HONEST: FERRAMENTA AUTOMATIZADA DE OTIMIZAÇÃO INTEGRADA DE WINDOWS E FIRMWARE PARA DESEMPENHO DE COMPUTADORES

DOI: https://doi.org/10.5281/zenodo.20761339
Recebido: 10/05/26 | ***Avaliado:*** 02/06/26 | ***Aceito:*** 09/06//26 | ***Publicado:*** 19/06//26

Danilo Souza Barreto
Cientista da Computação, egresso da Unijorge.
E-mail: danilobarreto93@hotmail.com

José Eduardo de Azevedo Ribeiro
Cientista da Computação, egresso da Unijorge.
E-mail: eduazevedo.contato@gmail.com

Marcos Santos Leite
Especialização em Educação a Distância, pela faculdade - Serviço Nacional de Aprendizagem Comercial Brasil (SENAC); Especialização em Educação Profissional e Tecnológica pelo Instituto Federal da Bahia (IFBA); Pós-graduação em Ciências de Dados e Inteligência Artificial pela FASUL Educacional; MBA em Gestão de Projetos pela FASUL Educacional; MBA em Empreendedorismo e Inovação pela FASUL Educacional. Graduado em Administração com Habilitação em Análises de Sistemas pela Fundação Visconde de Cairu; Graduado em Análise e Desenvolvimento de Sistemas, pelo Centro Universitário Jorge Amado (Unijorge); Licenciado em Educação Profissional e Tecnológica pelo Instituto Federal da Bahia (IFBA).
E-mail: profmarcosleite@hotmail.com

Celso Barreto da Silva
Doutorando em Difusão do Conhecimento pelo Instituto Federal da Bahia (IFBA); Mestrado em Sistemas e Computação pela UNIFACS; Graduado em Administração com Habilitação em Análise de Sistemas pela Faculdade Visconde de Cairu; Licenciado em Matemática para o Ensino Básico pela Faculdade Educacional da LAPA FAEL. Graduado em Análise e Desenvolvimento de Sistemas pelo Centro Universitário Jorge Amado (Unijorge); Pós-graduação em Metodologia do Ensino da Matemática e Docência no Ensino Superior pelo Centro Universitário Leonardo Da Vinci.
E-mail: profcelsobarreto@hotmail.com

Fabio Fonseca Barbosa Gomes
Mestrado em Sistemas e Computação pela Universidade Salvador; Graduação em Bacharelado em Sistemas de Informação pelo Centro Universitário Estácio da Bahia. Atualmente é professor assistente no Senai Cimatec e na Escola Bahiana de Medicina e Saúde Pública. Além disso, é professor universitário e coordenador do Centro Universitário Jorge Amado. Tem experiência na área de Ciência da Computação, com ênfase em Redes de Computadores, atuando principalmente nos seguintes temas: Saúde Digital, Ensino à Distância, Sistemas de Informação, Automação Residencial, Computação Ubíqua e Pervasiva e Redes de Computadores.
E-mail: fabiofbg@gmail.com

RESUMO

Este artigo apresenta o desenvolvimento de um sistema automatizado de otimização de desempenho computacional denominado Honest, voltado à integração de ajustes no sistema operacional Windows e no firmware (UEFI/BIOS). O objetivo principal foi analisar como a automação desses parâmetros pode melhorar o desempenho geral da máquina, proporcionando maior fluidez e estabilidade em aplicações de uso intensivo, como jogos eletrônicos. O estudo, de natureza aplicada, com abordagem quantitativa e caráter exploratório, foi conduzido por meio de experimentos comparativos realizados em computadores com diferentes configurações de hardware, utilizando ferramentas de benchmark como CapFrameX, OCAT e DWM. Os testes demonstraram ganhos de até 136% no desempenho gráfico e 28% em jogos eletrônicos, comprovando a eficácia das otimizações automatizadas. Além disso, foram incorporadas práticas de segurança baseadas em DevSecOps e SSDLC, assegurando a integridade do sistema e minimizando riscos associados a modificações sensíveis. O Honest consolida-se como uma ferramenta segura, eficiente e acessível, democratizando o acesso a otimizações avançadas de desempenho para diferentes perfis de usuários.

Palavras-chave: Sistema Automatizado. Desempenho Computacional. Benchmark.

ABSTRACT

This article presents the development of an automated computational performance optimization system called Honest, aimed at integrating adjustments in the Windows operating system and firmware (UEFI/BIOS). The main objective was to analyze how the automation of these parameters can improve the overall performance of the machine, providing greater fluidity and stability in intensive applications, such as electronic games. The study, of an applied nature, with a quantitative and exploratory approach, was conducted through comparative experiments performed on computers with different hardware configurations, using benchmark tools such as CapFrameX, OCAT, and DWM. The tests demonstrated gains of up to 136% in graphics performance and 28% in electronic games, proving the effectiveness of automated optimizations. In addition, security practices based on DevSecOps and SSDLC were incorporated, ensuring system integrity and minimizing risks associated with sensitive modifications. Honest establishes itself as a secure, efficient, and accessible tool, democratizing access to advanced performance optimizations for different user profiles.

Keywords: Automated System. Computational Performance. Benchmark.

RESUMEN

Este artículo presenta el desarrollo de Honest, un sistema automatizado de optimización del rendimiento computacional, diseñado para integrar ajustes en el sistema operativo Windows y el firmware (UEFI/BIOS). El objetivo principal fue analizar cómo la automatización de estos parámetros puede mejorar el rendimiento general del equipo, proporcionando mayor fluidez y estabilidad en aplicaciones exigentes, como los videojuegos. El estudio, de carácter aplicado, con un enfoque cuantitativo y exploratorio, se llevó a cabo mediante experimentos comparativos realizados en ordenadores con diferentes configuraciones de hardware, utilizando herramientas de evaluación comparativa como CapFrameX, OCAT y DWM. Las pruebas demostraron mejoras de hasta un 136 % en el rendimiento gráfico y un 28 % en videojuegos, lo que prueba la eficacia de las optimizaciones automatizadas. Además, se incorporaron prácticas de seguridad basadas en DevSecOps y SSDLC, garantizando la integridad del sistema y minimizando los riesgos asociados a modificaciones sensibles. Honest se consolida como una herramienta segura, eficiente y accesible, democratizando el acceso a optimizaciones de rendimiento avanzadas para diferentes perfiles de usuario.

Palabras clave: Sistema automatizado. Rendimiento computacional. Evaluación comparativa.

1 INTRODUÇÃO

O avanço das tecnologias de automação tem impulsionado o desenvolvimento e o uso de *softwares* voltados à otimização de sistemas computacionais. Essa tendência surge da crescente necessidade dos usuários em aprimorar o desempenho de seus computadores, especialmente daqueles que utilizam máquinas menos potentes, a fim de garantir melhor desempenho no processamento de dados, seja para melhorar a eficiência em trabalhos, estudos ou até mesmo em jogos eletrônicos. Isto é possível através de ajustes no hardware, sistema operacional e no *firmware* das placas-mães. Atualmente, o *firmware* usado na maioria dessas placas é UEFI (*United Extensible Firmware Interface*), sendo ela a evolução da BIOS tradicional (*Basic Input/Output System*) (HP, 2024).

Alguns softwares como OMEN AI da HP e APO da Intel, surgem como exemplos de que ajustes em sistemas operacionais do Windows acompanhado do uso de ferramentas de automação podem ajudar para melhorar eficiência da máquina e em suas performances, levando a otimizações de desempenhos (LIU, 2025), além de poder oferecer melhorias nos FPS (*Frames per Second*) de jogos de eletrônicos (MUFFASIL, 2025). Para Qi *et al.* (2022), configurações no *firmware* (UEFI/BIOS) apresentou ganhos de desempenho de 43.7% através de experimentos realizados em um servidor.

Embora existam esses sistemas, nenhum deles reúne, de forma integrada e automatizada, as configurações aplicadas tanto no sistema operacional quanto no firmware, não tornando o processo acessível para usuários sem conhecimento técnico, que como consequência, acabam recorrendo a ajustes manuais, que além de complexos, podem gerar riscos e demandam tempo. Nesse contexto, a necessidade de um sistema unificado também envolve considerar práticas estruturadas de segurança, como DevSecOps (*Development, Security, and Operations*) e SSDLC (*Secure Software Development Lifecycle*), que orientam a identificação e mitigação de vulnerabilidades desde as etapas iniciais do desenvolvimento (AMAZON, 2025). Assim, torna-se essencial uma solução capaz de unificar ambos os ambientes de configuração, oferecendo uma alternativa mais segura, prática e eficiente para alcançar melhorias de desempenho nos computadores dos usuários.

Diante desse panorama, este trabalho possui como objetivo a criação de um *software* que automatize ajustes no sistema operacional e no *firmware* para otimizar o uso do sistema computacional chamado de Honest. Para atingir este objetivo, é necessário:

(1) Implementar ajustes automáticos no Registro do Windows (*regedit*) e em *drivers* de vídeo de forma segura;

(2) Automatizar aplicações de ajustes básicos no *firmware*, com foco em estabilidade e ganho de performance;

(3) Testar o software em computadores de recursos limitados e em jogos eletrônicos para avaliar ganhos de desempenho e usabilidade;

(4) Realizar testes com computadores de diferentes configurações, utilizando mapas de *benchmark*, permitindo a comparação dos resultados antes e depois da aplicação das otimizações.

Espera-se que esta solução auxilie estudos do campo da automação de sistemas operacionais, explorando as relações entre configurações de software e firmware na busca por eficiência, pretendendo que os resultados obtidos aumentem as possibilidades de otimização, servindo também como um modo de referência para pesquisas futuras na área.

2 METODOLOGIA

O estudo realizado é de natureza aplicada, sendo definido por Gil (2017) como aquele que utiliza conhecimentos previamente adquiridos para coletar, selecionar e processar fatos e dados, a fim de obter e confirmar resultados que gerem impacto prático na solução de problemas concretos. Adota abordagem quantitativa, através da coleta e análise de dados numéricos que permitem generalizações estatísticas, conforme discutido por Flick (2019). O caráter exploratório é justificado segundo Gil (2017), uma vez que a pesquisa exploratória tem por finalidade permitir ao pesquisador uma maior familiaridade com a problemática, oferecendo flexibilidade para que sejam construídas as hipóteses e novas soluções.

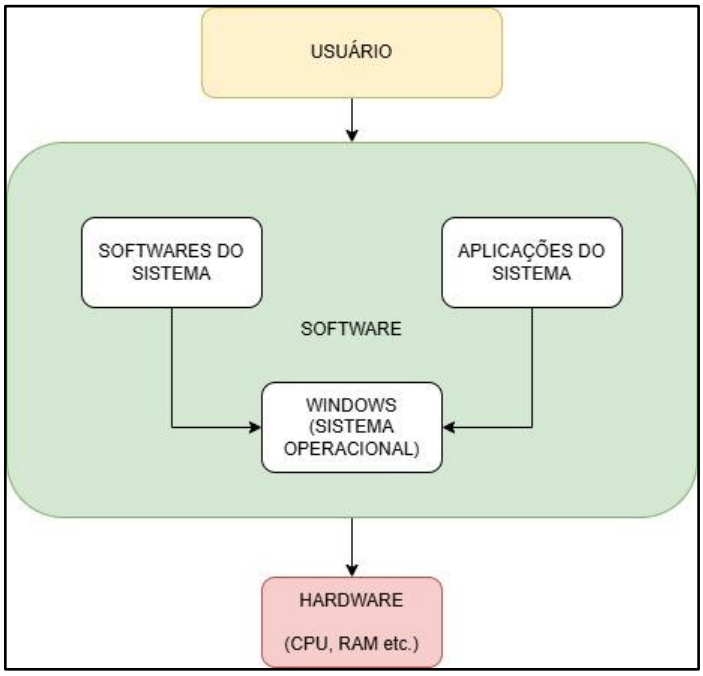
O estudo compreendeu pesquisas bibliográficas em artigos e relatórios técnicos, além da implementação prática do *software* Honest em computadores de diferentes configurações. A coleta de dados foi realizada por meio de *benchmarking* comparativo (antes e depois da aplicação das otimizações), permitindo a comparação quantitativa de métricas de desempenho em cenários de jogos eletrônicos e no desempenho geral do computador.

3 REFERENCIAL TEÓRICO

A partir da quarta geração da criação dos sistemas operacionais, o Windows surge no mercado e se consolida como um dos principais sistemas operacionais do mundo, contando

com diversas atualizações até os dias de hoje (NOLETO, 2022). Para compreendermos sua importância dentro de um computador e o porquê de o sistema operacional ser uma parte importante para realizar otimização para um computador, é necessário entender a arquitetura do Windows e como diferentes elementos do computador interagem com ele.

Figura 1 - Diagrama de relação entre usuário, softwares, sistema operacional e hardware



Fonte: Noleto (2022).

Conforme pode ser visto na Figura 1, na parte superior temos o usuário, que interage com o sistema ao utilizar programas e recursos. Esses programas são divididos em softwares do sistema, que incluem ferramentas e utilitários responsáveis por manter o computador funcionando, e aplicações do sistema, que são os programas usados diretamente para realizar tarefas do dia a dia, como navegadores ou editores de texto.

No centro do diagrama há o sistema operacional, representado pelo Windows, que atua como intermediário entre o usuário, os programas e o *hardware*, garantindo que os comandos dados pelo usuário sejam interpretados e executados de forma correta. Por fim, no final da figura está o hardware, representado por componentes físicos como processador (CPU), memória RAM e outros dispositivos. O sistema operacional organiza e distribui seus recursos para que todos os programas possam funcionar adequadamente.

Com base nessa estrutura, percebe-se a importância do Windows no desempenho do computador, evidenciando que ajustes de otimização dentro desse ambiente seriam de grande

valia, levando a melhor comunicação entre *software* e *hardware*. Para adquirir tal aprimoramento de desempenho, destacam-se entre os principais tipos de otimizações: (I) modificações no Registro do Windows (regedit), que permitem desativar funções desnecessárias; (II) configurações de planos de energia, que equilibram consumo e performance; (III) atualização e otimização de drivers de vídeo; e (IV) redução de serviços em segundo plano, liberando recursos importantes como CPU e memória RAM (DELL, 2024). Segundo a Microsoft (2022), aplicar essas otimizações pode resultar em melhorias significativas no desempenho geral da máquina.

Para validar tais melhorias, podem ser utilizados benchmarks, que são testes usados para realizar comparações de desempenho. Segundo Rezende *et al.*, (2024), o uso de benchmarks para parâmetros é fundamental para identificar principalmente gargalos de CPU e GPU, além de avaliar a estabilidade do sistema em cenários de uso intensivo. Dessa forma, nota-se a importância de diferentes tipos de otimizações dentro do sistema operacional, que leva a melhor eficiência da máquina, podendo ser comprovado através de testes realizados por meio de *benchmark* após tais alterações.

Todavia, o desempenho de um computador não se limita apenas das configurações do sistema operacional, havendo também ajustes no *firmware*, que é uma parte relevante da máquina, sendo igualmente importante compreender sua função e o impacto no panorama da eficiência computacional e otimizações de desempenho.

3.1 *FIRMWARE,* BIOS E UEFI NA EFICIÊNCIA COMPUTACIONAL

O *firmware* é um tipo de software incorporado a um *hardware,* controlando o processo de inicialização do mesmo e permitindo que ele e seus recursos funcionem corretamente antes que o sistema operacional assuma o controle, sendo o primeiro programa a ser executado, realizado inicialmente pelo BIOS, com o papel de orquestrar as suas etapas e preparação da plataforma (FLINDERS, 2024; KACPERSKI *et al*., 2024). Porém, à medida que os computadores foram evoluindo, o BIOS foi se tornando limitado em suas funções, não oferecendo nenhuma aplicabilidade avançada (KACPERSKI *et al.*, 2024), surgindo então o seu sucessor UEFI, oferecendo mais segurança e evoluções nessa tecnologia, como tabelas de dados que contêm informações relacionadas à máquina, além de chamadas de serviço de inicialização e tempo de execução disponíveis para o sistema operacional (UEFI, 2025).

Explorado por Qi *et al*., (2022), a otimização automatizada de mais de 100 parâmetros de BIOS em servidores demonstrou ganhos de até 43,7% comparado ao estado inicial,

alcançados através de um sistema inteligente que testou diferentes combinações de configurações de *firmware*, identificando quais combinações produziam melhor desempenho para cada tipo de tarefa, e então assim, aplicando automaticamente os ajustes mais eficientes. O algoritmo não só superou configurações tradicionais empíricas, como também demonstrou mais estabilidade e menor risco de falhas do que métodos puramente heurísticos, comprovando que as configurações adequadas de *firmware* melhoram significativamente o desempenho em plataformas computacionais.

Transpondo para computadores pessoais, Posey (2021) mostra que ajustes aplicados diretamente no firmware, como a desativação do *Multi-Threading* (*Hyper-Threading*) e a ativação do *Turbo Boost* são ajustes importantes que garantem melhor desempenho principalmente para usuários com hardware de menor capacidade. Segundo a Intel (2025) o *Hyper-Threading* executa tarefas em segundo plano sem interrupções, melhorando a eficiência em multitarefa e a responsividade geral do sistema, enquanto *Turbo Boost* permite aumentos dinâmicos de *clock speed* até a frequência máxima mantendo limites seguros de temperatura e energia (INTEL, 2025). Em jogos eletrônicos, esses ajustes melhoram fluidez, redução de *stutters* e maior estabilidade de *frame rates*, proporcionando experiência perceptivelmente superior durante a jogatina e validada através de *benchmarks.*

Ademais, podemos citar também a configuração XMP (*Extreme Memory Profile*), uma tecnologia desenvolvida pela Intel que permite aos usuários ativarem perfis pré-configurados de memória RAM que aumentam suas velocidades acima dos padrões de fábrica (KHULLAR, 2024). Por padrão, quando a memória RAM é instalada no computador, a placa-mãe executa a memória em velocidades padrão inferiores àquelas para as quais o módulo foi especificado, fazendo com que o usuário não aproveite o potencial completo do hardware adquirido. Ao ativar o XMP, o usuário pode alcançar velocidades mais altas de transferência de dados, resultando em redução de gargalos de memória e melhor responsividade do sistema, particularmente em tarefas como *gaming,* edição de vídeo e renderização 3D.

No entanto, deve-se pontuar que cada ajuste traz custos, como maior consumo de energia e aumento de temperatura caso o computador do usuário não possua um resfriamento adequado (Posey, 2021), sendo de grande valia avaliar bem os ajustes para diferentes tipos de máquinas. Pensando nisso, é essencial analisar algumas das soluções que já existem no mercado.

3.2 *SOFTWARES* DE OTIMIZAÇÕES EXISTENTES E SUAS LIMITAÇÕES

O avanço das tecnologias de automação trouxe aos usuários novos softwares voltados à otimização de sistemas, especialmente no ambiente Windows, ajustando automaticamente configurações de hardware e software para melhorar a eficiência e estabilidade das máquinas, reduzindo a necessidade de intervenção manual por parte do usuário.

Entre os *softwares* de otimização já existentes que contemplam ajustes no Windows e automação como uma dupla, destaca-se a Intel APO, que funciona como um protocolo de funcionamento específico para o DTT (*Intel Dynamic Tuning Technology).* A otimização é baseada em políticas avançadas de escalonamento de *threads* (*thread scheduling*) criadas pela própria Intel, tendo como referência padrões de uso. Sua otimização é aplicada apenas em jogos suportados e processadores compatíveis, com limitação de funcionamento apenas em CPUs híbridas da 14ª geração (KERBER, 2023).

Já a OMEN AI da HP, utiliza algoritmos de *machine learning* para treinamento com milhões de pontos de dados coletados de usuários de jogos eletrônicos ao redor do mundo. Pela grande quantidade das bases de dados, permite que o sistema identifique configurações específicas para cada combinação de hardware e jogo, permitindo uma otimização personalizada de forma automatizada, levando a melhorias de desempenho de até 35% em jogos intensivos de GPU (HP, 2024).

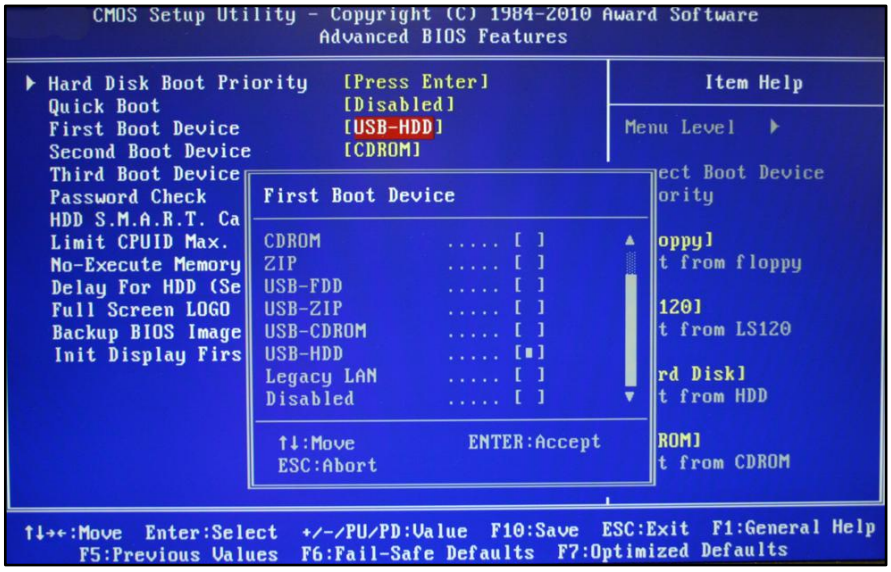
Apesar dos *softwares* apresentados serem exemplos de que a implementação de otimizações de forma automatizada no sistema operacional proporciona ganhos de desempenho significativos aos computadores dos usuários, eles apresentam limitações. Esses programas detêm grande foco apenas em jogos eletrônicos para melhoria de jogabilidade e FPS (*Frame per Second*), não explorando outros tipos de demandas e, sobretudo, não integram ajustes do sistema operacional acompanhado daqueles de *firmware* em um mesmo processo, que poderiam levar o computador do usuário a um melhor desempenho. Soma-se a isso o fato de que alterações manuais no *firmware* permanecem complexas e arriscadas, o que reforça a necessidade da automação como meio de simplificar esse processo.

3.3 AUTOMAÇÃO E SEGURANÇA EM AJUSTES DE SISTEMA

Com o crescimento de sistemas voltadas a automação de processos e atividades, considerar metodologias de segurança já nas fases iniciais do desenvolvimento garante um ambiente confiável ao usuário. Ajustes manuais, especialmente no *firmware*, são complexos,

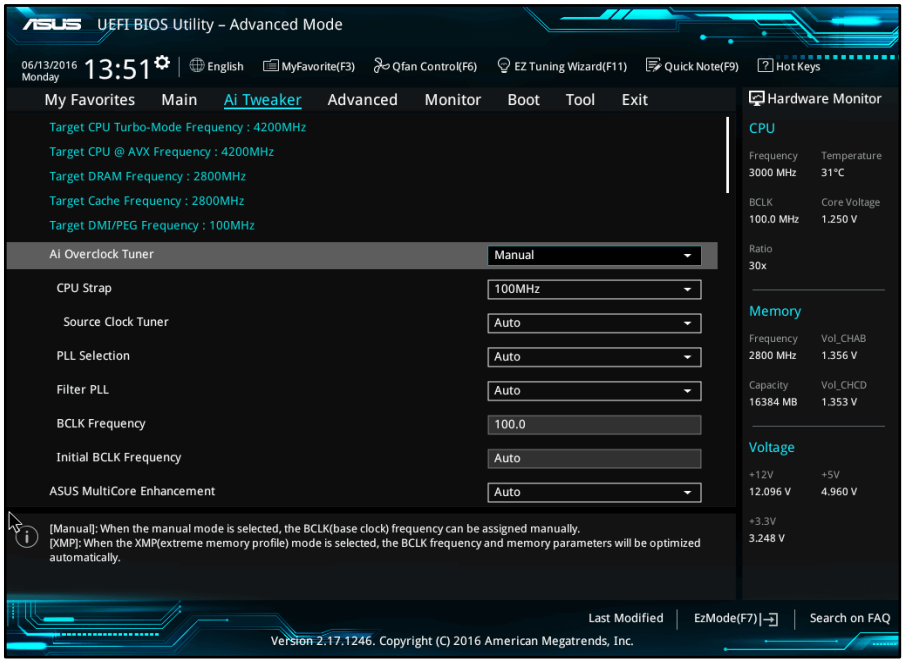
bastante demorados e sujeitos a erros, além da interface padrão do UEFI/BIOS, ilustrada nas Figuras 2 e 3, ser nada intuitiva. Para Kroth *et al.*, (2025), usuários que não possuem conhecimento técnico fazer configurações de forma manual nesses ambientes nos quais envolvem centenas de parâmetros capazes de interferir nos resultados, acaba sendo inviável. A automação é, portanto, a alternativa para minimizar a manualidade e reduzir falhas (OLAES, 2025).

Figura 2 - Interface do BIOS



Fonte: Kandwal (2022).

Figura 3 - Interface do UEFI



Fonte: Kandwal (2022).

Lowe *et al.* (2022) demonstraram através de 63 estudos de caso que intervenções automatizadas resultaram em reduções de riscos e aumentos de produtividade, ao mesmo tempo em que democratizaram o acesso a configurações avançadas. Evidencia-se assim, a possibilidade de usuários com hardware limitado ou com pouco conhecimento técnico em aplicar otimizações de alto nível, algo antes restrito a usuários já adeptos a esse processo. Mas para realizar um software automatizado dessas tarefas, é necessário garantir a integridade e a segurança do usuário, dado a complexidade e riscos dessas configurações.

Para tornar esse processo de automação seguro, práticas de segurança como DevSecOps (*Development, Security, and Operations*) e SSDLC (*Secure Software Development Life Cycle*) são de suma importância nessa aplicação pelo fato de garantir que seus princípios de segurança sejam incorporados desde as fases iniciais do desenvolvimento do sistema, evitando vulnerabilidades. De acordo com Pressman & Maxim (2021), a integração contínua de segurança no ciclo de vida do *software* representa um avanço fundamental na engenharia de *software* moderna, transformando segurança de uma preocupação pós-desenvolvimento para um imperativo do *design* inicial.

Na prática de DevSecOps, inclui etapas como análise de código, para identificar falhas antes da implantação; gerenciamento de alterações, que assegura rastreabilidade em cada modificação; modelagem de ameaças, antecipando riscos potenciais; e até mesmo o treinamento contínuo de equipes de desenvolvimento, que leva a manter as práticas de segurança sempre atualizadas (AMAZON, 2025).

Já no SSDLC, a ênfase está em aplicar segurança em todas as fases do ciclo de vida do software, desde a definição de requisitos até a manutenção. Isso significa, por exemplo, prever riscos de instabilidade em diferentes perfis de *hardware*, validar permissões de administrador durante a implementação e realizar testes de estresse ou *rollback* em cenários críticos, levando a garantia que a ferramenta mantenha a confiabilidade mesmo diante de falhas (AMAZON, 2025).

Além das citadas metodologias, ressalta-se a implementação de ferramentas que reforçam a proteção da aplicação, como o VMProtect, que abrange técnicas de obfuscação, criptografia e virtualização de código, que acaba dificultando tentativas de engenharia reversam ou modificações maliciosas no software (VMPROTECT, 2025).

Mostra-se então, que a união entre a necessidade de um *software* automatizado de processos de otimizações de computadores e a incorporação de práticas de segurança e mecanismos de proteção nos leva à eficiência e confiabilidade desta aplicação além da praticidade e acessibilidade, garantindo segurança ao usuário e à sua máquina.

3.4 DESENVOLVIMENTO DE SISTEMAS

O sistema de otimização, denominado Honest, foi desenvolvido utilizando a linguagem C# em conjunto com o *framework* .NET, escolhida devido à sua integração nativa com o Windows e ao acesso direto às APIs de baixo nível necessárias para realizar modificações no sistema. A arquitetura do aplicativo foi projetada para executar ajustes automatizados tanto em configurações das placas de vídeo NVIDIA e AMD quanto em parâmetros do sistema operacional, visando maximizar o desempenho em jogos eletrônicos e no uso geral do computador.

Por conta do Honest realizar configurações que são sensíveis no sistema, foi adotada a ferramenta VMProtect, que garante a proteção do código-fonte através da virtualização e ofuscação do código compilado, dificultando tentativas de descompilação e engenharia reversa. Também foram implementadas as metodologias de segurança DevSecOps (*Development, Security, and Operations*) e SSDLC (*Secure Software Development Life Cycle*), para mitigar riscos de instabilidade, falhas de execução e vulnerabilidades durante o uso do sistema. Além do mais, o sistema também cria pontos de restauração e backups automáticos antes da aplicação de qualquer ajuste, permitindo reverter as alterações em caso de falhas ou incompatibilidades.

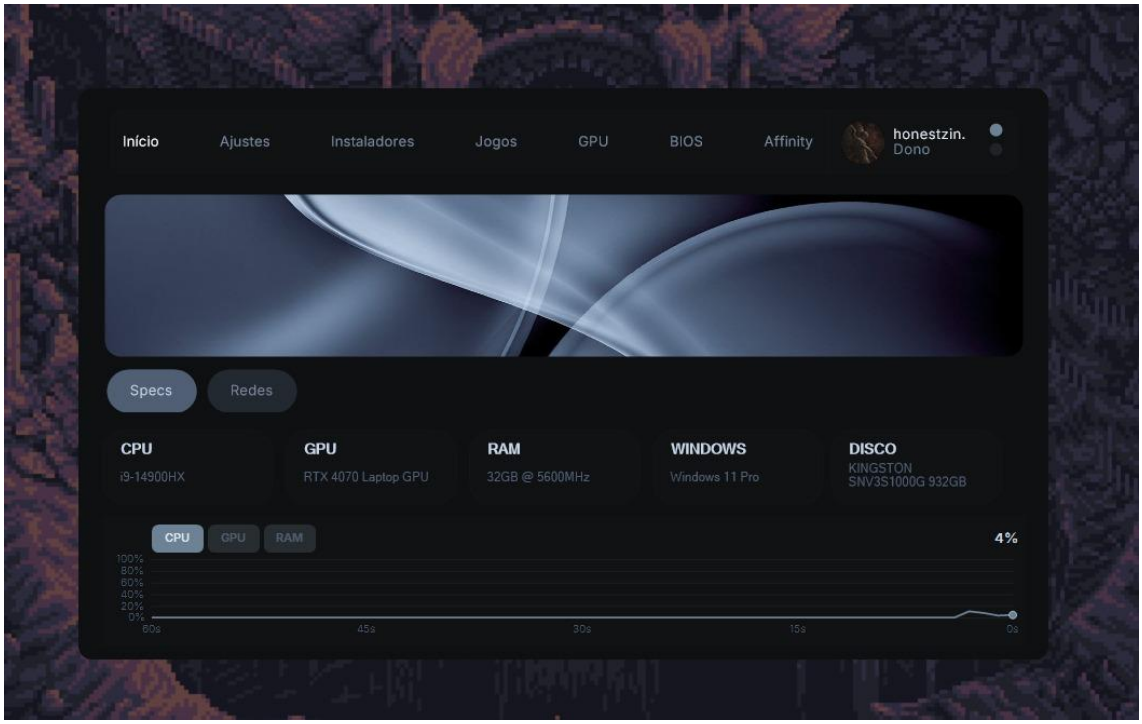
O Honest também identifica automaticamente qual placa de vídeo instalada e aplica as configurações adequadas para cada caso. Quando há uma GPU NVIDIA presente, o aplicativo utiliza chamadas diretas à NVAPI (NVIDIA API), biblioteca proprietária que possibilita o controle programático de diversas configurações do driver gráfico, como perfis de energia, latência e parâmetros de renderização. Já para placas AMD, as modificações são realizadas através do Registro do Windows e da ADL (*AMD Display Library*), que também traz ajustes parecidos em relação ao desempenho, gerenciamento térmico e perfis específicos de aplicações. Independente do fabricante da GPU, o sistema também realiza alterações em chaves específicas do Registro através das classes do *namespace* Microsoft.Win32, modificando configurações em relação a melhor performance do gráfico, gerenciamento de energia e prioridades de processos.

Quanto às otimizações do sistema operacional, foram implementados ajustes de prioridades de processos em tempo de execução usando a classe Process, enquanto modificações de baixo nível em alocação de memória, agendamento de threads e políticas de energia do processador foram viabilizadas através de chamadas a funções não gerenciadas da API Win32, utilizando P/Invoke (Platform Invocation Services). O sistema também gerencia

serviços do Windows por meio da classe ServiceController, desabilitando aqueles considerados não essenciais durante sessões de usos para liberar recursos computacionais.

A interface gráfica do *software*, mostrada nas Figuras 4, 5 e 6, foi desenvolvida em Windows Forms, oferecendo uma experiência intuitiva onde o usuário pode escolher quais otimizações aplicar e, se necessário, revertê-las posteriormente. Como medida de segurança, antes de qualquer modificação o aplicativo cria automaticamente pontos de restauração do sistema e realiza *backup* das configurações originais do Registro, garantindo que seja possível retornar ao estado anterior caso ocorra alguma instabilidade ou o usuário não fique satisfeito com os resultados.

Figura 4 - Interface inicial do Honest



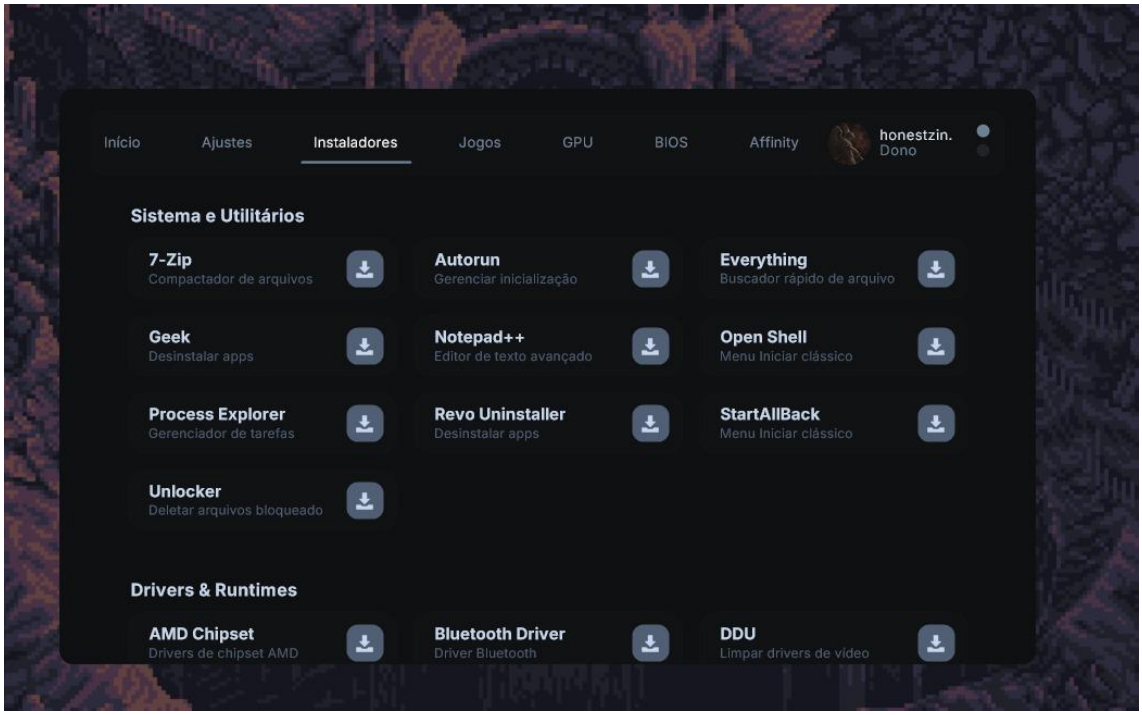
Fonte: Autoria própria (2026).

Na Figura 4 é possível verificar o painel do Honest. Nele, são exibidas as informações do sistema, incluindo o processador (Intel i9-14900HX), a placa de vídeo (RTX 4070 Laptop GPU), a memória RAM (32 GB a 5600 MHz), o sistema operacional (Windows 11 Pro) e o armazenamento (SSD Kingston de 932 GB).

Na parte inferior, há um gráfico de utilização em tempo real, com abas para monitoramento de CPU, GPU e memória RAM. O menu superior oferece acesso a diferentes módulos do sistema, como configurações, instaladores, jogos, GPU, BIOS e afinidade,

caracterizando uma interface voltada à administração e ao acompanhamento do desempenho do computador.

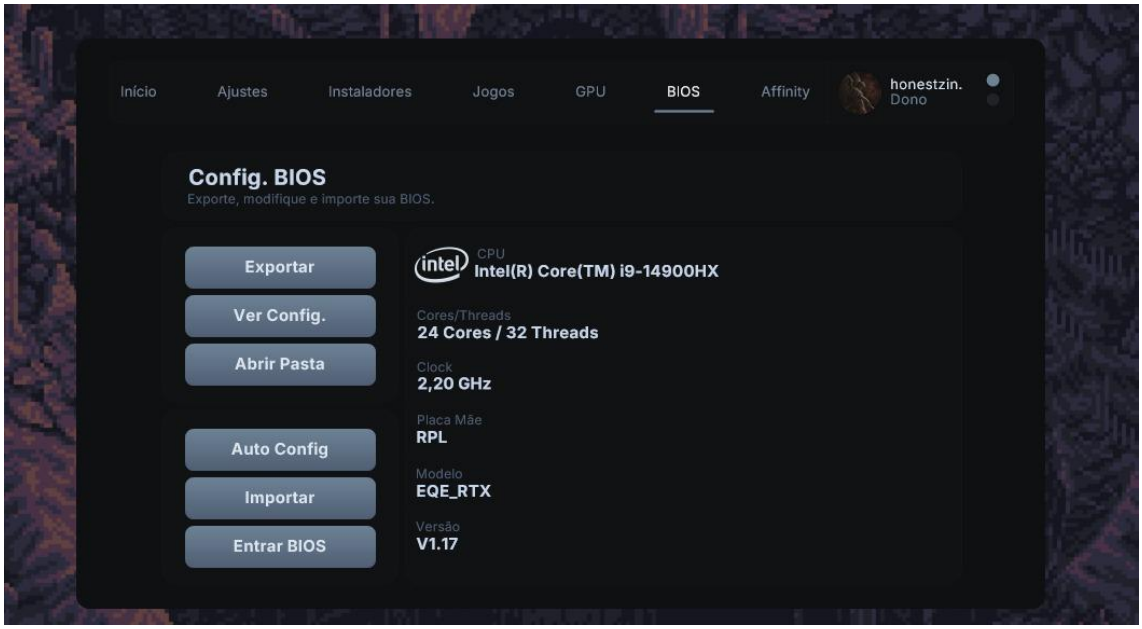
Figura 5 - Área para habilitar os ajustes no Windows



Fonte: Autoria própria (2026).

Na Figura 5 é possível verificar o painel de Instaladores do Honet. Nele estão mapeados todos os possíveis sistemas e utilitários do Sistema. Além disso também é possível identificar todos os drivers e arquivos de inicialização de programas.

Figura 6 - Área para habilitar os ajustes no firmware



Fonte: Autoria própria (2026).

Na Figura 6 é apresentado o campo de BIOS do Honest, em que é possível identificar todas as informações da máquina e, também, configurar a BIOS do sistema com a opção de acessar a BIOS ou auto configurar, dentre outras funcionalidades.

4 ANALISE DOS DADOS

Os testes de desempenho e validação das otimizações foram realizados em dois computadores com diferentes configurações de hardware, de modo a avaliar o comportamento e a eficácia do software proposto em contextos distintos de capacidade de processamento. A escolha desses equipamentos teve como objetivo representar cenários de uso real, comparando um sistema de alto desempenho com outro intermediário, permitindo observar a escalabilidade, estabilidade e possíveis limitações das otimizações automatizadas aplicadas.

O Computador 1 corresponde a um desktop de maior robustez, voltado para tarefas exigentes e com alto potencial de processamento gráfico. É equipado com processador AMD Ryzen 5 5600X, placa de vídeo NVIDIA RTX 2070 SUPER e 16 GB de memória RAM DDR4 a 3200 MHz, executando o Windows 11 Pro (versão 23H2) com *firmware* UEFI 2.9.

O Computador 2 é um notebook com configuração intermediária, representando um ambiente de hardware mais limitado e com maior restrição térmica, utilizando um Intel Core i5-12500H, placa de vídeo NVIDIA RTX 3050, 16 GB de RAM DDR4, e Windows 11 Home (versão 23H2) com *firmware* UEFI 2.7.

Através do Windows, as otimizações aplicadas incluíram *lopagelocklimit*, que se refere ao registro do Windows, o Auto Debloat, para desabilitar serviços em segundo plano, e a configuração FSE (*Full Screen Exclusive*) para os *drivers* de vídeos. Já nas configurações via firmware, foi utilizada XMP (*Extreme Memory Profile*) e *Hyper Threading.*

4.1 PROCEDIMENTO DE COLETA DE DADOS

Para a coleta de dados foram utilizados procedimentos de *benchmarking* comparativo e monitoramento de desempenho geral do computador, com o intuito de mensurar o impacto direto das modificações realizadas pelo *software* no Windows e no *firmware* das máquinas testadas.

Para o monitoramento geral do desempenho do sistema, foi realizado *benchmark* do *Desktop Window Manager* (DWM), processo nativo do Windows responsável pela renderização da interface gráfica e pela gestão de janelas e elementos visuais do sistema operacional, sendo avaliado através de FPS (*Frame Per Second*).

A análise do DWM foi adotada por representar o comportamento global do computador, permitindo mensurar de forma consistente o impacto das otimizações aplicadas sobre o desempenho geral do sistema, sem depender exclusivamente de cargas de trabalho externas. Assim, avaliando quantos quadros por segundo o sistema consegue renderizar na interface gráfica, conseguimos identificar, de modo geral, a fluidez da interface do Windows (menor travamento ou *delay visual*); a eficiência do sistema na renderização de janelas e animações e, de forma indireta, a capacidade de resposta da GPU e do sistema em tarefas gráficas gerais.

Os testes de *benchmark* foram realizados através das plataformas CapFrameX e OCAT (*Open Capture and Analytics Tool*), voltadas ao desempenho gráfico e estabilidade. O OCAT foi utilizado para capturar métricas precisas de *frames* por segundo (FPS), tempo de frame (*frametime*) e latência de renderização, enquanto o CapFrameX fornece a análise detalhada desses dados através de gráficos comparativos, que inclui médias, desvios padrão e percentis (1% e 0.2% *lows*). Essas aplicações nos leva a melhor avaliação da consistência da taxa de quadros e a fluidez geral, permitindo notar a estabilidade e a eficiência após a aplicação das otimizações.

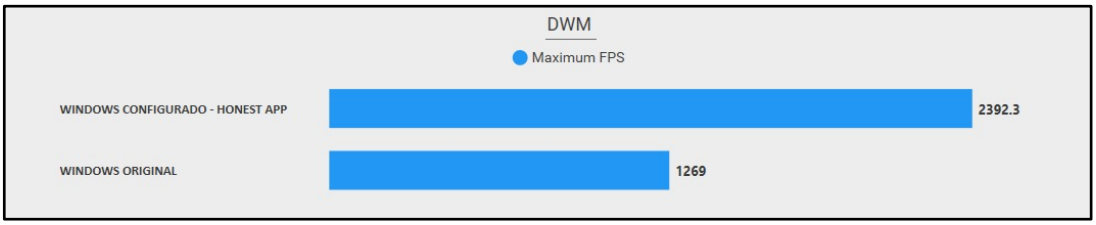
Os dados dos resultados foram registrados por meio de capturas de tela dos *benchmarks*, assegurando a fidelidade da coleta e a comparação entre os cenários antes e depois da aplicação das otimizações. Cada teste foi repetido 3 vezes para reduzir variabilidade. Esses registros

permitiram identificar visualmente o ganho percentual de desempenho e o impacto das configurações sobre o comportamento térmico e energético dos sistemas, validando a eficácia da ferramenta Honest quanto à melhoria de performance e estabilidade operacional.

5 RESULTADOS, LIMITAÇÕES E TENDÊNCIAS FUTURAS

Na Figura 7 e Figura 8 são apresentados os resultados obtidos nos testes de *benchmark* do DWM (*Desktop Window Manager*), responsável pela renderização das janelas, efeitos visuais e gerenciamento gráfico da interface do Windows. Esse processo impacta diretamente na fluidez e estabilidade do sistema, principalmente em tarefas que envolvem o uso intensivo da GPU, como jogos eletrônicos ou aplicações 3D.

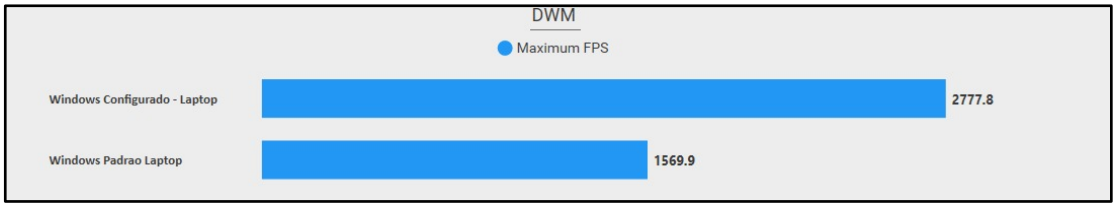
Figura 7 - Benchmark DWM do Computador 1



Fonte: Autoria própria (2026).

No Computador 1, o sistema configurado com o software Honest apresentou um aumento expressivo no desempenho, elevando o valor máximo de FPS de 1269 para 2392,3, representando uma melhora significativa na resposta gráfica e na estabilidade visual.

Figura 8 - Benchmark DWM do Computador 2



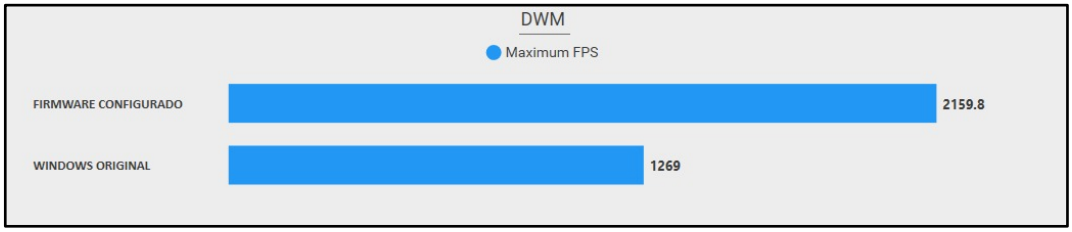
Fonte: Autoria própria (2026)..

De forma semelhante, no Computador 2, o FPS máximo subiu de 1569,9 para 2777,8 após a aplicação das otimizações automatizadas.

5.1 EXECUÇÃO DO SISTEMA COM CONFIGURAÇÕES DE FIRMWARE (UEFI/BIOS)

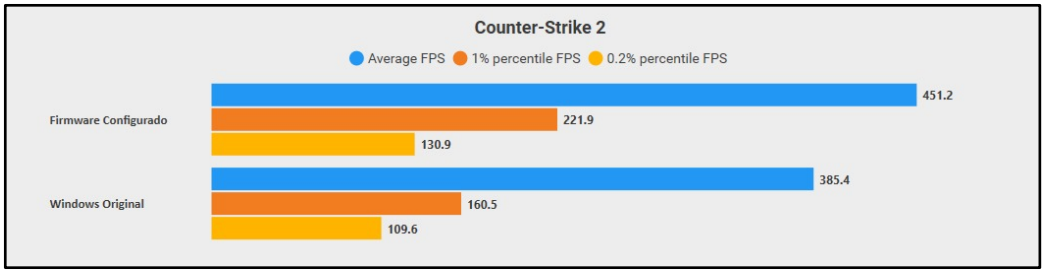
Os testes envolvendo configurações de *firmware* foram realizados exclusivamente no Computador 1, uma vez que o Computador 2, por se tratar de um notebook com sistema de resfriamento limitado, não suportaria ajustes avançados sem risco de superaquecimento.

Figura 9 - Benchmark DWM após configurações no firmware no Computador 1



Fonte: Autoria própria (2026).

Figura 10 - Benchmark FPS através do jogo eletrônico Counter-Strike 2 após configurações do firmware no Computador 1



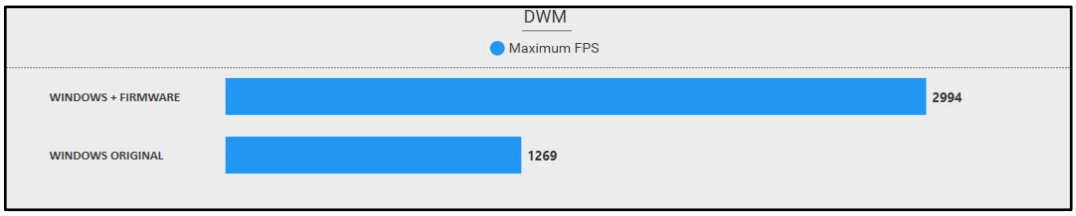
Fonte: Autoria própria (2026).

Na Figura 9 apresenta o resultado do benchmark DWM (Desktop Window Manager), no qual foi possível observar um aumento no FPS máximo de 1269 para 2159,8. Já na Figura 10, o teste foi realizado no jogo Counter-Strike 2, utilizando a mesma máquina. Após as otimizações aplicadas no firmware, o FPS médio subiu de 385,4 para 451,2.

5.2 DESEMPENHO COMBINADO: SISTEMA OPERACIONAL E *FIRMWARE*

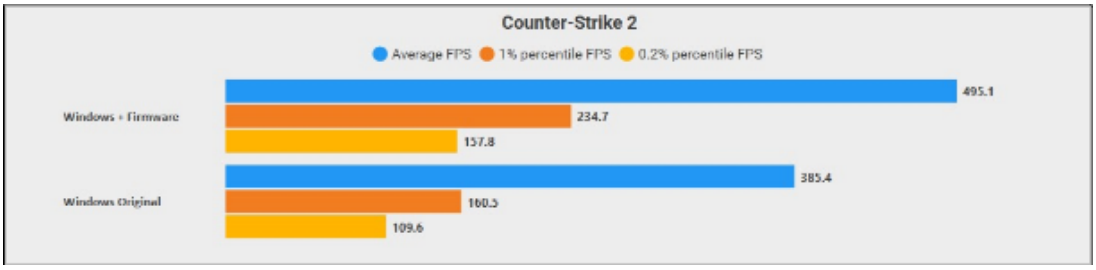
Com o objetivo de avaliar o impacto conjunto das configurações do Sistema Operacional e *Firmware* no desempenho geral da máquina, observa-se na Figura 11 que, após a aplicação dos ajustes em ambos os ambientes, o desempenho medido pelo DWM (*Desktop Window Manager*) apresentou um aumento expressivo, atingindo 2994 FPS em comparação aos 1269 FPS registrados na configuração original do sistema.

Figura 11 - Benchmark DWM após configurações do Windows e Firmware no Computador 1



Fonte: Autoria própria (2026).

Figura 12 - Benchmark FPS através do jogo eletrônico Counter-Strike 2 após configurações do Windows e Firmware no Computador 1



Fonte: Autoria própria (2026).

Na Figura 12, o teste de benchmark realizado no jogo Counter-Strike 2 através do Computador 1, saiu de 385,4 para 495,1 FPS.

5.3 ANÁLISE COMPARATIVA E INTERPRETAÇÃO DOS RESULTADOS

Os ajustes implementados pelo *software* Honest demonstraram impactos expressivos no desempenho do sistema, sobretudo em cenários que demandam maior processamento gráfico. De acordo com os resultados apresentados na Tabela 1, o Computador 1 registrou um aumento de 88,5% no DWM quando apenas as otimizações do sistema operacional foram aplicadas, elevando a taxa de 1269 FPS para 2392,3 FPS.

Na configuração em que somente o *firmware* foi otimizado, observou-se um incremento de 70,2%, alcançando 2159,8 FPS. A combinação dos ajustes no Windows somados às configurações do *firmware* resultou no melhor desempenho, atingindo 2994 FPS, valor que corresponde a um ganho total de 136% em comparação ao estado inicial da máquina.

Tabela 1 - DWM no Computador 1

Configuração	FPS Antes	FPS Depois	Ganhos em %
Windows Otimizado	1269	2392,3	+88,5%
Firmware Otimizado	1269	2159,8	+70,2%
Windows e Firmware Otimizados	1269	2994	+136%

Fonte: Autoria própria (2026).

Os testes realizados no jogo *Counter-Strike 2* no Computador 1 também evidenciaram ganhos relevantes após a aplicação das otimizações. Como apresentado na Tabela 2, apenas com os ajustes no sistema operacional, o FPS médio subiu de 385,4 para 451,2, representando um incremento aproximado de 17%.

O comportamento em cenários de maior exigência gráfica também melhorou de forma consistente. De acordo com a Tabela 3, o valor do percentil 1% aumentou de 160,5 para 221,9 (38%), enquanto o percentil 0,2% passou de 109,6 para 130,9 (19%). Esses resultados apontam para uma redução nas quedas severas de desempenho e para uma experiência de jogo mais estável.

Quando as configurações foram integradas entre o Windows e o *firmware*, houve um aumento ainda maior no desempenho do Computador 1. O FPS médio subiu para 495,1, representando um acréscimo de 28,4% em relação ao sistema original. O FPS do percentil 1% atingiu 234,7, um acréscimo de 46,2%, enquanto o do percentil 0,2% chegou a 157,8, equivalente a um ganho de 43,9%.

Tabela 2 - FPS Médio Counter Strike 2 no Computador 1

Configuração	FPS Médio Antes	FPS Médio Depois	Ganhos em %
Firmware Otimizado	385,4	451,2	+17%
Windows e *Firmware* Otimizados	385,4	495,1	+28,4%

Fonte: Autoria própria (2026).

Tabela 3 - Percentis Counter Strike 2 no Computador 1

Configuração	FPS 1% Percentil Antes	FPS 1% Percentil Depois	Ganhos em %	FPS 0.2% Percentil Antes	FPS 0.2% Percentil Depois	Ganhos em %
Firmware Otimizado	160,5	221,9	+38%	109,6	130,9	+19%
Windows e *Firmware* Otimizados	160,5	234,7	+46,2%	109,6	157,8	+43,9%

Fonte: Autoria própria (2026).

O Computador 2, de configuração intermediária, também apresentou resultados relevantes, na qual foi apenas realizada a configuração de otimização no Windows. Visualizado na Tabela 4, houve aumento de 77% no desempenho do DWM, passando de 1569,9 FPS para 2777,8 FPS. Os testes envolvendo os ajustes no *firmware* não foram executados neste equipamento devido às suas limitações térmicas e de resfriamento, que poderiam comprometer sua integridade.

Tabela 4 - DWM Computador 2

Configuração	FPS Antes	FPS Depois	Ganhos em %
Windows Otimizado	1569,9	2777.8	+77%

Fonte: Autoria própria (2026).

A partir desses resultados, podemos observar que houve consistência em relação à estabilidade, com baixa variação entre os valores de FPS obtidos nos diferentes testes, além do aumento dos percentis mais baixos, o que indica diminuição de microtravamentos e oscilações, reforçando que as melhorias estão diretamente relacionadas às configurações aplicadas, tornando o computador mais fluido e otimizado.

Os resultados obtidos nos comprovam que é possível alcançar um ambiente de melhor performance nos computadores através de um *software* de otimização automatizado como o Honest, seja sendo ela aplicada em máquinas de configuração intermediária ou mais robusta. A democratização do acesso á tais configurações, muito delas arriscadas, leva a melhor praticidade principalmente para aqueles usuários mais leigos.

A junção dos ajustes do Windows e do *firmware* oferecida pelo Honest, através dos testes realizados é observado ganhos significativos de performance no desempenho geral do computador (DWM) e na execução de jogos eletrônicos. Através dos resultados, que nos

mostram o aumento notório da taxa de quadros por segundo (FPS) após as configurações, comprovam que a automação desses ajustes foi capaz de reduzir gargalos de desempenho e melhorar a comunicação entre o *hardware* e o Windows. Estes ganhos corroboram os resultados documentados por Qi *et al*., [2022], que demonstraram melhorias de até 43,7% através de otimização automatizada de mais de 100 parâmetros de *firmware* em ambientes de servidor, validando que a automação inteligente de configurações de baixo nível produz grande impacto independente do ambiente, seja ele em servidor ou em computador pessoal.

Ademais, além dos ganhos de desempenho, constatou-se que o sistema apresentou comportamento estável durante os testes, sem falhas críticas ou interrupções, demonstrando assim o benefício dos modelos de segurança DevSecOps e SSDLC adotados no desenvolvimento do Honest, alinhados aos princípios de engenharia de *software* modernos propostos por Pressman & Maxim (2021). Estes resultados reforçam que a aplicação de rotinas automatizadas reduz significativamente a dependência da intervenção manual e minimiza o risco de configurações incorretas, aspecto já documentado por Lowe *et al.* (2022), que demonstraram através de 63 estudos de caso que intervenções automatizadas resultam em reduções expressivas de riscos e aumentos de produtividade. A automação, portanto, mostrou-se não apenas eficiente em termos de desempenho, mas também segura quanto à integridade do sistema e do usuário.

Por outro lado, foi observada uma limitação no Computador 2, em que ajustes de *firmware* não puderam ser aplicados devido à restrição térmica do equipamento por se tratar de um notebook, o que nos mostra que essas otimizações dependem diretamente das condições físicas do *hardware*, destacando a necessidade de estudos mais aprofundados para aprimorar a adaptação da ferramenta a diferentes cenários.

O Honest cumpre-se assim com os objetivos propostos ao trazer um sistema de otimização que realiza uma integração funcional entre o desempenho, segurança e acessibilidade, através da junção de configurações do Windows e do *firmware*, representando um avanço prático no campo da automação de otimizações computacionais.

6 CONSIDERAÇÕES FINAIS

Durante o desenvolvimento e a execução dos testes, algumas limitações foram identificadas que influenciam o alcance e a aplicabilidade do *software* Honest. A principal delas está relacionada às configurações de *firmware*, que, embora proporcionem ganhos

significativos de desempenho em máquinas com bom sistema de refrigeração, não puderam ser aplicadas em dispositivos com *hardware* mais limitado, como o notebook utilizado nos testes (Intel i5-12500H e RTX 3050). Nesse tipo de equipamento, como comentado por Posey [2021], ajustes no *firmware* podem causar superaquecimento, instabilidade e até desligamentos automáticos do sistema, devido à menor capacidade térmica e energética do dispositivo.

Podemos destacar também, que o Honest concentra suas otimizações apenas em sistemas operacionais Windows, não podendo realizar o uso da ferramenta em outros ambientes, como Linux. Outro ponto que se destaca é que, apesar dos resultados positivos, o aumento de desempenho pode variar conforme o perfil de uso e o tipo de *hardware*, não havendo um ganho uniforme entre diferentes máquinas.

Essas limitações não comprometem a relevância dos resultados obtidos, mas indicam oportunidades de aprimoramento para versões futuras, com foco em ampliar sua compatibilidade, segurança em diferentes contextos de uso. Uma das principais frentes de avanço é a adaptação do sistema para outros sistemas operacionais, como Linux e versões anteriores do Windows, de modo a ampliar o público-alvo e permitir comparações diretas entre ambientes distintos de gerenciamento de desempenho.

Outro ponto de desenvolvimento futuro está relacionado à automação segura de configurações no *firmware*, através da criação de um módulo que interprete de forma inteligente as limitações de cada *hardware* antes de aplicar ajustes nesse ambiente, visando prevenir superaquecimento e instabilidade, o que garante maior compatibilidade com sistemas menos robustos.

Além disso, recomenda-se a execução de novos testes práticos com uma gama mais ampla de *hardwares*, incluindo diferentes modelos de processadores, placas-mãe e GPUs, de fabricantes variados. Isso permitirá validar com maior precisão a escalabilidade das otimizações e ajustar parâmetros conforme o perfil de desempenho de cada equipamento.

A integração de um sistema de relatórios automáticos e comparação de métricas entre plataformas poderá contribuir para análises mais detalhadas e consistentes, fortalecendo o potencial do Honest como uma ferramenta universal de otimização e avaliação de eficiência computacional.

Desta forma, os resultados apresentados pelo Honest comprovaram que a otimização de computadores por meio da integração de ajustes do Windows e do *firmware* levou a um melhor desempenho dos hardwares testados, oferecendo maior fluidez e estabilidade, especialmente em ambientes que exigem alta capacidade de processamento, como jogos eletrônicos e sistemas instalados em máquinas com limitações de desempenho. A automação proposta simplificou

procedimentos tradicionalmente complexos, reduzindo etapas manuais e permitindo uma aplicação mais rápida e eficiente das otimizações. Além disso, a adoção de práticas de segurança como DevSecOps e SSDLC garantiu a integridade das operações realizadas, evidenciando a importância de estabelecer mecanismos de proteção desde as fases iniciais do desenvolvimento do software.

O Honest representa um marco de socialização da automação de desempenho computacional, tornando ajustes avançados acessíveis, seguros e confiáveis para usuários de diferentes níveis técnicos. Além de unificar em uma única ferramenta ajustes que antes exigiam processos fragmentados e manuais, o *software* amplia o entendimento sobre como interações entre sistema operacional e *firmware* podem gerar ganhos reais de *performance.*

REFERÊNCIAS

AMAZON WEB SERVICES. **O que é *DevSecOps?*** AWS, 2025. Disponível em: https://aws.amazon.com/pt/what-is/devsecops/. Acesso em: 10 maio 2026.

AMAZON WEB SERVICES. **O que é o *SDLC?*** AWS, 2025. Disponível em: https://aws.amazon.com/pt/what-is/sdlc/. Acesso em: 10 mai. 2026.

DELL. **Como melhorar o desempenho do seu computador *Dell* com *Windows* 11 ou *Windows* 10.** Dell Suporte, 2024. Disponível em: https://www.dell.com/support/kbdoc/pt-br/000192255/como-melhorar-o-desempenho-do-seu-computador-dell-com-windows-11-ou-windows-10. Acesso em: 10 maio 2026.

FLICK, U. **Introdução à pesquisa qualitativa e quantitativa.** 5. ed. Porto Alegre: Artmed, 2019.

FLINDERS, Mesh. ***What is firmware? IBM Think,*** 11 set. 2024. Disponível em: https://www.ibm.com/think/topics/firmware. Acesso em: 10 maio 2026.

GIL, A. C. **Como elaborar projetos de pesquisa.** 6. ed. São Paulo: Atlas, 2017.
HP BRASIL. O que é UEFI? Entenda o novo firmware para computadores. ***HP Tech Takes,*** 24 dez. 2024. Disponível em: https://www.hp.com/br-pt/shop/tech-takes/o-que-e-uefi. Acesso em: 10 maio 2026.

INTEL CORPORATION. **O que é *Hyper-Threading?*** Intel Brasil, 2025. Disponível em: https://www.intel.com.br/content/www/br/pt/gaming/resources/hyper-threading.html. Acesso em: 10 maio 2026.

INTEL CORPORATION. ***What is Intel® Turbo Boost Technology?*** Intel, 5 jul. 2025. Disponível em: https://www.intel.com/content/www/us/en/gaming/resources/turbo-boost.html. Acesso em: 10 maio 2026.

KACPERSKI, Kamil. ***Beyond the legacy:*** *embracing UEFI platform firmware.* CodiLime Blog, 1 ago. 2024. Disponível em: https://codilime.com/blog/beyond-the-legacy-embracing-uefi-platform-firmware/. Acesso em: 10 maio 2026.

KANDWAL, Vaibhav. UEFI vs BIOS – What's the Difference? freeCodeCamp, 2022. Disponível em: https://www.freecodecamp.org/news/uefi-vs-bios/. Acesso em: 10 maio 2026.

KERBER, Diego. **Milagres do Intel APO:** 20% de ganho só via otimizações? Adrenaline, 18 dez. 2023. Disponível em: https://www.adrenaline.com.br/artigos/milagres-do-intel-apo-20-de-ganho-so-via-otimizacoes/. Acesso em: 10 maio 2026.

KHULLAR, Kunal**.** ***How to Enable XMP to Improve RAM Speeds. Tom's Hardware****,* 14 dez. 2024. Disponível em: https://www.tomshardware.com/pc-components/ram/how-to-enable-xmp-to-improve-ram-speeds. Acesso em: 10 maio 2026.

KROTH, Brian; MATUSEVYCH, Sérgio; ZHU, Yiwen. ***Autotuning:*** *Systems, Techniques, Challenges, and Opportunities.* Microsoft, 2025. Disponível em: https://www.microsoft.com/en-us/research/publication/autotuning-systems-techniques-challenges-and-opportunities/. Acesso em: 10 maio 2026.

LIU, Z. ***Intel promises to keep updating its game-boosting software despite recent delays:*** *APO will only focus on current and upcoming Intel CPUs*. Disponível em: https://www.tomshardware.com/pc-components/cpus/intel-promises-to-keep-updating-its-game-boosting-software-despite-recent-delays-apo-will-only-focus-on-current-and-upcoming-intel-cpus?. Acesso em: 10 maio 2026.

LOWE, Brian D.; HAYDEN, Maria; ALBERS, James; NABER, Steven. Case Studies of Robots and Automation as Health/Safety Interventions in Small Manufacturing Enterprises. ***International Journal of Environmental Research and Public Health***, v. 20, n. 9, p. 5667, 2022. DOI: 10.3390/ijerph20095667. Disponível em: https://pubmed.ncbi.nlm.nih.gov/37206917/. 10 maio 2026.

MICROSOFT. **Windows Server 2022 Performance Tuning Guidelines**. Microsoft Learn, 5 jul. 2022. Disponível em: https://learn.microsoft.com/en-us/windows-server/administration/performance-tuning/. 10 maio 2026.

NOLETO, Caio. **O que é Sistema Operacional?** O guia completo! Blog da Trybe, 02 ago. 2022. Disponível em: https://blog.betrybe.com/tecnologia/sistema-operacional-tudo-sobre/. 10 maio 2026.

OLAES, Terry. ***What is security automation?*** Balbix, 16 jan. 2025. Disponível em: https://www.balbix.com/insights/what-is-security-automation/. Acesso em: 10 maio 2026.

POSEY, Brien. Ajuste essas 4 configurações de BIOS do servidor para melhorar o desempenho. ***Computer Weekly***, 16 ago. 2021. Disponível em: https://www.computerweekly.com/br/tip/Ajuste-essas-4-configuracoes-de-BIOS-do-servidor-para-melhorar-o-desempenho. Acesso em: 10 maio 2026.

PRESSMAN, R. S.; MAXIM, B. R. ***Software Engineering:*** A Practitioner's Approach. 9. ed. Nova York: McGraw-Hill, 2021. p. 320-321.

QI, Xianxian; YANG, Jianfeng; ZHANG, Yiyang; XIAO, Baonan. ***BIOS-Based Server Intelligent Optimization,*** 6 set. 2022. Disponível em: https://pmc.ncbi.nlm.nih.gov/articles/PMC9504261/. Acesso em: 10 maio 2026.

REZENDE, G. H. *et al*. Benchmark review and case study of stress test for a gaming computer applied in CPU. *In*: SIMPÓSIO BRASILEIRO DE TESTES DE SOFTWARE SISTEMÁTICO E AUTOMATIZADO (SAST), 9., 2024, Curitiba. **Anais** [...]. Porto Alegre: Sociedade Brasileira de Computação, 2024. p. 49-55. Disponível em: https://doi.org/10.5753/sast.2024.3848. Acesso em: 10 maio 2026.

TAAHA MUFFASIL. ***One-Click Performance: How OMEN AI is Revolutionizing Gaming Optimization.*** Disponível em: https://www.hp.com/us-en/shop/tech-takes/omen-ai-gaming-optimization?. Acesso em: 10 maio 2026.

UEFI, **Unified Extensible Firmware Interface Forum**. *UEFI FAQs*. UEFI.org, 2025. Disponível em: https://uefi.org/faq. Acesso em: 10 maio 2026.

VMP SOFTWARE. *VMProtect user manual*. ***VMPSoft***, 2025. Disponível em: https://vmpsoft.com/vmprotect/user-manual/. Acesso em: 10 maio 2026.

MINI CURRÍCULO E CONTRIBUIÇÕES AUTORES

TITULO DO ARTIGO	**HONEST:** FERRAMENTA AUTOMATIZADA DE OTIMIZAÇÃO INTEGRADA DE WINDOWS E FIRMWARE PARA DESEMPENHO DE COMPUTADORES
RECEBIDO	10/05/2026
AVALIADO	02/06/2026
ACEITO	09/06/2026

AUTOR 1	
PRONOME DE TRATAMENTO	Sr.
NOME COMPLETO	Danilo Souza Barreto
INSTITUIÇÃO/AFILIAÇÃO	Centro Universitário Jorge Amado - Unijorge
CIDADE	Salvador
ESTADO	Bahia
PAÍS	Brasil
RESUMO DA BIOGRAFIA	Cientista da Computação, egresso da Unijorge
AUTOR 2	
PRONOME DE TRATAMENTO	Sr.
NOME COMPLETO	José Eduardo de Azevedo Ribeiro
INSTITUIÇÃO/AFILIAÇÃO	Centro Universitário Jorge Amado - Unijorge
CIDADE	Salvador
ESTADO	Bahia
PAÍS	Brasil
RESUMO DA BIOGRAFIA	Cientista da Computação, egresso da Unijorge
AUTOR 3	
PRONOME DE TRATAMENTO	Sr.
NOME COMPLETO	Marcos Santos Leite
INSTITUIÇÃO/AFILIAÇÃO	Centro Universitário Jorge Amado - Unijorge
CIDADE	Salvador
ESTADO	Bahia
PAÍS	Brasil
RESUMO DA BIOGRAFIA	Especialização em Educação a Distância, pela faculdade - Serviço Nacional de Aprendizagem Comercial Brasil (SENAC); Especialização em Educação Profissional e Tecnológica pelo Instituto Federal da Bahia (IFBA); Pós-graduação em Ciências de Dados e Inteligência Artificial pela FASUL Educacional; MBA em Gestão de Projetos pela FASUL Educacional; MBA em Empreendedorismo e Inovação pela FASUL Educacional. Graduado em Administração com Habilitação em Análises de Sistemas pela Fundação Visconde de Cairu; Graduado em Análise e Desenvolvimento de Sistemas, pelo Centro Universitário Jorge Amado (Unijorge); Licenciado em Educação Profissional e Tecnológica pelo Instituto Federal da Bahia (IFBA).

AUTOR 4	
PRONOME DE TRATAMENTO	Sr.
NOME COMPLETO	Celso Barreto da Silva
INSTITUIÇÃO/AFILIAÇÃO	Centro Universitário Jorge Amado - Unijorge
CIDADE	Salvador
ESTADO	Bahia
PAÍS	Brasil
RESUMO DA BIOGRAFIA	Doutorando em Difusão do Conhecimento pelo Instituto Federal da Bahia (IFBA); Mestrado em Sistemas e Computação pela UNIFACS; Graduado em Administração com Habilitação em Análise de Sistemas pela Faculdade Visconde de Cairu; Licenciado em Matemática para o Ensino Básico pela Faculdade Educacional da LAPA FAEL. Graduado em Análise e Desenvolvimento de Sistemas pelo Centro Universitário Jorge Amado (Unijorge); Pós-graduação em Metodologia do Ensino da Matemática e Docência no Ensino Superior pelo Centro Universitário Leonardo Da Vinci.
AUTOR 5	
PRONOME DE TRATAMENTO	Sr.
NOME COMPLETO	Fabio Fonseca Barbosa Gomes
INSTITUIÇÃO/AFILIAÇÃO	Centro Universitário Jorge Amado - Unijorge
CIDADE	Salvador
ESTADO	Bahia
PAÍS	Brasil
RESUMO DA BIOGRAFIA	Mestrado em Sistemas e Computação pela Universidade Salvador; Graduação em Bacharelado em Sistemas de Informação pelo Centro Universitário Estácio da Bahia. Atualmente é professor assistente no Senai Cimatec e na Escola Bahiana de Medicina e Saúde Pública. Além disso, é professor universitário e coordenador do Centro Universitário Jorge Amado. Tem experiência na área de Ciência da Computação, com ênfase em Redes de Computadores, atuando principalmente nos seguintes temas: Saúde Digital, Ensino à Distância, Sistemas de Informação, Automação Residencial, Computação Ubíqua e Pervasiva e Redes de Computadores.
CONTRIBUIÇÃO DO(S) AUTOR(ES) NO ARTIGO	Todos os autores contribuíram na mesma proporção.

Endereço de Correspondência dos autores	**Autor 1:** danilobarreto93@hotmail.com **Autor 2:** eduazevedo.contato@gmail.com **Autor 3:** profmarcosleite@hotmail.com **Autor 4:** profcelsobarreto@hotmail.com **Autor 5:** fabiofbg@gmail.com