



Relatório de Análise e Funcionamento do Sistema de Gerenciamento de Registros de Atletas

Colaboradores:

Fábio Damas Valim (202410372)
Guilherme Lirio Miranda (202410367)
Marcos Vinicius Pereira (202411098)

Contents

1	Introdução	3
2	Estruturas de Dados Utilizadas	3
2.1	Estrutura Atleta	3
2.2	Estrutura Cabeçalho	3
3	Uso das Estruturas no Código	4
3.1	Estrutura Atleta	4
3.2	Estrutura Cabeçalho	4
4	Operações Realizadas pelo Programa	4
4.1	Inserção de Dados via Arquivo de Texto (.csv)	4
4.2	Inserção de Dados via Entrada Padrão	5
4.3	Remoção de Dados	5
4.4	Busca de Dados Específicos	5
4.5	Visualização de Dados	5
5	Divisão e Gerenciamento de Blocos	5
6	Ordenação dos Registros	5
7	Gestão de Cabeçalhos	6
8	Conclusão	6

1 Introdução

O presente relatório visa abordar a análise e o desempenho de um sistema projetado para o controle de registros de atletas. Para a manutenção de dados é utilizada a prática de arquivos binários. As operações efetuadas pelo sistema são: inserção, busca, remoção e exibição. As operações são realizadas tendo como base um sistema de blocos que contém as informações. Para a indexação utiliza-se a estrutura atleta e cabeçalho. Cada estrutura com uma função específica dentro do programa.

2 Estruturas de Dados Utilizadas

2.1 Estrutura Atleta

A estrutura “atleta” é usada para armazenar informações específicas dos atletas. Os campos desta estrutura são:

- **id:** Um inteiro que especifica o identificador único para cada atleta.
- **nome:** Um vetor de caracteres `char[150]` que armazena o nome do atleta.
- **sexo:** Um caractere que especifica o sexo do atleta (no formato M/ F).
- **idade:** Um número de ponto flutuante que especifica a idade do atleta.
- **altura:** Um número de ponto flutuante que especifica a altura do atleta.
- **peso:** Um número de ponto flutuante que especifica o peso do atleta.
- **time:** Um vetor de caracteres `char[150]` que armazena o nome do time ao qual o atleta pertence.

2.2 Estrutura Cabeçalho

A estrutura “cabeçalho” é utilizada para guardar informações sobre os blocos de dados no arquivo binário. Os campos da estrutura cabeçalho são os seguintes:

- **quantidade:** Um inteiro que indica a quantidade de registros de atletas presentes no bloco.
- **proximo:** Um número inteiro que armazena o índice do próximo bloco, ou o valor -1 se não houver próximo bloco.

3 Uso das Estruturas no Código

As estruturas “atleta” e “cabeçalho” são utilizadas em diversas funções dentro do programa para manipular os dados provenientes do arquivo .csv:

3.1 Estrutura Atleta

- **escreverNoBloco()**: Função que grava os dados de um atleta no arquivo binário.
- **buscarRegistro()**: Usa a estrutura “atleta” para armazenar e mostrar os dados do atleta encontrado na busca.
- **carregarArquivo()**: Completa os campos da estrutura “atleta” através dos dados presentes no arquivo CSV inserido no sistema.
- **removerRegistro()**: Usa a estrutura “atleta” para analisar, remover e reorganizar os registros dentro do arquivo binário.

3.2 Estrutura Cabeçalho

- **lerCabeçalho()** e **atualizarCabeçalho()**: As funções são responsáveis, respectivamente, por ler e atualizar os dados do cabeçalho presentes no arquivo binário.
- **criarBloco()**: Gera um novo arquivo binário com um cabeçalho inicial.
- **dividirBloco()**: Usa a estrutura “cabeçalho” para organizar a divisão dos blocos e manter a ordem sobre o próximo bloco a ser utilizado no programa.

4 Operações Realizadas pelo Programa

4.1 Inserção de Dados via Arquivo de Texto (.csv)

O programa possibilita a importação dos dados de um arquivo CSV, que possui informações sobre atletas. Ao inserir um novo atleta, a ordem dos identificadores é mantida, servindo como método de organização. Assim que um bloco atinge sua capacidade total, ele é dividido.

4.2 Inserção de Dados via Entrada Padrão

O programa permite que o usuário insira os dados de um atleta manualmente. Semelhante à inserção via arquivo de texto, a inclusão do novo atleta respeita a ordem dos IDs e divide os blocos quando for necessário.

4.3 Remoção de Dados

O sistema permite a remoção de um atleta específico, com base no seu identificador. Antes da remoção é necessário a confirmação do usuário e, após isso, os registros seguintes são deslocados para preencher a lacuna criada, e o cabeçalho do bloco é atualizado.

4.4 Busca de Dados Específicos

O usuário pode procurar registros específicos por ID. A busca é feita sequencialmente por meio dos blocos até encontrar o atleta ou concluir que o ID não existe.

4.5 Visualização de Dados

O sistema possibilita que o usuário visualize todos os registros inseridos no programa, percorrendo todos os blocos e exibindo os dados dos atletas cadastrados.

5 Divisão e Gerenciamento de Blocos

É essencial separar os dados em blocos para otimizar o armazenamento de forma eficiente. Quando um bloco atinge o limite de 50 registros no sistema é criado um novo bloco e uma parte dos registros é transferida para ele. Cada bloco possui um cabeçalho com detalhes sobre o número de registros e um ponteiro para o próximo bloco facilitando a navegação entre os blocos.

6 Ordenação dos Registros

Os registros dentro de cada bloco são ordenados por ID. A ordenação é realizada ao dividir blocos, para garantir que os dados permaneçam organizados.

7 Gestão de Cabeçalhos

Cada bloco começa com duas informações importantes no topo: o número de registros e o indicador do próximo bloco na sequência de blocos seguintes. Estas informações são modificadas à medida que novos dados são incluídos ou eliminados para permitir uma gestão eficiente do armazenamento e facilitar as transições entre os blocos adjacentes.

8 Conclusão

A gestão de registros de atletas é feita de maneira simples e eficiente por meio de arquivos binários bem estruturados no programa em questão. As operações como adicionar dados novos à base de dados ou remover informações são realizadas com eficiência para garantir integridade dos registros. O grupo reintera que o trabalho, ainda que complexo, foi uma gratificante experiência prática, aproximando-nos do ofício diário de quem atua na área da computação. Por fim, analisamos que entre as maiores dificuldades encontradas destaca-se a implementação do código em si, a subdivisão dos blocos em arquivos binários independentes foi desafiadora e, de maneira contrária, a maior facilidade observada foi na própria relação entre os colaboradores do projeto, o compromisso com o desenvolvimento do projeto foi apresentado através da boa organização do código para que todos os membros pudessem manuseá-lo e do respeito com datas e horários definidos para discutir e avançar no desenvolvimento do projeto final.