Exercício 5 – Biblioteca (identificação de objetos e atributos)

Construa o diagrama de classes para o problema que a seguir se apresenta.

Pretende-se um sistema de gestão de uma biblioteca.

A biblioteca dispõe apenas de livros, que podem ser requisitados por utentes.

Um livro é caracterizado por um código, título e autor.

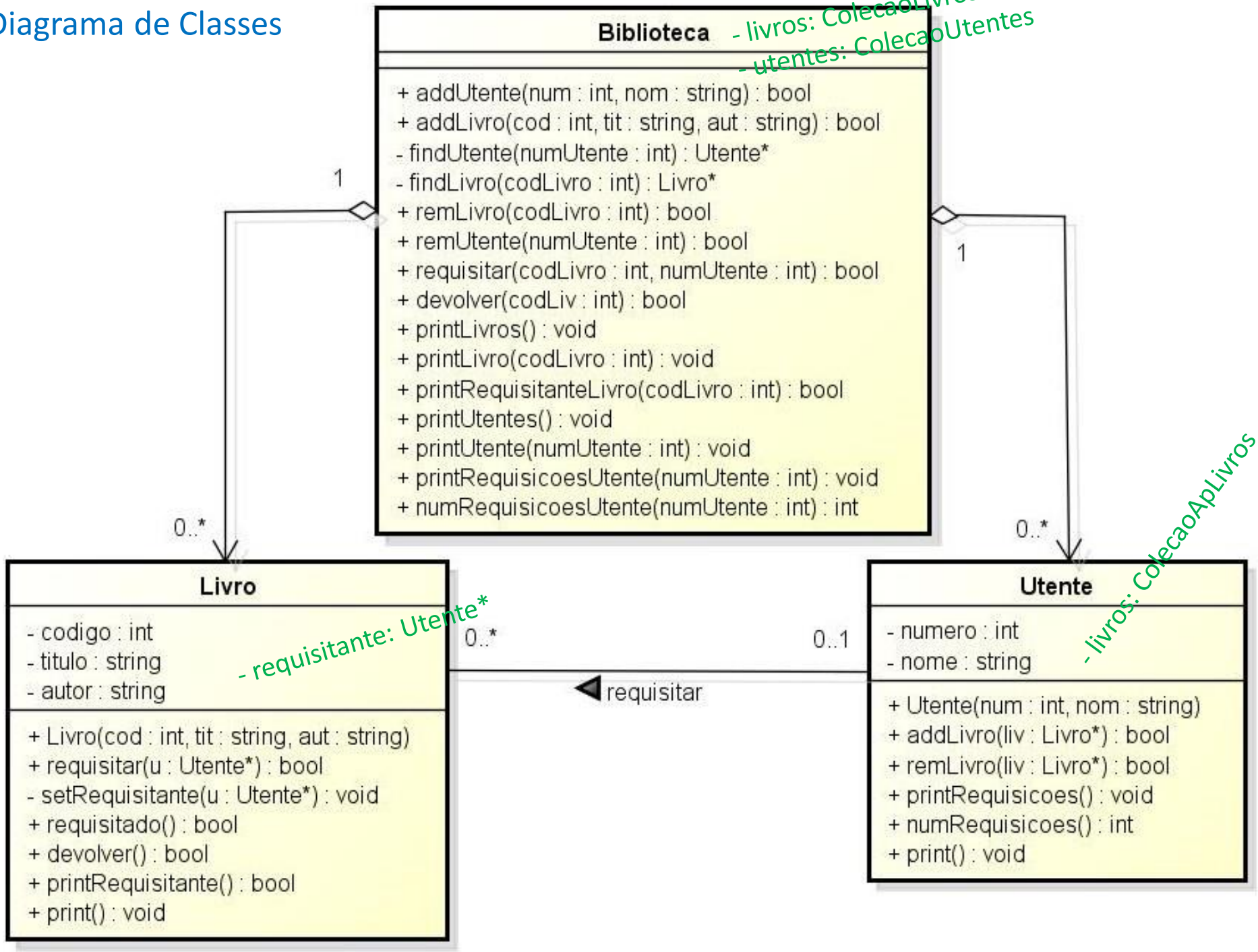
Um utente é caracterizado pelo número de utente e pelo nome.

Para não complicarmos demasiado a solução, vamos, por agora, introduzir duas importantes simplificações no problema: a biblioteca apenas dispõe de exemplares únicos e a aplicação não vai precisar de memorizar as requisições passadas (já devolvidas) – a aplicação apenas reportará o estado atual das requisições.

Um utente pode realizar diversas requisições, sendo apenas possível de concretizar se o exemplar a requisitar se encontrar disponível. Quando um exemplar é devolvido deve ficar livre para futuras requisições.

O sistema deve permitir o registo de novos utentes e a aquisição de novos livros, devendo ainda possibilitar a realização de consultas, nomeadamente: listagem de todos os livros, com indicação do código, título, autor e se o mesmo se encontra requisitado; listagem de um ou de todos os utentes, com indicação do número, nome e número de requisições do utente; e a consulta de todas a requisições de um dado utente.

Diagrama de Classes



Considerações de ordem geral

- Sempre que temos uma associação de um para muitos, isso significa que na implementação vamos ter uma colecção – cada objecto da 1ª classe vai conter uma colecção de objectos da 2ª classe

- Quando se trata de um problema de gestão de uma ou mais coleções de entidades, é habitual considerar-se uma classe principal que designamos centralizadora, a qual conterá as colecções principais do problema – neste caso, será a classe Biblioteca

- A interface da classe centralizadora (métodos públicos) deverá reflectir as várias funcionalidades da aplicação

- De certa forma, a interface da classe centralizadora constitui a interface (API) da própria aplicação – a ideia é o utilizador só interagir com essa classe

- Os métodos da classe centralizadora devem conter apenas parâmetros de tipo predefinido – não devem ser objectos de classes específicas do problema, por forma a facilitarmos a vida ao utilizador

- Por norma, cada funcionalidade tem início na invocação dum método da classe centralizadora, mas será o objecto a que se destina a acção a realizar a respectiva tarefa; em princípio, a classe centralizadora limitar-se-á a procurar o objecto coleccionado a quem deverá delegar a implementação de tarefa; podendo a tarefa ser delegada sucessivas vezes até chegar ao objecto final

- Numa associação de um para muitos (colecção), a classe agregadora deve conter as funcionalidades básicas de gestão da respectiva colecção – métodos add(), find() e rem()

- Numa associação de um para muitos, dá jeito que tb a classe colecionada veja a classe agregadora (navegabilidade bidireccional) – cada objecto coleccionado deve apontar para o objecto agregador