

TESTE DE PERFORMANCE

O QUE DEFINIR ANTES DE AUTOMATIZAR

O GUIA

#1



PASSO 1

QUER DESCOBRIR OU CONFIRMAR?

Seu teste de performance pode ter diversos objetivos, mas há uma coisa em comum: **A) Descobrir** até que ponto o sistema se comporta como esperado ou **B) Confirmar** que o sistema se comporta como esperado. O (A) é aplicado geralmente quando ainda **não temos informações** sobre como o sistema será consumido por quem o consome. Já o (B) é quando **sabemos exatamente** como o sistema será consumido.

PASSO 2

QUEM IRÁ CONSUMIR SEU SOFTWARE?

O consumidor aqui pode ser um **Ser Humano** ou um outro **Software**. Quando o consumidor do software que você testa é um ser humano, a forma de testar a performance deve levar em consideração **fatores como a quantidade que o humano leva para pensar na sua próxima ação** (User Think Time), algo que não acaba não sendo necessário em casos onde quem consome a API é um outro software que é executado de maneira automatizada, por exemplo.

PASSO 3

QUAL É O COMPORTAMENTO DO CONSUMIDOR?

O comportamento do consumidor inclui,
mas não se limita a:

- A) Quais são os principais cenários exercitados,
- B) Qual é a distribuição de execução dos cenários
- C) Quais são picos de utilização do sistema
- D) Quais são os principais dados de entrada utilizados
- E) Quais são os principais consumidores

PASSO 4

QUAL É A ARQUITETURA DO SISTEMA?

Entender qual é a arquitetura do sistema significa **saber quais componentes serão utilizados durante a execução dos testes de performance**. Alguns exemplos desses componentes podem ser A) APIs, B) Gateways, C) bancos de Dados, D) Filas, E) Serviços Externos, F) Micro-frontends e G) Sistemas Legados. Você precisa conhecê-los, entender sua relação com o sistema em si e prepará-los **antes mesmo de pensar em executar os testes de performance**, afinal, eles serão impactados diretamente.

PASSO 5

COMO MONITORAR OS COMPONENTES DURANTE O TESTE?

Ferramentas de testes de performance (ex. JMeter, Glatlin, Locust.io, etc) **geralmente apresentam apenas resultados baseados na requisição e resposta HTTP que foi enviada**, mas não vêem informações específicas do funcionamento dos componentes que compõe o sistema. Isso faz com que nossos resultados não sejam completos. Logo, para obter resultados **mais profundos, é necessário usar ferramentas de monitoramento de aplicações** (ex. New Relic).

PASSO 6

COMO LIDAR COM DADOS QUE SERÃO UTILIZADOS?

Testes de performance costumam usar um montante de dados muito grande para possibilitar a simulação de centenas ou milhares de usuários simultâneos. Pense em 500 usuários distintos fazendo login. Bem, você precisará de 500 usuários diferentes em sua base. Logo, **é necessário definir como lidar e preparar esses dados** para os testes.

COMO LIDAR COM SERVIÇOS EXTERNOS?

Durante fases mais iniciais, os serviços externos acabam sendo simulados. Logo, se meu sistema depende de um serviço da Amazon, por exemplo, eu uso um mock que simule esse serviço ao invés do serviço real. Conforme evoluímos no ciclo de desenvolvimento, passamos a usar o serviço real. Nesse caso, a execução de um teste de performance precisa ser um movimento planejado, de modo que o sistema externo esteja preparado para os testes (ou, em alguns caso, seja substituído por simuladores novamente).



**LEMBRE-SE: EM TESTES DE
PERFORMANCE, HÁ MUITO
MAIS DO QUE APENAS
ESCREVER UM TESTE
AUTOMATIZADO.**



TESTE DE PERFORMANCE

O QUE DEFINIR ANTES DE AUTOMATIZAR

O GUIA **#1**

Um passo a passo por vez, para ajudar você a aplicar algo na prática em seu dia a dia atuando em testes e qualidade de software



CURTA



COMPARTILHE



COMENTE



SALVE



@juliodelimasinsta