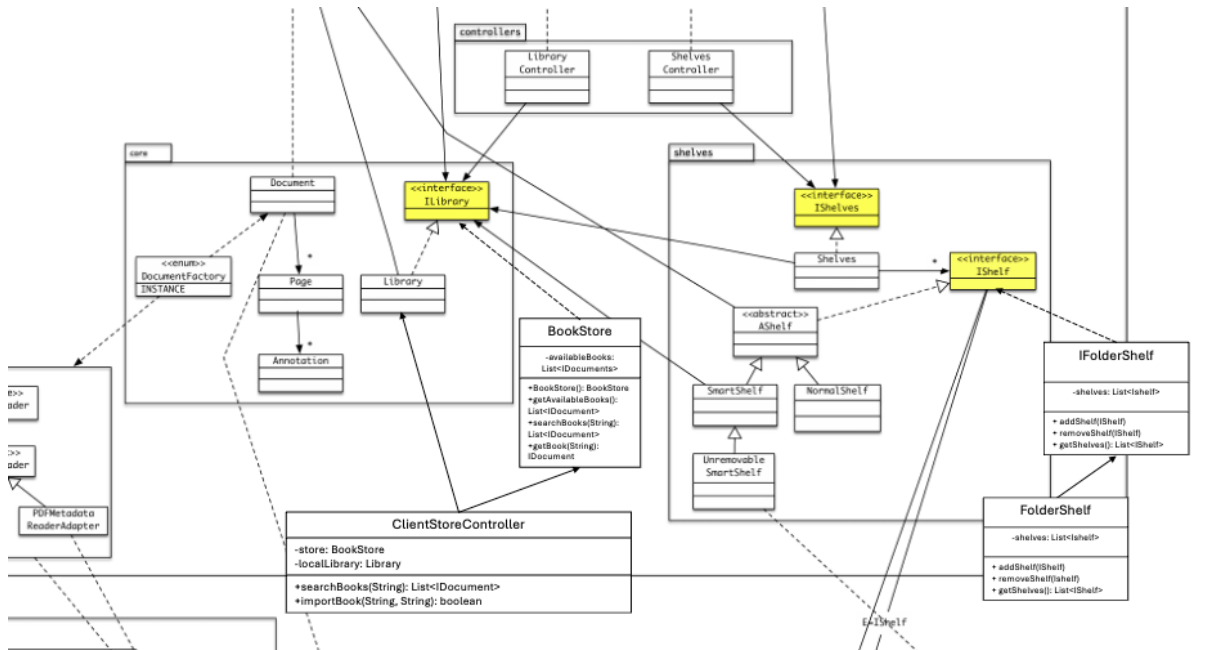


## Descrição da solução de desenho

Rodrigo Frutuoso, 61865

Tiago Leite, 61863



## 1. Introdução

Com base nos requisitos adicionais apresentados no ponto 3.2 do enunciado, foi necessário adaptar a solução de desenho original da aplicação **LEIBooks**. As principais alterações consistem na introdução de **prateleiras do tipo `FolderShelf`** (prateleiras que contêm outras prateleiras) e na **ligação a uma livraria online de livros gratuitos** (`BookStore`).

## 2. Adição do tipo FolderShelf

Para suportar prateleiras que contêm outras prateleiras, foi criada a interface `IFolderShelf`, estendendo `IShelf`, e implementada pela classe `FolderShelf`. Esta estrutura permite a organização hierárquica de prateleiras.

### Atributos de FolderShelf:

- `List<IShelf> shelves`

### Métodos principais:

- `void addShelf(IShelf shelf)`
- `void removeShelf(IShelf shelf)`
- `List<IShelf> getShelves()`

A `FolderShelf` permite armazenar outras prateleiras, quer normais quer inteligentes, sendo útil para estruturar temas ou categorias complexas (exemplo: "Universidade > DCO > Apontamentos").

---

## 3. Integração com livraria externa (`BookStore`)

Foi desenhada a classe `BookStore`, que representa uma livraria online com livros gratuitos disponíveis para inspeção e importação.

### Atributos de `BookStore`:

- `List<IDocument> availableBooks`

### Métodos principais:

- `BookStore BookStore()`
- `List<IDocument> getAvailableBooks()`
- `List<IDocument> searchBooks(String keyword)`
- `IDocument getBook(String title)`

Esta classe é usada pelo `ClientStoreController`, que interage com a `BookStore` para importar livros diretamente para a biblioteca local.

---

## 4. Considerações Finais

Estas alterações foram desenhadas de forma a respeitar os princípios de **modularidade** e **reutilização**.

As novas classes integram-se de forma natural na arquitetura existente, usando as **interfaces** e **mecanismos de eventos** já presentes no domínio.