

INF 2125 – PROJETO DE SISTEMAS DE SOFTWARE

Prof. Carlos J. P. de Lucena

Trabalho Experimental – Sistema de Biblioteca

1. Objetivo

Este trabalho tem o objetivo de consolidar o conhecimento sobre UML e garantir uma boa transição da fase de projeto para a de implementação de um sistema.

2. Visão Geral do Sistema

O sistema de gestão de produtividade acadêmica objetiva o gerenciamento da produção acadêmica de um laboratório de pesquisa, incluindo informações sobre projetos de pesquisa e colaboradores. O administrador do sistema é o usuário responsável pela manutenção de todas as informações do sistema.

Um laboratório de pesquisa é formado por colaboradores: alunos de graduação, alunos de mestrado, alunos de doutorado, professores e pesquisadores.

Os colaboradores de um laboratório podem ser alocados em projetos. Projetos possuem as seguintes informações básicas: título, data de início, data de término, agência financiadora, valor financiado, objetivo, descrição e participantes. Um projeto de pesquisa deve ter pelo menos um professor como participante.

O status inicial de um projeto é “Em elaboração”. Após finalizada sua elaboração, ou seja, constarem todas as informações básicas a respeito do projeto, o gerente pode iniciar o projeto, alterando seu status para “Em andamento”. A partir deste momento, o status do projeto somente poderá ser alterado pelo gerente para “Concluído”, e para que isto aconteça devem existir publicações associadas ao projeto.

Os colaboradores de um laboratório possuem dois tipos de produção acadêmica: publicações e orientações. Todos os colaboradores podem ter publicações, porém somente os professores podem ter orientações. Uma publicação é produzida por um ou mais autores. Uma publicação deve ter título, nome da conferência onde foi publicada, ano de publicação e projeto de pesquisa associado (se houver). Uma publicação só poderá ser associada a um projeto quando o status do projeto estiver “Em andamento”. As orientações devem ser feitas por professores, os quais devem orientar alunos do laboratório.

3. Requisitos Funcionais

1. O sistema deve permitir a edição de projetos de pesquisa
 - a. Alocação de participantes. Deve existir pelo menos um professor alocado para o projeto. Um aluno de graduação não pode participar de mais de dois projetos “em andamento”. A alocação deve ser permitida apenas quando o projeto estiver “em elaboração”.
 - b. Alteração do status

- i. “Em elaboração” para “Em andamento”. O gerente deve poder iniciar um projeto apenas se constarem todas as informações básicas a respeito do projeto cadastradas.
 - ii. “Em andamento” para “Concluído”. O gerente deve poder concluir um projeto apenas se existirem publicações associadas ao projeto.
2. O sistema deve permitir a inclusão de informações referentes à produção acadêmica. Uma publicação só poderá ser associada a um projeto quando o status do projeto estiver “Em andamento”.
3. O sistema deve oferecer as seguintes consultas:
 - a. Consulta por colaborador: dado um colaborador, o sistema deve mostrar suas informações: nome, e-mail, um histórico contendo a lista de projetos nos quais este colaborador participou, incluindo os projetos em andamento ordenados de forma decrescente pela data de término, incluindo também a lista de sua produção acadêmica. A produção acadêmica deve ser ordenada de forma decrescente de data (ano).
 - b. Consulta por projeto: dado um projeto, o sistema deve mostrar todos os dados do projeto, incluindo os colaboradores alocados e uma lista contendo toda a produção acadêmica do projeto, ordenada de forma decrescente de data (ano).
4. O sistema deve fornecer um relatório de produção acadêmica do laboratório, contendo:
 - i. Número de colaboradores
 - ii. Número de projetos em elaboração
 - iii. Número de projetos em andamento
 - iv. Número de projetos concluídos
 - v. Número total de projetos
 - vi. Número de produção acadêmica por tipo de produção

4. Critérios de Avaliação

O projeto será avaliado de acordo com os seguintes critérios:

1. Modelagem
 - a. Diagrama de casos de uso
 - i. Conformidade com a descrição do trabalho
 1. Os casos de uso atendem os requisitos?
 2. Todas as tarefas previstas estão modeladas?
 - ii. Uso correto de UML
 1. Relacionamentos (*extends*, *includes*, ...)
 2. Generalização
 3. Atores
 - iii. Descrição dos casos de uso
 - b. Diagrama de classes
 - i. Aplicação correta dos conceitos de OO
 1. Herança
 2. Polimorfismo
 3. Associações (Composição, Agregação, ...)
 4. Relacionamentos redundantes / cíclicos

5. Gerenciadores de objetos (OO)
 - ii. Conformidade com a descrição do trabalho
 - iii. Uso correto de UML:
 1. Multiplicidade
 2. Navegabilidade
 3. Papéis, nomes de relacionamentos, sentido de leitura
 4. Dependências
 5. Geral:
 - a. Classes soltas
 - b. Classes sem métodos e/ou atributos
 - c. Diagrama de seqüência
 - i. Conformidade com os diagramas de caso de uso e de classe
 1. Há pelo menos um diagrama de seqüência para cada caso de uso?
 2. Todos os objetos usados no diagrama de seqüência são instâncias de classes do diagrama de classes?
 3. Todos os métodos estão representados no diagrama de classes?
 4. Nas chamadas de métodos, as navegabilidades estão respeitadas?
 - ii. Uso correto de UML:
 1. Iteração
 2. Criação e destruição de objetos
 3. Atores
2. Implementação
 - a. Conformidade com todos os diagramas, especialmente com o de classes.
 - b. Todas as classes no código fonte (com exceção da API Java) estão representadas na modelagem?

Serão utilizados alguns casos de teste para verificar se o sistema desenvolvido atende aos mesmos. Portanto, é **MUITO** importante que os dados de teste sejam utilizados. Caso os dados de teste não sejam utilizados, os casos de teste falharão e a sua nota será prejudicada.

5. Quanto à Entrega do Trabalho

O dia da entrega do trabalho é **09/10/2017**. O trabalho deve ser feito individualmente. Os programas devem ser uma aplicação **web** redigidos em **Java** **preferencialmente**.

Devem ser entregues:

- Diagrama de casos de uso e descrição dos casos de uso;
- Diagrama de classes;
- Diagramas de seqüência;
- Manual de instalação do sistema. Caso as instruções do manual de instalação sejam seguidas e seu sistema não funcione, sua nota será prejudicada. O sistema deve ser entregues no formato **.zip**

Obs.: os diagramas devem estar **legíveis**.

Devem ser entregues em meio digital:

- Arquivos fonte e arquivos compilados para instalação do sistema;
- Arquivos contendo os modelos.

Obs. 1: O sistema não deve se preocupar com a persistência de dados. Entretanto, informações suficientes para efetuar os testes devem estar disponíveis desde o início da execução da aplicação.

Obs. 2: O sistema deve rodar no TomCat (<http://jakarta.apache.org>). Lembre-se que o seu sistema será instalado junto com o dos outros alunos, portanto não utilize o root e nem nomes de diretório como GESTAO_PRODUTIVIDADE. Crie um diretório com o seu nome para o seu sistema.

6. Dados de Teste

Alunos de graduação:

Id: 1001
Nome: Maria
e-mail: maria@email.br
Data ingresso: 01/03/2006
Orientador: 100

Id: 1002
Nome: João
e-mail: joao@email.br
Data ingresso: 01/03/2005
Orientador: 101

Id: 1003
Nome: Mário
e-mail: mario@email.br
Data ingresso: 01/03/2007
Orientador: 101

Alunos de mestrado:

Id: 1004
Nome: Soraia
e-mail: soraia@email.br
Data ingresso: 01/03/2006
Orientador: 100
Regime de curso: Parcial

Id: 1005
Nome: Rafael
e-mail: rafael@email.br
Data ingresso: 01/06/2007
Orientador: 102
Regime de curso: Parcial

Id: 1006
Nome: Marta
e-mail: marta@email.br
Data ingresso: 01/06/2007
Orientador: 102
Regime de curso: Integral

Id: 1007
Nome: Daniel
e-mail: daniel@email.br
Data ingresso: 01/03/2006
Orientador: 102
Regime de curso: Integral

Alunos de doutorado:

Id: 1008
Nome: Michael
e-mail: michael@email.br
Data ingresso: 01/03/2005
Orientador: 100
Regime de curso: Integral

Id: 1009
Nome: Bia
e-mail: bia@email.br
Data ingresso: 01/06/2004
Orientador: 100
Regime de curso: Integral

Professores:

Id: 100
Nome: Prof. Carlos
e-mail: carlos@email.br

Id: 101
Nome: Prof. Arnaldo
e-mail: arnaldo@email.br

Id: 102
Nome: Prof. Paulo
e-mail: paulo@email.br

Projetos:

Id: 20
Título: Engenharia de Software para Sistemas Multi-Agentes (ESMA)
Data de início: 02/02/2003
Data de término: 02/02/2010
Agência financiadora: FPCL
Valor financiado: R\$ 300.000,00
Objetivo: O objetivo geral deste projeto é desenvolver os fundamentos e as tecnologias da ESSMA.
Descrição: Pesquisar, aplicar e avaliar técnicas de desenvolvimento de software para sistemas multi-agentes..
Participantes: 100, 101, 1002, 1003, 1005, 1006, 1007, 1008, 1009
Status: Em andamento

Id: 30
Título: Engenharia de Software Orientada a Aspectos (ESOA)
Data de início: 02/02/2005
Data de término: 02/12/2011
Agência financiadora: FPCL
Valor financiado: R\$ 190.000,00
Objetivo: O objetivo geral deste projeto é desenvolver os fundamentos e as tecnologias da ESOA.
Descrição: Pesquisar, aplicar e avaliar técnicas de desenvolvimento de software orientado à aspectos..
Participantes: 101, 100, 1001, 1002, 1004, 1007, 1008
Status: Concluído

Id: 40
Título: Qualidade de Software
Data de início: 02/05/2006
Data de término: 02/10/2009
Agência financiadora: FPCL
Valor financiado: R\$ 100.000,00
Objetivo: O objetivo deste projeto é desenvolver os fundamentos e as tecnologias para desenvolvimento de software com qualidade.
Descrição: Pesquisar, aplicar e avaliar técnicas para qualidade em desenvolvimento de software.
Participantes: 102, 101, 1001, 1003, 1005, 1006, 1007, 1009
Status: Em elaboração

Id: 50
Título: Model-driven Software Product Lines Development
Data de início:
Data de término:
Agência financiadora: FPCL
Valor financiado: R\$ 500.000,00
Objetivo: O objetivo deste projeto é elaborar técnicas de engenharia de software dirigidas a modelos para o desenvolvimento de linhas de produtos de software.
Descrição: Pesquisar, aplicar e avaliar técnicas para o desenvolvimento de linhas de produtos de software.
Participantes: 102, 101, 1001, 1003, 1005, 1006, 1007, 1009
Status: Em elaboração

Id: 60
Título: Self-organizing Multi-agent Systems
Data de início: 15/07/2008
Data de término: 02/10/2010
Agência financiadora: FPCL
Valor financiado: R\$ 150.000,00
Objetivo: O objetivo deste projeto é desenvolver sistemas multi-agentes auto-organizáveis.
Descrição: Pesquisar, aplicar e avaliar técnicas para o desenvolvimento de sistemas multi-agentes auto-organizáveis.
Participantes: 1001, 1003, 1005, 1006, 1007, 1009
Status: Em elaboração

Produção acadêmica

Publicações:

Título: Abordagem Quantitativa para Desenvolvimento de Software Orientado a Aspectos
Autores: 1004, 1008, 100
Conferência: SBQS
Ano: 2006
Projeto de Pesquisa: 30

Título: Refactoring Product Lines
Autores: 1001, 1009, 101
Conferência: GPCE
Ano: 2007
Projeto de Pesquisa:

Título: Tratamento de Exceções Sensível ao Contexto
Autores: 1006, 102

Conferência: SBES
Ano: 2006
Projeto de Pesquisa:

Título: Integrating MAS in a component-based groupware environment
Autores: 1006, 1007, 102
Conferência: AOSE
Ano: 2006
Projeto de Pesquisa: 20

Título: Reputation Model Based on Testimonies
Autores: 1001, 1009, 100
Conferência: AAMAS
Ano: 2006
Projeto de Pesquisa: 20

Título: Extensions on Interaction Laws in Open Multi-Agent Systems
Autores: 1008
Conferência: SEAS
Ano: 2005
Projeto de Pesquisa:

Título: Aspect-oriented Patterns
Autores: 1004, 100
Conferência: FLOP
Ano: 2006
Projeto de Pesquisa: 30

Título: Classifying and Describing Agent Contracts and Norms
Autores: 1002, 1007
Conferência: AAMAS
Ano: 2005
Projeto de Pesquisa: 20

Orientações:

Orientador: 100
Aluno: 1001
Trabalho de Conclusão de Curso: Usabilidade no Portal do Banco do Brasil

Orientador: 100
Aluno: 1004
Dissertação: Framework para o Cálculo de Reputação de Agentes

Orientador: 102
Aluno: 1005

Dissertação: Arquitetura para Catálogos de Objetos baseado em Ontologias
--

Orientador: 102

Aluno: 1007

Dissertação: Framework para Smart Cards

Orientador: 100

Aluno: 1009

Tese: Linguagem de Modelagem para Sistemas baseados em Agentes
--