



# Curso

# Qualidade de software



# Performance



# Testes de performance

- A eficiência da performance é uma parte essencial de fornecer uma boa experiência aos usuários;
- Também chamado de testes de **desempenho**;
- É um tipo de teste **Não funcional**;
- Normalmente executado para ajudar a identificar gargalos do sistema, estabelecer uma *baseline* para futuras análises/testes.

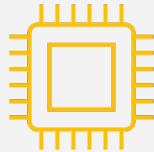


# Características



## Comportamento do tempo

Esse aspecto examina a capacidade de um componente ou sistema responder às entradas do usuário ou do sistema dentro de um tempo específico e sob condições específicas.



## Utilização de recursos

Avaliação da disponibilidade de recursos do sistema (ex., memória RAM) caso a utilização do recurso seja identificada como um risco.



## Capacidade

Avaliação do comportamento do sistema no limite da capacidade exigida (ex., carga de usuários).



# Experiência do usuário



# Principais tipos

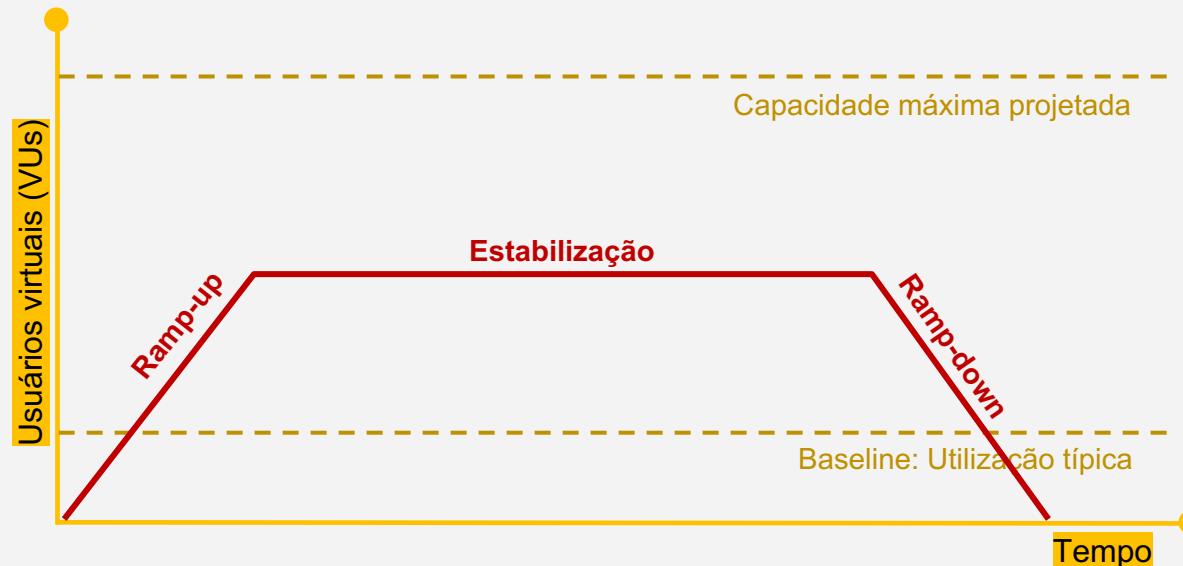
Teste de **performance** é um termo abrangente que inclui qualquer tipo de teste focado na performance (desempenho) do sistema ou componente sob diferentes volumes de carga:

- Teste de Carga (Load)
- Teste de Estresse (Stress)
- Testes de pico (Spike / Peak test)
- Testes de resistência (Endurance test)



# Testes de carga (Load test)

- O **teste de carga** concentra-se na capacidade de um sistema em lidar com níveis crescentes de cargas reais, de forma antecipada;
- As transações são geradas por quantidades de usuários virtuais, processos controlados ou concorrentes.

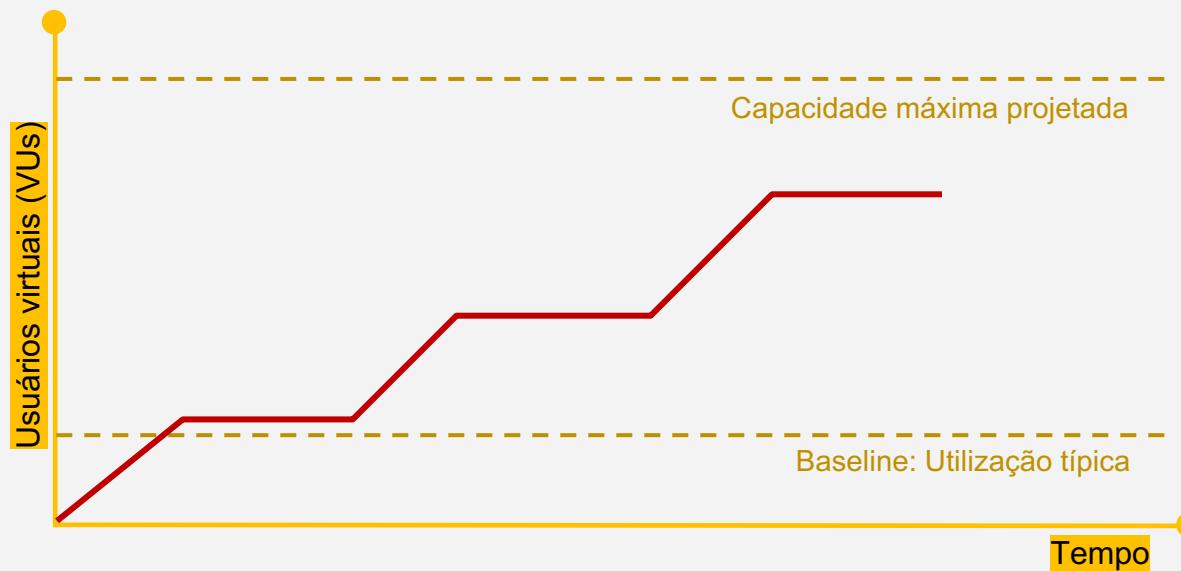


## Estratégia de teste:

- Número de VUs: 3000
- Ramp-up: 10 minutos
- Duração do teste: 1hr
- Ramp-down: 5 min

# Testes de estresse (Stress test)

- O **teste de estresse** concentra-se na capacidade de um sistema ou componente em lidar com picos de cargas que estão no limite ou além dos limites das cargas de trabalho previstas ou especificadas;
- Também usado para avaliar a capacidade de um sistema em lidar com a disponibilidade reduzida de recursos (processamento, rede, memória, etc.).



## Estratégia de teste:

- Número de VUs: 3000
- Ramp-up: De 5 em 5 minutos

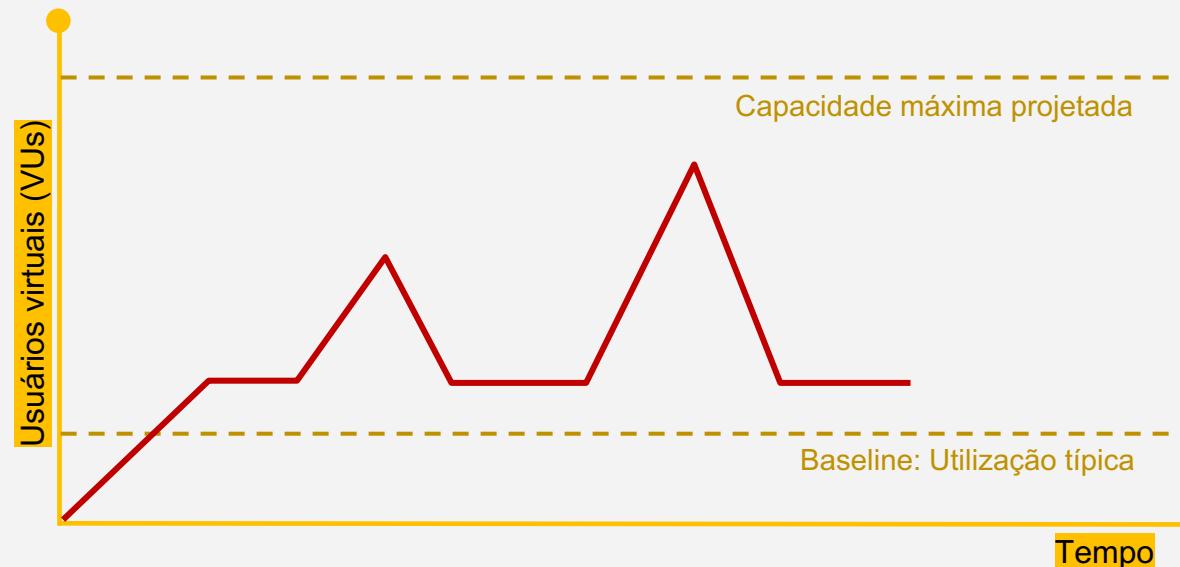


# Testes de pico

(Spike / Peak test)



- O **teste de pico** foca na capacidade de um sistema em responder corretamente a rajadas súbitas de cargas de pico e retornar depois a um estado estável.



# Testes de resistência

(Endurance test)



- O **teste de resistência** concentra-se na estabilidade do sistema ao longo de um período específico para o contexto operacional do sistema;
- Esse tipo de teste verifica se não há problemas de capacidade de recursos (vazamentos de memória, conexões de banco de dados, conjuntos de encadeamentos) que podem, eventualmente, degradar a performance e/ou causar falhas nos pontos de interrupção e segurança.



## Estratégia de teste:

- Número de VUs: 5000
- Ramp-up: 10 minutos
- Duração do teste: **5hs**
- Ramp-down: 10 min

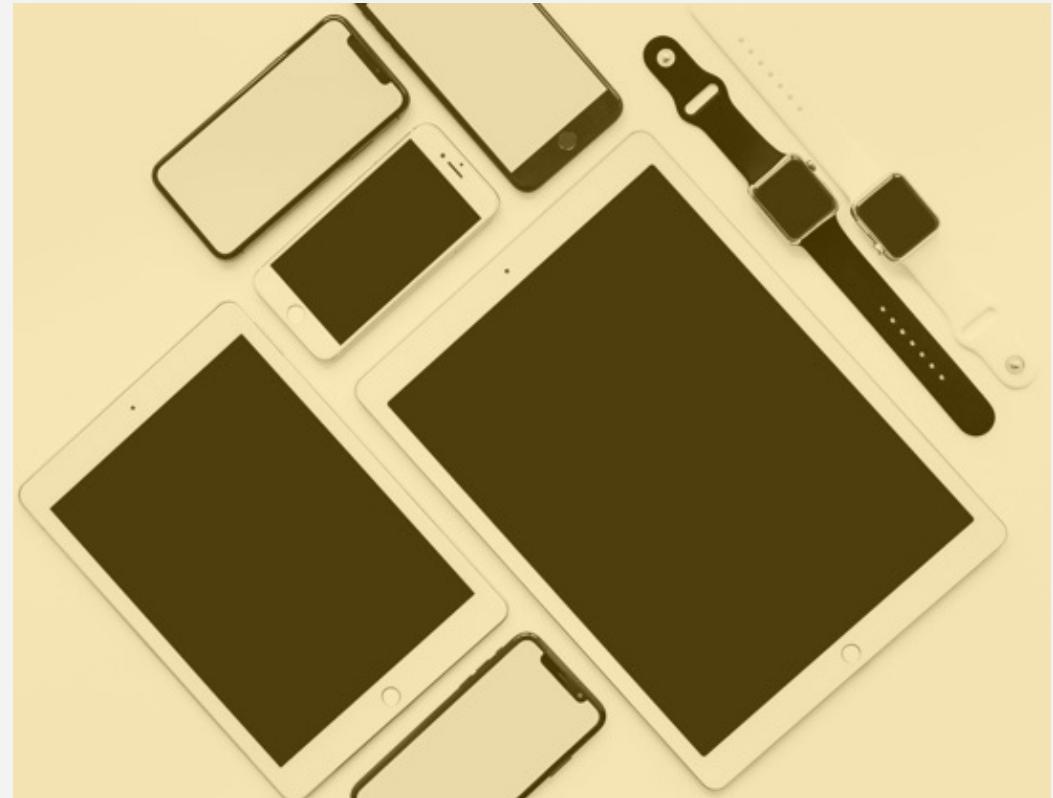


## Outros tipos

- **Teste de capacidade:** Determina quantos usuários e/ou transações um sistema suportará atendendo aos objetivos declarados de performance.
- **Teste de escalabilidade:** Determina a capacidade do sistema de expandir (Ex.: com mais usuários, grandes quantidades de dados armazenados) sem violar os requisitos de performance especificados no momento ou em sua falha.
- **Teste de concorrência:** Foca no impacto das situações em que ações específicas ocorrem simultaneamente (Ex.: quando muitos usuários fazem login ao mesmo tempo).

# O que testar?

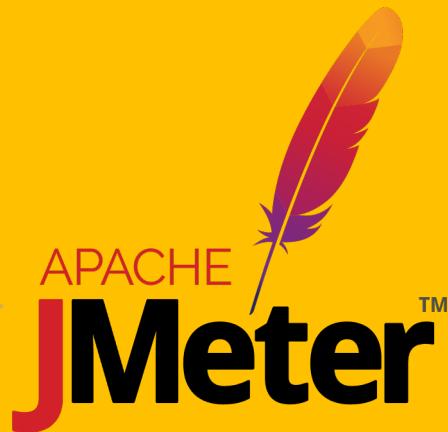
- Web
- Dispositivos móveis
- Internet das coisas (IoT)
- Dispositivos clientes de desktop
- Servidor
- Mainframe
- Bancos de Dados
- Redes
- Etc.





## Principais métricas

- **Tempo de resposta:** Ex.: por transação, por usuário simultâneo, tempos de carregamento da página.
- **Utilização de recursos:** Ex.: CPU, memória, largura de banda de rede, latência de rede, espaço em disco disponível, taxa de E/S, processamentos distribuídos ociosos e ocupados.
- **Taxa de transferência da transação-chave** (ou seja, o número de transações que podem ser processadas em um determinado período).
- **Tempo de processamento:** Ex.: tempos de espera, tempos de processamento, tempos de resposta da base de dados, tempos de conclusão.
- **Falhas:** Números de erros que afetam a performance.
- **Tempo de conclusão:** Ex.: para criar, ler, atualizar e excluir dados
- **Carregamento em segundo plano** em recursos compartilhados (especialmente em ambientes virtualizados).
- **Métricas de software:** Ex.: complexidade de código.



- O aplicativo **Apache JMeter™** é um software de código aberto (open source), um aplicativo Java projetado para testar o comportamento funcional de carga e medir o desempenho;
- Pode ser usado para testar o desempenho em recursos estáticos e dinâmicos;
- Pode ser usado para simular uma carga pesada em um servidor, grupo de servidores, rede ou objeto para testar sua força ou para analisar o desempenho geral sob diferentes tipos de carga.

Download: [https://jmeter.apache.org/download\\_jmeter.cgi](https://jmeter.apache.org/download_jmeter.cgi)



# Referências

- <https://blog.onedaytesting.com.br/testes-de-performance/>
- [https://bstqb.org.br/b9/doc/syllabus\\_ctfl\\_pt\\_2018br.pdf](https://bstqb.org.br/b9/doc/syllabus_ctfl_pt_2018br.pdf)
- <https://pricilahenry.medium.com/automa%C3%A7%C3%A3o-de-teste-de-performance-com-jmeter-f71064b2c056>
- <https://jmeter.apache.org/>
- <https://jmeter-plugins.org/wiki/PerfMon/>
- <https://github.com/undera/perfmon-agent/blob/master/README.md>
- Palestra de performance com Júlio de Lima, Fábio Araújo e Ernesto Barbosa (abril de 2020):  
[https://www.youtube.com/watch?v=QQj\\_pSS2Eeo](https://www.youtube.com/watch?v=QQj_pSS2Eeo)

