

**INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**  
RIO GRANDE DO NORTE



# Programação Orientada a Objetos

**UML**

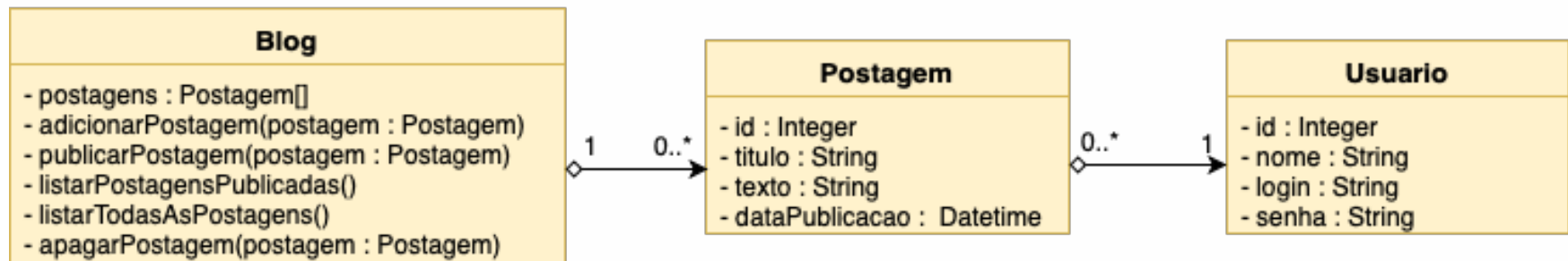
**PROFESSOR:**

Demetrios Coutinho

[Demetrios.coutinho@ifrn.edu.br](mailto:Demetrios.coutinho@ifrn.edu.br)

# Exercícios de UML

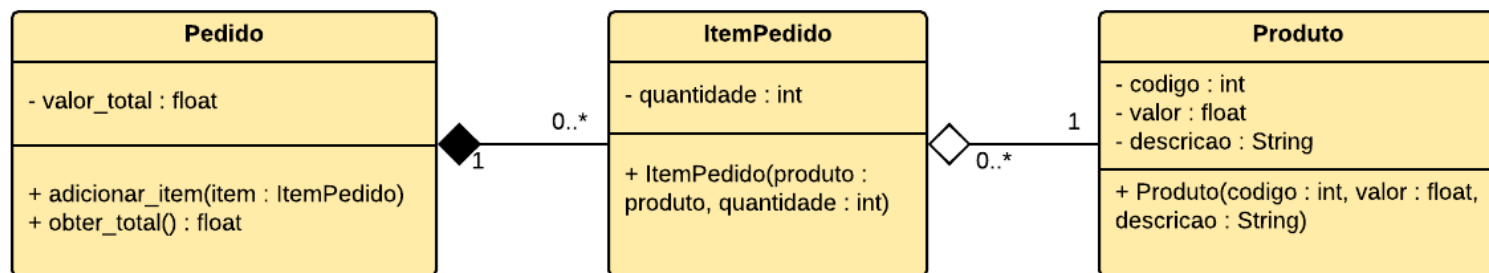
1. Crie um programa para representa um blog. O blog contém várias postagens que são cadastradas por usuários autenticados. Siga o diagrama de classes abaixo para a implementação:



- O blog contém uma lista de postagens.
- O método `listarPostagensPublicadas()` deverá listar apenas as postagens cuja data de publicação já foi atingida. Do contrário, as postagens serão visíveis apenas pelo método `listarTodasAsPostagens()`.
- Para criar uma postagem, o usuário deverá estar autenticado no sistema.
- A Postagem deverá identificar o usuário que fez a sua criação.

# Exercícios de UML

2. Implemente as classes descritas no diagrama abaixo:



- Um pedido é composto por um conjunto de itens pedidos.
- Um item pedido associa-se com um produto.
- O cálculo do valor total do pedido deverá ser feito mediante a soma do preço de cada produto incluído nos itens pedidos.

# Estudo de Caso

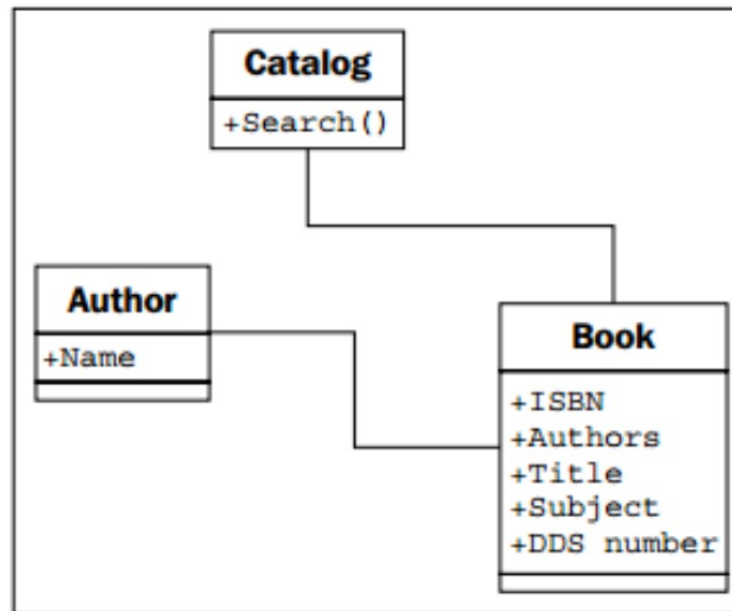
- Segue um exemplo de como modelar o sistema:

# Estudo de Caso

- Agora vamos modelar um caso mais prático: um sistema de catálogo de livrarias;
- Antes de falarmos com o cliente, o que sabemos sobre catálogos de livrarias:
  - Catálogos contém listas de livros;
  - Pessoas procurarm por livros de certos assuntos, títulos específicos ou por autores;
  - Livros são unicamente identificados por um *International Standard Book Number* (ISBN);
  - Cada livro tem um número de *Dewey Decimal System* (DDS) atribuído para ajudar a encontrá-lo em uma prateleira particular.
- A partir disso, podemos notar que Book será nosso principal objeto com muitos atributos como autor, título, assunto, ISBN, número DDS e catálogo;
- Observe que Autor será também um objeto, uma vez que um autor possui atributos como nome, filiação e etc.
  - A relação de livro e autor é uma associação, já que um livro TEM UM autor(es).
- Precisamos de uma classe *User*?
  - Apriori, não! Só precisamos de um catálogo e não precisamos rastrear os dados de usuários;

# Estudo de Caso

- Em relação ao catálogo?
  - Precisaremos de um método de busca;
    - A busca poderá ser realizada por autor, título e assunto;
  - Bem, já podemos criar um esquema para mostrá-lo ao bibliotecário:

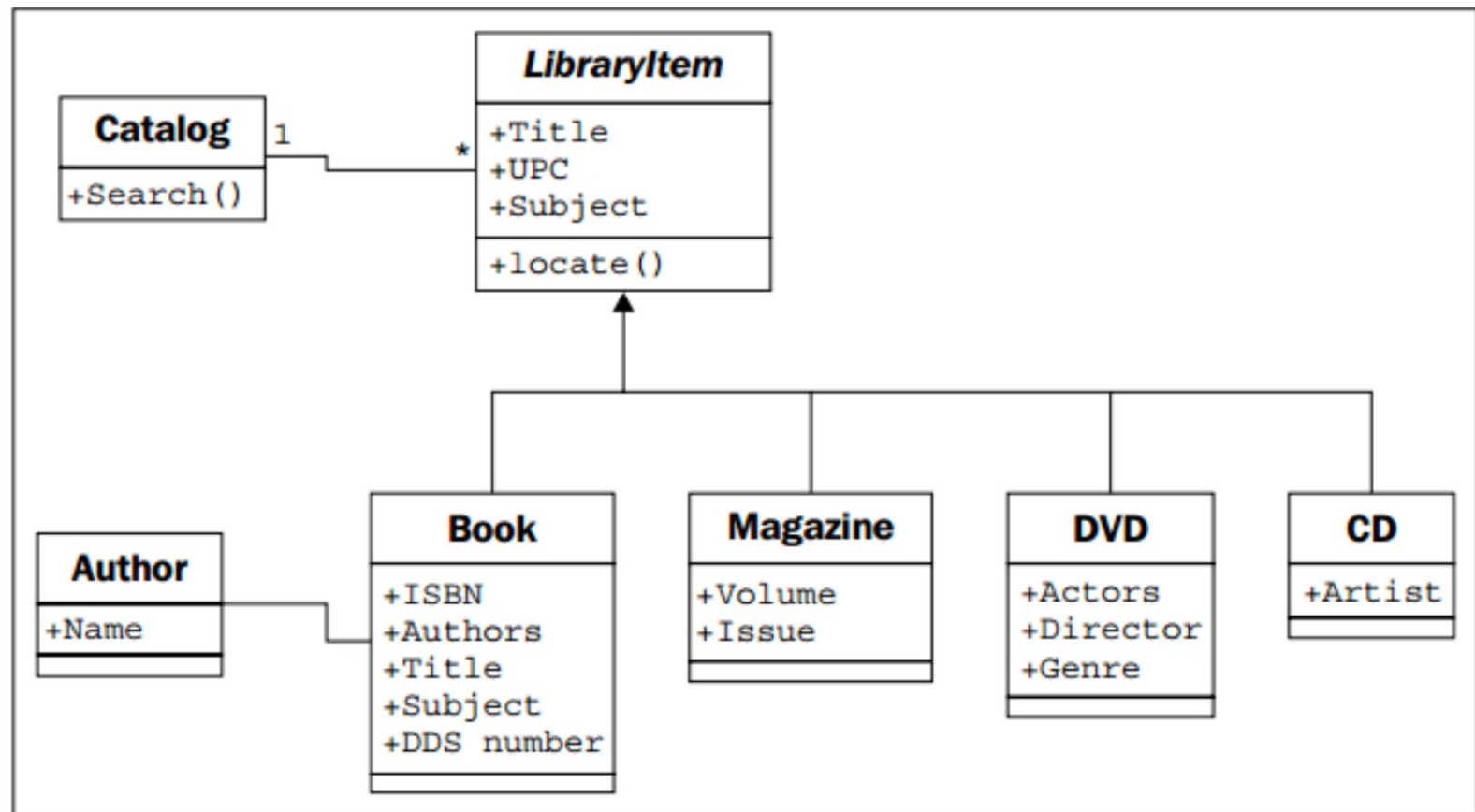


# Estudo de Caso

- Ao apresentar o simples esquema ao bibliotecário, este informou que livrarias não contêm somente livros, mas DVDs, revistas e CDs.
  - Nenhum desses possuem ISBN ou número DDS.
  - Como só existem poucos CDs, eles são organizados pelo último nome do autor;
  - Os DVDs são organizados por gênero e título;
  - Revistas são organizadas por título, volume e issue number;
  - Como dito anteriormente, os livros são organizados pelo DDS.

# Estudo de Caso

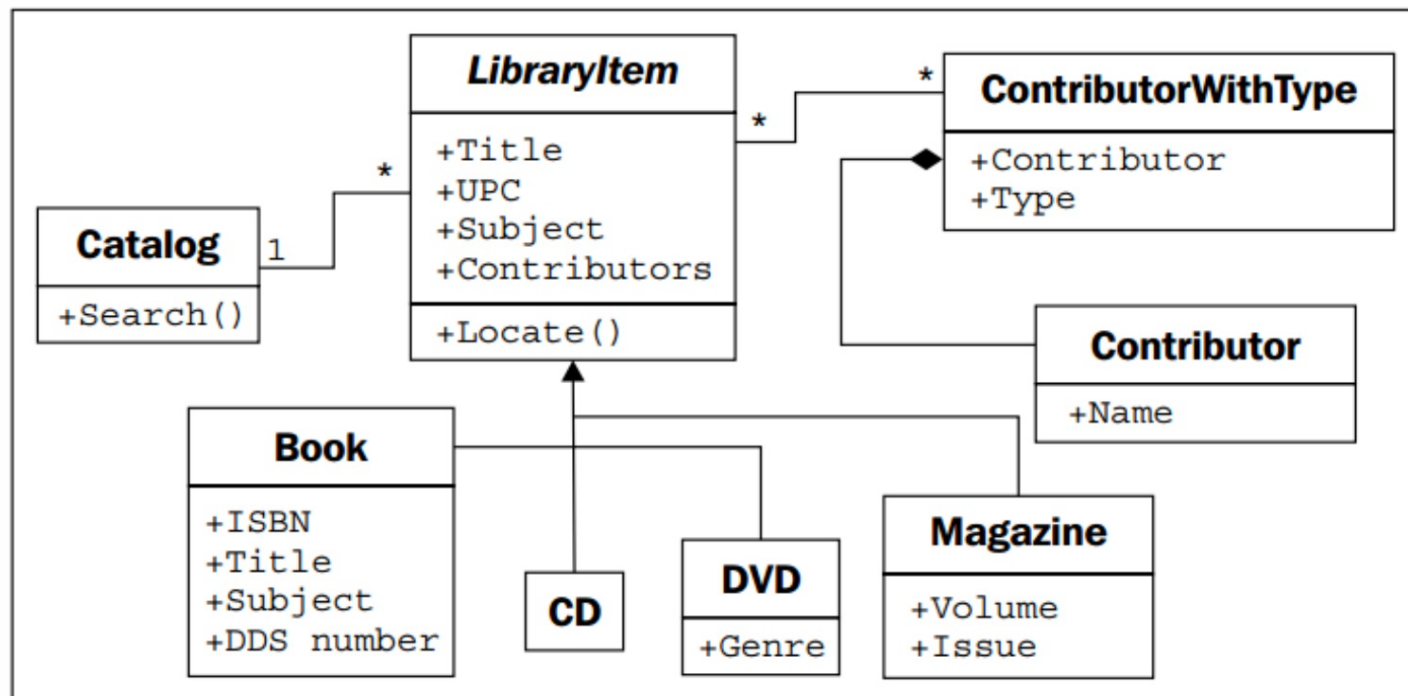
- Como todos os itens da livreria tem características em comum, podemos usar herança e definir nas subclasses as especificidades:





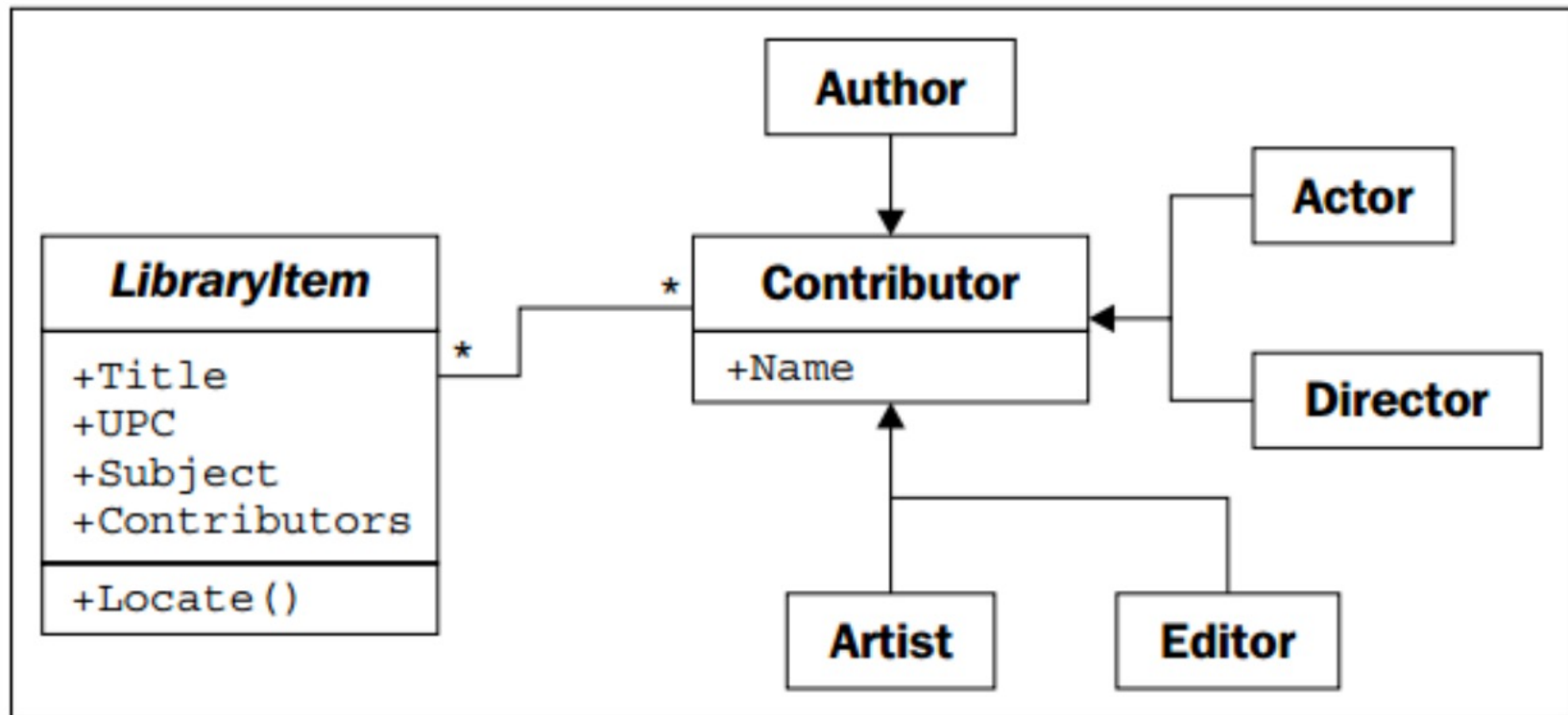
# Estudo de Caso

- O usuário faz uma requisição de busca ao catálogo. O catálogo procura na sua lista interna de itens e encontra um livro e um DVD onde aparece no título "bunnies". São retornados para o usuário a localização de ambos os itens.
- Várias pessoas podem contribuir com um item da livraria:
  - Para CDs, tem-se artistas, enquanto para livros temos os escritores responsáveis por contribuição.



# Estudo de Caso

Mais especificamente, ContributorWithType podem ser as várias classes de contribuidores:



# Empresa Aérea

3. Faça a modelagem em UML de um sistema de reserva para uma empresa aérea.

- cada voo terá várias reservas associadas a eles;
- cada voo pode informar o número de assentos livres e sua tripulação (conjunto de aeroviários);
- os voos podem ser internacionais ou nacionais;
- a um dado voo são atribuídos um código, um horário, uma data, um aeroporto de partida e um de destino;
- um aeroporto possui uma cidade onde se localiza, uma capacidade de decolagens por hora, e um nome;
- uma reserva possui um passageiro;
- operadores são funcionários da empresa responsáveis pela operacionalização das reservas; eles podem criar e cancelar reservas;
- os passageiros podem criar, cancelar e pagar reservas de voos.

# Empresa Aérea

4. Implemente a empresa Aérea com interface de interação com o usuário.

# DÚVIDAS?

