



INSTITUTO
SUPERIOR
TÉCNICO

Modelação Engenharia de Software

2º Semestre de 2008/2009

3ª entrega: Requisitos do sistema Test'O'Matic

Sistema Nacional para as Competências Profissionais de Utopia

11 de Maio de 2009

1 Introdução

Este documento descreve os requisitos para a terceira entrega do projecto de Engenharia de Software (ES). Os requisitos descritos neste enunciado assentam naqueles que foram apresentados no enunciado da segunda entrega.

Para a cadeira de Modelação esta terceira fase consistirá na entrega de dois artefactos:

- Pretende-se que agora os alunos revejam o que foi entregue na segunda fase e que modelem pela primeira vez as vistas ainda não modeladas do comportamento e estrutura de todos os sistemas envolvidos. No caso dos sistemas modelados em UML, deve ser feita ainda a modelação da respectiva arquitectura. Tudo isto deve ser modelado na ferramenta Enterprise Architect, devendo cada grupo entregar o respectivo ficheiro de projecto.
- Pretende-se ainda que, nesta fase, os alunos modelem uma ontologia, na ferramenta Portégé, devendo cada grupo entregar também o respectivo ficheiro. Os detalhes sobre esta entrega são descritos num documento complementar específico de Modelação.

2 Requisitos

Tanto o cliente como os utilizadores do sistema Test'O'Matic consideram crucial a implementação da noção de disciplina e das funcionalidades de eliminação de perguntas e de modelos.

2.1 Perguntas

Os utilizadores com a categoria de peritos passam a dispôr da funcionalidade de eliminação de perguntas.

Uma pergunta parcialmente introduzida ou que ainda não tenha sido utilizada pode ser eliminada instantaneamente. Quando se pretende eliminar uma pergunta que já tenha sido utilizada, esta deve ser marcada como eliminada. A pergunta original é mantida mas de forma a que não possa ser alterada, listada ou utilizada em novos testes.

2.2 Modelos

O docente pode eliminar modelos de teste. Um modelo que ainda não tenha sido utilizado é eliminado instantaneamente. Um modelo que já tenha sido utilizado deve ser marcado como eliminado.

Um docente pode listar os modelos de teste que criou. Os modelos que tenham sido eliminados não devem figurar nesta listagem.

2.3 Avaliações

Uma avaliação é definida por uma designação e passa a contar com a noção de disciplina.

As disciplinas são criadas por um utilizador com a categoria de técnico administrativo. Cada disciplina é composta por um código, por uma descrição, por um sumário dos tópicos leccionados e por uma indicação da carga horária semanal.

O técnico administrativo pode consultar as disciplinas que criou por forma a verificar quais os docentes que se associaram às mesmas, sendo-lhe ainda apresentado o número de alunos que se encontram inscritos a cada disciplina.



Qualquer situação anómala detectada pelo técnico administrativo é tratada directamente com os docentes. Para facilitar esse contacto, o sistema disponibiliza ao técnico os contactos do docente.

Um docente pode associar-se a uma ou mais disciplinas, escolhendo de entre a lista de disciplinas existentes. Apenas docentes associados a disciplina podem criar avaliações para essa mesma disciplina.

Após consultar os sumários dos tópicos leccionados de cada disciplina, um aluno pode increver-se nas disciplinas. O aluno poderá apenas realizar avaliações das disciplinas a que está inscrito.

Uma vez findado o período de avaliação, qualquer aluno que tenha realizado o teste pode visualizar o seu teste corrigido com a respectiva nota. O sistema apresenta os grupos em sequência, imprimindo as perguntas e as respostas possíveis para cada pergunta. A resposta correcta é sempre escrita em carregado. Caso a resposta do aluno esteja certa, esta é escrita a verde, caso contrário deve ser escrita a vermelho. Respostas erradas devem descontar na nota do teste. O valor a descontar é determinado dividindo a cotação da pergunta pelo número de respostas menos 1.

O aluno poderá ainda consultar uma pauta que agrega os resultados das avaliações efectuadas às diferentes disciplinas.

3 Desenvolvimento

3.1 Testes

Pelo menos os serviços (de ambos os componentes `core` e `web`) desenvolvidos para concretizar as secção ?? devem ser completamente testados de acordo com o *cookbook* da *framework*. A cobertura e qualidade dos testes para as restantes aplicações ficam ao critério de cada grupo, sendo no entanto uma das formas possíveis de demonstrar o funcionamento do componente `core` das aplicações.

3.2 Gestão de Projecto

Pretende-se que os alunos efectuem o levantamento de requisitos, modelem os diversos componentes a desenvolver, e estruturam o trabalho necessário em conjuntos de tarefas que demonstrem um evoluir constante do trabalho, ao longo do tempo disponível para a sua realização. O docente do laboratório fará o acompanhamento dessa gestão, em reuniões semanais.

O projecto encontra-se dividido em quatro sprints, sempre com início à 2ª feira. Todos os sprints devem ter a mesma duração (com a possível excepção do último sprint, que terá obrigatoriamente que terminar na 6ª feira, dia 5 de Junho). O planeamento deve ser realizado usando o template da folha de cálculo fornecida em <http://disciplinas.ist.utl.pt/leic-es/2008-2009/proj/scrums.xls>. Antes de cada sprint (exceptuando o 1º sprint, em que o prazo é o final de 3ª-feira, dia 12 de Maio), devem ser feitos o planeamento do novo sprint (em XLS) e a sprint retrospectiva incluindo conclusões da reunião relativas ao sprint anterior (em formato PDF). Estes dois documentos devem ter o nome grupo-sprint (em que grupo é o nome do grupo, correspondente ao módulo CVS; e sprint é o número do novo sprint – exemplo: A020310-3 seria o nome dos ficheiros do grupo A020310 relativo ao 3º sprint) e ser compactados num ficheiro ZIP com o mesmo nome. Este ficheiro deve ser submetido via Fénix, na página da cadeira de Engenharia de Software. Na reunião semanal, cada grupo deve apresentar em papel o estado actual do sprint e fazer uma demonstração sumária das histórias concretizadas no sprint anterior



É fundamental que o trabalho seja estruturado e dividido por forma a conseguir uma evolução incremental e sustentada. Segundo a filosofia SCRUM, cada sprint deve resultar numa *release* estável que permita demonstrar as funcionalidades desenvolvidas.

4 Entrega

A data limite para entrega é **5 de Junho de 2009 às 24:00**.

A entrega do trabalho é realizada através do repositório de CVS seguindo as regras descritas no documento “Utilização do CVS no projecto” (ver <http://disciplinas.ist.utl.pt/leic-es/2008-2009/proj/RegrasCVS.html>).

Os diagramas e descrição de casos de uso definidos devem ser colocados num directório **modelos**, ao passo que os artefactos resultantes de quaisquer tarefas que não sejam de desenvolvimento devem ser colocados, **em formato PDF**, num directório **gestao**, com nomes que indiquem explicitamente a tarefa a que se referem, juntamente com a folha de gestão correctamente preenchida.



A etiqueta a colocar para indicar a entrega desta fase do projecto é **RELEASE_3**.



O alvo `build` deve efectuar as operações necessárias para produzir aplicações que possam ser instaladas num servidor web, isto é, o resultado de efectuar `ant build` deve produzir os necessários ficheiros WAR nos respectivos directórios `dist`.

5 Avaliação

Esta terceira fase do projecto termina com:

- Visualização do projecto;
- Avaliação do código desenvolvido.

O primeiro ponto é realizado nas aulas de laboratório na semana seguinte à entrega do projecto. Consiste numa demonstração das funcionalidades do projecto e na discussão pedagógica com os elementos do grupo sobre as decisões tomadas e o seu impacto na qualidade do trabalho realizado.



O grupo deve preparar a demonstração num PC do laboratório, obtendo o projecto **a partir do repositório CVS**, efectuando o *deploy* e exercitando as várias funcionalidades desenvolvidas. ^a

^aA utilização do Eclipse é facultativa, pelo que fica ao critério de cada grupo decidir sobre a sua utilização ou não.

A segunda parte da avaliação é realizada pelo corpo docente e consiste na avaliação do projecto do ponto de vista da correcção da solução e do cumprimento das normas de utilização da arquitectura, seguindo os seguintes critérios:

- 35% Funcionalidade
- 5% Arquitectura
- 15% Modelação de Casos de Uso
- 5% Testes
- 40% Gestão de Projecto

FIM DO ENUNCIADO