***Programa de***

***TESTES & QUALIDADE DE SOFTWARE***

***Desafio 2***



**INTEGRANTES DO GRUPO:**

Aline Veras - [LinkedIn](http://linkedin.com/in/aalineveras)

Antônio Carlos - [LinkedIn](http://linkedin.com/in/antonio-carlos-6249b383)

Fernando Kenji - [LinkedIn](http://linkedin.com/in/fkenjikamei)

Tatiana Honda - [LinkedIn](http://linkedin.com/in/tatiana-honda)

**DESCRIÇÃO**

O **Grafana k6** é uma ferramenta de teste de carga, de código aberto (*open source)* e de software como serviço (*SaaS*) baseado nas nuvens, que torna o teste de desempenho fácil e produtivo para as equipes de engenharia. O k6 é gratuito, centrado no desenvolvedor e extensível.

Usando o k6, você pode testar a confiabilidade e o desempenho de seus sistemas, e identificar regressões e problemas de desempenho com antecedência. Assim, o k6 ajudará você a criar aplicativos resilientes e de alto desempenho que podem ser dimensionados.

**PRINCIPAIS FUNCIONALIDADES**

Seus principais recursos são:

* Geração de carga configurável. Mesmo as máquinas mais simples podem simular muito tráfego.
* Testes como código. Reutilize scripts, modularize lógica, controle de versão e integre testes com seu CI.
* Uma API completa. A API de script é repleta de recursos que ajudam a simular o tráfego real do aplicativo.
* Um mecanismo JavaScript incorporado. O desempenho de Go, a familiaridade de script de JavaScript.
* Suporte a múltiplos protocolos. HTTP, WebSockets, gRPC e muito mais.
* Ecossistema de grande extensão. Você pode estender o k6 para atender às suas necessidades. E muitas pessoas já compartilharam suas extensões com a comunidade!
* Armazenamento e visualização de métricas flexíveis. Estatísticas resumidas ou métricas granulares, exportadas para o serviço de sua escolha.

**3 PONTOS POSITIVOS**

**1.** Teste de carga  
O k6 é otimizado para consumo mínimo de recursos e projetado para executar testes de alta carga ([testes de pico](https://k6.io/docs/test-types/stress-testing/#spike-testing-in-k6), [estresse](https://k6.io/docs/test-types/stress-testing/) e [imersão](https://k6.io/docs/test-types/soak-testing/)).

**2.** Teste do navegador  
Por meio do [navegador k6](https://k6.io/docs/javascript-api/k6-browser/), você pode executar testes de desempenho baseados em navegador e detectar problemas relacionados apenas a navegadores, que podem ser totalmente ignorados no nível do protocolo.

**3.** Caos e testes de resiliência  
Você pode usar o k6 para simular o tráfego como parte de seus experimentos de caos, acioná-los a partir de seus testes k6 ou injetar diferentes tipos de falhas no Kubernetes com [xk6-disruptor](https://k6.io/docs/javascript-api/xk6-disruptor/).

**3 PONTOS NEGATIVOS**

**1.** Recurso não encontrado

No k6, não foi possível determinar uma janela para execução do cenário. Por exemplo: Execute 3000 requisições com 100 usuários em 1 minuto.

“Em 1 minuto” foi uma configuração não encontrada na ferramenta. Com isso, precisamos calibrar o número de requisições e usuários para caber o tempo necessário.

**2.** Não é executado nativamente em um navegador  
Por padrão, o k6 não renderiza páginas da Web da mesma forma que um navegador. Os navegadores podem consumir recursos significativos do sistema. Ignorar o navegador permite executar mais carga em uma única máquina.  
No entanto, com [o navegador k6](https://k6.io/docs/javascript-api/k6-browser/), você pode interagir com navegadores reais e coletar métricas de front-end como parte de seus testes k6 .

**3.** Não roda em NodeJS  
JavaScript geralmente não é adequado para alto desempenho. Para atingir o desempenho máximo, a própria ferramenta é escrita em Go, incorporando um tempo de execução JavaScript que permite scripts de teste fáceis.  
Se você deseja importar módulos ou bibliotecas npm usando APIs NodeJS, pode [agrupar módulos npm com webpack](https://k6.io/docs/using-k6/modules/#bundling-node-modules) e importá-los em seus testes.

**BOAS PRÁTICAS**

* Resiliente de código: executando [testes grandes](https://k6.io/docs/testing-guides/running-large-tests/);
* Use matriz compartilhada para obter dados: [SharedArray](https://k6.io/docs/javascript-api/k6-data/sharedarray/);
* Não coloque muita lógica na seção Init (consome muita CPU, portanto, nas execuções na nuvem, pode falhar);
* Use opções em um arquivo separado;
* Use cenários para separar seus resultados;
* Use tags em sua solicitação GET (caso contrário, seus relatórios terão um grande número de resultados).

**PREÇO**

Eles oferecem um teste gratuito para 50 testes, enquanto os planos pagos começam a partir de **US$ 89/mês** para 100 usuários virtuais e 600 execuções de testes/ano, podendo até existir um plano personalizado com base nas necessidades exclusivas da empresa.

**CURIOSIDADE**

O k6 foi lançado em 2017. k6 foi originalmente construído e é mantido por [Load Impact](https://loadimpact.com/) - um serviço de teste de carga de SaaS. O Load Impact tem várias pessoas trabalhando em tempo integral no k6 e isso, junto com as contribuições da comunidade, mantém o desenvolvimento deste software muito ativo.

Menos conhecido é o motivo pelo qual essa ferramenta é chamada de "k6", mas estou feliz em divulgar essa informação aqui: após uma longa batalha interna de nomes que terminou em um impasse, tínhamos um nome de 7 letras começando com "k" que a maioria das pessoas odiava, então o nome foi reduzido para "k6", o que pareceu resolver o problema.

**FONTES CONSULTADAS**

<https://github.com/grafana/k6>

<https://k6.io/>

<https://k6.io/docs/>

<https://k6.io/blog/comparing-best-open-source-load-testing-tools/#tool-overview>

<https://jdsc.me/category/arquitetura/>

<https://kinsta.com/pt/blog/ferramentas-teste-desempenho/>

[Live #34 - Boas práticas na adoção de testes de performance com k6](https://www.youtube.com/watch?v=HuXbR1rbVaM)

<https://community.k6.io/t/k6-best-practices/4715>

<https://k6.io/docs/javascript-api/k6-data/sharedarray/>

<https://k6.io/docs/testing-guides/running-large-tests/>

<https://k6.io/docs/testing-guides/running-large-tests/#make-your-test-code-resilient>