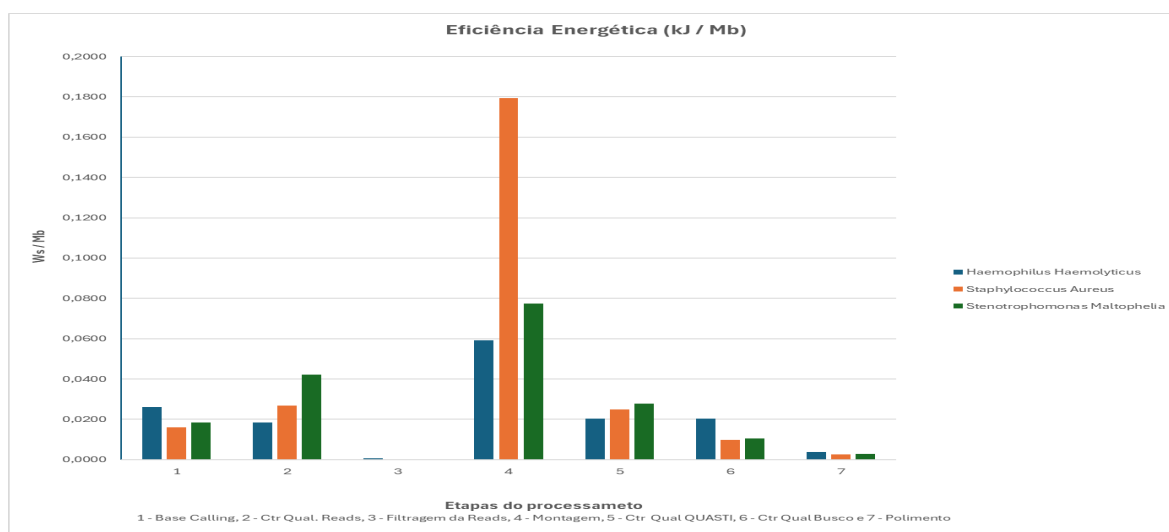
Podemos traçar um comparativo das simulações genômicas dos três organismos procariontes para relacionarmos a energia consumida durante o processo, este comparativo pode ser demonstrado pela tabela e gráficos abaixo.

Tabela 14 – Comparativo em kJ/Mb

Taxa Eficiência (kJ / Mb)			
Etapa	Haemophilus Haemolyticus	Staphylococcus Aureus	Stenotrophomonas Maltophilia
Basecalling	0,0260	0,0159	0,0184
Ctr Qual Reads	0,0184	0,0267	0,0421
Filtragem das reads	0,0007	0,0002	0,0003
Montagem	0,0591	0,1793	0,0775
Ctr Qual Quast	0,0202	0,0248	0,0277
Ctr Qual. Busco	0,0204	0,0097	0,0105
Polimento	0,0038	0,0026	0,0029

Fonte: O autor

Gráfico 14 – Comparativo por Etapa de Processamento



Fonte: O autor

A análise dos dados apresentados colabora para a transição da métrica experimental de Quilo Joule por Megabyte (kJ/Mb), ao invés da unidade de faturamento e análise industrial de quilowatt-hora por Megabyte (kW.h/Mb). Ao mapear o consumo energético detalhado em cada etapa do processamento bioinformático, onde como destaque a etapa de Montagem, apresentou os piores parâmetros de eficiência para os três organismos, os gráficos e tabelas fornecem a base empírica necessária para o escalonamento matemático da taxa. Considerando que 1 Joule = 1 / 3.600.000 kW.h, essa caracterização granular por microrganismo permite que a dissertação proponha um modelo de eficiência capaz de prever o impacto energético de grandes volumes de dados (Gb/Tb) em infraestruturas de computação de alto desempenho, transformando medições de tempo real em indicadores de sustentabilidade computacional, segundo a equação mostrada na figura 19.

Figura 19 – Taxa de Eficiência Energética

$$Taxa_{Eficiência} = \frac{Consumo_{Total}(kWh)}{Volume_{Dados}(Mb)}$$

Fonte: O autor

Em todos os cenários analisados, a etapa de Montagem (utilizando o software Flye) consolidou-se como o gargalo crítico de eficiência. No processamento da *H. haemolyticus*, a “Montagem” e o “Basecalling” (Dorado Fast) representaram 85% do impacto na taxa de eficiência. Embora o Basecalling registre o maior consumo absoluto e tempo de execução (devido ao uso intensivo de GPU), sua eficiência é superior à da Montagem. Para a *S. Aureus*, enquanto a “Montagem” processou apenas 49,40 Mb com uma taxa de 0,1793 kJ/Mb, o Basecalling processou 484,00 Mb com uma taxa mais favorável de 0,159 kJ/Mb. Tal fenômeno indica que a Montagem exige um esforço computacional desproporcional ao volume de saída, tornando-se o alvo primário para reengenharia de software sob a ótica da computação verde.

As etapas de Polimento (Medaka) e Filtragem das Reads (Filtlong) situam-se na zona menor impacto no consumo de energia do Diagrama de Pareto. Com taxas de eficiência na ordem 0,0002 a 0,0003 kJ/Mb, essas fases demonstram uma maturidade energética onde esforços adicionais de otimização resultariam em retornos marginais para a sustentabilidade total do sistema.

A integração entre Bioinformática, Internet das Coisas (IoT) e Computação Verde não é apenas uma necessidade técnica, mas um imperativo ético alinhado à Agenda 2030 (ONU, 2015). Esta relação é multidimensional e impacta diversos Objetivos de Desenvolvimento Sustentável. Entre eles os de Saúde, Bem-Estar e Meio Ambiente (ODS 3 e 13). A bioinformática acelera a descoberta de fármacos e a medicina personalizada (ODS 3). Contudo, o processamento massivo de dados contribui para as emissões de Gases de Efeito Estufa (GEE).

A Computação Verde atua como mecanismo de controle, garantindo que o progresso na saúde não ocorra à custa da degradação climática (ODS 13). Algoritmos energeticamente eficientes reduzem a demanda sobre datacenters, mitigando a poluição indireta que afeta a saúde pública.

A infraestrutura digital moderna (ODS 9) depende da resiliência dos Centros de Processamento de Dados. A aplicação de IoT para monitoramento de Smart Grids e a otimização de sistemas de refrigeração são cruciais para o ODS 7 (Energia Limpa). A democratização do acesso ao HPC através da nuvem e de algoritmos mais leves promove a igualdade de oportunidades na pesquisa (ODS 4). Além disso, a transição para uma economia de baixo carbono fomenta a criação de "Empregos Verdes" no setor de tecnologia, desacoplando o crescimento econômico da degradação ambiental (ODS 8).

A transição para uma Bioinformática Verde exige que o pesquisador adote uma visão holística, onde o sucesso de um pipeline é medido pela binômio acurácia-sustentabilidade. Os dados apresentados corroboram que a mitigação do impacto ambiental deve priorizar a reengenharia das etapas iniciais e de montagem. A redução de apenas 10% no consumo da etapa de Montagem possui um impacto ambiental superior à eliminação completa das etapas de polimento e filtragem combinadas. Em última análise, a eficiência energética por megabyte processado deve ser estabelecida como uma métrica padrão na publicação de novos protocolos genômicos.

6 CONCLUSÃO E TRABALHOS FUTUROS

Na referida Dissertação estabeleceu-se uma base teórica para quantificar o consumo de energia elétrica em um ambiente computacional, como nos trabalhos de Yao (2014)], foi quantificado o consumo de energia elétrica em simulações computacionais em bioinformática, visando um consumo de energia alinhado com ODS, e buscando a transição energética, contribuindo para a redução dos GEE, colaborando no consumo de energia fontes renováveis, FUENTES-SEPÚLVEDA (2019).

Na análise dos dados de consumo de energia de simulações computacionais utilizando sequenciadores Oxford Nanopore revela uma clara vantagem da Unidade de Processamento Gráfico (GPU) em termos de eficiência energética e temporal em comparação com a Unidade Central de Processamento (CPU). Conforme observado, a simulação via CPU demandou um tempo significativamente maior (5,84 horas ou 21.024 segundos) e resultou em um consumo de energia mais elevado (114,79 kJ). Em contraste, a simulação executada na GPU foi drasticamente mais rápida (0,41 horas ou 1.476 segundos) e consumiu consideravelmente menos energia (0,54 W). Essa disparidade sublinha o potencial da GPU para acelerar processos computacionais intensivos, como a análise de dados de sequenciamento genômico, um campo central da bioinformática conforme Gauthier (2016) destaca em "A brief history of bioinformatics", que traça a evolução da disciplina e a crescente demanda por poder computacional.

Em resposta ao primeiro objetivo, "examinar os fatores que impactam o consumo energético na modelagem genômica", a pesquisa identificou os fatores determinantes do consumo de energia no sequenciamento genômico. O estudo concluiu que os principais elementos incluem a complexidade intrínseca dos algoritmos empregados, o volume massivo de dados processado, a arquitetura de hardware subjacente — com destaque para a distinção de eficiência entre CPUs e GPUs — e as configurações de software, como a paralelização e a otimização de código. A análise demonstrou que não há um único fator, mas sim uma interação complexa entre esses elementos que dita o perfil energético das simulações. Levando-se em consideração apenas a medição deste em um ambiente computacional específico, e também se considerar fatores como climatização e qualidade do fornecimento de energia.

A eficiência da GPU neste contexto pode ser atribuída à sua arquitetura paralela, otimizada para o processamento simultâneo de grandes volumes de dados. Este é um diferencial crucial para a bioinformática, onde a análise de sequências, montagem de

genomas e alinhamentos exigem processamento massivo, como descrito por Feltes (2007) em "Bioinformática: da biologia à flexibilidade molecular", que explora a complexidade e a necessidade de métodos computacionais robustos na área.

É importante ressaltar a significativa redução no tempo de simulação e no consumo de energia ao utilizar a GPU ressoa com a crescente preocupação com a pegada de carbono da computação, um tópico abordado por Grealey (2022) em "Green Algorithms: Quantifying the Carbon Footprint of Computation". A adoção de GPUs para tarefas bioinformáticas não só acelera a pesquisa, mas também contribui para uma computação mais sustentável, alinhando-se com os princípios de "algoritmos verdes".

Em relação ao segundo objetivo, "Analisar as metodologias utilizadas para medir esse consumo", foram analisadas as metodologias de quantificação do gasto energético. A investigação verificou que, embora ferramentas baseadas em hardware, como wattímetros, ofereçam medições de alta precisão, plataformas de software, a exemplo da Interface de Limite de Potência Média de Execução (RAPL), proporcionam maior granularidade e facilidade de implementação para análises em nível de componente. A escolha da metodologia, portanto, deve ser alinhada à profundidade e ao escopo da investigação desejada. No entanto há a necessidade de quantificação de fatores externos que podem afetar a simulação.

No que tange ao terceiro objetivo, "Investigar estratégias e técnicas adotadas para sua redução", o estudo investigou e sistematizou um conjunto de estratégias para a redução do consumo energético. As abordagens identificadas como mais promissoras incluem a otimização algorítmica para diminuir a complexidade computacional, a adoção de hardware especializado e de baixo consumo, e a implementação de técnicas de gerenciamento de energia,

Em suma, a transição para o uso de GPUs em simulações computacionais relacionadas a sequenciadores Oxford Nanopore não é apenas uma questão de velocidade, mas também de sustentabilidade. Esta otimização é vital para o avanço da "Era da Bioinformática", como Jawdat (2018) sugere em "The Era of Bioinformatics", onde a capacidade de processar dados genômicos de forma eficiente e ambientalmente consciente é cada vez mais determinante para o progresso científico e tecnológico.

Devido a eficiência demonstrada pelas GPUs na quantificação do consumo de energia em simulações de sequenciamento de Oxford Nanopore, e alinhado aos objetivos de sustentabilidade e redução de GEE, o próximo passo natural e crucial deste projeto será a realização de uma simulação completa de um organismo eucarionte no

Cluster do C3, aproveitando a capacidade de processamento paralelo e a otimização energética das GPUs para lidar com a complexidade e o volume massivo de dados inerentes a genomas eucariontes.

A relação entre Bioinformática, Internet das Coisas (IoT) e Computação Verde é fundamental para o cumprimento do Objetivo de Desenvolvimento Sustentável 3 (ODS 3) da ONU, que visa assegurar uma vida saudável e promover o bem-estar para todos. Enquanto a bioinformática acelera a descoberta de medicamentos e vacinas através de simulações genômicas e medicina personalizada, a IoT permite o monitoramento instantâneo do consumo de energia e da qualidade ambiental, facilitando gerenciamento energético e a redução da emissão de GEE.

A disseminação e a integração entre tecnologias avançadas e o ODS 4 (Educação de Qualidade) destaca-se pela capacidade de democratizar o acesso à pesquisa científica e otimizar a gestão escolar. O uso de Bioinformática e Computação de Alto Desempenho (HPC) na nuvem permite que estudantes de instituições menos favorecidas realizem pesquisas de ponta, promovendo a igualdade de oportunidades. Além disso, plataformas digitais e o ensino a distância facilitam a aprendizagem contínua em áreas estratégicas, enquanto a inclusão de temas como Computação Verde e IoT nos currículos escolares capacita os alunos com competências essenciais para a sustentabilidade energética e uma consciência de economia circular.

Podemos fazer uma relação entre tecnologia e o ODS 7 (Energia Acessível e Limpa) é caracterizada por um papel dual: ao mesmo tempo que o processamento de alto desempenho (HPC) em Bioinformática e as redes de IoT elevam a demanda energética, essas mesmas ferramentas são fundamentais para a transição energética.

O alto consumo de energia em Centros de Processamento de Dados (CPDs) pode ameaçar a segurança e a acessibilidade energética se depender de fontes caras e não renováveis. Para mitigar esse impacto, a Computação Verde utiliza servidores eficientes, sistemas de refrigeração otimizados e Algoritmos Verdes para reduzir a demanda e aliviar a pressão sobre a rede elétrica pública. Além da mitigação, a tecnologia atua como um otimizador direto da matriz energética sustentável:

- IoT e Smart Grids: A Internet das Coisas é essencial para criar redes inteligentes que equilibram cargas e gerenciam fontes renováveis em tempo real.
- Bioinformática e Biocombustíveis: A análise genômica acelera a pesquisa de organismos otimizados para a produção de fontes de energia limpa.

- **Eficiência Operacional:** A Computação Verde transforma a infraestrutura digital de uma grande consumidora em um pilar de eficiência energética fundamental para o acesso universal à energia sustentável.

No tocante a intersecção entre tecnologia e o ODS 8 (Trabalho Decente e Crescimento Econômico) revela que a Bioinformática e a IoT são motores fundamentais para a inovação e o aumento da produtividade em setores de alto valor, como a biotecnologia e as fábricas inteligentes. Essa expansão tecnológica fomenta a criação de postos de trabalho altamente qualificados e de "Empregos Verdes" voltados à sustentabilidade, fortalecendo a competitividade econômica e promovendo o trabalho decente a toda a população do planeta.

Entretanto, o alto consumo energético exigido por essas áreas representa um desafio à sustentabilidade do crescimento econômico. Nesse contexto, a Computação Verde — através de Algoritmos Verdes e hardware eficiente — surge como uma solução estratégica para otimizar recursos e reduzir custos operacionais. Essa abordagem permite desacoplar o progresso econômico da degradação ambiental, garantindo que o desenvolvimento tecnológico na bioinformática seja duradouro e ecologicamente responsável.

Visando a integração entre Computação Verde, Bioinformática e IoT é o pilar da infraestrutura digital moderna e motor essencial para o ODS 9 (Indústria, Inovação e Infraestrutura). A IoT fortalece a resiliência de infraestruturas físicas e digitais ao monitorar o desempenho em tempo real e prever falhas. Paralelamente, a Computação Verde assegura que os Centros de Processamento de Dados (CPDs) — espinha dorsal dessa infraestrutura — operem com alta confiabilidade e eficiência energética, utilizando Algoritmos Verdes e refrigeração otimizada para reduzir riscos de sobrecarga e aumentar o tempo de atividade.

No campo da inovação, a Bioinformática e o uso de Computação de Alto Desempenho (HPC) impulsionam avanços científicos em saúde, agricultura e novos materiais. A redução de custos energéticos promovida pela TI Sustentável torna essas tecnologias mais inclusivas, permitindo que diversas instituições participem de atividades de Pesquisa e Desenvolvimento (P&D). Além disso, a aplicação de sensores IoT na Indústria 4.0 permite a maximização da eficiência de recursos e o desenvolvimento de processos industriais mais limpos, como a biorremediação, garantindo que a

industrialização ocorra de maneira ecologicamente responsável e com menor pegada de carbono.

Em suma, a convergência entre Bioinformática, IoT e Computação Verde consolida-se como a base da infraestrutura digital resiliente e um motor indispensável para o ODS 9. Enquanto a IoT e a Bioinformática impulsionam a inovação científica e a eficiência na Indústria 4.0, a Computação Verde atua como o balizador ético e operacional, garantindo que o processamento de alto desempenho (HPC) seja energeticamente sustentável.

A implementação de Algoritmos Verdes e o monitoramento inteligente de dados não apenas reduzem a pegada de carbono e os custos operacionais, mas também democratizam o acesso à pesquisa avançada. Dessa forma, a integração dessas tecnologias permite que a industrialização e a inovação tecnológica avancem de maneira inclusiva e ecologicamente responsável, alinhando o progresso técnico à preservação ambiental.

Com a integração entre Computação Verde, Bioinformática e IoT desempenha um papel duplo e essencial no suporte aos ODS 13 (Ação Contra a Mudança Global do Clima) e ODS 15 (Vida Terrestre). Embora o alto consumo energético dos Centros de Processamento de Dados (CPD) para bioinformática e IoT seja uma fonte significativa de emissões de Gases de Efeito Estufa (GEE), a Computação Verde surge como uma medida de mitigação urgente. Através do uso de Algoritmos Verdes, hardware eficiente e monitoramento preciso, é possível reduzir a pegada de carbono da infraestrutura digital, transformando o impacto ambiental da tecnologia em uma operação sustentável. Além da mitigação, essas tecnologias servem como ferramentas críticas para a preservação ambiental:

- **Monitoramento e Prevenção:** A IoT é a principal ferramenta de monitoramento climático e terrestre em tempo real, fornecendo dados cruciais sobre florestas, desmatamento e saúde do solo, além de alimentar sistemas de alerta precoce contra eventos extremos.
- **Resiliência e Conservação:** A Bioinformática utiliza a computação de alto desempenho (HPC) para desenvolver culturas agrícolas resistentes ao clima e analisar genomas vitais para a conservação de espécies e restauração de ecossistemas.
- **Sustentabilidade Integrada:** A combinação de IoT e Bioinformática viabiliza a agricultura de precisão, combatendo a degradação da terra e a desertificação,

enquanto a Computação Verde resolve o paradoxo do alto consumo energético necessário para essas ações de preservação.

Por fim, “compreender como diferentes domínios de aplicação de simulações computacionais impactam o consumo de energia elétrica”, ao analisar a influência de diferentes domínios de aplicação, a pesquisa permitiu compreender que, embora áreas como engenharia e física compartilhem desafios de simulação intensiva, a modelagem genômica possui um perfil de consumo particular, frequentemente caracterizado por operações intensivas de entrada/saída (I/O) e manipulação de memória. Essa constatação reforça a necessidade de desenvolver estratégias de otimização energética específicas para as cargas de trabalho da bioinformática.

Dessa forma, a pesquisa cumpre sua proposta ao fornecer um diagnóstico claro dos desafios e das soluções para a eficiência energética no sequenciamento genômico. As conclusões apresentadas contribuem para o entendimento para a necessidade do avanço na utilização da computação verde, oferecendo um guia para que pesquisadores, desenvolvedores e gestores de infraestrutura de TI possam tomar decisões mais conscientes, minimizando o impacto ambiental da pesquisa científica sem comprometer a qualidade e a velocidade das descobertas no campo da genômica. O próximo passo do presente projeto será quantificar o consumo de energia do sequenciamento completo de organismos eucarionte e procariontes de organismos mais complexos, com maior complexidade computacional e identificando demais fatores (temperatura ambiente, configuração do sistema, etc..) , no Cluster do C3 da FURG, quantificado o consumo energético do mesmo durante o doutorado.

REFERÊNCIAS

- [1] J. Gauthier, A. T. Vincent, S. J. Charette, e N. Derome, “A brief history of bioinformatics”, *Briefings in Bioinformatics*, vol. 20, no 6, p. 1981–1996, nov. 2019, doi: [10.1093/bib/bby063](https://doi.org/10.1093/bib/bby063).
- [2] D. C. B. Mariano, C. Leite, L. H. S. Santos, e R. C. de Melo-Minardi, “A guide to performing systematic literature reviews in bioinformatics”.
- [3] *ABNT NBR 6023: informação e documentação: referências: elaboração*. ABNT, 2018.
- [4] N. S. Silva, J. G. Costa, e L. Antônio, “Análise de Consumo de Energia Utilizando Arduino”, 2019.
- [5] STALLINGS, William., *Arquitetura e organização de computadores*. 2017.
- [6] et al Bruno César Feltes, *Bioinformática: da biologia à flexibilidade molecular*. SBBq, 2015.
- [7] J. Breda, J. Martins Medeiros, e J. Vitor Gomes De Araújo, “Desenvolvimento de um Medidor de Consumo de Energia Elétrica em Tempo Real por meio de Medidor não Invasivo de Corrente Utilizando a Plataforma Arduino”, em *Anais do 14o Simpósio Brasileiro de Automação Inteligente*, Galoa, 2019. doi: [10.17648/sbai-2019-111428](https://doi.org/10.17648/sbai-2019-111428).
- [8] G. Shankar, S. Ray, S. R. Goyal, e R. Srinivas, “Development of a Low Cost Phasor Measurement Unit for Laboratory Studies”, em *2021 1st International Conference on Power Electronics and Energy (ICPEE)*, Bhubaneswar, India: IEEE, jan. 2021, p. 1–6. doi: [10.1109/ICPEE50452.2021.9358673](https://doi.org/10.1109/ICPEE50452.2021.9358673).
- [9] R. W. Fransiska, E. M. P. Septia, W. K. Vessabhu, W. Frans, W. Abednego, e Hendro, “Electrical power measurement using Arduino Uno microcontroller and LabVIEW”, em *2013 3rd International Conference on Instrumentation, Communications, Information Technology and Biomedical Engineering (ICICI-BME)*, Bandung: IEEE, nov. 2013, p. 226–229. doi: [10.1109/ICICI-BME.2013.6698497](https://doi.org/10.1109/ICICI-BME.2013.6698497).
- [10] M. Errbii, U. R. Ernst, A. Lajmi, E. Privman, J. Gadau, e L. Schrader, “Evolutionary genomics of socially polymorphic populations of *Pogonomyrmex californicus*”, *BMC Biol*, vol. 22, no 1, p. 109, maio 2024, doi: [10.1186/s12915-024-01907-z](https://doi.org/10.1186/s12915-024-01907-z).
- [11] S. K. Addya, A. Satpathy, B. C. Ghosh, S. Chakraborty, S. K. Ghosh, e S. K. Das, “Geo-Distributed Multi-Tier Workload Migration Over Multi-Timescale Electricity Markets”, *IEEE Trans. Serv. Comput.*, vol. 16, no 5, p. 3385–3396, set. 2023, doi: [10.1109/TSC.2023.3270921](https://doi.org/10.1109/TSC.2023.3270921).

- [12] D. Raimi, Y. Zhu, R. G. Newell, B. C. Prest, e A. Bergman, “Global Energy Outlook 2023: Sowing the Seeds of an Energy Transition”.
- [13] A. Di Nisio, T. Di Noia, C. G. C. Carducci, e M. Spadavecchia, “High Dynamic Range Power Consumption Measurement in Microcontroller-Based Applications”, *IEEE Trans. Instrum. Meas.*, vol. 65, no 9, p. 1968–1976, set. 2016, doi: [10.1109/TIM.2016.2549818](https://doi.org/10.1109/TIM.2016.2549818).
- [14] K. Luechaphonthara e V. A., “IoT based application for monitoring electricity power consumption in home appliances”, *IJECE*, vol. 9, no 6, p. 4988, dez. 2019, doi: [10.11591/ijece.v9i6.pp4988-4992](https://doi.org/10.11591/ijece.v9i6.pp4988-4992).
- [15] Y. Chernyavskaya, X. Zhang, J. Liu, e J. Blackburn, “Long-read sequencing of the zebrafish genome reorganizes genomic architecture”, *BMC Genomics*, vol. 23, no 1, p. 116, dez. 2022, doi: [10.1186/s12864-022-08349-3](https://doi.org/10.1186/s12864-022-08349-3).
- [16] H. Jameel e H. K. Farhan, “Low-Cost Energy-Efficient Smart Monitoring System Using Open- Source Microcontrollers”, *IREACO*, vol. 9, no 6, p. 423, nov. 2016, doi: [10.15866/ireaco.v9i6.10315](https://doi.org/10.15866/ireaco.v9i6.10315).
- [17] R. S. Wazlawick, “Metodologia de Pesquisa para Ciência da Computação”. 2009.
- [18] M. C. Pelegrin e P. J. Martins, “Monitoramento de Consumo de Energia, Utilizando Arduino e Sensor de Tensão e Corrente Elétrica”.
- [19] P. P. Machado, T. P. Abud, M. Z. Fortes, e B. S. M. C. Borba, “Power factor metering system using Arduino”, em *2017 IEEE Workshop on Power Electronics and Power Quality Applications (PEPQA)*, Bogota: IEEE, maio 2017, p. 1–6. doi: [10.1109/PEPQA.2017.7981633](https://doi.org/10.1109/PEPQA.2017.7981633).
- [20] J. Yao, X. Liu, e C. Zhang, “Predictive Electricity Cost Minimization Through Energy Buffering in Data Centers”, *IEEE Trans. Smart Grid*, vol. 5, no 1, p. 230–238, jan. 2014, doi: [10.1109/TSG.2013.2274525](https://doi.org/10.1109/TSG.2013.2274525).
- [21] M. Wang, S. Ren, Y. Du, Y. Li, e X. Ma, “Research on electric energy information acquisition system based on embedded platform”, *Multimed Tools Appl*, vol. 83, no 10, p. 29377–29397, set. 2023, doi: [10.1007/s11042-023-16716-9](https://doi.org/10.1007/s11042-023-16716-9).
- [22] G. Dodig-Crnkovic, “Scientific Methods in Computer Science”.
- [23] H. E. Imai, L. F. Bósio, A. A. D. Silva Junior, L. K. Berto, N. U. Yamaguchi, e L. C. S. H. Rezende, “Simulação computacional como ferramenta de otimização na geração de energia solar fotovoltaica”, *urbe, Rev. Bras. Gest. Urbana*, vol. 12, p. e20190343, 2020, doi: [10.1590/2175-3369.012.e20190343](https://doi.org/10.1590/2175-3369.012.e20190343).
- [24] J. Grealey *et al.*, “The Carbon Footprint of Bioinformatics”, *Molecular Biology and Evolution*, vol. 39, no 3, p. msac034, mar. 2022, doi: [10.1093/molbev/msac034](https://doi.org/10.1093/molbev/msac034).

- [25] D. Jawdat, “The Era of Bioinformatics”, em *2006 2nd International Conference on Information & Communication Technologies*, Damascus, Syria: IEEE, 2006, p. 1860–1865. doi: [10.1109/ICTTA.2006.1684672](https://doi.org/10.1109/ICTTA.2006.1684672).
- [26] Y. Berezovskaya, C.-W. Yang, e V. Vyatkin, “Towards Multi-Agent Control in Energy-Efficient Data Centres”, em *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, Singapore, Singapore: IEEE, out. 2020, p. 3574–3579. doi: [10.1109/IECON43393.2020.9255232](https://doi.org/10.1109/IECON43393.2020.9255232).
- [27] ONU, “Transformando Nosso Mundo: A Agenda 2030 para o Desenvolvimento Sustentável”, *Ambientalmente sustentável*, vol. 25, no 1, p. 171–190, jan. 2018, doi: [10.17979/ams.2018.25.1.4655](https://doi.org/10.17979/ams.2018.25.1.4655).
- [28] [Algoritmos Verdes | Algoritmos verdes](#) – www.green-algorithms.org – Acesso 17/04/2025
- [29] FUENTES-SEPÚLVEDA, José; LADRA, Susana. Consumo de energia em vetores inteiros compactos: um estudo de caso. Acesso IEEE, v. 7, p. 155625-155636, 2019.
- [30] L. Lannelongue, J. Grealey, e M. Inouye, “Green Algorithms: Quantifying the Carbon Footprint of Computation”, *Advanced Science*, vol. 8, no 12, jun. 2021, doi: [10.1002/advs.202100707](https://doi.org/10.1002/advs.202100707).
- [31] De Araújo Macêdo et al. - 2025 - A Brazilian Perspective on Computing for the Plane - <https://doi.org/10.1145/3724498>
- [32] Sociedade Brasileira de Computação, “Tecnologias Digitais para o Meio Ambiente: Manifesto SBC”, Sociedade Brasileira de Computação, jul. 2022. doi: [10.5753/sbc.rt.2022.07.01](https://doi.org/10.5753/sbc.rt.2022.07.01).
- [33] Satam, H., Joshi, K., Mangrolia, U., Waghoo, S., Zaidi, G., Rawool, S., Thakare, R. P., Banday, S., Mishra, A. K., Das, G., and Malonia, S. K. (2023). Next-Generation Sequencing Technology: Current Trends and Advancements. *Biology*, 12(7):997.
- [34] WANG, Yunhao et al. Nanopore sequencing technology, bioinformatics and applications. **Nature biotechnology**, v. 39, n. 11, p. 1348-1365, 2021.
- [35] WICK, Ryan R.; JUDD, Louise M.; HOLT, Kathryn E. Performance of neural network basecalling tools for Oxford Nanopore sequencing. *Genome biology*, v. 20, n. 1, p. 129, 2019.
- [36] N. D. O. Volpini, V. S. Conceição, R. L. Pontes, e D. Guedes, “Uma análise do consumo de energia de ambientes de processamento de dados massivos em nuvem”, em *Anais do Workshop em Desempenho de Sistemas Computacionais e de*

Comunicação (WPerformance), Sociedade Brasileira de Computação - SBC, jul. 2018. doi: 10.5753/wperformance.2018.3338.

[37] R. Flora Calili, "POLÍTICAS DE EFICIÊNCIA ENERGÉTICA NO BRASIL: UMA ABORDAGEM EM UM AMBIENTE ECONÔMICO SOB INCERTEZA", DOUTOR EM CIÊNCIAS EM ENGENHARIA ELÉTRICA, PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO, Rio de Janeiro, Brazil, 2013. doi: 10.17771/PUCRio.acad.23849

[38] A. dos R. Araújo, Controle estatístico da qualidade. Bookman, 2012.

[39] Wick, R. "ONT-only accuracy: 5 kHz and Dorado. Ryan Wick's bioinformatics blog; 2023." 2025

[40] SCHACKSEN, Patrick Skov et al. Complete pipeline for Oxford Nanopore Technology amplicon sequencing (ONT-AmpSeq): from pre-processing to creating an operational taxonomic unit table. FEBS Open Bio, v. 14, n. 11, p. 1779-1787, 2024.

ANEXO 1

Desenvolveu-se um algoritmo para executar o monitoramento utilizando uma placa de ARDUINO UNO e um sensor PZEMT-004T

➤ Algoritmo "Medição de Consumo de Energia", em Portugol

```
// Inclui a biblioteca SoftwareSerial para comunicação serial com o PZEM-004T
#include <SoftwareSerial.h>

// Define os pinos para comunicação serial com o PZEM-004T
const int rxPin = 2; // Pino RX do Arduino conectado ao TX do PZEM-004T
const int txPin = 3; // Pino TX do Arduino conectado ao RX do PZEM-004T

// Cria um objeto SoftwareSerial
SoftwareSerial pzemSerial(rxPin, txPin);

// Endereço do dispositivo PZEM-004T (o padrão é 0x01)
byte pzemAddress = 0x01;

// Variáveis para armazenar os dados lidos do PZEM-004T
float tensaoRedeLida = 0.0;
float correnteLida = 0.0;
float potenciaLida = 0.0;
float energiaLida = 0.0;
float frequenciaLida = 0.0;
float fatorPotenciaLido = 0.0;

void setup() {
    // Inicializa a comunicação serial com o computador
    Serial.begin(9600);
    Serial.println("Iniciando leitura do PZEM-004T...");

    // Inicializa a comunicação serial com o PZEM-004T
    pzemSerial.begin(9600);

    // Aguarda um breve período para inicialização
    delay(100);
}
```

```

void loop() {
  // Solicita os dados do PZEM-004T
  if (requestPZEMData()) {
    // Se os dados foram lidos com sucesso, exibe-os
    Serial.print("Tensão: ");
    Serial.print(tensaoRedeLida);
    Serial.println(" V");

    Serial.print("Corrente: ");
    Serial.print(correnteLida);
    Serial.println(" A");

    Serial.print("Potência: ");
    Serial.print(potenciaLida);
    Serial.println(" W");

    Serial.print("Energia: ");
    Serial.print(energiaLida);
    Serial.println(" kWh");

    Serial.print("Frequência: ");
    Serial.print(frequenciaLida);
    Serial.println(" Hz");

    Serial.print("Fator de Potência: ");
    Serial.println(fatorPotenciaLido);
  } else {
    // Se houve falha na leitura, exibe uma mensagem de erro
    Serial.println("Erro ao ler dados do PZEM-004T.");
  }

  // Espera um intervalo antes da próxima leitura
  delay(1000); // Lê os dados a cada 1 segundo
}

// Função para solicitar os dados do PZEM-004T
bool requestPZEMData() {
  byte requestFrame[] = {pzemAddress, 0x04, 0x00, 0x00, 0x00, 0x0A, 0x21, 0xCB};
  byte responseFrame[25];

```

```

unsigned int timeout = 1000; // Tempo limite para a resposta (ms)
unsigned long startTime = millis();
int bytesReceived = 0;

// Envia a solicitação
pzemSerial.write(requestFrame, sizeof(requestFrame));

// Aguarda a resposta com timeout
while (millis() - startTime < timeout && bytesReceived < 25) {
    if (pzemSerial.available()) {
        responseFrame[bytesReceived++] = pzemSerial.read();
    }
}

// Verifica se a resposta tem o tamanho esperado e o endereço correto
if (bytesReceived == 25 && responseFrame[0] == pzemAddress) {
    // Extrai os dados da resposta (conforme o protocolo do PZEM-004T)
    tensaoRedeLida = (responseFrame[3] * 256.0 + responseFrame[4]) / 10.0;
    correnteLida = (responseFrame[5] * 256.0 + responseFrame[6] + (responseFrame[7] * 256.0 +
responseFrame[8]) / 1000.0) / 1000.0;
    potenciaLida = (responseFrame[9] * 256.0 + responseFrame[10]) / 10.0;
    energiaLida = (responseFrame[11] * 256.0 * 256.0 + responseFrame[12] * 256.0 +
responseFrame[13]) / 1000.0;
    frequenciaLida = (responseFrame[15] * 256.0 + responseFrame[16]) / 10.0;
    fatorPotenciaLido = (responseFrame[17] * 256.0 + responseFrame[18]) / 100.0;
    return true;
} else {
    return false;
}
}

```

ANEXO 2

Programa de Controle do Arduino Uno:

```

void setup() {
    // put your setup code here, to run once:

}

void loop() {
    // put your main code here, to run repeatedly:

}
#include <PZEM004Tv30.h>
#include <Wire.h>

PZEM004Tv30 pzem(8, 9); // Software Serial pin 8 (RX) & 9 (TX)

void setup() {
    Serial.begin(9600);
}

void loop() {
    float voltage = pzem.voltage();
    if (voltage != NAN) {
        Serial.print("Voltage: ");
        Serial.print(voltage);
        Serial.println("V");
    } else {
        Serial.println("Error reading voltage");
    }

    float current = pzem.current();
    if (current != NAN) {

```

```
Serial.print("Current: ");
Serial.print(current);
Serial.println("A");
} else {
    Serial.println("Error reading current");
}

float power = pzem.power();
if (current != NAN) {
    Serial.print("Power: ");
    Serial.print(power);
    Serial.println("W");
} else {
    Serial.println("Error reading power");
}

float energy = pzem.energy();
if (current != NAN) {
    Serial.print("Energy: ");
    Serial.print(energy, 3);
    Serial.println("kWh");
} else {
    Serial.println("Error reading energy");
}

float frequency = pzem.frequency();
if (current != NAN) {
    Serial.print("Frequency: ");
    Serial.print(frequency, 1);
    Serial.println("Hz");
} else {
    Serial.println("Error reading frequency");
}
```

```

float pf = pzem.pf();
if (current != NAN) {
    Serial.print("PF: ");
    Serial.println(pf);
} else {
    Serial.println("Error reading power factor");
}

Serial.println();
delay(2000);
}

```

A utilização da placa de Arduino Uno integrado ao sensor de monitoramento de energia PZEM-004T, segue os seguintes passos:

9. Inclusão de Biblioteca:

- ✓ Conceito: A comunicação com o sensor PZEM-004T ocorre através da interface serial TTL (níveis lógicos de tensão). O Arduino Uno possui uma porta serial principal que geralmente é utilizada para a comunicação com o computador (via USB).
- ✓ Implementação: Para utilizar a comunicação serial com o sensor em pinos digitais do Arduino, emprega-se a biblioteca SoftwareSerial. Esta biblioteca permite emular uma porta serial em qualquer par de pinos digitais do Arduino.
- ✓ Código: `#include <SoftwareSerial.h>` - Esta linha inclui a biblioteca SoftwareSerial no projeto, tornando suas funcionalidades disponíveis.

10. Definição de Pinos Seriais:

- ✓ Conceito: É necessário definir quais pinos digitais do Arduino serão utilizados para a comunicação serial com o PZEM-004T. Os pinos de transmissão (TX) de um dispositivo devem ser conectados aos pinos de recepção (RX) do outro, e vice-versa.
- ✓ Implementação:

- `const int rxPin = 2;` - Define o pino digital 2 do Arduino como o pino de recepção (RX) dos dados enviados pelo PZEM-004T. Este pino será conectado ao pino TX do PZEM-004T.
- `const int txPin = 3;` - Define o pino digital 3 do Arduino como o pino de transmissão (TX) dos dados para o PZEM-004T. Este pino será conectado ao pino RX do PZEM-004T.
- ✓ Importância: A correta conexão dos pinos RX e TX é crucial para o estabelecimento da comunicação serial. Uma conexão invertida impedirá a troca de dados entre o Arduino e o sensor.

11. Criação do Objeto SoftwareSerial:

- ✓ Conceito: Para utilizar as funcionalidades da biblioteca SoftwareSerial, é necessário criar um objeto dessa classe. Este objeto representará a porta serial virtual que será utilizada para se comunicar com o PZEM-004T.
- ✓ Implementação: `SoftwareSerial pzemSerial(rxPin, txPin);` - Esta linha cria um objeto chamado `pzemSerial`, associando-o aos pinos digitais 2 (RX) e 3 (TX) definidos anteriormente. Todas as operações de comunicação com o PZEM-004T serão realizadas através deste objeto.

12. Endereço do Dispositivo:

- ✓ Conceito: O PZEM-004T se comunica utilizando um protocolo que permite que múltiplos dispositivos sejam conectados a um mesmo barramento serial. Cada dispositivo possui um endereço único para identificação.
- ✓ Implementação: `byte pzemAddress = 0x01;` - Esta linha declara uma variável do tipo `byte` chamada `pzemAddress` e a inicializa com o valor `0x01`. Este é o endereço padrão de fábrica da maioria dos módulos PZEM-004T.
- ✓ Consideração para Múltiplos Módulos: Caso haja a necessidade de conectar vários sensores PZEM-004T a um único Arduino, o endereço de cada módulo precisará ser alterado (através de comandos específicos enviados ao módulo) para evitar conflitos na comunicação.

13. Variáveis para Dados:

- ✓ Conceito: Para armazenar os diferentes parâmetros de energia lidos do PZEM-004T, é necessário declarar variáveis com o tipo de dado apropriado para representar valores numéricos com casas decimais.
- ✓ Implementação: São declaradas variáveis do tipo float (ponto flutuante) para cada parâmetro:
 - `tensaoRedeLida` (Tensão em Volts)
 - `correnteLida` (Corrente em Amperes)
 - `potenciaLida` (Potência em Watts)
 - `energiaLida` (Energia consumida em kWh)
 - `frequenciaLida` (Frequência da rede em Hz)
 - `fatorPotenciaLido` (Fator de potência)

14. Função `setup()`:

- ✓ Conceito: A função `setup()` é executada apenas uma vez quando o Arduino é ligado ou resetado. Ela é utilizada para realizar configurações iniciais.
- ✓ Implementação:
 - `Serial.begin(9600);` - Inicializa a comunicação serial principal (via USB) com o computador a uma taxa de transmissão de 9600 bits por segundo (baud). Isso permite que o Arduino envie informações para o monitor serial do IDE.
 - `Serial.println("Iniciando leitura do PZEM-004T...");` - Exibe uma mensagem no monitor serial indicando o início do processo de leitura do sensor.
 - `pzemSerial.begin(9600);` - Inicializa a comunicação serial com o sensor PZEM-004T através da porta serial virtual (`pzemSerial`) na mesma taxa de baud de 9600. Esta é a taxa de comunicação padrão do PZEM-004T.
 - `delay(100);` - Introduz uma pequena pausa de 100 milissegundos para garantir que os dispositivos tenham tempo suficiente para inicializar corretamente antes de iniciar a comunicação.

15. Função `loop()`:

- ✓ Conceito: A função `loop()` é executada continuamente após a função `setup()` ter sido concluída. Ela contém o código principal que será repetido indefinidamente.
- ✓ Implementação:

- `if (requestPZEMData()) { ... } else { ... }` - Chama a função `requestPZEMData()` para solicitar os dados do sensor. A estrutura condicional `if-else` verifica o valor de retorno desta função:
- Se `requestPZEMData()` retornar `true`, significa que os dados foram lidos com sucesso, e o bloco de código dentro do `if` é executado para exibir os valores lidos no monitor serial.
- Se `requestPZEMData()` retornar `false`, indica que ocorreu um erro durante a leitura, e o bloco de código dentro do `else` é executado para exibir uma mensagem de erro.
- `Serial.print(...)` e `Serial.println(...)` - Dentro do bloco `if`, estas funções são utilizadas para enviar os valores de tensão, corrente, potência, energia, frequência e fator de potência para o monitor serial do Arduino IDE, permitindo que o usuário visualize os dados lidos.
- `delay(1000);` - Introduz uma pausa de 1000 milissegundos (1 segundo) antes da próxima iteração do loop, controlando a frequência com que os dados são lidos do sensor.

16. Função `requestPZEMData()`:

- ✓ Conceito: Esta função é responsável por enviar a solicitação de dados ao sensor PZEM-004T e receber a resposta contendo as informações de energia.
- ✓ Implementação:
 - Frame de Requisição:
 - ❖ `byte requestFrame[] = {pzemAddress, 0x04, 0x00, 0x00, 0x00, 0x0A, 0x21, 0xCB};` - Define um array de bytes chamado `requestFrame`. Este array contém o comando específico para solicitar os dados de leitura do PZEM-004T (função de leitura com código 0x04).
 - ❖ Os bytes individuais têm o seguinte significado:
 - ❖ `pzemAddress`: Endereço do dispositivo PZEM-004T (definido como 0x01).
 - ❖ 0x04: Código da função de leitura (Read Input Registers).
 - ❖ 0x00, 0x00: Endereço inicial do registro a ser lido (neste caso, o primeiro registro de dados).
 - ❖ 0x00, 0x0A: Número de registros a serem lidos (10 registros contendo todos os parâmetros de energia).

- ❖ 0x21, 0xCB: CRC (Cyclic Redundancy Check) de 16 bits para garantir a integridade do frame de requisição. O CRC é calculado com base nos bytes anteriores do frame.
- Buffer de Resposta:
 - ❖ `byte responseFrame[25];` - Declara um array de bytes chamado `responseFrame` com tamanho 25. Este array será utilizado para armazenar os bytes de resposta enviados pelo PZEM-004T. O tamanho da resposta para esta solicitação é de 25 bytes, conforme a documentação do sensor.
- Timeout:
 - ❖ `unsigned int timeout = 1000;` - Define um tempo limite de 1000 milissegundos (1 segundo) para aguardar a resposta do sensor.
 - ❖ `unsigned long startTime = millis();` - Armazena o tempo em que a solicitação foi enviada.
 - ❖ `int bytesReceived = 0;` - Inicializa um contador para o número de bytes recebidos.
 - ❖ O loop `while (millis() - startTime < timeout && bytesReceived < 25)` aguarda a chegada dos bytes de resposta até que o tempo limite seja atingido ou todos os 25 bytes esperados sejam recebidos.
 - ❖ `if (pzemSerial.available()) { responseFrame[bytesReceived++] = pzemSerial.read(); }` - Dentro do loop, verifica se há dados disponíveis na porta serial virtual (`pzemSerial`). Se houver, um byte é lido e armazenado no `responseFrame`, e o contador `bytesReceived` é incrementado.
- Envio da Requisição:
 - ❖ `pzemSerial.write(requestFrame, sizeof(requestFrame));` - Envia o array `requestFrame` (contendo o comando de leitura) para o PZEM-004T através da porta serial virtual.
- Verificação da Resposta:

- ❖ if (bytesReceived == 25 && responseFrame[0] == pzemAddress) - Após o período de espera, esta condição verifica se:
- ❖ O número de bytes recebidos (bytesReceived) é igual ao tamanho esperado da resposta (25).
- ❖ O primeiro byte da resposta (responseFrame[0]) corresponde ao endereço do dispositivo solicitado (pzemAddress), garantindo que a resposta veio do sensor correto.

- Extração dos Dados:

Se a resposta for válida, os bytes relevantes do responseFrame são extraídos e combinados para formar os valores dos parâmetros de energia. As fórmulas de conversão (multiplicação e divisão por diferentes potências de 10) são específicas do protocolo de comunicação do PZEM-004T e estão detalhadas em sua documentação técnica.

ANEXO 3

➤ Configuração do processador Intel® Core™ i7-13700K da 13ª Geração.

É um processador de alto desempenho com uma arquitetura híbrida, que combina dois tipos de núcleos para otimizar tanto a performance em tarefas pesadas quanto a eficiência em tarefas leves.

Aqui estão os detalhes principais da configuração:

- Arquitetura e Núcleos - Total de Núcleos: 16
- Total de Threads: 24
- Arquitetura Híbrida: 8 Performance-cores (P-cores): Focados em alto desempenho para jogos e aplicações exigentes. 8 Efficient-cores (E-cores): Focados em eficiência energética e tarefas em segundo plano.
- Frequências (Velocidade do Clock): Frequência Turbo Máxima: Até 5.4 GHz. Frequência da Tecnologia Intel® Turbo Boost Max 3.0: Até 5.4 GHz. Frequência Turbo Máx. do P-core: Até 5.3 GHz. Frequência Turbo Máx. do E-core: Até 4.2 GHz. Frequência Base do P-core: 3.4 GHz. Frequência Base do E-core: 2.5 GHz
- Memória Cache: Intel® Smart Cache (Cache L3): 30 MB. Cache L2 Total: 24 MB
- Memória RAM Suportada. Tipos de Memória: DDR5 (até 5600 MT/s) e DDR4 (até 3200 MT/s). Máximo de Canais de Memória: 2. Tamanho Máximo de Memória: 192 GB
- Gráficos Integrados. Modelo: Gráficos UHD Intel® 770. Frequência Dinâmica Máxima: 1.60 GHz
- Outras Especificações: Litografia: Intel 7 (processo de fabricação de 10 nm). Soquete: FCLGA1700. Potência Base do Processador: 125 W. Potência Turbo Máxima: 253 W
- Desbloqueado para Overclock: Sim (indicado pela letra "K" no nome)

ANEXO 4

➤ Configuração da GPU NVIDIA GeForce GTX 1050 Ti:

Hardware: Placa de Vídeo - NVIDIA GeForce GTX 1050 Ti: A GTX 1050 Ti é uma placa de vídeo de entrada/intermediária baseada na arquitetura Pascal da NVIDIA..

As especificações técnicas principais são:

- Arquitetura: Pascal (Chip GP107);
- Núcleos CUDA: 768 (são os processadores paralelos que executam os cálculos);
- Memória de Vídeo (VRAM): 4 GB GDDR5. Interface da Memória: 128-bit;
- Clock do Processador Gráfico (Base): Aproximadamente 1290 MHz;
- Clock do Processador Gráfico (Boost): Aproximadamente 1392 MHz (pode variar um pouco dependendo do fabricante da placa, como ASUS, Gigabyte, MSI, etc.);
- Consumo de Energia (TDP): 75 Watts (não exige conector de energia extra da fonte na maioria dos modelos);
- Software: Driver e API: Estes são os componentes que permitem que o sistema operacional e os programas utilizem o hardware da placa de vídeo de forma eficiente. NVIDIA-SMI / Driver Version: 535.247.01. NVIDIA-SMI (System Management Interface): É uma ferramenta de linha de comando que permite monitorar e gerenciar as placas de vídeo NVIDIA. É com ela que você obteve essas informações.
- Driver Version 535.247.01: Este é o software que "traduz" as instruções do sistema operacional (Windows ou Linux) para a placa de vídeo. Um driver atualizado garante compatibilidade com programas mais recentes, correções de bugs e otimizações de desempenho.
- CUDA Version: 12.2: CUDA (Compute Unified Device Architecture): É a plataforma de computação paralela e modelo de programação da NVIDIA. Essencialmente, é uma API (Interface de Programação de Aplicações) que permite que desenvolvedores (e programas como os de simulação em bioinformática) usem o poder de processamento dos milhares de núcleos da GPU para tarefas de cálculo intensivo, não apenas para gráficos.

A diferença de arquitetura entre CPU e GPU sob a ótica da obra de Stallings, nos traz que com a evolução da computação contemporânea ocorreu a consolidação da coexistência de duas arquiteturas de processamento distintas e complementares, a Unidade Central de Processamento (CPU) e a Unidade de Processamento Gráfico (GPU). Para o autor, Stallings (2018), a divergência fundamental entre ambas não reside apenas na contagem de núcleos, mas no modelo de desenvolvimento de cada projeto, voltada para diferentes classes de problemas computacionais.

A CPU é caracterizada por ser um processador de uso geral, otimizado para a redução da latência de execução de threads individuais. Sua arquitetura é robusta em termos de lógica de controle e memória cache. A CPU é dedicada a predição de desvios (branch prediction), execução fora de ordem e unidades de controle complexas, visando manter o fluxo de instruções sequenciais o mais contínuo possível. Então a CPU é ideal para tarefas que exigem tomadas de decisão rápidas e manipulação de fluxos de trabalho com execução variados.

Já a GPU foi concebida sob a ótica da computação massivamente paralela, focada no aumento do throughput (vazão) de dados. Então a GPU dedica a maior parte de seus transistores ao processamento de dados propriamente dito (unidades lógicas e aritméticas - ALUs) em detrimento da lógica de controle e dos grandes níveis de cache. Stallings (2018), destaca que as GPUs operam principalmente sob o modelo SIMD (Single Instruction, Multiple Data), onde uma única instrução é aplicada simultaneamente a um vasto conjunto de dados. Nas arquiteturas modernas, como a Fermi citada pelo autor, a organização se dá por meio de Multiprocessadores de Streaming (MS), que gerenciam milhares de threads ativas para reduzir a latência de acesso à memória.

Com relação ao desempenho computacional, a distinção é quantificada pela capacidade de operações de ponto flutuante (FLOPs). Enquanto a CPU apresenta um crescimento de desempenho linear e limitado, devido à complexidade do paralelismo de nível de instrução (ILP), a GPU exhibe um crescimento exponencial em sua capacidade de cálculo. Isso ocorre porque o desempenho da GPU é escalonado pela adição de mais núcleos de processamento simples, permitindo que ela execute tarefas matemáticas intensivas com uma eficiência de ordem de magnitude superior à da CPU.

Stallings ressalta que o alto desempenho da GPU é condicionado à natureza do algoritmo. Em um modelo de computação heterogênea (GPGPU), a CPU assume o papel de "host", encarregada da lógica de controle e das tarefas seriais, para que a GPU atue

como um acelerador ou "device", processando os núcleos computacionais (kernels) que apresentam alto grau de paralelismo de dados.

Portanto, a superioridade de desempenho de uma sobre a outra não é absoluta, mas dependente da carga de trabalho, onde a CPU domina na complexidade lógica e baixa latência, e a GPU é a ferramenta de escolha para o processamento de grandes volumes de dados que podem ser decompostos paralelamente.