

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**EVERTON SOUZA WIENCE**

**AVALIAÇÃO DE DESEMPENHO DO SERVIDOR NGINX EM CONFIGURAÇÃO  
PADRÃO E COMO PROXY REVERSO COM CACHE EM APLICAÇÕES WEB**

**GUARAPUAVA**

**2025**

**EVERTON SOUZA WIENCE**

**AVALIAÇÃO DE DESEMPENHO DO SERVIDOR NGINX EM CONFIGURAÇÃO  
PADRÃO E COMO PROXY REVERSO COM CACHE EM APLICAÇÕES WEB**

**Performance evaluation of NGINX server in default configuration and as  
reserve proxy for web applications**

Proposta de Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Tecnólogo em Sistemas para Internet do Curso de Tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná.

Orientador: Prof<sup>a</sup>. Dr<sup>a</sup>. Sediane Carmen Lunardi Hernandes

Coorientador: Prof. Dr. Hermano Pereira

**GUARAPUAVA**

**2025**



Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

[4.0 Internacional](#)

## **LISTA DE FIGURAS**

<b>Figura 1 – Ilustração de um servidor web (FOROUZAN; MOSHARRAF, 2013).</b>	<b>6</b>
<b>Figura 2 – Arquitetura de um servidor <i>proxy</i> (SÁ; SOUZA, 2015).</b>	<b>7</b>
<b>Figura 3 – Arquitetura de um <i>proxy</i> reverso (SÁ; SOUZA, 2015).</b>	<b>7</b>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>3</b>
<b>1.1</b>	<b>Objetivos . . . . .</b>	<b>3</b>
1.1.1	Objetivo geral . . . . .	3
1.1.2	Objetivos específicos . . . . .	4
<b>1.2</b>	<b>Justificativa . . . . .</b>	<b>4</b>
<b>1.3</b>	<b>Estrutura do trabalho . . . . .</b>	<b>5</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO . . . . .</b>	<b>6</b>
<b>2.1</b>	<b>Servidores web . . . . .</b>	<b>6</b>
2.1.1	O Servidor NGINX . . . . .	6
<b>2.2</b>	<b><i>Proxy e proxy reverso</i> . . . . .</b>	<b>7</b>
<b>2.3</b>	<b>Mecanismos de <i>cache</i> em aplicações web . . . . .</b>	<b>8</b>
<b>2.4</b>	<b>Trabalhos relacionados . . . . .</b>	<b>8</b>
<b>3</b>	<b>PROPOSTA . . . . .</b>	<b>10</b>
<b>4</b>	<b>CONSIDERAÇÕES FINAIS . . . . .</b>	<b>11</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>12</b>

## 1 INTRODUÇÃO

As aplicações web têm se tornado parte essencial da vida contemporânea, sendo utilizadas em diversos setores, como comunicação, entretenimento, comércio eletrônico e sistemas corporativos. Com a crescente digitalização da sociedade, tais aplicações passaram a desempenhar um papel crítico não apenas no cotidiano dos usuários, mas também no funcionamento de empresas e instituições. Nesse cenário, aspectos como desempenho e disponibilidade são fundamentais para garantir experiências satisfatórias e confiáveis por parte dos usuários.

Entre as técnicas disponíveis, destaca-se a utilização de *proxy* reverso aliado ao uso de mecanismos de *cache* em servidores web. O *proxy* reverso atua como intermediário entre os clientes e um ou mais servidores de aplicação, trazendo benefícios como balanceamento de carga, proteção contra acessos indevidos e armazenamento temporário de conteúdos em *cache* (CLOUDFLARE, 2024b). O *cache*, por sua vez, evita o processamento repetido de requisições idênticas, reduzindo a latência percebida pelo usuário e otimizando o uso de *Central Processing Unit* (CPU) e memória do servidor (AIVALIOTIS, 2016).

Nesse contexto, o servidor NGINX tem ganhado destaque devido à sua ampla adoção em ambientes corporativos e acadêmicos, além de sua eficiência no tratamento de requisições simultâneas. Reconhecido por sua capacidade de atuar tanto como servidor web quanto como *proxy* reverso, o NGINX oferece recursos avançados de caching que permitem melhorar o desempenho de aplicações web em diferentes cenários.

Diante disso, este trabalho busca investigar o impacto da utilização de *proxy* reverso com cache no servidor NGINX em aplicações web estáticas e dinâmicas, avaliando seus efeitos em métricas como tempo de resposta, consumo de memória e uso de CPU. O estudo tem como objetivo fornecer subsídios práticos e científicos que auxiliem acadêmicos, profissionais de Tecnologia da Informação e pequenas empresas na adoção de soluções eficientes para otimização de desempenho em serviços web.

### 1.1 Objetivos

#### 1.1.1 Objetivo geral

O objetivo geral do trabalho consiste em realizar uma análise comparativa do desempenho do servidor NGINX em diferentes configurações, especificamente sua utilização em modo padrão (sem cache) e como *proxy* reverso com cache, a fim de avaliar os impactos sobre aplicações web estáticas e dinâmicas.

### 1.1.2 Objetivos específicos

- Analisar como funciona o *proxy* reverso e o mecanismo de cache do servidor NGINX, bem como a relação entre *proxy* reverso e *caching*.
- Realizar levantamento de aplicações/serviços web de diferentes portes para compor os cenários de testes experimentais.
- Sintetizar os dados coletados em testes de carga realizados em ambientes com e sem *proxy* reverso com cache para medir parâmetros como latência, tempo de resposta e uso de recursos (p.e., memória e processador).
- Avaliar os resultados coletados propondo boas práticas de *proxy* reverso para os diferentes tipos de aplicações.

## 1.2 Justificativa

Com a crescente dependência de serviços web em diversos setores, o desempenho dos servidores tornou-se um fator crítico para a experiência do usuário. Páginas lentas não apenas impactam a satisfação e a retenção de usuários, mas também podem gerar prejuízos financeiros significativos. O tempo de resposta elevado, especialmente em requisições de conteúdo estático (como imagens, arquivos CSS e JavaScript) é um problema. Sem um *proxy* reverso, o servidor de origem precisa processar cada solicitação, gerando latência e sobrecarga. Em contrapartida, o uso do *cache* no *NGINX* resolve esse problema. Ele armazena uma cópia do conteúdo na primeira requisição e, a partir da segunda, entrega a resposta diretamente do *cache*. Esse mecanismo tem como objetivo reduzir o tempo de espera do cliente, oferecendo uma experiência de navegação mais rápida e fluida.

Além disso, servidores web sem *cache* precisam de mais recursos de CPU e memória para lidar com o tráfego, o que pode levar a um gargalo de desempenho. A arquitetura orientada a eventos do *NGINX*, combinada com o *cache*, libera o servidor de origem de grande parte das requisições repetidas. Assim, com o *NGINX* servindo as respostas em *cache*, o servidor original pode focar apenas em processar novas requisições. Com isso, a carga computacional do servidor diminui e tem-se uma otimização dos recursos existentes.

Nesse contexto, a investigação do uso do *NGINX* como *proxy* reverso com *cache* justifica-se por seu potencial em solucionar problemas práticos de desempenho e escalabilidade. O *NGINX*, em sua configuração com *cache*, atua como uma camada intermediária que optimiza o fluxo de tráfego, servindo como uma solução eficiente e de baixo custo para ambientes de alta demanda.

Além disso, este estudo contribui para a literatura e para a prática profissional. Ao oferecer uma análise aplicada e a demonstração dos benefícios do *NGINX*, o trabalho se posiciona

como um estudo de caso valioso. A relevância no mercado é evidente, uma vez que a técnica é amplamente adotada por grandes empresas de tecnologia.

### **1.3 Estrutura do trabalho**

Este trabalho está organizado da seguinte forma. O Capítulo 1 apresenta a introdução, objetivos e justificativa. O Capítulo 2 aborda o referencial teórico sobre servidores web, NGINX, *proxy reverso* e *cache*. A proposta deste Trabalho de Conclusão de Curso é descrita no Capítulo 3 e o Capítulo 4 apresenta as considerações finais.

## 2 REFERENCIAL TEÓRICO

No referencial teórico, serão discutidos os principais conceitos que embasam este estudo, começando com uma introdução sobre **servidores web**. Em seguida, serão exploradas as definições e funções de **proxy reverso** e **mechanismos de caching**, destacando a importância dessas tecnologias para a otimização de recursos. Por fim, o capítulo abordará as características do servidor **NGINX**, ressaltando sua relevância para o contexto da pesquisa.

### 2.1 Servidores web

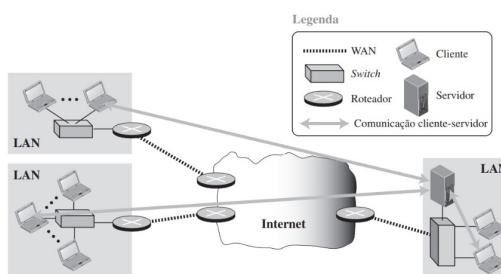
Os servidores web são *softwares* responsáveis por receber e processar requisições do *Hyper Transfer Protocol* (HTTP) ou do *Hypertext Transfer Protocol Secure* (HTTPS) realizadas por clientes (navegadores ou outras aplicações) e retornar os recursos solicitados, como páginas *Hypertext Markup Language*(HTML), imagens, arquivos ou dados estruturados (AIVALIOTIS, 2016). Tradicionalmente, os servidores web desempenham papel central na arquitetura cliente-servidor da web, atuando como ponto de contato entre o usuário e os sistemas hospedados(AIVALIOTIS, 2016). A Figura 1 ilustra a ideia de um servidor servindo a dois usuários na Internet.

Dentre as características mais relevantes de um servidor web, destacam-se: capacidade de lidar com múltiplas conexões simultâneas, suporte a protocolos modernos (como HTTP/2 e HTTP/3), escalabilidade, segurança e compatibilidade com diferentes tecnologias de aplicação. O desempenho do servidor web é um fator determinante para a experiência do usuário final e para a eficiência no consumo de recursos computacionais (BUTLER, 2017).

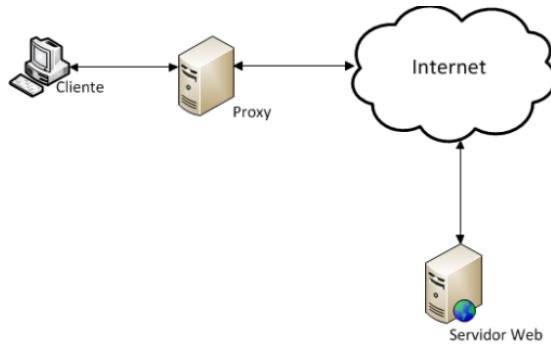
Alguns exemplos de servidores web são o *Apache HTTP Server*, o *Lighttpd* e o *NGINX*.

#### 2.1.1 O Servidor NGINX

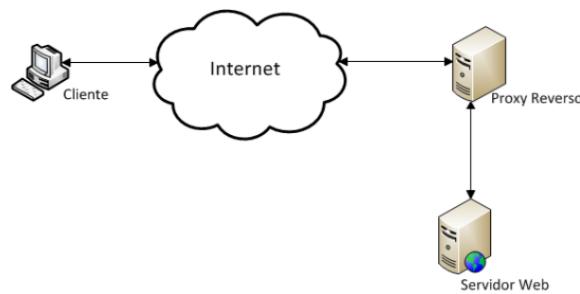
O NGINX é um servidor web de alto desempenho que se consolidou como uma das principais ferramentas utilizadas mundialmente em razão de sua capacidade de lidar com grande volume de conexões simultâneas com baixo consumo de recursos(AIVALIOTIS, 2016). Sua ar-



**Figura 1 – Ilustração de um servidor web (FOROUZAN; MOSHARRAF, 2013).**



**Figura 2 – Arquitetura de um servidor proxy (SÁ; SOUZA, 2015).**



**Figura 3 – Arquitetura de um proxy reverso (SÁ; SOUZA, 2015).**

Arquitetura baseada em eventos assíncronos permite escalabilidade superior quando comparada a modelos tradicionais baseados em *threads*.

Além de atuar como servidor web, o NGINX também é amplamente empregado como proxy reverso e平衡ador de carga. O NGINX *Cookbook* (BUTLER, 2017) destaca casos práticos de configuração que demonstram ganhos de desempenho significativos em aplicações que lidam com tráfego intenso e requisições repetidas.

## 2.2 Proxy e proxy reverso

Um *proxy* é um servidor intermediário utilizado entre o cliente e o servidor (CLOUDFLARE, 2024b). Assim, quando um cliente faz uma requisição para o servidor, o *proxy* intercepta a requisição e se comunica com o servidor como um intermediário, ou seja, em nome do servidor. O *proxy* recebe a resposta e encaminha ao cliente. O cliente não acessa diretamente o servidor do domínio, mas passa pelo *proxy*, ou seja, acessa a Internet através dele. No contexto de redes e aplicações web, o *proxy* pode desempenhar funções como filtragem de tráfego, anonimização, restrição de acesso ou *caching* de conteúdos. A Figura 2 ilustra a arquitetura de um servidor *proxy*.

O *proxy* reverso, por outro lado, atua de forma oposta, ficando próximo do servidor de origem (CLOUDFLARE, 2024b). O cliente sempre se comunica com esse *proxy* reverso como se ele fosse o próprio servidor do domínio. Logo, o *proxy* reverso intercepta as solicitações, encaminha ao servidor, recebe a resposta e envia para o cliente. A Figura 3 ilustra a arquitetura de um *proxy* reverso.

Entre os principais benefícios do uso de *proxy* reverso, destacam-se (CLOUDFLARE, 2024b):

- Balanceamento de carga, permitindo que requisições sejam distribuídas entre múltiplos servidores.
- Segurança, protegendo os servidores de aplicação contra ataques diretos.
- *Caching*, armazenando temporariamente conteúdos e reduzindo a necessidade de processamento repetitivo.

Em resumo, as diferenças entre os dois é que com o *proxy*, o cliente acessa a Internet através dele, enquanto que com o *proxy* reverso, o cliente acessa um servidor, mas a requisição passa primeiro pelo *proxy* reverso que representa o servidor.

»» Colocar no bibtex—> DE SÁ, Sant'Anna et al. Uma proposta de arquitetura para melhoria de desempenho no ambiente virtual de aprendizagem Moodle utilizando proxy reverso. 2015.

### **2.3 Mecanismos de *cache* em aplicações web**

Um *cache* é um local de armazenamento temporário para cópias de arquivos ou outros tipos de dados que busca oferecer às aplicações acesso mais rápido(CLOUDFLARE, 2024a). Em aplicações web, o mecanismo de *cache* também é utilizado.

Desta forma, existem diferentes níveis de cache para aplicações web:

1. *Cache* no navegador, armazenando recursos no dispositivo do cliente.
2. *Cache* em proxies, interceptando tráfego e guardando conteúdos frequentemente requisitados.
3. *Cache* no servidor de aplicação, otimizando respostas dinâmicas.

No contexto do NGINX, o cache integrado ao *proxy* reverso permite que respostas a requisições repetidas sejam entregues rapidamente sem necessidade de acessar o servidor de aplicação, diminuindo o uso de *Central Processing Unit* (CPU) e memória.

### **2.4 Trabalhos relacionados**

Diversos estudos têm investigado o desempenho de servidores web, seja em análises experimentais ou em revisões de literatura. O objetivo comum desses trabalhos é identificar vantagens e limitações de soluções como Apache e Nginx, assim como propor arquiteturas para otimização em contextos específicos.

Sá e Souza (2015) propuseram uma arquitetura baseada em *proxy reverso* utilizando o Varnish para melhorar o desempenho do Moodle, um Ambiente Virtual de Aprendizagem (AVA). Os testes realizados com diferentes números de usuários simultâneos (50, 100, 150 e 200) mostraram ganhos significativos de latência e disponibilidade ao empregar o cache reverso (Sá; SOUZA, 2015).

Kunda, Chihana e Muwane (2017) realizaram uma *revisão sistemática da literatura* sobre o desempenho dos servidores Apache e Nginx. O estudo reuniu resultados de múltiplos experimentos já publicados, destacando que o Nginx apresenta melhor escalabilidade, menor consumo de memória e CPU em cenários de alto volume de conexões, enquanto o Apache tende a obter vantagem em conteúdos dinâmicos, como scripts PHP (KUNDA; CHIHANA; MUWANE, 2017).

Mais recentemente, Pinastawa, Pradana e Maulana (2024) conduziram um *estudo experimental* comparando Apache e Nginx com o uso da ferramenta K6. Foram aplicados diferentes métodos de teste: carga (*load*), pico (*spike*), saturação (*soak*) e desempenho (*performance*). Os resultados mostraram que o Nginx obteve melhor desempenho geral, com tempos de resposta mais baixos, menor taxa de falhas e maior quantidade de requisições processadas por segundo, especialmente em cenários de saturação e carga constante (PINASTAWA; PRADANA; MAULANA, 2024).

Esses trabalhos evidenciam a relevância de estudar comparativamente os servidores web. Enquanto revisões de literatura consolidam tendências gerais de desempenho, estudos experimentais recentes reforçam a superioridade do Nginx em situações de alta concorrência, ao passo que o Apache mantém relevância em aplicações que demandam maior flexibilidade e suporte a conteúdos dinâmicos.

### 3 PROPOSTA

Este trabalho se propõe a estudar e analisar os impactos do servidor NGINX como *proxy* reverso com *cache* quanto ao desempenho para aplicações web. Logo, o trabalho consistirá na instalação, configuração e análise de ambientes web utilizando o servidor NGINX com e sem *proxy* reverso com *cache*. Serão realizadas comparações de desempenho em aplicações estáticas (conteúdo fixo) e dinâmicas (com processamento de dados), de modo a identificar ganhos, limitações e melhores práticas.

A metodologia envolverá a criação de cenários de teste, a execução de testes de carga utilizando ferramentas específicas (como *Apache Benchmark*, *wrk* ou *k6*) e a coleta de métricas de desempenho. Os resultados obtidos serão analisados de forma crítica e comparativa. Desta forma, acadêmicos, profissionais de tecnologia e pequenas empresas podem utilizar este estudo como base para otimizar o desempenho de aplicações web sem recorrer a soluções proprietárias de alto custo.

Espera-se demonstrar, por meio de experimentação, os benefícios da utilização de *proxy* reverso com *cache* no NGINX, fornecendo evidências quantitativas de redução de latência e melhor aproveitamento de recursos. Ao final, pretende-se propor um conjunto de recomendações práticas que possam ser aplicadas em diferentes contextos.

## 4 CONSIDERAÇÕES FINAIS

Este projeto de Trabalho de Conclusão de Curso tem como foco a investigação dos efeitos da implementação de *proxy* reverso com *cache* utilizando o servidor NGINX em aplicações web, tanto estáticas quanto dinâmicas. A análise será conduzida por meio de métricas de desempenho como tempo de resposta, uso de memória e carga de CPU, com o intuito de oferecer fundamentos práticos e científicos que auxiliem na escolha de soluções eficazes para a otimização de serviços online.

A importância desta pesquisa aplicada está diretamente ligada à crescente demanda por sistemas web mais ágeis, escaláveis e confiáveis, especialmente em um contexto onde organizações dependem fortemente de plataformas digitais. Com isso, espera-se que os resultados beneficiem tanto o meio acadêmico — ao consolidar práticas recomendadas — quanto profissionais da área de tecnologia, ao fornecer orientações técnicas aplicáveis no cotidiano.

Entre os principais resultados almejados, destaca-se a comprovação de que o uso de *proxy* reverso com *cache* no NGINX pode diminuir significativamente a latência e melhorar o aproveitamento dos recursos computacionais, promovendo ambientes mais eficientes. O impacto dessa melhoria pode se refletir em uma experiência de usuário mais fluida e em maior competitividade para empresas que adotam tais estratégias.

Ainda assim, o projeto reconhece a existência de desafios, como a criação de cenários de teste que representem fielmente o uso real, a escolha criteriosa das métricas de avaliação e a consideração das variações de desempenho conforme o tipo de aplicação. Esses aspectos exigirão atenção especial ao longo da pesquisa para assegurar a precisão e a confiabilidade dos resultados obtidos.

## REFERÊNCIAS

- AIVALIOTIS, D. **Mastering NGINX - Second Edition**. Birmingham: Packt Publishing, 2016.
- BUTLER, T. **NGINX Cookbook: Over 70 recipes for real-world configuration, deployment, and administration**. Birmingham: Packt Publishing, 2017.
- CLOUDFLARE. **O que é caching?** 2024. Acesso em: 10 set. 2025. Disponível em: <https://www.cloudflare.com/pt-br/learning/cdn/what-is-caching>.
- CLOUDFLARE. **O que é um proxy reverso?** 2024. Acesso em: 02 set. 2025. Disponível em: <https://www.cloudflare.com/pt-br/learning/cdn/glossary/reverse-proxy/>.
- FOROUZAN, B. A.; MOSHARRAF, F. **Redes de Computadores**. Porto Alegre: Grupo A, 2013. E-book. ISBN 9788580551693.
- KUNDA, D.; CHIHANA, S.; MUWANE, S. Web server performance of apache and nginx: A systematic literature review. **International Journal of Computer Applications**, v. 179, n. 47, p. 1–7, nov. 2017.
- PINASTAWA, I. W. R.; PRADANA, M. G.; MAULANA, N. Web server performance evaluation: Comparing nginx and apache using k6 testing methods for load, spike, soak, and performance. In: **Proceedings of the 2024 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)**. [S.I.]: IEEE, 2024. p. 1002–1010.
- Sá, S. S. de; SOUZA, S. C. de. Uma proposta de arquitetura para melhoria de desempenho no ambiente virtual de aprendizagem moodle utilizando proxy reverso. In: **Quinta Conferência de Diretores de Tecnologia de Informação e Comunicação em Instituições de Educação Superior**. Brasil: [s.n.], 2015. p. 1–12.