|  |
| --- |
| Algoritmos e Tipos Abstratos de Dados  de Dados 2018/2019  Licenciatura em Engª. Informática |
| Relatório Técnico |

Turma: INF-04

Docente TP: Bruno Silva

Docente PL: Hugo Santos

Alunos:

180221070 – Rafael Trindade

180221109 – Joanã Costa

Índice

[1. TAD Utilizadas 3](#_Toc11458449)

[1.1. List - ArrayList 3](#_Toc11458450)

[1.2. Map – ArrayList 3](#_Toc11458451)

[1.3. Queue – ArrayList 4](#_Toc11458452)

# TAD Utilizadas

Os TAD utilizados para o este projeto foi o ***List***, ***Map*** e ***Queue*** onde cada um teve uma implementação específica na estrutura do programa.

## List - ArrayList

Usado maioritariamente em toda a estrutura do programa onde a estrutura de acesso é baseada por ***ranks*** permitindo assim o acesso a aleatório a qualquer instante a um elemento presente na lista. Esta estrutura foi usada para fazer carregar a lista de pacientes e dados clínicos. O esquema da estrutura de dados segue a seguinte imagem:

typedef struct patient {

int id;

Date birthdate;

char gender;

String hospital;

String district;

ClinicalData clinicalData;

} Patient;

/\* Estrutura dos dados do Paciente\*/

typedef struct clinicalData {

float age;

float bmi;

float glucose;

float insulin;

float mcp1;

int disease\_type;

int clinicalDataCount;

} ClinicalData;

/\* Estrutura dos dados Clínicos\*/

Estrutura ClinicalData: Tem todos os dados clínicos da paciente.

Estrutura Patient: Contém as informações pessoais do paciente para além de conter mais duas estruturas do tipo Date e ClinicalData.

## Map – ArrayList

Um mapa representa um contentor de elementos que implementa uma memória associativa (dicionário) onde são armazenados tuplos, **{chave: valor}.** A chave está apenas associada a um valor para além de não existir chaves duplicadas, há valores duplicados, nuca chaves duplicadas. Os acessos são feitos indicando a **chave**.

Esta estrutura foi escolhida para fazer a média de dados clínicos de cada paciente. A estrutura ficou com a seguinte configuração:

/\* definicao do tipo do chave\*/

typedef String MapKey;

/\* definicao do tipo do valor\*/

typedef ClinicalDataStats MapValue;

typedef struct clinicalDataStats{

float age;

float bmi;

float glucose;

float insulin;

float mcp1;

int disease\_type;

int clinicalDataCount;

} ClinicalDataStats;

/\* Estrutura médias de dados Clínicos\*/

## Queue – ArrayList

A fila é um contentor de elementos que são inseridos e removidos de acordo com o princípio FIFO.

* Os elementos podem ser inseridos na fila a qualquer altura, mas apenas o que se encontra na fila há mais tempo pode ser removido.
* Os elementos são inseridos no final da fila e removidos do início da fila.

Este TAD foi utilizado na implementação para ver os dados clínicos médios num intervalo mínimo e máximo e de uma idade mínima e máxima, onde a cada comando **next** dentro comando **QUEUE** mostrava os dados do paciente a seguir, e assim sucessivamente.