

CURSO:	Análise e Desenvolvimento de Sistemas Back-End	Avaliação Formadora (Competência 3)
Compreender modelagem orientada a objetos com diagramas UML, desenvolver análise crítica, trabalhar em equipe, implementar classes em MVC e fortalecer pensamento analítico e argumentação técnica.		

Implementação no MVC

Objetivo

Praticar a implementação da camada Model no padrão MVC, integrando a teoria de modelagem com a prática de desenvolvimento de software, e aprimorar a compreensão da arquitetura em camadas.

Tipo de Atividade

- Trabalho grupo
- Avaliação: Apresentação do desenvolvimento e funcionamento em video
- **Entrega: 09 de novembro de 2025, as 23:59, na Plataforma AVA**
- Valor total: 15,0 pontos

Como funcionará a atividade

Abaixo, será apresentado um diagrama de classes a ser utilizado para implementação. A estrutura mínima de front-end, back-end e a arquitetura MVC já estão disponíveis em um repositório público no GitHub.

O aluno deverá implementar a(s) classe(s) Model correspondentes ao diagrama de classes proposto. Classes: **Livro, Venda, ItemLivroVenda, Cliente, ClienteFisico e ClienteJuridico**

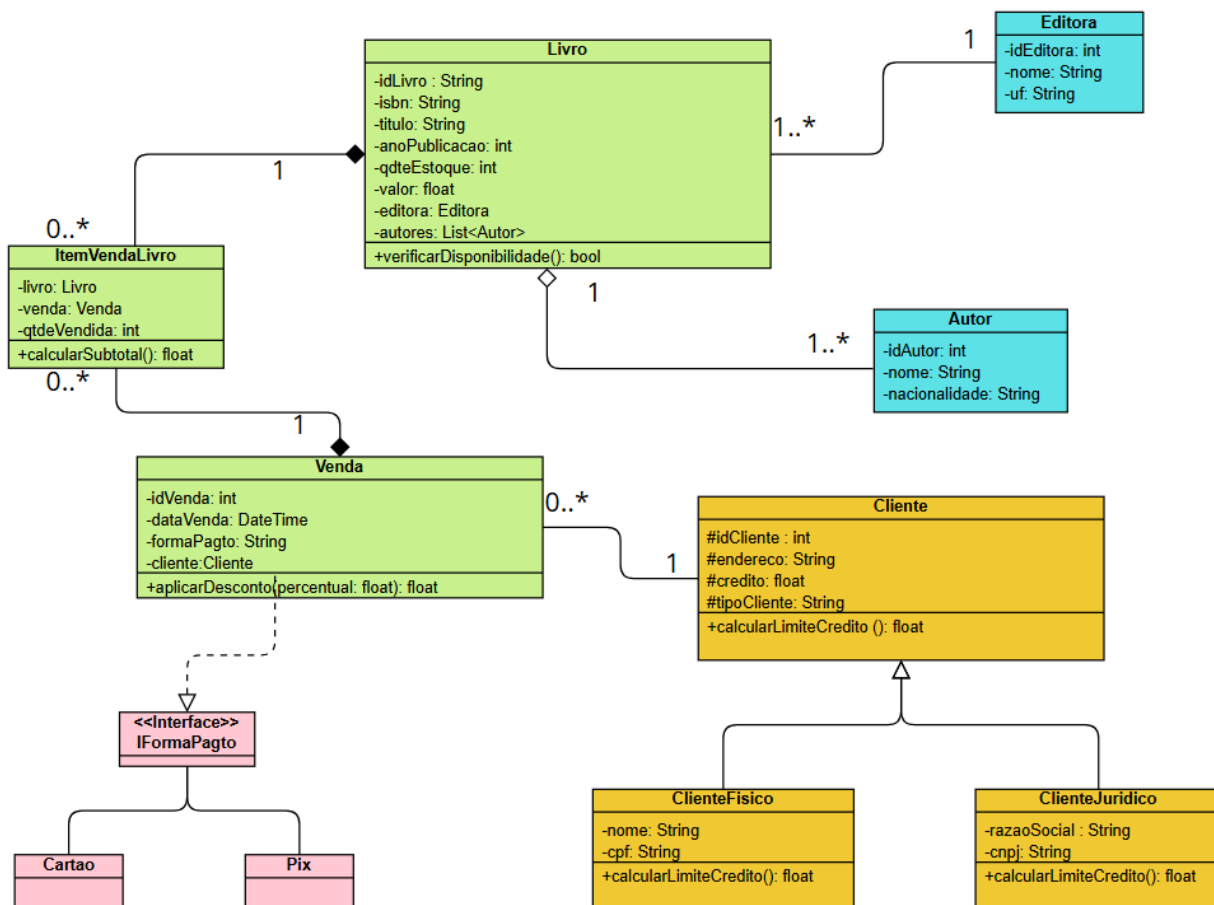
Cada classe deve conter:

- ✓ Todos os atributos indicados no diagrama;
- ✓ Os métodos de acesso (getters e setters) para cada atributo;
- ✓ Um método toArray(), responsável por converter o objeto em um array associativo com todos os seus atributos.

Instruções

1. Baixar do repositório público no GitHub a estrutura mínima do projeto, incluindo front-end e back-end.
2. A implementação deverá integrar-se corretamente ao sistema já fornecido.
3. O aluno deverá apresentar o código desenvolvido, realizar testes e demonstrar seu funcionamento ao professor por meio de um vídeo, que será disponibilizado na Plataforma AVA através de um link.
4. Todos os alunos deverão postar o link na Plataforma, mesmo se o trabalho for em grupo.

DIAGRAMA DE CLASSE PROPOSTO



☒ Critérios de Avaliação (15,0 pontos)

Critério	Excelente (5 pts)	Parcial (3 pts)	Insuficiente (0 pt)
Estrutura da Classe	Todas as 6 classes implementadas corretamente.	Classes incompletas ou com erros em atributos / relacionamentos.	Classes muito incompletas ou incorretas.
Métodos de Acesso (Getters e Setters)	Atributos com getters e setters corretos e nomes claros.	Atributos com getters/setters ausentes ou nomes inconsistentes.	Atributos sem métodos de acesso.
Método toArray()	Cada classe possui toArray() corretamente implementado.	toArray() incompleto ou converte objetos incorretamente.	toArray() ausente ou errado.
Clareza e Organização do Código	Código organizado, legível e com boas práticas.	Código pouco legível ou desorganizado.	Código desorganizado ou confuso.

Template para Download

<https://github.com/prof-kellenery/AvaliacaoFormadora3>