

SISTEMA DE GERENCIAMENTO DE BIBLIOTECA

1. INTRODUÇÃO

O sistema de gerenciamento de biblioteca apresentado no código é estruturado em várias classes que interagem de maneira coesa, seguindo princípios da programação orientada a objetos, como encapsulamento, herança e abstração. A abordagem adotada visa à organização e modularidade do código, facilitando a manutenção e a extensão futura do sistema.

2. CLASSE ITEM BIBLIOTECA

A classe **ItemBiblioteca** é definida como uma classe abstrata, servindo como base para outros itens, como livros. Essa classe possui duas propriedades fundamentais: **Título** e **Código**, que são essenciais para armazenar informações identificadoras do item. A definição de métodos abstratos, como **Emprestar** e **Devolver**, assegura que cada tipo de item tenha sua própria lógica para essas operações, permitindo que as subclasses implementem suas particularidades.

3. CLASSE LIVRO

A classe **Livro**, que herda de **ItemBiblioteca**, representa um livro específico dentro da biblioteca. Além das propriedades herdadas, que fornecem a identificação básica do livro, a classe contém informações adicionais relevantes, como **Autor**, **ISBN**, **Gênero** e **Quantidade em Estoque**. O construtor da classe inicializa todas essas propriedades e invoca o construtor da classe base, garantindo que todas as informações necessárias sejam definidas no momento da criação do objeto.

Os métodos **Emprestar** e **Devolver** são implementados para gerenciar a disponibilidade do livro, permitindo que um usuário o empreste e o devolva. Esses métodos também atualizam a quantidade em estoque conforme necessário, assegurando que a quantidade correta de cópias disponíveis seja sempre refletida.

4. CLASSE USUARIO

A classe **Usuario** desempenha um papel vital ao representar os usuários da biblioteca. Esta classe armazena informações importantes, incluindo **Nome**, **Número de Identificação**, **Endereço** e **Contato**, além de uma lista privada chamada **Histórico de Empréstimos**, que registra todos os empréstimos realizados pelo usuário. O construtor da classe inicializa as propriedades do usuário e cria uma nova lista para o histórico, preparando o sistema para registrar qualquer operação de empréstimo.

Os métodos disponíveis na classe permitem exibir informações detalhadas sobre o usuário, consultar seu histórico de empréstimos e adicionar novos registros a esse

histórico, proporcionando uma visão completa da interação do usuário com a biblioteca.

5. CLASSE BIBLIOTECA

A classe **Biblioteca** desempenha um papel central na gestão dos livros e usuários, atuando como o controlador principal do sistema. Ela possui listas privadas que armazenam objetos do tipo **Livro** e **Usuario**, o que permite um gerenciamento eficiente e organizado dos recursos da biblioteca.

Os métodos da classe incluem funcionalidades para cadastrar novos livros e usuários, listar todos os registros existentes e gerenciar as operações de empréstimo e devolução de livros. O método de empréstimo verifica cuidadosamente a existência do livro e do usuário antes de permitir a operação, enquanto o método de devolução confirma se o livro estava realmente emprestado. Essa verificação é fundamental para garantir a integridade do sistema e prevenir erros, como a devolução de um livro que não estava emprestado.

6. PESQUISA DE LIVROS E USUÁRIOS

A implementação de métodos para pesquisa de livros e usuários proporciona uma experiência mais dinâmica e interativa. Os usuários podem buscar livros por título, autor ou gênero, facilitando o acesso à informação desejada. Essa funcionalidade é especialmente importante em bibliotecas com um acervo extenso, onde a localização de um título específico pode ser um desafio.

7. CÓDIGOS PESQUISADOS

Para a pesquisa de livros e usuários, foram utilizados alguns códigos que auxiliam na busca eficiente dos dados armazenados:

Where: Esse método permite a filtragem de coleções, como listas de livros ou usuários. Ele é usado para encontrar itens que atendem a uma condição específica. Por exemplo, pode-se buscar um livro pelo título ou autor.

StringComparison.OrdinalIgnoreCase: Essa opção de comparação de strings permite a busca de títulos e autores de maneira com que seja desconsiderado a diferença entre maiúsculas e minúsculas. Tornando a busca mais fácil para o usuário e para o programador, sem ter que exigir uma correspondência exata no uso de letras maiúsculas e minúsculas.

FirstOrDefault: Esse método retorna o primeiro item que satisfaz uma condição especificada, ou o valor padrão (null) se nenhum item for encontrado. É útil em

buscas que esperam retornar um único resultado, como a busca por um livro com um título específico.

8. CONCLUSÃO SOBRE O APRENDIZADO

Em resumo foi adquirido mais conhecimentos sobre conteúdos utilizados em aula e sobre os pilares da Programação Orientada a Objetos, além de novos comandos para realizar certas funções.

9. CONCLUSÃO

Em suma, o sistema de gerenciamento de biblioteca implementa uma estrutura organizada e modular, seguindo normas da programação orientada a objetos. Essa abordagem não só facilita a manutenção do código, mas também permite que novas funcionalidades sejam incorporadas de maneira ágil e sem complicações. Cada componente do sistema tem responsabilidades bem definidas e interage de forma coesa, criando um ambiente robusto e eficiente para o gerenciamento de livros e usuários. O sistema também pode ser expandido para incluir funcionalidades adicionais, como notificações de empréstimos e devoluções, integração com sistemas externos ou a implementação de uma interface gráfica para facilitar a interação dos usuários.

Se houver interesse em discutir aspectos específicos do código, suas aplicações ou explorar possíveis melhorias e extensões do sistema, estou à disposição para esclarecer e aprofundar a discussão.

10. REFERÊNCIAS BIBLIOGRÁFICAS UTILIZADAS

1. O Alquimista de Paulo Coelho.
2. 1984 de George Orwell.
3. Harry Potter e a Pedra Filosofal de J.K. Rowling