



TESTE DE SISTEMAS: VISÃO GERAL

Uma jornada completa pela validação de software em grandes empresas

Thales & Eduardo

Sistema Nacional de Aprendizagem Industrial - SENAI - Cascavel



O QUE É TESTE DE SISTEMAS?

O **teste de sistemas** é a fase em que o software é validado como um todo, já integrado, para verificar se ele funciona conforme os requisitos definidos.

Grandes empresas como **Google, Amazon, Netflix, Microsoft, Meta e bancos digitais** investem fortemente em testes de sistemas, pois qualquer falha pode gerar prejuízos financeiros e impacto na experiência do usuário.

PRINCIPAIS TÉCNICAS UTILIZADAS

TESTES FUNCIONAIS

Validam se o sistema faz o que foi especificado

TESTES DE REGRESSÃO

Garantem que novas mudanças não quebrem funcionalidades existentes

TESTES DE CARGA E DESEMPENHO

Avaliam o comportamento do sistema sob alto volume de usuários

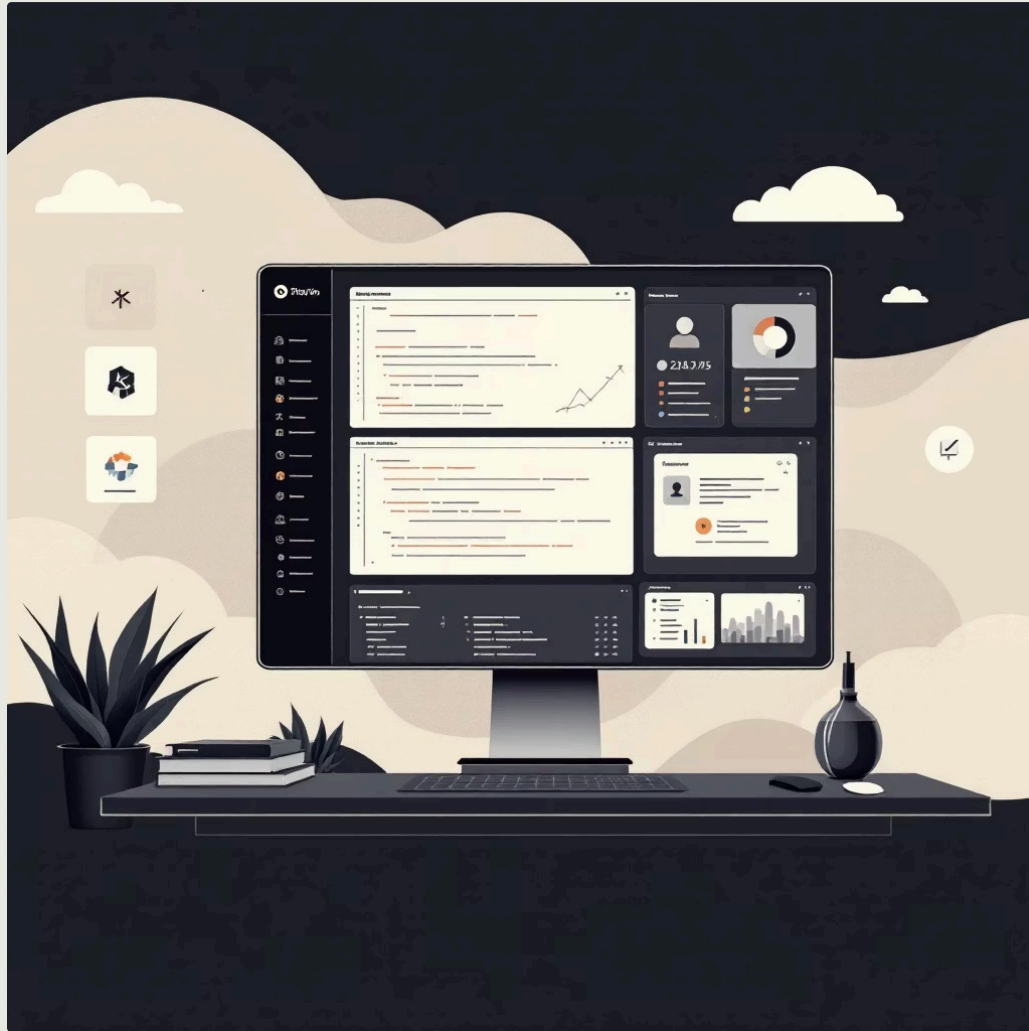
TESTES DE SEGURANÇA

Focados em vulnerabilidades

TESTES EXPLORATÓRIOS

O testador avalia o sistema sem roteiros rígidos

FERRAMENTAS MAIS USADAS NO MERCADO



AUTOMAÇÃO DE TESTES FUNCIONAIS

Selenium e Cypress para automação de testes funcionais

TESTES EM JAVA

JUnit e TestNG para testes em aplicações Java

TESTES DE APIS

Postman para testes de APIs

CARGA E DESEMPENHO

JMeter para carga e desempenho

SEGURANÇA

Burp Suite para segurança

ORGANIZAÇÃO

TestRail e Jira são amplamente adotadas para organização e acompanhamento dos testes

POR QUE TESTE DE SISTEMAS É ESSENCIAL?



QUALIDADE

Garante que o software atenda aos padrões de qualidade estabelecidos



CONFIABILIDADE

Assegura que o sistema funcione de forma consistente e previsível



SEGURANÇA

Protege contra vulnerabilidades e ataques externos

Em resumo, o teste de sistemas é essencial para garantir **qualidade, confiabilidade e segurança**, ajudando as empresas a entregarem softwares mais estáveis e alinhados às expectativas dos usuários.

SITUAÇÕES REAIS

CASOS DE TESTE DE SISTEMAS EM GRANDES EMPRESAS

A seguir, você verá exemplos reais de como empresas globais enfrentaram desafios críticos e utilizaram testes de sistemas para resolver problemas complexos.

CASOS REAIS: AMAZON, NETFLIX E META

1

AMAZON – LENTIDÃO E QUEDAS NO SITE

O sistema havia sido planejado para um volume menor de acessos, mas milhões de usuários estavam conectados ao mesmo tempo.

Pergunta: Como garantir que o site suportasse picos extremos de usuários?

Resposta: Simular grandes volumes de acesso antes dos eventos, para ajustar escalabilidade e desempenho.

*Técnica: Teste de carga e estresse /
Ferramenta: Apache JMeter*

2

NETFLIX – FALHAS INESPERADAS EM PRODUÇÃO

Mesmo com testes realizados, usuários relataram que o serviço ficava indisponível quando um servidor apresentava falha.

Pergunta: Como garantir que o sistema funcionasse mesmo quando partes da infraestrutura falhassem?

Resposta: Provocar falhas para validar a resistência do sistema.

Técnica: Teste de caos (Chaos Engineering) / Ferramenta: Chaos Monkey

3

FACEBOOK (META) – INDISPONIBILIDADE GLOBAL EM 2021

Durante uma alteração na configuração da infraestrutura, os principais serviços da Meta ficaram fora do ar por horas.

Pergunta: Como evitar que mudanças de infraestrutura causem quedas globais?

Resposta: Reforçou testes de sistema e regressão em mudanças críticas antes do deploy.

*Técnica: Teste de sistema e regressão /
Ferramenta: Automação interna baseada em Selenium*

CASOS REAIS: GOOGLE, UBER E MICROSOFT

1

GOOGLE – QUEDA NA TAXA DE CLIQUES

Após uma atualização visual na página de busca, analistas notaram uma redução no número de cliques dos usuários.

Pergunta: Como avaliar se uma mudança realmente melhora a experiência do usuário?

Resposta: O Google comparou versões diferentes da interface com usuários reais.

Técnica: Teste A/B | Ferramenta: Google Optimize (interno)

2

UBER – COBRANÇAS INCORRETAS APÓS ATUALIZAÇÃO

Alguns usuários relataram cobranças duplicadas.

Pergunta: Como garantir que serviços integrados funcionem corretamente após mudanças?

Resposta: A equipe intensificou os testes de integração entre APIs antes das liberações.

Técnica: Teste de integração | Ferramenta: Postman

3

MICROSOFT – REJEIÇÃO DO WINDOWS 8

Usuários demonstraram dificuldade para realizar tarefas básicas, como acessar programas.

Pergunta: Como identificar problemas de usabilidade antes do lançamento?

Resposta: A Microsoft passou a priorizar testes de usabilidade com usuários reais.

Técnica: Teste de usabilidade | Ferramenta: UserTesting

CASOS REAIS: PAYPAL, SPOTIFY E TESLA

1

PAYPAL – RISCOS DE SEGURANÇA

Durante auditorias internas, foi identificado que algumas APIs poderiam ser exploradas por ataques externos se não fossem testadas adequadamente.

Pergunta: Como detectar falhas de segurança antes que sejam exploradas?

Resposta: Foram realizados testes de invasão (pentests) de forma contínua.

Técnica: Teste de segurança / Ferramenta: Burp Suite

2

SPOTIFY – FALHAS NO MODO OFFLINE

Usuários relataram que músicas baixadas desapareciam ao alternar entre dispositivos.

Pergunta: Como encontrar problemas que surgem apenas no uso real do sistema?

Resposta: A equipe utilizou testes exploratórios, simulando diferentes comportamentos de usuários.

Técnica: Teste exploratório / Ferramenta: TestRail

3

TESLA – COMPORTAMENTO INESPERADO APÓS ATUALIZAÇÃO

Alguns veículos apresentaram respostas inesperadas do sistema de assistência ao motorista.

Pergunta: Como validar atualizações críticas antes e depois da liberação?

Resposta: Reforçar testes de sistema em ambientes simulados e reais, analisando dados coletados dos veículos.

Técnica: Teste de sistema / Ferramenta: Simuladores internos e telemetria



CASO REAL: BANCO ITAÚ

FALHAS APÓS ATUALIZAÇÃO DO APP

Funcionalidades que antes funcionavam corretamente apresentaram erros.

Pergunta: Como evitar que novas versões quebrem funções já existentes?

RESPOSTA

O banco adotou testes de regressão automatizados antes de publicar novas versões.

Técnica: Teste de regressão

Ferramenta: Cypress