

Relatório de Testes de performance da API serverest

Cenário 1: Teste de Carga do fluxo de compra completo da API e-commerce

Objetivo: Avaliar a resposta da API com um número crescente de usuários simultâneos fazendo requisições 'http' para a aplicação do fluxo completo de criação de usuário, login com este usuário, cadastrar produto, cadastrar carrinho e finalizar a compra deletando o carrinho.

Usuários Virtuais: 20

Duração do Teste: 3 minutos por carga.

Principais métricas: Tempo médio de resposta e taxa de erro.

Os seguintes parâmetros serão utilizados:

- Checks status code 200
- Check response time menor que 3seg
- Average http requisition duration: <1000ms
- Http requisition fail request: < 0.01
- Capacidade de carga: suportar subida gradual até 20 usuários simultâneos e manter iterações desses usuários por 3 minutos.
- Escalabilidade: Planejar um aumento de 20% na capacidade para suportar crescimento futuro.

Resultados obtidos:

```

  AKG.io

execution: local
script: fluxo-comprou.js
output: -

scenarios: (100.00%) 1 scenario, 20 max VUs, 5m30s max duration (incl. graceful stop):
    * default: Up to 20 looping VUs for 5m0s over 3 stages (gracefulRampDown: 30s, gracefulStop: 30s)

✓ POST /usuarios status is 201
✓ POST /usuarios response time is acceptable
✓ POST /login status is 200
✓ POST /login response time is acceptable
✓ POST /produtos status is 201
✓ POST /produtos response time is acceptable
✓ POST /carrinhos status is 201
✓ POST /carrinhos response time is acceptable
✓ DELETE /carrinhos/concluir-comprou status is 200
✓ DELETE /carrinhos/concluir-comprou response time is acceptable

checks .....: 100.00% ✓ 7260 X 0
data_received.....: 1.9 MB 6.5 kB/s
data_sent.....: 1.2 MB 4.0 kB/s
http_req_blocked.....: avg=7.92µs min=0s med=0s max=13.99ms p(90)=0s p(95)=0s
http_req_connecting.....: avg=3.65µs min=0s med=0s max=1.1ms p(90)=0s p(95)=0s
✓ http_req_duration.....: avg=541.61ms min=16.49ms med=522.95ms max=1.78s p(90)=980.04ms p(95)=1.13s
  { expected_response:true }...: avg=541.61ms min=16.49ms med=522.95ms max=1.78s p(90)=980.04ms p(95)=1.13s
✓ http_req_failed.....: 0.00% ✓ 0 X 3630
http_req_receiving.....: avg=211.51µs min=0s med=0s max=1.1ms p(90)=877.04µs p(95)=999.5µs
http_req_sending.....: avg=3.36µs min=0s med=0s max=1.01ms p(90)=0s p(95)=0s
http_req_tls_handshaking.....: avg=0s min=0s med=0s max=0s p(90)=0s p(95)=0s
http_req_waiting.....: avg=541.4ms min=16.49ms med=522.95ms max=1.78s p(90)=979.04ms p(95)=1.13s
http_reqs.....: 3630 12.065478/s
iteration_duration.....: avg=6.73s min=4.31s med=7.01s max=8.57s p(90)=7.63s p(95)=7.8s
iterations.....: 726 2.413096/s
vus.....: 1 min=1 max=20
vus_max.....: 20 min=20 max=20

running (5m00.9s), 00/20 VUs, 726 complete and 0 interrupted iterations
default ✓ [=====] 00/20 VUs 5m0s
```

Iterations: 726

Duration: Traffic ramp-up from 1 to 20 users over 1 minutes. Stay at 20 users for 3 minutes. ramp-down to 0 users

http requests duration: average: 541.61ms

http requests failed: 0%

Status code: 200

Response duration < 3sec (x) ✓ | () ✗

Cenário 2: Teste de estresse do fluxo de compra completo da API e-commerce

Objetivo: Avaliar a resposta da API com um número crescente de usuários simultâneos fazendo requisições 'http' para estressar a aplicação do fluxo completo de criação de usuário, login com este usuário, cadastrar produto, cadastrar carrinho e finalizar a compra deletando o carrinho.

Usuários Virtuais: 50

Duração do Teste: 15 minutos por carga.

Principais métricas: Tempo médio de resposta e taxa de erro.

Os seguintes parâmetros serão utilizados:

- Checks status code 200
- Check response time menor que 3seg
- Average http requisition duration: <1000ms
- Http requisition fail request: < 0.01
- Capacidade de carga: suportar subida para 40 usuários simultâneos e manter iterações desses usuários por 10 minutos.

Resultado:

```

  MK6.io

execution: local
script: fluxo-comp.ra.js
output: -

scenarios: (100.00%) 1 scenario, 50 max VUs, 15m30s max duration (incl. graceful stop):
  * default: Up to 50 looping VUs for 15m0s over 3 stages (gracefulRampDown: 30s, gracefulStop: 30s)

✓ POST /usuarios status is 201
✗ POST /usuarios response time is acceptable
  ↳ 25% - ✓ 532 / ✗ 1526
✓ POST /login status is 200
✓ POST /login response time is acceptable
✓ POST /produtos status is 201
✗ POST /produtos response time is acceptable
  ↳ 24% - ✓ 496 / ✗ 1562
✗ POST /carrinhos status is 201
  ↳ 99% - ✓ 2057 / ✗ 1
✗ POST /carrinhos response time is acceptable
  ↳ 21% - ✓ 445 / ✗ 1613
✓ DELETE /carrinhos/concluir-comp.ra status is 200
✗ DELETE /carrinhos/concluir-comp.ra response time is acceptable
  ↳ 25% - ✓ 515 / ✗ 1543

checks.....: 69.65% ✓ 14335 ✗ 6245
data_received.....: 5.5 MB 6.1 kB/s
data_sent.....: 3.4 MB 3.8 kB/s
http_req_blocked.....: avg=5.62µs min=0s med=0s max=13.05ms p(90)=0s p(95)=0s
http_req_connecting.....: avg=2.8µs min=0s med=0s max=1.22ms p(90)=0s p(95)=0s
✗ http_req_duration.....: avg=2.87s min=22.02ms med=3.5s max=5.03s p(90)=4.09s p(95)=4.27s
  { expected_response:true }...: avg=2.86s min=22.02ms med=3.5s max=4.98s p(90)=4.09s p(95)=4.27s
✓ http_req_failed.....: 0.00% ✓ 1 ✗ 10289
http_req_receiving.....: avg=188.74µs min=0s med=0s max=1.15ms p(90)=872.8µs p(95)=999.6µs
http_req_sending.....: avg=2.23µs min=0s med=0s max=969.5µs p(90)=0s p(95)=0s
http_req_tls_handshaking.....: avg=0s min=0s med=0s max=0s p(90)=0s p(95)=0s
http_req_waiting.....: avg=2.86s min=21.08ms med=3.5s max=5.03s p(90)=4.09s p(95)=4.27s
http_reqs.....: 10290 11.38684/s
iteration_duration.....: avg=18.38s min=4.42s med=20.85s max=22.35s p(90)=21.59s p(95)=21.74s
iterations.....: 2058 2.277368/s
vus.....: 1 min=1 max=50
vus_max.....: 50 min=50 max=50

running (15m03.7s), 00/50 VUs, 2058 complete and 0 interrupted iterations
default ✓ [=====] 00/50 VUs 15m0s
ERRO[0904] thresholds on metrics 'http_req_duration' have been crossed
```

Iterations: 2058

Duration: traffic ramp-up from 1 to a higher 40 users over 3 minutes. stay at higher 40 users for 10 minutes. 2 min for ramp-down to 0 users

http requests duration: average: 2.87seg

http requests failed: 0%

Status code: 200

Response duration < 3sec () ✓ | (x) ✗

Comentários: Durante o teste uma requisição POST /carrinhos falhou, rota mais problemática em termos de performance. Com status code 408 Request Failed.

```
Windows PowerShell
POST /produtos 201 4434.329 ms - 82
POST /usuarios 201 3712.505 ms - 82
DELETE /carrinhos/concluir-compra 200 3829.885 ms - 51
POST /login 200 1838.731 ms - 277
POST /carrinhos 408 5000.867 ms - 53
Error [ERR_HTTP_HEADERS_SENT]: Cannot set headers after they are sent to the client
    at ServerResponse.setHeader (node:_http_outgoing:652:11)
    at ServerResponse.header (C:\Users\gabri\AppData\Local\npm-cache\npx\7083535e72394f6b\node_modules\express\lib\response.js:795:10)
    at ServerResponse.send (C:\Users\gabri\AppData\Local\npm-cache\npx\7083535e72394f6b\node_modules\express\lib\response.js:175:12)
    at ServerResponse.json (C:\Users\gabri\AppData\Local\npm-cache\npx\7083535e72394f6b\node_modules\express\lib\response.js:279:15)
    at ServerResponse.json (C:\Users\gabri\AppData\Local\npm-cache\npx\7083535e72394f6b\node_modules\dd-trace\packages\datadog-instrumentations\src\express.js:34:17)
    at ServerResponse.send (C:\Users\gabri\AppData\Local\npm-cache\npx\7083535e72394f6b\node_modules\express\lib\response.js:163:21)
    at exports.post (C:\Users\gabri\AppData\Local\npm-cache\npx\7083535e72394f6b\node_modules\serverest\src\controllers\carrinhos-controller.js:43:19)
POST /produtos 201 4256.673 ms - 82
POST /login 200 1951.069 ms - 277
POST /usuarios 201 3778.141 ms - 82
POST /login 200 1985.336 ms - 277
POST /carrinhos 201 4883.168 ms - 82
POST /produtos 201 4257.948 ms - 82
DELETE /carrinhos/concluir-compra 200 3958.803 ms - 51
DELETE /carrinhos/concluir-compra 200 3964.246 ms - 51
POST /usuarios 201 3733.735 ms - 82
POST /usuarios 201 3605.396 ms - 82
POST /produtos 201 4403.973 ms - 82
```

Cenário 3: Teste de resistência do fluxo de compra completo da API e-commerce

Objetivo: Avaliar a resposta da API com um número crescente de usuários simultâneos fazendo requisições 'http' por um tempo específico para testar a resistência da aplicação do fluxo completo de criação de usuário, login com este usuário, cadastrar produto, cadastrar carrinho e finalizar a compra deletando o carrinho.

Usuários Virtuais: 30

Duração do Teste: 19 minutos por carga.

Principais métricas: Tempo médio de resposta e taxa de erro.

Os seguintes parâmetros serão utilizados:

- Checks status code 200

- Check response time menor que 3seg
- Average http requisition duration: <1000ms
- Http requisition fail request: < 0.01
- Capacidade de carga: suportar subida gradual de 30 usuários simultâneos e manter iterações desses usuários por 15 minutos.

Resultado:

```

MKG.io

execution: local
script: fluxo-compra.js
output: -

scenarios: (100.00%) 1 scenario, 30 max VUs, 19m30s max duration (incl. graceful stop):
    * default: Up to 30 looping VUs for 19m0s over 3 stages (gracefulRampDown: 30s, gracefulStop: 30s)

✓ POST /usuarios status is 201
X POST /usuarios response time is acceptable
  ↳ 95% - ✓ 1818 / X 93
✓ POST /login status is 200
✓ POST /login response time is acceptable
✓ POST /produtos status is 201
X POST /produtos response time is acceptable
  ↳ 62% - ✓ 1196 / X 715
X POST /carrinhos status is 201
  ↳ 99% - ✓ 1910 / X 1
X POST /carrinhos response time is acceptable
  ↳ 23% - ✓ 458 / X 1453
✓ DELETE /carrinhos/concluir-compra status is 200
X DELETE /carrinhos/concluir-compra response time is acceptable
  ↳ 74% - ✓ 1415 / X 496

checks.....: 85.56% ✓ 16352    X 2758
data_received.....: 5.1 MB 4.5 kB/s
data_sent.....: 3.2 MB 2.8 kB/s
http_req_blocked.....: avg=3.79µs    min=0s    med=0s    max=12.66ms p(90)=0s    p(95)=0s
http_req_connecting.....: avg=1.8µs    min=0s    med=0s    max=1.08ms p(90)=0s    p(95)=0s
X http_req_duration.....: avg=2.41s    min=41.48ms med=2.72s max=4.47s p(90)=3.37s p(95)=3.58s
  { expected_response:true }...: avg=2.41s    min=41.48ms med=2.72s max=4.47s p(90)=3.37s p(95)=3.58s
✓ http_req_failed.....: 0.01% ✓ 1    X 9554
http_req_receiving.....: avg=197.38µs min=0s    med=0s    max=1.15ms p(90)=879.22µs p(95)=999.5µs
http_req_sending.....: avg=2.73µs    min=0s    med=0s    max=842.5µs p(90)=0s    p(95)=0s
http_req_tls_handshaking.....: avg=0s    min=0s    med=0s    max=0s p(90)=0s    p(95)=0s
http_req_waiting.....: avg=2.41s    min=41.48ms med=2.72s max=4.47s p(90)=3.37s p(95)=3.58s
http_reqs.....: 9555    8.347278/s
iteration_duration.....: avg=16.12s    min=4.67s    med=17.16s max=18.9s p(90)=17.95s p(95)=18.16s
iterations.....: 1911    1.669456/s
vus.....: 1    min=1    max=30
vus_max.....: 30    min=30    max=30

running (19m04.7s), 00/30 VUs, 1911 complete and 0 interrupted iterations
default ✓ [=====] 00/30 VUs 19m0s
ERRO[1145] thresholds on metrics 'http_req_duration' have been crossed

```

Iterations: 1911

Duration: traffic ramp-up from 1 to 30 users over 5 minutes. stay at 30 users for 15 minutes. ramp-down to 0 users

http requests duration: average: 2.87ms

http requests failed: 0%

Status code: 200

Response duration < 3sec () ✓ | (x) ✗

Comentários: Durante o teste uma requisição POST /carrinhos falhou, uma das rota mais problemática em termos de performance, retornou com status code 408 Request Failed.

Cenário 4: Teste de pico do fluxo de compra completo da API e-commerce

Objetivo: Avaliar a resposta da API com um número crescente e espontâneo de usuários simultâneos fazendo requisições 'http' em um período rápido para a aplicação do fluxo completo de criação de usuário, login com este usuário, cadastrar produto, cadastrar carrinho e finalizar a compra deletando o carrinho.

Usuários Virtuais: 20

Duração do Teste: 2 minutos por carga.

Principais métricas: Tempo médio de resposta e taxa de erro.

Os seguintes parâmetros serão utilizados:

- Checks status code 200
- Check response time menor que 3seg
- Average http requisition duration: <1000ms
- Http requisition fail request: < 0.01
- Capacidade de carga: suportar subida gradual de 30 usuários simultâneos e manter iterações desses usuários por 15 minutos.

Resultado:



execution: local
script: fluxo-compra.js
output: -

scenarios: (100.00%) 1 scenario, 20 max VUs, 2m30s max duration (incl. graceful stop):
* default: Up to 20 looping VUs for 2m0s over 2 stages (gracefulRampDown: 30s, gracefulStop: 30s)

✓ POST /usuarios status is 201
✓ POST /usuarios response time is acceptable
✓ POST /login status is 200
✓ POST /login response time is acceptable
✓ POST /produtos status is 201
✓ POST /produtos response time is acceptable
✓ POST /carrinhos status is 201
✗ POST /carrinhos response time is acceptable
 ↳ 85% - ✓ 118 / ✗ 20
✓ DELETE /carrinhos/concluir-compra status is 200
✓ DELETE /carrinhos/concluir-compra response time is acceptable

checks.....: 98.55% ✓ 1360 ✗ 20
data_received.....: 370 kB 3.0 kB/s
data_sent.....: 228 kB 1.8 kB/s
http_req_blocked.....: avg=39.68µs min=0s med=0s max=14.07ms p(90)=0s p(95)=0s
http_req_connecting.....: avg=17.18µs min=0s med=0s max=1.11ms p(90)=0s p(95)=0s
✗ http_req_duration.....: avg=1.12s min=37.53ms med=966.07ms max=3.78s p(90)=2.24s p(95)=2.6s
 { expected_response:true }...: avg=1.12s min=37.53ms med=966.07ms max=3.78s p(90)=2.24s p(95)=2.6s
✓ http_req_failed.....: 0.00% ✓ 0 ✗ 690
http_req_receiving.....: avg=202.64µs min=0s med=0s max=1.14ms p(90)=809.5µs p(95)=999.81µs
http_req_sending.....: avg=3.13µs min=0s med=0s max=563.1µs p(90)=0s p(95)=0s
http_req_tls_handshaking.....: avg=0s min=0s med=0s max=0s p(90)=0s p(95)=0s
http_req_waiting.....: avg=1.12s min=37.53ms med=966.05ms max=3.78s p(90)=2.24s p(95)=2.6s
http_reqs.....: 690 5.554386/s
iteration_duration.....: avg=9.68s min=4.78s med=9.33s max=14.99s p(90)=14.21s p(95)=14.46s
iterations.....: 138 1.110877/s
vus.....: 1 min=1 max=20
vus_max.....: 20 min=20 max=20

running (2m04.2s), 00/20 VUs, 138 complete and 0 interrupted iterations
default ✓ [=====] 00/20 VUs 2m0s
ERRO[0124] thresholds on metrics 'http_req_duration' have been crossed

Iterations: 138

Duration: fast ramp-up to a high point, quick ramp-down to 0 users

http requests duration: average: 1.12s

http requests failed: 0%

Status code: 200

Response duration < 3sec (x) ✓ | () ✗

Cenário 5: Teste de fumaça do fluxo de compra completo da API e-commerce

Objetivo: Avaliar a resposta da API com um número mínimo de usuários simultâneos fazendo requisições 'http' por um tempo curto para testar o funcionamento do fluxo completo de criação de usuário, login com este usuário, cadastrar produto, cadastrar carrinho e finalizar a compra deletando o carrinho.

Principais métricas: Tempo médio de resposta e taxa de erro.

Resultado:

[illegible]

Duration: Traffic ramp-up from 1. Key for Smoke test. Keep max 5 VUs. Duration can be shorter or just a few iterations.

http requests duration: average: 168.08ms

http requests failed: 0%

Status code: 200

Response duration < 3sec (x) ✓ | () ✗

Cenário 6: Teste de Carga de POST da rota /usuários

Objetivo: Avaliar a resposta da API com um número crescente de usuários simultâneos.

Usuários Virtuais: 20

Duração do Teste: 3 minutos por carga.

Principais métricas: Tempo médio de resposta e taxa de erro.

Os seguintes parâmetros serão utilizados:

- Checks status code 200
- Check response time menor que 3seg
- Average http requisition duration: <1000ms
- Http requisition fail request: < 0.01
- Capacidade de carga: suportar subida gradual até 20 usuários simultâneos e manter iterações desses usuários por 3 minutos.

Resultado obtido:

```
PS C:\Users\gabriel\Desktop> k6 serverest performance --project-base-k6 \tests> k6 run postUsuarios.js
```

```

      /\_/\
     /  _  \ io
    /_____\

execution: local
script:   postUsuarios.js
output: -

scenarios: (100.00%) 1 scenario, 20 max VUs, 5m30s max duration (incl. graceful stop):
    * default: Up to 20 looping VUs for 5m0s over 3 stages (gracefulRampDown: 30s, gracefulStop: 30s)


✓ POST /usuarios status was 201
✓ POST /usuarios response time is acceptable


checks.....: 100.00% ✓ 7714          X 0
data_received.....: 1.9 MB   6.5 kB/s
data_sent.....: 922 kB   3.1 kB/s
http_req_blocked.....: avg=8.26µs   min=0s       med=0s        max=15.72ms   p(90)=0s      p(95)=0s
http_req_connecting.....: avg=3.32µs   min=0s       med=0s        max=1.12ms    p(90)=0s      p(95)=0s
✓ http_req_duration.....: avg=247.28ms min=47.06ms  med=235.27ms  max=861.18ms  p(90)=418.14ms p(95)=480.01ms
  { expected_response:true }...: avg=247.28ms min=47.06ms  med=235.27ms  max=861.18ms  p(90)=418.14ms p(95)=480.01ms
✓ http_req_failed.....: 0.00% ✓ 0           X 3857
http_req_receiving.....: avg=169.03µs min=0s       med=0s        max=2.11ms    p(90)=666.18µs p(95)=997.32µs
http_req_sending.....: avg=3.3µs    min=0s       med=0s        max=1.55ms    p(90)=0s      p(95)=0s
http_req_tls_handshaking.....: avg=0s       min=0s       med=0s        max=0s        p(90)=0s      p(95)=0s
http_req_waiting.....: avg=247.11ms min=47.06ms  med=235.22ms  max=861.18ms  p(90)=417.99ms p(95)=480.01ms
http_reqs.....: 3857    12.812567/s
iteration_duration.....: avg=1.25s    min=1.04s    med=1.24s     max=1.87s     p(90)=1.42s   p(95)=1.48s
iterations.....: 3857    12.812567/s
vus.....: 1         min=1         max=20
vus_max.....: 20      min=20        max=20


running (5m01.0s), 00/20 VUs, 3857 complete and 0 interrupted iterations
default ✓ [=====] 1 00/20 VUs  5m0s
```

Iterations: 3857

Duration: Traffic ramp-up from 1 to 20 users over 1 minutes. Stay at 20 users for 3 minutes. ramp-down to 0 users

http requests duration: average: 247.28ms

http requests failed: 0%

Status code: 200

Response duration < 3sec (x) ✓ | () ✗

Cenário 7: Teste de estresse de POST rota /usuários

Objetivo: Avaliar a resposta da API com um número crescente de usuários simultâneos fazendo requisições 'http' para estressar a aplicação do fluxo completo de criação de usuário.

Usuários Virtuais: 50

Duração do Teste: 15 minutos por carga.

Principais métricas: Tempo médio de resposta e taxa de erro.

Os seguintes parâmetros serão utilizados:

- Checks status code 200
- Check response time menor que 3seg
- Average http requisition duration: <1000ms
- Http requisition fail request: < 0.01
- Capacidade de carga: suportar subida para 40 usuários simultâneos e manter iterações desses usuários por 10 minutos.

Resultado:

```
PS C:\Users\gabri\Desktop\k6 serverest performance\projeto-base-k6\tests> k6 run postUsuarios.js

      /\_/\
     /  _  \
    /_____\ \
   /         \
  /           \
 /             \
/               \
\               /
 \             /
  \           /
   \         /
    \_____/

execution: local
script: postUsuarios.js
output: -

scenarios: (100.00%) 1 scenario, 50 max VUs, 15m30s max duration (incl. graceful stop):
  * default: Up to 50 looping VUs for 15m0s over 3 stages (gracefulRampDown: 30s, gracefulStop: 30s)

✓ POST /usuarios status is 201
✗ POST /usuarios response time is acceptable
  ↳ 79% - ✓ 8944 / ✗ 2350

checks.....: 89.59% ✓ 20238 ✗ 2350
data_received.....: 5.7 MB 6.3 kB/s
data_sent.....: 2.7 MB 3.0 kB/s
http_req_blocked.....: avg=3.9µs min=0s med=0s max=9.93ms p(90)=0s p(95)=0s
http_req_connecting.....: avg=2.78µs min=0s med=0s max=2.64ms p(90)=0s p(95)=0s
✗ http_req_duration.....: avg=2.32s min=67.05ms med=2.59s max=4.53s p(90)=3.27s p(95)=3.5s
  { expected_response:true }...: avg=2.32s min=67.05ms med=2.59s max=4.53s p(90)=3.27s p(95)=3.5s
✓ http_req_failed.....: 0.00% ✓ 0 ✗ 11294
http_req_receiving.....: avg=152.3µs min=0s med=0s max=1.3ms p(90)=565.47µs p(95)=831µs
http_req_sending.....: avg=2.14µs min=0s med=0s max=1.99ms p(90)=0s p(95)=0s
http_req_tls_handshaking.....: avg=0s min=0s med=0s max=0s p(90)=0s p(95)=0s
http_req_waiting.....: avg=2.32s min=67.05ms med=2.59s max=4.53s p(90)=3.27s p(95)=3.5s
http_reqs.....: 11294 12.547647/s
iteration_duration.....: avg=3.33s min=1.07s med=3.59s max=5.54s p(90)=4.28s p(95)=4.51s
iterations.....: 11294 12.547647/s
vus.....: 1 min=1 max=50
vus_max.....: 50 min=50 max=50

running (15m00.1s), 00/50 VUs, 11294 complete and 0 interrupted iterations
default ✓ [=====] 00/50 VUs 15m0s
ERRO[0900] thresholds on metrics 'http_req_duration' have been crossed
```

Iterations: 2058

Duration: traffic ramp-up from 1 to a higher 40 users over 3 minutes. stay at higher 40 users for 10 minutes. 2 min for ramp-down to 0 users

http requests duration: average: 2.32seg

http requests failed: 0%

Status code: 200

Response duration < 3sec () ✓ | (x) ✗

Comentários: Durante o teste uma requisição POST /usuários só cumpriu cerca de 80% a regra de 3 segundos de tempo máximo para a resposta da requisição, mas já que a média está em 2seg e o percentil(90) em 3.27, foi considerado aceitável e não será relatado como bug esse cenário de estresse.

Cenário 8: Teste de resistência de POST rota /usuários

Objetivo: Avaliar a resposta da API com um número crescente de usuários simultâneos fazendo requisições 'http' por um tempo específico para testar a resistência da aplicação do fluxo completo de criação de usuário.

Usuários Virtuais: 30

Duração do Teste: 19 minutos por carga.

Principais métricas: Tempo médio de resposta e taxa de erro.

Os seguintes parâmetros serão utilizados:

- Checks status code 200
- Check response time menor que 3seg
- Average http requisition duration: <1000ms
- Http requisition fail request: < 0.01
- Capacidade de carga: suportar subida gradual de 30 usuários simultâneos e manter iterações desses usuários por 15 minutos.

Resultado:

```

PS C:\Users\gabri\Desktop\k6 serverest performance\projeto-base-k6\tests> k6 run postUsuarios.js

      M K6 .io

execution: local
  script: postUsuarios.js
  output: -

scenarios: (100.00%) 1 scenario, 25 max VUs, 19m30s max duration (incl. graceful stop):
  * default: Up to 25 looping VUs for 19m0s over 3 stages (gracefulRampDown: 30s, gracefulStop: 30s)

✓ POST /usuarios status is 201
✓ POST /usuarios response time is acceptable

checks.....: 100.00% ✓ 21974      ✗ 0
data_received.....: 5.5 MB  4.9 kB/s
data_sent.....: 2.6 MB  2.3 kB/s
http_req_blocked.....: avg=2.64µs  min=0s      med=0s      max=14.02ms p(90)=0s    p(95)=0s
http_req_connecting.....: avg=1.15µs  min=0s      med=0s      max=826.3µs p(90)=0s    p(95)=0s
✗ http_req_duration.....: avg=1.32s   min=87.64ms med=1.37s   max=2.61s   p(90)=1.95s p(95)=2.06s
  { expected_response:true }...: avg=1.32s   min=87.64ms med=1.37s   max=2.61s   p(90)=1.95s p(95)=2.06s
✓ http_req_failed.....: 0.00% ✓ 0      ✗ 10987
http_req_receiving.....: avg=159.98µs min=0s      med=0s      max=1.79ms  p(90)=582.4µs p(95)=820.37µs
http_req_sending.....: avg=1.92µs  min=0s      med=0s      max=1.11ms  p(90)=0s     p(95)=0s
http_req_tls_handshaking.....: avg=0s      min=0s      med=0s      max=0s      p(90)=0s     p(95)=0s
http_req_waiting.....: avg=1.32s   min=87.64ms med=1.37s   max=2.61s   p(90)=1.95s p(95)=2.06s
http_reqs.....: 10987  9.633896/s
iteration_duration.....: avg=2.32s   min=1.09s   med=2.38s   max=3.62s   p(90)=2.95s p(95)=3.07s
iterations.....: 10987  9.633896/s
vus.....: 1      min=1      max=25
vus_max.....: 25      min=25     max=25

running (19m00.5s), 00/25 VUs, 10987 complete and 0 interrupted iterations
default ✓ [=====] 00/25 VUs  19m0s
ERRO[1141] thresholds on metrics 'http_req_duration' have been crossed

```

Iterations: 10987

Duration: traffic ramp-up from 1 to 30 users over 5 minutes. stay at 30 users for 15 minutes.
ramp-down to 0 users

http requests duration: average: 1.32s

http requests failed: 0%

Status code: 200

Response duration < 3sec (x) ✓ | () ✗

Comentários: A duração da requisição http ultrapassou um pouco a métrica de 1 segundo mas não foi considerado como bug pelo resultado ser um valor muito próximo.

Cenário 9: Teste de pico da POST rota /usuários

Objetivo: Avaliar a resposta da API com um número crescente e espontâneo de usuários simultâneos fazendo requisições 'http' em um período rápido para a aplicação do fluxo completo de criação de usuário.

Status code: 200

Response duration < 3sec (x) ✓ | 0 ✗

Cenário 10: Teste de fumaça da POST rota /usuários

Objetivo: Avaliar a resposta da API com um número mínimo de usuários simultâneos fazendo requisições 'http' por um tempo curto para testar o funcionamento do fluxo completo de criação de usuário, login com este usuário.

Usuários Virtuais: 5

Duração do Teste: 9 segundos por carga.

Principais métricas: Tempo médio de resposta e taxa de erro.

Os seguintes parâmetros serão utilizados:

- Checks status code 200
- Check response time menor que 3seg
- Average http requisition duration: <800ms
- Http requisition fail request: < 0.01
- Capacidade de carga: suportar subida gradual de 30 usuários simultâneos e manter iterações desses usuários por 15 minutos.

Resultado:

```
PS C:\Users\gabri\Desktop\k6 serverest performance\projeto-base-k6\tests> k6 run postUsuarios.js

      M K6
     .io

execution: local
  script: postUsuarios.js
  output: -
scenarios: (100.00%) 1 scenario, 5 max VUs, 39s max duration (incl. graceful stop):
            * default: Up to 5 looping VUs for 9s over 3 stages (gracefulRampDown: 30s, gracefulStop: 30s)

✓ POST /usuarios status was 201
✓ POST /usuarios response time is acceptable

checks.....: 100.00% ✓ 42      ✗ 0
data_received.....: 11 kB    1.1 kB/s
data_sent.....: 5.0 kB    500 B/s
http_req_blocked.....: avg=956.69µs min=0s      med=0s      max=16.69ms p(90)=1.15ms p(95)=1.2ms
http_req_connecting.....: avg=156.08µs min=0s      med=0s      max=1.15ms  p(90)=525.8µs p(95)=560.69µs
✓ http_req_duration.....: avg=98.38ms  min=53.55ms med=86.25ms max=169.18ms p(90)=152.65ms p(95)=153.7ms
  { expected_response:true }...: avg=98.38ms  min=53.55ms med=86.25ms max=169.18ms p(90)=152.65ms p(95)=153.7ms
✓ http_req_failed.....: 0.00% ✓ 0      ✗ 21
http_req_receiving.....: avg=257.77µs min=0s      med=178.9µs max=1.06ms  p(90)=572.2µs p(95)=662.4µs
http_req_sending.....: avg=0s       min=0s      med=0s      max=0s      p(90)=0s     p(95)=0s
http_req_tls_handshaking.....: avg=0s       min=0s      med=0s      max=0s      p(90)=0s     p(95)=0s
http_req_waiting.....: avg=98.12ms  min=53.55ms med=86.25ms max=169.18ms p(90)=152.41ms p(95)=153.16ms
http_reqs.....: 21      2.092868/s
iteration_duration.....: avg=1.1s     min=1.05s   med=1.09s   max=1.18s   p(90)=1.15s   p(95)=1.16s
iterations.....: 21      2.092868/s
vus.....: 1      min=1      max=5
vus_max.....: 5      min=5      max=5

running (10.0s), 0/5 VUs, 21 complete and 0 interrupted iterations
default ✓ [=====] 0/5 VUs  9s
```

Iterations: 21

Duration: Traffic ramp-up from 1. Key for Smoke test. Keep max 5 VUs. Duration can be shorter or just a few iterations.

http requests duration: average: 98.38ms

http requests failed: 0%

Status code: 200

Response duration < 3sec (x) ✓ | () ✗

Cenário 11: Teste de Carga de POST da rota /produtos

Objetivo: Avaliar a resposta da API com um número crescente de usuários simultâneos fazendo requisições 'http' para a aplicação cadastrar produto.

Usuários Virtuais: 20

Duração do Teste: 3 minutos por carga.


Principais métricas: Tempo médio de resposta e taxa de erro.

Os seguintes parâmetros serão utilizados:

- Checks status code 200
- Check response time menor que 3seg
- Average http requisition duration: <1000ms
- Http requisition fail request: < 0.01
- Capacidade de carga: suportar subida gradual até 20 usuários simultâneos e manter iterações desses usuários por 3 minutos.
- Escalabilidade: Planejar um aumento de 20% na capacidade para suportar crescimento futuro.

Resultados obtidos:

```
PS C:\Users\gabriel\Documents\projeto-base-k6\tests> k6 run postProdutos.js
```



```

execution: local
  script: postProdutos.js
  output: -

scenarios: (100.00%) 1 scenario, 20 max VUs, 5m30s max duration (incl. graceful stop):
    * default: Up to 20 looping VUs for 5m0s over 3 stages (gracefulRampDown: 30s, gracefulStop: 30s)

✓ POST /produtos status is 201
✓ POST /produtos response time is acceptable

checks.....: 100.00% ✓ 2150      ✗ 0
data_received.....: 1.8 MB   6.1 kB/s
data_sent.....: 963 kB   3.2 kB/s
http_req_blocked.....: avg=8.52µs   min=0s       med=0s       max=13.98ms  p(90)=0s     p(95)=0s
http_req_connecting.....: avg=3.4µs    min=0s       med=0s       max=927.2µs  p(90)=0s     p(95)=0s
✓ http_req_duration.....: avg=499.24ms min=17.49ms  med=426.07ms max=1.59s    p(90)=997.73ms p(95)=1.1s
  { expected_response:true }...: avg=499.24ms min=17.49ms  med=426.07ms max=1.59s    p(90)=997.73ms p(95)=1.1s
✓ http_req_failed.....: 0.00% ✓ 0      ✗ 3225
http_req_receiving.....: avg=208.56µs min=0s       med=0s       max=1.53ms   p(90)=886.87µs p(95)=999.7µs
http_req_sending.....: avg=2.04µs   min=0s       med=0s       max=960.8µs  p(90)=0s     p(95)=0s
http_req_tls_handshaking.....: avg=0s       min=0s       med=0s       max=0s       p(90)=0s     p(95)=0s
http_req_waiting.....: avg=499.03ms min=16.49ms  med=425.52ms max=1.59s    p(90)=997.06ms p(95)=1.1s
http_reqs.....: 3225    10.731668/s
iteration_duration.....: avg=4.52s    min=3.24s    med=4.7s     max=5.26s    p(90)=4.97s   p(95)=5.04s
iterations.....: 1075    3.577223/s
vus.....: 1      min=1      max=20
vus_max.....: 20    min=20     max=20

running (5m00.5s), 00/20 VUs, 1075 complete and 0 interrupted iterations
default ✓ [=====] 00/20 VUs  5m0s
```

Iterations: 1075

Duration: Traffic ramp-up from 1 to 20 users over 1 minutes. Stay at 20 users for 3 minutes. ramp-down to 0 users

http requests duration: average: 499.03ms

http requests failed: 0%

Status code: 200

Response duration < 3sec (x) ✓ | () ✗

Cenário 12: Teste de estresse de POST rota /produtos

Objetivo: Avaliar a resposta da API com um número crescente de usuários simultâneos fazendo requisições 'http' para estressar o cadastrar produto.

Usuários Virtuais: 40

Duração do Teste: 15 minutos por carga.

Principais métricas: Tempo médio de resposta e taxa de erro.

Os seguintes parâmetros serão utilizados:

- Checks status code 200
- Check response time menor que 3seg
- Average http requisition duration: <1000ms
- Http requisition fail request: < 0.01
- Capacidade de carga: suportar subida para 40 usuários simultâneos e manter iterações desses usuários por 10 minutos.

Resultados obtidos:

```
PS C:\Users\gabri\Desktop\k6 serverest performance\projeto-base-k6\tests> k6 run postProdutos.js

      M K6
     .io

execution: local
script: postProdutos.js
output: -

scenarios: (100.00%) 1 scenario, 40 max VUs, 15m30s max duration (incl. graceful stop):
 * default: Up to 40 looping VUs for 15m0s over 3 stages (gracefulRampDown: 30s, gracefulStop: 30s)

WARN[0373] Request Failed                      error="Post \"http://localhost:3000/usuarios\": read tcp 127.0.0.1:57122->127.0.0.1:3000: wsarecv: Foi forçado o cancela
elo host remoto."
WARN[0398] Request Failed                      error="Post \"http://localhost:3000/usuarios\": read tcp 127.0.0.1:57158->127.0.0.1:3000: wsarecv: Foi forçado o cancela
elo host remoto."
WARN[0411] Request Failed                      error="Post \"http://localhost:3000/usuarios\": read tcp 127.0.0.1:57172->127.0.0.1:3000: wsarecv: Foi forçado o cancela
elo host remoto."
WARN[0411] Request Failed                      error="Post \"http://localhost:3000/usuarios\": EOF"
WARN[0411] Request Failed                      error="Post \"http://localhost:3000/usuarios\": EOF"

X POST /produtos status is 201
  ↳ 92% - ✓ 2497 / X 192
X POST /produtos response time is acceptable
  ↳ 23% - ✓ 632 / X 2057

checks.....: 58.18% ✓ 3129 X 2249
data_received.....: 4.6 MB 5.1 kB/s
data_sent.....: 2.4 MB 2.7 kB/s
http_req_blocked.....: avg=20.83µs min=0s med=0s max=14.8ms p(90)=0s p(95)=0s
http_req_connecting.....: avg=15.11µs min=0s med=0s max=1.39ms p(90)=0s p(95)=0s
X http_req_duration.....: avg=2.73s min=509.6µs med=2.89s max=5.14s p(90)=4.53s p(95)=4.79s
  { expected_response:true }...: avg=2.68s min=26.08ms med=2.82s max=5.07s p(90)=4.44s p(95)=4.66s
X http_req_failed.....: 2.57% ✓ 206 X 7859
http_req_receiving.....: avg=201.3µs min=0s med=0s max=1.25ms p(90)=894.34µs p(95)=999.6µs
http_req_sending.....: avg=1.83µs min=0s med=0s max=2.05ms p(90)=0s p(95)=0s
http_req_tls_handshaking.....: avg=0s min=0s med=0s max=0s p(90)=0s p(95)=0s
http_req_waiting.....: avg=2.73s min=509.6µs med=2.89s max=5.14s p(90)=4.53s p(95)=4.79s
http_reqs.....: 8067 8.929349/s
iteration_duration.....: avg=11.23s min=3.29s med=12.58s max=13.7s p(90)=13.08s p(95)=13.21s
iterations.....: 2689 2.97645/s
vus.....: 1 min=1 max=40
vus_max.....: 40 min=40 max=40

running (15m03.4s), 00/40 VUs, 2689 complete and 0 interrupted iterations
default / [=====] 00/40 VUs 15m0s
ERRO[0904] thresholds on metrics 'http_req_duration, http_req_failed' have been crossed
PS C:\Users\gabri\Desktop\k6 serverest performance\projeto-base-k6\tests> |
```

Iterations: 2689

Duration: traffic ramp-up from 1 to a higher 40 users over 3 minutes. stay at higher 40 users for 10 minutes. 2 min for ramp-down to 0 users

http requests duration: average: 2.68s

http requests failed: 0%

Status code: 200

Response duration < 3sec () ✓ | (x) ✗

Comentários: Durante o teste uma requisição POST /produtos falhou, uma das rota mais problemática em termos de performance, retornou com status code 408 Request Failed.

Cenário 13: Teste de resistência de POST rota /produtos

Objetivo: Avaliar a resposta da API com um número crescente de usuários simultâneos fazendo requisições 'http' por um tempo específico para testar a resistência de cadastrar produto.

Usuários Virtuais: 25

Duração do Teste: 19 minutos por carga.

Principais métricas: Tempo médio de resposta e taxa de erro.

Os seguintes parâmetros serão utilizados:

- Checks status code 200
- Check response time menor que 3seg
- Average http requisition duration: <1000ms
- Http requisition fail request: < 0.01
- Capacidade de carga: suportar subida gradual de 30 usuários simultâneos e manter iterações desses usuários por 15 minutos.

Resultado:

aplicação do fluxo completo de criação de usuário, login com este usuário, cadastrar produto, cadastrar carrinho e finalizar a compra deletando o carrinho.

Usuários Virtuais: 20

Duração do Teste: 2 minutos por carga.

Principais métricas: Tempo médio de resposta e taxa de erro.

Os seguintes parâmetros serão utilizados:

- Checks status code 200
- Check response time menor que 3seg
- Average http requisition duration: <1000ms
- Http requisition fail request: < 0.01
- Capacidade de carga: suportar subida gradual de 30 usuários simultâneos e manter iterações desses usuários por 15 minutos.

Resultado:

```
PS C:\Users\gabriel\Desktop\k6 serverest performance\projeto-base-k6\tests> k6 run postProdutos.js

      M K6
     .io

execution: local
script: postProdutos.js
output: -

scenarios: (100.00%) 1 scenario, 20 max VUs, 2m30s max duration (incl. graceful stop):
            * default: Up to 20 looping VUs for 2m0s over 2 stages (gracefulRampDown: 30s, gracefulStop: 30s)

✓ POST /produtos status is 201
✗ POST /produtos response time is acceptable
  98% - ✓ 219 / ✗ 4

checks.....: 99.10% ✓ 442 ✗ 4
data_received.....: 380 kB 3.1 kB/s
data_sent.....: 200 kB 1.6 kB/s
http_req_blocked.....: avg=40.12µs min=0s med=0s max=14.51ms p(90)=0s p(95)=0s
http_req_connecting.....: avg=18.32µs min=0s med=0s max=1.13ms p(90)=0s p(95)=0s
✓ http_req_duration.....: avg=922.76ms min=50.05ms med=747.44ms max=3.27s p(90)=1.96s p(95)=2.27s
  { expected_response:true }...: avg=922.76ms min=50.05ms med=747.44ms max=3.27s p(90)=1.96s p(95)=2.27s
✓ http_req_failed.....: 0.00% ✓ 0 ✗ 669
http_req_receiving.....: avg=207.35µs min=0s med=0s max=1.11ms p(90)=843.2µs p(95)=999.4µs
http_req_sending.....: avg=4.87µs min=0s med=0s max=1.03ms p(90)=0s p(95)=0s
http_req_tls_handshaking.....: avg=0s min=0s med=0s max=0s p(90)=0s p(95)=0s
http_req_waiting.....: avg=922.55ms min=49.04ms med=747.44ms max=3.27s p(90)=1.96s p(95)=2.27s
http_reqs.....: 669 5.488238/s
iteration_duration.....: avg=5.79s min=3.39s med=5.76s max=9.08s p(90)=8.32s p(95)=8.65s
iterations.....: 223 1.829413/s
vus.....: 1 min=1 max=20
vus_max.....: 20 min=20 max=20

running (2m01.9s), 00/20 VUs, 223 complete and 0 interrupted iterations
default ✓ [=====] 00/20 VUs 2m0s
```

Iterations: 223

Duration: fast ramp-up to a high point, quick ramp-down to 0 users

http requests duration: average: 922.76ms

http requests failed: 0%

Status code: 200

Response duration < 3sec (x) ✓ | () ✕

Comentários: A duração da requisição http ultrapassou um pouco a métrica de 1 segundo em 2% das requisições totais mas não foi considerado como bug pelo resultado ser um valor muito próximo e aceitável.

Cenário 15: Teste de fumaça da POST rota /produtos

Objetivo: Avaliar a resposta da API com um número mínimo de usuários simultâneos fazendo requisições 'http' por um tempo curto para testar o funcionamento do fluxo completo de criação de usuário, login com este usuário, cadastrar produto, cadastrar carrinho e finalizar a compra deletando o carrinho.

Usuários Virtuais: 5

Duração do Teste: 9 segundos por carga.

Principais métricas: Tempo médio de resposta e taxa de erro.

Os seguintes parâmetros serão utilizados:

- Checks status code 200
- Check response time menor que 3seg
- Average http requisition duration: <800ms
- Http requisition fail request: < 0.01
- Capacidade de carga: suportar subida gradual de 30 usuários simultâneos e manter iterações desses usuários por 15 minutos.

Resultado:

```
PS C:\Users\gabri\Desktop\k6 serverest performance\projeto-base-k6\tests> k6 run postProdutos.js

      /\_/\
     /  _  \
    /_____\ \
   /         \
  /           \
 /             \
/               \
.io

execution: local
  script: postProdutos.js
  output: -
  scenarios: (100.00%) 1 scenario, 5 max VUs, 39s max duration (incl. graceful stop):
    * default: Up to 5 looping VUs for 9s over 3 stages (gracefulRampDown: 30s, gracefulStop: 30s)

✓ POST /produtos status is 201
✓ POST /produtos response time is acceptable

checks.....: 100.00% ✓ 18      ✗ 0
data_received.....: 15 kB   1.4 kB/s
data_sent.....: 7.9 kB   707 B/s
http_req_blocked.....: avg=533.33µs min=0s    med=0s    max=12.22ms p(90)=526.5µs p(95)=946.18µs
http_req_connecting.....: avg=117.49µs min=0s    med=0s    max=1.06ms  p(90)=526.5µs p(95)=898.44µs
✓ http_req_duration.....: avg=112.86ms min=28.49ms med=96.66ms max=279.25ms p(90)=227.23ms p(95)=236.36ms
  { expected_response:true }...: avg=112.86ms min=28.49ms med=96.66ms max=279.25ms p(90)=227.23ms p(95)=236.36ms
✓ http_req_failed.....: 0.00% ✓ 0      ✗ 27
http_req_receiving.....: avg=256.84µs min=0s    med=0s    max=1.25ms  p(90)=936.74µs p(95)=1.04ms
http_req_sending.....: avg=21.31µs min=0s    med=0s    max=507.4µs p(90)=0s      p(95)=47.73µs
http_req_tls_handshaking.....: avg=0s      min=0s    med=0s    max=0s      p(90)=0s      p(95)=0s
http_req_waiting.....: avg=112.58ms min=28.49ms med=96.66ms max=279.25ms p(90)=227.23ms p(95)=236.36ms
http_reqs.....: 27      2.405374/s
iteration_duration.....: avg=3.36s   min=3.24s med=3.39s max=3.46s   p(90)=3.43s   p(95)=3.44s
iterations.....: 9      0.801791/s
vus.....: 1      min=1    max=5
vus_max.....: 5      min=5    max=5

running (11.2s), 0/5 VUs, 9 complete and 0 interrupted iterations
default ✓ [=====] 0/5 VUs  9s
```

Iterations: 9

Duration: Traffic ramp-up from 1. Key for Smoke test. Keep max 5 VUs. Duration can be shorter or just a few iterations.

http requests duration: average: 112.86ms

http requests failed: 0%

Status code: 200

Response duration < 3sec (x) ✓ | () ✗

Cenário 16: Teste de Carga de POST da rota /delete-carrinhos-concluir compra

Objetivo: Avaliar a resposta da API com um número crescente de usuários simultâneos fazendo requisições 'http' para a aplicação do verbo para deletando o carrinho.

Usuários Virtuais: 20

Duração do Teste: 3 minutos por carga.

Principais métricas: Tempo médio de resposta e taxa de erro.

Os seguintes parâmetros serão utilizados:

- Checks status code 200
- Check response time menor que 3seg
- Average http requisition duration: <1000ms
- Http requisition fail request: < 0.01
- Capacidade de carga: suportar subida gradual até 20 usuários simultâneos e manter iterações desses usuários por 3 minutos.
- Escalabilidade: Planejar um aumento de 20% na capacidade para suportar crescimento futuro.

Resultados obtidos:

```
PS C:\Users\gabriel\Desktop\k6 serverest performance\projeto-base\k6\tests> k6 run deleteCarrinho.js
```

```

      /\_/\
     /  _  \
    /_____\ \   .io

```

```

execution: local
script: deleteCarrinho.js
output: -

scenarios: (100.00%) 1 scenario, 20 max VUs, 5m30s max duration (incl. graceful stop):
  * default: Up to 20 looping VUs for 5m0s over 3 stages (gracefulRampDown: 30s, gracefulStop: 30s)

✓ DELETE /carrinhos/concluir-compra status is 200
✓ DELETE /carrinhos/concluir-compra response time is acceptable

checks.....: 100.00% ✓ 1148 X 0
data_received.....: 1.5 MB 5.1 kB/s
data_sent.....: 948 kB 3.1 kB/s
http_req_blocked.....: avg=8.81µs min=0s med=0s max=13.89ms p(90)=0s p(95)=0s
http_req_connecting.....: avg=4.1µs min=0s med=0s max=1.09ms p(90)=0s p(95)=0s
✓ http_req_duration.....: avg=898.09ms min=20.62ms med=947.08ms max=2.62s p(90)=1.5s p(95)=1.68s
  { expected_response:true }...: avg=898.09ms min=20.62ms med=947.08ms max=2.62s p(90)=1.5s p(95)=1.68s
✓ http_req_failed.....: 0.00% ✓ 0 X 2870
http_req_receiving.....: avg=198.46µs min=0s med=0s max=2.46ms p(90)=875.39µs p(95)=999.7µs
http_req_sending.....: avg=2.07µs min=0s med=0s max=897µs p(90)=0s p(95)=0s
http_req_tls_handshaking.....: avg=0s min=0s med=0s max=0s p(90)=0s p(95)=0s
http_req_waiting.....: avg=897.89ms min=20.49ms med=947.08ms max=2.62s p(90)=1.5s p(95)=1.68s
http_reqs.....: 2870 9.425678/s
iteration_duration.....: avg=8.52s min=4.44s med=9.07s max=10.81s p(90)=9.99s p(95)=10.2s
iterations.....: 574 1.885136/s
vus.....: 1 min=1 max=20
vus_max.....: 20 min=20 max=20

running (5m04.5s), 00/20 VUs, 574 complete and 0 interrupted iterations
default ✓ [=====] 00/20 VUs 5m0s
```

Iterations: 574

Duration: Traffic ramp-up from 1 to 20 users over 1 minutes. Stay at 20 users for 3 minutes. ramp-down to 0 users

http requests duration: average: 898.09ms

http requests failed: 0%

Status code: 200

Response duration < 3sec (x) ✓ | () ✗

Cenário 17: Teste de estresse de POST rota /delete-carrinhos-concluir

Objetivo: Avaliar a resposta da API com um número crescente de usuários simultâneos fazendo requisições 'http' para estressar a aplicação do verbo para deletando o carrinho.

Usuários Virtuais: 40

Duração do Teste: 15 minutos por carga.

Principais métricas: Tempo médio de resposta e taxa de erro.

Os seguintes parâmetros serão utilizados:

- Checks status code 200
- Check response time menor que 3seg
- Average http requisition duration: <1000ms
- Http requisition fail request: < 0.01
- Capacidade de carga: suportar subida para 40 usuários simultâneos e manter iterações desses usuários por 10 minutos.

Resultados obtidos:

```
PS C:\Users\gabir\Desktop\k6 serverest performance\projeto-base-k6\tests> k6 run deleteCarrinho.js

      /\_/\
     /__  \
    (oo)\_____)
    (__/       )\/\
    ||----w |
    ||     ||

.io

execution: local
script: deleteCarrinho.js
output: -

scenarios: (100.00%) 1 scenario, 40 max VUs, 15m30s max duration (incl. graceful stop):
  * default: Up to 40 looping VUs for 15m0s over 3 stages (gracefulRampDown: 30s, gracefulStop: 30s)

WARN[0198] Request Failed                                error="Delete \"http://localhost:3000/carrinhos/concluir-compra\": EOF"

X DELETE /carrinhos/concluir-compra status is 200
  ↳ 99% - ✓ 1628 / X 1
X DELETE /carrinhos/concluir-compra response time is acceptable
  ↳ 22% - ✓ 360 / X 1269

checks.....: 61.01% ✓ 1988      X 1270
data_received.....: 4.4 MB 4.8 kB/s
data_sent.....: 2.7 MB 3.0 kB/s
http_req_blocked.....: avg=10.33µs min=0s med=0s max=15.37ms p(90)=0s p(95)=0s
http_req_connecting.....: avg=4.8µs min=0s med=0s max=3.16ms p(90)=0s p(95)=0s
X http_req_duration.....: avg=2.91s min=27.54ms med=3.4s max=5.11s p(90)=4.12s p(95)=4.36s
  { expected_response:true }...: avg=2.9s min=27.54ms med=3.39s max=5.02s p(90)=4.11s p(95)=4.33s
✓ http_req_failed.....: 0.51% ✓ 42      X 8103
http_req_receiving.....: avg=199.77µs min=0s med=0s max=1.57ms p(90)=899.1µs p(95)=999.6µs
http_req_sending.....: avg=2.96µs min=0s med=0s max=1.81ms p(90)=0s p(95)=0s
http_req_tls_handshaking.....: avg=0s min=0s med=0s max=0s p(90)=0s p(95)=0s
http_req_waiting.....: avg=2.91s min=26.49ms med=3.4s max=5.11s p(90)=4.12s p(95)=4.36s
http_reqs.....: 8145 9.030854/s
iteration_duration.....: avg=18.6s min=4.53s med=20.46s max=22.94s p(90)=21.51s p(95)=21.77s
iterations.....: 1629 1.806171/s
vus.....: 1 min=1 max=40
vus_max.....: 40 min=40 max=40

running (15m01.9s), 00/40 VUs, 1629 complete and 0 interrupted iterations
default ✓ [=====] 00/40 VUs 15m0s
ERR! [0003] thresholds on metrics 'http_req_duration' have been crossed
```

Iterations: 1629

Duration: traffic ramp-up from 1 to a higher 40 users over 3 minutes. stay at higher 40 users for 10 minutes. 2 min for ramp-down to 0 users

http requests duration: average: 2.32seg

http requests failed: 0%

Status code: 200

Response duration < 3sec () ✓ | (x) ✗

Comentários: Uma requisição não foi status code 200 corretamente e resultou em 408 request timeout e 78% das requisições foram acima do nível aceitável de tempo de resposta http, mas já que a média ficou em 2.32seg não será classificado como bug considerando os problemas de performance.

Cenário 18: Teste de resistência de DELETE rota /delete-carrinhos-concluir

Objetivo: Avaliar a resposta da API com um número crescente de usuários simultâneos fazendo requisições 'http' por um tempo específico para testar a resistência de concluir a compra.

Usuários Virtuais: 25

Duração do Teste: 19 minutos por carga.

Principais métricas: Tempo médio de resposta e taxa de erro.

Os seguintes parâmetros serão utilizados:

- Checks status code 200
- Check response time menor que 3seg
- Average http requisition duration: <1000ms
- Http requisition fail request: < 0.01
- Capacidade de carga: suportar subida gradual de 30 usuários simultâneos e manter iterações desses usuários por 15 minutos.

Resultado:

```
PS C:\Users\gabri\Desktop\k6 serverest performance\projeto-base-k6\tests> k6 run deleteCarrinho.js

      M K6 .io

execution: local
script: deleteCarrinho.js
output: -

scenarios: (100.00%) 1 scenario, 25 max VUs, 19m30s max duration (incl. graceful stop):
* default: Up to 25 looping VUs for 19m0s over 3 stages (gracefulRampDown: 30s, gracefulStop: 30s)

✓ DELETE /carrinhos/concluir-comprou status is 200
X DELETE /carrinhos/concluir-comprou response time is acceptable
  ↳ 73% - ✓ 1170 / X 423

checks.....: 86.72% ✓ 2763      X 423
data_received.....: 4.3 MB 3.7 kB/s
data_sent.....: 2.6 MB 2.3 kB/s
http_req_blocked.....: avg=6.9µs min=0s med=0s max=13.95ms p(90)=0s p(95)=0s
http_req_connecting.....: avg=3µs min=0s med=0s max=9.22ms p(90)=0s p(95)=0s
X http_req_duration.....: avg=2.42s min=34.49ms med=2.67s max=5.12s p(90)=3.53s p(95)=3.83s
  { expected_response:true }...: avg=2.41s min=34.49ms med=2.67s max=5.04s p(90)=3.52s p(95)=3.81s
✓ http_req_failed.....: 0.06% ✓ 5      X 7960
http_req_receiving.....: avg=191.14µs min=0s med=0s max=1.15ms p(90)=864µs p(95)=999.5µs
http_req_sending.....: avg=2.55µs min=0s med=0s max=1ms p(90)=0s p(95)=0s
http_req_tls_handshaking.....: avg=0s min=0s med=0s max=0s p(90)=0s p(95)=0s
http_req_waiting.....: avg=2.42s min=34.49ms med=2.67s max=5.12s p(90)=3.53s p(95)=3.83s
http_reqs.....: 7965 6.974875/s
iteration_duration.....: avg=16.13s min=4.65s med=17.23s max=19.62s p(90)=18.26s p(95)=18.6s
iterations.....: 1593 1.394975/s
vus.....: 1 min=1 max=25
vus_max.....: 25 min=25 max=25

running (19m02.0s), 00/25 VUs, 1593 complete and 0 interrupted iterations
default ✓ [=====] 00/25 VUs 19m0s
ERRO[1143] thresholds on metrics 'http_req_duration' have been crossed
```

Iterations: 1593

Duration: traffic ramp-up from 1 to 30 users over 5 minutes. stay at 30 users for 15 minutes.
ramp-down to 0 users

http requests duration: average: 2.41s

http requests failed: 0%

Status code: 200

Response duration < 3sec () ✓ | (x) ✗

Comentários: A duração da requisição http ultrapassou a métrica de response time aceitável onde 70% das requisições deram falhas mas com a média em 2.41seg e p(90) em 3.52seg não será classificado como bug e considerado aceitável.

Cenário 19: Teste de pico da DELETE rota /delete-carrinhos-concluir

Objetivo: Avaliar a resposta da API com um número crescente e espontâneo de usuários simultâneos fazendo requisições 'http' em um período rápido para a aplicação da finalização a compra deletando o carrinho.

Status code: 200

Response duration < 3sec (x) ✓ | () ✗

Comentários: A threshold estabelecida em tempo de resposta da requisição menor que 1seg mas pela proximidade do resultado de 1.18s, o resultado será considerado como positivo.

Cenário 20: Teste de fumaça da DELETE rota /delete-carrinhos-concluir

Objetivo: Avaliar a resposta da API com um número mínimo de usuários simultâneos fazendo requisições 'http' por um tempo curto para testar o funcionamento da finalização a compra deletando o carrinho.

Usuários Virtuais: 5

Duração do Teste: 9 segundos por carga.

Principais métricas: Tempo médio de resposta e taxa de erro.

Os seguintes parâmetros serão utilizados:

- Checks status code 200
- Check response time menor que 3seg
- Average http requisition duration: <800ms
- Http requisition fail request: < 0.01
- Capacidade de carga: suportar subida gradual de 30 usuários simultâneos e manter iterações desses usuários por 15 minutos.

Resultado:

```
PS C:\Users\gabri\Desktop\k6 serverest performance\projeto-base-k6\tests> k6 run deleteCarrinho.js

      M K6 .io

execution: local
script: deleteCarrinho.js
output: -

scenarios: (100.00%) 1 scenario, 5 max VUs, 39s max duration (incl. graceful stop):
    * default: Up to 5 looping VUs for 9s over 3 stages (gracefulRampDown: 30s, gracefulStop: 30s)

✓ DELETE /carrinhos/concluir-compra status is 200
✓ DELETE /carrinhos/concluir-compra response time is acceptable

checks.....: 100.00% ✓ 16      ✗ 0
data_received.....: 21 kB    1.6 kB/s
data_sent.....: 13 kB    995 B/s
http_req_blocked.....: avg=524.64µs min=0s      med=0s      max=18.24ms p(90)=522.6µs p(95)=578.61µs
http_req_connecting.....: avg=67.79µs min=0s      med=0s      max=571.1µs p(90)=522.54µs p(95)=535.91µs
✓ http_req_duration.....: avg=118.52ms min=17.61ms med=80.94ms max=315.86ms p(90)=246.15ms p(95)=249.07ms
  { expected_response:true }...: avg=118.52ms min=17.61ms med=80.94ms max=315.86ms p(90)=246.15ms p(95)=249.07ms
✓ http_req_failed.....: 0.00% ✓ 0      ✗ 40
http_req_receiving.....: avg=272.42µs min=0s      med=0s      max=1.04ms p(90)=915.53µs p(95)=999.78µs
http_req_sending.....: avg=1.46µs min=0s      med=0s      max=58.7µs p(90)=0s p(95)=0s
http_req_tls_handshaking.....: avg=0s min=0s      med=0s      max=0s p(90)=0s p(95)=0s
http_req_waiting.....: avg=118.25ms min=17.49ms med=80.49ms max=315.86ms p(90)=246.12ms p(95)=248.9ms
http_reqs.....: 40    3.039498/s
iteration_duration.....: avg=4.62s min=4.25s med=4.61s max=4.93s p(90)=4.91s p(95)=4.92s
iterations.....: 8    0.6079/s
vus.....: 1    min=1    max=5
vus_max.....: 5    min=5    max=5

running (13.2s), 0/5 VUs, 8 complete and 0 interrupted iterations
default ✓ [=====] 0/5 VUs 9s
PS C:\Users\gabri\Desktop\k6 serverest performance\projeto-base-k6\tests> |
```

Iterations: 8

Duration: Traffic ramp-up from 1. Key for Smoke test. Keep max 5 VUs. Duration can be shorter or just a few iterations.

http requests duration: average: 118.52ms

http requests failed: 0%

Status code: 200

Response duration < 3sec (x) ✓ | () ✗

Cenário 21: Teste de Carga de POST da rota /login

Objetivo: Avaliar a resposta da API com um número crescente de usuários simultâneos.

Usuários Virtuais: 20

Duração do Teste: 3 minutos por carga.


Principais métricas: Tempo médio de resposta e taxa de erro.

Os seguintes parâmetros serão utilizados:

- Checks status code 200
- Check response time menor que 3seg
- Average http requisition duration: <1000ms
- Http requisition fail request: < 0.01
- Capacidade de carga: suportar subida gradual até 20 usuários simultâneos e manter iterações desses usuários por 3 minutos.

Resultado obtido:

```
PS C:\Users\gabri\Desktop\k6 serverest performance\projeto-base-k6\tests> k6 run postLogin.js
```



```

execution: local
  script: postLogin.js
  output: -

scenarios: (100.00%) 1 scenario, 20 max VUs, 5m30s max duration (incl. graceful stop):
    * default: Up to 20 looping VUs for 5m0s over 3 stages (gracefulRampDown: 30s, gracefulStop: 30s)

✓ POST /login status was 200
✓ POST /login response time is acceptable

checks.....: 100.00% ✓ 5508      ✗ 0
data_received.....: 3.3 MB   11 kB/s
data_sent.....: 1.2 MB   3.9 kB/s
http_req_blocked.....: avg=5µs      min=0s      med=0s      max=15.18ms  p(90)=0s      p(95)=0s
http_req_connecting.....: avg=1.97µs   min=0s      med=0s      max=1.07ms   p(90)=0s      p(95)=0s
✓ http_req_duration.....: avg=375.52ms min=16.49ms med=345.57ms max=1.01s    p(90)=659.27ms p(95)=711.15ms
    { expected_response:true }...: avg=375.52ms min=16.49ms med=345.57ms max=1.01s    p(90)=659.27ms p(95)=711.15ms
✓ http_req_failed.....: 0.00% ✓ 0      ✗ 5508
http_req_receiving.....: avg=188.28µs min=0s      med=0s      max=2.07ms   p(90)=846.64µs p(95)=999.4µs
http_req_sending.....: avg=1.88µs   min=0s      med=0s      max=1.32ms   p(90)=0s      p(95)=0s
http_req_tls_handshaking.....: avg=0s       min=0s      med=0s      max=0s       p(90)=0s      p(95)=0s
http_req_waiting.....: avg=375.33ms min=16.49ms med=345.26ms max=1.01s    p(90)=659.27ms p(95)=711.15ms
http_reqs.....: 5508      18.312522/s
iteration_duration.....: avg=1.75s    min=1.07s   med=1.85s    max=2.55s    p(90)=2.04s    p(95)=2.12s
iterations.....: 2754      9.156261/s
vus.....: 1      min=1      max=20
vus_max.....: 20      min=20     max=20

running (5m00.8s), 00/20 VUs, 2754 complete and 0 interrupted iterations
default ✓ [=====] 00/20 VUs  5m0s
```

Iterations: 2754

Duration: Traffic ramp-up from 1 to 20 users over 1 minutes. Stay at 20 users for 3 minutes. ramp-down to 0 users

http requests duration: average: 375.52ms

http requests failed: 0%

Status code: 200

Response duration < 3sec (x) ✓ | () ✗

Cenário 22: Teste de estresse de POST rota /login

Objetivo: Avaliar a resposta da API com um número crescente de usuários simultâneos fazendo requisições 'http' para estressar a realização de login.

Usuários Virtuais: 40

Duração do Teste: 15 minutos por carga.

Principais métricas: Tempo médio de resposta e taxa de erro.

Os seguintes parâmetros serão utilizados:

- Checks status code 200
- Check response time menor que 3seg
- Average http requisition duration: <1000ms
- Http requisition fail request: < 0.01
- Capacidade de carga: suportar subida para 40 usuários simultâneos e manter iterações desses usuários por 10 minutos.

Resultados obtidos:

```

execution: local
script: postLogin.js
output: -

scenarios: (100.00%) 1 scenario, 40 max VUs, 15m30s max duration (incl. graceful stop):
    * default: Up to 40 looping VUs for 15m0s over 3 stages (gracefulRampDown: 30s, gracefulStop: 30s)

WARN[0229] Request Failed                      error="Post \"http://localhost:3000/login\": EOF"
ERRO[0229] GoError: the body is null so we can't transform it to JSON - this likely was because of a request error getting the response

(61)) executor=ramping-vus scenario=default source=stacktrace
WARN[0229] Request Failed                      error="Post \"http://localhost:3000/login\": EOF"
ERRO[0229] GoError: the body is null so we can't transform it to JSON - this likely was because of a request error getting the response

(61)) executor=ramping-vus scenario=default source=stacktrace
WARN[0229] Request Failed                      error="Post \"http://localhost:3000/login\": EOF"
ERRO[0229] GoError: the body is null so we can't transform it to JSON - this likely was because of a request error getting the response

(61)) executor=ramping-vus scenario=default source=stacktrace
WARN[0229] Request Failed                      error="Post \"http://localhost:3000/login\": EOF"
ERRO[0229] GoError: the body is null so we can't transform it to JSON - this likely was because of a request error getting the response

(61)) executor=ramping-vus scenario=default source=stacktrace
WARN[0229] Request Failed                      error="Post \"http://localhost:3000/login\": EOF"
ERRO[0229] GoError: the body is null so we can't transform it to JSON - this likely was because of a request error getting the response

(61)) executor=ramping-vus scenario=default source=stacktrace
WARN[0229] Request Failed                      error="Post \"http://localhost:3000/login\": EOF"
ERRO[0229] GoError: the body is null so we can't transform it to JSON - this likely was because of a request error getting the response

(61)) executor=ramping-vus scenario=default source=stacktrace
WARN[0236] Request Failed                      error="Post \"http://localhost:3000/login\": EOF"
ERRO[0236] GoError: the body is null so we can't transform it to JSON - this likely was because of a request error getting the response

(61)) executor=ramping-vus scenario=default source=stacktrace

running (13m21.7s), 34/40 VUs, 5209 complete and 0 interrupted iterations
default [=====>] 34/40 VUs 13m21.7s/15m00.0s

X POST /login status is 200
  99% - ✓ 5802 / X 7
✓ POST /login response time is acceptable

checks.....: 99.93% / 11611      X 7
data_received.....: 7.0 MB 7.7 kB/s
data_sent.....: 2.5 MB 2.8 kB/s
http_req_blocked.....: avg=4.05µs min=0s med=0s max=12.88ms p(90)=0s p(95)=0s
http_req_connecting.....: avg=2.33µs min=0s med=0s max=2ms p(90)=0s p(95)=0s
X http_req_duration.....: avg=2.08s min=6.4ms med=1.57s max=5.08s p(90)=3.92s p(95)=4.12s
  { expected_response:true }...: avg=2.08s min=23.49ms med=1.57s max=5.03s p(90)=3.92s p(95)=4.12s
✓ http_req_failed.....: 0.12% / 14      X 11604
http_req_receiving.....: avg=200.74µs min=0s med=0s max=1.47ms p(90)=902.28µs p(95)=999.5µs
http_req_sending.....: avg=2.61µs min=0s med=0s max=1ms p(90)=0s p(95)=0s
http_req_tls_handshaking.....: avg=0s min=0s med=0s max=0s p(90)=0s p(95)=0s
http_req_waiting.....: avg=2.08s min=6.4ms med=1.57s max=5.08s p(90)=3.92s p(95)=4.12s
http_reqs.....: 11618 12.905366/s
iteration_duration.....: avg=5.18s min=1.11s med=5.93s max=7.66s p(90)=6.34s p(95)=6.45s
iterations.....: 5809 6.452683/s
vus.....: 1 min=1 max=40
vus_max.....: 40 min=40 max=40

running (15m00.2s), 00/40 VUs, 5809 complete and 0 interrupted iterations
default ✓ [=====] 00/40 VUs 15m0s
ERRO[0901] thresholds on metrics 'http_req_duration' have been crossed

```

Iterations: 5809

Duration: traffic ramp-up from 1 to a higher 40 users over 3 minutes. stay at higher 40 users for 10 minutes. 2 min for ramp-down to 0 users

http requests duration: average: 2.08seg

http requests failed: 0%

Status code: 200

Response duration < 3sec () ✓ | (x) ✗

Comentários: 99% das requisições foram status code 200 por request fail.

Cenário 23: Teste de resistência de POST rota /login

Objetivo: Avaliar a resposta da API com um número crescente de usuários simultâneos fazendo requisições 'http' por um tempo específico para testar a resistência de realizar login na API.

Usuários Virtuais: 25

Duração do Teste: 19 minutos por carga.

Principais métricas: Tempo médio de resposta e taxa de erro.

Os seguintes parâmetros serão utilizados:

- Checks status code 200
- Check response time menor que 3seg
- Average http requisition duration: <1000ms
- Http requisition fail request: < 0.01
- Capacidade de carga: suportar subida gradual de 30 usuários simultâneos e manter iterações desses usuários por 15 minutos.

Resultado:

```

PS C:\Users\gabri\Desktop\k6 serverest performance\projeto-base-k6\tests> k6 run postLogin.js

      M K6 .io

execution: local
script: postLogin.js
output: -

scenarios: (100.00%) 1 scenario, 25 max VUs, 19m30s max duration (incl. graceful stop):
  * default: Up to 25 looping VUs for 19m0s over 3 stages (gracefulRampDown: 30s, gracefulStop: 30s)

✓ POST /login status is 200
✓ POST /login response time is acceptable

checks.....: 100.00% ✓ 12730      X 0
data_received.....: 7.6 MB 6.7 kB/s
data_sent.....: 2.7 MB 2.4 kB/s
http_req_blocked.....: avg=2.2µs min=0s med=0s max=14.18ms p(90)=0s p(95)=0s
http_req_connecting.....: avg=984ns min=0s med=0s max=685.6µs p(90)=0s p(95)=0s
X http_req_duration.....: avg=1.5s min=33.48ms med=1.32s max=2.88s p(90)=2.49s p(95)=2.56s
  { expected_response:true }...: avg=1.5s min=33.48ms med=1.32s max=2.88s p(90)=2.49s p(95)=2.56s
✓ http_req_failed.....: 0.00% ✓ 0      X 12730
http_req_receiving.....: avg=190.5µs min=0s med=0s max=2.29ms p(90)=897.5µs p(95)=999.8µs
http_req_sending.....: avg=1.06µs min=0s med=0s max=999.1µs p(90)=0s p(95)=0s
http_req_tls_handshaking.....: avg=0s min=0s med=0s max=0s p(90)=0s p(95)=0s
http_req_waiting.....: avg=1.5s min=33.48ms med=1.32s max=2.88s p(90)=2.49s p(95)=2.56s
http_reqs.....: 12730 11.158208/s
iteration_duration.....: avg=4.02s min=1.14s med=4.29s max=5.01s p(90)=4.51s p(95)=4.58s
iterations.....: 6365 5.579104/s
vus.....: 1 min=1 max=25
vus_max.....: 25 min=25 max=25

running (19m00.9s), 00/25 VUs, 6365 complete and 0 interrupted iterations
default ✓ [=====] 00/25 VUs 19m0s
ERRO[1142] thresholds on metrics 'http_req_duration' have been crossed
PS C:\Users\gabri\Desktop\k6 serverest performance\projeto-base-k6\tests> |

```

Iterations: 6365

Duration: traffic ramp-up from 1 to 30 users over 5 minutes. stay at 30 users for 15 minutes.
ramp-down to 0 users

http requests duration: average: 1.5s

http requests failed: 0%

Status code: 200

Response duration < 3sec () ✓ | (x) ✗

Comentários: A duração da requisição http ultrapassou a métrica de response time aceitável onde metade das requisições deram falhas, mas a média permaneceu em 1.5seg e o percentil(90) em 2.49seg considerado aceitável para o e-commerce.

Cenário 24: Teste de pico da POST rota /login

Objetivo: Avaliar a resposta da API com um número crescente e espontâneo de usuários simultâneos fazendo requisições 'http' em um período rápido para a aplicação do fluxo completo de criação de usuário, login com este usuário.

Usuários Virtuais: 20

Duração do Teste: 2 minutos por carga.

Principais métricas: Tempo médio de resposta e taxa de erro.

Os seguintes parâmetros serão utilizados:

- Checks status code 200
- Check response time menor que 3seg
- Average http requisition duration: <1000ms
- Http requisition fail request: < 0.01
- Capacidade de carga: suportar subida gradual de 30 usuários simultâneos e manter iterações desses usuários por 15 minutos.

Resultado:

```
PS C:\Users\gabri\Desktop\k6 serverest performance\projeto-base-k6\tests> k6 run postLogin.js

      M K6
     .io

execution: local
script: postLogin.js
output: -

scenarios: (100.00%) 1 scenario, 20 max VUs, 2m30s max duration (incl. graceful stop):
 * default: Up to 20 looping VUs for 2m0s over 2 stages (gracefulRampDown: 30s, gracefulStop: 30s)

✓ POST /login status is 200
✓ POST /login response time is acceptable

checks.....: 100.00% ✓ 1032 X 0
data_received.....: 620 kB 5.2 kB/s
data_sent.....: 222 kB 1.8 kB/s
http_req_blocked.....: avg=25.1µs min=0s med=0s max=14.52ms p(90)=0s p(95)=0s
http_req_connecting.....: avg=10.57µs min=0s med=0s max=1.13ms p(90)=0s p(95)=0s
✓ http_req_duration.....: avg=702.89ms min=37.25ms med=577.86ms max=2.16s p(90)=1.57s p(95)=1.82s
  { expected_response:true }...: avg=702.89ms min=37.25ms med=577.86ms max=2.16s p(90)=1.57s p(95)=1.82s
✓ http_req_failed.....: 0.00% ✓ 0 X 1032
http_req_receiving.....: avg=203.24µs min=0s med=0s max=1.82ms p(90)=903.02µs p(95)=1ms
http_req_sending.....: avg=2.3µs min=0s med=0s max=744.3µs p(90)=0s p(95)=0s
http_req_tls_handshaking.....: avg=0s min=0s med=0s max=0s p(90)=0s p(95)=0s
http_req_waiting.....: avg=702.68ms min=37.07ms med=577.69ms max=2.16s p(90)=1.57s p(95)=1.82s
http_reqs.....: 1032 8.577955/s
iteration_duration.....: avg=2.41s min=1.16s med=2.38s max=4.12s p(90)=3.7s p(95)=3.84s
iterations.....: 516 4.288978/s
vus.....: 1 min=1 max=20
vus_max.....: 20 min=20 max=20

running (2m00.3s), 00/20 VUs, 516 complete and 0 interrupted iterations
default ✓ [=====] 00/20 VUs 2m0s
```

Iterations: 516

Duration: fast ramp-up to a high point, quick ramp-down to 0 users

http requests duration: average: 702.89ms

http requests failed: 0%

Status code: 200

Response duration < 3sec (x) ✓ | () ✕

Cenário 25: Teste de fumaça da POST rota /login

Objetivo: Avaliar a resposta da API com um número mínimo de usuários simultâneos fazendo requisições 'http' por um tempo curto para testar o funcionamento do fluxo completo de criação de usuário, login com este usuário.

Usuários Virtuais: 5

Duração do Teste: 9 segundos por carga.

Principais métricas: Tempo médio de resposta e taxa de erro.

Os seguintes parâmetros serão utilizados:

- Checks status code 200
- Check response time menor que 3seg
- Average http requisition duration: <800ms
- Http requisition fail request: < 0.01
- Capacidade de carga: suportar subida gradual de 30 usuários simultâneos e manter iterações desses usuários por 15 minutos.

Resultado:

```

PS C:\Users\gabri\Desktop> k6 serverless performance\projeto-base-k6\tests> k6 run postLogin.js

      M K6
     io

execution: local
  script: postLogin.js
  output: -
scenarios: (100.00%) 1 scenario, 5 max VUs, 39s max duration (incl. graceful stop):
    * default: Up to 5 looping VUs for 9s over 3 stages (gracefulRampDown: 30s, gracefulStop: 30s)

✓ POST /login status was 200
✓ POST /login response time is acceptable

checks.....: 100.00% ✓ 40      X 0
data_received.....: 24 kB    2.6 kB/s
data_sent.....: 8.6 kB    941 B/s
http_req_blocked.....: avg=499.46µs min=0s      med=0s      max=17.15ms p(90)=520.35µs p(95)=573.3µs
http_req_connecting.....: avg=83.4µs min=0s      med=0s      max=1.24ms p(90)=515.24µs p(95)=523.38µs
✓ http_req_duration.....: avg=62.44ms min=17.22ms med=65.04ms max=120.49ms p(90)=97.85ms p(95)=107.23ms
  { expected_response:true }...: avg=62.44ms min=17.22ms med=65.04ms max=120.49ms p(90)=97.85ms p(95)=107.23ms
✓ http_req_failed.....: 0.00% ✓ 0      X 40
http_req_receiving.....: avg=283.56µs min=0s      med=0s      max=1.08ms p(90)=989.44µs p(95)=1ms
http_req_sending.....: avg=12.63µs min=0s      med=0s      max=505.4µs p(90)=0s p(95)=0s
http_req_tls_handshaking.....: avg=0s min=0s      med=0s      max=0s p(90)=0s p(95)=0s
http_req_waiting.....: avg=62.15ms min=16.5ms med=64.55ms max=119.88ms p(90)=97.83ms p(95)=107.23ms
http_reqs.....: 40    4.37781/s
iteration_duration.....: avg=1.13s min=1.08s med=1.12s max=1.21s p(90)=1.16s p(95)=1.21s
iterations.....: 20    2.188905/s
vus.....: 2    min=1    max=5
vus_max.....: 5    min=5    max=5

running (09.1s), 0/5 VUs, 20 complete and 0 interrupted iterations
default ✓ [=====] 0/5 VUs  9s

```

Iterations: 20

Duration: Traffic ramp-up from 1. Key for Smoke test. Keep max 5 VUs. Duration can be shorter or just a few iterations.

http requests duration: average: 62.44ms

http requests failed: 0%

Status code: 200

Response duration < 3sec (x) ✓ | () ✗