PUCPR/ EP / BCC / Modelagem de Sistemas Computacionais / Prof. Edson Emilio Scalabrin / Segundo semestre de 2024

Equipe: (até 4 elementos) Mesma equipe do Trabalho 01, preferencialmente.

Trabalho 02

Essa atividade é continuação da Trabalho 01 vista na primeira parte do curso, cujo foco foi sobre: casos de uso, modelo de domínio, atribuição de responsabilidades (Framework GRASP + Diagrama de Sequencia) e classes de projeto; i.e., classes detalhadas incluindo atributos e métodos.

Esse Trabalho 02 foca nos modelos de atividades, estados, componentes e arquitetura de implantação.

Em resumo, o enunciado do Trabalho 02 é uma expansão do enunciado da Trabalho 01, incluindo detalhes com vistas a criação de modelos de atividades, estados, componentes e arquitetura de implantação.

O que já foi feito?

a) ler o texto, do item 1 ao 11 começando na página 5, para relembrar o problema que foi tratado no primeiro bimestre e também tomar ciência de pequenos ajustes feitos no texto, como por exemplo, interface. Para chamar atenção sobre esses ajustes, os mesmos estão em destaque na cor azul.
b) retomar o arquivo ASTAH que a equipe produziu no contexto do Trabalho 01, fazer os pequenos ajustes que achar relevante em função da leitura do item (a).

Relembrando o Enunciado do Trabalho 01

Enunciado: A empresa *iB2B* deseja informatizar o seu processo de empréstimo de livros. Para tal pediu ajuda a fabricante de software *BCC2022* para desenvolver um software que permite realizar reserva, empréstimo e devolução de livros.

Reserva, empréstimo e devolução são casos particulares de transação. Toda transação ocorre em uma data e envolve um livro e dois papéis: um estudante e um atendente. Tanto o papel estudante como atendente é assumido por uma pessoa. Todo papel tem um id (identificador). Toda pessoa tem nome, sobrenome e CPF. Todo livro tem título, um id (identificador) e um estado (que pode ser “livre”, “reservado” ou “emprestado”).

O software deve ser concebido e construído seguindo o paradigma da orientação a objetos e sua diagramação deve ser feita em UML. O software implementa as funcionalidades para os papéis já supracitados: Atendente e Estudante. O Estudante interage com o sistema *por meio da classe TratadorDeEmprestimo.* O Atendente interage com o sistema por meio da classe *TratadorDeDevolucao.* O Administrador da biblioteca interage com o sistema por meio da classe *TratadorDeAcervo*. Os livros são cadastrados no Acervo pelo Administrador. O Administrador da biblioteca também interage com o software por meio das classes: *TratadorDePessoa*, *TratadorDeEstudante* e *TratadorDeAtendente*. Esses três tratadores implementam a mesma interface dada pelas operações: incluir, excluir, atualizar, consultar Pessoa, Estudante e Atendente respectivamente.

O software deve fornecer os seguintes serviços: a) ao Estudante: solicitar reserva de livro (f1); consultar reservas de livros (f2); realizar/efetivar empréstimo de livro (f3); renovar empréstimo de livro (f4); b) ao Atendente: registrar devolução de livro (f5); emitir boleto de multa—caso livre em atraso (f6). As métodos correspondentes aos serviços f1, f2, f3 e f4 devem ser implementados na classe *TratadorDeEmprestimo*; e os serviços f5 e f6 devem ser implementados na classe *TratadorDeDevolucao*. A movimentação dada por reservas, empréstimos e devoluções deve ter um único ponto de acesso interno—chamado Movimentação—, porém mantidas por associações diferentes. O TratadorDeAcervo deve implementar a interface dada pelas operações: incluir, excluir, atualizar, consultar livro. O Acervo também implementa uma interface que permite incluir, excluir, atualizar, consultar um livro. Os tratadores *TratadorDePessoa*, *TratadorDeEstudante* e *TratadorDeAtendente* devem implementar a interface dada pelas operações: incluir, excluir, atualizar, consultar Pessoa, Estudante e Atendente respectivamente. A movimentação dadas por pessoa, estudante e atendente deve ter um único ponto de acesso interno—Relacionamento—, porém mantidas por

associações diferentes. O mecanismo a ser implementado para a comunicação entre os tratadores e as classes: Relacionamento, Acervo e Movimentação é *delegação*.

A dinâmica de uso do software é a seguinte: Estudante reserva um livro interagindo com software; Estudante consulta livro(s) reservado(s) interagindo com software. Estudante realiza/efetiva empréstimo de livro reservado/registrado no software; Estudante renova empréstimo de livro interagindo com software; Atendente entrega o livro emprestado à Estudante; Estudante devolve o livro emprestado à Atendente. Atendente registrar devolução de livro no software; (em caso de atraso), o Atendente, interage com o software, para solicitar a geração de multa no valor de R$ 1,00 p/dia de atraso e envia boleto ao Estudante por e-mail.

O que foi descrito acima representa um comportamento deseja para o software. Cada equipe pode incrementar tanto os elementos que definem o modelo de domínio como o comportamento geral do sistema, incluindo outros elementos de interface.

Relembrando Atividade/Trabalho 01: Dado o enunciado em questão:

a) identificar as classes e seus atributos;
b) identificar os relacionamentos (ex.: generalização, associação, agregação e/ou composição) entre as classes;
c) identificar os nomes dos relacionamentos, os nomes dos papéis para cada relacionamento, bem a multiplicidade/cardinalidade para cada relacionamento.
d) Identificar as funcionalidades do sistema;
e) Associar as funcionalidades identificadas as suas classes;
f) Desenhar o que foi identificado em UML;
g) Codificar em Java as classes e relacionamentos desenhados no item (g);
h) Identificar os casos de uso;
i) Identificar os relacionamentos entre casos de uso;
j) Identificar os atores; e
k) Desenhar o que foi identificado do item (h) até (j) em UML.
l) Desenhar em UML um diagrama de sequencia pelo menos para cada caso de uso de cada ator;

1. Regras para Manter Livro:

1.1. Inserir Livro: [se o livro já está cadastrado], então encerrar o procedimento de inserção e mostrar os dados do livro cadastro; [se o livro não está cadastrado], então criar e inserir o livro no **Acervo** de Livros.

1.2. Atualizar Livro: [se o livro não está cadastrado], então encerrar o procedimento de atualização; [se o livro está cadastrado], então atualizar os dados do livro.

1.3. Consultar Livro: [se o livro está cadastrado], então mostrar os dados do livro; [se o livro não está cadastrado], então encerrar o procedimento de consulta.

1.3. Excluir Livro: [se o livro não está cadastrado], então encerrar o procedimento de exclusão; [se o livro está cadastrado], então excluir o livro; por uma questão de simplificação não vamos tratar aqui integridade referencial.

OBS 1. Todos as funcionalidades da interface iLivro—inserir, consultar, atualizar, excluir—devem ser implementados na classe TratadorDeAcervo.

2. Regras para Manter Pessoa:

2.1. Inserir Pessoa: [se a pessoa já está cadastrada], então encerrar o procedimento de inserção e mostrar os dados da pessoa cadastra; [se a pessoa não está cadastrada], então criar e inserir a pessoa na associação **pessoa** da classe **Relacionamento**.

2.2. Atualizar Pessoa: [se a pessoa não está cadastrada], então encerrar o procedimento de atualização; [se a pessoa já está cadastrada], então atualizar os dados da pessoa.

2.3. Consultar Pessoa: [se a pessoa está cadastrada], então mostrar os dados da pessoa; [se a pessoa não está cadastrada], então encerrar o procedimento de consulta.

2.3. Excluir Pessoa: [se a pessoa não está cadastrada], então encerrar o procedimento de exclusão; [se a pessoa está cadastrada], então excluir a pessoa; por uma questão de simplificação não vamos tratar aqui integridade referencial.

OBS 2. Todos as funcionalidades da interface iPessoa—inserir, consultar, atualizar, excluir—devem ser implementados na classe TratadorDePessoa.

3. Regras para Manter Estudante:

3.1. Inserir Estudante: deve-se notar que todo(a) estudante deve estar ligada a uma pessoa. [se a pessoa já está cadastrada e o(a) estudante não está cadastrado(a)], então criar e inserir o(a) estudante na associação **estudante** da classe **Relacionamento**, [se a pessoa não está cadastrada ou o(a) estudante já está cadastrado(a)], encerrar o procedimento de inserção e mostrar os dados do(a) estudante cadastro(a).

3.2. Atualizar Estudante: [se o(a) estudante não está cadastrado(a)], então encerrar o procedimento de atualização; [se a pessoa está cadastrada e o(a) estudante está cadastrado(a)], então atualizar os dados do(a) estudante.

3.3. Consultar Estudante: [se o(a) estudante está cadastrado(a)], então mostrar os dados do(a) estudante; [se o(a) estudante não está cadastrado(a)], então encerrar o procedimento de consulta.

3.3. Excluir Estudante: [se o(a) estudante não cadastrado(a)], então encerrar o procedimento de exclusão; [se o(a) estudante está cadastrado(a)], então excluir o(a) estudante; por uma questão de simplificação não vamos tratar aqui integridade referencial.

OBS 3. Todos as funcionalidades da interface *iEstudante*—inserir, consultar, atualizar, excluir—devem ser implementados na classe TratadorDeEstudante.

4. Regras para Manter Atendente:

4.1. Inserir Atendente: deve-se notar que todo(a) atendente deve estar ligada a uma pessoa. [se a pessoa está cadastrada e o(a) atendente não está cadastrado(a)], então criar e inserir o(a) atendente na associação **atendente** da classe **Relacionamento**, [se a pessoa não está cadastrada ou o(a) atendente já está cadastrado(a)], encerrar o procedimento de inserção e mostrar os dados do(a) atendente cadastro(a).

4.2. Atualizar Atendente: [se o(a) atendente não está cadastrado(a)], então encerrar o procedimento de atualização; [se a pessoa está cadastrada e o(a) atendente está cadastrado(a)], então atualizar os dados do(a) atendente.

4.3. Consultar Atendente: [se o(a) atendente está cadastrado(a)], então mostrar os dados do(a) atendente; [se o(a) atendente não está cadastrado(a)], então encerrar o procedimento de consulta.

4.3. Excluir Atendente: [se o(a) atendente não está cadastrado(a)], então encerrar o procedimento de exclusão; [se o(a) atendente está cadastrado(a)], então excluir o(a) atendente; por uma questão de simplificação não vamos tratar aqui integridade referencial.

OBS 4. Todos as funcionalidades da interface iAtendente—inserir, consultar, atualizar, excluir—devem ser implementados na classe TratadorDeAtendente.

5. Regras para o caso de uso Reservar Livro:

5.1. consultar se o livro L1 está cadastrado no **Acervo** de livros**.**

5.2. consultar se o estudante E1 está cadastrado/inserido na associação **estudante** da classe **Relacionamento.**

5.3. [se L1 está cadastrado e E1 está cadastrado e estado de L1 é "livre"], então criar **Reserva**, indicado o idDoLivro, a matrícula do estudante; ajustar o estado do livro para "reservado"; adicionar a instância de **Reserva** em **Movimentação.**

6. Regras para o caso de uso Consultar reserva de Livro:

6.1. consultar se a reserva R1 está registrada na **Movimentação**, usando o idDoLivro e o idDoEstudante**.**

6.2. [se reserva R1 está OK], então mostrar dados da reserva.

7. Regras para o caso de uso Emprestar Livro:

7.1. consultar se a reserva R1 está registrada na **Movimentação**, envolvendo o idDoLivro e o idDoEstudante**.**

7.2. [se reserva R1 está OK], então emprestar livro, indicando o idDoLivro, a matrícula do estudante; ajustar o estado do livro para "emprestado". Remover a instância de **Reserva** de Movimentação e inserir a instância de **Empréstimo**.

8. Regras para o caso de uso Renovar Livro:

8.1. consultar se o empréstimo E1 está registrado na **Movimentação**, envolvendo o idDoLivro e o idDoEstudante**.**

8.2. [se estado do livro é “emprestado” e número de renovações é igual a 0], então renovar empréstimo do livro; mudar o estado do livro para “renovado” e ajustar data de vencimento da transação (em + 30 dias) e incrementar em 1 no número de renovações.

8.3. [se estado do livro é “renovado” e número de renovações é menor que 2], então renovar empréstimo de livro; manter o estado do livro para “renovado” e ajustar data de vencimento da transação (em + 30 dias) e incrementar em 1 no número de renovações.

9. Regras para o caso de uso Registrar livro devolvido:

9.1. consultar se o empréstimo E1 está registrado em **Movimentação**, envolvendo o idDoLivro e o idDoEstudante**.**

9.2. [se E1 está na Movimentação], então inserir uma instância de **Devolução** em Movimentação e remover E1 de Movimentação; ajustar o estado do livro para “livre”.

9.3. [se data da devolução de E1 ultrapassou o prazo], então invocar [extensão] caso de uso “10. Emitir boleto de multa.”

10. Regras para o caso de uso Emitir boleto de multa:

10.1. consultar se a devolução D1 está registrada na **Movimentação**, envolvendo o idDoLivro e o idDoEstudante**.**

10.2. [se D1 está em Movimentação e excedeu a data de devolução], emitir segunda via do boleto de cobrança de multa por atraso (R$ 1,00/dia de atraso).

INICIO DO TRABALHO 02

Antes de iniciar a definição do Trabalho 02, considere os seguintes ajustes no que foi feito no Trabalho 01: De forma resumida, o software deve fornecer as seguintes funcionalidades: manter Livro, manter Pessoa, manter Estudante, manter Atendente. Todas as funcionalidades do tipo "manter" são de responsabilidade do ator Administrador. Cada funcionalidade "manter" é definida por uma interface com os métodos: inserir, consultar, atualizar e excluir; por exemplo, a interface *iLivro* contém os métodos: inserir, consultar, atualizar, excluir; a interface *iPessoa* contém os métodos: inserir, consultar, atualizar, excluir; e de forma similar para as interfaces *iEstudante* e *iAtendente*. Essas interfaces são implementadas pelas classes *TratadorDePessoa*, *TratadorDeEstudante* e *TratadorDeAtendente*. **Restrições:** só pode ser cadastrado um(a) estudante ou um(a) atendente se a pessoa que será associada a estudante ou a atendente já esteja cadastrada.

Os itens do Trabalho 2 são 4, a saber:

1. representar o comportamento do sistema face aos estados e suas mudanças do objeto livro (item 11).
2. representar o comportamento do sistema face as atividades e fluxos de dados, controle e objetos entre atores e subsistemas (item 12).
3. representar a arquitetura do sistema em termos de componentes com suas interfaces fornecidas e requeridas.
4. Representar a arquitetura de implantação do sistema em termos de nós dispositivos e de execução.

---

11. Comportamento – ciclo de uso de um livro (diagrama de estados)

Dado o comportamento apresentado para cada caso de uso supra descrito, há uma dinâmica particular em torno dos estados de disponibilidade ou não de um livro. Em outras palavras, o ciclo de uso de um livro segue os seguintes passos:

a) quando ele (o livro) é cadastrado no sistema, ele é indicado no sistema como "livre" (ou disponível);
b) quando ele é associado a uma transação do tipo **Reserva**, ele é indicado no sistema como "reservado"; ele só pode ser indicado como "reservado" se ele se encontrava anteriormente como "livre";

c) quando ele é associado a uma transação do tipo **Empréstimo**, ele é indicado no sistema como “emprestado”; ele só pode ser indicado como “emprestado” se ele se encontrava anteriormente como “reservado”
d) quando ele é associado a uma renovação, ele é indicado no sistema como “renovado”; ele só pode ser indicado como “renovado” se ele se encontrava anteriormente como “emprestado” ou “renovado”; deve-se notar que a renovação só pode ocorrer 2 vezes.
e) quando ele é associado a uma transação do tipo **Devolução**, ele é indicado no sistema como “devolvido”; ele só pode ser indicado como “devolvido” se ele se encontrava anteriormente como “emprestado” ou “renovado”.
f) quando ele (o livro) vai para o setor de reparação, ele é indicado no sistema como “reparação”. Ele só pode ser indicado como “reparação” se ele se encontrava anteriormente como “livre” ou “emprestado” ou “renovado”.
g) quando ele sai para o setor de reparação, ele é indicado no sistema como “emReparação”. Ele pode ser indicado como “livre” se ele se encontrava anteriormente como “emReparação”, “emprestado”, “renovado” ou acabou de ser cadastro.

Ver exemplo, slide 11, 13, 16, 19, “Modelo dinâmico usando diagrama de estados 2024.pdf”

12. Comportamento – ciclo de uso de um livro (diagrama de atividades)

Dado o comportamento apresentado para cada caso de uso supra descrito e dos estados que um livro pode assumir, tem-se uma dinâmica particular em torno dos atividades realizadas pelas pessoas—estudante e atendente—com o sistema, em que a dinâmica básica é a seguinte:

a) primeiramente, os atores (administrador, estudante, atendente, banco) e os departamentos (Acadêmico, Financeiro, Biblioteca, RH (Recursos Humanos ou funcionários).
b) O ator Administrador mantem os cadastros de Pessoa, Estudante, Atendente, Livro. Em outras palavras, Administrador recebe em mãos formulários com os dados de Pessoa, Estudante, Atendente, Livro, e ele providencia os registros/cadastros destes dados no sistema. Deve-se notar que os estudantes são registrados no departamento Acadêmico, as pessoas e os atendentes são registrados no departamento de RH e os livros são registrados no departamento Biblioteca.
c) O ator Estudante pode registrar uma reserva no sistema (ou no departamento/subsistema Biblioteca). Essa reserva só tem sucesso se o livro requerido está cadastrado na Biblioteca e ele está livre para reserva. Se a reserva tem sucesso, o estado do livro é atualizado para reservado.
d) O ator Estudante pode registrar um empréstimo no sistema (ou no departamento/subsistema Biblioteca). Esse empréstimo só tem sucesso se o livro requerido está cadastrado na Biblioteca e ele já está reservado para o estudante requerente do empréstimo. Se o empréstimo tem sucesso, o estado do livro é atualizado para emprestado e o atendente entrega o livro ao estudante.
e) O ator Estudante pode registrar uma renovação de empréstimo no sistema (ou no departamento/subsistema Biblioteca). Essa renovação só tem sucesso se o livro está cadastrado na Biblioteca e ele já está emprestado para o referido estudante ou está como renovado e o número de renovações não excedeu 2.
f) O ator Atendente pode registrar devolução no sistema (ou no departamento/subsistema Biblioteca). Essa devolução só tem sucesso se o livro está cadastrado na Biblioteca e ele está efetivamente emprestado ou renovado para o referido estudante. O departamento Financeiro emite uma multa caso o livro tenha sido entregue fora do prazo definido, usando o CPF do referido estudante; esse último recebe um boleto bancário a ser quitado.
g) O Banco recebe o pagamento do estudante e informa quitação da divida/boleto ao departamento Financeiro.

h) O ator Atendente ao receber o livro, no contexto de uma devolução, verifique a situação do livro: caso o livro esteja em perfeito estado, registrar (no departamento/subsistema Biblioteca) a devolução e o livro já está novamente disponível/livre para reservas; caso o livro não esteja em bom estado de conservação, o livro vai para reparação e o livro fica como “emReparação”. Quando o livro voltar da reparação, o ator Atendente registra no sistema da Biblioteca que o livro está novamente livre.

Ver exemplo, slide 22, 23, 28, “Diagrama de atividades UML 2024.pdf”

13. Estruturação/arquitetura

13.1. Diagrama de componentes

Dados os atores Administrador, Estudante, Atendente, Banco e os departamentos ou subsistemas Acadêmico, Financeiro, Biblioteca, RH (Recursos Humanos), defina uma arquitetura baseada em componentes. A conexão dos componentes devem ser expressas em termos de interface requerida ou fornecida. Deve-se notar que um dado componente pode ter uma ou mais interfaces requeridas ou fornecidas.

Ver exemplo, slide 27, 21, 20 de “03_MSC-Diagrama Componente e Implantacao 2024.pdf”

13.2 Diagrama de implantação e instanciação

A implementação do sistema será observando os seguintes requisitos:

a) os gerenciadores de dados serão “MySQL” e “Oracle”. O gerenciador “Oracle” é um espelho do “MySQL”. O MySQL deve executar no ambiente operacional Windows 10 e o “Oracle” sobre Ubuntu. Tanto o “MySQL” e “Oracle” deve estar instalados em servidores diferentes e do tipo “Dell Edge 3600”.

b) a aplicação (lógica de negócio) deve executar no ambiente operacional Ubuntu, JVM 1090, Tomcat 10.0.11, https.

b) a interface com o usuário deve executar WebBrowser e conexão com o servidor de aplicação é http.

Ver exemplo, slide 37 de “03_MSC-Diagrama Componente e Implantacao 2024.pdf”