

2021/2022

Programação Avançada – Projeto prático 1

1. Introdução

Com este projeto espera-se que os alunos demonstrem o conhecimento adquirido no primeiro terço da disciplina de Programação Avançada. Assim espera-se a entrega de um projeto em Java que resolva um problema descrito abaixo de uma forma paralela. De uma forma mais específica espera-se a utilização de:

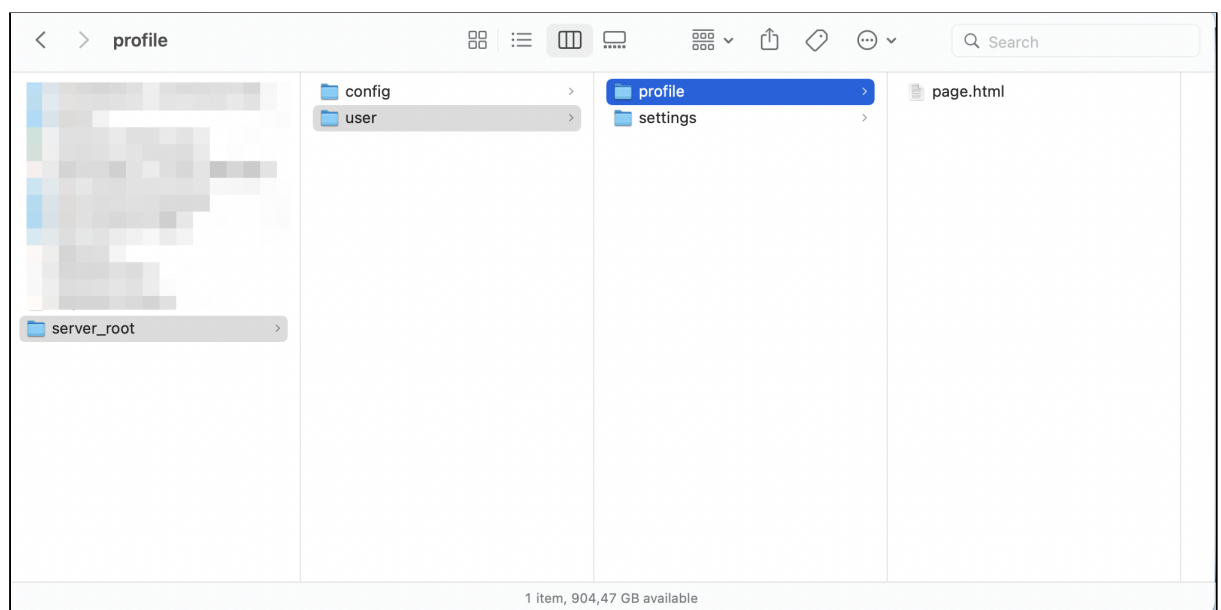
- Threads
- Semáforos
- Trancas
- Padrões Arquiteturais para o desenvolvimento paralelo

2. Descrição do problema

O projeto proposto, definido de agora adiante como “pa-web-server”, tem como objetivo a criação de um servidor Web em JAVA, tendo como ponto de partida [o código partilhado no repositório da disciplina](#). Espera-se que cada grupo estenda o projeto fornecido de forma a que este possa garantir o acesso paralelo aos recursos do servidor.

Neste servidor o utilizador deverá ser definido, um caminho para a raiz de um diretório. O servidor deverá depois ser capaz de “servir” todas as páginas .html estáticas neste diretório, e nos seus subdiretórios. Por exemplo, o documento “page.html” abaixo deverá ser retornado com o pedido <http://serverip:port/user/profile/page.html>.

As configurações do servidor, deverão ser lidas do ficheiro server/server.config partilhado no repositório do projeto base. Toda a parametrização adicional do servidor deverá ser adicionada a esse ficheiro.



Na prática serão avaliados os seguintes requisitos de implementação, estes requisitos terão que ser cumpridos para a aprovação do projeto:

2.1 Requisitos de implementação

1. Terão que ser integrados **pelo menos 2 padrões** de desenvolvimento distribuído discutidos nas aulas para a resolução do problema acima.
2. O servidor deverá suportar pelo menos 15 pedidos simultâneos.
3. Terão que ser utilizadas *Threads/Runnables*, Trancas e Semáforos.
4. O servidor deverá garantir que a qualquer momento, apenas uma thread tem acesso a um determinado documento .html do servidor.
5. Caso o documento html pretendido não esteja especificado no pedido, o servidor deverá retornar o ficheiro index.html de cada diretório.
 - a. Caso não exista nenhum ficheiro index.html no diretório em questão deverá ser retornada uma mensagem de erro 404.
6. Deverá ser mantido um ficheiro server.log com um registo de todos os pedidos ao servidor, sua origem e timestamp
 - a. O acesso a este ficheiro deverá ser protegido, garantido que o mesmo não é corrompido por acesso simultâneos.
 - b. O sistema deverá garantir que **todos** os pedidos sejam registados neste ficheiro.
 - c. O ficheiro server.log deverá seguir o seguinte formato:

<timestamp>-Method:<tipo de pedido>-Route:<rota/documento>-<origem>

```
2022-03-11 14:40:48.528-Method:GET-Route:/-/10.101.129.109
2022-03-11 14:40:48.612-Method:GET-Route:/favicon.ico-/10.101.129.109
2022-03-11 14:41:04.69-Method:GET-Route:/site1/index.html-/10.101.129.109
2022-03-11 14:41:04.76-Method:GET-Route:/site1/css/styles.css-/10.101.129.109
2022-03-11 14:41:07.867-Method:GET-Route:/site1/img/madeira.jpg-/10.101.129.109
```

3. Requisitos de entrega

1. O projeto deverá ser entregue até ao **dia 10 de Abril de 2022**. A entrega deverá ser realizada através do repositório da disciplina.
2. Juntamente com a implementação, deverão ser entregues testes unitários e relatório de *code coverage* gerado pela biblioteca jacoco (já incluída no pom.xml do projeto partilhado).
3. Projetos que não compilem, e cuja sua execução seja, assim, impossível não serão considerados para avaliação.

4. Todas as classes, e os métodos do programa, deverão apresentar documentação de suporte de acordo com as práticas de documentação JAVA (Javadoc) discutidas na aula TP2

De forma a garantir os requisitos nº 2,3 e 4 é obrigatório que os alunos implementem GitHub Actions para a execução dos testes, criação do relatório de *code-coverage*, e criação da documentação de suporte. Poderão seguir os exemplos criados na aula TP2.

4. Avaliação Individual

A contribuição individual de cada aluno será avaliada tendo em conta o histórico de atividade do mesmo no repositório do projeto. Logo que iniciem a implementação, **todos** os grupos deverão partilhar os seus repositórios privados com os docentes da disciplina (**Dntfreitas** e **Lipegno**).

5. Código de ética e honestidade académica

Nesta disciplina, espera-se que cada aluno subscreva os mais altos padrões de honestidade académica. Isto significa que cada ideia que não seja do aluno deve ser explicitamente creditada ao(s) respetivo(s) autor(es). O não cumprimento do disposto constitui uma prática de plágio. O plágio inclui a utilização de ideias, código ou conjuntos de soluções de outros alunos ou indivíduos, ou qualquer outra fonte para além dos textos de apoio à disciplina, sem dar o respectivo crédito a essas fontes. A menção das fontes não altera a classificação, mas os alunos não devem copiar código de outros colegas, ou dar o seu próprio código a outros colegas em qualquer circunstância. De notar que a responsabilidade de manter o acesso ao código somente para os colegas de grupo é de todos os elementos.