

Relatório Técnico: Teste de Desempenho da API de Checkout

Engenharia de Performance

24 de novembro de 2025

1 Resumo Executivo

O objetivo destes testes foi avaliar os limites da *Ecommerce Checkout API*, focando em dois comportamentos distintos: I/O Bound (rota `/simple`) e CPU Bound (rota `/crypto`).

- **Cenário de I/O (`/checkout/simple`):** A aplicação demonstrou alta resiliência. Suportou a carga nominal de **50 VUs** e picos repentinos de **300 VUs** sem apresentar erros, mantendo a latência estável ($p95 \approx 294\text{ms}$).
- **Cenário de CPU (`/checkout/crypto`):** A aplicação falhou catastroficamente sob estresse. Ao tentar escalar para 1000 usuários, a taxa de erro atingiu **90%** e a latência $p95$ subiu para mais de **5.3 segundos**. O sistema não suporta alta concorrência nesta rota devido ao bloqueio do *Event Loop*.

2 Evidências e Métricas (k6)

Abaixo estão os resultados consolidados das execuções dos scripts k6.

2.1 1. Smoke Test (Disponibilidade)

Objetivo: Validar saúde da API sob carga mínima.

Métrica	Resultado
Usuários Virtuais (VUs)	1
Taxa de Sucesso	100%
Latência Média	0.45 ms
Latência p95	1.04 ms
Total de Requisições	30

2.2 2. Load Test (Carga Normal - I/O Bound)

Objetivo: Simular tráfego esperado de promoção (50 usuários).

Métrica	Resultado
VUs Máximos	50
Taxa de Erro	0.00% (Aprovado)
Latência p95	294.93 ms (Meta: < 500ms)
Throughput	≈ 32.6 req/s
Total de Requisições	6872

2.3 3. Stress Test (Ponto de Ruptura - CPU Bound)

Objetivo: Encontrar o limite da API com cálculos pesados.

Métrica	Resultado
VUs Máximos Tentados	1000
Taxa de Erro	90.02% (Falha Crítica)
Latência p95	5322.67 ms
Latência Máxima	58166.41 ms
Requisições com Sucesso	6076
Requisições com Falha	54805

2.4 4. Spike Test (Pico Repentino - I/O Bound)

Objetivo: Salto de 10 para 300 usuários em 10s.

Métrica	Resultado
VUs Máximos	300
Taxa de Erro	0.00%
Latência p95	293.64 ms
Throughput	≈ 160.1 req/s
Total de Requisições	17795

3 Análise de Estresse e Conclusão

O teste de estresse revelou que a rota de criptografia é o gargalo crítico do sistema. Enquanto a rota de I/O (`/simple`) manteve a performance constante mesmo com 300 usuários simultâneos (Spike Test), a rota de CPU (`/crypto`) colapsou.

Diagnóstico: A taxa de falha de 90% indica que o servidor parou de responder adequadamente muito antes de atingir os 1000 usuários. Diferente da latência de I/O que é assíncrona, o cálculo de hash bloqueia a thread principal do Node.js.

Recomendação: Para suportar a carga de estresse testada, a funcionalidade de hash deve ser movida para um *Worker Thread*, uma arquitetura de microsserviços dedicada ou executada de forma assíncrona real, evitando o bloqueio do *Event Loop*.