Relatório da 1ªFase do Trabalho Prático

de Estruturas de Dados e Algoritmos II

Diogo Solipa(l43071) Leonardo Catarro(l43025)



# Estruturas de Dados

## Descrição

## Razões da escolha

## Dimensionamento

## Ligação entre as estruturas

## Localização

## Descrição dos Elementos das estruturas

## Memória ocupada pelas estruturas em memória central

# Ficheiros de Dados

Todos os ficheiros possuíram a mesma estrutura.

Teremos um total de 195 ficheiros neste projeto.

Destes 195, teremos 1 ficheiro que conterá todos os estudantes existentes, 193 em cada um deles representará um país em especifico e 1 ficheiro suplementar para o caso de um dos países possuir mais de 5 000 000 estudantes.

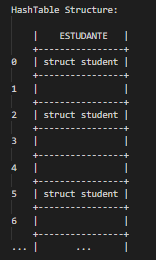
Em memória central teremos um array de structs do tipo student com 5 000 000 posições que servirá para guardar os vários estudantes para as várias operações.

O valor 5 000 000 é o total de alunos que conseguimos ter na memória central. Valor dado por:

64 000 000 / tamanho da Struct(12bytes) = 5 333 333.

Por questão de organização e visto que o total de alunos possível é de 10 000 000, optamos por utilizar os 5 000 000 de forma a deixar alguma memória livre, para caso de necessidade.

A estrutura de cada ficheiro será a seguinte:



Cada posição da Tabela conterá uma struct que representará um aluno em específico. Struct essa que conterá os seguintes elementos:

🡪O id do aluno (char[7] id) : **7 bytes**

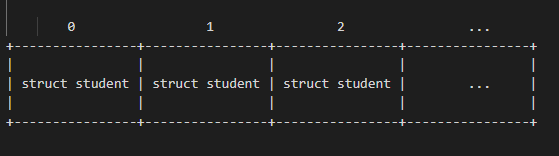
🡪O p\_id do aluno (char[3] p\_id) : **3 bytes**

🡪Um boleano para verificar se o aluno terminou o curso (bool finished) : **1 byte**

🡪Um boleano para verificar se o aluno desistiu do curso (bool quit) : **1 byte**

Tamanho da struct = (7 x 1 byte) + ( 3 x 1 byte) + (1 byte) + (1 byte) = **12 bytes**

A estrutura do array em memoria central será a seguinte:



Como se trata de struct do mesmo tipo do ficheiros, o seu tamanho será o mesmo: **12 bytes.**

# O tamanho do ficheiro é variável, de acordo com o número de alunos existente, mas no pior caso, o ficheiro poderá ter um tamanho máximo de:

# Tamanho máximo do ficheiro:

# Tamanho do ficheiro = 10 000 000 x Tamanho da Struct = 10 000 000 x 12 bytes = 120 000 000 bytes = 120 MB

# 

# Operações

## Introduzir um novo estudante

## Ordenar a lista

## Remover um identificado

## Assinalar que um estudante terminou o curso

## Assinalar o abandono de um estudante

## Obter dados de um país

# Início e fim da execução

Em início de execução criaremos uma HashTable e uma Struct do tipo student, ambas em memória central. Caso esta tabela já exista, estará vazia.

No fim da execução, fechamos os ficheiros existentes, dado que as atualizações aos mesmos serão escritas em disco e libertamos toda a memória alocada ao longo de todo o programa.

# Bibliografia e Webgrafia

🡪[Thomas H. Cormen](http://www.cs.dartmouth.edu/~thc/), [Charles E. Leiserson](http://supertech.csail.mit.edu/~cel/) e [Ronald L. Rivest](http://theory.csail.mit.edu/~rivest/), [Clifford Stein](http://www.columbia.edu/~cs2035/) , *Introduction to Algorithms* , 3ªEdição.

🡪UNESCO, wikipedia : <https://pt.wikipedia.org/wiki/Organiza%C3%A7%C3%A3o_das_Na%C3%A7%C3%B5es_Unidas_para_a_Educa%C3%A7%C3%A3o,_a_Ci%C3%AAncia_e_a_Cultura>

🡪Prof.Vasco Pedro, Estruturas de Dados e Algoritmos 2, plataforma moodle: <https://www.moodle.uevora.pt/1920/course/view.php?id=1390> [2 de Abril de 2020].

🡪Geeks for Geeks, Secondary Memory Acess: <https://www.geeksforgeeks.org/introduction-of-secondary-memory/>

🡪Prof.Dr.Dalius Mazeika, Data Structures and Algorithms:

<http://dma.vgtu.lt/DS/DS12.pdf> [1 Abril de 2020].

🡪Dan Bernstein, djb2 algorithm: <http://www.cse.yorku.ca/~oz/hash.html>