



UNIVERSIDADE ESTADUAL DO NORTE DO PARANÁ
Centro de Ciências Tecnológicas
Curso de Ciência da Computação

PROF. ANDRÉ LUIS ANDRADE MENOLLI
ALUNA: MARIA LUISA MOREIRA DE SOUZA

TRABALHO FINAL

BANDEIRANTES
2023

1. Descrição da Arquitetura do Sistema.....	2
2. Funcionalidades cadastrar Livros e Alunos.....	3
3. Emprestar Livro.....	4
3.1. Caso de Uso.....	4
3.2. Diagrama de Caso de Uso.....	5
3.3. Atualização do diagrama de classe.....	6
3.4. Diagrama de Sequência.....	6
3.5. Implementação do caso de uso.....	6
4. Devolver Livro.....	7
4.1. Caso de Uso.....	7
4.2. Diagrama de Caso de Uso.....	8
4.3. Diagrama de Classes.....	9
4.4. Diagrama de Sequência.....	9
4.5. Implementação do caso de uso.....	10
5. Discussão.....	10

1. Descrição da Arquitetura do Sistema

A arquitetura proposta consiste em uma abordagem modular e escalável, que visa automatizar e otimizar as operações relacionadas ao empréstimo, consulta e reserva de livros na biblioteca da universidade. A arquitetura é composta por diferentes camadas, cada uma com responsabilidades específicas, garantindo a separação de preocupações e facilitando a manutenção e evolução do sistema.

Model-View-Controller (MVC):

- ***Model:*** Representa a lógica de negócios do sistema. Inclui componentes que gerenciam os dados da biblioteca, como informações sobre livros, empréstimos e reservas. O modelo também lida com a verificação de disponibilidade, regras de renovação e reservas.
- ***View:*** Representa a camada de interface do usuário. Inclui a interface do usuário para pesquisa avançada, renovação de empréstimos e reserva de livros. A *View* é responsável por coletar informações do usuário e exibir resultados.
- ***Controller:*** Atua como um intermediário entre o Modelo e a Visão. Para o caso de uso "*Reservar Livro*", o *Controller* gerencia a solicitação do usuário de reserva, verifica disponibilidade e encaminha os resultados à Visão. Para "*Pesquisa*", o *Controller* lida com a definição de critérios e recuperação de resultados do Modelo.

DAO (Data Access Object):

O padrão ***DAO*** é utilizado na camada de persistência para abstrair o acesso ao banco de dados. Isso permite desacoplar a lógica de acesso a dados da lógica de negócio, facilitando a substituição ou atualização do banco de dados sem afetar outras partes do sistema. Além disso, o ***DAO*** promove a reutilização de código ao fornecer uma interface consistente para acesso aos dados.

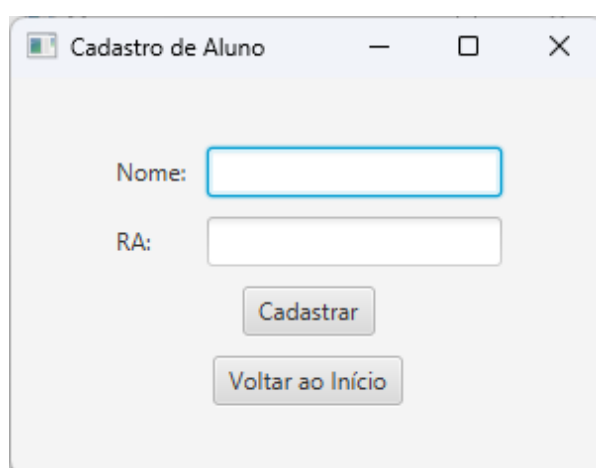
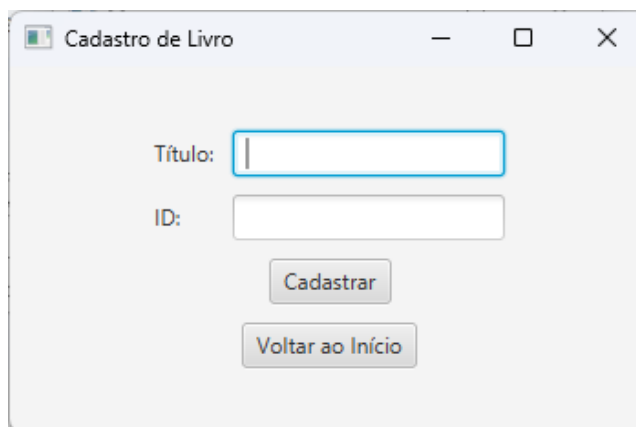
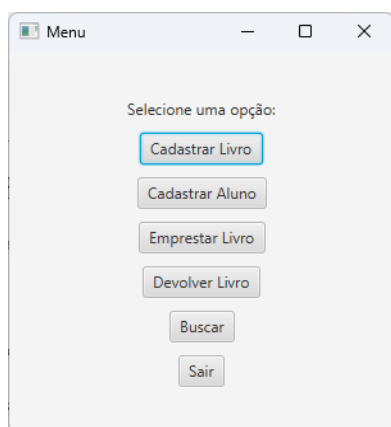
Justificação:

O padrão arquitetural ***MVC*** foi escolhido porque ele separa claramente a lógica de negócios, a interface do usuário e o controle, facilitando a manutenção e a escalabilidade do sistema. O Modelo pode ser responsável por gerenciar os dados dos livros, usuários e empréstimos, aplicando regras de negócios. A Visão fornece uma interface amigável para os usuários interagirem com o sistema. O *Controller* gerencia a comunicação entre o Modelo e a Visão, garantindo que as operações de reserva, renovação e pesquisa sejam realizadas de maneira eficaz. Além disso, a utilização do padrão ***DAO*** na camada de persistência proporciona flexibilidade e facilita a integração com diferentes tecnologias de banco de dados. Isso é importante para garantir que o sistema possa ser adaptado a diferentes ambientes e requisitos de negócio no futuro.

2. Funcionalidades cadastrar Livros e Alunos

Adicionar funcionalidades para cadastrar Livros e Alunos (inclusive com interface gráfica).

Foram implementadas as funcionalidades de cadastro de Livros e Alunos, aliadas à criação de uma interface gráfica intuitiva. Através dessa interface, os usuários podem inserir informações, como o nome do aluno e seu *RA*, ou o nome do livro e seu código, e posteriormente submeter esses dados para serem armazenados no banco de dados, mediante o acionamento do botão "*Cadastrar*". Essa abordagem proporciona uma maneira simples e eficiente de interagir com o sistema, permitindo que os usuários registrem novos alunos e livros de forma rápida e conveniente. Abaixo segue algumas imagens da interface gráfica:



3. Emprestar Livro

3.1. Caso de Uso

O caso de uso "*Emprestar Livro*" descreve a interação entre o usuário e o sistema quando o usuário deseja emprestar um livro da biblioteca.

Caso de Uso: Emprestar Livro

Ator Principal: Bibliotecário

Pré-condições:

- O livro e o aluno devem estar cadastrados no sistema.
- O livro deve estar disponível para empréstimo.
- O aluno não deve possuir débitos pendentes.

Fluxo Principal:

1. O bibliotecário seleciona a opção de emprestar um livro no sistema.
2. O sistema solicita ao bibliotecário que insira o *RA* (*Registro Acadêmico*) do aluno.
3. O bibliotecário insere o *RA* do aluno.
4. O sistema solicita ao bibliotecário que insira o código do livro.
5. O bibliotecário insere o código do livro.
6. O sistema solicita ao bibliotecário que insira a data prevista para a devolução do livro.
7. O bibliotecário insere a data prevista para a devolução do livro.
8. O sistema valida os dados fornecidos.
9. O sistema verifica se o aluno possui débitos pendentes.
10. O sistema verifica se o livro está disponível para empréstimo.
11. O sistema registra o empréstimo do livro para o aluno.
12. O sistema altera o status do livro para indisponível.
13. O sistema registra a data prevista para a devolução do livro.
14. O sistema confirma o empréstimo para o bibliotecário.

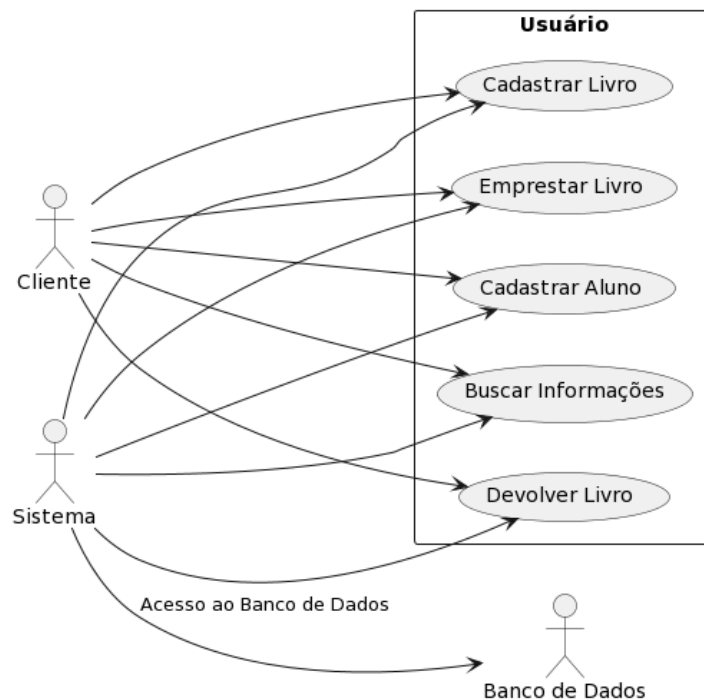
Pós-condições:

- O livro é emprestado para o aluno.
- O status do livro é alterado para indisponível.
- O registro do empréstimo é armazenado no sistema.

Fluxo Alternativo:

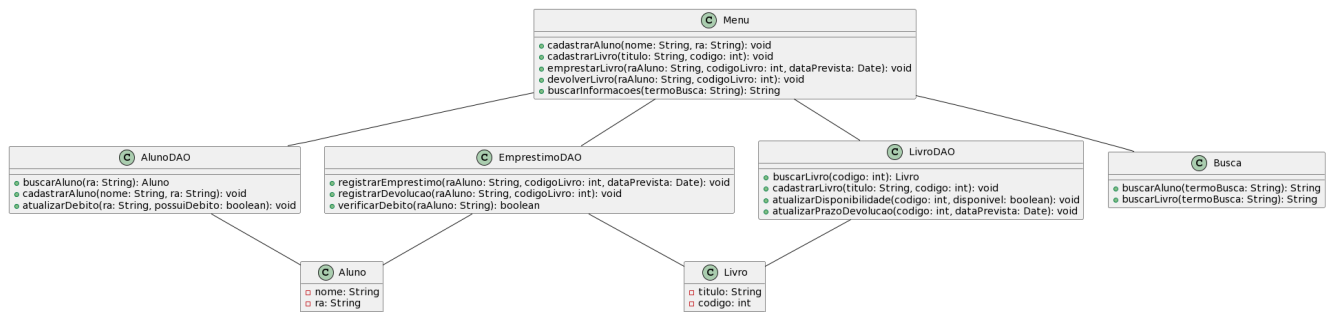
- Se os dados fornecidos forem inválidos, o sistema exibe uma mensagem de erro e retorna ao *passo 2*.
- Se o aluno possuir débitos pendentes, o sistema exibe uma mensagem informando que o aluno não pode realizar empréstimos.
- Se o livro não estiver disponível para empréstimo, o sistema exibe uma mensagem informando que o livro não está disponível.
- Se ocorrer algum erro durante o processo, o sistema exibe uma mensagem de erro e retorna ao *passo 2*.

3.2. Diagrama de Caso de Uso

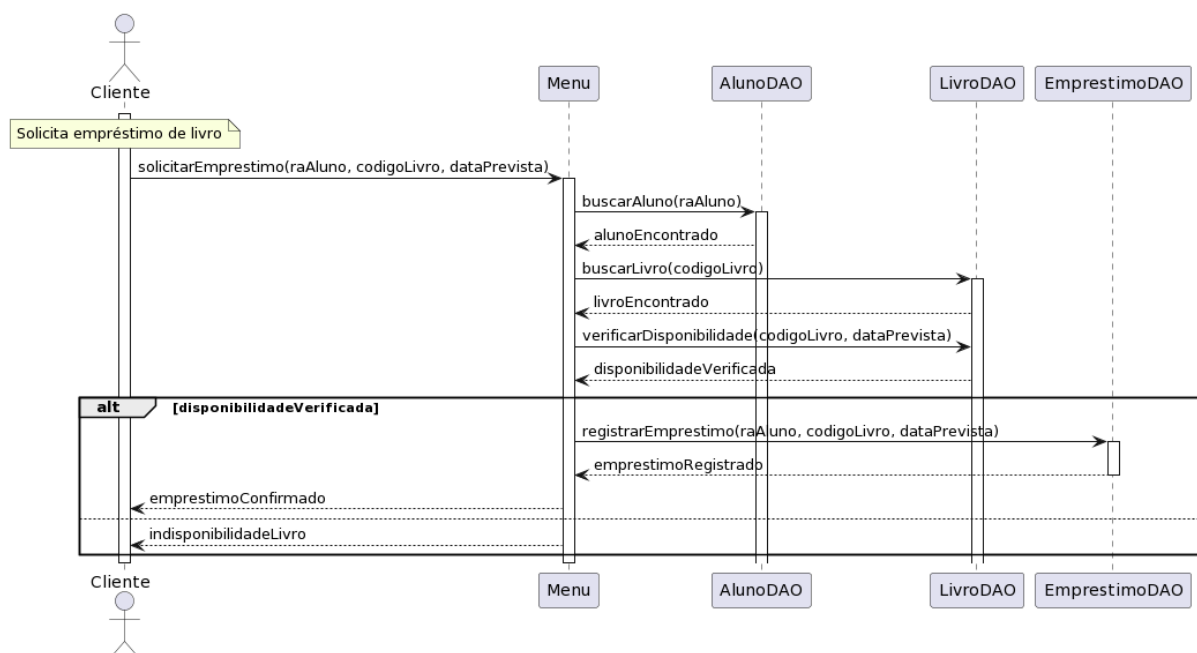


3.3. Atualização do diagrama de classe

Inserção da camada de persistência e aplicação do padrão DAO.



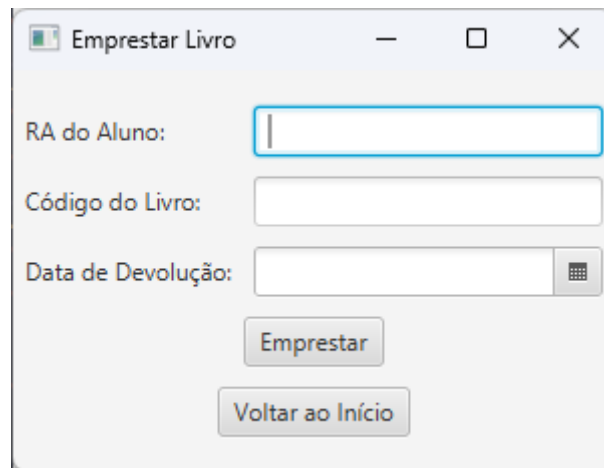
3.4. Diagrama de Sequência



3.5. Implementação do caso de uso

Foi implementado o caso de uso *"Emprestar Livro"* na interface gráfica, que funciona da seguinte maneira: o usuário fornece o *RA* do aluno e o código do livro a ser emprestado, juntamente com a data prevista para a devolução do livro. Contudo, algumas regras são aplicadas: é necessário que tanto o livro quanto o aluno estejam registrados no banco de dados, o livro deve estar disponível para empréstimo e o aluno não pode possuir débitos pendentes.

Uma vez cumpridas essas condições, o banco de dados é atualizado com essas informações. O empréstimo é registrado, incluindo sua data prevista de devolução. Além disso, o status do livro é alterado para *indisponível* e sua data prevista de devolução é registrada. Essas medidas garantem a integridade das operações de empréstimo, contribuindo para a gestão eficiente dos recursos do sistema.

A imagem mostra uma janela de software intitulada "Emprestar Livro". Ela contém três campos de entrada: "RA do Aluno:" com um campo de texto e uma seta para alternar entre maiúsculas/minúsculas; "Código do Livro:" com um campo de texto; e "Data de Devolução:" com um campo de texto e um ícone de calendário. Abaixo dos campos, há dois botões: "Emprestar" e "Voltar ao Início".

4. Devolver Livro

4.1. Caso de Uso

O caso de uso "*Devolver Livro*" descreve a interação entre o usuário e o sistema quando o usuário deseja devolver um livro da biblioteca.

Caso de Uso: Devolver Livro

Ator Principal: Cliente

Pré-condições: O cliente possui um livro emprestado e deseja devolvê-lo.

Fluxo Principal:

1. O cliente seleciona a opção de *devolver livro* na interface do sistema.
2. O sistema solicita ao cliente que insira o *RA* do aluno e o código do livro a ser devolvido.
3. O cliente fornece o *RA* do aluno e o código do livro.
4. O sistema verifica se o livro está emprestado para o aluno informado.
5. Se o livro estiver emprestado para o aluno:
6. O sistema registra a data e hora atuais como a data de devolução do livro.

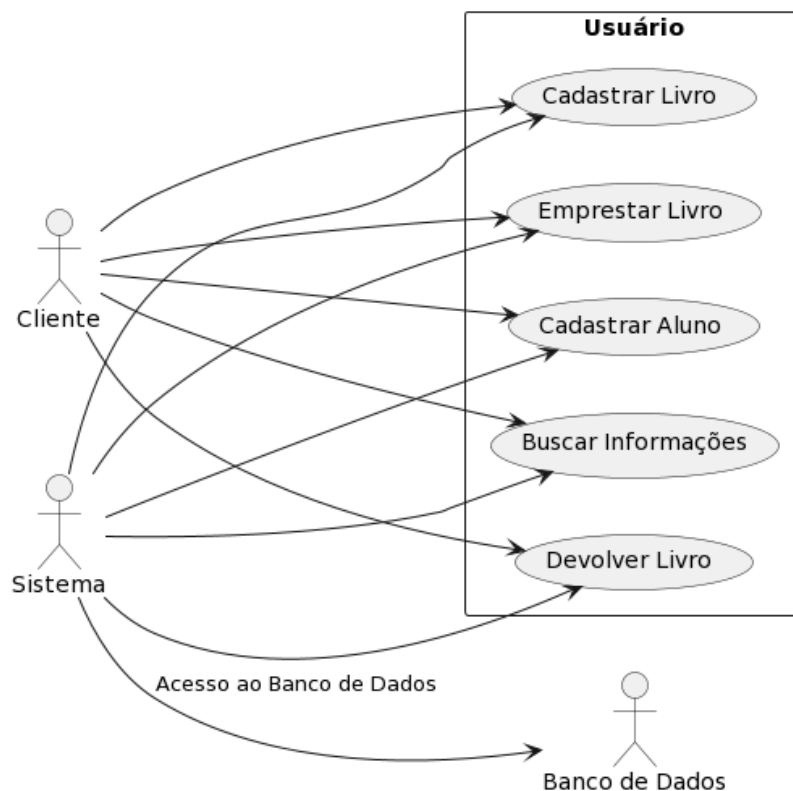
7. O sistema marca o livro como disponível novamente no banco de dados.
8. O sistema remove o registro de empréstimo do livro para o aluno.
9. O sistema exibe uma mensagem de confirmação de devolução bem-sucedida ao cliente.

Fluxo Alternativo:

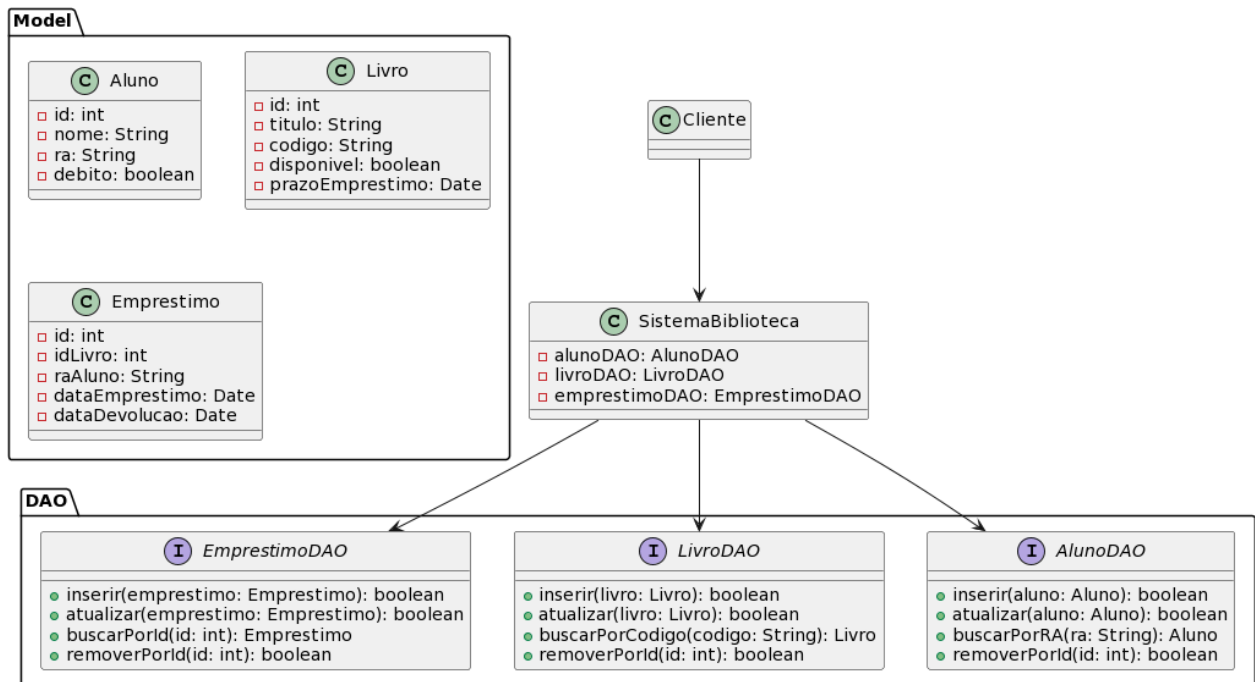
- Se o livro não estiver emprestado para o aluno informado, o sistema exibe uma mensagem de erro e retorna ao *passo 3* do fluxo principal.
- Se ocorrer algum erro durante o processo, o sistema exibe uma mensagem de erro e retorna ao *passo 3* do fluxo principal.

Pós-condições: O livro é devolvido com sucesso e está disponível para empréstimo novamente.

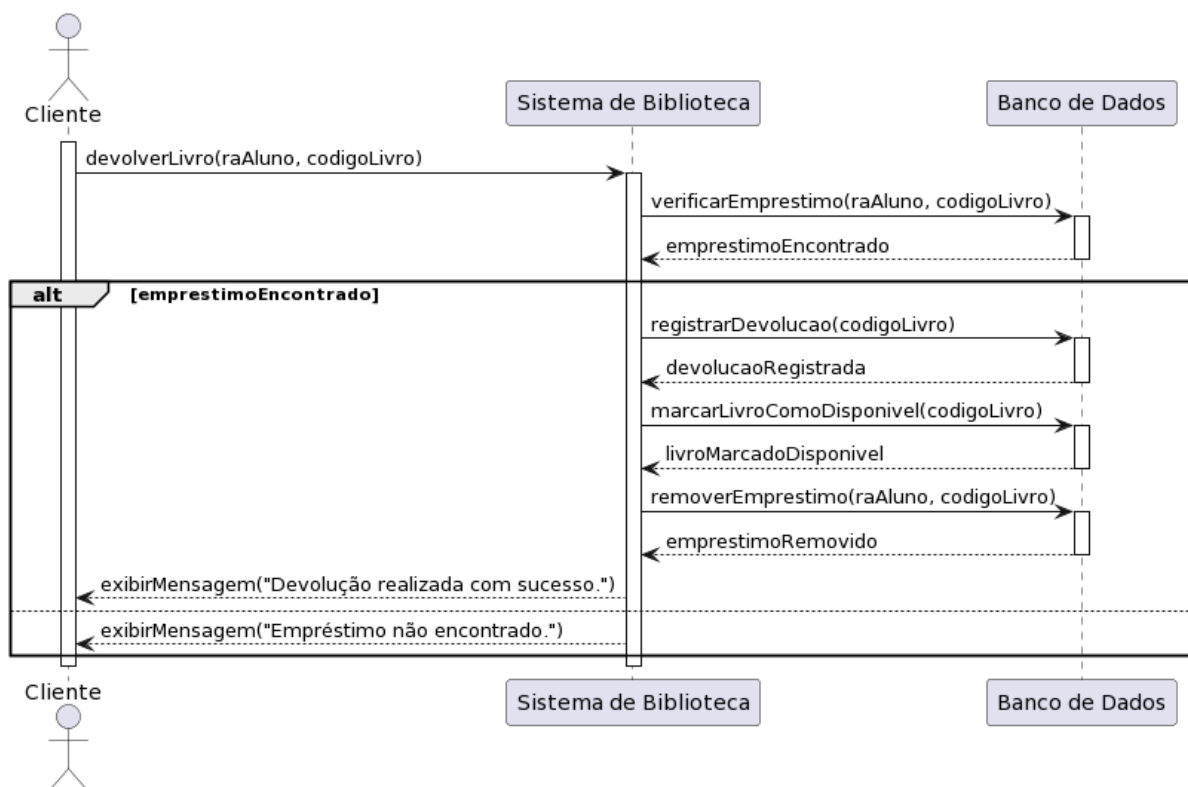
4.2. Diagrama de Caso de Uso



4.3. Diagrama de Classes



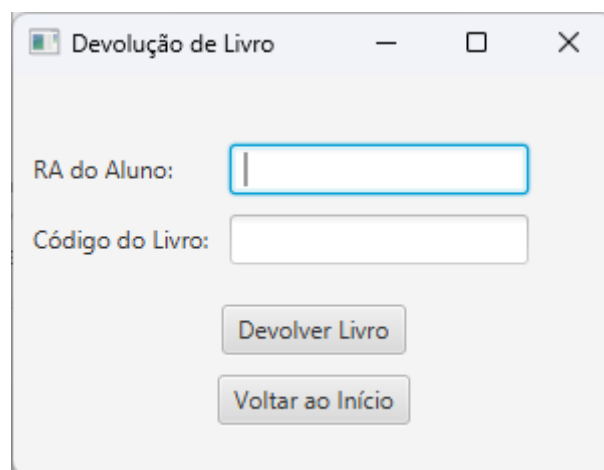
4.4. Diagrama de Sequência



4.5. Implementação do caso de uso

Foi implementado o caso de uso "*Devolver Livro*" na interface gráfica, cujo funcionamento requer que o usuário forneça o *RA* do aluno e o código do livro a ser devolvido. O sistema então registra no banco de dados o horário e a data atual como a data de devolução do livro. Algumas regras são aplicadas durante esse processo: tanto o livro quanto o aluno devem estar registrados no banco de dados, o aluno deve ter o empréstimo atualmente registrado em seu nome para que a devolução seja permitida. Após essas verificações, o banco de dados é atualizado: a disponibilidade do livro é registrada como "*disponível*", a data de empréstimo é removida dos dados do aluno e a data de devolução é registrada no banco.

Além disso, caso a data prevista de devolução seja superior à data de devolução atual, é registrado um débito para o aluno. Essas medidas asseguram a integridade do processo de devolução e contribuem para a gestão eficiente do sistema.

A imagem mostra uma janela de software intitulada "Devolução de Livro". Dentro da janela, há dois campos de entrada de texto. O primeiro campo é rotulado "RA do Aluno:" e o segundo "Código do Livro:". Abaixo dos campos, há dois botões: "Devolver Livro" e "Voltar ao Início". A interface é simples, com uma barra de título e botões de controle padrão.

5. Discussão

Qual seria o custo para que o projeto suporte um outro tipo de *SGBD* como mecanismo de armazenamento persistente.

A discussão sobre a migração para outro tipo de Sistema de Gerenciamento de Banco de Dados (*SGBD*) implica em considerar diversos aspectos que afetariam o custo e a complexidade do projeto.

1. Compatibilidade de SQL: Se o novo *SGBD* suporta a mesma linguagem de consulta *SQL* (*Structured Query Language*) ou requer adaptações significativas no código *SQL* existente. Migrações entre *SGBDs* diferentes muitas vezes exigem ajustes nas consultas *SQL*, o que pode exigir tempo e esforço consideráveis.

2. Conexão e Configuração: É necessário avaliar se as conexões com o novo *SGBD* podem ser estabelecidas facilmente a partir do código existente. Além disso, a

configuração do novo *SGBD*, incluindo permissões de acesso e configurações de segurança, deve ser gerenciada.

3. Treinamento e Familiaridade: A equipe de desenvolvimento deve estar familiarizada com o novo *SGBD* ou disposta a adquirir conhecimento sobre ele. Isso pode exigir treinamento adicional e tempo para se familiarizar com as melhores práticas e peculiaridades do novo sistema.

4. Migração de Dados: Um dos maiores desafios é migrar os dados existentes do banco de dados atual para o novo *SGBD*, garantindo a integridade e consistência dos dados durante o processo. Isso pode exigir scripts de migração complexos e extensivos testes para garantir que nenhum dado seja perdido ou corrompido.

5. Testes e Validação: Após a migração, é essencial realizar testes abrangentes para verificar se todas as funcionalidades do sistema continuam operacionais e se os dados foram migrados corretamente. Isso pode envolver testes de unidade, integração e aceitação, bem como testes de desempenho para garantir que o sistema atenda aos requisitos de performance.

6. Manutenção Contínua: Uma vez que a migração tenha sido concluída com sucesso, é necessário considerar os custos contínuos de manutenção e suporte do novo *SGBD*. Isso inclui a aplicação de patches de segurança, atualizações de software e otimizações de desempenho conforme necessário.