



Mini-Projeto 1: Servidor myWeb

O objetivo do Mini-Projeto 1 (MP1) é desenvolver, em linguagem Java, um servidor Web concorrente (*multi-threaded*), o *myWeb*, que seja capaz de responder a múltiplos pedidos de clientes em paralelo. Os alunos irão implementar um subconjunto do protocolo HTTP/1.1, tal como definido no RFC 2616 [1]. O HTTP/1.1 cria ligações TCP separadas para cada par pedido/resposta. Assim, o vosso servidor deve ser concorrente, criando uma *thread* por ligação, e deve ainda ter sempre uma *thread* principal usada pelo servidor para aguardar por novos pedidos de ligação.

Com este projeto vão ganhar experiência na programação de um serviço de Internet, nomeadamente:

- construindo um sistema realista e não-trivial,
- que obedece a um standard real usado pela indústria e pelos programadores de serviços Internet, e
- vão refletir acerca dos múltiplos serviços oferecidos pelos protocolos Internet que os programadores usam para desenvolver as suas aplicações.

Desenvolvimento

O código do *myWeb* será desenvolvido em três fases:

1. Na primeira há apenas uma *thread*, e o servidor apenas imprime o conteúdo do pedido HTTP feito para o ecrã. O pedido do cliente deve usar o método GET e incluir apenas o campo Host, devendo estar bem estruturado.
2. Na segunda fase o servidor já é concorrente, criando uma *thread* por pedido, mas continua apenas a imprimir o conteúdo do pedido HTTP para o ecrã.
3. Na terceira fase, o *myWeb* já responde, de forma apropriada, ao cliente, de acordo com os seguintes requisitos do protocolo HTTP/1.1 [1]:
 - a. O servidor só suporta o método GET (ver secção 5.1.1 de [1]). Para todos os outros métodos (PUT, HEAD, etc.), o servidor deve retornar o código “501 Method Unimplemented”.
 - b. O *myWeb* deve suportar 5 códigos de erro (ver secção 6.1.1 de [1]). Além do código 501, deve suportar: 400 (Bad request), 404 (Not Found), 408 (Request Timeout) e 505 (HTTP Version Not Supported). Mais detalhes na secção respetiva do capítulo 10 de [1].

Para resolver o MP1, e em particular o ponto 3, os alunos devem analisar o RFC 2616 [1]. Um RFC é um documento escrito com um estilo peculiar, com o qual a maior parte dos alunos não estará familiarizado. De qualquer modo, é importante ganharem alguma familiaridade com este tipo de *standard*, portanto esta é uma experiência relevante para o futuro académico e profissional de muitos alunos. Sendo um documento longo e “peculiar”, não é esperado que o

leiam na totalidade. Devem dar uma leitura na diagonal, lendo com alguma atenção a introdução, e depois lendo com mais detalhe as secções referentes ao que necessitam de implementar no ponto 3 (em particular as secções referidas em cima e outras relacionadas). Os slides da aula TP1 . HTTP também poderão ser úteis.

Referências de apoio ao projeto

Como ponto de partida de programação da aplicação sugere-se aos alunos a leitura do Capítulo 5 do livro Fundamentos de Redes de Computadores, com especial foco na secção 5.3. O livro está disponível online:

<http://legatheaux.eu/documents/cnfbook/cnfbook-pub.pdf>

No zip que contém este documento encontram também um cliente e um servidor básicos, assim como um servidor concorrente, escritos em Java, que podem usar como referência inicial, assim como um ficheiro `makefile`, que podem usar para compilar os programas (encontram alguns tutoriais simples sobre `makefile` nas referências [2] e [3]). Se necessário, os alunos devem alterar este `makefile`, ou criar um novo, de forma a permitir a compilação do programa que vão desenvolver. O zip contém também um ficheiro `mp1.html` que podem usar para testes (ver secção a seguir).

Avaliação autónoma do sistema

À medida que os alunos desenvolvem o código, devem ir testando o seu servidor Web usando como cliente um browser Web à sua escolha. Notem, no entanto, que não vão estar a usar a porta tradicional do HTTP (80), por isso devem especificar o número de porta que vão usar, juntamente com o URL que especificam no browser. Por exemplo, se usarem o URL do vosso sistema, `localhost`, o servidor estiver à escuta na porta 11111, e quiserem obter o ficheiro `mp1.html`, devem especificar o URL da seguinte maneira:

```
http://localhost:11111/mp1.html
```

Além disso, como boa metodologia de avaliação do programa desenvolvido, também aconselhamos os alunos a prepararem alguns scripts de teste, que incluam vários tipos de pedidos HTTP, verificação de condições de erro, etc. Os docentes usarão scripts deste tipo para avaliar os programas.

Entrega

Os alunos devem entregar, através do Moodle, três ficheiros zip, um por cada fase. Esse ficheiro deve incluir **todos** os ficheiros necessários para compilar e correr o servidor Web da fase respetiva, incluindo ficheiro(s) fonte, `makefile`, e (opcionalmente) script(s) de teste.

O trabalho deverá ser entregue até ao **final do dia 27 de Outubro de 2019**.

Avaliação do MP1

O MP1 vai ser avaliado pelos docentes da disciplina com base em 3 critérios:

1. Scripts de teste para verificação de funcionalidade do programa, fundamentalmente para verificar se o servidor trata corretamente os vários pedidos, válidos e inválidos;
2. Análise de código, incluindo como critérios de avaliação o estilo de programação e o uso apropriado de comentários no código;
3. Discussão com os alunos (opcional).

A distribuição de cotação pelas 3 fases é feita da seguinte forma:

- Primeira fase: 3 valores.
- Segunda fase: 5 valores.
- Terceira fase: 12 valores.

Nota: se o código não compilar e/ou não correr, por qualquer tipo de erro (incluindo a não entrega de algum ficheiro, como o `makefile` ou outro), o projeto é avaliado com zero valores.

Dúvidas e comunicação

Como neste projeto se procura fomentar a **autonomia** dos alunos, por regra os docentes não responderão a dúvidas na sala de aula. Assim, os alunos devem utilizar o Fórum da disciplina para tentarem esclarecer as dúvidas que vão surgindo.

Nota importante: Não é permitido os alunos partilharem código com soluções, ainda que parciais, a nenhuma parte do MP1, com alunos de outros grupos (nem através do Fórum, nem por qualquer outro meio). Se isso acontecer são anulados os projetos de todos os alunos envolvidos.

Referências

- [1] <https://tools.ietf.org/html/rfc2616>
- [2] <http://www.cs.colby.edu/maxwell/courses/tutorials/maketutor/>
- [3] <https://www.cs.swarthmore.edu/~newhall/unixhelp/javamakefiles.html>