Sistema de Catalogação de Livros

Evelyn Suzarte Fernandes

Engenharia da Computação – Universidade Estadual de Feira de Santana (UEFS)  
Avenida Transnordestina s/n – 44036-900 – Feira de Santana – BA – Brasil

evelynsuzarte@hotmail.com

**Resumo.** Este relatório tem como objetivo descrever como foi desenvolvido o software, que tem como função principal ser a base de dados de livros do clube de leitura Comunidade da Batata, facilitando a otimização em buscas e pesquisas dos conteúdos do sistema.

# 1. Introdução

O Project Gutenberg oferece mais de 60 mil livros eletrônicos gratuitos. O site é mantido por diversos voluntários que digitalizam e revisam cada obra para mandar a alta qualidade. Esse projeto permite busca online e offline, através do buscador e de um sumário o livro é acessado online e offline o livro fica disponível no formato epub, de texto (.txt) e *Kindle*.

O clube da leitura Comunidade da Batata é fã deste projeto e adota o mesmo formato de alimentar a base de dados, porém sofre com a otimização para busca e pesquisa dos conteúdos do sistema, sendo assim, pediu para que os alunos do curso de Engenharia de Computação criassem um software que armazene, controle e gerencie esse livros de forma a otimizar o tempo de busca offline.

Para a resolução deste problema será feito um software em Linguagem Java pelo programa NetBeans, utilizando árvore binária, listas, manipulação de arquivos, classes e modularização para o sistema operacional Windows.

# 2. Fundamentação teórica

O modelo de projeto escolhido para organização do código do projeto foi o *Data Access Object* (DAO). Segundo MACORATTI (2011) é um padrão de projeto que abstrai e encapsula os mecanismos de acesso a dados escondendo os detalhes da execução da origem dos dados. Seu objetivo é remover a alta dependência do modelo de acesso a dados da nossa aplicação, de forma que se no futuro resolvermos trocar o banco de dados ou adotar outra estratégia de persistência a alteração deverá ser feita apenas em um único lugar (MACORATTI, 2010). Na Figura 1 mostra um exemplo de como é organizado o DAO:

public class DAO{

public static Lista listaCompras = new Lista();

public static void inserirProduto(String produto){

listaCompras.inserirProduto(produto);

}

public static int tamanhoLista (){

return listaCompras.tamanhoLista();

}

}

O DAO faz a conexão com as classes que vão salvar e editar o

Figura 1. Organização da classe DAO

Para a resolução do problema foi utilizada a Árvore como estrutura de dados principal. Segundo BOERES (2019), Árvore é um conjunto finito de elementos, onde um elemento é chamado de raiz e os outros são divididos em subconjuntos disjuntos,qual cada nó tem 0, 1 ou 2 filhos.

Existem vários tipos de Árvores, como a binária, rubro negra e AVL, mas a escolhida foi a Árvore Binária.

Foi utilizada como base a Árvore Binária de BARBOSA (2017) para construção do código.

# 3. Metodologia

A metodologia utilizada para o planejamento da lógica e do código foi a do *Problem Based Learning* (PBL), traduzido para o português como Aprendizagem Baseada Em Problemas.

O método traz uma abordagem curricular centrada no aluno que o capacita a realizar pesquisas, integra teoria e prática, além de possibilitar a aplicação de conhecimentos, habilidades e atitudes para o desenvolvimento de uma solução viável para um problema definido (GEN.NEGÓCIOS & GESTÃO, 2018).

Sendo assim, a partir do debate feito em grupo na sala de aula podemos discutir e concluir soluções para cada requisito do projeto.

# 4. Resultados e discussão

Os seguintes tópicos descrevem um pouco sobre as classes e sobre os requisitos pedidos.

## 4.1. Classes

As classes de programação são receitas de um objeto, aonde têm características e comportamentos, permitindo assim armazenar propriedades e métodos dentro dela (GARCIA, 2015).

No software são usadas 6 classes: Árvore Binária, Dao, Folha, Livro, Manipulação de arquivos e Principal.

### 4.1.1. Árvore Binária

A árvore será usada para armazenar os livros do banco de dados. Ela possui os métodos principais de inserção e remoção, métodos privados auxiliares, métodos de buscas e de listagem de acordo com o parâmetro desejado, e os métodos de carregamento e gravação de dados a partir de um arquivo de texto.

Os métodos de todos os requisitos são executados a partir dessa classe.

### 4.1.2 DAO

Como explicado no tópico 2, o DAO faz a conexão com a interface e os métodos internos. Nele é instanciado a Árvore que será usada e uma lista de *Strings* que será usada para poder armazenar o nome dos autores de cada livro que será usada em algumas partes do código.

Não é necessária fazer a instância dessa classe, pois ela é formada por métodos estáticos, que fazem o retorno ou a chamada dos métodos da Árvore, como mostrada na Figura 2.

Figura 2. Trecho do código da classe DAO

private static ArvoreBinaria arvore\_livros = new ArvoreBinaria();

private static ArrayList<String> autores = new ArrayList();

public static void inserir(Livro livros, int n\_ebook){

arvore\_livros.inserir(livros, n\_ebook);

}

Os métodos *static* tem um relacionamento com uma classe como um todo, enquanto os métodos que não são *static* são associados a uma instância de classe específica (objeto) e podem manipular as variáveis de instância do objeto (DEV MEDIA, 2012).

### 4.1.3 Folha

Nessa classe é criado os nós da Árvore, possui os métodos básicos para manipulações do nó, como o *get* e o *set*.

### 4.1.4. Livro

Com essa classe é possível instanciar um objeto do tipo livro, guardando seus dados: número de Ebook, título, autor, mês, ano e link para acesso.

### 4.1.5. Manipulação de arquivos

Essa classe possui apenas dois métodos estáticos: ler e escrever. Ela vai ajudar na leitura e escrita dos arquivos com a ajuda de um *ArrayList* dos tipos Folha e Livro*.*

Segundo Goodrich e Tamassia (2013), *Arraylist* em sua forma mais simples, é uma coleção de nós que juntos formam uma ordem linear, no qual cada nó é um objeto que guarda a referência para o elemento e para o próximo elemento.

Para a chamada dos métodos dessa classe não é necessária fazer a instância já que são do tipo estáticos.

### 4.1.6. Principal

Nela funciona toda a interface para a comunicação com o usuário, ela possui o comando switch case, para fazer a seleção das opções do menu.

O comando switch avalia uma expressão inteira ou enumeração e faz com que o fluxo de controle desvie para o povo marcado com o valor dessa expressão (GOODRICH E TAMASSIA, 2013).

Para se executar o que deseja, é nessa classe que chamamos os métodos do DAO.

## 4.2. Cadastrar livro

Após o usuário selecionar a opção “Cadastrar livro”, ele digita os dados do livro, exceto o número do ebook que é gerado automaticamente com o método “Dao.geradorNEbook”, onde é verificado qual o maior número de ebook e é somado mais um para gerar um novo código para um novo livro.

É executado o método “Dao.inserir” na classe principal, esse método cria um novo nó para a árvore, adicionando na posição certa.

## 4.3. Carregar base de dados

Assim que é inicializado o programa, automaticamente a base de dados é carregada, a partir do método “Dao.carregarDados”. O arquivo a ser lido já está definido de forma padrão. Esse método recupera os dados de um arquivo de texto em formato CSV e armazena todo o conteúdo na Árvore.

Segundo Microsoft (2020), um arquivo CSV (Valores Separados por Vírgula) é um tipo especial de arquivo. Em vez de armazenar informações em colunas, os arquivos CSV armazenam informações separadas por vírgulas. Quando o texto e os números são salvos em um arquivo CSV, é fácil movê-los de um programa para outro.

## 4.4. Gravar arquivo

O arquivo final com todos os livros é gerado quando o usuário seleciona a opção “Encerrar e gravar programa”. O método “Dao.gravarDados” salva todos os livros da Árvore no arquivo de texto em formado CSV, atualizando-o toda vez que é selecionada essa opção.

## 4.5. Listar autores e sua quantidade de livros

Quando o usuário seleciona a opção “Listar Autores/Quantidade”, é mostrado o nome de todos os autores e a quantidade de livros de cada um. O método executado é o “Dao. listarAutoresQuantidade”.

## 4.6. Listar livros de determinado autor

Após o usuário selecionar a opção “Listar Autores/Livros”, ele digita o nome do autor que deseja buscar. O método “Dao.listarAutoresLivro” busca na Árvore e exibe os livros que pertencem ao autor desejado e ao mesmo tempo cria um arquivo de texto salvo com o nome do autor e dentro todos os seus livros.

## 4.7. Listar todos os livros

Selecionando a opção “Listar livros”, o método “Dao.listarTodosLivros” percorre toda a Árvore exibindo todos os livros.

## 4.8. Buscar livro por número do Ebook

Na opção “Buscar Livro”, o usuário digita o número do Ebook. O método “Dao.buscarLivro\_nEbook” busca o número de Ebook na Árvore e caso ele exista, é exibido o link pra acesso do livro, caso contrário, o software notifica o usuário avisando que o livro buscado não existe no banco de dados.

## 4.9. Buscar livro por ano

Na opção “Buscar Livro/Ano”, o usuário digita o ano que deseja visualizar seus livros. O método “Dao.buscar\_ano” busca o ano do livro na Árvore e todos os livros daquele ano são exibidos. Ao mesmo tempo é criado um arquivo de texto salvo com o ano que desejou ser buscado e dentro do arquivo possui os livros daquele ano.

## 4.10. Excluir livro

Selecionando “Excluir livro”, o usuário digita o número do Ebook. O método “Dao.excluirLivro” busca na Árvore o livro para ser excluído e remove ele. Ele retorna True ou False para poder saber se o livro desejado foi excluído na árvore.

# 5. Conclusões

Todos os requisitos pedidos para a produção do software foram cumpridos. Os pontos fortes são a organização do menu e a possibilidade de estar sempre atualizando o banco de dados através do arquivo de texto, mantendo sempre salvo as últimas alterações, já o ponto fraco, a Árvore poderia ser do tipo balanceada, o que garantiria melhor facilidade na busca e inserção de um objeto.

A partir do desenvolvimento do software passei a entender e perceber a importância de uma Árvore, aprendi mais sobre como ler e escrever em arquivos e também em como organizar melhor meu código utilizando o modelo de projeto DAO.

# 6. Referências

Boeres, C. (2019), Estrutura de dados, Árvores, árvores binárias e percursos.

Macoratti, J. C. (2011) “Apresentando o padrão DAO – Data Access Object.”, http://www.macoratti.net/11/10/pp\_dao1.htm, Janeiro.

Macoratti, J. C. (2010) “Usando padrões de projeto na prática – 2”, http://www.macoratti.net/10/07/vbn\_ppp2.htm, Janeiro.

Barbosa, D. (2017) “ArvoreBinariaApp”, https://gist.github.com/divanibarbosa/819e7cfcf1b9bae48c4e0f5bd74fb658, Dezembro.

Garcia, V. G. A. (2015) “Classes, Objetos, Atributos e Métodos em Java”, http://fabrica.ms.senac.br/2015/03/classes-objetos-atributos-e-metodos-em-java/, Janeiro.

Goodrich, M. T. e Tamassia, R. (2013) “Estruturas de Dados & Algoritmos em Java”, Bookman Editora.

Gen. Négocios & Gestão (2018) “O que é PBL? Entenda o método de aprendizagem baseada em problemas”, https://gennegociosegestao.com.br/o-que-e-pbl/, Janeiro.

Dev Media (2012) “Trabalhando com métodos em Java”, https://www.devmedia.com.br/trabalhando-com-metodos-em-java/25917, Janeiro.

Microsoft (2020) “Criar ou editar arquivos .csv para importação para o Outlook”, https://support.office.com/pt-br/article/criar-ou-editar-arquivos-csv-para-importa%C3%A7%C3%A3o-para-o-outlook-4518d70d-8fe9-46ad-94fa-1494247193c7, Janeiro.