Relatório — Análise de Desempenho da API de Checkout

Resumo
Este relatório apresenta os procedimentos experimentais e os resultados obtidos ao submeter a API de checkout a uma bateria de ensaios de desempenho. Foram avaliados cenários de verificação inicial (smoke), carga sustentada (load) e picos abruptos (spike) com o objetivo de quantificar latência e confiabilidade dos endpoints representativos de operações I/O-bound (/checkout/simple) e CPU-bound (/checkout/crypto). O ensaio de estresse completo não produziu um resumo final utilizável devido a violações de limiares e recusas de conexão durante a execução; por isso, as conclusões relativas ao comportamento CPU-bound são apresentadas de forma conservadora.

Introdução
A avaliação de desempenho buscou identificar limites operacionais e propriedades de degradação sob diferentes padrões de carga. A API em estudo contém dois endpoints com comportamentos contrastantes: um que simula latência de I/O sem bloquear o event loop, e outro que executa operações de CPU de forma síncrona (bcrypt). Esta distinção permite observar, na prática, diferenças entre gargalos de I/O e de CPU e inferir estratégias de mitigação.

Metodologia experimental
Os experimentos foram conduzidos em ambiente local, direcionados ao host http://localhost:3000. Foram utilizados scripts k6 que implementam os seguintes cenários: smoke (1 VU, 30 s, endpoint /health), load (rampa 0 a 50 VUs em 1 minuto, platô de 50 VUs por 2 minutos, rampa de descida em 30 s, endpoint /checkout/simple), spike (salto rápido de 10 para 300 VUs, endpoint /checkout/simple) e stress (rampa agressiva até 1000 VUs, endpoint /checkout/crypto). Para cada execução foram exportados resumos JSON do k6 quando disponíveis. As métricas extraídas e analisadas foram percentil 95 da latência (p95), taxa de requisições com falha (http_req_failed) e contagem de checks que validam o código de resposta esperado.

Resultados principais
No ensaio smoke, a API mostrou disponibilidade consistente, com todos os checks aprovados. No ensaio de load, a execução com até 50 VUs sustentados apresentou p95 de latência em aproximadamente 296 ms, com todas as requisições POST ao endpoint /checkout/simple retornando o código 201 conforme esperado. Esse resultado indica que, para a carga projetada de 50 usuários simultâneos, o endpoint I/O-bound atende o objetivo de latência definido no escopo da atividade.
No ensaio spike, a elevação súbita de carga causou degradação significativa: observou-se um aumento expressivo nas latências e uma taxa elevada de falhas nas requisições. Esse comportamento evidencia que picos instantâneos sem mecanismos de amortecimento (como filas, limites de taxa ou buffering) resultam em degradação rápida da experiência do usuário.

O ensaio de stress para o endpoint CPU-bound não gerou um summary final utilizável; em execuções anteriores verificaram-se recusas de conexão e violações de thresholds enquanto a rampa aumentava. Dado que o endpoint realiza hashing de forma síncrona, é plausível atribuir o comportamento observado à saturação de CPU e ao bloqueio do loop de eventos, o que reduz a capacidade de atender novas conexões conforme cresce a concorrência. Com base nas evidências do spike e nas características da implementação, estimamos de forma conservadora que o ponto de ruptura para carga CPU-heavy ocorre substancialmente abaixo de 1000 VUs e possivelmente na ordem de algumas centenas de VUs.

Discussão
Os ensaios confirmam que a aplicação exibe dois modos distintos de limitação: para operações I/O-bound, a latência permanece controlada sob carga moderada sustentada; para operações CPU-bound, a execução síncrona de cargas compute-intensive compromete rapidamente a capacidade do servidor. O ensaio spike demonstrou que, apesar de a aplicação suportar cargas sustentadas esperadas, bursts intensos provocam colapso parcial ou total do serviço, refletido em alta latência e elevado número de falhas. A ausência de um resumo do stress final impede a quantificação precisa do breaking point, mas o padrão dos dados disponíveis motiva recomendações práticas imediatas.

Conclusões e recomendações
Conclui-se que, para o cenário de promoção com 50 usuários simultâneos, o endpoint I/O-bound comporta-se adequadamente em termos de latência (p95 dentro do limite desejado). Para cargas CPU-bound, recomenda-se evitar processamento síncrono no fluxo de requisição; em vez disso, delegar tarefas pesadas a workers ou filas, ou executar hashing em processos separados para não bloquear o loop principal. Adicionalmente, recomenda-se a adoção de mecanismos de mitigação de picos, tais como rate limiting, circuit breaker e enfileiramento de requisições. Para determinar o breaking point do endpoint CPU-bound com precisão, é recomendada uma abordagem incremental de testes (probe): subir a carga progressivamente em passos controlados (por exemplo, 50 -> 100 -> 200 -> 400 VUs), monitorando p95, p99, uso de CPU e taxa de erros.

Limitações do estudo
A principal limitação é a ausência de um artefato final do teste de stress sobre /checkout/crypto; sem esse resumo não é possível fornecer um valor numérico definitivo para o ponto de ruptura da carga CPU-bound. Outras limitações incluem a execução em ambiente local, sem replicação em infraestrutura distribuída, e a ausência de coleta sistemática de métricas de sistema (CPU, memória, fila de conexões) correlacionadas com as métricas de aplicação.

Anexo — comandos executados (reprodutibilidade)
- Inicializar e executar a API:
npm install
npm start (executa src/server.js em http://localhost:3000)
- Exportar resumos k6 (exemplos):
k6 run –summary-export=results/smoke-summary.json tests/smoke.js

k6 run –summary-export=results/load-summary.json tests/load.js
k6 run –summary-export=results/spike-summary.json tests/spike.js
- Recomenda-se para análise adicional:
k6 run –summary-export=results/stress-probe-summary.json tests/stress-probe.js

Considerações finais
Este documento sintetiza as evidências quantitativas disponíveis e traduz observações técnicas em recomendações concretas. Caso sejam disponibilizados os summaries adicionais do teste de stress ou de um probe incremental, a análise pode ser refinada e o relatório atualizado com os valores numéricos do ponto de ruptura e com gráficos de degradação.