

# Load Test (Teste de Carga )

## Critérios de Aceitação – Fluxo Produtos

### Objetivo

Validar se o sistema consegue responder rapidamente às consultas de produtos, mesmo sob carga constante de 50 VUs.

### Critérios de aceitação

- **Todas as requisições devem retornar status 200**
- O JSON deve conter o campo **produtos** e ser um array válido
- O produto detalhado deve ter o campo **nome**
- **Taxa de falhas** deve ser **menor que 1%**
- **Taxa de checks bem-sucedidos** deve ser **maior que 95%**
- **95% das requisições devem responder abaixo de 100 ms**
- Deve executar com:
  - **50 VUs**
  - **Durante 10 minutos**

---

## Critérios de Aceitação – Fluxo Usuários

### Objetivo

Validar se o sistema suporta criação e consulta de usuários de forma extremamente rápida, mesmo com carga paralela.

### Critérios de aceitação

- **Criação de usuário deve retornar status 201**
- Consultas devem retornar **status 200**
- **Taxa de falhas** deve ser **menor que 1%**

- **Taxa de checks com sucesso** deve ser **acima de 95%**
- **95% das requisições** devem responder abaixo de 50 ms
- Deve executar com:
  - **5 VUs**
  - **Durante 10 minutos**

## Script Load Test

```
import http from 'k6/http';
import { check, group } from 'k6';

const BASE_URL = 'http://localhost:3000';

export const options = {
  scenarios: {
    produtos: {
      executor: 'constant-vus',
      vus: 50,
      duration: '10m',
      exec: 'fluxoProdutos',
    },
    usuarios: {
      executor: 'constant-vus',
      vus: 5,
      duration: '10m',
      exec: 'fluxoUsuarios',
    },
  },
  thresholds: {
    http_req_failed: ['rate<0.01'],
    checks: ['rate>0.95'],

    'http_req_duration{group:::Fluxo de Produtos}': ['p(95)<100'],
    'http_req_duration{group:::Fluxo de Usuários}': ['p(95)<50'],
  }
}
```

```

};

export function fluxoProdutos() {
  group('Fluxo de Produtos', () => {
    const resListar = http.get(`${BASE_URL}/produtos`);
    check(resListar, {
      'Listar Produtos - status 200': (r) => r.status === 200,
      'Listar Produtos - array valido': (r) => Array.isArray(r.json('produtos')),
    });

    const produtos = resListar.json('produtos');

    const index = Math.floor(Math.random() * produtos.length);
    const id = produtos[index]._id;

    const resDetalhe = http.get(`${BASE_URL}/produtos/${id}`);
    check(resDetalhe, {
      'Detalhe Produto - status 200': (r) => r.status === 200,
      'Detalhe tem nome': (r) => r.json('nome') !== undefined,
    });
  });
}

export function fluxoUsuarios() {
  group('Fluxo de Usuários', () => {
    const payload = JSON.stringify({
      nome: `Usuario${__VU}-${Date.now()}`,
      email: `usuario${__VU}-${Date.now()}@teste.com`,
      password: 'senha123',
      administrador: 'true',
    });

    const resCriar = http.post(`${BASE_URL}/usuarios`, payload, { header
s: { 'Content-Type': 'application/json' } });

    check(resCriar, {
      'Criar Usuário - status 201': (r) => r.status === 201,
    });
  });
}

```

```
});  
const id = resCriar.json('_id');  
  
const resDetalhe = http.get(`${BASE_URL}/usuarios/${id}`);  
check(resDetalhe, {  
  'Detalhe Usuário - status 200': (r) => r.status === 200,  
});  
});  
}
```