

Problema 2: Sistema de Catalogação de Livros

O **Project Gutenberg** (disponível em www.gutenberg.org), lhe oferece mais de 60.000 livros eletrônicos gratuitos. Esse projeto é mantido através da ajuda de milhares de voluntários que digitalizam e revisam cuidadosamente cada obra a fim de manter uma alta qualidade. Todos os seus livros foram previamente publicados por editores genuínos.

Esse projeto permite o acesso on-line e off-line. Para o acesso on-line, o site disponibiliza a busca pelos motores de busca do Yahoo!, Google e através de um sumário separado por autores, títulos e idiomas. Já para o acesso off-line, são disponibilizadas bases de dados gratuitas com atualizações semanais no formato *epub*, Kindle e em formato de texto (.txt).

O clube da leitura **Comunidade da Batata**, grande fã desse projeto, utiliza o formato de texto para alimentar a sua base de dados. Porém sofre muito com a falta de otimização em buscas e pesquisas dos conteúdos em seu sistema. Pensando em otimizar esse processo e conhecendo a alta capacidade dos alunos do MI de Programação do curso de Engenharia de Computação da UEFS, o clube solicita para os alunos um protótipo de um *software* que armazene, controle e gerencie esse livros de forma a minimizar o tempo de busca off-line.

O objetivo é agilizar o tempo de busca dos livros conforme o desejo do usuário, como por exemplo as listagens de todos os livros de um autor ou dos livros catalogados por ano de publicação. Após uma tentativa frustrada do clube de otimizar a sua base dados, a única conclusão que chegaram foi que deveria ser utilizada uma estrutura de dados com complexidade média de $O(\log n)$ para as buscas, inserção ou remoção e a linguagem de programação Java.

Para garantir que o sistema funcione perfeitamente e sem erros, é necessário que sejam construídos os testes de unidade com o propósito de possibilitar testar o sistema com mais afinco, segurança e verificar se o mesmo está atendendo aos requisitos solicitados. Todas as classes e testes devem estar documentadas utilizando o padrão **javadoc**.

A partir de uma pesquisa entre seus membros, os casos de uso para o Sistema de Catalogação de Livros foram:

Tabela 1.: Casos de uso

Nº	Título	Descrição
1	Cadastrar - Livro	Um novo livro deve ser inserido no sistema.
2	Carregar base de dados	Um arquivo CSV contendo uma listagem de livros deve ser carregado na estrutura do sistema.
3	Gravar arquivo	Um arquivo CSV contendo os dados de livros devem ser gravado em disco.
4	Listar - Autores/Quantidade	Todos os autores, juntamente com a quantidade de obras catalogadas, devem ser listados.
5	Listar - Autor/Livros	Dado um autor, todos os seus livros devem ser listados.
6	Listar - Livros	Todos os livros catalogados devem ser listados.
7	Buscar Livro	Dado o nº do eBook deve ser verificado se o livro existe no sistema e, em caso positivo, mostrar o seu endereço para acesso.
8	Buscar Livro/Ano	Dado o ano de publicação, todos os livros devem ser listados juntamente com todas as suas informações.
9	Excluir Livro	Um determinado livro deve ser excluído da estrutura do sistema.
10	Testes automatizados	Todos os requisitos do sistema devem poder ser extensivamente testados e isso deve ser feito de modo automatizado. Os testes devem ser devidamente descritos para que os próximos desenvolvedores entendam rapidamente o que exatamente cada teste verifica em relação ao funcionamento do sistema.

A **Comunidade da Batata** enviou um exemplo do arquivo utilizado para alimentar a sua base de dados para que os alunos manipulem exatamente o mesmo tipo arquivo utilizado em seu clube.

TRECHO DO ARQUIVO CSV

N_EBOOK;TITULO;AUTOR;MES;ANO;LINK

267322;The Opal Pin;Rufus H Gillmore;Nov;2019;http://gutenberg.net.au/ebooks19/1901131h.html

267306;Trail Smoke;Ernest Haycox;Nov;2019;http://gutenberg.net.au/ebooks19/1901121h.html

267290;The Red Ledger;Frank L Packard;Nov;2019;http://gutenberg.net.au/ebooks19/1901111h.html

267424;Never Again;Frank L Packard;Nov;2019;http://gutenberg.net.au/ebooks19/1901101h.html

267162;Corporal Bob;Frank L Packard;Nov;2019;http://gutenberg.net.au/ebooks19/1901091h.html

267146;Coogan's Last Run;Frank L Packard;Nov;2019;http://gutenberg.net.au/ebooks19/1901081h.html

267130;The Whispering Lane;Fergus Hume;Nov;2019;http://gutenberg.net.au/ebooks19/1901071h.html

267114;Leathermouth;Carlton Dawe;Oct;2019;http://gutenberg.net.au/ebooks19/1901061h.html

267098;More Lives than One;Carolyn Wells;Oct;2019;http://gutenberg.net.au/ebooks19/1901051h.html

Um outro requisito exigido pela **Comunidade da Batata** é que todas as buscas realizadas pelo sistema sejam salvas em arquivos CSV utilizando como nome do arquivo os parâmetros definidos na busca. Por exemplo: "Rufus H Gillmore.txt", busca de todos os livros do autor.

Produto:

Você deve fazer a implementação deste software (**duplas são permitidas para desenvolvimento do código fonte**), seguindo as boas práticas de Programação Orientada a Objetos, usando a linguagem de programação Java. Poderá ser utilizada a API do Java **apenas** para as Listas e Filas, qualquer outra estrutura de dados você terá que fazer a sua própria implementação.

Você deve enviar um e-mail com o **Código-Fonte** para o seu tutor até às **23 horas 59 minutos do dia 23/12/2019 (segunda)**, anexando o arquivo compactado com o código-fonte e os testes de unidade, e **informar a IDE e versões de framework** utilizadas para desenvolvimento. O código-fonte deve estar todo documentado utilizando o padrão javadoc.

Também deverá ser entregue um relatório técnico escrito no formato de artigo da SBC (**relatório individual**). Deverá ter entre seis a oito páginas até o dia **27/12/2019**, contendo:

- Uma introdução com a apresentação resumida da motivação, contexto e problema;
- A fundamentação teórica;
- A metodologia utilizada;
- Resultados e discussão (com a descrição detalhada da sua solução, declarando as decisões que você teve de tomar e fundamentando-as, sempre que possível);
- As suas conclusões (com as lições aprendidas e os pontos positivos e negativos deste projeto).

Você encontra as recomendações para a elaboração do relatório e produto no site da disciplina (<http://sites.ecomp.uefs.br/mip-20192/home/tutorial>).

Avaliação e Cronograma:

O desempenho nas sessões tutoriais equivale a 30% da nota no Problema. O relatório equivale a 30% da nota e o código fonte (produto) equivale a 40%. Haverá penalidade de 1 ponto por descumprimento do prazo de entrega e 1 ponto por dia de atraso na entrega a contar da entrega do código-fonte, até o máximo de cinco dias. Após este prazo, o trabalho não será mais aceito.

Data	Sessão Tutorial
29/11	Entrega Problema 2
03/12	Sessão Tutorial - Problema 2
06/12	Sessão Tutorial - Problema 2
10/12	Sessão Tutorial - Problema 2
13/12	Sessão Tutorial - Problema 2
17/12	Sessão Tutorial - Problema 2
20/12	Sessão Final - Problema 2
23/12	Entrega do Código-Fonte
27/12	Entrega do Relatório