



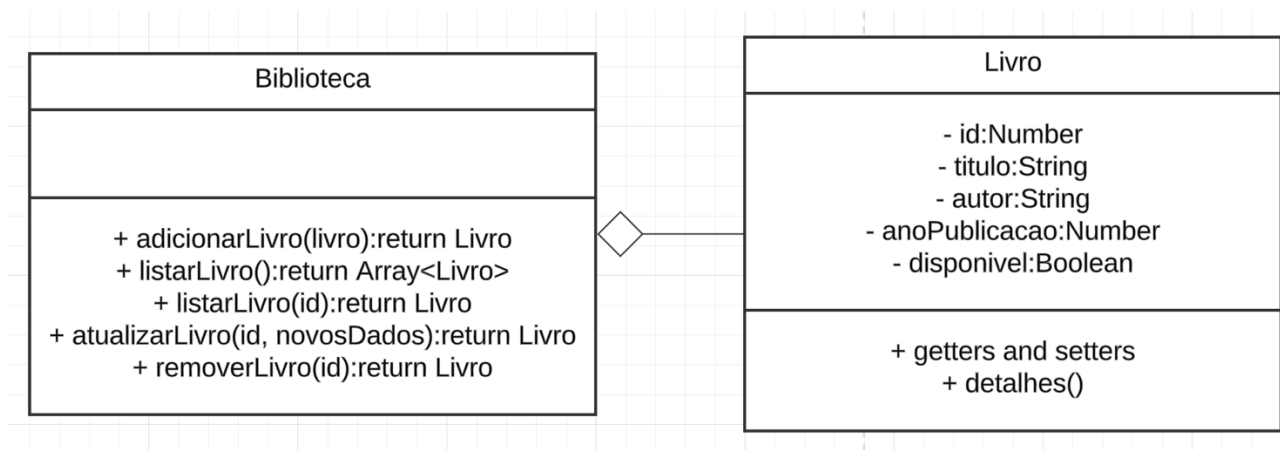
Laboratório de CRUD com Classes e Armazenamento no Web Storage

Nesta prática, o aluno terá uma introdução a conceitos práticos de **CRUD**, orientação a objetos, e gestão de dados em memória - habilidades essenciais para aplicações modernas.

Gerenciamento de Biblioteca

Desenvolveremos um sistema simples para gerenciamento de livros para uma biblioteca. O sistema deve permitir o **Cadastro, Leitura, Atualização e Exclusão** (CRUD) de informações sobre os livros utilizando **classes e objetos** em JavaScript.

Considere o diagrama de classes abaixo:



Cada livro deve conter as seguintes informações:

- **id**: um identificador único (gerado automaticamente pelo sistema).
- **titulo**: o título do livro.
- **autor**: o autor do livro.
- **anoPublicacao**: o ano de publicação do livro.
- **disponivel**: um booleano indicando se o livro está disponível ou não para empréstimo.

Além de realizar as operações de **Cadastro, Leitura, Atualização e Exclusão** (CRUD), a biblioteca deve persistir os dados dos livros utilizando o **Web Storage** do navegador (**localStorage** ou **sessionStorage**). Isso garante que os registros não sejam perdidos ao recarregar a página.

Tarefas

1. Criar a Classe **Livro**:

- Crie a classe **Livro** com os atributos descritos acima.
- Adicione um método `detalhes()` que retorna uma string formatada com todas as informações do livro.

2. Criar a Classe **Biblioteca**:

- A classe **Biblioteca** será responsável por gerenciar os livros.
- Inclua os seguintes métodos na classe:
 - `adicionarLivro(livro)`: adiciona um novo livro à coleção.
 - `listarLivros()`: exibe todos os livros cadastrados.
 - `atualizarLivro(id, novosDados)`: permite atualizar as informações de um livro pelo `id`.

- `removerLivro(id)`: remove um livro pelo `id`.

3. Simular o Uso do Sistema:

- Crie um menu (simulado no console ou utilizando prompts) que permita ao usuário executar as operações de CRUD na biblioteca.

4. Persistência de Dados:

- Os livros cadastrados devem ser salvos no **localStorage**.
- Ao iniciar o sistema, a biblioteca deve carregar os dados do **localStorage**.

5. Modificações na Classe Biblioteca:

- Adicione métodos para salvar e carregar dados do **localStorage**:
 - `salvarDados()`: armazena o array de livros no **localStorage**.
 - `carregarDados()`: recupera os dados salvos no **localStorage** e atualiza o array livros.

6. Simulação do Uso do Sistema:

- Teste as funcionalidades de CRUD e verifique se os dados persistem após recarregar a página.

Validando seu código:

1. Persistência de Dados:

- Certifique-se de que os livros persistem no **localStorage** e são carregados ao reiniciar a página.

2. Uso do **localStorage**:

- Verifique se as funções `salvarDados` e `carregarDados` estão funcionando corretamente.

3. CRUD Completo:

- Teste todas as operações: `adicionar`, `listar`, `atualizar` e `remover` livros.

4. Organização do Código:

- Estrutura clara e uso adequado de classes e métodos.