ServeRest

# Escopo de Teste

Analisar e apontar a performance da API durante a listagem de usuários cadastrados utilizando 5 usuários simultâneos durante 5 minutos.

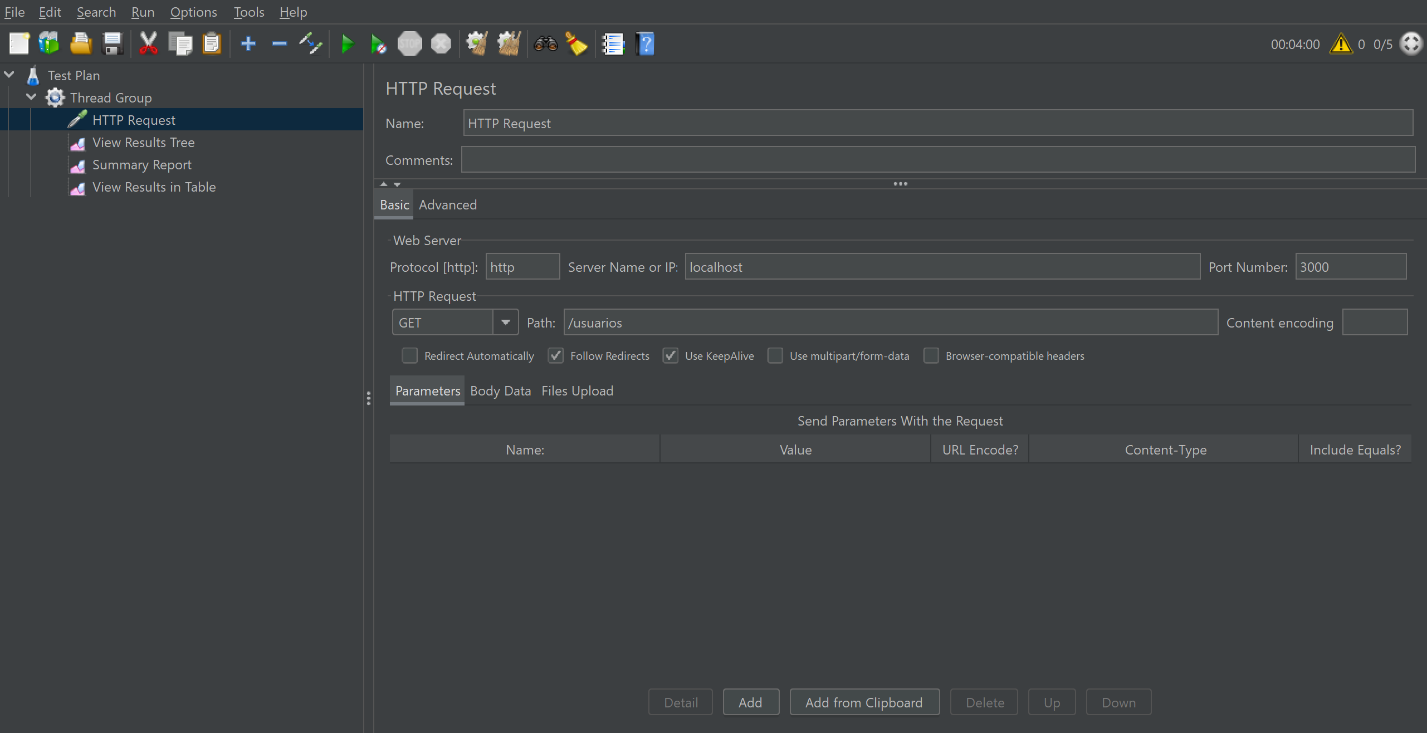
# Descrição Técnica da Execução dos Casos de Teste

Para a execução dos Testes de Performance foi utilizada a ferramenta JMeter e os testes foram executados localmente utilizando Docker.

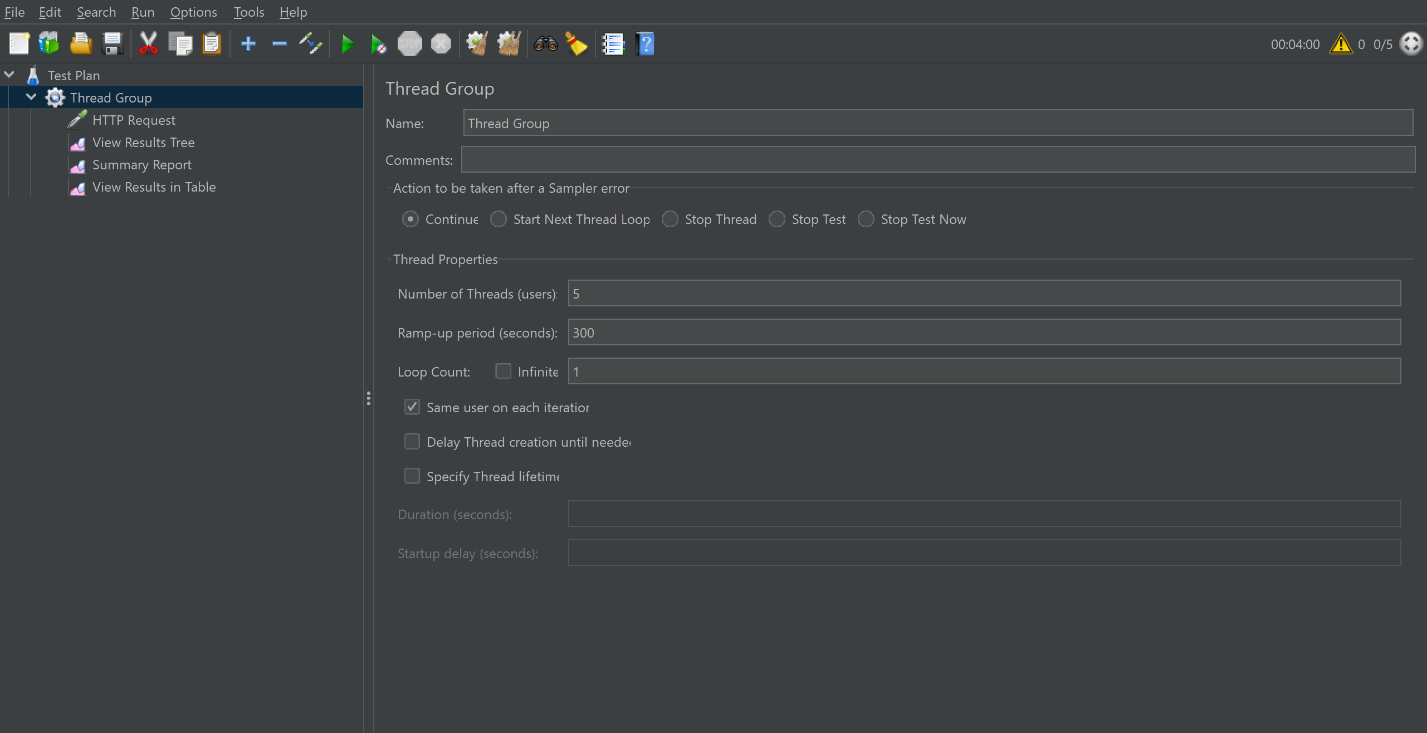
Conforme a descrição do escopo, foi feito apenas teste de resistência (tipo de teste de performance) com o objetivo de observar a estabilidade do site/API ao receber simultaneamente 5 requisições durante um período de 5 minutos e assim, verificar se não há problemas de capacidade de recurso, como: vazamentos de memória e conexões com a base de dados.

**Parâmetros de execução:**

* Requisição HTTP do tipo GET e Path “/usuarios” (caminho para visualizar a listagem de usuários cadastrados).



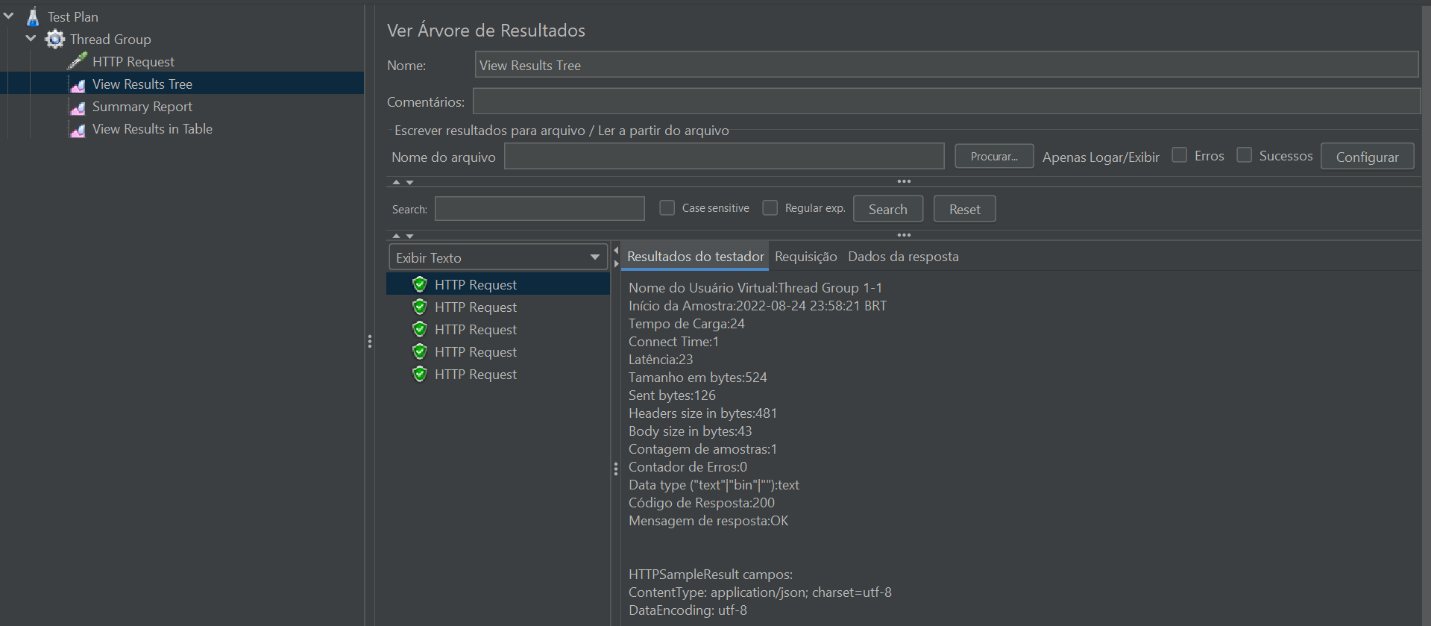
* Especificação do grupo de thread, onde foi solicitado uma quantidade de 5 threads (usuários virtuais) para realizar apenas 1 requisição e durante um período de 300 segundos que equivale a 5 minutos.



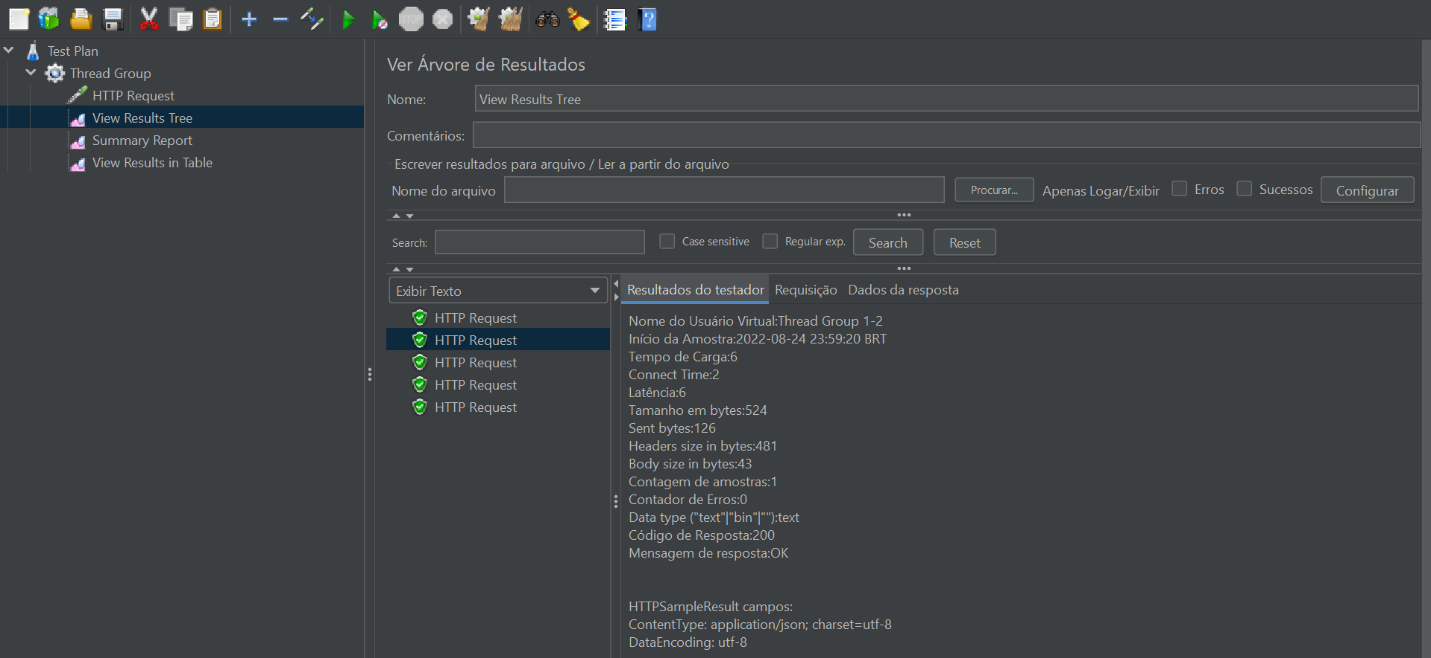
Grupo de Thread

**Resultado obtido ao executar o scrip de teste:**

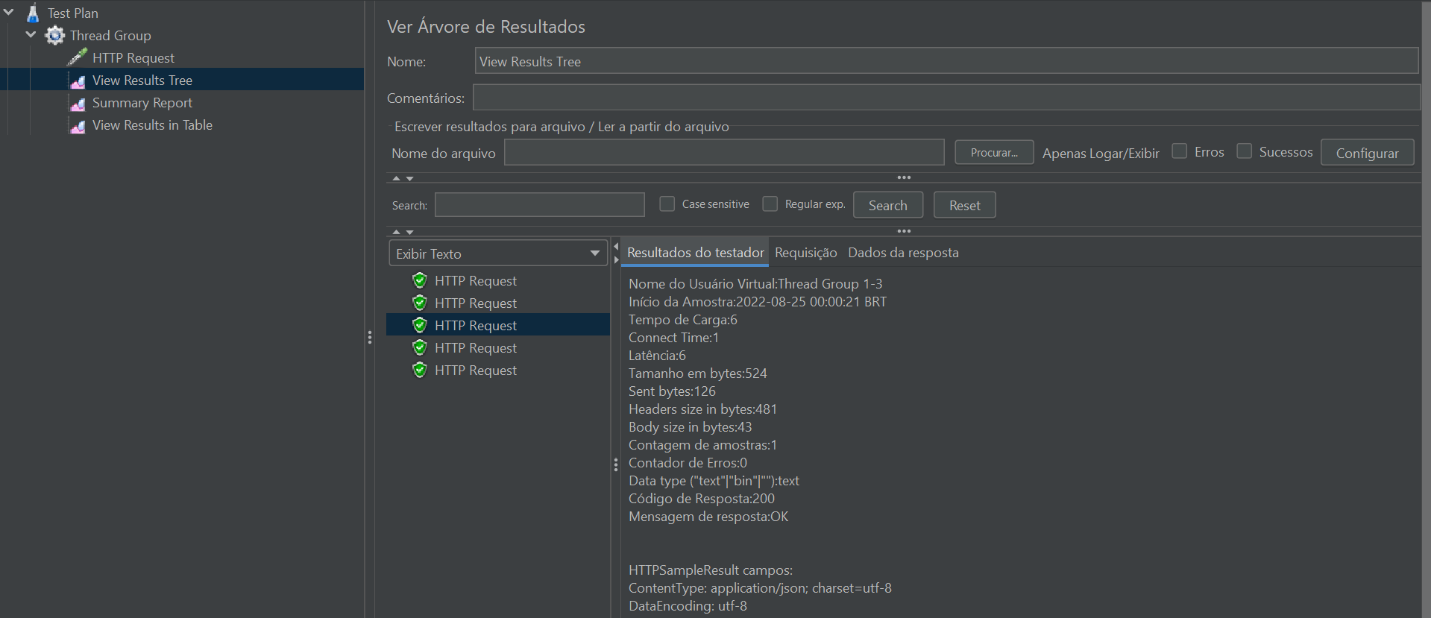
Por meio das imagens abaixo da árvore de resultados, é possível verificar as 5 requisições e para cada requisição é exibida como principais informações: o tempo de carga, tempo para realizar a conexão, latência, tamanho em bytes que a requisição consumiu, tamanho em byte enviado, tamanho em byte do cabeçalho e do corpo, contagem de erros e código da resposta.



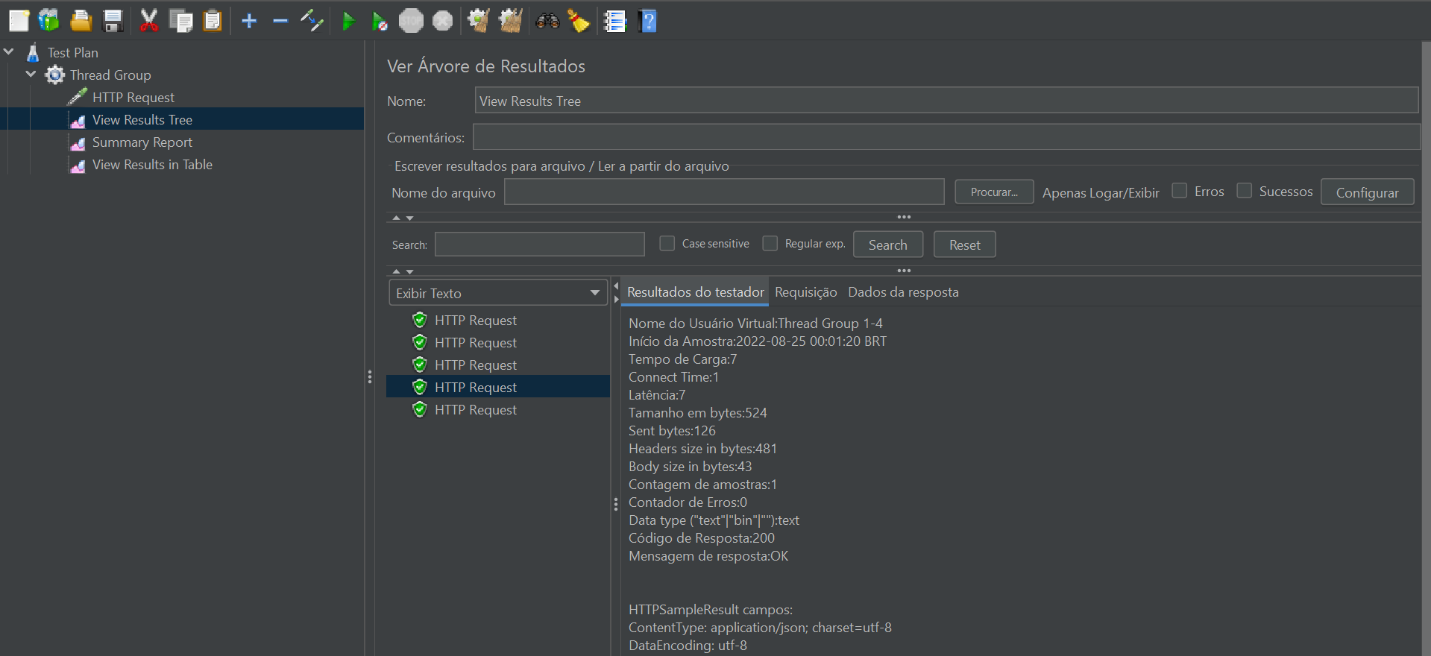
Árvore de resultados da Thread Group 1-1



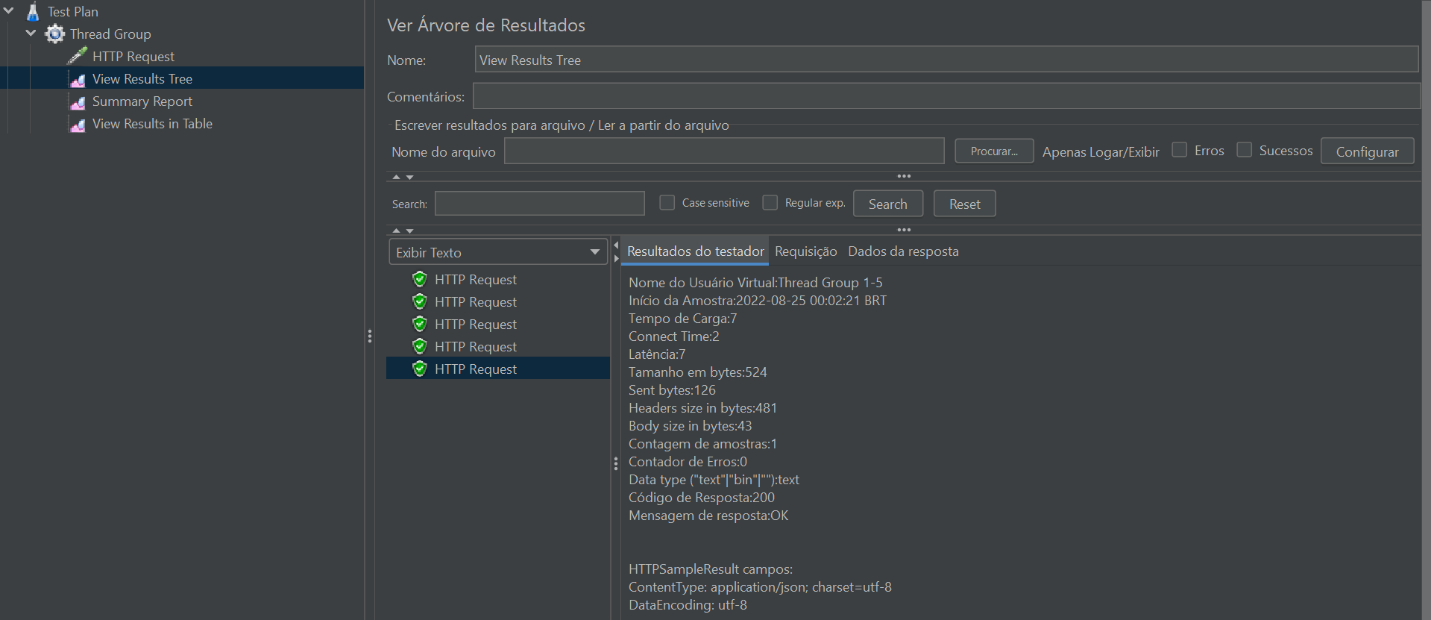
Árvore de resultados da Thread Group 1-2



Árvore de resultados da Thread Group 1-3

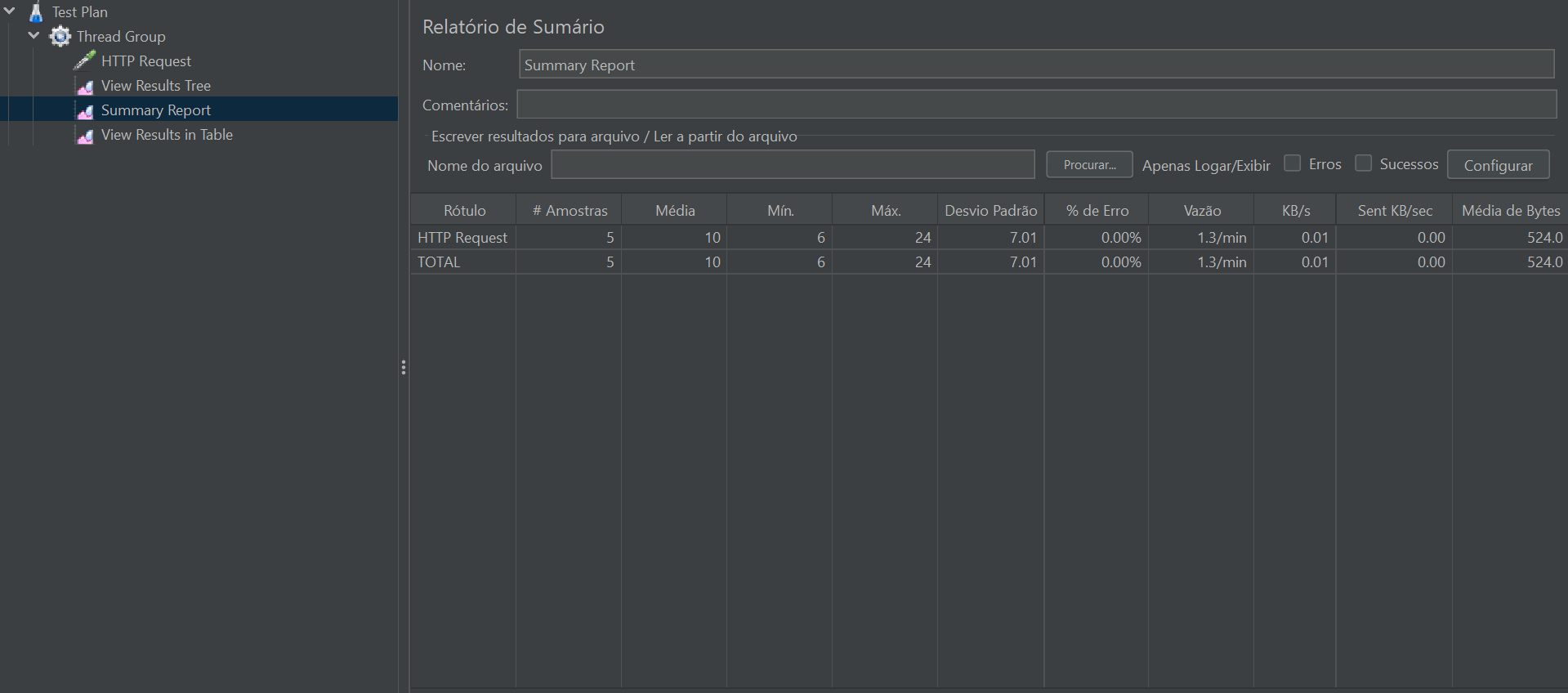


Árvore de resultados da Thread Group 1-4



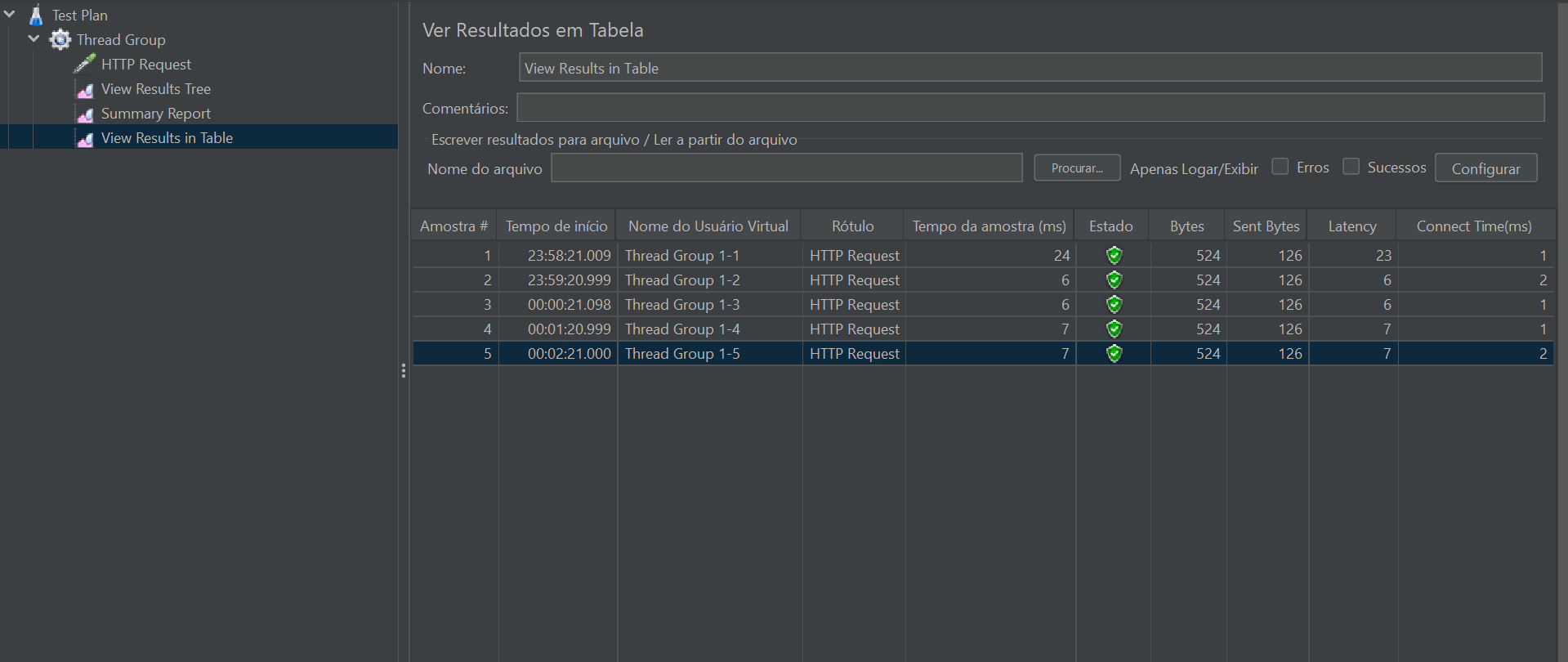
Árvore de resultados da Thread Group 1-5

O relatório de sumário abaixo diferencia-se da árvore de resultado, pois ele sintetiza os resultados das amostras, exibindo o tempo médio, máximo e mínimo das amostras em milissegundos, o desvio padrão, a taxa de erro, a vazão por minuto, KB por minutos consumido, KB por segundos que foram enviados e a média de consumo em Bytes.



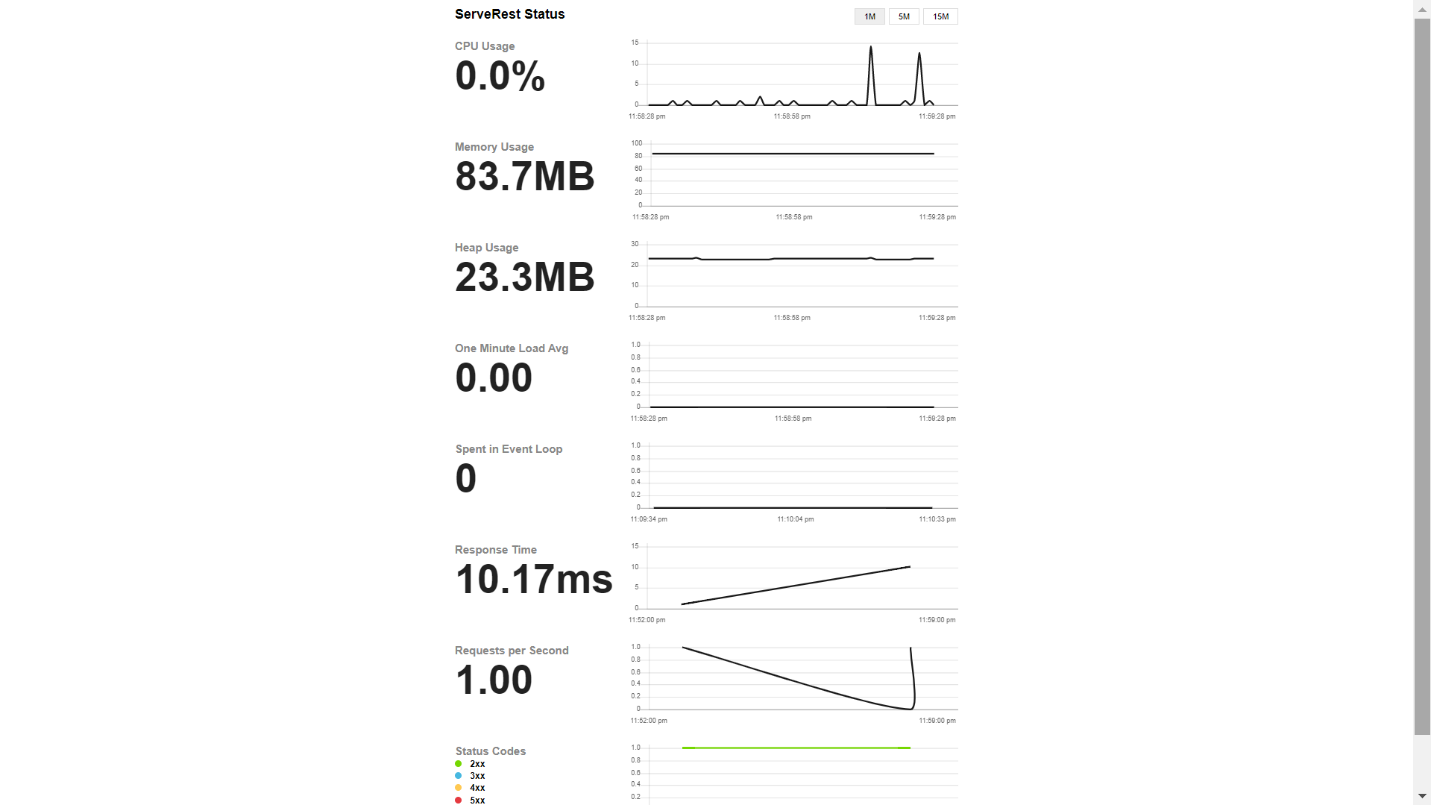
Relatório de Sumário

É exibido a seguir, o resultado em tabela que mostra as mesmas informações ditas anteriormente, entretando apresentandas em tabela.



Resultados em Tabela

Durante a execução do scrip de teste, foi possível observar outras variáveis disponibilizadas pela própria API por meio a URL <http://localhost:3000/status>. Logo abaixo vemos uma imagem que apresenta a porcentagem de utilização da CPU, a quantidade de memória utilizada em MB, a quantidade de memória alocada utilizada em MB, tempo de resposta e etc.



Status da API

# Conclusões

É certo que o objetivo do teste de performance é coletar métricas sobre o comportamento da aplicação sob determinada carga de usuários durante um período predefinido. No caso desse teste, o site/API foi submetido a uma carga de 5 usuários simultâneos durante um período de 5 minutos (300 segundos) e com isso podemos apontar alguns dos principais baseadas em métricas coletada:

* Tempo de resposta: A ferramenta JMeter por meio do Relatório de Sumário apresenta três medidas sobre o tempo de resposta, que são elas: a média, o mínimo e máximo tempo de resposta das amostras. É possível observar pela imagem dos Resultados em Tabela que a primeira amostra teve um valor para o tempo de resposta altamente divergente em relação as outras quatro amostras. Sendo assim, apontar o tempo médio de resposta como um indicador de performance para o tempo de resposta, não é uma boa escolha. Por isso, o tempo de resposta será indicado através da análise do 80 Percentil dos tempos de resposta que equivale a 7 milissegundos, ou seja, 7 milissegundos é o tempo de resposta do site/API e esse é indicador muito satisfatório e positivo.
* Taxa de erro: Esse indicador mostra a porcentagem de erros nas requisições em relação ao total de requisições e é comum que erros ocorram, mas para o teste realizado, foi obtido uma taxa de erro zero, conforme mostra a imagem de Relatório de Sumário. Contudo, esse é mais um indicador positivo para a performance.
* Latência: Apesar de ser um indicador de performance, esse é um indicador que pode sofrer diversas influências, como: distância entre a requisição e o destino, programas e protocolos no meio do caminho, tipo de conexão etc., por isso esse indicador não será levando em consideração.
* Bytes enviados x Bytes recebidos: O teste nos mostra essa métrica, que por sua vez, também é vista como positiva, pois é para esse caso é normal que o a quantidade de byte enviada seja menor que a quantidade recebida, visto que foi envio uma requisição a API e ela retornou um conjunto de dados dos usuários cadastrados.

Os status da API mostram que a porcentagem de processamento é na maioria do tempo 0, a memória utilizada é relativamente mínima e a taxa de erro é zero. Posto isso, por meio dos status da API e dos indicadores citados acima, é possível concluir que as requisições de 5 usuários virtuais simultâneos durante um período de 5 minutos não causaram nenhum impacto negativo para a API, ela se manteve estável e performática.

# Apêndices

O script de teste encontra-se disponível em <https://github.com/anaizaalencar/Teste-de-Performace-ServeRest>. Para realizar o teste de performance, é preciso ter instalado o JMeter, o Docker e executar o seguinte comando no terminal: docker run -p 3000:3000 paulogoncalvesbh/serverest:latest e por fim, realizar o download do projeto. Com o JMeter instalado, a ferramenta deve ser executada. Ao abrir a ferramenta, é preciso seguir os seguintes passos para abrir e executar o script:

1. Clicar em Arquivo;
2. Clicar em Abrir;
3. Selecionar o arquivo que contém o projeto baixado;
4. Clicar em Abrir;
5. Clicar em Iniciar.

Após iniciar, é preciso aguardar um período de 5 minutos até que o teste finalize e assim, conseguir observar as variáveis de resposta.

Para mais informação sobre a API ServeRest consultar o site <https://serverest.dev/>.