

Testes em Aplicações Web com k6 (Browser)

O K6 nasceu como uma ferramenta focada em **testes de performance de back end**, permitindo medir diretamente o tempo de resposta de APIs, taxas de erro e throughput. Entretanto, testes apenas HTTP **não capturam a experiência do usuário**, que depende de como a interface é renderizada e interage com o usuário final.

Por exemplo, testes HTTP **não conseguem avaliar**:

- Lentidão no carregamento de scripts JavaScript
- Problemas de layout ou renderização visual
- Elementos que aparecem atrasados ou não são visíveis na tela
- A experiência de interação real, como cliques, preenchimento de formulários e navegação

Para preencher essa lacuna, o K6 passou a oferecer o módulo **K6 Browser**, que permite realizar testes de performance **diretamente no front end**, simulando a navegação de usuários reais.

Como funciona o K6 Browser

O módulo inicia uma instância do navegador, baseado na arquitetura do **Playwright**, o que permite simular a navegação de usuários reais. Com isso, o navegador carrega toda a página , HTML, CSS, scripts e imagens e executa as interações **como se fosse um usuário humano**, incluindo cliques, preenchimento de formulários e navegação entre páginas.

Isso possibilita medir aspectos que o HTTP sozinho não captura:

- **Renderização e layout:** tempo até a página estar completamente visível
 - **Interações de interface:** cliques, scroll, formulários e buscas
 - **Percepção do usuário:** quanto tempo leva até que a interface esteja responsiva
-

Tipos de testes de performance em aplicações web

Quando falamos em testes na camada web, podemos pensar em três abordagens principais:

1. Teste baseado em navegador

- Simula o comportamento de usuários reais navegando no site
- Foca na performance do front end
- Útil para medir tempo de renderização e interações visuais

2. Teste baseado em protocolo

- Mede a performance do back end através das solicitações que o front end faz
- Por exemplo, clicando em um botão e registrando o tempo de resposta do servidor
- Não mede renderização visual, mas avalia a eficiência do servidor

3. Teste híbrido

- Combina teste de navegador e teste de protocolo
 - Permite analisar métricas tanto do front end quanto do back end simultaneamente
 - Ideal para detectar gargalos em qualquer camada da aplicação
-