Dashboard Nativo do K6

1. Introdução

O K6 sempre foi conhecido por ser extremamente leve, eficiente e centrado na linha de comando.

Porém, por muitos anos, quem queria **acompanhar métricas em tempo real** dependia de ferramentas externas, como:

- Grafana + Prometheus
- InfluxDB
- Plugins de terceiros que exibiam gráficos ao vivo

Isso funcionava, mas exigia configuração adicional.

Para resolver isso, a partir da versão **0.49**, o K6 passou a incluir **um dashboard nativo**, integrado ao próprio binário, sem dependências externas.

Esse painel oferece:

- Visualização das métricas *durante* a execução
- Painéis organizados por categorias
- Exportação final em HTML

---

2. Como o Dashboard Nativo Funciona

Quando você habilita esse recurso, o K6 inicia um pequeno servidor local no próprio processo.

Esse servidor:

- expõe um endpoint HTTP
- envia as métricas que o K6 está coletando
- renderiza um dashboard moderno e interativo diretamente no navegador

Ele **não salva dados depois que a execução termina**, a menos que você peça para exportar.

O fluxo é assim:

1. Você roda o K6 com a flag K6_WEB_DASHBOARD
2. O K6 exibe no terminal:

   “Dashboard available at http://localhost:####”
3. Você abre no navegador e acompanha tudo ao vivo
4. No final da execução, pode exportar o dashboard como HTML

---

3. Habilitando o Dashboard

Comando básico:

```
$env:K6_WEB_DASHBOARD="true" ; k6 run script.js
```

O K6 automaticamente escolherá uma porta disponível e vai imprimir algo assim:

```
web dashboard: http://127.0.0.1:5665
```

---

4. O que você encontra dentro do Dashboard

O dashboard tem várias seções.

Aqui está um detalhamento aprofundado do que aparece e para que serve cada gráfico.

4.1. Resumo geral

Uma visão rápida do comportamento do teste:

- Taxa de requisições por segundo
- Taxa de envio (*send rate*)
- Erros recentes
- Status das VUs (ativas, pendentes, terminadas)

Essa área serve para identificar imediatamente se:

- a carga está sendo aplicada corretamente
- o sistema está respondendo dentro do esperado

- há falhas concentradas em determinado intervalo

4.2. Métricas de Requisições HTTP

Exibe:

- **Duração (Response time)**

  Distribuição do tempo de resposta ao longo do teste
- **RPS (Requests per second)**
- **Taxa de erros por segundo**
- **HTTP Status codes**

  (200, 400, 404, 500 etc.)

Esses gráficos ajudam a responder perguntas importantes como:

- O sistema mantém desempenho sob carga?
- Os tempos de resposta estão estáveis ou crescentes?
- Em que momento começaram a surgir erros?

4.3. Métricas de Iteração (VU)

Cada VU executa um ciclo de iteração.

O dashboard mostra:

- Tempo médio por iteração
- Variação ao longo da execução
- Possíveis gargalos dentro da lógica do script

Quando o tempo de iteração aumenta demais, geralmente significa:

- A API ficou lenta
- Alguma dependência externa travou
- O servidor saturou

4.4. Taxa de Transferência (Throughput)

Inclui:

- bytes enviados por segundo

- bytes recebidos por segundo

Isso permite avaliar se existe:

- consumo exagerado de banda
- picos de tráfego inesperados
- estabilidade no fluxo de transferência

4.5. Resumo final (Summary)

É o *summary* padrão do K6 renderizado de forma visual.

Inclui:

- p(90), p(95), p(99)
- média e mediana
- total de requisições
- taxa de erro global
- duração total

Essa visão é ideal para anexar em relatórios ou apresentações.

---

5. Exportando o Dashboard em HTML

O K6 permite exportar todos os gráficos como um arquivo HTML estático.

Esse arquivo contém:

- todos os dados do teste
- todos os gráficos
- uma visualização praticamente idêntica ao dashboard ao vivo

Para gerar automaticamente ao final:

```
$env:K6_WEB_DASHBOARD="true" ; $env:K6_WEB_DASHBOARD_EXPORT="report.htm"; k6 run script.js
```

O nome do arquivo você escolhe, exemplo:

```
meu_teste_api.html
resultado_carga.html
dashboard_login.html
```

Não há necessidade de servidor para visualizar o arquivo:

é só abrir no navegador.

Excelente para:

- enviar para gestores
- anexar no CI/CD (Jenkins, GitLab, GitHub Actions)
- armazenar histórico
- fazer comparações entre execuções