

Disciplina de Desenvolvimento de Sistemas - DSIS4

Curso Superior de ADS - 4º Semestre

(Professor Dênis Leonardo Zaniro)

Projeto Prático

Data de Entrega: 07/12/2018

Valor: 2,0 (Peso de 20% na média final da disciplina)

Informações básicas

- O projeto deve ser elaborado de maneira individual ou em duplas, e a entrega deve ser efetuada pelo sistema Moodle. No caso de duplas, apenas um aluno deve submeter o projeto. O projeto será apresentado ao professor na aula do dia 07/12/2018.
- A especificação de requisitos apresentada aqui é um subconjunto de requisitos do projeto prático da disciplina de Banco de Dados II (BD2S4). Alguns requisitos foram refinados para dar melhor suporte à implementação do sistema. Embora a estrutura do banco de dados seja a mesma, os projetos serão avaliados de forma independente.
- Entregas com atraso de, até dois dias, valerão 50% do valor inicial. Entregas com atraso superior a dois dias serão desconsideradas.
- Projetos cujos códigos sejam idênticos entre si serão anulados.

Objetivo

O objetivo do projeto é desenvolver um sistema de software em Java que gerencie empréstimos e devoluções de obras literárias em uma biblioteca de uma instituição acadêmica. Qualquer funcionário da biblioteca poderá utilizar o sistema, e os usuários da biblioteca (referenciados aqui como leitores) são alunos e servidores. A descrição do problema está organizada em requisitos funcionais e não funcionais, conforme pode ser observado a seguir.

Requisitos funcionais

RF1) O sistema deve permitir o cadastro de obras literárias da biblioteca. Cada obra possui os seguintes atributos: ISBN, título, categoria, autores, palavras-chave, data de publicação, número de edição e editora. A categoria (por exemplo, livro, revista, jornal, etc.) deve ser selecionada a partir de um conjunto de categorias inseridas no banco de dados. Considere que as categorias possíveis já estejam cadastradas. Cada obra possui um ou mais exemplares na biblioteca. O sistema deve atribuir um identificador único à cada um dos exemplares. Cada exemplar ou está disponível ou está emprestado.

RF2) O sistema deve permitir cadastrar obras literárias a partir da importação de dados das obras nos formatos XML e JSON. Caso o usuário escolha o formato XML, o sistema deverá fornecer três opções de API a serem utilizadas para leitura do XML: SAX, DOM e JAXB. Caso o usuário escolha o formato JSON, o sistema deverá fornecer duas opções de API: Gson e Jackson. Além disso, a aplicação deve solicitar que o usuário selecione o arquivo que contém os dados a serem importados (utilize o componente JFileChooser). Utilize como base o arquivo obras.xml (a estrutura do arquivo no formato JSON deve corresponder exatamente à estrutura do arquivo XML).

RF3) O sistema deve permitir realizar empréstimos de exemplares de obras literárias. Cada empréstimo possui os seguintes atributos: data de empréstimo da obra, data prevista para devolução, leitor, funcionário responsável e identificação dos exemplares das obras emprestadas. No ato do empréstimo, um dado leitor pode emprestar mais de um exemplar (apenas um exemplar de cada obra). Considere as seguintes regras: 1) Ao todo, cada leitor pode manter até 3 exemplares emprestados de obras diferentes; e 2) O empréstimo somente pode ser realizado se o leitor não possuir alguma devolução pendente com atraso. O sistema deve calcular a data prevista para devolução com base na categoria do leitor. Para servidores, o prazo para devolução é de 20 dias corridos e, para alunos, o prazo é de 15 dias corridos. Considere que leitores, suas categorias e funcionários já estejam cadastrados no banco de dados. Além disso, não é necessário considerar a realização de reservas.

RF4) O sistema deve permitir realizar devoluções de exemplares de obras literárias, com o registro da data de devolução. Observe que um leitor pode realizar a devolução de apenas um subconjunto de todas as obras emprestadas por ele. Caso um leitor não devolva um exemplar dentro do prazo, ele será considerado suspenso a partir do primeiro dia de atraso. Durante a suspensão, o leitor não poderá realizar empréstimos, conforme já foi mencionado.

RF5) O sistema deve permitir consultar, a partir de uma determinada data informada pelo funcionário, os dados de empréstimos que possuem exemplares cuja devolução está pendente e fora do prazo. O relatório deve exibir a data de empréstimo da obra, data que foi prevista para devolução, título da obra, número de edição, identificação dos exemplares com devolução pendente, nome e categoria do leitor. O relatório deve ser impresso na tela na forma de uma tabela, e cada linha da tabela representa um exemplar emprestado. Apenas 10 linhas devem ser exibidas ao mesmo tempo, isto é, elas devem ser exibidas de 10 em 10, conforme o funcionário navega pelo relatório (para frente ou para trás).

RF6) O sistema deve gerar e exportar um relatório com os dados de todas as obras. Para cada obra, o relatório deve exibir a quantidade total de exemplares no acervo e a quantidade disponível. O sistema deve oferecer três formatos para exportação: XML (utilize a API JAXB), JSON (utilize Gson ou Jackson) e PDF. Para exportação em PDF, pesquise uma API disponível em Java. Sugestões: iText e PDFBox.

Requisitos não funcionais

RNF1) Considerando-se o requisito RF2, durante a importação de dados, a tela deve permanecer responsiva, mesmo que o arquivo com os dados a serem importados seja grande. Durante a leitura do arquivo (XML ou JSON), a mensagem “Lendo arquivo” deve ser exibida e, ao término da leitura, a quantidade total de obras lidas do arquivo deve ser exibida. Uma vez que a leitura tenha sido finalizada, o sistema deve perguntar se o funcionário deseja confirmar a gravação dos dados. Caso o funcionário confirme, uma barra de progresso deve ser exibida para permitir acompanhar o andamento da gravação no banco de dados.

RNF2) Considerando-se o requisito RF5, utilize paginação para a consulta no banco de dados de tal forma que apenas os registros necessários para exibição sejam buscados.

RNF3) O sistema deve possuir interface gráfica implementada por meio de menus e janelas. Portanto, todas as operações devem ser adequadamente distribuídas entre janelas. Defina teclas de atalho associadas aos itens de menus. Adote uma estratégia para reutilização de código entre janelas.

RNF4) Utilize os componentes visuais mais adequados para a resolução do problema. Por exemplo, para satisfazer ao requisito RF5, deve-se utilizar o componente JTable (classes JTable, DefaultTableModel e JScrollPane).

RNF5) Todo o acesso ao banco de dados deve ser encapsulado por classes no padrão DAO. A estrutura de classes DAO deve ser projetada considerando-se a possibilidade (em versões posteriores do sistema) de se utilizar outra forma de comunicação com o banco de dados. Por exemplo, é possível em algum momento utilizar um framework como Hibernate em vez de se utilizar puramente a API JDBC. Observe, no entanto, que as implementações de DAO devem ser baseadas apenas na API JDBC (conforme foi estudado na disciplina).

RNF6) Os algoritmos necessários para leitura e escrita de arquivos XML e JSON devem ser encapsulados de tal forma que eles possam variar independentemente de seus clientes. Portanto, o código das classes clientes (classes de janela e quaisquer outras classes) deve ser desligado de qualquer uma das implementações. Em outras palavras, caso novos algoritmos sejam considerados pela aplicação, o código das classes clientes não deverá ser alterado.