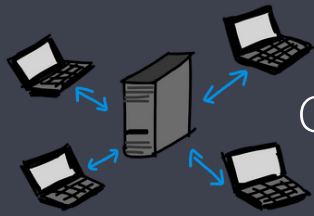


# Checklist de Performance de Testes: Guia Prático para Otimizar Seu Software

## ■ Introdução:

Testes de performance avaliam como seu software se comporta sob carga, garantindo estabilidade, velocidade e escalabilidade. Este checklist ajuda você a planejar, executar e analisar testes de forma eficiente. Siga os passos abaixo e marque cada item conforme avança. Ferramentas recomendadas: JMeter, LoadRunner ou Gatling.

Dica: Use este checklist em projetos de apps móveis, websites ou APIs. Se precisar de mais detalhes, confira nossos recursos premium abaixo.



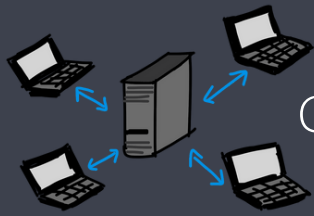
# Checklist de Performance de Testes: Guia Prático para Otimizar Seu Software

## 🧩 Planejamento e Preparação

Antes de começar, defina o escopo para evitar surpresas.

- [ ] Defina objetivos claros: O que medir? (Ex.: tempo de resposta < 2s, suporte a 1000 usuários simultâneos).
- [ ] Identifique cenários de teste: Liste casos reais, como login em pico de tráfego ou upload de arquivos grandes.
- [ ] Configure o ambiente: Use servidores similares ao produção (ex.: AWS, Azure). Inclua dados de teste realistas.
- [ ] Escolha ferramentas: Selecione software gratuito (JMeter) ou pago (LoadRunner). Instale e teste a configuração.
- [ ] Defina métricas: Tempo de resposta, throughput, uso de CPU/memória, taxa de erro.

;

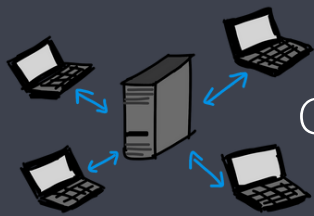


# Checklist de Performance de Testes: Guia Prático para Otimizar Seu Software

## ■ Execução dos Testes

Execute com precisão para obter dados confiáveis.

- [ ] **Teste de carga (Load Testing):** Simule usuários crescentes (ex.: 10, 100, 1000) e monitore gargalos.
- [ ] **Teste de estresse (Stress Testing):** Vá além dos limites (ex.: 2000 usuários) para ver quando o sistema quebra.
- [ ] **Teste de volume (Volume Testing):** Teste com grandes volumes de dados (ex.: banco com milhões de registros).
- [ ] **Teste de pico (Spike Testing):** Simule picos repentinos de tráfego e avalie recuperação.
- [ ] **Monitore em tempo real:** Use dashboards para CPU, memória e rede. Registre logs de erros.

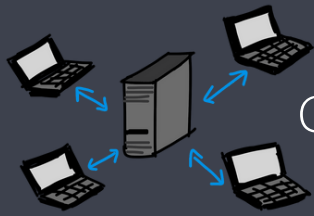


# Checklist de Performance de Testes: Guia Prático para Otimizar Seu Software

## ■ Análise e Relatórios

Avalie os resultados e proponha melhorias.

- [ ] **Analise métricas:** Compare com benchmarks. Identifique bottlenecks (ex.: queries lentas no banco).
- [ ] **Gere relatórios:** Crie gráficos de performance (ex.: latência vs. usuários). Use ferramentas como Grafana.
- [ ] **Identifique problemas:** Liste falhas (ex.: memory leaks) e priorize correções.
- [ ] **Teste correções:** Reexecute testes após ajustes e valide melhorias.
- [ ] **Documente lições aprendidas:** Anote o que funcionou e o que não, para futuros projetos.

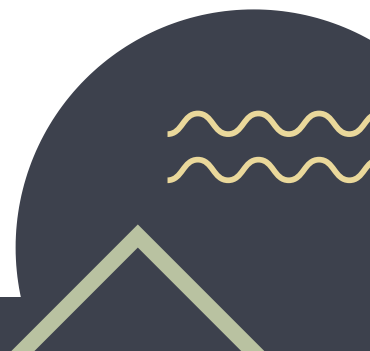


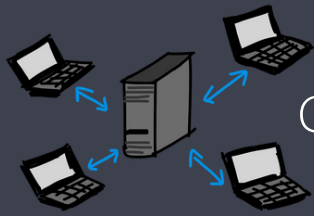
# Checklist de Performance de Testes: Guia Prático para Otimizar Seu Software

## Boas Práticas e Dicas Avançadas

Para resultados profissionais.

- [ ] **Automatize onde possível:** Use scripts para repetir testes (ex.: via CI/CD com Jenkins).
- [ ] **Considere fatores externos:** Teste em diferentes redes (3G, 5G) e dispositivos.
- [ ] **Inclua testes de segurança:** Combine com testes de carga para detectar vulnerabilidades.
- [ ] **Equipe e comunicação:** Trabalhe com devs e stakeholders. Compartilhe relatórios regularmente.
- [ ] **Atualize-se:** Siga blogs como o do Google Testing Blog ou cursos online.
- 





# Checklist de Performance de Testes: Guia Prático para Otimizar Seu Software

## **Conclusão**

Seguindo este checklist, você garante que seu software seja robusto e pronto para o mundo real. Marque todos os itens e veja a diferença na qualidade dos seus projetos!

### **Quer aprofundar em automação ?**

Confira nossos produtos digitais premium para levar seus testes ao próximo nível:

[Aulas de performance com Jmeter](#) < Gratuito

[50 Comandos em Cypress Prontos](#)

[50 Comandos de Playwright prontos](#)

[BDD como funciona e exemplos práticos](#)

[Guia para Robot Framework no VS Code](#)

[Curso completo - Automação de testes](#)

