

Avaliação Comparativa de Desempenho entre APIs

João Henrique Calza, Murilo Tonello Badim e Erik de Chaves Lavarda

1

1. Introdução

O desenvolvimento de APIs modernas exige não apenas funcionalidades corretas, mas também desempenho, escalabilidade e eficiência no consumo de recursos. Este trabalho tem como objetivo comparar três frameworks distintos utilizados na implementação de APIs, analisando seu comportamento sob carga e estresse.

Foram implementadas três APIs equivalentes para a entidade *Users*, utilizando **FastAPI (Python)**, **C#.NET** e **Lumen (PHP)**. Todas usam PostgreSQL e foram executadas em contêineres Docker. Os testes de desempenho foram realizados com o Apache JMeter, avaliando principalmente tempo médio de resposta, throughput, taxa de erros, uso de CPU e memória RAM.

2. Sobre as Tecnologias Avaliadas

2.1. FastAPI

FastAPI é um framework moderno em Python que utiliza ASGI e execução assíncrona. É reconhecido por seu alto desempenho, tipagem automática, validação integrada e excelente escalabilidade.

2.2. C#.NET

A implementação em C.NET foi desenvolvida com o .NET 8, plataforma conhecida por sua maturidade e pelo amplo uso em sistemas corporativos. O ambiente oferece um conjunto sólido de ferramentas e bibliotecas, o que facilita a construção de APIs de forma organizada e consistente.

2.3. Lumen

Lumen é um micro-framework derivado do Laravel, voltado para aplicações menores com foco em rapidez de resposta. Embora seja mais leve que Laravel, seu desempenho está ligado às características do ambiente PHP, que opera de forma diferente de soluções nativamente assíncronas.

3. Implementação

As três APIs seguiram exatamente o mesmo conjunto de operações CRUD e utilizaram schemas independentes no PostgreSQL. A infraestrutura Docker utilizada incluiu:

- 2 instâncias de cada API;
- gateway com balanceamento;
- banco PostgreSQL compartilhado;
- rede interna Docker.

A entidade *Users* contém: `id`, `name`, `email`, `user`, `password`.

4. Testes de Desempenho

Foram realizados dois tipos principais de teste:

- **Teste de Carga:** aumento gradual dos usuários simultâneos;
- **Teste de Estresse:** aumento contínuo até o ponto de saturação.

4.1. Teste de Carga

O teste de carga foi executado de forma gradual, utilizando cinco etapas independentes com 10, 30, 50, 100 e 150 requisições simultâneas. Ao final das execuções, o total processado foi de 340 requisições, distribuídas entre os diferentes níveis de carga. Nessa avaliação, o Lumen apresentou o maior throughput (3,7 req/s), seguido pelo FastAPI (3,4 req/s) e pelo C#.NET (2,8 req/s). Quanto ao tempo médio de resposta, o FastAPI obteve o melhor resultado (8 ms), seguido pelo C#.NET (14 ms) e pelo Lumen (24 ms). Em todas as etapas, foi observado 0% de erros, indicando estabilidade das três APIs sob cargas leves.

4.2. Teste de Estresse

As figuras a seguir apresentam as principais métricas de desempenho obtidas nos testes de estresse pelo JMeter, cada um realizado com 24.800 requisições por API

FASTAPI

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	24800	2821	2	27117	4076.63	0.00%	111.7/sec	16.90	14.72	155.0
TOTAL	24800	2821	2	27117	4076.63	0.00%	111.7/sec	16.90	14.72	155.0

A API em FastAPI apresentou o melhor desempenho entre as três: tempo médio de apenas **2821 ms**, **0% de erros** e o maior throughput, atingindo **111.7 requisições/segundo**. Esses resultados indicam excelente estabilidade e alta capacidade de processamento sob carga.

C#.NET

Label ↓	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	24800	7795	0	60564	14887.77	4.84%	77.2/sec	14.47	10.15	192.0
TOTAL	24800	7795	0	60564	14887.77	4.84%	77.2/sec	14.47	10.15	192.0

A API em .NET apresentou desempenho intermediário: tempo médio de **7795 ms**, com **4.84% de erros** e throughput de **77.2 requisições/segundo**. Apesar de melhor que o Lumen, ainda demonstrou instabilidades e menor eficiência geral em alta carga.

LUMEN

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	24800	11576	0	24642	6070.90	6.51%	87.7/sec	57.18	9.96	667.8
TOTAL	24800	11576	0	24642	6070.90	6.51%	87.7/sec	57.18	9.96	667.8

A API em Lumen exibiu o pior desempenho entre as três: tempo médio elevado de **11576 ms**, a maior taxa de erros (**6.51%**) e throughput de apenas **87.7 requisições/segundo**. Os resultados mostraram dificuldade significativa em lidar com altas cargas simultâneas.

4.3. Consumo de Recursos (Docker Stats)

PYTHON

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
62677dc582b6	gateway	19.22%	19.21MiB / 7.728GiB	0.24%	9.34MB / 9.42MB	0B / 8.19kB	5
1f2f94592cdf	lumen_api_1	0.00%	17.24MiB / 7.728GiB	0.22%	7.33kB / 107kB	532kB / 0B	4
90c1ba22de5d	lumen_api_2	0.00%	8.676MiB / 7.728GiB	0.11%	1.86kB / 126B	0B / 0B	3
00bad4cea8a4	api_python_1	36.60%	49.27MiB / 7.728GiB	0.62%	2.66MB / 2.28MB	0B / 28.7kB	36
f873c48ff8ff	api_python_2	42.23%	48.44MiB / 7.728GiB	0.61%	2.73MB / 2.42MB	0B / 28.7kB	31
1eadb555aa10	api_csharp_1	0.01%	52.98MiB / 7.728GiB	0.67%	30.2kB / 14.1kB	0B / 0B	17
a1b7ff954de0	api_csharp_2	0.00%	21.55MiB / 7.728GiB	0.27%	2.01kB / 126B	0B / 0B	16
ae73e4b20760	postgres	14.51%	20.29MiB / 7.728GiB	0.26%	18.8kB / 31kB	0B / 389kB	6

O FastAPI apresentou um consumo de CPU moderado e um uso de memória intermediário em comparação com as demais APIs.

LUMEN

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
34413dc82464	gateway	7.55%	35.41MiB / 7.728GiB	0.45%	9.68MB / 13.6MB	3.6MB / 8.19kB	5
698d320bf1de	lumen_api_2	86.97%	37.54MiB / 7.728GiB	0.47%	11.1MB / 8.4MB	19.4MB / 0B	6
cb25ee7aaedc	lumen_api_1	91.21%	33.5MiB / 7.728GiB	0.42%	11.2MB / 8.45MB	16.4MB / 0B	7
3e425d43244b	api_csharp_2	0.00%	23.46MiB / 7.728GiB	0.30%	1.9kB / 126B	33.1MB / 0B	16
02cb725514ec	api_python_1	0.10%	48.41MiB / 7.728GiB	0.61%	1.86kB / 126B	18.7MB / 28.7kB	5
5656b95e3174	api_python_2	0.09%	44.6MiB / 7.728GiB	0.56%	1.73kB / 126B	8.72MB / 28.7kB	5
5c45c535fd9d	api_csharp_1	0.00%	43.05MiB / 7.728GiB	0.54%	28.7kB / 13.3kB	42.4MB / 0B	17
ae73e4b20760	postgres	116.68%	55.73MiB / 7.728GiB	0.70%	11MB / 14.5MB	40.3MB / 496kB	19

O Lumen apresentou um uso de CPU mais elevado, porém com baixo consumo de memória.

C#.NET

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
34413dc82464	gateway	6.21%	16.12MiB / 7.728GiB	0.20%	35.1MB / 47MB	3.6MB / 8.19kB	5
698d320bf1de	lumen_api_2	0.00%	33.6MiB / 7.728GiB	0.42%	36.2MB / 29MB	19.4MB / 0B	4
cb25ee7aaedc	lumen_api_1	0.19%	29.49MiB / 7.728GiB	0.37%	36.2MB / 29MB	16.4MB / 0B	5
3e425d43244b	api_csharp_2	56.14%	139.2MiB / 7.728GiB	1.76%	3.09MB / 2.94MB	33.1MB / 0B	25
02cb725514ec	api_python_1	1.36%	48.41MiB / 7.728GiB	0.61%	2.06kB / 126B	18.7MB / 28.7kB	5
5656b95e3174	api_python_2	0.51%	44.6MiB / 7.728GiB	0.56%	2kB / 126B	8.72MB / 28.7kB	5
5c45c535fd9d	api_csharp_1	73.24%	158.2MiB / 7.728GiB	2.00%	3.05MB / 2.88MB	46MB / 0B	26
ae73e4b20760	postgres	160.77%	139.9MiB / 7.728GiB	1.77%	40.9MB / 53.3MB	40.5MB / 496kB	96

A API em C.NET apresentou um uso de CPU mediano em comparação com as outras APIs, porém com um consumo de memória mais elevado.

5. Conclusão

Os testes realizados indicam que o FastAPI apresentou o melhor desempenho geral entre as três tecnologias avaliadas, combinando alta estabilidade, bom throughput e eficiência no uso de recursos. O C#.NET se mostrou uma alternativa equilibrada, com desempenho consistente, porém inferior ao FastAPI, especialmente sob maior carga. O Lumen, embora leve em memória, apresentou limitações significativas em cenários de alta concorrência, demonstrando maior uso de CPU e menor eficiência em processamento. Em cargas leves, todas as APIs mantiveram estabilidade, com diferenças concentradas principalmente na capacidade de processamento e tempo de resposta.