

## Revista Abakós do Instituto de Ciências Exatas e de Informática Pontifícia Universidade Católica de Minas Gerais\*

Model - Magazine Abakós - ICEI - PUC Minas

Gabriel Alves Costa Nardy <sup>1</sup>

Diogo Casal Ferreira <sup>2</sup>

### Resumo

O documento que se segue irá descrever, de forma objetiva e clara, o projeto desenvolvido por Diogo Casal Ferreira e Gabriel Nardy para solucionar a primeira entrega do trabalho prático proposto na disciplina de AEDS III. O trabalho para a primeira entrega consiste na criação de um sistema de cadastros de prontuários para uma empresa de planos de saúde que poderá inserir, alterar, excluir e imprimir os prontuários. Futuramente em entregas subsequentes serão realizadas aplicações com índices e com impressões de posições específicas de dados.

---

\* Artigo apresentado à Revista Abakós

<sup>1</sup> Programa de graduação em Ciência da Computação da PUC Minas, Brasil - gabriel.nardy@sga.pucminas.b

<sup>2</sup> Programa de graduação em Ciência da Computação da PUC Minas, Brasil - diogo.casal@sga.pucminas.b

## **1 ESPECIFICAÇÕES**

### **1.1 Proposta apresentada**

Para o trabalho prático o grupo deveria criar uma sistematização em JAVA ou C++ do tipo CRUD em arquivos, escolhemos a linguagem JAVA em função de maior familiarização com linguagem orientada a objetos. Deveria ser feito um sistema de prontuários para uma empresa de planos de saúde de forma otimizada sem que houvesse um uso excessivo de memória.

### **1.2 Composição do trabalho**

A princípio, a primeira entrega desse trabalho apresenta apenas operações realizadas no arquivo mestre diretamente, compostos por arquivo CLASS e o Main que será executado para possibilitar análise do arquivo mestre.

#### ***1.2.1 Arquivo Mestre - Prontuário***

O arquivomestre.db armazena as informações do Prontuário, possuindo um cabeçalho seguido das informações registradas dos pacientes e anotação do médico responsável.

### **1.3 Descrição das técnicas implementadas**

As técnicas utilizadas foram a implementação de um sistema de CRUD simples

Os dados que irão persistir, cabeçalhos e prontuários irão ser salvos em memória secundária, mas quando o programa é iniciado e há leitura do cabeçalho e atribuição de variáveis estáticas de ambiente há uso de memória principal.

### **1.4 Descrição do Hardware.**

Os componentes utilizados para executar os testes do sistema desenvolvido foram, um processador Intel i5-4440 e 8gb de Ram ddr3, testes executados em ssd.

## **2 DESCRIÇÃO S.O.**

Utilizou-se o Windows 10 PRO, na versão 11.0.1904 build 19043.1165.

## **3 IMPLEMENTAÇÃO DO CÓDIGO**

No momento de inicialização do programa, primeiramente uma função é chamada afim de checar a existencia de documentos presentes previamente no sistema, caso a tentativa tenha sucesso em verificar a existência de certas informações, as mesmas serão carregadas em memória principal afim de facilitar seu uso no programa.

Já caso a tentativa falhe funções adjacentes serão chamadas em sequência para obter certas informações iniciais e criar os arquivos necessarios afim de permitir o funcionamento do sistema. É requisitado o tamanho de m(numero de caracteres da anotação do prontuário).

Após a coleta das informações iniciais é impresso na tela do prompt um menu com as opções:

- 1) Cadastrar prontuário
- 2) Alteração das anotações
- 3) Excluir
- 4) Sair do sistema.

### **3.1 Opção 1 - Cadastrar novo prontuário**

Para desbloquear o funcionamento do resto do sistema, primeiramente é necessário cadastrar um prontuário (opção 1), onde é requerido nome, data de nascimento, cpf e sexo do paciente.

### **3.2 Opção 2 - Alteração do campo de anotações**

Após a criação do prontuário, o campo de anotações é iniciado vazio, e é necessario selecionar a opção de Alteração do campo de anotações afim de preenche-lo ou alterá-lo.

A opção ao ser chamada inicia requerindo o cpf do paciente que deve ter seu campo de anotação modificado

### **3.3 Opção 3- Excluir**

Inicia-se o processo com a busca de um dado cpf de um prontuário ,sendo este localizado, ele é apagado de maneira logica através da inserção de uma lápide.

No arquivo mestre ao ocorrer a exclusão duas ações ocorrem, o cabeçalho de registros ativos sofre uma alteração de subtração em 1, e é posto uma lápide no primeiro prontuário.

## **4 REFERÊNCIAS BIBLIOGRÁFICAS**

RATHI, Ashok; LU, Huizhu; HENDRICK, George E. Performance comparison of extendible hashing and linear hashing techniques. In SIGSMALL/PC Symposium, p. 178-185, 1990.

ENBODY, Richard J.. ; DU, Hung-Chang. Dynamic hashing schemes. ACM Computing Surveys (CSUR), v. 20, n. 2, p. 850-113, 1988.

LITWIN, Witold. Linear hashing: a new tool for file and table addressing. In Proceedings of the sixth international conference on Very Large Data Bases - Volume 6 (VLDB '80). VLDB Endowment, p. 212–223, 1980.

FAGIN, Ronald et al. Extendible hashing—a fast access method for dynamic files. ACM Transactions on Database Systems (TODS), v. 4, n. 3, p. 315-344, 1979.

## **Referências**