Relatório de Teste de Carga K6

Grupo:

Rhuan Soares

Thauan Fabrício

Gabriel de Oliveira

1. OBJETIVOS DO PROJETO

1.1. OBJETIVO GERAL

- Realizar a análise de desempenho de serviços internos do sistema através de testes de carga, identificando métricas críticas de performance e propondo hipóteses para possíveis gargalos.

1.2. OBJETIVOS ESPECÍFICOS

- Selecionar um conjunto mínimo de dois serviços internos para análise, garantindo que pelo menos um deles execute operações de escrita no banco de dados.
- Projetar e executar testes de carga nos serviços selecionados, submetendo-os a diferentes cenários de demanda.
- Coletar e analisar métricas de desempenho durante os testes, incluindo:
    - Latência (tempo médio de resposta)
    - Vazão (requisições processadas por segundo em intervalos específicos: 1s, 5s, 10s, 30s e 1min)
    - Capacidade de concorrência (número máximo de requisições simultâneas suportadas)

2. RESULTADO DO TERMINAL

```
  execution: local
     script: index.js
     output: json (resultado.json)

  scenarios: (100.00%) 1 scenario, 10 max VUs, 1m0s max duration (incl. graceful stop):
           * default: 10 looping VUs for 30s (gracefulStop: 30s)


  TOTAL RESULTS

  checks_total.......: 392     12.325396/s
  checks_succeeded...: 100.00% 392 out of 392
  checks_failed......: 0.00%   0 out of 392

  ✓ GET /saldo - 200
  ✓ POST /registrar - 201

  CUSTOM
  concorrencia_obter_saldo.....: 1      min=1        max=1
  concorrencia_registrar_cliente...: 1      min=1        max=1
  latencia_obter_saldo.........: avg=30.375829  min=5.8346   med=10.55275 max=330.6775 p(90)=60.82755 p(95)=129.712775
  latencia_registrar_cliente.....: avg=543.154458 min=381.3508 med=536.9724 max=814.1836 p(90)=631.2235 p(95)=713.834
  vazao_obter_saldo.............: 196    6.162698/s
  vazao_registrar_cliente.....: 196    6.162698/s

  HTTP
  http_req_duration.............: avg=287.11ms   min=5.83ms   med=381.35ms max=814.18ms p(90)=578.55ms p(95)=630.05ms
    { expected_response:true }.....: avg=287.11ms   min=5.83ms   med=381.35ms max=814.18ms p(90)=578.55ms p(95)=630.05ms
  http_req_failed...............: 0.00%  0 out of 393
  http_reqs.....................: 393    12.356838/s

  EXECUTION
  iteration_duration.............: avg=1.57s      min=1.39s    med=1.54s    max=2.09s    p(90)=1.67s    p(95)=1.81s
  iterations.....................: 196    6.162698/s
  vus............................: 8      min=8        max=10
  vus_max........................: 10     min=10       max=10

  NETWORK
  data_received..................: 83 kB  2.6 kB/s
```

3. MEDIÇÕES DO SLA

a) Serviço: Obter Saldo (ObterSaldo.js)

- Tipo de operação: Leitura
- Arquivos envolvidos: https://github.com/thauanhub/paper-bank/blob/main/backend/auth.py
- Arquivos de medição SLA: https://github.com/GbosDev/TestesDeCargaK6-PaperBank.git
- Data da medição: 16/11/2025
- Configurações: 12th Gen Intel(R) Core(TM) i5-1235U (1.30 GHz), 16,0 GB, 64bit, Node.js v25.2.1, Banco MySQL + MongoDB

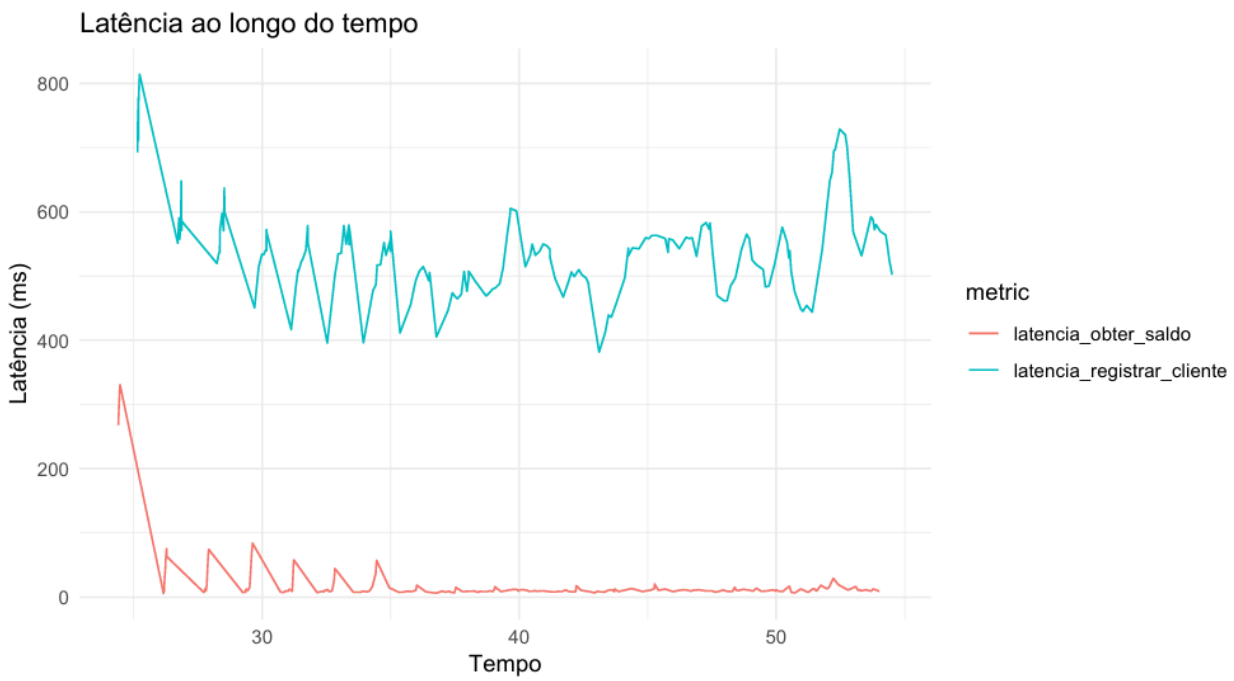
- Testes de carga:
    - Latência: 30.37 ms (média), 129.71 ms (p95)
    - Vazão: 6.16 requisições/segundo
    - Concorrência: 1 requisição simultânea por usuário virtual

b) Serviço: Registrar Cliente (RegistrarCliente.js)

- Tipo de operação: Inserção
- Arquivos envolvidos: https://github.com/thauanhub/paper-bank/blob/main/backend/auth.py
- Arquivos de medição SLA: https://github.com/GbosDev/TestesDeCargaK6-PaperBank.git
- Data da medição: 16/11/2025
- Configurações: 12th Gen Intel(R) Core(TM) i5-1235U (1.30 GHz), 16,0 GB, 64bit, Node.js v25.2.1, Banco MySQL + MongoDB
- Testes de carga:
    - Latência: 543.15 ms (média), 713.83 ms (p95)
    - Vazão: 6.16 requisições/segundo
    - Concorrência: 1 requisição simultânea por usuário virtual

4. ANÁLISE DOS RESULTADOS DO TESTE DE CARGA

4.1. Gráfico de Latência × Tempo

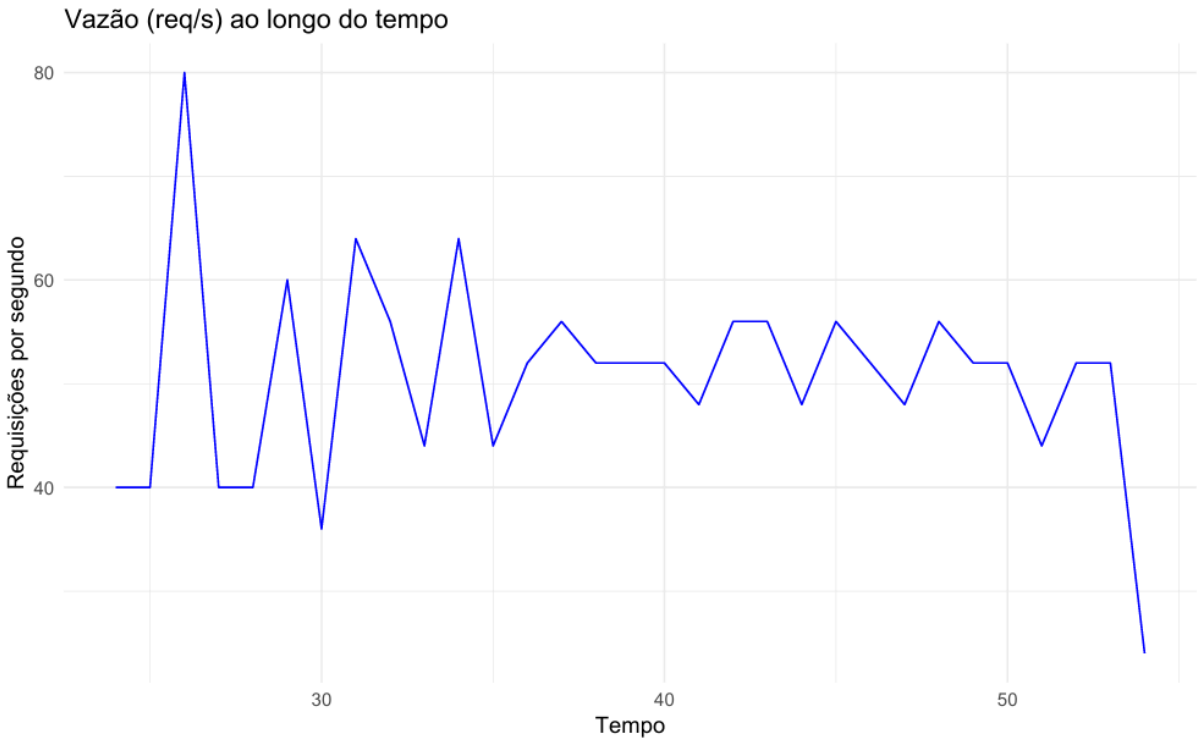


- Para este cenário foram feitas as seguintes análises:

O gráfico demonstra uma diferença significativa no desempenho entre as duas operações analisadas. A consulta de saldo (GET /saldo) manteve-se consistentemente rápida e estável, com tempo médio de resposta de 30,38 ms. Em contraste, a operação de registro (POST /registrar) apresentou tempo médio consideravelmente superior (543,15 ms), com variações mais acentuadas durante o teste.

Este comportamento era esperado, uma vez que operações de escrita tendem a ser naturalmente mais lentas que operações de leitura.

4.2. Gráfico de Vazão (Requisição por segundo) X Tempo

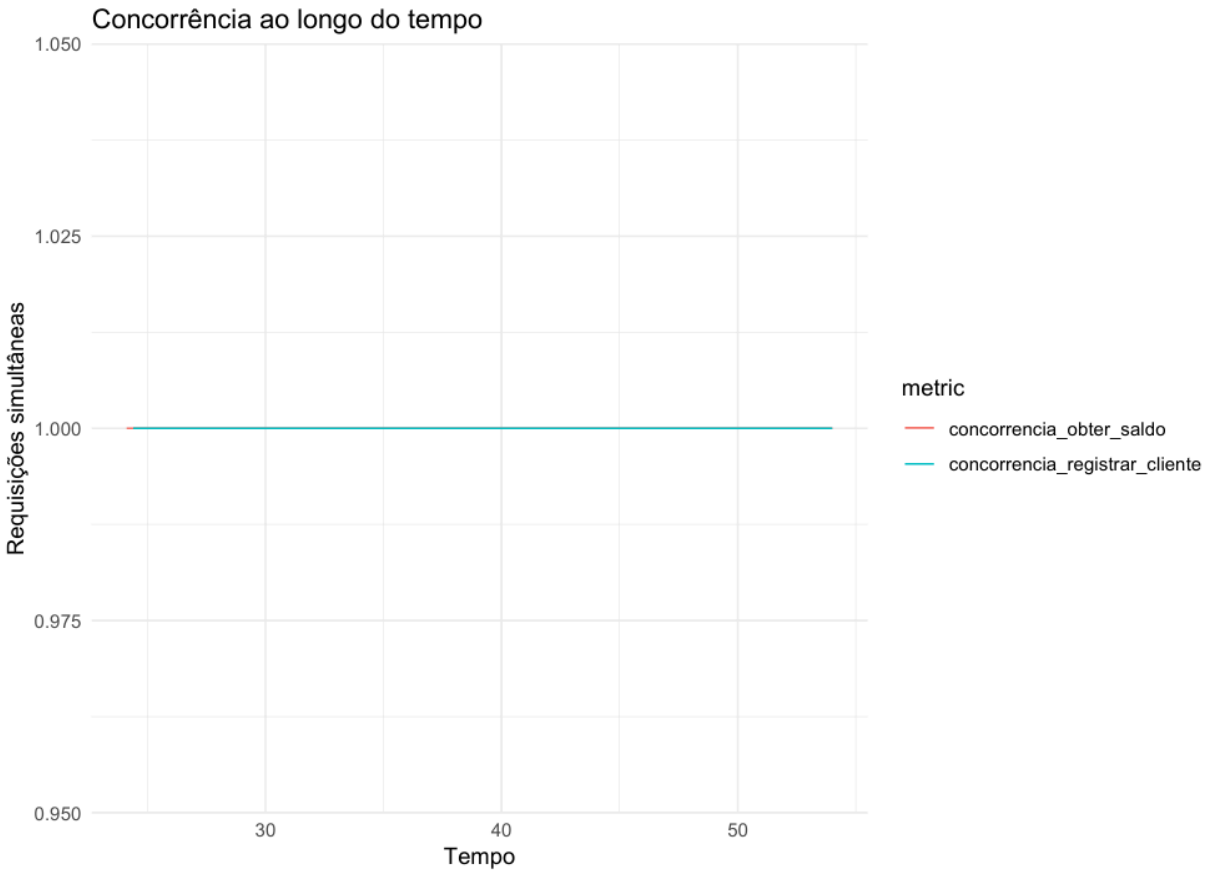


- Para este cenário foram feitas as seguintes análises:

Embora a média de requisições por segundo tenha sido de 12,36, o gráfico revela uma instabilidade considerável neste indicador, com flutuações frequentes ao longo do tempo. Essas variações sugerem que o sistema enfrenta dificuldades para manter um throughput constante, possivelmente devido a limitações na conexão com o banco de dados ou na alocação de recursos do servidor.

Tal instabilidade merece atenção, pois pode impactar diretamente a experiência do usuário final.

4.3. Gráfico de Concorrência X Tempo



- Para este cenário foram feitas as seguintes análises:

O gráfico de concorrência apresentou um comportamento estável para ambas as operações, mantendo-se em aproximadamente 1 requisição simultânea processada por vez. Este padrão indica que o sistema gerencia adequadamente a distribuição do processamento entre operações rápidas e lentas, sem acumular requisições em excesso.

5. CONCLUSÃO E IDENTIFICAÇÃO DE PONTOS CRÍTICOS

Os testes confirmaram a confiabilidade do sistema, com 100% das requisições processadas com sucesso. Entretanto, a análise identificou aspectos que requerem melhorias:

- Operações de registro com desempenho limitado: O tempo de resposta elevado no endpoint de registro indica possíveis gargalos, provavelmente relacionados a transações de banco de dados ou processos de validação.

- Instabilidade na vazão: A flutuação no número de requisições por segundo sugere dificuldades do sistema em manter performance consistente sob carga contínua.

- Capacidade limitada de processamento simultâneo: A baixa concorrência mantida durante os testes pode indicar restrições na arquitetura para aproveitar todo o potencial de processamento paralelo.

Próximas prioridades levantadas:

1. Otimizar as operações de registro, com foco nas transações de banco de dados

2. Revisar a configuração de pools de conexão e recursos compartilhados

3. Avaliar a implementação de processamento assíncrono para operações mais demoradas

4. Realizar testes de estresse para determinar os limites de escalabilidade do sistema

Em resumo, o sistema apresenta base sólida de funcionamento, mas requer ajustes específicos para melhorar o desempenho das operações de escrita e garantir uma vazão mais estável sob carga.