



Stress





Tópicos

Introdução

Fundamentos do Stress
Testing em Projetos Web

Planejamento e Preparação para o Stress Test

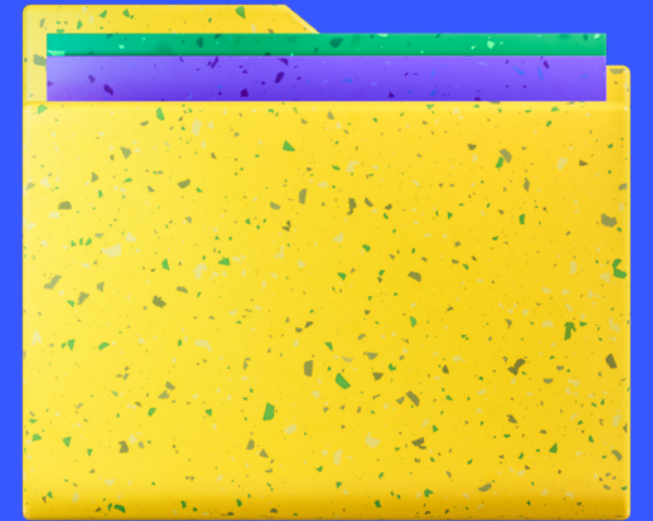
Execução do Stress Test

Conclusão

Análise de Resultados e
Melhoria Pós-Teste

O QUE É ?

O teste de Stress é um tipo de teste de performance que tem como objetivo verificar como um sistema ou aplicação web se comporta quando submetido a condições extremas de carga.



OBJETIVOS

- Entender até onde o sistema pode ir
- Verificar/descobrir falhas
- Verificar desempenho

METAS

- Desenvolver uma aplicação que suporte
- Resolver falhas antes do projeto ir pro user
- Alto desempenho

Objetivo

é identificar os limites do sistema em condições extremas, ou seja, determinar como ele lida com uma carga de usuários ou requisições muito superior ao esperado



Tambem ajuda em:



IDENTIFICAR GARGALOS:

Verificar pontos do sistema que podem não suportar um alto volume de requisições, como servidores, banco de dados, ou código.



AVALIAR A ESCALABILIDADE:

Entender como o sistema responde a um aumento massivo de tráfego.



GARANTIR A ESTABILIDADE:

- Assegurar que o sistema permaneça estável, mesmo quando operando em níveis de carga elevados, evitando quedas ou falhas catastróficas.

IDENTIFICAR CENÁRIOS

Pico de tráfego

Carga contínua

Falhas de rede

Sobrecarga no banco de dados

Teste de recuperação

FERRAMENTAS DE TESTE

JMeter

Gatling

LoadRunner

Ambiente para Teste

Servidores de aplicação

Banco de dados

Serviços de rede

Monitores de desempenho



Metodologias

Usuários Simultâneos: Iniciar com uma carga baixa e aumentar gradualmente até identificar o ponto de falha.

Duração: O teste deve durar entre 30 minutos e 2 horas.

Intensidade: Começar com uma carga moderada e aumentar a intensidade ao longo do teste.





Monitoramento e Coleta de Dados

Tempo de Resposta: Tempo para processar e responder requisições.

Taxa de Falhas: Número de erros (ex: timeouts, falhas de conexão).

Utilização de Recursos: CPU, memória, disco e rede.

Comportamento do Banco de Dados: Desempenho de consultas e transações.

Recuperação: Tempo necessário para o sistema se recuperar após falhas.

Escalonamento Progressivo

Carga Baixa: Iniciar com tráfego normal.

Aumento Gradual: Aumentar a carga progressivamente (10% a 20% por vez).

Pico de Carga: Testar o sistema com carga máxima e forçar ao limite.

Teste de Sustentação: Manter a carga máxima por um tempo.

Recuperação: Observar como o sistema se recupera após a falha.

O objetivo é identificar o ponto de falha do sistema, avaliar sua resiliência e desempenho sob carga extrema.

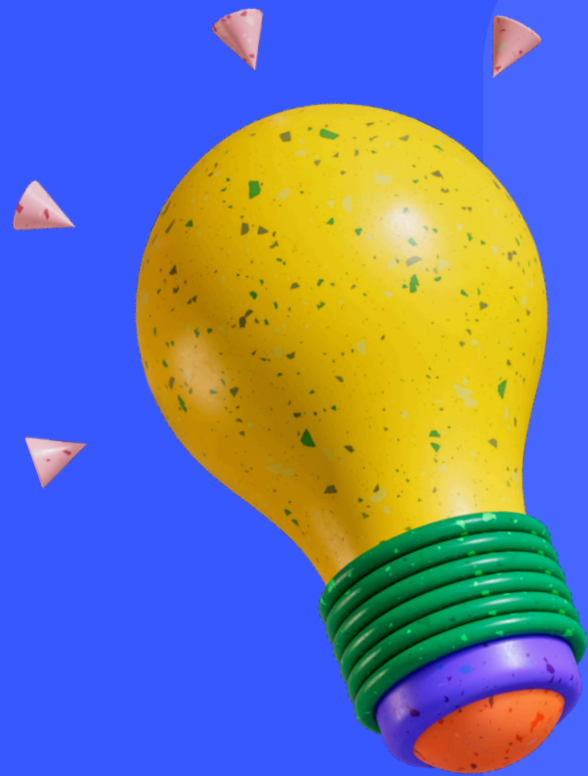
Interpretação de Dados

Ela é obtida durante os testes de desempenho e uma das etapas mais críticas do processo. Ao coletar métricas como tempos de resposta, uso de CPU, memória, largura de banda e taxas de erro, podemos obter uma visão clara do comportamento do sistema sob diferentes cargas de trabalho. A análise desses dados permite identificar onde o sistema está apresentando falhas ou gargalos de desempenho. Por exemplo, se houver um aumento significativo nos tempos de resposta durante picos de tráfego, isso pode indicar a necessidade de melhorias na infraestrutura ou ajustes no código. A interpretação eficaz dos dados é essencial para direcionar as ações corretivas de forma precisa e eficaz.



Dica: a colaboração facilita o trabalho em equipe! Clique em "Compartilhar" e convide seus colegas para preencher o quadro branco. Use esta página para reuniões, brainstorms e outras ideias divertidas da equipe!

Ações Corretivas



O próximo passo é identificar e implementar ações corretivas que visem melhorar o desempenho do sistema. Essas ações podem variar desde a otimização de código até melhorias na infraestrutura ou ajustes no balanceamento de carga. A otimização de código, por exemplo, pode incluir a refatoração de algoritmos para reduzir a complexidade computacional ou a implementação de técnicas como cache para reduzir a necessidade de processamento repetido. Já as melhorias na infraestrutura podem envolver a adição de mais recursos de hardware, como servidores mais rápidos ou maior capacidade de rede, para lidar com o volume de tráfego.

Revisão e Iteração



Após implementar as ações corretivas, é fundamental realizar uma revisão contínua do sistema por meio de novos testes de desempenho. Isso garante que as mudanças implementadas tenham realmente solucionado os problemas identificados e não tenham causado impactos negativos em outras áreas do sistema. A iteração é uma parte fundamental do processo de melhoria contínua. A cada novo teste, novas áreas podem ser descobertas que precisam de ajustes, ou o desempenho pode ser ainda mais otimizado.

Obrigada pela atenção