

HeatSync: Sistema de Refrigeração Externa Automatizado para Notebooks

Gabriel C. Mourão, Luís A.L. Oliveira, Mateus F. Barbosa,
Rafael F.B.C. Oliveira, Victor F. Moraes, Rafael H.N. Diniz,
Matheus B. Pereira, Felipe D. Cunha, Matheus A. Souza

Depto. de Ciência da Computação – Pontifícia Universidade Católica de Minas Gerais
Belo Horizonte – MG – Brasil

{gabriel.mourao.1449838, luis.oliveira.695168}@sga.pucminas.br,
{rafael.barcellos, vfmoraes, mateus.barbosa}@sga.pucminas.br,
rafahdiniz@yahoo.com.br, matheuspereira@dcc.ufmg.br,
{felipe, matheusalcantara}@pucminas.br

Abstract. *This work proposes an intelligent external cooling pad for laptops, aiming to mitigate overheating without generating excessive noise. The system, controlled by an Arduino via PWM, adjusts the fan speed according to the temperature received via Bluetooth. Its effectiveness was validated by comparing it with scenarios of no external cooling and full-speed manual operation. The results demonstrate that the automatic control reduces noise by up to 24.3 dB under low load, while maintaining thermal and computational performance similar to the maximum speed mode.*

Resumo. *Este trabalho propõe uma base de refrigeração externa inteligente para notebooks, visando mitigar o superaquecimento em notebooks sem gerar ruído excessivo. O sistema, controlado por Arduino via PWM, ajusta a velocidade da ventoinha conforme a temperatura recebida por Bluetooth. Sua eficácia foi validada em comparação com cenários de ausência de refrigeração e de operação manual máxima. Os resultados demonstram que o controle automático reduz o ruído em até 24,3 dB em baixa carga, mantendo um desempenho térmico e computacional similar ao modo de máxima velocidade.*

1. Introdução

O uso de *laptops* em atividades de alto desempenho computacional, como jogos eletrônicos e renderização gráfica, tem-se tornado cada vez mais comum. Nesse contexto, um dos grandes fatores a dificultar a realização de tais tarefas é o superaquecimento da unidade de processamento gráfico (GPU) e, principalmente, da unidade central de processamento (CPU).

Visto que os sistemas de refrigeração internos são insuficientes para essa alta demanda, soluções como *coolers* externos popularizam-se [Ahmad Khalid et al. 2018].

Gabriel, Luís, Mateus, Rafael e Victor são estudantes de graduação do curso de Ciência da Computação, orientados pelos demais autores.

Contudo, grande parte desses dispositivos possui controle manual, o que acarreta duas desvantagens: a dificuldade em prever a rotação ideal em tempo real e a geração desnecessária de ruído quando se opta por manter as ventoinhas em rotação máxima.

Esse ruído em ambiente interno, abordado como um "estressor ambiental"[Campbell 1983], está correlacionado com a maior dificuldade para lidar com atividades que exigem concentração [Wallenius 2004] e também age como "estressor crônico", potencialmente reduzindo a capacidade de lidar com outras fontes de tensão [Homburg et al. 2007], como aquelas provenientes do trabalho ou da realização de projetos pessoais.

Por mais que haja estudos buscando soluções eficazes, como em [Ahmad Khalid et al. 2018], este ainda consiste em um problema em aberto. Diante desse cenário, neste trabalho desenvolveu-se um sistema de refrigeração automático, comparando seu desempenho com cenários de ausência de *cooler* externo e de utilização de ventoinha com controle manual. Os testes envolvem *benchmarks* específicos para avaliar a temperatura, o nível de ruído gerado e o desempenho computacional do equipamento sob diferentes condições de uso.

Este artigo está organizado da seguinte forma: a Seção 2 revisa os trabalhos relacionados. A Seção 3 apresenta a metodologia de avaliação e a Seção 4 detalha a implementação. A Seção 5 analisa os resultados, e a Seção 6 conclui o trabalho e aponta sugestões para pesquisas futuras.

2. Trabalhos Relacionados

A literatura apresenta sistemas de resfriamento ativo para *notebooks* com abordagens similares. O trabalho de [Ahmad Khalid et al. 2018], por exemplo, propõe um sistema baseado em Arduino UNO com uma interface gráfica (GUI) para monitoramento. Sua principal inovação reside na coleta de dados térmicos diretamente da CPU, diferenciando-se de abordagens que utilizam apenas sensores externos. A GUI permite o monitoramento de métricas como temperatura e uso de CPU, além de controlar a velocidade da ventoinha com base em limiares pré-definidos.

Em testes comparativos com modelos comerciais, o protótipo demonstrou maior eficácia na redução térmica. Contudo, o trabalho apresenta limitações, como a ausência de conectividade sem fio, testes de ruído e de *benchmarks* que avaliem o impacto no desempenho computacional (e.g., FPS) em cenários de uso prático, lacunas que o presente projeto busca preencher.

Outra abordagem explorada é o uso de módulos termoelétricos (TEC) para o resfriamento [Ranchagoda et al. 2016]. Embora eficaz na redução de temperatura, a implementação descrita revela-se inviável para o uso prático devido a desvantagens significativas: alto consumo energético (2-150 W), controle manual, complexidade e grandes dimensões. Essas limitações justificam a opção deste projeto por um sistema puramente ativo (ventoinhas), focado em portabilidade e conforto acústico.

Apesar da inspiração conceitual no Llano v12 [NLLano 2025], desenvolveu-se um projeto estrutural próprio com ventoinha de desktop para garantir compatibilidade com o controle PWM, um requisito não verificável em soluções comerciais por falta de especificações técnicas sobre as *fans* utilizadas.

3. Metodologia

A validação experimental do sistema de refrigeração considerou três aspectos principais: temperatura, desempenho computacional (pontuação de *benchmark*) e nível de ruído. A análise comparou três cenários de operação: sem refrigeração externa ('sem base'), com a base em modo manual e rotação máxima ('base manual') e com o controle automático de velocidade proposto ('base automática').

3.1. Especificação dos Materiais

O aparato experimental utilizou um *notebook* Acer Nitro 5 AN515-54 (Intel i5-9300H, NVIDIA GTX 1650), um microcontrolador Arduino UNO R3 com um módulo *Bluetooth* HC-05, e uma ventoinha IceGALE Xtra de 140 mm (2500 RPM). O circuito foi alimentado por uma fonte de 12 V/1.5 A DC, e resistores de 220 Ω foram empregados para o ajuste de tensão do sinal PWM, conforme detalhado na Seção 4.1.

3.2. Procedimentos dos Testes

A metodologia foi estruturada em dois protocolos de teste para avaliar o sistema sob diferentes perfis de carga:

- **Alta Carga (Estresse de Desempenho):** Execução de dois *benchmarks* distintos: o teste de desempenho do software PassMark e a ferramenta de avaliação do jogo *Forza Horizon 4*. Para garantir a robustez estatística, cada teste foi executado 5 vezes consecutivas em cada um dos três cenários.
- **Baixa Carga (Uso Cotidiano):** Reprodução contínua de vídeo em alta definição por 30 minutos em cada cenário.

A coleta de dados de temperatura dos componentes foi realizada com o software HWiNFO, enquanto o RPM da ventoinha foi registrado pela aplicação Java via tacômetro do Arduino. A análise comparativa entre os cenários baseou-se em estatística descritiva, calculando-se a média e a moda dos valores coletados para cada métrica, a fim de avaliar a eficácia do sistema proposto.

4. Aspectos da Implementação

A implementação do sistema consiste em uma aplicação *desktop* e um *firmware* embarcado, cuja arquitetura de comunicação é ilustrada na Figura 1.

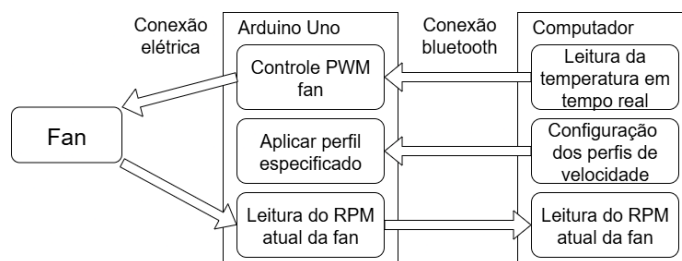


Figura 1. Diagrama de blocos da arquitetura do sistema

A aplicação *desktop*, desenvolvida em Java (com interface *Swing*), monitora as temperaturas do notebook via biblioteca *JSensors* e gerencia a comunicação bidirecional

Bluetooth com o Arduino através da biblioteca *BlueCove*. A aplicação transmite os dados de temperatura dos sensores ao Arduino e recebe a telemetria de RPM para exibição na interface. Além disso, ela configura os perfis de operação do *firmware*.

O *firmware*, desenvolvido sobre o *framework* Arduino, utiliza a biblioteca *SoftwareSerial* para comunicar-se com o módulo *Bluetooth* HC-05, por onde recebe comandos e envia a telemetria de RPM. O controle da ventoinha é feito via PWM: seu pino de controle é conectado a uma saída digital com capacidade PWM na placa do Arduino. A velocidade é modulada pela variação do *duty cycle*. A leitura do RPM, por sua vez, é feita no pino 3 (tacômetro), que emite dois pulsos por rotação [Noctua 2023]. A interrupção externa do Arduino (INT0, pino 2) captura esses pulsos na borda de subida, permitindo o cálculo de RPM pela fórmula:

$$\text{RPM} = \frac{\text{count} \times 60}{2}$$

Em que *count* é a contagem de pulsos por segundo.

O *firmware* possui dois perfis de operação.

O primeiro perfil, denominado ‘Constante’, permite ao usuário definir uma velocidade de operação fixa (0–100%), que será convertida para o valor de *duty cycle* correspondente para controlar a ventoinha.

O segundo perfil, ‘Automático’, ajusta a velocidade da ventoinha dinamicamente com base nos limites de temperatura (T_{\min} e T_{\max}) definidos pelo usuário. A partir desses limites, o *duty cycle* é calculado pelo Arduino conforme a seguinte equação:

$$\text{DutyCycle} = \text{map}(T_a, T_{\min}, T_{\max}, 0, 255)$$

Em que T_a é a temperatura atual (°C), e $[0, 255]$ é o intervalo de saída do controle *PWM* do Arduino.

4.1. Aspectos do MVP e dificuldades na implementação

A estrutura do produto mínimo viável (MVP) (Figura 2), foi confeccionada em papelão, uma alternativa à impressão 3D, considerada inviável devido às dimensões do projeto.



Figura 2. Protótipo da base de refrigeração (MVP) implementado.

Durante os testes, após um longo período de operação, identificou-se uma instabilidade no controle da ventoinha, caracterizada por pulsos incorretos de velocidade seguidos de paradas abruptas. Nessas condições, o dispositivo ignorava o sinal de controle *PWM*, tornando o ajuste de velocidade inoperante. Adicionalmente, o tacômetro registrava leituras de RPM anormalmente altas, mesmo com a ventoinha parada ou em baixa rotação, o que não condizia com o estado físico do componente.

Observou-se que o problema em questão foi causado por uma incompatibilidade de tensão entre o sinal PWM de 5V do Arduino e do componente utilizado. Embora as especificações de referência [Noctua 2023] fossem atendidas, a ausência de

documentação técnica do fabricante da ventoinha (Icegale) dificultou o diagnóstico. O problema foi solucionado com a implementação de um circuito divisor de tensão, que utiliza resistores para ajustar o sinal a um nível estável (entre 2V e 4V) e com a limitação do duty cycle máximo para 95% via software, como medida de segurança.

5. Testes de Ruído

A quantificação do ruído foi realizada com o decibelímetro para *smartphones* do NIOSH, ferramenta selecionada por sua precisão de ± 2 dB [NIOSH 2021]. Para garantir medições com mínima interferência externa, os testes foram conduzidos em um ambiente acusticamente controlado, uma sala de reuniões na biblioteca da Pontifícia Universidade Católica de Minas Gerais, unidade Coração Eucarístico. Este controle do ambiente foi crucial para permitir a correlação fidedigna entre a velocidade das *fans* e o nível de ruído emitido.

5.1. Cenários e Procedimentos de Teste

A análise do nível de ruído foi realizada em dois cenários para isolar e, em seguida, combinar as fontes sonoras. Primeiramente, mediu-se o ruído gerado apenas pelo *notebook*, variando a velocidade de suas *fans* internas. Posteriormente, analisou-se a interação sonora, mantendo a rotação do *notebook* fixa em diferentes patamares enquanto se variava a velocidade da base refrigeradora. Os dados coletados em ambos os cenários foram utilizados para gerar os gráficos apresentados na Figura 3.

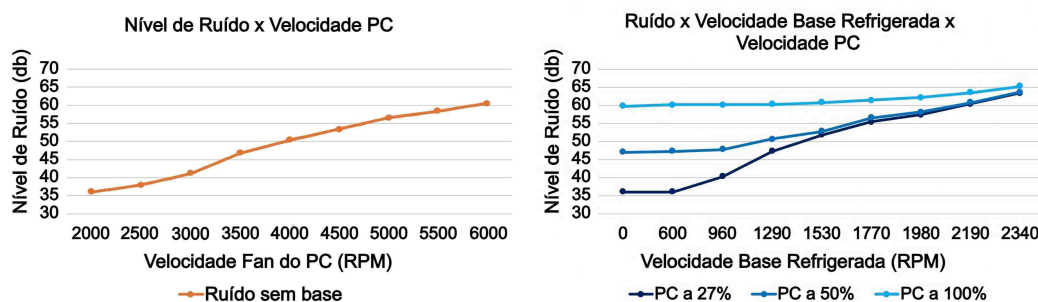


Figura 3. Ruído em função da velocidade de rotação

A análise do gráfico à esquerda evidencia uma forte correlação positiva entre o nível de ruído e a velocidade de rotação das *fans* internas do *notebook*. Este resultado corrobora os princípios apresentados por Cheremisinoff em sua obra “*Noise Control in Industry: A Practical Guide*” [Cheremisinoff 1996].

O gráfico à direita expande a análise ao mostrar o ruído combinado do sistema. Cada uma das três séries parte de um patamar fixo de rotação do notebook (27%, 50% e 100%), ilustrando o acréscimo no ruído total conforme a velocidade da base (eixo X) aumenta. Essa elevação do nível sonoro geral é consistente com a teoria acústica sobre fontes não correlacionadas, [O’Neill 2007].

5.2. Resultados dos Testes

A análise da interação sonora entre a base e o computador permitiu o desenvolvimento de uma fórmula que estima o ruído resultante, em decibéis, em função das velocidades de operação da Base (*VB*) e do *Notebook* (*VPC*). O que permitiu a realização dos testes de desempenho fora de um ambiente controlado.

$$\begin{aligned} \text{Ruído} = & 24,61 + (8,03 \times 10^{-3}) \cdot \text{VB} + (3,33 \times 10^{-3}) \cdot \text{VPC} + (7,35 \times 10^{-6}) \cdot \text{VB}^2 \\ & - (4,43 \times 10^{-6}) \cdot \text{VB} \cdot \text{VPC} + (1,37 \times 10^{-6}) \cdot \text{VPC}^2 - (5 \times 10^{-10}) \cdot \text{VB}^3 \\ & - (6 \times 10^{-10}) \cdot \text{VB}^2 \cdot \text{VPC} + (4 \times 10^{-10}) \cdot \text{VB} \cdot \text{VPC}^2 - (2 \times 10^{-10}) \cdot \text{VPC}^3 \end{aligned}$$

A fórmula foi gerada por meio de uma regressão polinomial, utilizando dados experimentais previamente divididos em conjuntos de treino e teste. A acurácia do modelo foi validada pelas métricas de Erro Médio Absoluto (MAE) e Raiz do Erro Quadrático Médio (RMSE). O modelo foi avaliado com um MAE de 0,634 dB (treino) e 0,782 dB (teste), e um RMSE de 0,804 (treino) e 0,981 (teste). Os baixos valores de MAE e RMSE indicam, respectivamente, alta precisão da fórmula em relação aos dados reais e ausência de *outliers* significativos.

6. Resultados e Discussão

Para os testes, o perfil da Base Automática foi configurado com limites mínimos de 30 °C (CPU e GPU) e máximos de 85 °C (CPU) e 65 °C (GPU). Os limites máximos foram definidos 5°C abaixo das temperaturas de risco observadas no cenário sem refrigeração, 90 °C para a CPU (início do *thermal throttling*) e 70 °C para a GPU (máximo observado para o *core* durante os testes), visando otimizar o tempo de resposta do sistema.

6.1. Benchmark: Forza Horizon 4

No benchmark de alta carga do *Forza Horizon 4*, observou-se um uso simultâneo e intenso dos componentes, com a utilização média da CPU em 70% e da GPU em 96%. O cenário 'Sem Base' resultou em severo *thermal throttling* da CPU (97,8% do tempo), com temperaturas médias de 89,6 °C. Em contraste, tanto a 'Base Manual' quanto a 'Base Automática' eliminaram completamente o *throttling*. A Base Manual reduziu a temperatura média da CPU para 83,4 °C (-6,2 °C) e da GPU para 59,4 °C (-10 °C). A Base Automática apresentou desempenho térmico equivalente, com uma variação marginal de até 1,5 °C em relação à manual, validando sua eficácia sob estresse máximo (Figura 4).

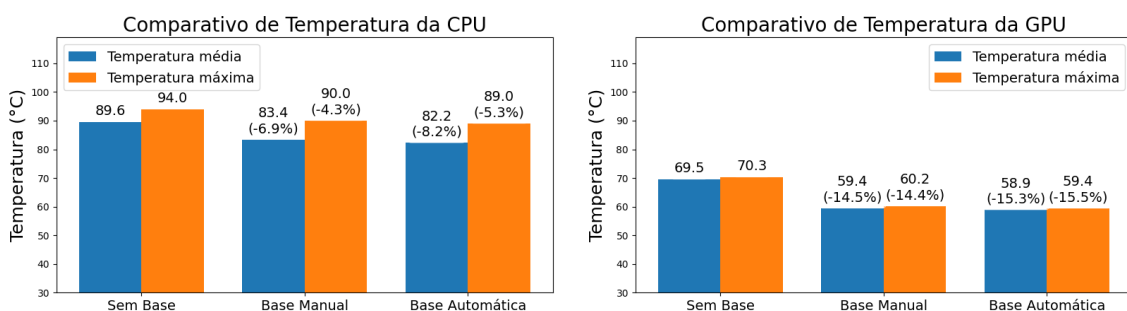


Figura 4. Temperaturas de CPU (esquerda) e GPU (direita) durante a execução.

A eliminação do *throttling* com o uso das bases proporcionou um ganho de desempenho similar em ambas as soluções, sendo 2,5% na simulação da CPU e 1,7% na taxa de quadros por segundo (FPS) da GPU. Acusticamente, os modos Manual (62,16 dB) e Automático (61,92 dB) foram similares entre si, registrando um ruído aproximadamente 3,8 dB superior ao cenário sem base, um aumento justificado pelo expressivo ganho térmico.

6.2. Benchmark: PassMark

No teste intensivo de CPU (CPU Mark), a competição por recursos entre a aplicação de monitoramento e o *benchmark* (com picos de 100% de uso) gerou latência na resposta do controle automático. Além disso, sugere-se que o 2D Mark foi impactado pelo estresse do teste anterior, resultando em um pico de temperatura de 72°C, valor idêntico ao do cenário ‘Sem Base’. Conseqüentemente, o desempenho térmico da Base Automática foi prejudicado: as temperaturas médias no CPU Mark (64°C) e no 2D Mark (61°C) foram equivalentes às do cenário ‘Sem Base’ e cerca de 7°C superiores às da ‘Base Manual’.

Nos demais testes (3D, Memory, Disk Mark), de menor estresse para a CPU, a Base Automática manteve as temperaturas controladas com uma redução de ruído significativa em comparação à Base Manual (Figura 5). Notavelmente, no Disk Mark, como o sistema não monitora o disco, a ventoinha operou em rotação mínima, resultando em um nível de ruído inferior inclusive ao do cenário ‘Sem Base’. Este fato evidencia a eficácia acústica do controle, mas também uma limitação de seu algoritmo, que não contempla o aquecimento de todos os componentes do sistema.

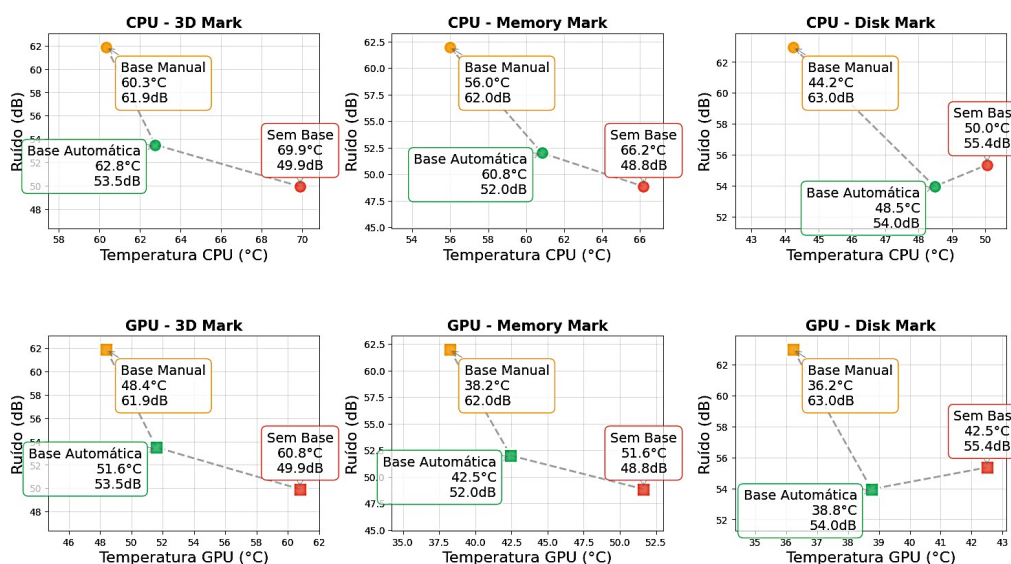


Figura 5. Ruído em função da temperatura nos testes (CPU e GPU) do PassMark.

6.3. Streaming de Vídeo 4K

No cenário de baixa carga, a principal vantagem da Base Automática foi revelada. A Base Manual, apesar da redução drástica de temperatura (CPU 37,8°C, GPU 33,5°C), impôs um custo acústico elevado de 62,9 dB (+26 dB em relação ao cenário ‘Sem Base’). Em contraste, a Base Automática proporcionou uma redução térmica menor, porém suficiente (CPU 44,8°C, GPU 39,5°C) com impacto sonoro mínimo, operando a 38,6 dB (acréscimo de apenas 1,7 dB).

Este equilíbrio foi alcançado porque a rotação da ventoinha externa no modo automático foi 73,7% inferior à do modo manual, enquanto o RPM da *fan* do *notebook* permaneceu equivalente em todos os testes. O resultado demonstra a eficácia do sistema em adequar a refrigeração à demanda real, otimizando o conforto acústico em cenários de baixo estresse, onde a potência máxima é desnecessária.

7. Conclusão e Trabalhos Futuros

Este trabalho demonstrou com sucesso a viabilidade de um sistema de refrigeração automático para notebooks, que se adapta de forma eficiente a diferentes perfis de uso. Em cenários de alta carga computacional o sistema preveniu o *thermal throttling* com um desempenho térmico equivalente ao da operação manual em rotação máxima. Em contrapartida, em cenários de baixa carga sua principal vantagem foi revelada: uma drástica redução do ruído ao adequar a velocidade da ventoinha à demanda térmica real. Dessa forma, a solução proposta atinge um equilíbrio eficaz entre desempenho computacional, controle térmico e conforto acústico.

Como trabalhos futuros, propõem-se: a otimização da coleta de dados para mitigar a latência por concorrência de processos e expandir o monitoramento para outros componentes (e.g., disco); a análise e otimização do *overhead* computacional da aplicação de monitoramento; e, por fim, o aprimoramento dos perfis de controle com a investigação de curvas de ventilação mais complexas (e.g., não lineares) para otimizar a relação desempenho-ruído.

Referências

- Ahmad Khalid, S. K., Samsudin, N., Nordin, N., and Aripin, M. (2018). Laptop cooling pad temperature monitoring system. *Indonesian Journal of Electrical Engineering and Computer Science*, 12:420.
- Campbell, J. M. (1983). Ambient stressors. *Environment and Behavior*, 15(3):355–380.
- Cheremisinoff, N. P. (1996). *Noise Control in Industry: A Practical Guide*. Noyes Publications, Park Ridge, NJ, USA.
- Homburg, A., Stolberg, A., and Wagner, U. (2007). Coping with global environmental problems: Development and first validation of scales. *Environment and Behavior*, 39(6):754–778.
- NIOSH (2021). NIOSH Sound Level Meter App. <https://www.cdc.gov/niosh/noise/about/app.html>. Accessed: 2025-05-09.
- NLLano (2025). Llano v12 gaming laptop cooling pad. <https://www.nllano.com/electronic-series/gaming-laptop-cooling-pad>. Acessado em 20 de março de 2025.
- Noctua (2023). Noctua pwm specifications white paper. Technical report, Noctua. Acesso em 20 de março de 2025.
- O'Neill, K. (2007). Acoustics and psychoacoustics: Introduction to sound – part 3.
- Ranchagoda, N., Akram, M., Vithanage, C., and Jayasundere, N. (2016). Implementation of an external intelligent cooling system for laptops using tecs. In *2016 IEEE 6th International Conference on Consumer Electronics-Berlin (ICCE-Berlin)*, pages 104–107. IEEE.
- Wallenius, M. A. (2004). The interaction of noise stress and personal project stress on subjective health. *Journal of Environmental Psychology*, 24(2):167–177.