

Projeto Prático – 2024/2025 Sistema de Gestão de Bibliotecas

Descrição Geral

O objetivo deste projeto é criar um sistema de gestão de uma biblioteca que utilize extensivamente os conceitos de herança e polimorfismo para modelar diversos tipos de livros, leitores e funcionalidades. O sistema permitirá o registo de livros e leitores e respetiva atualização/manutenção, realizar e gerir empréstimos (requisições de livros por parte dos leitores), e controlar outros aspetos como multas e reservas de livros.

Funcionalidades Requeridas

1. Registo de Livros (com Hierarquia):

- Diferentes tipos de livros devem ser registados, com informações específicas dependendo do tipo. Os tipos de livros incluem:
 - LivroFiccao
 - LivroCientifico
 - LivroEducativo
 - Revista
 - Jornal
- Cada subclasse de livro deve ter atributos e métodos específicos (ex.: LivroEducativo tem um "grau de escolaridade", Revista tem "número de edição").

2. Inscrição de Leitores (com Hierarquia):

- Os leitores podem ser divididos em categorias diferentes, como:
 - LeitorComum
 - Estudante
 - Professor
 - Senior
- Cada tipo de leitor terá características diferentes (por exemplo, o número máximo de livros que pode requisitar ou se pode receber descontos em multas).

3. Realização de Empréstimos (Polimorfismo):

- O processo de empréstimo deve adaptar-se ao tipo de leitor (usando polimorfismo). Por exemplo:
 - Estudante pode ter mais tempo para devolver livros educativos.
 - Professor pode requisitar livros científicos por mais tempo.
 - LeitorComum tem prazos normais para todos os tipos de livros.
- A classe Empréstimo será responsável por registar as informações do empréstimo, e a data de devolução será ajustada de acordo com o tipo de livro e o tipo de leitor.

4. Devolução e Controlo de Multas:

- Os livros têm um tempo máximo que podem estar na posse de um leitor (que depende do tipo de livro e do tipo de leitor). O não cumprimento deste prazo implica a aplicação de multas, cujo valor varia de acordo com o tipo de leitor. Estudantes e professores podem ter descontos em multas, enquanto leitores comuns pagam o valor total. Isso deve ser implementado usando o polimorfismo.

5. Sistema de Reservas (Polimorfismo):

- Os leitores podem reservar livros que estão atualmente emprestados a outro leitor. Cada tipo de leitor pode ter regras diferentes para o número de reservas que pode

fazer e a prioridade na fila de reservas (por exemplo, `Estudante` pode ter prioridade na reserva de livros educativos).

6. Pesquisa e Listagem de Livros (Herança e Polimorfismo):

- O sistema deve permitir a pesquisa de livros por tipo (polimorfismo), e a listagem pode ser filtrada por categorias, como ficção, revistas ou científicos.

7. Relatório de Empréstimos por Tipo de Livro:

- Implementar um relatório que mostre os empréstimos categorizados por tipo de livro (livros ficcionais, científicos, etc.) e o tipo de leitor que mais os requisita.

8. Histórico de Empréstimos de Leitores:

- Cada tipo de leitor terá seu próprio histórico de empréstimos, e o sistema deve exibir o histórico de empréstimos de cada leitor, categorizando os livros por tipo.

9. Relatório de Multas Pendentes:

- Gerar um relatório que exibe leitores com multas pendentes, mostrando os valores por tipo de leitor (polimorfismo aplicado no cálculo de multas).

10. Edição de Informações de Leitores:

- Implementar a funcionalidade de edição de informações de leitores, onde se pode atualizar dados específicos de cada subclasse (ex.: descontos em multas para professores ou número máximo de livros para estudantes).

11. Edição de Informações de Livros:

- Permitir a atualização de informações de livros, com atributos específicos de cada tipo (por exemplo, a categoria de um livro educativo ou a edição de uma revista).

12. Sistema de Notificações de Atraso:

- Os leitores receberão notificações de atraso personalizadas de acordo com seu tipo (herança). Por exemplo, um professor pode receber uma notificação com lembrete sobre as vantagens de prorrogação, enquanto um leitor comum receberá uma notificação direta sobre a multa.

13. Prorrogação de Empréstimos:

- Implementar um sistema onde apenas certos tipos de leitores (como professores ou estudantes) podem prorrogar os seus empréstimos, enquanto leitores comuns não podem.

14. Relatório de Livros por Categoria:

- Implementar relatórios que exibem os livros agrupados por suas subclasses (ex.: listar todos os livros científicos disponíveis ou todos os jornais registados).

15. Gravação/Recuperação de toda a informação referente à biblioteca:

- Gravar num dado ficheiro (introduzido pelo utilizador)
- Ler de um ficheiro (introduzido pelo utilizador)

16. Interface Gráfica:

- Os alunos devem desenvolver uma interface gráfica que utilize menus para navegar entre as funcionalidades.

Requisitos Técnicos

• **Herança e Polimorfismo:**

- As classes devem ser projetadas de modo que a herança e o polimorfismo sejam explorados intensivamente. A classe base para os livros será `Livro` e, a partir dela, serão criadas subclasses como `LivroFiccao`, `LivroCientifico`, `Revista`, etc.
- A classe `Leitor` será a classe base para diferentes tipos de leitores, e cada subclasse (como `Estudante`, `Professor`, `Senior`) terá seu próprio comportamento para empréstimos, multas e prorrogações.

• **Encapsulamento:**

- As classes devem implementar encapsulamento, com os atributos privados e métodos públicos para manipulação de dados.

- **Classes e Arquitetura:**

- Organizar o código em múltiplos ficheiros, separando as classes em módulos distintos para facilitar a manutenção e expansão do sistema.

Sugestão de Classes

1. Classes Base:

- Livro (Classe base para todos os tipos de livros)
- Leitor (Classe base para os diferentes tipos de leitores)
- Empréstimo (Classe responsável por registar e gerir os empréstimos)

2. Subclasses de Livros:

- LivroFiccao
- LivroCientifico
- LivroEducativo
- Revista
- Jornal

3. Subclasses de Leitores:

- LeitorComum
- Estudante
- Professor
- Senior

4. Outras Classes:

- Biblioteca (Gere toda a interação entre livros, leitores e empréstimos)
- Multa (Classe para cálculo de multas)

Observações:

- Podem e devem criar novos métodos auxiliares (privados) com os nomes que acharem por bem! E chamar nos métodos principais (públicos);
- Sempre na metodologia orientada a objetos;

Avaliação / Observações:

- Se a gestão da memória, não estiver correta, haverá uma penalização de 4 valores;
- Se for detetado "copiar", trabalho anulado!
- Soluções do problema que não usem os conceitos explorados na UC de Programação Orientada a Objetos, ainda que funcionais, não serão consideradas;
- Os grupos são constituídos por até 4 alunos;
- O código deve estar comentado;
- O código deve ser documentado (através de uma aplicação tipo **doxygen** ou elaborando um relatório em que se explicita a hierarquia de classes e se expliquem as funcionalidades implementadas).
- Eventuais dúvidas serão esclarecidas pelos docentes da disciplina.

Entrega:

- Para avaliação em **época normal**, a data limite para entrega do projeto no moodle é **15/12/2024** (a defesa/avaliação do trabalho decorrerá no período de compensação);
- É ainda permitida a entrega para avaliação em **época recurso**, mas com a inclusão de funcionalidades adicionais, posteriormente anexadas a este enunciado. Neste caso, a data limite para entrega do projeto no moodle é **02/02/2025** (a avaliação será em data a definir).