

Relatório - Trabalho 01

Implementação de um servidor Web Multithreaded

Anaísa Forti da Fonseca

11811ECP012

1. Objetivo

O objetivo deste trabalho é a implementação de um servidor Web "multithreaded", que deverá ser capaz de processar múltiplas requisições simultâneas de serviços em paralelo.

2. Introdução

O servidor Web é um dos componentes da aplicação Web e baseia-se em um programa servidor. Ao ser executado, o processo servidor é criado e ele mantém se "na escuta" para clientes Web que desejam extrair documentos por meio do protocolo HTTP.

Através de conexões TCP, esse servidor recebe pedidos (requests) e envia respostas (responses) até que o cliente decida finalizar a conexão.

3. Desenvolvimento

O código apresentado foi desenvolvido baseando-se no texto "Tarefa de Programação 1: Construindo um servidor Web multithreaded", disponível na sub-pasta "Primeiro Trabalho" da pasta "Trabalhos" do canal Geral da equipe Redes de Comunicações I no Microsoft Teams.

A versão 1.0 do HTTP foi implementada, definida na RFC-1945, na qual requisições HTTP separadas são enviadas a cada componente da página Web. Na thread principal, o servidor "escuta" uma porta fixa e, quando recebe uma requisição de conexão TCP, ele ajusta a conexão TCP através de outra porta e atende essa requisição em uma thread separada.

O código desenvolvido possui nome de "WebServer.java" e pode ser visto na pasta "multithreaded" enviada juntamente a este relatório.

4. Testes

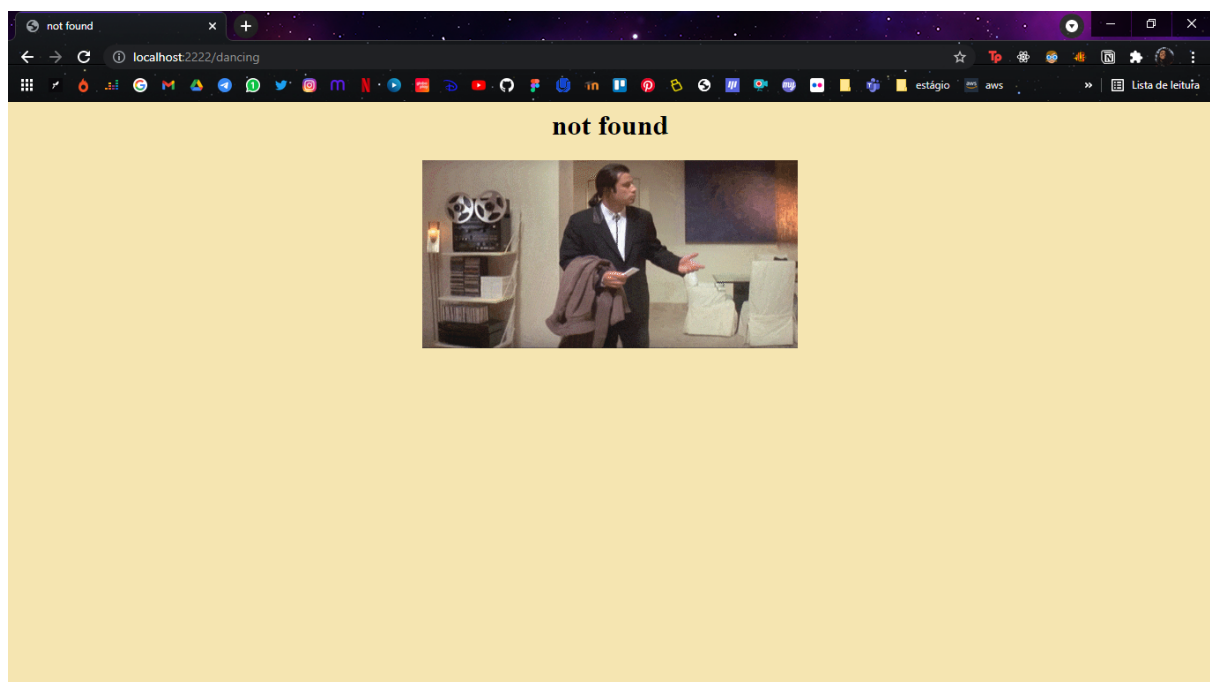
A fim de testar este servidor, foram utilizados quatro arquivos principais: "index.html", "dancing.gif", "relatorio.pdf" e "milkshake.jpeg". Todos eles estão disponíveis na pasta

"multithreaded" mencionada anteriormente.

Antes de explorar cada um dos testes, é válido ressaltar que nosso servidor foi definido para ser executado na porta 2222 do localhost. Então, caso o cliente não especifique a porta, não será possível conectar-se ao servidor. Também, é necessário que o arquivo desejado também seja especificado na URL. Portanto, a única maneira de acessar os quatro arquivos mencionados (html, gif, pdf e jpeg) é através destas quatro URLs:

```
1- http://localhost:2222/index.html
2- http://localhost:2222/dancing.gif
3- http://localhost:2222/relatorio.pdf
4- http://localhost:2222/milkshake.jpeg
```

Se o cliente tentar inserir qualquer outro link, seja sem especificar o arquivo ou o tipo do arquivo (como colocar apenas "dancing" ao invés de "dancing.gif"), foi definido que o servidor envie uma mensagem de resposta com a fonte HTML exibindo o erro na janela do browser:



A seguir, serão explorados os quatro testes feitos, considerando que todas URLs tenham sido inseridas corretamente, como comentado anteriormente.

4.1 index.html

Ao inserirmos a primeira URL para teste, obtemos como resposta do servidor a seguinte página HTML no browser:



Já no terminal, podemos confirmar que tudo correu bem. As seguintes informações são printadas:

```
[Running] cd "c:\Users\anaís\Documents\ufu\redes1\trabalho 1\multithread\" && javac WebServer.java && java WebServer

GET /index.html HTTP/1.1
Host: localhost:2222
Connection: keep-alive
Cache-Control: max-age=0
sec-ch-ua: "Google Chrome";v="93", " Not;A Brand";v="99", "Chromium";v="93"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/93.0.4577.63 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: pt-BR,pt;q=0.9,en;q=0.8
/index.html
HTTP/1.0 200 OK

Content-type: text/html

./index.html

GET /assets/index.gif HTTP/1.1
Host: localhost:2222
Connection: keep-alive
sec-ch-ua: "Google Chrome";v="93", " Not;A Brand";v="99", "Chromium";v="93"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/93.0.4577.63 Safari/537.36
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: image
Referer: http://localhost:2222/index.html
Accept-Encoding: gzip, deflate, br
Accept-Language: pt-BR,pt;q=0.9,en;q=0.8
/assets/index.gif
HTTP/1.0 200 OK

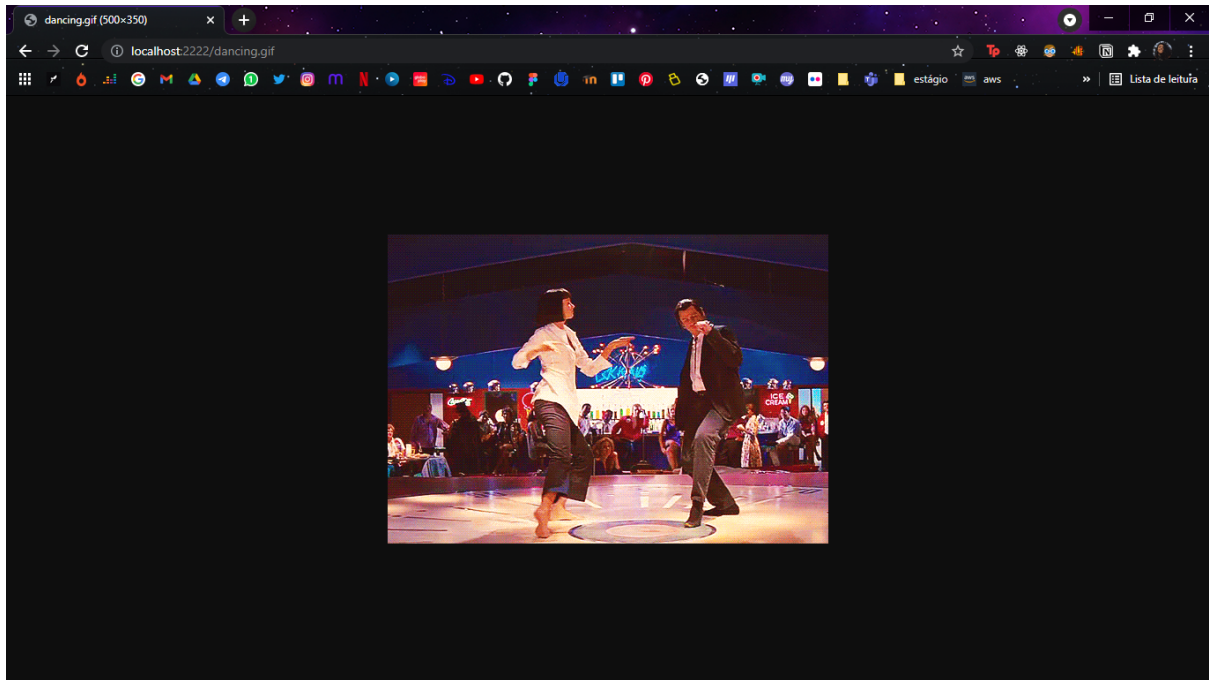
Content-type: image/gif

./assets/index.gif
```

O segundo "bloco" de informações é mostrado pelo fato de o browser carregar o arquivo "index.html".

4.2 dancing.gif

Ao acessar a segunda URL para teste, o servidor nos responde o gif "dancing.gif" (presente na pasta multithread enviada):



Novamente, as informações mostradas no terminal nos confirmam que tudo correu como esperado:

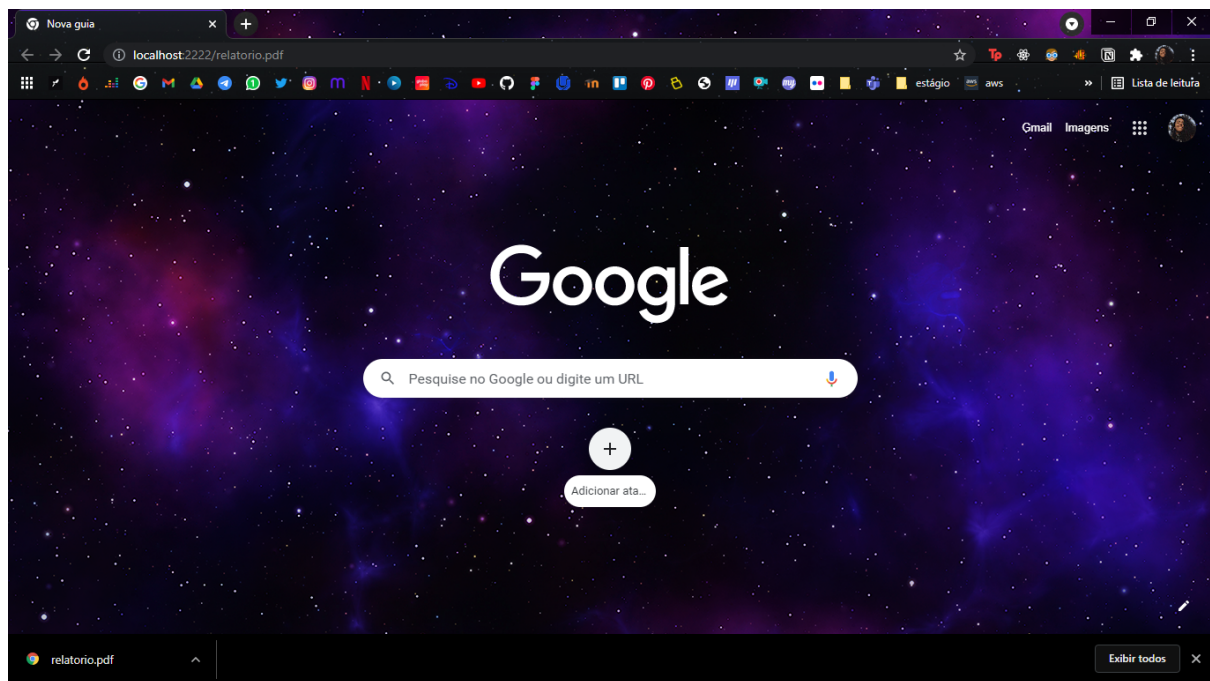
```
[Running] cd "c:\Users\anais\Documents\ufu\redes1\trabalho 1\multithread\" && javac WebServer.java && java WebServer

GET /dancing.gif HTTP/1.1
Host: localhost:2222
Connection: keep-alive
Cache-Control: max-age=0
sec-ch-ua: "Google Chrome";v="93", " Not;A Brand";v="99", "Chromium";v="93"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/93.0.4577.63 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: pt-BR,pt;q=0.9,en;q=0.8
/dancing.gif
HTTP/1.0 200 OK

Content-type: image/gif
./dancing.gif
```

4.3 relatorio.pdf

Neste teste, por se tratar de um pdf, o arquivo é baixado ao entrar na terceira URL. Observa-se que o arquivo foi baixado na parte inferior esquerda do navegador:



Já no terminal, confirma-se, novamente, que tudo se saiu bem:

```
[Running] cd "c:\Users\anaís\Documents\ufu\redes1\trabalho 1\multithread\" && javac WebServer.java && java WebServer

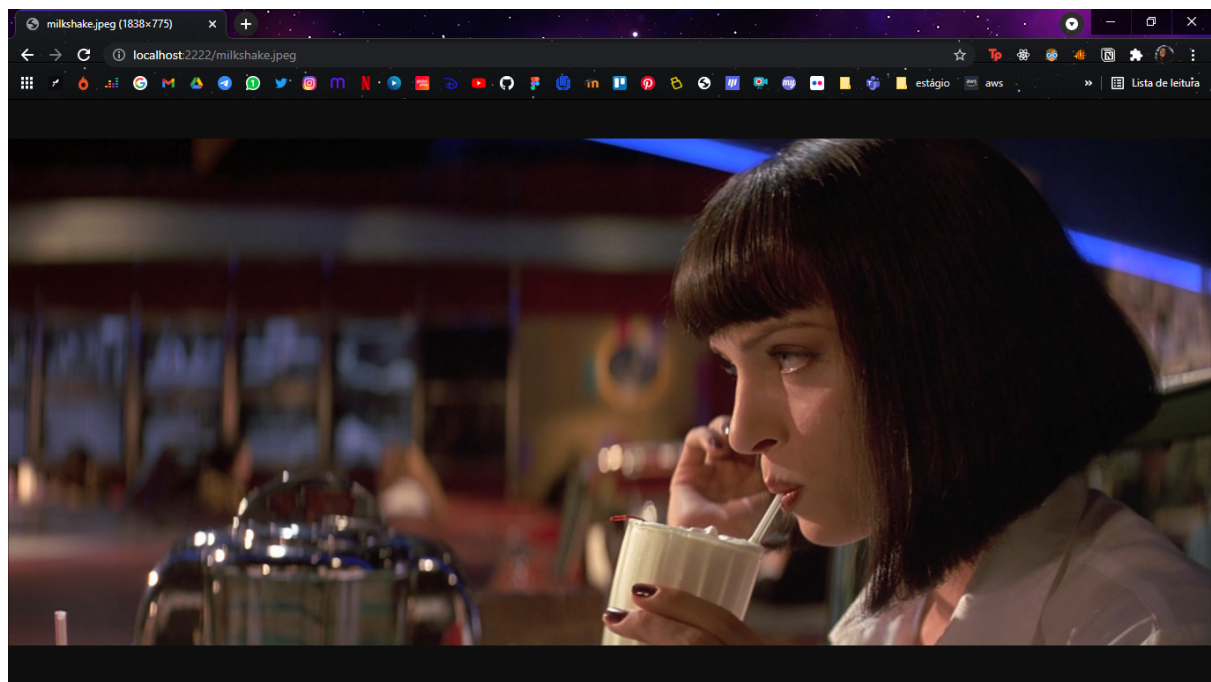
GET /relatorio.pdf HTTP/1.1
Host: localhost:2222
Connection: keep-alive
sec-ch-ua: "Google Chrome";v="93", " Not;A Brand";v="99", "Chromium";v="93"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/93.0.4577.63 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: pt-BR,pt;q=0.9,en;q=0.8
/relatorio.pdf
HTTP/1.0 200 OK

Content-type: application.pdf/jpeg

./relatorio.pdf
```

4.4 milkshake.jpeg

Na quarta (e última) URL, o navegador nos responde a imagem "milkshake.jpeg" (presente na pasta multithreaded enviada):



Aqui, também foi possível conferir que o processo se saiu como esperado através do terminal:

```
[Running] cd "c:\Users\anaís\Documents\ufu\redesi\trabalho 1\multithread\" && javac WebServer.java && java WebServer

GET /milkshake.jpeg HTTP/1.1
Host: localhost:2222
Connection: keep-alive
sec-ch-ua: "Google Chrome";v="93", " Not;A Brand";v="99", "Chromium";v="93"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/93.0.4577.63 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: pt-BR,pt;q=0.9,en;q=0.8
/milkshake.jpeg
HTTP/1.0 200 OK

Content-type: image/jpeg
./milkshake.jpeg
```

Assim, conclui-se os testes com sucesso em todos eles.

5. Conclusão

Esse trabalho permitiu melhor compreensão e análise de um servidor multithreaded, passo importante no desenvolvimento dos estudantes da turma da disciplina.