

Trabalho Prático

Enunciado da Etapa 2

Projetar uma aplicação.

Entrega

O grupo DESIGNER deve fazer um projeto orientado a objetos de uma aplicação (diagrama de classes e diagrama(s) de sequência) com base em uma especificação recebida. Este projeto será entregue para um grupo DEVELOPER, que deverá implementá-lo.

Entrega como DESIGNER (um por grupo)

Enviar via moodle um arquivo compactado (formato zip) contendo:

- Relatório contendo: (i) diagrama de classes – deve ser legível, possivelmente dividido em mais de um diagrama, contendo classes projetadas para toda aplicação (desde a interface ao armazenamento de dados); (ii) diagramas de sequência – devem ser legíveis, mostrando o fluxo de execução completo das funcionalidades projetadas; e (iii) explicação textual do projeto. O relatório deve ser entregue em um arquivo único no formato PDF.
 - ***Entregar o documento impresso (de preferência, frente-e-verso) até às 13h do dia seguinte à data de entrega. Caso o dia seguinte seja em final de semana, entregar 2ª-feira. Colocar o documento no meu escaninho (número 87 – Ingrid Nunes) localizado na sala 224 do prédio 43425(73). Não serão aceitos documentos entregues de outra forma.***
- Arquivo do projeto da ferramenta CASE escolhida (Astah ou StarUML), contendo os diagramas desenvolvidos.

Observações:

- Na próxima etapa, alguém tomará seu projeto, e desenvolverá o sistema.
- **Os projetos serão distribuídos seguindo o princípio de equivalência de qualidade.** A qualidade é definida pela qualidade do projeto, bem como respeito à especificação do projeto.
- Não deve ser implementada persistência no sistema. Crie uma classe que represente o banco de dados da aplicação, e métodos para a manipulação dos dados.
- **Atenção aos prazos! Caso você não entregue o projeto até o prazo da etapa 2, ou o projeto esteja incompleto, a etapa 3 valerá apenas 50% da nota!**

Especificação do Projeto a ser Realizado

Apesar de não ser um modo perfeito, a revisão por pares (*peer review*) é a melhor forma que temos hoje para decidir quais artigos científicos são de fato contribuições que devem ser publicadas. Por exemplo, uma conferência científica tem pesquisadores que formam o seu comitê de programa, que serão avaliadores de artigos submetidos à conferência. Assim, pesquisadores (parte ou não do comitê de programa) submetem artigos científicos com o intuito dos mesmos serem aceitos e publicados nos anais desta conferência. Os artigos submetidos são avaliados por N revisores (membros do comitê de programa), e cada um dá um parecer (quantitativo e qualitativo). Quem atribui um artigo para ser revisado para um membro do comitê de programa é o coordenador do comitê. Com base nestes pareceres feitos pelos revisores, o coordenador do comitê de programa decide quais artigos são aceitos e quais são rejeitados.

O seu objeto é desenvolver uma aplicação que auxilie o coordenador de comitê de programa a atribuir artigos para serem revisados por membros do comitê de programa, e selecionar os artigos a serem aceitos.

A inserção de dados iniciais é feita diretamente no banco de dados¹, e o cadastro de novos dados ainda não será implementado nesta versão da aplicação. Os seguintes dados são armazenados no sistema, e devem ser carregados no início da aplicação:

- **Pesquisador:** que possui id, nome, afiliação (universidade), tópicos de pesquisa de interesse.
- **Artigo:** que possui id, título, autor (que é um pesquisador), conferência, tópico de pesquisa relacionado.
- **Conferência:** que possui sigla, artigos submetidos, membros do comitê de programa (que são pesquisadores).

A aplicação deve ter três funcionalidades, descritas a seguir.

Funcionalidade

Alocação de Artigos a Membros do Comitê de Programa

1. Usuário (um coordenador do comitê de programa) solicita ao sistema a alocação de artigos a membros do comitê de programa.
2. Sistema solicita para qual conferência deve ser feita a alocação (para a qual não foi feita alocação ainda), e quantos revisores (membros do comitê de programa) devem ser alocados para cada artigo, que deve ser um número entre 2 e 5.
3. Usuário informa os dados solicitados.
4. Sistema faz a alocação dos artigos realizando o seguinte algoritmo
 - a. Sistema inicializa um conjunto de alocação com todos artigos da conferência
 - b. Sistema escolhe o artigo com menor id do conjunto
 - c. Sistema seleciona membros do comitê de programa como candidatos, que devem atender os seguintes critérios
 - i. membro que é o autor do artigo ou que pertence à mesma universidade que o autor do artigo é excluído
 - ii. membro cujos tópicos de pesquisa de interesse não inclui o tópico de pesquisa relacionado ao artigo é excluído
 - iii. membro que já foi alocado para revisar o artigo é excluído
 - d. Sistema ordena os membros candidatos pelo número de artigos já alocados a eles até o momento, em ordem crescente. O id do pesquisador é utilizado como critério de desempate (pesquisador com menor id é antecessor de pesquisador com maior id).
 - e. Sistema atribui a revisão do artigo ao primeiro membro da lista.
 - f. Sistema exclui o artigo atribuído do conjunto de alocação, e volta ao passo “b”
 - g. Caso os artigos já tenham o número de revisores informados pelo usuário, o algoritmo termina. Caso contrário, voltar ao passo “a”.
5. Sistema salva a alocação realizada, e exibe para o usuário uma lista dos artigos da conferência e os seus respectivos revisores, e um log do processo de alocação da seguinte forma:
 - a. Iniciando alocação.
 - b. Artigo id X alocado ao revisor id Y
 - c. [...]
 - d. Fim da Locação.

¹ Isto significa que dados devem estar hard-coded na classe que simula o banco de dados.

Atribuição de Notas a Artigos

1. Usuário solicita informar uma nota a um artigo, e sistema exibe a lista de artigos.
2. Usuário seleciona um artigo, e sistema exibe a lista de revisores
3. Usuário seleciona um revisor.
4. Sistema solicita uma nota entre -3 e 3 para o artigo, e usuário informa.
5. Sistema armazena a nota informada, relacionando ela com o revisor e com o artigo.

Seleção de Artigos

1. Usuário solicita ao sistema relatório de artigos aceitos.
2. Sistema solicita a seleção de uma conferência, e usuário informa.
3. Caso haja revisões (notas) não recebidas para aquela conferência o sistema emite alerta.
4. Caso contrário, sistema calcula a média das notas dos revisores e exibe os artigos em duas listas
 - a. Artigos cuja média de notas é ≥ 0 , aparecem na lista de artigos aceitos, em ordem decrescente
 - b. Artigos cuja média de notas é < 0 , aparecem na lista de artigos rejeitados, em ordem crescente

Dados Iniciais do Sistema

Pesquisadores

Id	Nome	Afiliação	Tópicos de Pesquisa
1	João	UFRGS	Software Product Lines Software Reuse Modularity
2	Ana	USP	Software Architecture Modularity Software Reuse
3	Manoel	UFRGS	Software Product Lines Software Testing
4	Joana	UFRJ	Software Product Lines Software Reuse Software Architecture Aspect-oriented Programming
5	Miguel	UFRGS	Software Architecture Modularity Software Testing
6	Beatriz	UFRJ	Software Reuse Software Testing Aspect-oriented Programming
7	Suzana	UFRGS	Aspect-oriented Programming Modularity Software Reuse
8	Natasha	UFRJ	Modularity Software Reuse Software Quality Software Product Lines
9	Pedro	USP	Aspect-oriented Programming Software Architecture
10	Carlos	USP	Software Reuse Modularity Software Testing

Conferências

Sigla	Membros do Comitê de Programa
ICSE	1, 2, 3, 4, 5, 6, 7
FSE	1, 2, 3, 4, 5, 6, 7
SBES	4, 5, 6, 7, 8, 9, 10

Artigos

Id	Título	Autor	Conferência	Tópico de Pesquisa
1	Coupling and Cohesion	1	SBES	Modularity
2	Design Patterns	6	FSE	Software Reuse
3	AspectJ	7	FSE	Aspect-oriented Programming
4	Feature Model	8	FSE	Software Product Lines
5	Architecture Recovery	9	FSE	Software Architecture
6	Funcional Testing	10	FSE	Software Testing
7	COTs	6	ICSE	Software Reuse
8	Pointcut	7	ICSE	Aspect-oriented Programming
9	Product Derivation	8	ICSE	Software Product Lines
10	Architecture Conformance	9	ICSE	Software Architecture
11	Structural Testing	10	ICSE	Software Testing

Atribuição de Revisões e Notas

Artigo	Revisor	Nota
1	8	2
1	10	
2	7	2
2	2	3
3	4	-1
3	6	1
4	1	1
4	3	0
5	4	-3
5	5	-3
6	3	-1
6	6	0