

## Teste de Recuperação

- **Técnicas:** Focadas em simular falhas, como quedas de energia ou interrupções de rede, para verificar a recuperação do sistema.
- **Ferramentas:** Ferramentas como Chaos Monkey são usadas para simular falhas controladas, ajudando a verificar a resiliência do sistema.
- **Artefatos:** Relatórios de recuperação documentam os cenários de falha e tempos de recuperação alcançados.
- **Exemplo de Relatório:** Registra o tempo de recuperação e as falhas detectadas, evidenciando a eficiência dos mecanismos de redundância e failover

[Monitora](#)

[RoleCatcher | Your Competitive Edge](#)

.

## Teste de Regressão

- **Técnicas:** Testes automatizados para garantir que novas mudanças não afetem funcionalidades existentes.
- **Ferramentas:** Selenium e TestNG para automação de testes de interface e funcionalidade.
- **Artefatos:** Scripts automatizados e relatórios de execução que comparam versões para verificar a integridade do sistema.
- **Exemplo de Relatório:** Inclui uma matriz de rastreamento com funcionalidades testadas e eventuais defeitos reintroduzidos

[LoadView](#)

[Monitora](#)

.

## Teste de Confiabilidade

- **Técnicas:** Envolve testes de carga e estresse para avaliar estabilidade ao longo do tempo.
- **Ferramentas:** LoadRunner e Apache JMeter para testes de estresse e carga em diversos ambientes.
- **Artefatos:** Logs de execução e relatórios de falhas de estabilidade.
- **Exemplo de Relatório:** Documenta a estabilidade sob carga contínua e identifica falhas intermitentes que podem comprometer a confiabilidade a longo prazo

[Guru99](#)

## Teste de Segurança

- **Técnicas:** Análise de vulnerabilidades e pentesting para identificar falhas de segurança.
- **Ferramentas:** OWASP ZAP e Burp Suite são populares para verificação de vulnerabilidades em aplicativos web.
- **Artefatos:** Relatórios de vulnerabilidade que listam pontos fracos e riscos de segurança.
- **Exemplo de Relatório:** Detalha vulnerabilidades, com gravidade e recomendações de mitigação para cada ponto fraco identificado

## [Monitora](#)

## Teste de Desempenho

- **Técnicas:** Envolve simulação de usuários simultâneos para avaliar resposta e resistência.
- **Ferramentas:** Apache JMeter e Gatling são amplamente usados para simular tráfego de rede e carga de usuário.
- **Artefatos:** Gráficos de tempo de resposta e logs de carga máxima.
- **Exemplo de Relatório:** Mostra tempos de resposta e capacidade máxima antes da degradação do desempenho

## [Guru99](#)

## Teste de Usabilidade

- **Técnicas:** Observação de interações reais de usuários para otimizar a experiência.
- **Ferramentas:** UsabilityHub e Crazy Egg para capturar comportamento e mapear cliques.
- **Artefatos:** Relatórios de feedback de usabilidade e gravações de sessão.
- **Exemplo de Relatório:** Inclui mapas de calor e gravações, destacando áreas de melhoria

## [LoadView](#)

## [Monitora](#)

## Teste de Aceitação

- **Técnicas:** Avaliação feita pelo cliente para garantir o atendimento de requisitos.

- **Ferramentas:** FitNesse e Cucumber para automação de testes de aceitação baseados em especificações.
- **Artefatos:** Casos de teste de aceitação com validação do cliente.
- **Exemplo de Relatório:** Documenta se as funcionalidades atendem às especificações contratuais

[Guru99](#)

### Teste de Sistema

- **Técnicas:** Testes de ponta a ponta para garantir que o sistema funcione em conjunto.
- **Ferramentas:** IBM Rational e TestRail para orquestração e execução de testes completos.
- **Artefatos:** Casos de teste e logs de execução.
- **Exemplo de Relatório:** Um relatório de teste de sistema lista as funcionalidades principais e os problemas de integração detectados

[Monitora](#)

### Teste de API

- **Técnicas:** Envolve testar respostas, tempo de execução e segurança de APIs.
- **Ferramentas:** Postman e SoapUI para automação de requisições e validação de respostas.
- **Artefatos:** Documentação de endpoints e logs de resposta de APIs.
- **Exemplo de Relatório:** Inclui resultados de validação de campos e tempo de resposta, validando a conformidade com especificações

[Monitora](#)

[RoleCatcher | Your Competitive Edge](#)

### Teste de Integração

- **Técnicas:** Verifica a interação entre diferentes módulos e sistemas.
- **Ferramentas:** JUnit para integração de componentes e Jenkins para automação de integração contínua.
- **Artefatos:** Scripts de integração e logs de execução.
- **Exemplo de Relatório:** Documenta problemas de compatibilidade entre módulos e logs de integração

## [Monitora](#)

.

### Teste de Unidade

- **Técnicas:** Verifica funcionalidades isoladas para garantir precisão.
- **Ferramentas:** JUnit para Java e NUnit para C#.
- **Artefatos:** Scripts de teste unitário e logs de execução.
- **Exemplo de Relatório:** Inclui a cobertura de código e taxas de falha para cada unidade testada

## [Guru99](#)

.

### Teste Exploratórios

- **Técnicas:** O teste não-estruturado onde o testador explora a aplicação para descobrir problemas.
- **Ferramentas:** TestRail para documentação dos casos explorados e gravação de erros.
- **Artefatos:** Documentação de casos de teste exploratórios e feedback imediato.
- **Exemplo de Relatório:** Inclui observações de usabilidade e falhas inesperadas

## [Monitora](#)

## [RoleCatcher | Your Competitive Edge](#)

.