

Análise e Teste de Software

2018-2019

Projeto: UMeR

No contexto da disciplina de Programação Orientada a Objectos, no ano lectivo 2016-2017 foi proposto aos alunos criar um serviço de transporte de passageiros que faz concorrência a um serviço muito conhecido (e que tem um nome muito parecido com UMeR...). O enunciado desse projecto descreve informalmente os requisitos funcionais do sistema de software que os alunos tiveram de desenvolver, usando a linguagem de programação Java.

Uma possível solução para esse projeto é disponibilizada com este enunciado. De facto, essa solução foi desenvolvida por um grupo de alunos que frequentou a disciplina nesse ano lectivo, e foi avaliada pela equipa docente com a classificação de *Bom*. Esta avaliação foi feita com base em vários factores, incluindo a análise do funcionamento do sistema de software, demonstrado pelos alunos. Porém, estas demonstrações são muitas vezes feitas com poucos dados - não mostrando uma execução real da aplicação! - uma vez que não é possível no contexto de uma disciplina com um grande número de alunos/projeto testar devidamente este software.

Análise e teste da Aplicação UMer

Neste projecto de Análise e Teste de Software, pretendemos analisar e testar com grande detalhe o software desenvolvido pelos alunos. Assim, serão usadas técnicas e ferramentas que permitirão aferir da qualidade da aplicação desenvolvida pelos alunos.

O projecto deve ser realizado por grupos de 3 alunos e tem um peso de 50% da nota da disciplina.

1ª Fase: Validação da Solução dos Alunos de POO

O projecto será feito em três fases. Na primeira fase os grupos devem avaliar a solução apresentada pelos alunos de POO. Isto é, devem verificar se o software cumpre os requisitos definidos no seu enunciado. A solução deve ainda ser executada utilizando uma grande volume de informação. Assim, disponibiliza-se um ficheiro de *log* que contém o registo de todas as operações realizadas pela UMeR durante um certo período de tempo. Obviamente, a aplicação deve executar correctamente ao efectuar todas essas operações.

De notar que o ficheiro de *log* define uma linguagem (de domínio específico), que deve ser processada por um parser. Para desenvolver este parser, os alunos deverão utilizar o sistema ANTLR (www.antlr.org/): um gerador de parsers para Java. O parser deve ser integrado no software da UMer de modo a executar todas as operações em *batch*.

2ª Fase: Teste do Software UMeR

Enquanto que a 1ª Fase incidia sobre a funcionalidade do projecto UMeR a nível do seu uso por um utilizador, nesta 2ª Fase pretende-se avaliar a tanto a funcionalidade do código como das funcionalidades disponíveis aos utilizadores. Desta forma, será necessário trabalhar sob dois aspectos:

- Testes unitários;
- Testes de sistema.

2.1: Testes unitários

O código de cada classe poderá ter erros. Implemente testes que permitem encontrar os erros eventualmente presentes do programa e tome nota caso detecte algum erro. Na definição de testes unitários tenha em atenção os critérios de qualidade de testes definidos nas aulas teóricas (*critérios cobertura e uso de mutação de software em testing*).

2.1.1: Automação da Geração dos Testes Unitários

Utilize o sistema EvoSuite de modo a gerar testes unitários para a aplicação UMeR.

2.2: Testes de sistema

À semelhança da 1ª Fase, para testar o sistema será necessário executar instruções como se um utilizador as executasse. Para isso, será necessário gerar *inputs* aleatórios – semelhantes aos ficheiros de *log* que foram dados na 1ª Fase – e possivelmente com elementos adicionais para testar funcionalidades que os ficheiros de *log* fornecidos não permitiam.

Para realizar esta tarefa, terá de gerar novos casos de teste (*inputs de log*) e executá-los.

Recomenda-se a utilização do sistema QuickCheck. Poderá ser necessário a actualização da gramática da 1ª Fase.

Resultado esperado

Como resultado desta fase, pretende-se que a framework de teste da aplicação UMer, que terão de desenvolver neste fase, produza um relatório que atesta a qualidade dos testes e da aplicação em causa.

Indique no relatório que documenta esta fase, que partes do código tem erros e como os encontraram, entre outros pontos que achem de relevo. É pretendido também saber que alterações fizeram ao código, tanto ao da 1ª Fase como ao do projecto UMeR. Estão no papel de *testers* com o objectivo de informar os *developers* de erros.

3ª Fase: Análise da Qualidade do Software UMer

Esta fase incide na análise da qualidade do software desenvolvido. A análise de qualidade do software é feita em dois eixos: em primeiro lugar será feita uma análise da qualidade do código da aplicação, em segundo lugar será feita uma análise de performance desse mesmo software.

A qualidade do código é definida baseada em métricas extraídas do código fonte do projecto UMeR e da análise e eliminação de “bad smells” que aí ocorram. A análise de performance será feita monitorizando o tempo de execução, consumo de energia e consumo de memória do software UMer.

3.1 Métricas do Código Fonte

Use métricas de código fonte e indicações da ocorrência de bad smells (utilizando ferramentas próprias apresentadas nas aulas) para indicar a qualidade do código do projecto UMeR.

3.2 Métricas dos Testes

Use métricas para apresentar uma ideia da qualidade do código dos testes ao código do projecto UMeR desenvolvidos na segunda fase deste projecto.

3.3 Refactoring e Análise de Execução

Utilize os catálogos de **bad smells/refactorings** e ainda os **red smells** apresentados na aula teórica para efetuar o refactoring do código da aplicação UMeR. Valide os refactorings considerados executando o software UMeR na sua versão original e após os refactorings com um conjunto de inputs gerados na 2ª fase.

Utilize a ferramenta apresentada na aula para a análise de performance (tempo de execução e consumo de energia via RAPL) - e ainda um profiler de memória do Java - de modo a poder analisar o efeitos dos refactorings na performance do software. Efectue um teste o mais exaustivo possível sobre que refactorings melhoram (ou não) a performance (tempo, energia e memória) do software,

Avaliação:

A avaliação do projeto terá em conta os seguintes pontos:

- Qualidade das soluções entregues em cada uma das 3 fases de entrega;
- Qualidade da solução final;
- Uso do sistema de gestão de versões;
(acesso a um sistema de controlo de versões será disponibilizado)
- Apresentação Final;
- Relatório Final.

Datas de Entrega:

1ª Fase: 09 de Outubro, 2018 (estendido para 16 de Outubro)

2ª Fase: 13 de Novembro, 2018 (estendido para 19 de Novembro)

3ª Fase: 18 de Dezembro, 2018 (a confirmar na aula de 11 de Dezembro)

Apresentação Final: 08 de Janeiro, 2019.

