

3. Versão com Groups + Tags

As **tags** no K6 servem para **classificar, organizar e filtrar os dados** do teste de performance.

Elas não mudam o comportamento da requisição, mas **mudam completamente como você analisa os resultados**.

Quando você adiciona tags assim:

```
tags: {  
  endpoint: 'produtos',  
  acao: 'listar',  
  modulo: 'catalogo'  
}
```

Você está “marcando” essa requisição com informações que vão aparecer nos relatórios, dashboards (Grafana), métricas e logs.

```
import http from 'k6/http';  
import { group } from 'k6';  
  
export default function () {  
  
  group('01 - Listar produtos', () => {  
    http.get('http://localhost:3000/produtos', {  
      tags: {  
        endpoint: 'produtos',  
        acao: 'listar',  
        modulo: 'catalogo'  
      }  
    });  
  });  
  
  group('02 - Detalhar produto', () => {  
    http.get('http://localhost:3000/produtos/BeeJh5Iz3k6kSlzA', {  
      tags: {  
        endpoint: 'produtos',  
        acao: 'detalhar',  
      }  
    });  
  });  
}
```

```

    modulo: 'catalogo'
  }
});
});

group('03 - Listar usuários', () => {
  http.get('http://localhost:3000/usuarios', {
    tags: {
      endpoint: 'usuarios',
      acao: 'listar',
      modulo: 'admin'
    }
  });
});
});

}

```

Por que isso é extremamente útil no mundo real?

1. Você sabe exatamente o que está lento

Ex.:

- p95 alto apenas em `/produtos/BeeJh5Iz3k6kSIzA`
- apenas o módulo `admin` está degradado
- apenas operações de `detalhar` estão lentas

Isso dá munição técnica para conversar com o time de Dev.

2. Thresholds ficam muito mais inteligentes

Você pode criar metas separadas por módulo, endpoint ou ação:

```

export const options = {
  vus: 10,
  duration: '10s',
  thresholds: {
    'http_req_duration{modulo:catalogo}': ['p(95)<9'],
    'http_req_duration{modulo:admin}': ['p(95)<9'],
  }
}

```

```
'http_req_duration{acao:listar}': ['p(95)<9'],  
  
'http_req_duration{group:::01 - Listar produtos}': ['p(95)<8'],  
},  
};
```

Agora o teste falha **apenas no que realmente importa**, e não no conjunto inteiro.

3. Dashboard extremamente claro

Com Tags, o Grafana mostra gráficos filtráveis por:

- endpoint
- módulo
- tipo de ação
- fluxo
- operações críticas

Assim você consegue responder perguntas reais de negócio, como:

┃ “O catálogo está rápido, mas o detalhamento está lento?”

┃ “O módulo administrativo está piorando conforme subimos usuários?”

4. Facilita encontrar gargalos em sistemas grandes

Quando o sistema tem muitas rotas, como:

- catálogo
- estoque
- usuários
- pedidos
- pagamentos
- shipping
- carrinho

Tags permitem medir cada parte separadamente, sem precisar criar arquivos separados.