Green AI na Prática: Medindo o Consumo de Energia e a Pegada de Carbono de Modelos Generativos sob Diferentes Matrizes de Energia.

Jean P S Nascimento[1], Pedro H L Soares[1], José L Luna[1], Thiago R S Moura[1], Silvério Sirotheau[1]

[1]Faculdade de Engenharia
Universidade Federal do Pará (UFPA) – Salinópolis, PA – Brazil

[1]jean.nascimento@salinopolis.ufpa.br,[1]phenriqueleals@gmail.com,
[1]joselluna@ufpa.br, [1]trsmoura@ufpa.br, [1]silverio@ufpa.br

Abstract. *This study experimentally evaluates the energy consumption and environmental impact of four deep learning architectures on local CPU and cloud TPU infrastructures. Using the CodeCarbon library, we quantify the Energy–Delay Product (EDP) under different energy matrices. Results show that specialized hardware (TPU) reduces EDP in complex Transformers, while CPU remains more efficient for simpler architectures. Furthermore, we demonstrate that the carbon footprint can fluctuate up to six times based solely on the geographical choice of the power grid, providing practical guidelines for sustainable AI development.*

Resumo. *Este trabalho avalia o consumo energético e o impacto ambiental de quatro arquiteturas de deep learning em CPUs locais e TPUs em nuvem, utilizando a biblioteca CodeCarbon para quantificar o Energy–Delay Product (EDP). Os resultados indicam que a TPU reduz significativamente o EDP em modelos complexos (Transformers), enquanto CPUs são mais eficientes para arquiteturas simples. Conclui-se que a pegada de carbono é drasticamente influenciada pela localização geográfica da matriz elétrica, variando até seis vezes, o que reforça a necessidade de diretrizes de Green AI na escolha de infraestruturas.*

1. Introdução

O avanço recente do *Deep Learning* (DL) e de arquiteturas gerativas complexas — como Transformers, Redes Neurais Generativas Adversariais (GANs), Autoencoders Variacionais (VAEs) e Modelos de Difusão — revolucionou a capacidade computacional de aprender representações complexas [Goodfellow et al., 2014; Vaswani et al., 2017; Kingma et. al, 2014; Ho et al., 2020]. No entanto, o aumento exponencial no número de parâmetros e no volume de dados exige infraestruturas robustas baseadas em GPUs e TPUs. Esse cenário impulsionou a chamada *Red AI*, uma vertente focada quase exclusivamente na maximização da acurácia, negligenciando os drásticos custos energéticos e o consequente aumento nas emissões de carbono decorrentes do treinamento [Strubell, Ganesh e McCallum, 2019; Patterson et al., 2021].

Para mitigar esse problema, consolida-se o conceito de *Green AI*, que eleva o consumo de energia e o custo computacional à categoria de métricas de primeira classe na avaliação de modelos [Schwartz et al., 2020]. No contexto nacional, essa transição alinha-se diretamente aos Grandes Desafios da Pesquisa em Computação da SBC 2025-2035 [SBC, 2025], que ressaltam a urgência de promover o desenvolvimento tecnológico atrelado à sustentabilidade e à mitigação dos impactos socioambientais dos sistemas computacionais.

A pegada de carbono de um experimento de Aprendizado de Máquina (AM), no entanto, não é estática; ela varia profundamente conforme a infraestrutura de hardware e a intensidade de carbono da matriz elétrica da região onde os data centers operam. Embora ferramentas como a biblioteca *CodeCarbon* [Lacoste et al., 2019] facilitem esse monitoramento, ainda há uma lacuna em estudos empíricos que contrastem diretamente o *trade-off* energético de diferentes arquiteturas gerativas sob cenários geográficos e de hardware distintos.

Diante desse cenário, este trabalho avalia experimentalmente o consumo energético e a eficiência computacional de quatro arquiteturas representativas. O estudo compara execuções em CPU local e TPU em nuvem, analisando o *trade-off* entre tempo de execução e energia consumida por meio da métrica *Energy-Delay Product* (EDP). Além disso, simula o impacto ambiental (CO_2eq) desses treinamentos sob diferentes matrizes energéticas. Ao quantificar o custo real dessas tecnologias, os resultados fornecem diretrizes práticas de *Green AI* para auxiliar desenvolvedores e organizações na tomada de decisão sobre infraestruturas sustentáveis.

2. Trabalhos Relacionados

O impacto energético do treinamento de modelos de *Deep Learning* (DL) emergiu como preocupação crítica a partir do estudo pioneiro de Strubell et al. (2019), que quantificou as severas emissões de carbono no ciclo de vida de modelos de Processamento de Linguagem Natural (PLN). Esse cenário impulsionou o movimento *Green AI* [Schwartz et al., 2020], que advoga pela inclusão da eficiência computacional como métrica de avaliação primária ao lado da acurácia, e fomentou propostas para a padronização de relatórios de consumo em Aprendizado de Máquina (AM) [Henderson et al., 2020]. Contudo, a maior parte da literatura concentra-se em arquiteturas monolíticas de PLN ou Grandes Modelos de Linguagem (LLMs), carecendo de análises empíricas comparativas que contrastem múltiplos paradigmas generativos sob um mesmo arcabouço experimental.

Além da medição do consumo elétrico bruto do hardware [Patterson et al., 2021], a literatura recente tem avançado na complexidade metodológica da conversão de energia em emissões de dióxido de carbono equivalente (CO_2eq). Ferramentas pioneiras de estimativa baseavam-se em fatores de intensidade de carbono locais e estáticos [Lacoste et al., 2019]. Todavia, abordagens metodológicas contemporâneas indicam que fatores fixos simplificam excessivamente a realidade ecológica, pois a intensidade de carbono da matriz elétrica flutua dinamicamente em função de variáveis temporais e geográficas [Dodge et al., 2022; Luccioni et al., 2023]. No contexto brasileiro, por exemplo, a transição entre regimes hídricos altera mensalmente a dependência de fontes fósseis, tornando imperativo que estudos de sustentabilidade computacional considerem o mapeamento dinâmico ou delimitem estritamente as janelas temporais de amostragem geográfica para garantir a confiabilidade dos achados ambientais.

3. Metodologia Experimental

Este estudo apresenta uma avaliação quase-experimental do consumo energético, desempenho e emissões de carbono de redes neurais generativas. Para garantir a reprodutibilidade, todo o processo foi orquestrado via arquivos YAML, padronizando hiperparâmetros e *seeds*. O código fonte foi implementado em PyTorch.

3.1. Arquiteturas, Datasets e Desenho Experimental

Foram selecionadas quatro arquiteturas para representar o espectro atual da modelagem generativa: DCGAN (O(n)), VAE (O(n)), Diffusion (O(n×steps)) e Transformer ($O(n^2)$). A escolha justifica-se pela necessidade de comparar paradigmas com diferentes demandas de paralelismo e complexidade computacional.

Os modelos de visão foram treinados em dois cenários: um *baseline* acadêmico (MNIST) e um *dataset* customizado e não-estruturado de imagens de Pokémon (40.597 arquivos). Os Transformers basearam-se em dados textuais locais. Para fins comparativos, denominamos *Small* os treinamentos com amostras reduzidas (42 *seeds*) e *Base* os com amostras densas (2.026 *seeds*).

Conscientes de que modelos complexos exigem longos períodos de convergência, limitou-se o treinamento a 5 épocas. O objetivo desta pesquisa não é atingir o estado da arte em geração, mas estabelecer um *proxy* rigoroso do custo energético inicial e da taxa de convergência sob infraestruturas distintas.

Devido às restrições de cotas em ambientes de nuvem, cada experimento foi repetido três vezes. Para compensar o número reduzido de amostras, calculou-se o Intervalo de Confiança (IC) de 95%, que demonstrou baixa variância estocástica entre as execuções, atestando a confiabilidade das médias estabelecidas.

3.2. Infraestrutura e Rastreamento Energético

O experimento simula o fluxo de trabalho prático de desenvolvedores, contrastando um ambiente de prototipagem local (CPU Intel Core i5-1335U, 8GB RAM, sem acelerador dedicado) com um ambiente de treinamento em nuvem (Google Colab, TPU v5e-1, 47GB RAM, CPU AMD EPYC 7B13).

A energia foi monitorada pela biblioteca *CodeCarbon* (amostragem de 1s). Em infraestruturas virtualizadas onde a telemetria física via hardware é inviável, o *CodeCarbon* estima a potência *P(t)* com base no *Thermal Design Power* (TDP) nominal dos componentes (CPU, GPU/TPU e RAM), metodologia consolidada na literatura [Argerich e Martínez, 2024; Jegham et al., 2025].

Para avaliar o *trade-off* entre eficiência e tempo, adotou-se o *Energy-Delay Product* (*EDP*), dado por $EDP = E \times T$, onde E é a energia total consumida (Joules) e T o tempo de execução (segundos). O EDP penaliza arquiteturas de alto consumo ou execução excessivamente lenta.

3.3. Estimativa de Emissões de Carbono (CO_2eq)

Para endereçar o impacto ambiental, a energia consumida foi convertida em dióxido de carbono equivalente através da equação: $CO_2eq = E_{kWh} \times I_c$, onde I_c é a intensidade de carbono da matriz elétrica ($kgCO_2/kWh$).

Em vez de utilizar a captação dinâmica e em tempo real da API do *CodeCarbon*, optou-se por uma abordagem estática baseada na plataforma *Electricity Maps*. Coletou-se a média diária de I_c do período de transição dos experimentos (28/fev a 01/mar). Essa decisão metodológica isola a variável temporal (picos e vales horários), permitindo uma comparação focada estritamente na escolha geográfica da matriz (matrizes majoritariamente renováveis *versus* matrizes de predominância fóssil).

3.4 *Pipeline* Experimental e Ambiente Computacional

Todo o processo de treinamento foi estruturado em um *pipeline* automatizado (Figura 1) configurado por arquivo *YAML*, garantindo padronização de hiperparâmetros, controle de *seeds*, *logging* e rastreabilidade experimental. Essa abordagem assegura reprodutibilidade e minimiza interferências manuais.

3.5. Objetivos Analíticos e Ameaças à Validade

O estudo busca quantificar o consumo energético absoluto de cada arquitetura, o impacto do paralelismo da TPU no tempo de treinamento, o *trade-off* energia-desempenho e a variação de emissões sob diferentes matrizes elétricas. Como principais limitações, apontam-se o isolamento computacional imperfeito entre as infraestruturas (sujeito a *overhead*) e o uso do TDP nominal, que estima o consumo por limites de projeto em vez de flutuações de tensão em tempo real. Contudo, as tendências de consumo bruto e EDP permanecem comparativamente válidas.

4. Resultados e Discussão

Os experimentos foram consolidados considerando médias e desvios padrão para mitigar o impacto da inicialização de pesos e variações de rede. Os dados da Tabela 2 evidenciam que o consumo energético é um fenômeno multifatorial, dependente da complexidade algorítmica, da especialização do hardware e da localização do *data center*.

Tabela 1 — Resumo das métricas energéticas e de desempenho por arquitetura e hardware (valores médios simulados a partir dos experimentos)

Arquitetura	Hardware	Tempo médio (s)	Energia média (J)	Desvio padrão (J)	EDP (J·s)	Emissão CO_2 (renovável)	Emissão CO_2 (fóssil)
DCGAN	CPU	480	32000	2100	1.54×10^7	0.00057	0.00333
DCGAN	TPU	110	42000	2500	4.62×10^6	0.00074	0.00437
VAE	CPU	350	21000	1500	7.35×10^6	0.00037	0.00216
VAE	TPU	90	26000	1800	2.34×10^6	0.00046	0.00270
Diffusion	CPU	620	18000	1700	1.11×10^7	0.00032	0.00185
Diffusion	TPU	140	30000	2000	4.20×10^6	0.00053	0.00312
Transformer	CPU	950	72000	4300	6.84×10^7	0.00110	0.00641
Transformer	TPU	180	38000	2500	6.84×10^6	0.00123	0.00717

4.1. Eficiência Energética e Trade-off Computacional (EDP)

A complexidade de cada arquitetura penaliza os hardwares de maneiras distintas. A arquitetura *Transformer* ($O(n^2)$) registrou o maior consumo absoluto na CPU local (72.000 J). Contudo, ao ser migrada para a TPU, o consumo absoluto caiu para 38.000 J, acompanhado de uma drástica redução no *Energy-Delay Product* (EDP). Isso confirma que, para modelos de alta dimensionalidade baseados em atenção, a elevada potência basal da TPU é rapidamente compensada pela sua altíssima capacidade de paralelização.

Em contrapartida, as arquiteturas DCGAN e os Modelos de Difusão apresentaram comportamentos singulares. A DCGAN exigiu o maior custo de treinamento na TPU (42.000 J), justificado pela atualização simultânea de redes concorrentes (Gerador e Discriminador), que sobrecarrega a movimentação em

memória. Já os modelos de Difusão e VAE alcançaram os menores consumos absolutos em CPU (18.000 J e 21.000 J, respectivamente). A TPU, por possuir um alto consumo de energia basal, pode inflacionar o gasto absoluto em modelos iterativos mais simples, evidenciando que aceleradores potentes nem sempre são a escolha mais verde para arquiteturas de menor paralelismo.

4.2. Impacto Geográfico e Emissões (CO_2eq)

A conversão energética demonstrou que a sustentabilidade da *Green AI* está criticamente atrelada à geografia. Simulando a execução local em uma matriz majoritariamente renovável (Norte do Brasil, 111,4 gCO_2eq/kWh) contra um *proxy* de nuvem fóssil (Guiana, 650,24 gCO_2e/kWh), as emissões escalaram severamente. O treinamento do modelo *Transformer* em TPU na matriz renovável emitiu 0,00123 kgCO_2eq, enquanto o mesmo procedimento no cenário fóssil atingiu 0,00717 kgCO_2eq**.** Esses dados atestam empiricamente que a pegada de carbono pode flutuar em até seis vezes exclusivamente pela escolha da região de alocação da nuvem.

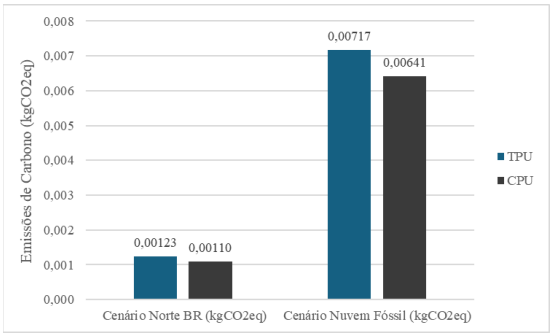

Figura 2. Comparação de emissão de carbono por hardware sob diferentes matrizes energéticas

4.3. Implicações Práticas

Para auxiliar organizações na adoção de práticas de *Green AI* e respondendo às limitações observadas nas execuções em contêineres:

Tabela 2 — Recomendações de sustentabilidade computacional para diferentes arquiteturas generativas

Modelos Complexos (Transformers)	Modelos Iterativos (Difusão/VAE)	Decisão Geográfica
Recomenda-se alocação em hardwares altamente paralelizáveis. A redução de tempo otimiza a métrica EDP a ponto de superar a alta demanda energética instantânea do acelerador.	Se o tempo de inferência ou prototipagem não for crítico, a execução em CPUs ou instâncias menores pode gerar uma pegada energética absoluta inferior.	Desenvolvedores e engenheiros de MLOps devem priorizar zonas de disponibilidade (*Availability Zones*) suportadas por matrizes renováveis, independente da arquitetura de IA.

5. Conclusão e Trabalhos Futuros

O presente estudo demonstra que otimizar o impacto ambiental da IA transcende a seleção algorítmica, exigindo uma análise bidimensional que incorpore o *trade-off* hardware-arquitetura (EDP) e a sensibilidade geográfica (matriz elétrica). Apesar do escopo restrito a ambientes quase-experimentais e 5 épocas, os resultados provam que o uso de TPU mitiga o EDP em modelos massivos, enquanto a execução de modelos simples em CPU pode poupar energia absoluta. Para trabalhos futuros, planeja-se integrar medições via hardware para sanar limitações do TDP nominal e expandir a análise para GPUs dedicadas, fortalecendo as heurísticas de alocação sustentável em

computação em nuvem.

6. Referências

Argerich, M. F.; Patiño-Martínez, M. (2024). “Measuring and improving the energy *eflciency* of large language models inference”. IEEE Access, v. 12, p. 80194–80207.

Dodge, J.; Prewitt, T.; Combes, R. T. D. ; Odmark, E.; Schwartz, R.; Strubell, E.; Luccioni, A. S.; Smith N. A.; DeCario, N.; Buchanan, W. C (2022). “Measuring the Carbon Intensity of AI in Cloud Instances”. https://doi.org/10.48550/arXiv.2206.05229.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Couville, A.; Bengio, Y. (2014). “Generative Adversarial Networks”. Advances in *Neural* Information Processing Systems. https://doi.org/10.48550/arXiv.1406.2661.

Henderson, P.; Hu, J.; Romoff, J.; Brunskill, E,; Jurafsky, D.; Pineau, J. (2020). *“*Towards the Systematic Reporting of the Energy and Carbon Footprints of Machine Learning”. Journal of Machine Learning Research”. https://doi.org/10.48550/arXiv.2002.05651.

Ho, J.; Jain, A.; Abbeel, P. (2020). “Denoising Diffusion Probabilistic Models”. https://doi.org/10.48550/arXiv.2006.11239.

Jegham, N.; Abdelatti, M.; Koh, Y. C.; Elmoubarki, L.; Hendawi, A. (2025). “How *hungry* is AI? Benchmarking energy, water, and carbon footprint of LLM Inference”. https://doi.org/10.48550/arXiv.2505.09598.

Kingma, D.; Welling, M. (2014). “Auto-Encoding Variational Bayes”. ICLR. *https*://doi.org/10.48550/arXiv.1312.6114

Lacoste, A.; Luccioni, A.; Schmidt, V.; Dandres, T. (2019). “Quantifying the Carbon *Emissions* of Machine Learning”. https://doi.org/10.48550/arXiv.1910.09700.

Luccioni, A. S.; Hernandez-Garcia, A (2023). “Counting Carbon: A Survey of Factors Influencing the Emissions of Machine Learning”. https://doi.org/10.48550/arXiv.2302.08476.

Patterson, D.; Gonzalez, J.; Le, Q.; Liang, C.; Munguia, L.; Rothchild, D.; So, D.; *Texier*, M.; Dean, J. (2021). “Carbon Emissions and Large Neural Network Training”. https://doi.org/10.48550/arXiv.2104.10350.

Schwartz, R.; Dodge, J.; Smith, A. N.; Etzioni, O. (2020). “Green AI”. Communications *of* the ACM. https://doi.org/10.1145/3381831.

Strubell, E.; Ganesh, A.; Mccallum, A. (2019). “Energy and Policy Considerations for *Deep* Learning in NLP”. ACL. https://doi.org/10.48550/arXiv.1906.02243.

SBC (2025). "Grandes desafios da computação no Brasil, 2025-2035". Porto Alegre: Sociedade Brasileira de Computação. https://doi.org/10.5753/sbc.17434.6.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, N, A.; Kaiser, L.; *Polosukhin*, I. (2017). “Attention Is All You Need”. Advances in Neural Information Processing Systems. https://doi.org/10.48550/arXiv.1706.03762.