

Desafio Semana 1

Atividade

1. Tire um tempo para refletir sobre tudo que vimos nesta aula que, apesar de teórica, é muito importante. Hoje em dia é muito fácil obter trechos de código usando ferramentas na internet mas, é preciso entender o porquê e se faz sentido. E como futura pessoa desenvolvedora, você é que vai fornecer essas respostas aos porquês.

Acredito que entender os conceitos básicos é importante para que a gente seja agnóstico com relação a linguagens , e tendo os fundamentos conseguiremos mais facilmente trabalhar com diferentes linguagens. Além disso, é importante entender sobre a complexidade dos algoritmos e recursos computacionais, pois assim seremos profissionais mais completas. Por último, sei que na maioria dos trabalhos teremos que diagnosticar bugs em códigos de outras programadoras, o que geralmente não é tarefa fácil, e para que possamos escrever código mais legível precisamos nos aperfeiçoar.

2. Imagine que você está desenvolvendo um sistema de gerenciamento de uma biblioteca. Neste sistema, é necessário modelar livros, autores, usuários, empréstimos e devoluções. Cada livro possui informações como título, autor e gênero. Os autores têm nome, nacionalidade e obras publicadas. Os usuários da biblioteca têm nome, idade e histórico de empréstimos. Os empréstimos registram a data de retirada e devolução, além do livro e usuário envolvidos.

Com base no cenário descrito, analise qual paradigma de programação seria mais indicado para implementar o sistema de gerenciamento da biblioteca:

- a) programação estruturada
- b) programação imperativa
- c) programação orientada a objetos ou
- d) programação funcional

Justifique sua escolha considerando a estrutura e as interações entre os objetos do sistema.

Para facilitar, considere pontos como:

- Estrutura e interação das entidades do sistema (Livros, autores, usuários e demais)
- Reuso
- Qual paradigma facilitaria a evolução do sistema, no futuro
- Eficiência, dado o cenário apresentado

Dica: A combinação dos paradigmas de programação pode ser viável em certos contextos, mas geralmente não é recomendada devido a diferenças fundamentais entre os paradigmas

Eu escolheria programação orientada a objetos, pois em primeiro lugar as entidades desse sistema têm relações complexas, os livros têm autores, os empréstimos envolvem livros e usuários, por exemplo, e programação imperativa por exemplo, não seria adequada para lidar com tantos detalhes.

Relativo ao reuso, a programação orientada a objetos ajuda na modularização e reutilização do código, pois facilmente separamos as classes e objetos que bem definidos podem ser reusados.

O paradigma funcional também não seria eficiente pois, por exemplo, a programação funcional depende da recursão em vez de loops imperativos, o que pode ser menos eficiente. Ainda, esse paradigma não é tão maduro em comparação com o paradigma orientado a objetos.

O paradigma estrutural também não seria o mais adequado, pois por exemplo é focado em fluxos de controle e não em estruturas de dados complexas, como é o nosso caso. Já a programação orientada a objetos oferece, por exemplo, o encapsulamento que permite proteger o estado interno, encapsulando dados e comportamentos dos objetos.

3. Você deve ter percebido que nossas aulas tem vários diagramas. Eles ajudam na correta compreensão dos conceitos e, quando projetamos sistemas reais, esses recursos nos ajudam a entender alguns detalhes de implementação e viabilidade do que estamos fazendo. Neste exercício, você vai criar seu primeiro diagrama!

Antes de começar a desenhar, tenha em mãos as seguintes respostas:

- Quais serão os elementos (classes) do diagrama?
- Com os conhecimentos que você já tem em programação, consegue pensar no tipo de cada propriedade? Por exemplo, no caso do livro, qual seria o tipo da propriedade "Título"? Seria uma string, um number, ou outro tipo?
- Qual seria o relacionamento entre os elementos? Pense em relacionamentos 1 para 1 e 1 para muitos

Regras:

- Seu diagrama deve estar de acordo com o que você definiu no exercício anterior, em Elementos de diagrama e Relações entre elementos

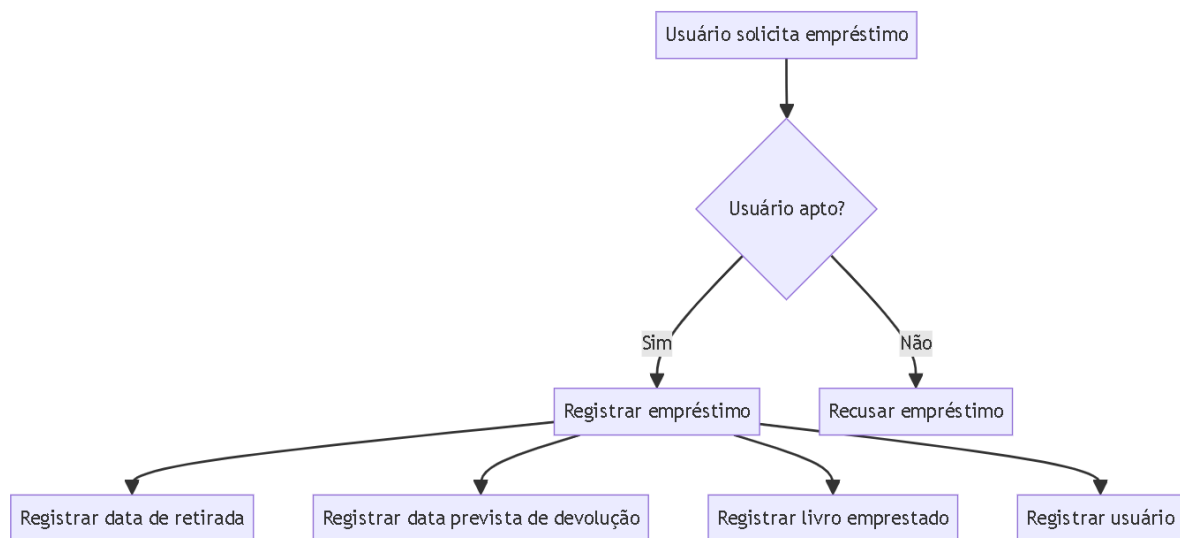
4. Agora, vamos acrescentar mais alguns detalhes ao seu diagrama! Já desenhamos o diagrama de classes, contendo propriedades e relacionamentos entre os objetos.

Agora, você vai receber os fluxos do nosso sistema de gerenciamento! Precisamos dessas informações para completar nosso exercício, que vai ser adicionar os métodos às nossas classes, de acordo com o que o sistema deve fazer. Para facilitar, vamos deixar os fluxos mais simples para o nosso projeto de aula.

Fluxo de Empréstimo de Livros

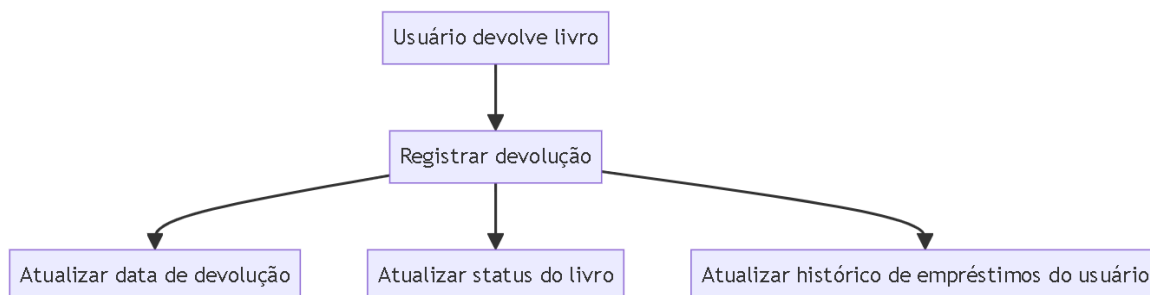
1. O usuário solicita o empréstimo de um ou mais livros.
2. O sistema verifica se o usuário está apto a realizar o empréstimo:

- a. Consulta o nome do usuário para identificá-lo
 - b. Verificar o histórico de empréstimos do usuário para ver se ele não possui livros vencidos.
 - c. Verificar a idade do usuário, caso o livro solicitado seja de um gênero impróprio para menores de 18 anos.
3. Se o usuário estiver apto, o sistema:
 - a. Registra a data de retirada do livro.
 - b. Registra a data prevista de devolução.
 - c. Associar o livro emprestado e o usuário que realizou o empréstimo.
4. Caso o usuário não esteja apto, o sistema recusa o empréstimo



Fluxo de Devolução de Livros

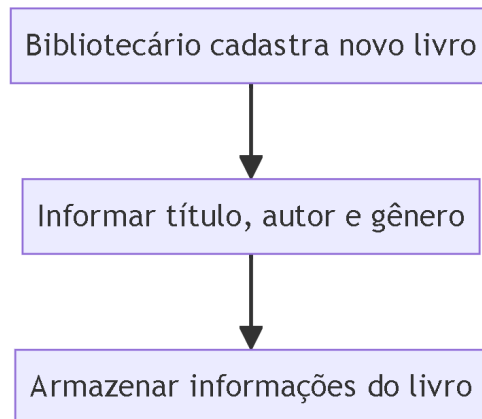
1. O usuário devolve um ou mais livros.
2. O sistema registra a devolução, atualizando:
 - a. A data de devolução do livro.
 - b. O status do livro como disponível na biblioteca.
 - c. O histórico de empréstimos do usuário.



Fluxo de Cadastro de Livros

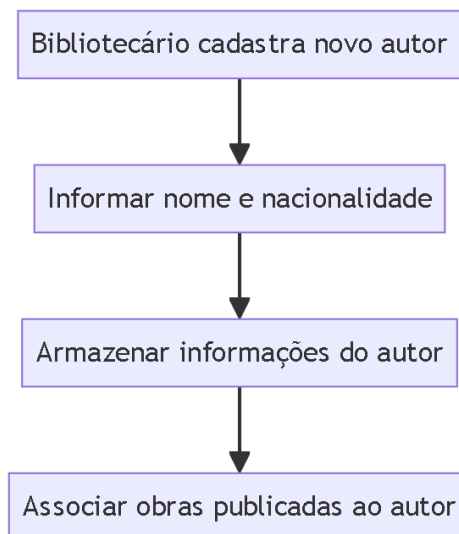
1. O bibliotecário cadastra um novo livro no sistema, informando:
 - a. O título do livro.
 - b. O autor do livro.
 - c. O gênero do livro.

2. O sistema armazena as informações do novo livro.



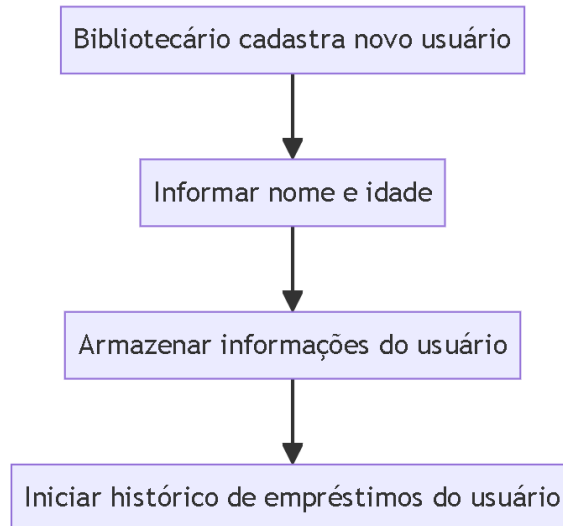
Fluxo de Cadastro de Autores

1. O bibliotecário cadastra um novo autor no sistema, informando:
 - a. O nome do autor.
 - b. A nacionalidade do autor.
2. O sistema armazena as informações do novo autor e associa as obras publicadas a ele.



Fluxo de Cadastro de Usuários

1. O bibliotecário cadastra um novo usuário no sistema, informando:
 - a. O nome do usuário.
 - b. A idade do usuário.
2. O sistema armazena as informações do novo usuário e inicia seu histórico de empréstimos.



Com base em todos os fluxos e regras de negócio, quais seriam os métodos que precisamos acrescentar em nossas classes?

Banco Comunitário

Como apresentado, o [Projeto Integrador](#) será um Banco Comunitário feito por vocês. Pensem qual comunidade vocês desejam atender e possíveis necessidades dessa comunidade.

Como funcionará o projeto e métodos de entrega

Você deverá criar de forma individual, o seu projeto no Github, seguindo o exemplo: <https://github.com/reprograma/ON36-IJS-SEU-BANCO-COMUNITARIO>

Crie a branch feature/semana-1 e coloque lá seu desenho do seu banco comunitário.