PUCPR
GRUPO MARISTA

Trabalho RA01

"Modelagem e implementação OO – primeiros passos"

Disciplina:	Modelagem de Sistemas Computacionais	**Ano/Semestre**:	2024/2
Professor:	Edson Emilio Scalabrin	**Metodologia:**	Produção de Modelos e Programas.
Resultado de Aprendizagem:	RA1. Emprega técnicas orientadas a objetos na criação de modelos estruturais, comportamentais e responsabilidades contratuais de acordo com as características do problema e contexto, e preceitos éticos e legais.		
Indicador de desempenho:	ID1. Mapeia classes, interfaces, relacionamentos e componentes em diagramas UML		

Enunciado: A empresa *iB2B* deseja informatizar o seu processo de empréstimo de livros. Para tal pediu ajuda a fabricante de software *BCC2024* 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/usuários: um estudante e um atendente. Tanto o papel/entidade estudante como atendente é assumido por um indivíduo do tipo pessoa. Todo papel tem um id (identificador). Toda pessoa tem nome, sobrenome e cpf. Todo livro tem título 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 atores/usuários: Atendente e Estudante. O ator Estudante interage com o sistema *por meio da classe TratadorDeEmprestimo e* O ator Atendente *interage com o sistema por meio da classe TratadorDeDevolucao.*

O software deve fornecer os seguintes serviços ou casos de uso: 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 dinâmica de uso do software pode ser a seguinte: o ator Estudante reserva um livro interagindo com software; o ator Estudante consulta livro(s) reservado(s) interagindo com software. O ator Estudante realiza/efetiva empréstimo de livro reservado/registrado no software; o ator Estudante renova empréstimo de livro junto interagindo com software; o ator Atendente entrega o livro emprestado pela entidade estudante; o ator Estudante devolve o livro emprestado ao ator Atendente. O ator Atendente registrar devolução de livro no software; (em caso de atraso), o ator 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.

Os livros disponíveis para reserva, consulta e empréstimo pertencem ao **Acervo de Livros** da biblioteca. As transações (reserva, empréstimo e devolução) pertencem à **Movimentação** da biblioteca. Os usuários acadêmicos (ex. estudante) da biblioteca pertencem ao **Grupo de Acadêmicos**. Os usuários colaboradores (ex. atendente) da biblioteca pertencem ao **Grupo de Colaboradores**. As pessoas usuárias da biblioteca pertencem ao **Grupo de Pessoas**. As entidades Livro, Estudante, Atendente, Pessoa implementam uma interface CRUD (Create—ou inserir, Update—ou atualizar, Read—ou consultar e Delete—ou excluir). De forma resumida o sistema fornece as seguintes funcionalidades: manter Livro, manter Pessoa, manter Estudante, manter Atendente. Todas as funcionalidades do tipo "manter" são de responsabilidade do ator Gerente Operacional. Todos os métodos (inserir, consultar, atualizar e excluir) devem ser implementados na classe TratadorDeCadastro. **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.

1. Regras para Manter Livro:

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

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

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

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

OBS 1. Todos as funcionalidades—inserir, consultar, atualizar e excluir—devem ser implementados na classe TratadorDeCadastro e renomeados como segue: inserirLivro, consultarLivro, atualizarLivro e excluirLivro.

2. Regras para Manter Pessoa:

2.1. Inserir Pessoa: [se pessoa cadastrada], então encerrar o procedimento de inserção e mostrar os dados da pessoa cadastra; [se pessoa não cadastrada], então criar e inserir a pessoa no **Grupo De Pessoas**.

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

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

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

OBS 2. Todos as funcionalidades—inserir, consultar, atualizar e excluir—devem ser implementados na classe TratadorDeCadastro e renomeados como segue: inserirPessoa, consultarPessoa, atualizarPessoa e excluirPessoa.

3. Regras para Manter Estudante:

3.1. Inserir Estudante: deve-se notar que todo(a) estudante deve estar ligada a uma pessoa. [se pessoa cadastrada e estudante não cadastrado(a)], então criar e inserir o(a) estudante no **Grupo De Acadêmicos**, [caso contrário] encerrar o procedimento de inserção e mostrar os dados do(a) estudante cadastro(a).

3.2. Atualizar Estudante: [se estudante não cadastrado(a)], então encerrar o procedimento de atualização; [se pessoa cadastrada e estudante cadastrado(a)], então atualizar os dados do(a) estudante.
3.3. Consultar Estudante: [se estudante cadastrado(a)], então mostrar os dados do(a) estudante; [se estudante não cadastrado(a)], então encerrar o procedimento de consulta.
3.3. Excluir Estudante: [se estudante não cadastrado(a)], então encerrar o procedimento de exclusão; [se estudante cadastrado(a)], então excluir o(a) estudante. [não vamos tratar aqui integridade referencial]
OBS 3. Todos as funcionalidades—inserir, consultar, atualizar e excluir—devem ser implementados na classe TratadorDeCadastro e renomeados como segue: inserirEstudante, consultarEstudante, atualizarEstudante e excluirEstudante.

4. Regras para Manter Atendente:
4.1. Inserir Atendente: deve-se notar que todo(a) atendente deve estar ligada a uma pessoa. [se pessoa cadastrada e atendente não cadastrado(a)], então criar e inserir o(a) atendente no **Grupo De Colaboradores**, [caso contrário] encerrar o procedimento de inserção e mostrar os dados do(a) atendente cadastro(a).
4.2. Atualizar Atendente: [se atendente não cadastrado(a)], então encerrar o procedimento de atualização; [se pessoa cadastrada e atendente cadastrado(a)], então atualizar os dados do(a) atendente.
4.3. Consultar Atendente: [se atendente cadastrado(a)], então mostrar os dados do(a) atendente; [se atendente não cadastrado(a)], então encerrar o procedimento de consulta.
4.3. Excluir Atendente: [se atendente não cadastrado(a)], então encerrar o procedimento de exclusão; [se atendente cadastrado(a)], então excluir o(a) atendente. [não vamos tratar aqui integridade referencial]
OBS 4. Todos as funcionalidades—inserir, consultar, atualizar e excluir—devem ser implementados na classe TratadorDeCadastro e renomeados como segue: inserirAtendente, consultarAtendente, atualizarAtendente e excluirAtendente.

5. Regras para o caso de uso Reservar Livro:
5.1. livro = consultar se o livro pertence ao **Acervo de Livros.**
5.2. estudante = consultar se o estudante pertence ao **Grupo de Acadêmicos.**
5.3. [se livro cadastrado e estudante cadastrado e estado do livro é “livre”], então criar **Reserva**, indicado o id do livro, a matrícula do estudante; ajustar o estado do livro para “reservado”. Inserir/adicionar a instância de **Reserva** em **Movimentação.**
6. Regras para o caso de uso Consultar reserva de Livro:
6.1. reserva = consultar se há uma reserva registrada na **Movimentação**, envolvendo o livro idLivro e o estudante idEstudante**.**
6.2. [se reserva está OK], então mostrar dados da reserva.

7. Regras para o caso de uso Emprestar Livro:
7.1. reserva = consultar se há uma reserva registrada na **Movimentação**, envolvendo o livro idLivro e o estudante idEstudante**.**
7.1. [se há reserva], então emprestar livro, indicado o id do livro, 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. empréstimo = consultar se há um empréstimo registrado na **Movimentação**, envolvendo o livro idLivro e o estudante idEstudante**.**
8.1. [se estado do empréstimo é "emprestado"], então renovar empréstimo de livro; ajustar o estado do livro para "renovado" e data de vencimento da transação (em + 30 dias). [se estado do empréstimo é "renovado"], então encerrar procedimento sem renovação, pois o livre já foi renovado uma vez.

9. Regras para o caso de uso Registrar livro devolvido:
9.1. empréstimo = consultar se há um empréstimo registrado em **Movimentação**, envolvendo o livro idLivro e o estudante idEstudante**.**
9.2. [se há empréstimo], então inserir a instância de **Devolução** do livro em Movimentação. ajustar o estado do livro para "livre".
9.3. [se data de devolução > hoje], então invocar [extensão] caso de uso "10. Emitir boleto de multa"

10. Regras para o caso de uso Emitir boleto de multa:
10.1. devolução = consultar se há uma devolução registrada na **Movimentação**, envolvendo o livro idLivro e o estudante idEstudante**.**
10.2. [se há devolução e data de devolução > hoje], emitir boleto de cobrança de multa por atraso (R$ 1,00/dia de atraso).

Pede-se para fazer:

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; e
f) Desenhar o que foi identificado do item (a) até (e) em UML.
g) Identificar os casos de uso;
h) Identificar os relacionamentos entre casos de uso;
i) Identificar os atores; e
j) Desenhar o que foi identificado do item (g) até (i) em UML.
k) Fazer os diagramas DE SEQUÊNCIA para os seguintes casos de uso:
   a. Manter Pessoa, Manter Estudante e Manter Atendente.
   b. Emprestar Livro
   c. Renovar empréstimo de Livro
   d. Registrar devolução de Livro
   e. Emitir boleto de multa
l) Montar uma apresentação (preferencialmente no PPT) incluindo os seguintes diagramas:
   a. Casos de uso.
   b. Domínio (aqui o diagrama pode conter apenas classes, relacionamentos (associação, agregação, composição, generalização), cardinalidades. NÃO PRECISA DOS

ATRIBUTOS E MÉTODOS, pois a ideia é ter um modelo conceitual para o problema em questão.

c. Sequencias (um diagrama de sequência para cada caso de uso, sigam os exemplos fornecidos para “manter livro”, “reservar livro”, “consultar reserva”).
d. Classes de projeto (aqui o diagrama de classes deve conter todos os métodos e atributos).
e. **Data de entrega e apresentação**: Ver plano de ensino

IMPORTANTE: Caso se perceba que faltam métodos, parâmetros, atributos, acrescente no modelo de classes. Cuide para os nomes dos métodos inseridos nas classes são os mesmos presentes que representam as mensagens dos diagramas de sequência.

Não deixe de perguntar ao professor suas dúvidas!!!