

Documentação - Teste de Carga/Estresse

FarmacoCheck

Registro:

Início	Fim
12/01/2025 - 13:00h	12/01/2025 - 15:03h

Descrição: Este documento descreve o teste de estresse realizado na aplicação FarmacoCheck em ambiente de produção, utilizando a ferramenta K6. O teste teve como objetivo avaliar o desempenho da aplicação ao acessar o endpoint da tela inicial da aplicação, ponto de partida para acesso a todas as funcionalidades incluindo registrar-se ou efetuar o login:

O objetivo deste teste é simular um número crescente de usuários simultâneos acessando o serviço, monitorar o comportamento sob alta carga e verificar se o sistema suporta o aumento de tráfego sem degradação severa de performance ou falhas.

Requisito Associado:

RNF003			
Nome:	Desempenho e Escalabilidade		
Descrição:	<p>O sistema deve ser capaz de suportar pelo menos 100 usuários simultâneos e responder a requisições em até 3 segundos, para que o usuário consiga usar o sistema mesmo em condições não previstas.</p> <p>O tempo de resposta de até 3 segundos e o quantitativo de 100 usuários simultâneos foram estimados com base na análise inicial do público-alvo e na expectativa de uso durante picos de acesso, considerando a infraestrutura disponível e o planejamento de escalabilidade futura.</p> <p>obs: As métricas foram definidas após uma análise breve do público-alvo e com base nos interesses do cliente.</p>		
Prioridade:	1	Anexo:	----
Entradas e pré-condições:	<ol style="list-style-type: none">1. Carga estimada de usuários durante picos de acesso.2. Estrutura de infraestrutura disponível (servidores, banco de dados).		
Saídas e pós-condições:	<ol style="list-style-type: none">1. O sistema é capaz de atender às requisições de forma eficiente, mesmo em situações de alta demanda.		
Critérios de aceitação:			

1. O tempo de resposta médio para requisições deve ser inferior a 3 segundos em 95% das interações.
2. O sistema deve suportar pelo menos 100 usuários simultâneos sem perda de desempenho.
3. Em situações de carga máxima, o sistema deve ser capaz de escalar automaticamente para acomodar a demanda.
4. Testes de carga devem ser realizados periodicamente para garantir que o sistema possa lidar com picos de usuários sem comprometer o desempenho.

Caso de teste 01

Código do Teste:

```
import http from 'k6/http';
import { sleep } from 'k6';

export const options = {
  //config
  stages: [
    { duration: '2m', target: 100 }, // aumento gradual para 100 usuários em 2 min
    { duration: '5m', target: 100 }, // manutenção de 100 usuários por 5 min
    { duration: '2m', target: 50 }, // redução gradual para 50 usuários
    { duration: '2m', target: 0 }, // redução gradual para 0 usuários
  ],
  thresholds: {
    // Testa se 95% das requisições estão abaixo de 3 segundos
    http_req_duration: ['p(95)<3000']
  }
};

export default () => {
  const urlRes = http.get('endpoint de um catálogo aleatório');
  sleep(1);
};
```

Descrição:

Este caso de teste foi elaborado para realizar um teste de carga na aplicação, com o objetivo de validar o comportamento do sistema ao receber um número crescente de usuários

simultâneos, atingindo um pico de 100 usuários, seguido por uma manutenção e uma redução gradual até 0 usuários. O teste visa verificar se 95% das requisições realizadas ao endpoint da página inicial do FarmacoCheck retornam com um tempo de resposta inferior a 3 segundos.

- **Objetivo do Teste:** Validar o desempenho do endpoint de um catálogo aleatório sob carga, com ênfase no tempo de resposta das requisições.
- **Carga Inicial:** O número de usuários virtuais (VUs) começa em 0 e vai até 100 de forma gradual ao longo de 2 minutos.
- **Manutenção de Carga:** A carga de 100 VUs é mantida por 5 minutos.
- **Redução Gradual de Usuários:** A carga é reduzida para 50 VUs e, em seguida, para 0 VUs ao longo de 4 minutos.
- **Critério de Sucesso:** 95% das requisições devem ter um tempo de resposta inferior a 3 segundos.

Métrica de desempenho:

- O teste verifica se 95% das requisições feitas ao endpoint retornam com um tempo de resposta abaixo de 3 segundos.

Resultados do Caso de teste 01:

```
silvio@TERMINAL MINGW64 ~/OneDrive/Área de Trabalho/k6 teste
```

```
execution: local
```

```
script: k6_test.js
```

```
output: -
```

```
scenarios: (100.00%) 1 scenario, 100 max VUs, 11m30s max duration (incl. graceful stop):
```

```
* default: Up to 100 looping VUs for 11m0s over 4 stages (gracefulRampDown: 30s, gracefulStop: 30s)
```

```
data_received.....: 75 MB 114 kB/s
data_sent.....: 978 kB 1.5 kB/s
http_req_blocked.....: avg=2.65ms min=0s med=0s max=5.62s p(90)=0s p(95)=0s
http_req_connecting.....: avg=935.39µs min=0s med=0s max=319.47ms p(90)=0s p(95)=0s
✓ http_req_duration.....: avg=2.47s min=224.87ms med=2.46s max=2.90s p(90)=2.77s p(95)=2.79s
http_req_failed.....: 0.00% 0 out of 10825
http_req_receiving.....: avg=12.21ms min=0s med=1.52ms max=9.93s p(90)=4.06ms p(95)=6.04ms
http_req_sending.....: avg=85.93µs min=0s med=0s max=1.2ms p(90)=253.9µs p(95)=290.7µs
http_req_tls_handshaking.....: avg=1.71ms min=0s med=0s max=5.31s p(90)=0s p(95)=0s
http_req_waiting.....: avg=2.46s min=223.89ms med=2.48s max=6.81s p(90)=4.13s p(95)=4.42s
http_reqs.....: 10825 16.373174/s
iteration_duration.....: avg=3.47s min=1.22s med=3.49s max=12.3s p(90)=5.14s p(95)=5.44s
iterations.....: 10825 16.373174/s
vus.....: 1 min=1 max=100
vus_max.....: 100 min=100 max=100
```

```
running (11m01.1s), 000/100 VUs, 10825 complete and 0 interrupted iterations
```

```
default ✓ [=====] 000/100 VUs 11m0s
```

ANÁLISE DOS RESULTADOS:

- **Taxa de Requisições:** Durante a execução, foram realizadas 10.825 requisições.
- **Taxa de Requisições por Segundo:** Aproximadamente 16,37 requisições por segundo.
- **Tempo de Resposta:**
 - O tempo médio das requisições foi de 2,47 segundos.
 - 95% das requisições foram processadas em até 2,79 segundos, atendendo ao critério do teste de carga.
 - O tempo de resposta mínimo foi de 224,87ms e o máximo foi de 2,90 segundos.
- **Taxa de Erros:** Não houve falhas nas requisições (0% de falhas).
- **Latência e Conectividade:** O tempo médio de conexão foi de 935,39µs, com o tempo médio de recepção de dados sendo 12,21ms. O tempo médio de envio foi muito baixo, em torno de 85,93µs.

Conclusão: O teste de carga foi bem-sucedido. A aplicação atendeu aos critérios estabelecidos, com a maioria das requisições processadas dentro do tempo limite de 3 segundos, e sem falhas nas requisições. O desempenho da aplicação foi estável, mesmo com a carga de 100 usuários.

Resultado esperado: o caso de teste 01passa.

Resultado obtido: o caso de teste 01passa.

Caso de teste 02

Código do Teste:

```
import http from 'k6/http';
import { sleep } from 'k6';

export const options = {
  // Config
  stages: [
    { duration: '2m', target: 50 },
    { duration: '2m', target: 100 },
    { duration: '1m', target: 200 },
    { duration: '5m', target: 200 },
    { duration: '2m', target: 300 },
    { duration: '1m', target: 300 },
    { duration: '1m', target: 150 },
    { duration: '1m', target: 75 },
    { duration: '10m', target: 0 },
  ],
  thresholds: {

    http_req_duration: ['p(95)<3000']    }
};

export default () => {
  const urlRes = http.get('https://orca-app-9k2ie.ondigitalocean.app/');
  sleep(1);
};
```

Descrição:

Este caso de teste foi projetado para realizar um teste de estresse na aplicação, com o objetivo de avaliar o comportamento do sistema sob alta carga de usuários simultâneos. O teste aumenta gradualmente o número de usuários virtuais (VUs), atingindo até 300 VUs, para identificar o ponto de falha ou o limite de capacidade do sistema.

- **Objetivo do Teste:** Avaliar o desempenho da aplicação sob carga extrema e verificar até onde o sistema pode suportar usuários simultâneos, mantendo a resposta dentro de tempos aceitáveis.
- **Crescimento de Carga:** O número de VUs começa em 50 e vai até 300 de forma gradual ao longo do teste, com picos de carga chegando a 300 VUs por até 5 minutos.

- **Redução Gradual:** Após o pico de carga, a quantidade de usuários é reduzida, até 0 VUs ao longo de 10 minutos.
- **Critério de Sucesso:** 95% das requisições devem ter um tempo de resposta inferior a 3 segundos.

Métrica de desempenho:

- O teste verifica se 95% das requisições feitas ao endpoint retornam com um tempo de resposta abaixo de 3 segundos.
-

Resultados do Caso de teste 02:

```

      /\      Grafana  /\
     /\  /\    | \   /\
    /\  /\  \  |  |  /\
   /\  /\    \ |  |  /\
  /\  /\      \|  |  /\
 /\  /\       \|  |  /\
/\  /\        \|  |  /\
\  /\         \|  |  /\
 \ /\          \|  |  /\
  \ /\          \|  |  /\
   \ /\          \|  |  /\
    \ /\          \|  |  /\
     \ /\          \|  |  /\
      \ /\          \|  |  /\
       \ /\          \|  |  /\
        \ /\          \|  |  /\
         \ /\          \|  |  /\
          \ /\          \|  |  /\
           \ /\          \|  |  /\
            \ /\          \|  |  /\
             \ /\          \|  |  /\
              \ /\          \|  |  /\
               \ /\          \|  |  /\
                \ /\          \|  |  /\
                 \ /\          \|  |  /\
                  \ /\          \|  |  /\
                   \ /\          \|  |  /\
                    \ /\          \|  |  /\
                     \ /\          \|  |  /\
                      \ /\          \|  |  /\
                       \ /\          \|  |  /\
                        \ /\          \|  |  /\
                         \ /\          \|  |  /\
                          \ /\          \|  |  /\
                           \ /\          \|  |  /\
                            \ /\          \|  |  /\
                             \ /\          \|  |  /\
                              \ /\          \|  |  /\
                               \ /\          \|  |  /\
                                \ /\          \|  |  /\
                                 \ /\          \|  |  /\
                                  \ /\          \|  |  /\
                                   \ /\          \|  |  /\
                                    \ /\          \|  |  /\
                                     \ /\          \|  |  /\
                                      \ /\          \|  |  /\
                                       \ /\          \|  |  /\
                                        \ /\          \|  |  /\
                                         \ /\          \|  |  /\
                                          \ /\          \|  |  /\
                                           \ /\          \|  |  /\
                                            \ /\          \|  |  /\
                                             \ /\          \|  |  /\
                                              \ /\          \|  |  /\
                                               \ /\          \|  |  /\
                                                \ /\          \|  |  /\
                                                 \ /\          \|  |  /\
                                                  \ /\          \|  |  /\
                                                    \ /\          \|  |  /\
                                                     \ /\          \|  |  /\
                                                      \ /\          \|  |  /\
                                                       \ /\          \|  |  /\
                                                        \ /\          \|  |  /\
                                                         \ /\          \|  |  /\
                                                          \ /\          \|  |  /\
                                                           \ /\          \|  |  /\
                                                            \ /\          \|  |  /\
                                                             \ /\          \|  |  /\
                                                              \ /\          \|  |  /\
                                                               \ /\          \|  |  /\
                                                                \ /\          \|  |  /\
                                                                 \ /\          \|  |  /\
                                                                  \ /\          \|  |  /\
                                                                   \ /\          \|  |  /\
                                                                    \ /\          \|  |  /\
                                                                     \ /\          \|  |  /\
                                                                      \ /\          \|  |  /\
                                                                       \ /\          \|  |  /\
                                                                        \ /\          \|  |  /\
                                                                         \ /\          \|  |  /\
                                                                          \ /\          \|  |  /\
                                                                           \ /\          \|  |  /\
                                                                            \ /\          \|  |  /\
                                                                             \ /\          \|  |  /\
                                                                              \ /\          \|  |  /\
                                                                               \ /\          \|  |  /\
                                                                                \ /\          \|  |  /\
                                                                                 \ /\          \|  |  /\
                                                                                  \ /\          \|  |  /\
                                                                                                                                
execution: local
script: scripy.js
output: -

scenarios: (100.00%) 1 scenario, 300 max VUs, 25m30s max duration (incl. graceful stop):
    * default: Up to 300 looping VUs for 25m0s over 9 stages (gracefulRampDown: 30s, gracefulStop: 30s)

data_received.....: 865 MB 576 kB/s
data_sent.....: 4.8 MB 3.2 kB/s
http_req_blocked.....: avg=244.09µs min=0s med=0s max=476.33ms p(90)=0s p(95)=0s
http_req_connecting.....: avg=103.8µs min=0s med=0s max=349.95ms p(90)=0s p(95)=0s
X http_req_duration.....: avg=2.34s min=240.5ms med=2s max=8.84s p(90)=5.01s p(95)=5.73s
  { expected_response:true }...: avg=2.34s min=240.5ms med=2s max=8.84s p(90)=5.01s p(95)=5.73s
http_req_failed.....: 0.01% 7 out of 51574
http_req_receiving.....: avg=14.11ms min=0s med=4.27ms max=3.41s p(90)=14.48ms p(95)=120.3ms
http_req_sending.....: avg=36.62µs min=0s med=0s max=46.71ms p(90)=0s p(95)=329.33µs
http_req_tls_handshaking.....: avg=138.32µs min=0s med=0s max=126.38ms p(90)=0s p(95)=0s
http_req_waiting.....: avg=2.32s min=233.04ms med=1.99s max=7.5s p(90)=5.01s p(95)=5.71s
http_reqs.....: 51574 34.362893/s
iteration_duration.....: avg=3.34s min=1.24s med=3s max=9.88s p(90)=6.01s p(95)=6.73s
iterations.....: 51574 34.362893/s
vus.....: 1 min=1 max=300
vus_max.....: 300 min=300 max=300

running (25m00.9s), 000/300 VUs, 51574 complete and 0 interrupted iterations
default ✓ [=====] 000/300 VUs 25m0s

```

ANÁLISE DOS RESULTADOS:

- **Taxa de Requisições:** Durante o teste, foram realizadas 51.574 requisições.
- **Taxa de Requisições por Segundo:** Aproximadamente 34,36 requisições por

segundo.

- **Tempo de Resposta:**
 - O tempo médio das requisições foi de 2,34 segundos.
 - 95% das requisições foram processadas em até 5,73 segundos, o que não atendeu ao critério de 3 segundos, indicando uma sobrecarga no sistema.
 - O tempo de resposta mínimo foi de 240,5ms, mas o máximo foi de 8,84 segundos, sugerindo que o sistema experimentou picos de latência sob carga extrema.
- **Taxa de Erros:** A taxa de falhas foi de 0,01%, com 7 falhas em 51.574 requisições, indicando que, embora o sistema tenha se comportado bem na maior parte do tempo, houve algumas falhas durante o pico de carga.
- **Latência e Conectividade:** O tempo médio de conexão foi de 103,8μs e o tempo médio de recepção de dados foi de 14,11ms, com variações dependendo da carga. O tempo de envio de dados foi baixo, em torno de 36,62μs.

Conclusão: O teste de estresse revelou que o sistema pode suportar uma carga significativa de até 300 usuários simultâneos, mas com uma degradação no tempo de resposta (95% das requisições foram processadas em até 5,73 segundos, acima do limite de 3 segundos esperado). Além disso, houve uma pequena taxa de falhas, indicando que o sistema começou a mostrar sinais de instabilidade à medida que a carga aumentava.

Resultado esperado: o caso de teste 02 falhe.

Resultado obtido: o caso de teste 02 falhe.

REFERÊNCIAS:

<https://k6.io/open-source/>