UNIVERSIDADE ESTADUAL DE CAMPINAS

FACULDADE DE TECNOLOGIA

**RELATÓRIO**

ESTRUTURAS DE ARQUIVOS – ST562 B

Alunos: Caroline Resende Silveira (165921)

Mirelle Candida Bueno (174909)

Otavio Passarelli Praça (175390)

Vitor Artoni de Marcio (178379)

Professor: Dr. Celmar Guimarães da Silva

Novembro de 2016

Este documento contém o trabalho da disciplina Estrutura de Arquivos, orientado pelo Prof. Celmar Guimarães da Silva com entrega no dia 21 de novembro de 2016.

O objetivo do projeto é implementar um sistema de gerenciamento de arquivo de registros, visando treinar os conceitos aprendidos sobre gerência de arquivos de registros e indexação baseada em listas simples.

**Sumário**

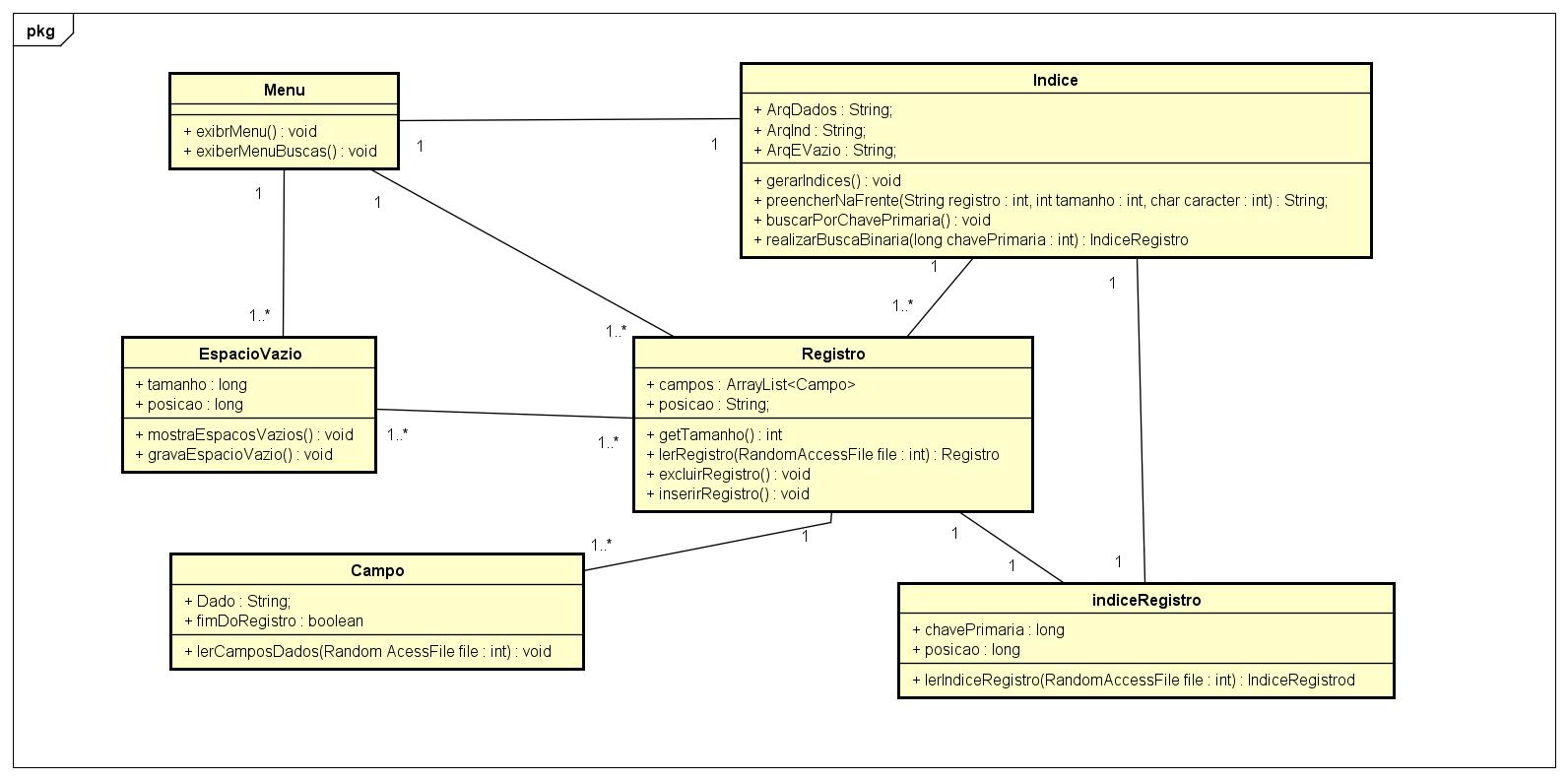
[Diagrama UML 4](#_Toc467534503)

[Instruções de compilação e execução 5](#_Toc467534504)

[Relatório detalhado 6](#_Toc467534505)

[Referências 14](#_Toc467534506)

# Diagrama UML



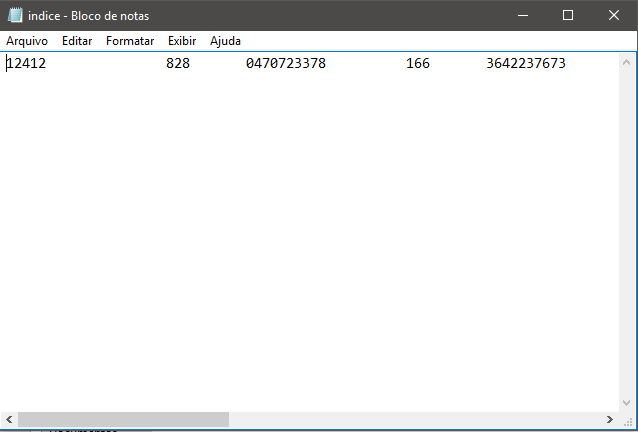
No projeto há a classe Menu que será responsável na escolha das opções que o usuário fizer. Já a classe indice tem a função de gerar um arquivo de indices, buscar chaves primárias e realizar busca binária . A classe registro tem a função de ler os registros de um arquivo de dados,inserir registros e excluir,sendo possível além disso obter os campos e a posição de um determinado registro. A classe espaço vazio tem a função de mostrar os espaços vazios e grava-los , para isso é necessário saber o tamanho e a posição do registro. A classe Campo tem a função de ler os campos de dados de um registro e encontrar o final dos registros. A classe Indice Registro, a partir da chave primária e da posição de um registro é capaz de ler o indice do registro.

# Instruções de compilação e execução

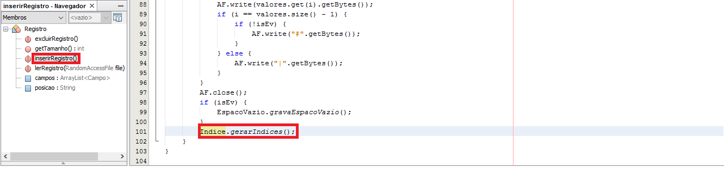
* Foi utilizado a linguagem Java.
* O Desenvolvimento do projeto ocorreu no NetBeans (Oracle).
* A compilação do projeto deve ocorrer diretamente na plataforma ou em algum ambiente de desenvolvimento semelhante;
* O arquivo “dados.txt” deve estar na mesma pasta na qual está o código implementado;

# Relatório detalhado

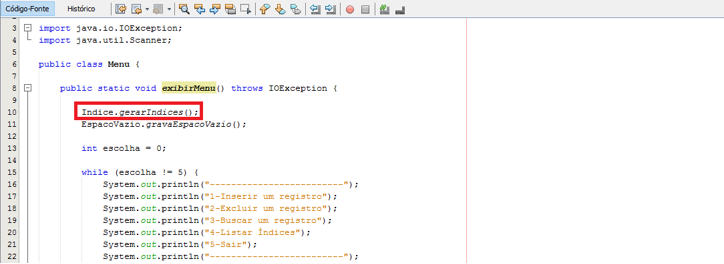
1. O sistema deve criar índices sobre todos os campos do arquivo de dados



**Índice sobre a chave primária (ISBN) gerado pelo código**

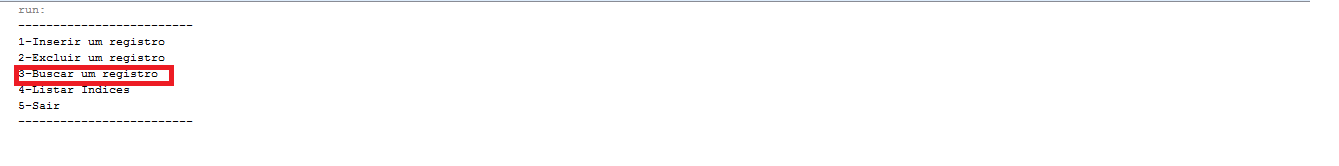
****

**Código para gerar índices.**

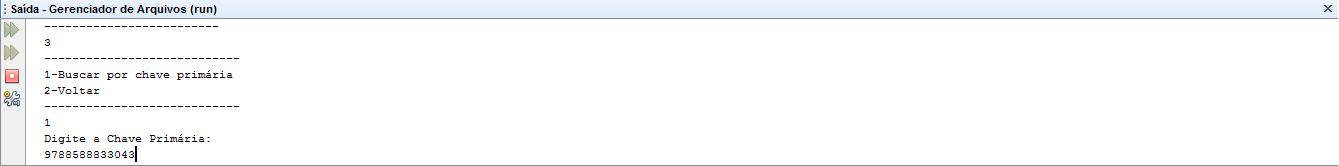
****

**Código para gerar índices via “Indice.gerarIndices()”**

1. O sistema deve permitir ao usuário consultar registros por chave primária ou por chave secundária (sempre com índice)



**Menu de opções -> Buscar um registro**

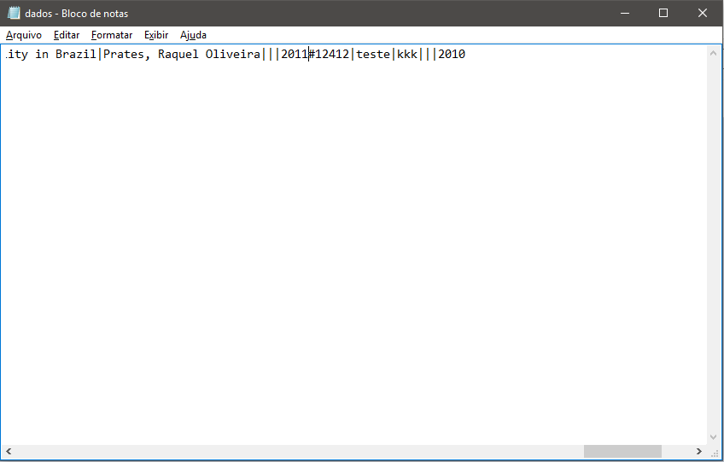
****

**Opção “Buscar por chave primária”**

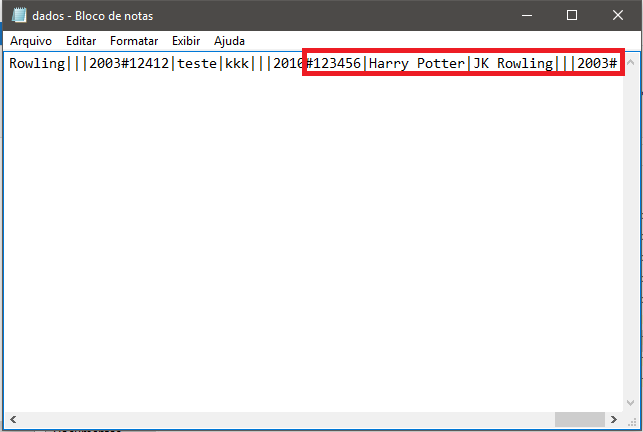
****

**Resultado da busca por chave primária via índice**

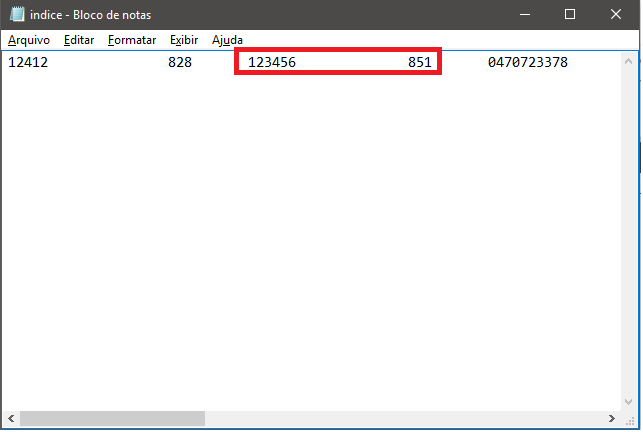
1. O sistema deve permitir ao usuário inserir registro, atualizando todos os arquivos adequadamente

****

**Arquivo “dados.txt” antes da nova inserção.**

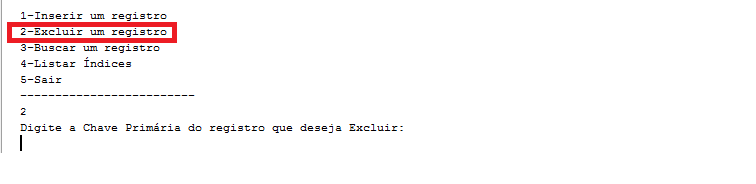


**Arquivo “dados.txt” com novo registro inserido.**

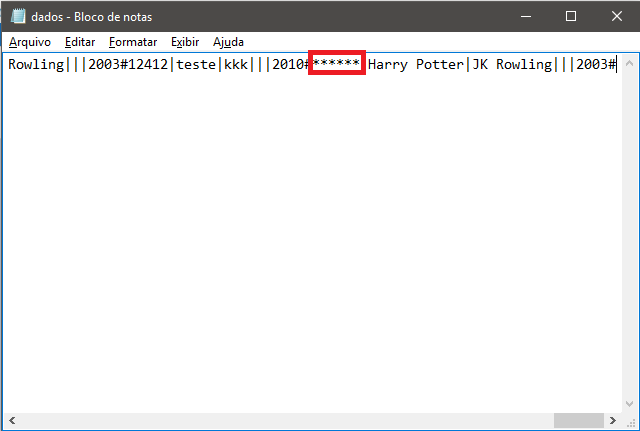
****

**Arquivo de índice utilizado para inserção do registro.**

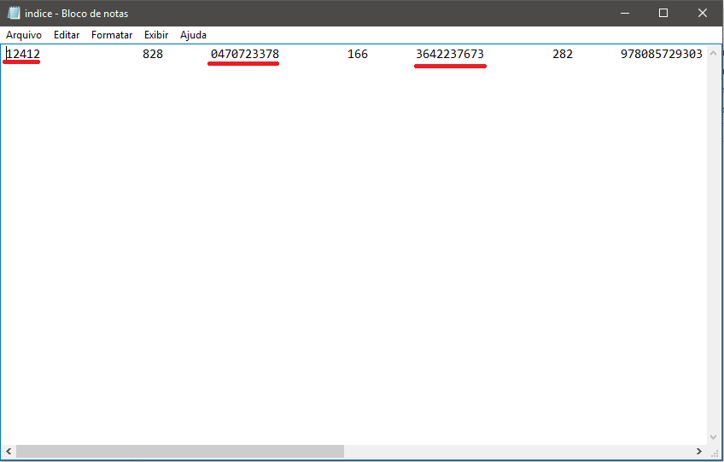
1. O sistema deve permitir ao usuário remover registro (dada sua chave primária), atualizando todos os arquivos adequadamente



**Menu de opções -> Excluir um registro**

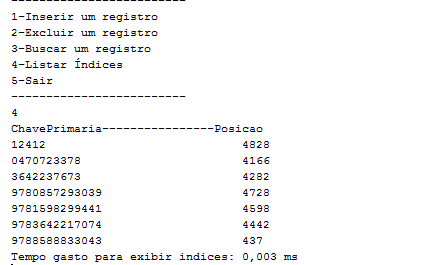
****

**Registro foi excluido**

****

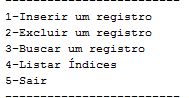
**Arquivo de índices foi atualizado.**

1. O sistema deve permitir ao usuário ver os índices utilizados. Estes devem ser impressos na tela em formato de tabelas, e devem ser legíveis ao usuário



**Exibição dos índices utilizados**

1. O sistema deve ser interativo (não necessariamente em tela gráfica), permitindo ao usuário informar os nomes ou valores de campos conforme as opções disponibilizadas



**Menu com as opções disponibilizadas**

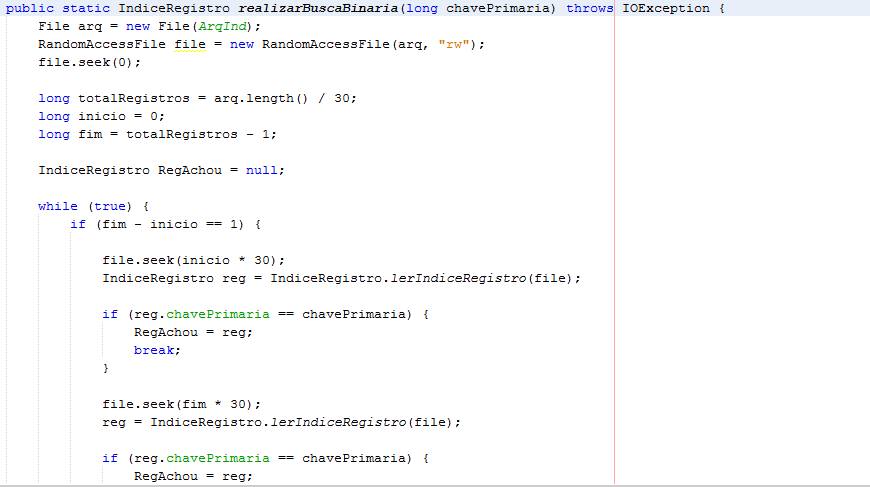
7. Índices devem ser implementados como listas, e em arquivo

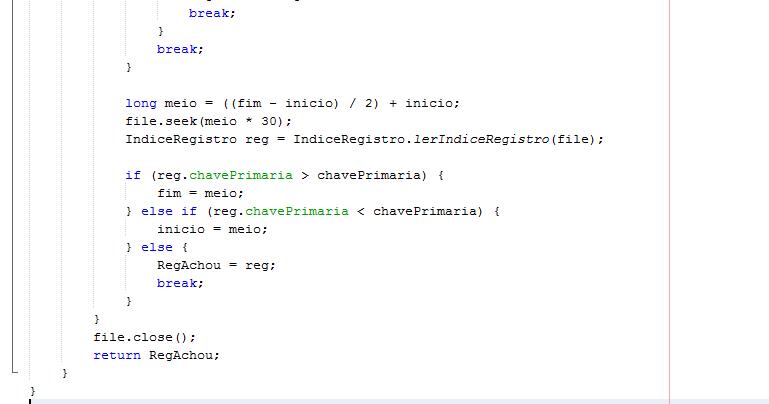
8. Operações de consulta devem ser feitas diretamente em arquivo (não vale trazer o arquivo inteiro para a RAM para então fazer consulta)



**Operações realizadas diretamente em Disco.**

9. Buscas feitas nos índices devem ser buscas binárias

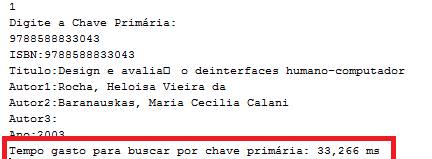




**Busca Binária**

10. O sistema deve usar listas invertidas de chaves primárias como apoio à implementação de índices secundários

11. Operações de consulta devem exibir tempo gasto



**Operação de Consulta e seu respectivo tempo gasto.**

12. O sistema também deve funcionar corretamente com outros arquivos que não o exemplificado neste trabalho, mas que tenham registro de cabeçalho com nomes de registros, em formato semelhante a este.

# Referências

[1] Folk, M. J. Zoellick, B. File Structures. 2ª ed. Addison-Wesley, 1991.

[2] Neto, J. P. Programação, Algoritmos e Estruturas de Dados. 3ª ed. Escolar, 2014.

[3] Deitel, H. C++ Como Programar. 5ª ed. Pearson, 2006.

[4] Mizrahi, V. V. Treinamento em linguagem C. 2ª ed. Prentice Hall, 2008.

[5] Edicezar, L. N. Estudo de sistemas de arquivos estruturados em IOG e uma proposta de implementação para o ambiente LINUX. UNICAMP, 1998. Disponível em: http://www.bibliotecadigital.unicamp.br/document/?code=000129878 (Acesso em 11 de novembro de 2016)