# Engenharia de Software sob Restrições de Hardware: Uma Análise Técnica do Motor de Wolfenstein 3D

Arthur Pereira, Heitor Rodrigues, Matheus Moura,
Samuel Paula e Thomás Kriger

8 de abril de 2026

**Resumo**

Este artigo analisa os fundamentos técnicos e os desafios de hardware superados no desenvolvimento de *Wolfenstein 3D* (1992), baseando-se nos capítulos iniciais da obra *Game Engine Black Book*, de Fabien Sanglard. O estudo explora a transição do PC de uma máquina de escritório para uma plataforma de jogos de alta performance, focando na arquitetura do processador Intel 80386 e na ausência de unidades de ponto flutuante (FPU). Discute-se como a engenhosidade da id Software, através do uso de aritmética de ponto fixo e da técnica de *raycasting*, permitiu a criação de um ambiente pseudo-3D fluido em um hardware hostil. A análise conclui que o sucesso do título derivou da subversão das limitações físicas através de otimizações matemáticas e de software.

## Sumário

# 1 Introdução

O lançamento de *Wolfenstein 3D* em 1992 pela id Software não representou apenas o nascimento do gênero *First-Person Shooter* (FPS) moderno, mas também um marco de engenharia de software. Diferente do desenvolvimento contemporâneo, onde há abundância de recursos e aceleração gráfica dedicada (GPUs), a era do início dos anos 90 exigia que os programadores compreendessem intimamente as entranhas do hardware.

Este artigo, fundamentado na análise técnica de Fabien Sanglard, explora como o motor do jogo (*engine*) foi projetado para operar em uma arquitetura de computador pessoal (PC) originalmente destinada a tarefas de produtividade e escritório. O objetivo é detalhar os obstáculos técnicos — como a baixa largura de banda de vídeo e a falta de suporte nativo a cálculos decimais — e as soluções algorítmicas que permitiram a ilusão da tridimensionalidade em tempo real.

# 2 Contexto Tecnológico: O PC vs. Consoles

Uma das premissas fundamentais da obra de Sanglard é a justificativa da escolha do PC como plataforma, em detrimento dos consoles dominantes na época, como o Super Nintendo e o Mega Drive.

Enquanto os consoles possuíam hardware especializado para manipular *sprites* e realizar *scrolling* lateral em 2D, eles careciam de poder de processamento bruto para cálculos geométricos complexos. O PC, por outro lado, embora desprovido de aceleração gráfica, possuía uma Unidade Central de Processamento (CPU) significativamente mais rápida. O processador Intel 386 oferecia a flexibilidade necessária para que o motor do jogo assumisse o papel que hoje cabe às GPUs: o cálculo pixel a pixel do cenário através do acesso direto ao *framebuffer* no modo VGA.

# 3 Análise do Hardware e Limitações

## 3.1 A Arquitetura do Processador Intel 80386

O motor de *Wolfenstein 3D* foi otimizado para a arquitetura de 32 bits do Intel 80386. No entanto, o mercado era fragmentado entre as versões SX e DX.

- **386-SX:** Uma versão de baixo custo com barramento externo de 16 bits, criando um gargalo na transferência de dados.
- **386-DX:** A versão completa com barramento de 32 bits, permitindo maior performance.

A equipe da id Software precisou garantir que o pipeline de renderização fosse eficiente o suficiente para ser executado mesmo no modelo SX, onde a comunicação entre a CPU e a memória RAM era limitada.

## 3.2 O Problema da Ausência de FPU

O maior desafio técnico identificado no Capítulo 2 é a inexistência de uma *Floating Point Unit* (FPU) na maioria dos computadores domésticos de 1991. O cálculo de senos, cossenos e distâncias euclidianas, essenciais para a perspectiva 3D, exige o uso de números decimais. Sem hardware dedicado, o processador precisava emular números de ponto flutuante via software, o que resultava em uma perda de performance de até dez vezes, tornando o jogo inviável.

# 4 Metodologia e Soluções de Engenharia

## 4.1 A Técnica de Raycasting

Para contornar a incapacidade de processar polígonos reais, John Carmack implementou o *Raycasting*. Em vez de projetar um mundo 3D complexo, o motor dispara "raios"horizontais a partir da posição do jogador. Quando um raio atinge uma parede, a distância é calculada. A engine então desenha uma coluna vertical de pixels cuja altura é inversamente proporcional à distância encontrada, criando a ilusão de profundidade com um custo computacional reduzido.

## 4.2 Aritmética de Ponto Fixo (Fixed-Point)

A solução definitiva para o gargalo da CPU foi a substituição de números reais por aritmética de ponto fixo. Em vez de utilizar o padrão IEEE 754 para decimais, o jogo tratava todos os valores como inteiros de 32 bits. Nesse sistema, os primeiros 16 bits representam a parte inteira e os 16 bits finais simulam a parte fracionária (formato 16:16). Isso permitiu que o motor utilizasse a Unidade Lógica e Aritmética (ALU) — a parte mais rápida da CPU — para realizar cálculos trigonométricos, garantindo que o jogo mantivesse uma taxa de quadros aceitável mesmo em máquinas modestas.

# 5 Conclusão

*Wolfenstein 3D* é um testemunho da capacidade da engenharia de software de superar limites físicos hostis. A análise dos capítulos iniciais de Sanglard revela que a "mágica"do 3D em 1992 não foi fruto de um hardware avançado, mas de uma subversão inteligente da arquitetura existente.

O uso do *Raycasting* aliado à aritmética de ponto fixo permitiu transformar uma máquina de escritório em uma poderosa plataforma de entretenimento. Conclui-se que o legado deste motor reside na lição de que a otimização matemática e o conhecimento profundo do hardware são ferramentas tão poderosas quanto a potência bruta de processamento, servindo de inspiração para a engenharia de sistemas modernos.