

## **Relatório do projeto prático de “Estruturas de Dados” – Parte 1**

**Participantes: Esther Silva de Magalhães, Ítalo Alves Rabelo, Pedro Henrique Cabral Moreira**

O objetivo desse projeto prático é fazer a leitura de um arquivo de dados no formato .csv e convertê-lo para arquivo binário, local em que as operações serão feitas. Para desenvolvimento deste trabalho, foi utilizado a base de dados "data\_athlete\_event.csv".

### **- Descrição da estrutura de dados utilizada:**

Declaração de uma classe denominada “Dados” com os campos:

“mId” (identificação), “mNome” (nome do atleta), “mCidade” (cidade do atleta), “mEsporte” (esporte do atleta), “mEvento” (evento) e “mNoc” (Comitê Olímpico Nacional).

Suas funções membros são:

“conversao()” – ler o arquivo .csv e transformar para um arquivo tipado.

“exibir()” – imprime na tela todos os registros.

“exibir\_intervalo()” – imprime na tela todos os registros de um intervalo específico.

“alterar\_dado()” – altera qualquer campo de um registro existente.

“trocar\_elementos()” – troca dois registros de posição.

“adicionar()” – adiciona um novo registro no arquivo.

### **- Descrição em alto nível explicando a lógica do programa:**

No programa principal, o arquivo .csv é aberto para leitura e os registros são convertidos para código binário de forma sequencial e transferidos para um arquivo binário.

A função “conversao()” realiza esse processo. Ela utiliza o getline para ler, linha por linha, os campos separados por vírgula. Durante a leitura, trata os casos em que um campo contém mais de um par de aspas, além de vírgulas entre aspas, e os campos lidos são armazenados por um construtor da classe Dados e escritos no arquivo binário.

Durante esse processo, o progresso da conversão é exibido na tela, exibindo a porcentagem de linhas que foram processadas.

Em seguida, é exibido um menu de opções (a fim de “chamar” os subprogramas), em que o usuário escolhe a opção que lhe convém.

## **- Conclusão**

Durante a realização deste projeto, ficou claro a importância de organizar os dados de forma eficiente, a fim do processamento de grande quantidade de dados ser realizado da maneira mais rápida possível. Portanto, minimizando os índices de erros.