

# SISTEMA DE MATRÍCULAS

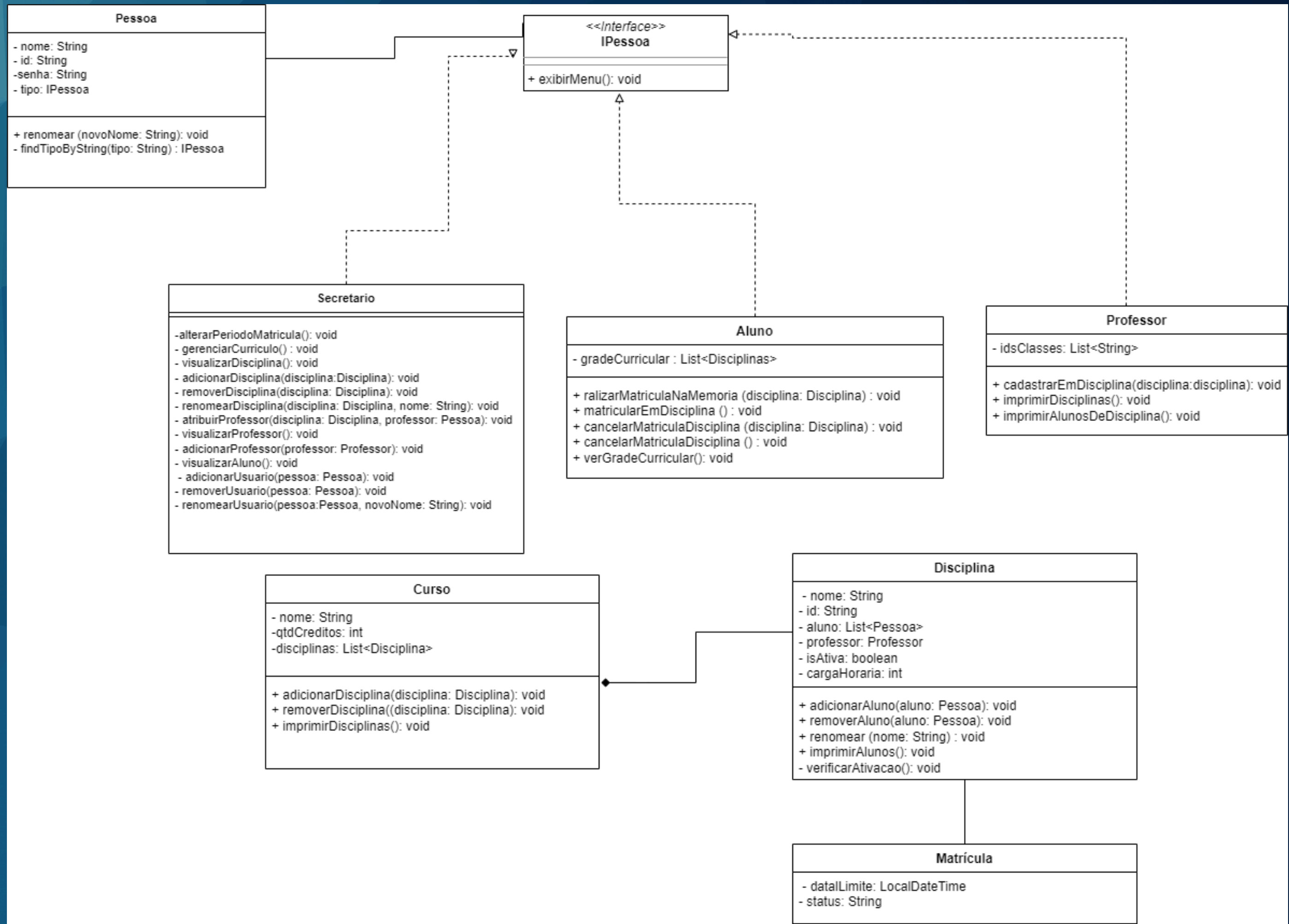
---

## LABORATÓRIO DE DESENVOLVIMENTO DE SOFTWARE

Engenharia de Software  
PUC MINAS

Ana Flávia de Carvalho Santos  
Estevão de Faria Rodrigues  
Lorrayne Marayze Silva de Oliveira

# DIAGRAMA DE CLASSE





UC1. Como secretária, quero gerar o currículo para disponibilizar para os alunos.

UC2. Como secretário, quero poder abrir e fechar o período de matrículas.

UC3. Como secretária, quero atualizar as informações sobre disciplinas, alunos e professores para manter os dados em dia.

UC4. Como aluno, quero realizar minha matrícula para me regularizar no curso.

UC5. Como aluno, quero realizar a matrícula em matérias obrigatórias para conseguir completar minha matrícula geral.

UC6. Como aluno, quero realizar a matrícula em matérias optativas para cumprir as horas complementares do curso.

UC7. Como aluno, quero consultar o número de créditos de uma disciplina para gerenciar quantos créditos preciso para me formar.

UC8. Como aluno, quero cancelar a matrícula para não precisar cursar mais uma disciplina.

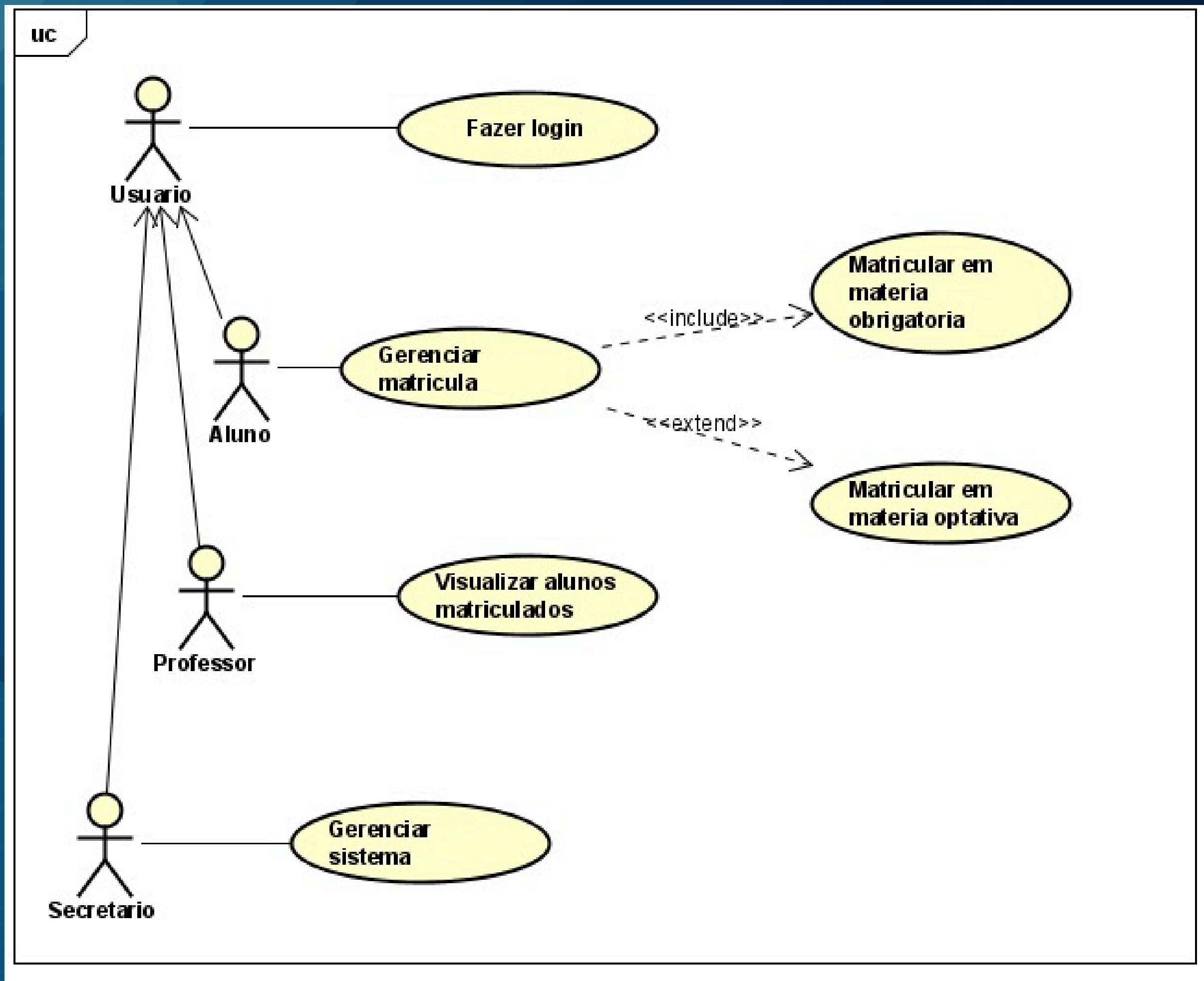
UC9. Como professor, quero visualizar os alunos matriculados para planejar minhas aulas.

UC10. Como professor, quero visualizar minhas turmas para gerenciar meus horários.

UC11. Como usuário, quero fazer login para acessar o sistema.

UC12. Como aluno, quero ser notificado sobre a cobrança das minhas disciplinas.

# DIAGRAMA DE CASO DE USO





A arquitetura MVC (Model-View-Controller) é um padrão de design de software que separa uma aplicação em três componentes principais:

1. **Model (Modelo):** Responsável pela lógica de dados da aplicação. Ele gerencia os dados, as regras de negócio, e a lógica de acesso e armazenamento. O Model notifica a View sobre mudanças nos dados.
2. **View (Visão):** Encapsula a apresentação dos dados. Ela é responsável por exibir a interface de usuário e os dados fornecidos pelo Model. A View não contém lógica de negócios, apenas a lógica para renderizar a interface.
3. **Controller (Controlador):** Atua como intermediário entre o Model e a View. Ele recebe entradas do usuário (por exemplo, cliques de botões), processa essas entradas (geralmente envolvendo o Model), e retorna a resposta apropriada à View para que ela seja atualizada.

# OBRIGADA!

---

Ana Flávia de Carvalho Santos  
Estevão de Faria Rodrigues  
Lorrayne Marayze Silva de Oliveira