

Paradigmas de Programação

2019/2020

Projeto (TP3)

Enunciado

O trabalho prático de Paradigmas de Programação (PPROG) será realizado em articulação com a unidade curricular de Engenharia de Software (ESOFT). O trabalho de PPROG consiste na implementação da funcionalidade definida no seguinte Caso de Uso do projeto que está a ser desenvolvido em ESOFT, intitulado “Plataforma de Outsourcing de Tarefas”:

- Caso de Uso UC10 – seriar anúncio (este caso de uso apenas é introduzido na iteração 2 de ESOFT)

Na implementação do projeto devem ser considerados os requisitos da iteração 2 do trabalho de ESOFT. Os enunciados das iterações 1 e 2 encontram-se em anexo. Os respetivos artefactos podem ser consultados no site <https://bitbucket.org/pafomaio/esoft2019-2020-resolucao/> (os artefatos relativos à iteração 2 estarão disponíveis a partir do dia 4 de maio). Este repositório é partilhado pelo que o seu conteúdo não deve ser alterado.

Na implementação devem ser considerados dois tipos de seriação. Cada um destes tipos de seriação ordena as propostas através da aplicação sucessiva de um conjunto de critérios descritos de seguida:

- **Seriação 1**
 - Maior média dos níveis de proficiência em cada uma das competências técnicas exigidas para a tarefa;
 - Preço mais baixo;
 - Proposta registada mais cedo;
- **Seriação 2**
 - Maior média dos níveis de proficiência em cada uma das competências técnicas exigidas para a tarefa;
 - Menor desvio padrão dos níveis de proficiência em cada uma das competências técnicas exigidas para a tarefa;
 - Preço mais baixo;
 - Proposta registada mais cedo;

Contudo, ao longo do tempo, prevê-se a adoção de mais e variados tipos de seriação, como tal, pretende-se que a adição de novos tipos de seriação esteja facilitada e, se possível, possa até ser realizada por terceiros.

No processo de seriação apenas devem ser considerados os freelancers (candidatos) que possuam o grau de proficiência mínimo exigido a todas as competências técnicas obrigatórias para a tarefa em causa.

A interface com o utilizador da aplicação deve permitir:

- a visualização de todas as tarefas publicadas pelo colaborador e a seleção de uma dessas tarefas;
- a visualização das candidaturas à realização da tarefa selecionada;
- o despoletar do processo de seriação;

- a visualização do resultado da serialização.

De forma a que a aplicação desenvolvida seja funcional, será necessário criar objetos de várias classes. A implementação deverá incluir um mecanismo a executar no arranque da aplicação que permita criar os objetos das várias classes a partir de dados armazenados em ficheiros de texto. Por exemplo, o ficheiro usado para armazenar informação acerca de instâncias da classe Tarefa que possui os atributos “referência”, “designação”, “descrição informal”, “descrição técnica”, “duração estimada” e “custo estimado” poderá ter o seguinte formato:

```
ref1; designacao1; descrição informal 1; descrição técnica 1; duração 1; custo 1
ref2; designacao2; descrição informal 2; descrição técnica 2; duração 2; custo 2
...
```

Desta forma, após o arranque da aplicação, existirão em memória objetos, das várias classes, necessários ao funcionamento da aplicação.

Os requisitos funcionais da implementação incluem:

- implementação das funcionalidades descritas;
- implementação de uma classe que instancie todas as classes necessárias para que as funcionalidades descritas possam ser utilizadas/testadas durante a apresentação do trabalho;
- implementação de uma interface com o utilizador que permita usar as funcionalidades implementadas; a interface poderá ser baseada em consola ou poderá consistir numa interface gráfica (esta última com valorização adicional – ver tópico avaliação);
- implementação de um mecanismo de instanciação de classes a partir do conteúdo de ficheiros de texto – os ficheiros deverão estar disponíveis no momento da apresentação do projeto contendo informação para a instanciação de todas as classes necessárias ao funcionamento da aplicação.

Os requisitos não funcionais incluem:

- desenvolvimento em linguagem Java e ferramenta Maven);
- documentação do código, usando a ferramenta Javadoc;
- diagrama de classes em formato PDF;
- documento contendo a descrição das funcionalidades implementadas;
- implementação de testes unitários usando JUnit excetuando métodos getter, setter e toString (os métodos getter e setter deverão ser implementados apenas quando estritamente necessário);
- geração do relatório de cobertura de testes usando o plugin JaCoCo;
- utilização do repositório Bitbucket e da ferramenta de controlo de versões Git – **o repositório deverá ser privado e partilhado com o professor das aulas práticas – o nome do repositório deve respeitar o seguinte formato: Turma_TP3_NúmeroAluno1_NúmeroAluno2;**
- execução regular de commits por parte de cada um dos elementos do grupo durante todo o período de desenvolvimento do projeto;
- a utilização do repositório Bitbucket e da ferramenta de controlo de versões Git é obrigatória; a não existência deste repositório, assim como a falta de evidências acerca de *commits* regulares implica a não avaliação do trabalho.

Normas de Funcionamento

Constituição dos grupos

O trabalho será realizado em grupos constituídos por dois alunos da mesma turma PL. Os alunos deverão informar por e-mail os respetivos professores acerca da constituição dos grupos até ao dia **5 de maio de 2020**. O não cumprimento deste requisito implicará a não avaliação do trabalho.

No caso de não ser possível proceder à formação de um grupo constituído por elementos da mesma turma PL, serão admissíveis situações excecionais de grupos constituídos por elementos de diferentes turmas, desde que devidamente justificadas e após aprovação por parte dos professores das aulas PL.

Avaliação

A avaliação levará em consideração fatores como a implementação de funcionalidades, utilização de boas práticas de codificação, qualidade da documentação e apresentação.

A interface com o utilizador terá um peso de 10% na avaliação do trabalho. No caso de ser implementada uma interface de consola, o respetivo item de avaliação será avaliado no máximo em 5%, sendo este máximo de 10% no caso de ser implementada uma interface gráfica.

Alunos que não frequentam ESOFT

Sugere-se que os alunos que não frequentem ESOFT constituam grupo com um aluno que se encontre a frequentar ESOFT. Serão disponibilizados no site da Wiki de ESOFT propostas de modelo de casos de uso, modelo de domínio e modelo de *design* (diagrama de classes e diagrama de sequência).

Entrega

A entrega será realizada através de submissão no Moodle até às **23h30 do dia 24 de maio de 2020 (domingo)**. Esta entrega é constituída por um único ficheiro em formato ZIP, não sendo consideradas submissões de ficheiros com outros formatos. O ficheiro ZIP deve cumprir os seguintes requisitos:

- O nome do ficheiro deve seguir o seguinte formato **Turma_TP3_NúmeroAluno1_NúmeroAluno2** (exemplo: 1DA_TP3_1870049_1120999.zip);
- O ficheiro ZIP deve incluir o seguinte:
 - Pasta contendo o projeto criado no IDE NetBeans, cujo nome segue as regras definidas para o nome do ficheiro ZIP;
 - Pasta com nome Doc contendo a documentação em formato HTML gerada pela ferramenta Javadoc;
 - Ficheiro em formato PDF com o nome requisitos.pdf contendo a enumeração das funcionalidades implementadas, assim como uma breve descrição de cada uma delas;
 - Ficheiro em formato PDF com o nome diagramaClasses.pdf contendo o diagrama de classes implementado.

Não serão consideradas para avaliação submissões que não cumpram completamente os requisitos definidos para a entrega. Poderão ser feitas várias

submissões, sendo considerada apenas a última. As submissões devem ser efetuadas por **todos** os elementos do grupo.

Apresentação

Após a submissão, os trabalhos terão que ser apresentados pelos membros do grupo ao respetivo professor das aulas práticas durante a semana de 25 a 29 de maio através de videoconferência e em data e hora a definir.

Para a discussão todos os alunos deverão poder ser identificados de forma inequívoca, devendo ter disponíveis equipamentos com IDE e Java, câmara web e microfone, podendo, se necessário recorrer a um dispositivo móvel para o efeito. Durante a discussão os estudantes terão que partilhar o ecrã com o docente quando este o solicitar.