

1. Resumo

O propósito deste teste de performance da API REST ServeRest é validar que a API suporta um número X de usuários interagindo com diversas requisições por Y minutos, que serão descritos nos próximos pontos deste documento.

2. Metodologia

Os testes foram realizados utilizando a ferramenta JMeter da Apache, em um ambiente local disponibilizado pelo site da ServeRest.

- O ambiente foi montado localmente no caminho “http://localhost:3000” utilizando o seguinte comando disponibilizado pelo site da ServeRest em NodeJS:

```
Microsoft Windows [versão 10.0.22631.4317]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\Jhonatan Nascimento>npx serverest@latest

Serverest v2.29.5 está em execução
Teste o funcionamento acessando http://localhost:3000/usuarios no navegador

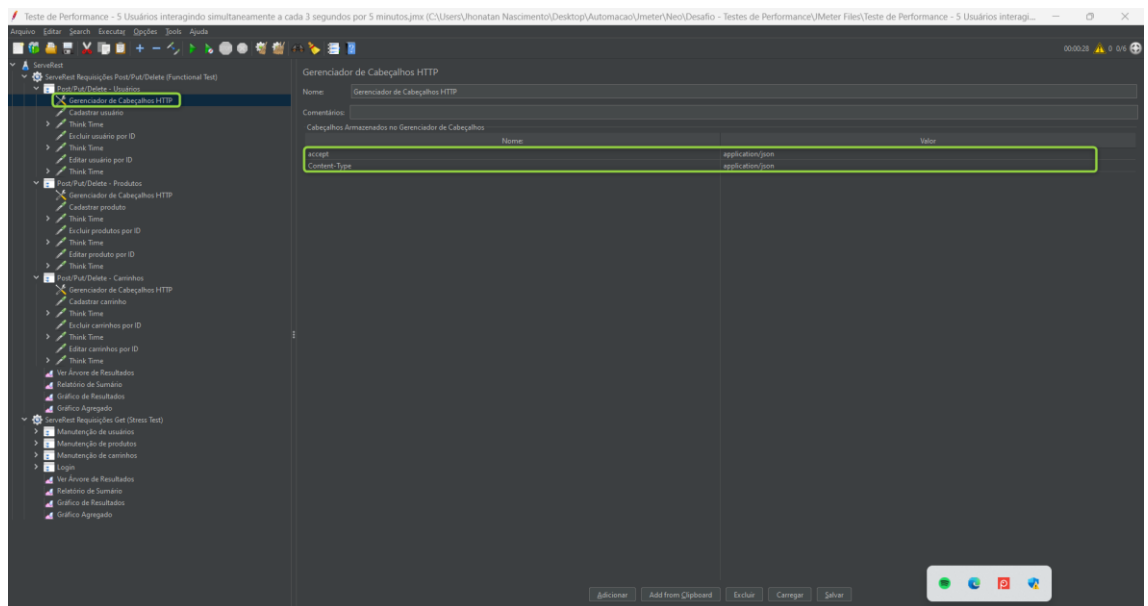
Quer saber as rotas disponíveis e como utilizá-las? Acesse http://localhost:3000
Quer alterar porta de execução, timeout do token, etc? Execute npx serverest --help
Para outras dúvidas acesse github.com/ServeRest/Serverest

Feito com ♥ para todos os QAs e apoiado por Wellington Avelino

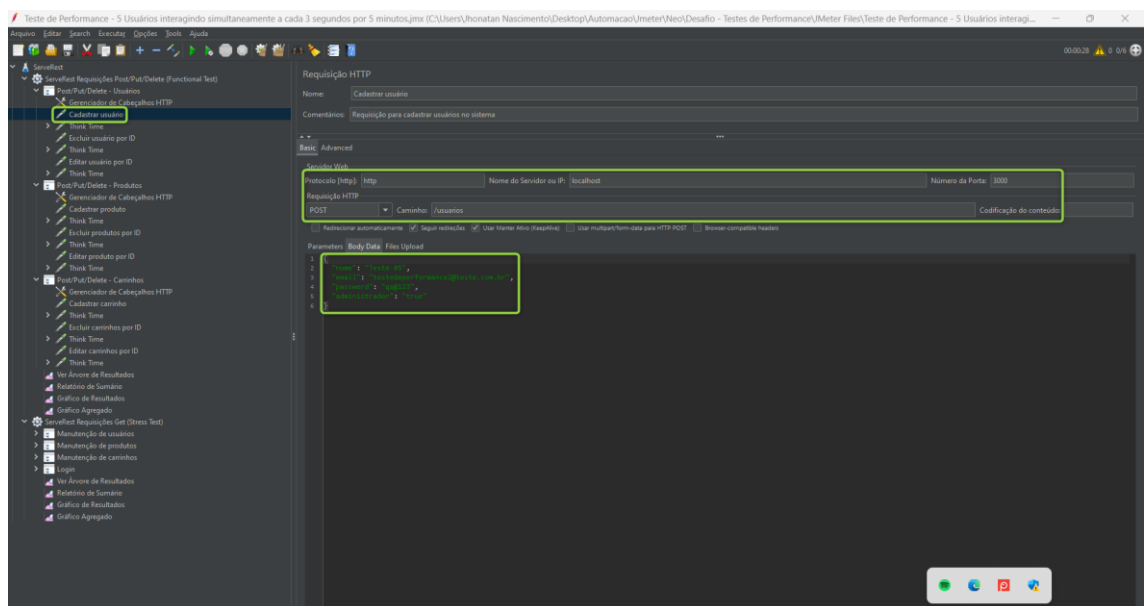
POST /login 200 17.466 ms - 265
POST http://localhost:3000/login 200 11.413 ms - 275
POST /usuarios 200 5.254 ms - 82
DELETE /usuarios/CNcGUGcf90bdFOEP 200 17.059 ms - 51
PUT /usuarios/lr9t0qJlm346861 200 4.688 ms - 50
POST /produtos 401 0.877 ms - 103
DELETE /produtos/y6AY8c1luREhQZEP 401 0.391 ms - 103
PUT /produtos/K6leHdfte03j8B1 401 0.249 ms - 103
POST /carrinhos 401 0.314 ms - 103
PUT /carrinhos/afDUqntef4ia0uMfg 405 0.379 ms - 165
GET /usuarios 200 0.800 ms - 1335
GET /usuarios 200 0.473 ms - 1335
GET /usuarios 200 0.866 ms - 1335
GET /usuarios 200 0.559 ms - 1335
GET /usuarios/0uxuPY0cbmQhpEz1 200 0.482 ms - 148
GET /usuarios 200 1.233 ms - 1335
GET /usuarios/0uxuPY0cbmQhpEz1 200 0.523 ms - 148
GET /usuarios/0uxuPY0cbmQhpEz1 200 1.134 ms - 148
GET /usuarios/0uxuPY0cbmQhpEz1 200 0.535 ms - 148
GET /usuarios 200 0.952 ms - 1335
GET /usuarios/0uxuPY0cbmQhpEz1 200 0.772 ms - 148
GET /usuarios 200 0.290 ms - 1335
GET /usuarios 200 0.505 ms - 1335
GET /usuarios 200 0.807 ms - 1335
GET /usuarios/0uxuPY0cbmQhpEz1 200 0.900 ms - 148
GET /usuarios 200 0.554 ms - 1335
GET /usuarios/0uxuPY0cbmQhpEz1 200 0.737 ms - 148
GET /usuarios/0uxuPY0cbmQhpEz1 200 0.521 ms - 148
GET /usuarios/0uxuPY0cbmQhpEz1 200 0.601 ms - 148
GET /usuarios 200 0.831 ms - 1335
GET /usuarios/0uxuPY0cbmQhpEz1 200 0.439 ms - 148
GET /usuarios 200 0.342 ms - 1335
GET /usuarios 200 0.489 ms - 1335
GET /usuarios 200 0.616 ms - 1335
GET /usuarios/0uxuPY0cbmQhpEz1 200 0.841 ms - 148
```

No JMeter, foi criado um plano de teste, ou TestPlan, que foi dividido em dois grupos de usuários. O primeiro grupo, nomeado “ServeRest Requisições Post/Put/Delete (Functional Test)” foi criado para abranger testes funcionais de requisições Post, Put e Delete. Para esse grupo foram criados os cenários de validação de criar, de atualizar e de deletar dados de requisições de usuários, de produtos e de carrinhos. Também foram adicionados dois relatórios e dois gráficos para consulta e validação dos resultados das requisições. Foi também utilizado somente um usuário e apenas uma interação para cada requisição, com o objetivo somente de validar se elas estão funcionando de maneira correta. E também, foi adicionado um tempo de espera de um segundo entre cada requisição (Think Time), para simular um usuário real interagindo com o sistema.

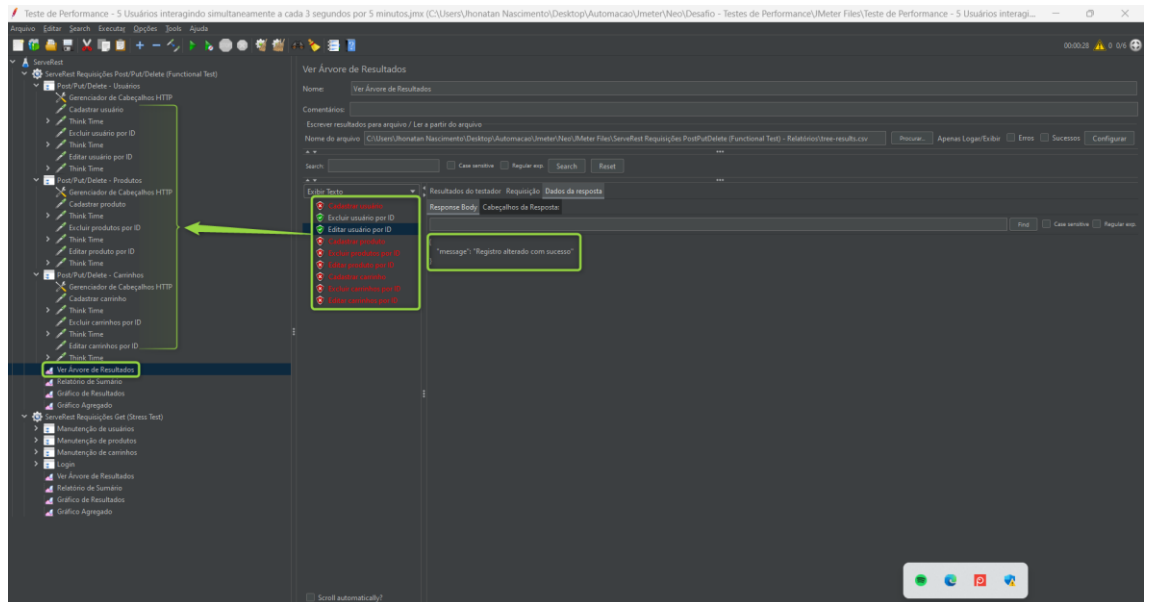
- Cabeçalho das requisições:



- Exemplo de uma requisição Post:

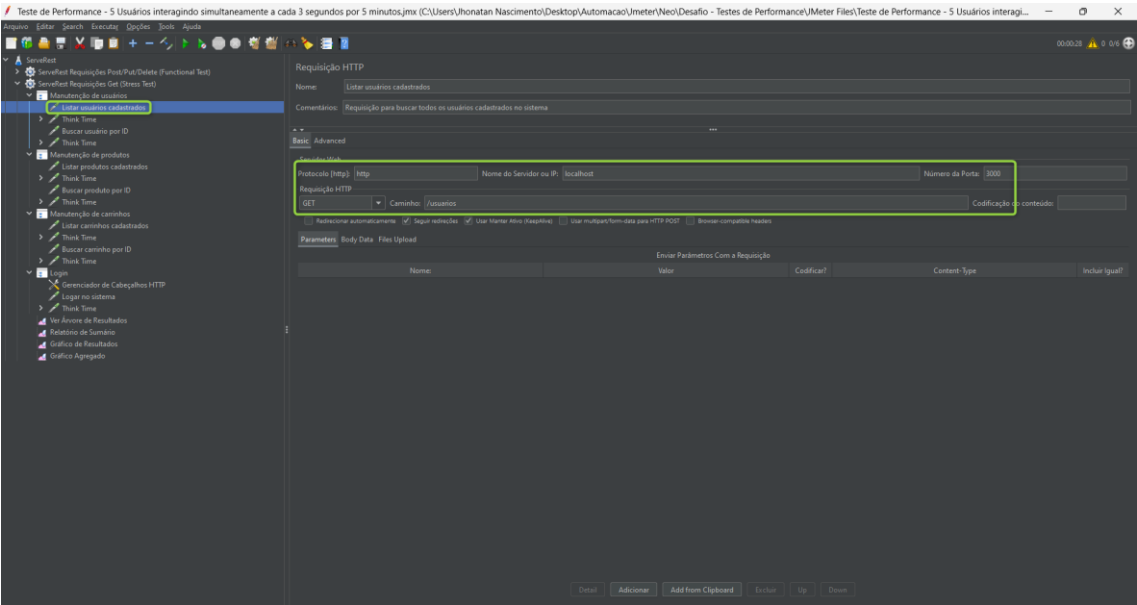


- Exemplo de um relatório (Obs: Os relatórios são salvos no caminho do campo “Nome do arquivo” em Excel):



Já o segundo grupo foi nomeado “ServeRest Requisições Get (Stress Test)” e foi criado para abranger testes de performance de requisições Get (e uma requisição Post de Login). Para esse grupo foram criados os cenários de validação de listar e buscar dados de usuários, de produtos e de carrinhos e um cenário de login. Também foram adicionados dois relatórios e dois gráficos para consulta e validação dos resultados das requisições. Foi também utilizado cinco usuário e interações infinitas em um tempo de 5 minutos (300 segundos), com o objetivo de validar se o sistema suporta diversas interações de 5 usuários em 5 minutos. E também, foi adicionado um tempo de espera de um segundo entre cada requisição (Think Time), para simular um usuário real interagindo com o sistema.

- Exemplo de requisição:



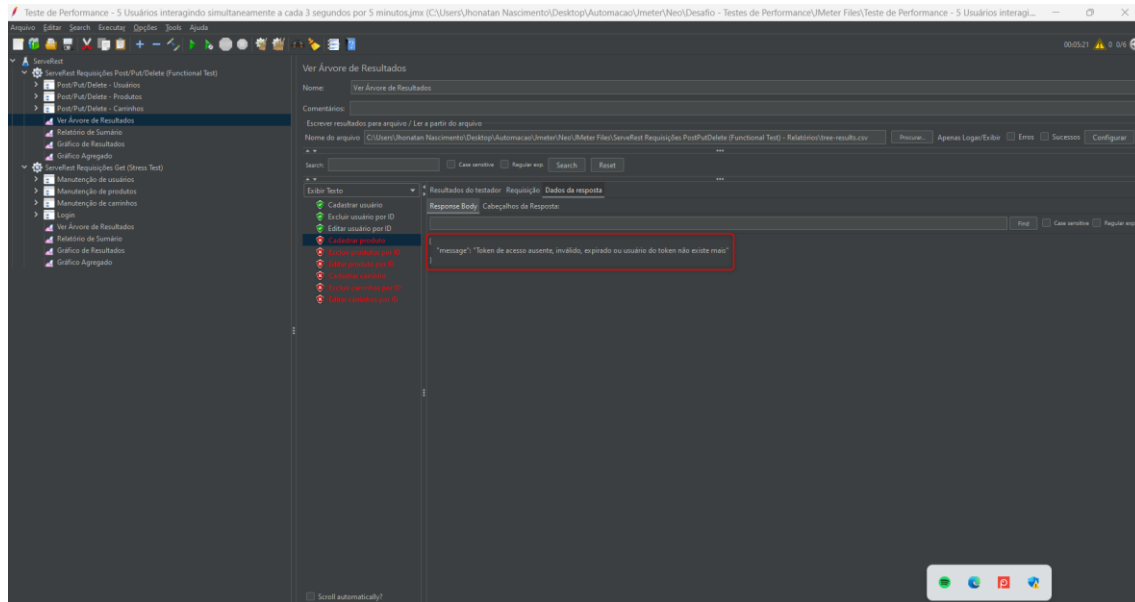
- Exemplo de relatório de performance:

The screenshot shows the JMeter GUI for the 'Relatório de Sumário' (Summary Report) for the 'Listar usuários cadastrados' test. The table displays performance metrics for various test elements:

| Rótulo | # Amostras | Média | Mín. | Máx. | Desvio Padrão | % de Erro | Veloc. | KB/s | Send KB/sec | Média de Bytes |
|--------------------------------|------------|-------|------|------|---------------|-----------|---------|------|-------------|----------------|
| Listar usuários cadastrados... | 9 | 4 | 3 | 8 | 1,49 | 0,00% | 1,3/sec | 2,40 | 0,19 | 1720,9 |
| Buscar usuário por ID | 5 | 3 | 3 | 4 | 0,40 | 0,00% | 2,0/sec | 1,13 | 0,29 | 567,6 |
| TOTAL | 14 | 4 | 3 | 8 | 1,41 | 0,00% | 2,4/sec | 3,57 | 0,31 | 1388,7 |

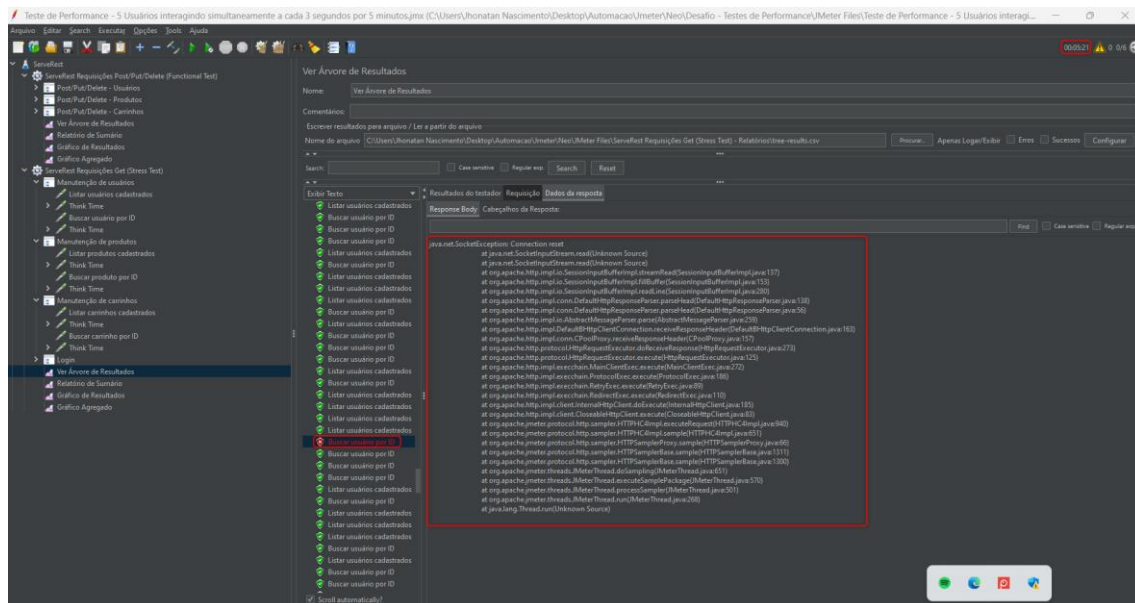
3. Análise dos resultados de teste:

Analisando os resultados do teste, foi verificado que no primeiro grupo de usuário (ServeRest Requisições Post/Put/Delete (Functional Test)) algumas requisições retornaram uma mensagem de problema no token, mostrando que o token de acesso não foi preenchido:

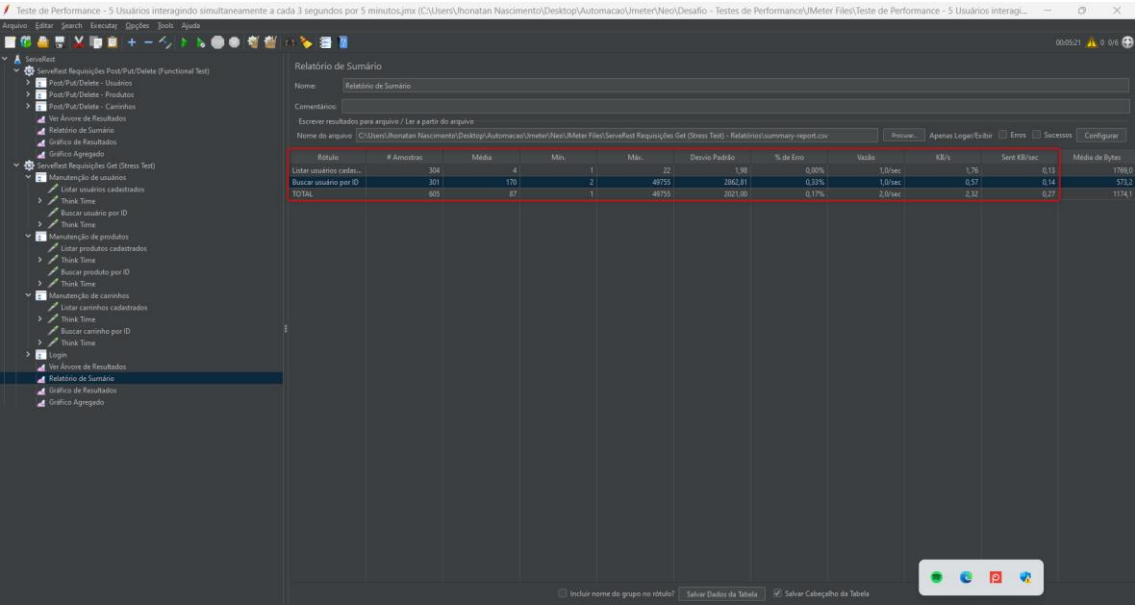


Já no segundo grupo de usuário foi verificada a seguinte análise:

Quando 5 usuários estavam executando requisições Get simultaneamente por 5 minutos com 1 segundo de intervalo entre as requisições, em um determinado momento do teste, a aplicação falhou ao executar uma requisição conforme a imagem abaixo:



As seguintes métricas foram analisadas na performance:



The screenshot shows the JMeter Summary Report window. The left sidebar lists various test elements, including 'Relatório de Sumário' which is currently selected. The main area displays a table of performance metrics for the selected test. The table has columns for Test Name, Number of Samples, Mean, Minimum, Maximum, Deviation Standard, % of Errors, Value, KB/s, Sent KB/sec, and Mean of Bytes. The data is organized into a hierarchical structure with sub-headers for different test phases.

| Teste | # Amostras | Média | Mín. | Máx. | Desvio Padrão | % de Erro | Valor | KB/s | Sent KB/sec | Média de Bytes |
|-----------------------------|------------|-------|------|-------|---------------|-----------|---------|------|-------------|----------------|
| Testar usuários cadastrados | 305 | 4 | 1 | 22 | 1.98 | 0.00% | 1.01sec | 6.76 | 6.10 | 1768.0 |
| Listar usuários cadastrados | 301 | 179 | 2 | 48755 | 2862.81 | 0.33% | 1.01sec | 6.57 | 6.14 | 571.2 |
| TOTAL | 605 | 87 | 1 | 48755 | 2821.80 | 0.17% | 2.01sec | 2.32 | 6.27 | 1134.5 |

Se pegarmos a métrica “% de Erro”, veremos que a porcentagem de erro da aplicação é pequena com base na quantidade de requisições que foram realizadas pelos usuários no tempo estimado (foram realizadas 605 requisições), porém um ponto a favor disso é a base de dados também ser pequena, diminuindo a chance de erros nas requisições.

Também podemos analisar utilizando a métrica “Média de Bytes” que a requisição de listar usuários traz mais dados do que a requisição de buscar usuários por ID, devido a média de Bytes de cada uma.