Métricas

2 — Métricas padrão do K6

O K6 gera **métricas automaticamente** em toda execução, sem precisar configurar nada.

Elas são a base para analisar o comportamento da aplicação.

Métrica	O que representa	Exemplo prático
`http_reqs`	Quantidade total de requisições HTTP	Se 10 usuários fazem 5 requisições cada → 50
`http_req_duration`	Tempo total de uma requisição (DNS + connect + send + wait + receive)	Mede a **latência total** percebida pelo usuário
`iterations`	Quantas vezes a função `default()` foi executada	Cada iteração equivale a um ciclo completo de teste
`vus` / `vus_max`	Usuários virtuais ativos e máximo atingido	Mostra o nível de carga aplicado
`data_sent` / `data_received`	Volume total de bytes enviados e recebidos	Útil para entender o consumo de rede
`checks`	Contador de validações (passaram ou falharam)	Mostra se o sistema manteve consistência sob carga

2.1 — Métricas detalhadas de tempo (`res.timings`)

O K6 mede o **tempo total da requisição HTTP** e também **quanto tempo foi gasto em cada etapa** dela.

Pense em uma requisição HTTP como uma **conversa entre seu navegador e o servidor**:

1. Primeiro você **abre uma conexão** com o servidor (connect).
2. Em seguida, **envia os dados** da requisição (send).
3. O servidor **processa e começa a responder** (waiting).
4. Por fim, você **recebe todos os dados da resposta** (receiving).

Cada uma dessas etapas tem um tempo próprio, e o K6 mede tudo isso automaticamente dentro do objeto `res.timings`.

Exemplo prático

```
import http from 'k6/http';

export const options = {
  vus: 1,
  iterations: 1,
};

export default function () {
  const res = http.get('https://serverest.dev/carrinhos');

  console.log(`
  Tempo total: ${res.timings.duration} ms
  Conexão: ${res.timings.connecting} ms
  Envio (request): ${res.timings.sending} ms
  Espera (servidor pensando): ${res.timings.waiting} ms
  Recebimento (response): ${res.timings.receiving} ms
 `);
}
```

Exemplo real de saída no console

```
  Tempo total: 206.1665 ms
  Conexão: 19.3373 ms
  Envio (request): 0 ms
  Espera (servidor pensando): 188.1432 ms
  Recebimento (response): 18.0233 ms
```

Entendendo cada campo

Campo	O que mede	Interpretação prática
`res.timings.duration`	Tempo total da requisição (do início ao fim)	Tempo “percebido” pelo usuário
`res.timings.connecting`	Tempo gasto para abrir a conexão TCP	Se alto, pode haver lentidão na rede
`res.timings.sending`	Tempo para enviar a requisição	Geralmente baixo (dados pequenos)
`res.timings.waiting`	Tempo que o servidor leva para começar a responder (TTFB – *Time To First Byte*)	**Principal métrica de backend**, mostra tempo de processamento
`res.timings.receiving`	Tempo para receber toda a resposta	Pode indicar lentidão na entrega dos dados
`res.timings.blocked`	Tempo total bloqueado antes do envio (ex.: fila de conexão)	Relevante em cenários com muitos usuários virtuais