# Biblioteca Gestor Dados Trabalho Prático G14

1.0

Gerado por Doxygen 1.9.8

# Capítulo 1

# Biblioteca GestorDadosIO

A Biblioteca `GestorDadosIO` é responsável por fornecer funcionalidades de importação e exportação de dados a partir de ficheiros. Esta biblioteca é parte integrante do projeto Bem Estar, que tem como objetivo apoiar o bem-estar e cuidados nutricionais.

## 1.1 Funcionalidades

### 1.1.1 Importar Dados

A biblioteca suporta a importação de diferentes tipos de dados, incluindo pacientes, dietas e planos nutricionais. Para importar dados, utilize as seguintes funções:

- `ImportarDadosFichBinario`: Importa dados de um ficheiro binário.
- `ImportarPacientesFichTexto`: Importa dados de pacientes de um ficheiro de texto.
- `ImportarDietasFichTexto`: Importa dados de dietas de um ficheiro de texto.
- `ImportarPlanosFichTexto`: Importa dados de planos de um ficheiro de texto.

### 1.1.2 Exportar Dados

Além da importação, a biblioteca permite exportar dados para ficheiros. A função correspondente é:

- `ExportarDadosFichBinario`: Exporta dados para um ficheiro binário.

# Capítulo 2

# Índice dos ficheiros

## 2.1 Lista de ficheiros

Lista de todos os ficheiros documentados com uma breve descrição:

# Capítulo 3

# Documentação do ficheiro

## 3.1 Referência ao ficheiro ExportarDados.c

Funções para exportar dados para ficheiros binários.

```
#include <stdio.h>
#include "IODadosAPI.h"
#include "../LESI_PI_TP_a28602/Constantes.h"
#include "../LESI_PI_TP_a28602/EstruturaDados.h"
```

### Funções

- int ExportarDadosFichBinario (const char *nomeFicheiro, void *dados, int tamanho, size_t tamanho↩ Elemento)

  *Exporta os dados para o ficheiro binário.*

### 3.1.1 Descrição detalhada

Funções para exportar dados para ficheiros binários.

Este ficheiro contém implementações de funções para exportar dados de pacientes, dietas e planos para ficheiros binários.

**Autor**

Enrique George Rodrigues

**Data**

Created: 28.11.2023

Last modified: 17.12.2023

**Copyright**

### 3.1.2 Documentação das funções

#### 3.1.2.1 ExportarDadosFichBinario()

```
int ExportarDadosFichBinario (
            const char * nomeFicheiro,
            void * dados,
            int tamanho,
            size_t tamanhoElemento )
```

Exporta os dados para o ficheiro binário.

**Parâmetros**

| | |
|---|---|
| *nomeFicheiro* | - O nome do ficheiro binário. |
| *dados* | - Os dados na memória. |
| *tamanho* | - O número de elementos na memória. |
| *tamanhoElemento* | - O tamanho de cada elemento no array. |

**Valores retornados**

| | |
|---|---|
| *0* | - Sucesso na execução (SUCESSO). |
| *12* | - Erro ao abrir ficheiro (ERRO_ABRIR_FICHEIRO). |

## 3.2 Referência ao ficheiro ImportarDados.c

Funções para importar dados a partir de ficheiros (binários e de texto).

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "IODadosAPI.h"
#include "../LESI_PI_TP_a28602/Constantes.h"
#include "../LESI_PI_TP_a28602/EstruturaDados.h"
#include "../LESI_PI_TP_a28602/Utils.h"
```

**Funções**

- int ImportarDadosFichBinario (const char ∗nomeFicheiro, void ∗dados, size_t tamanhoRegisto, int ∗num↩ Registos)

  *Importa os dados a partir do ficheiro binário.*
- int ImportarPacientesFichTexto (FILE ∗file, const char ∗nomeFicheiro, char ∗formatoFicheiro, Paciente pacientes[TAMANHO_MAX_ARRAY], int ∗numPacientes)

  *Importa dados de pacientes de um ficheiro de texto e adiciona-os aos dados guardados, ignorando os recordes duplicados.*
- int ImportarDietasFichTexto (FILE ∗file, const char ∗nomeFicheiro, char ∗formatoFicheiro, Dieta dietas[TAMANHO_MAX_ARRAY], int ∗numDietas)

  *Importa dados de dietas de um ficheiro de texto e adiciona-os aos dados guardados, ignorando os recordes duplicados.*

- int ImportarPlanosFichTexto (FILE ∗file, const char ∗nomeFicheiro, char ∗formatoFicheiro, Plano planos[TAMANHO_MAX_ARRAY], int ∗numPlanos)

  *Importa dados de planos de um ficheiro de texto e adiciona-os aos dados guardados, ignorando os recordes duplicados.*

### 3.2.1 Descrição detalhada

Funções para importar dados a partir de ficheiros (binários e de texto).

Este ficheiro contém implementações de funções para importar dados de pacientes, dietas e planos a partir de ficheiros binários e de texto. As funções são projetadas para adicionar novos dados aos dados já existentes, ignorando registros duplicados quando adequado.

**Autor**

Enrique George Rodrigues

**Data**

Created: 28.11.2023

Last modified: 18.12.2023

**Copyright**



### 3.2.2 Documentação das funções

#### 3.2.2.1 ImportarDadosFichBinario()

```
int ImportarDadosFichBinario (
            const char * nomeFicheiro,
            void * dados,
            size_t tamanhoRegisto,
            int * numRegistos )
```

Importa os dados a partir do ficheiro binário.

**Parâmetros**

| | |
|---|---|
| *nomeFicheiro* | - O nome do ficheiro binário. |
| *dados* | - O array de dados a ser importado. |
| *tamanhoRegisto* | - O tamanho de cada registro em bytes. |
| *numRegistos* | - Um apontador para a variável que guarda o número de registros. |

**Valores retornados**

| | |
|---|---|
| *0* | - Sucesso na execução (SUCESSO). |
| *12* | - Erro ao abrir o ficheiro (ERRO_ABRIR_FICHEIRO). |

**Valores retornados**

| | |
|---|---|
| *13* | - Erro, tamanho máximo excedido. (ERRO_TAMANHO_EXCEDIDO). |

#### 3.2.2.2 ImportarDietasFichTexto()

```
int ImportarDietasFichTexto (
            FILE * file,
            const char * nomeFicheiro,
            char * formatoFicheiro,
            Dieta dietas[TAMANHO_MAX_ARRAY],
            int * numDietas )
```

Importa dados de dietas de um ficheiro de texto e adiciona-os aos dados guardados, ignorando os recordes duplicados.

**Parâmetros**

| | |
|---|---|
| *file* | - Apontador do tipo FILE∗ do ficheiro. Caso não tenha, pode mandar NULL. |
| *nomeFicheiro* | - O nome do ficheiro de texto. |
| *formato* | - O formato dos dados do ficheiro de texto (FORMATO_DIETAS_TXT ou FORMATO_DIETAS_TAB). |
| *dietas* | - O array de dietas. |
| *numDietas* | - Um apontador para a variável que guarda o número de dietas. |

**Nota**

Só é preciso mandar o nome do ficheiro ou o apontador FILE∗.

**Valores retornados**

| | |
|---|---|
| *0* | - Sucesso na execução (SUCESSO). |
| *10* | - Erro formato ficheiro (ERRO_FORMATO_INVALIDO). |
| *12* | - Erro ao abrir o ficheiro (ERRO_ABRIR_FICHEIRO). |

#### 3.2.2.3 ImportarPacientesFichTexto()

```
int ImportarPacientesFichTexto (
            FILE * file,
            const char * nomeFicheiro,
            char * formatoFicheiro,
            Paciente pacientes[TAMANHO_MAX_ARRAY],
            int * numPacientes )
```

Importa dados de pacientes de um ficheiro de texto e adiciona-os aos dados guardados, ignorando os recordes duplicados.

**Parâmetros**

| | |
|---|---|
| *file* | - Apontador do tipo FILE∗ do ficheiro. Caso não tenha, pode mandar NULL. |
| *nomeFicheiro* | - O nome do ficheiro de texto. |
| *formato* | - O formato dos dados do ficheiro de texto (FORMATO_PACIENTES_TXT ou FORMATO_PACIENTES_TAB). |
| *pacientes* | - O array de pacientes. |
| *numPacientes* | - Um apontador para a variável que guarda o número de pacientes. |

**Nota**

Só é preciso mandar o nome do ficheiro ou o apontador FILE∗.

**Valores retornados**

| | |
|---|---|
| *0* | - Sucesso na execução (SUCESSO). |
| *10* | - Erro formato ficheiro (ERRO_FORMATO_INVALIDO). |
| *12* | - Erro ao abrir o ficheiro (ERRO_ABRIR_FICHEIRO). |

#### 3.2.2.4 ImportarPlanosFichTexto()

```
int ImportarPlanosFichTexto (
            FILE * file,
            const char * nomeFicheiro,
            char * formatoFicheiro,
            Plano planos[TAMANHO_MAX_ARRAY],
            int * numPlanos )
```

Importa dados de planos de um ficheiro de texto e adiciona-os aos dados guardados, ignorando os recordes duplicados.

**Parâmetros**

| | |
|---|---|
| *file* | - Apontador do tipo FILE∗ do ficheiro. Caso não tenha, pode mandar NULL. |
| *nomeFicheiro* | - O nome do ficheiro de texto. |
| *planos* | - O array de planos. |
| *numPlanos* | - Um apontador para a variável que guarda o número de planos. |

**Nota**

Só é preciso mandar o nome do ficheiro ou o apontador FILE∗.

**Valores retornados**

| | |
|---|---|
| *0* | - Sucesso na execução (SUCESSO). |
| *8* | - Ficheiro vazio (ERRO_FICHEIRO_VAZIO). |
| *9* | - Erro na leitura do ficheiro (ERRO_LEITURA_FICHEIRO). |
| *12* | - Erro ao abrir o ficheiro (ERRO_ABRIR_FICHEIRO). |
| *13* | - Erro, tamanho máximo excedido. (ERRO_TAMANHO_EXCEDIDO). |

## 3.3 Referência ao ficheiro IODadosAPI.h

Funções de importar e exportar dados dos ficheiros ImportarDados.c e ExportarDados.c.

```
#include "../LESI_PI_TP_a28602/Constantes.h"
#include "../LESI_PI_TP_a28602/EstruturaDados.h"
#include "../LESI_PI_TP_a28602/Utils.h"
```

### Funções

- IO_DADOS_API int ExportarDadosFichBinario (const char *nomeFicheiro, void *dados, int tamanho, size_t tamanhoElemento)

  *Exporta os dados para o ficheiro binário.*
- IO_DADOS_API int ImportarDadosFichBinario (const char *nomeFicheiro, void *dados, size_t tamanho↩ Registo, int *numRegistos)

  *Importa os dados a partir do ficheiro binário.*
- IO_DADOS_API int ImportarPacientesFichTexto (FILE *file, const char *nomeFicheiro, char *formato↩ Ficheiro, Paciente pacientes[TAMANHO_MAX_ARRAY], int *numPacientes)

  *Importa dados de pacientes de um ficheiro de texto e adiciona-os aos dados guardados, ignorando os recordes duplicados.*
- IO_DADOS_API int ImportarDietasFichTexto (FILE *file, const char *nomeFicheiro, char *formatoFicheiro, Dieta dietas[TAMANHO_MAX_ARRAY], int *numDietas)

  *Importa dados de dietas de um ficheiro de texto e adiciona-os aos dados guardados, ignorando os recordes duplicados.*
- IO_DADOS_API int ImportarPlanosFichTexto (FILE *file, const char *nomeFicheiro, char *formatoFicheiro, Plano planos[TAMANHO_MAX_ARRAY], int *numPlanos)

  *Importa dados de planos de um ficheiro de texto e adiciona-os aos dados guardados, ignorando os recordes duplicados.*

### 3.3.1 Descrição detalhada

Funções de importar e exportar dados dos ficheiros ImportarDados.c e ExportarDados.c.

~

**Autor**

Enrique George Rodrigues

**Data**

23.12.2023

**Copyright**



### 3.3.2 Documentação das funções

#### 3.3.2.1 ExportarDadosFichBinario()

```
IO_DADOS_API int ExportarDadosFichBinario (
            const char * nomeFicheiro,
            void * dados,
            int tamanho,
            size_t tamanhoElemento )
```

Exporta os dados para o ficheiro binário.

**Parâmetros**

| | |
|---|---|
| *nomeFicheiro* | - O nome do ficheiro binário. |
| *dados* | - Os dados na memória. |
| *tamanho* | - O número de elementos na memória. |
| *tamanhoElemento* | - O tamanho de cada elemento no array. |

**Valores retornados**

| | |
|---|---|
| *0* | - Sucesso na execução (SUCESSO). |
| *12* | - Erro ao abrir ficheiro (ERRO_ABRIR_FICHEIRO). |

#### 3.3.2.2 ImportarDadosFichBinario()

```
IO_DADOS_API int ImportarDadosFichBinario (
            const char * nomeFicheiro,
            void * dados,
            size_t tamanhoRegisto,
            int * numRegistos )
```

Importa os dados a partir do ficheiro binário.

**Parâmetros**

| | |
|---|---|
| *nomeFicheiro* | - O nome do ficheiro binário. |
| *dados* | - O array de dados a ser importado. |
| *tamanhoRegisto* | - O tamanho de cada registro em bytes. |
| *numRegistos* | - Um apontador para a variável que guarda o número de registros. |

**Valores retornados**

| | |
|---|---|
| *0* | - Sucesso na execução (SUCESSO). |
| *12* | - Erro ao abrir o ficheiro (ERRO_ABRIR_FICHEIRO). |
| *13* | - Erro, tamanho máximo excedido. (ERRO_TAMANHO_EXCEDIDO). |

#### 3.3.2.3 ImportarDietasFichTexto()

```
IO_DADOS_API int ImportarDietasFichTexto (
            FILE * file,
            const char * nomeFicheiro,
            char * formatoFicheiro,
            Dieta dietas[TAMANHO_MAX_ARRAY],
            int * numDietas )
```

Importa dados de dietas de um ficheiro de texto e adiciona-os aos dados guardados, ignorando os recordes duplicados.

**Parâmetros**

| | |
|---|---|
| *file* | - Apontador do tipo FILE∗ do ficheiro. Caso não tenha, pode mandar NULL. |
| *nomeFicheiro* | - O nome do ficheiro de texto. |
| *formato* | - O formato dos dados do ficheiro de texto (FORMATO_DIETAS_TXT ou FORMATO_DIETAS_TAB). |
| *dietas* | - O array de dietas. |
| *numDietas* | - Um apontador para a variável que guarda o número de dietas. |

**Nota**

Só é preciso mandar o nome do ficheiro ou o apontador FILE∗.

**Valores retornados**

| | |
|---|---|
| *0* | - Sucesso na execução (SUCESSO). |
| *10* | - Erro formato ficheiro (ERRO_FORMATO_INVALIDO). |
| *12* | - Erro ao abrir o ficheiro (ERRO_ABRIR_FICHEIRO). |

#### 3.3.2.4 ImportarPacientesFichTexto()

```
IO_DADOS_API int ImportarPacientesFichTexto (
            FILE * file,
            const char * nomeFicheiro,
            char * formatoFicheiro,
            Paciente pacientes[TAMANHO_MAX_ARRAY],
            int * numPacientes )
```

Importa dados de pacientes de um ficheiro de texto e adiciona-os aos dados guardados, ignorando os recordes duplicados.

**Parâmetros**

| | |
|---|---|
| *file* | - Apontador do tipo FILE∗ do ficheiro. Caso não tenha, pode mandar NULL. |
| *nomeFicheiro* | - O nome do ficheiro de texto. |
| *formato* | - O formato dos dados do ficheiro de texto (FORMATO_PACIENTES_TXT ou FORMATO_PACIENTES_TAB). |
| *pacientes* | - O array de pacientes. |
| *numPacientes* | - Um apontador para a variável que guarda o número de pacientes. |

**Nota**

Só é preciso mandar o nome do ficheiro ou o apontador FILE∗.

**Valores retornados**

| | |
|---|---|
| *0* | - Sucesso na execução (SUCESSO). |
| *10* | - Erro formato ficheiro (ERRO_FORMATO_INVALIDO). |
| *12* | - Erro ao abrir o ficheiro (ERRO_ABRIR_FICHEIRO). |

#### 3.3.2.5 ImportarPlanosFichTexto()

```
IO_DADOS_API int ImportarPlanosFichTexto (
            FILE * file,
            const char * nomeFicheiro,
            char * formatoFicheiro,
            Plano planos[TAMANHO_MAX_ARRAY],
            int * numPlanos )
```

Importa dados de planos de um ficheiro de texto e adiciona-os aos dados guardados, ignorando os recordes duplicados.

**Parâmetros**

| | |
|---|---|
| *file* | - Apontador do tipo FILE∗ do ficheiro. Caso não tenha, pode mandar NULL. |
| *nomeFicheiro* | - O nome do ficheiro de texto. |
| *planos* | - O array de planos. |
| *numPlanos* | - Um apontador para a variável que guarda o número de planos. |

**Nota**

Só é preciso mandar o nome do ficheiro ou o apontador FILE∗.

**Valores retornados**

| | |
|---|---|
| *0* | - Sucesso na execução (SUCESSO). |
| *8* | - Ficheiro vazio (ERRO_FICHEIRO_VAZIO). |
| *9* | - Erro na leitura do ficheiro (ERRO_LEITURA_FICHEIRO). |
| *12* | - Erro ao abrir o ficheiro (ERRO_ABRIR_FICHEIRO). |
| *13* | - Erro, tamanho máximo excedido. (ERRO_TAMANHO_EXCEDIDO). |

## 3.4 IODadosAPI.h

Ir para a documentação deste ficheiro.

```
00001
00011 #ifndef IO_DADOS_API_H
00012 #define IO_DADOS_API_H
00013
00014 #include "../LESI_PI_TP_a28602/Constantes.h"
00015 #include "../LESI_PI_TP_a28602/EstruturaDados.h"
00016 #include "../LESI_PI_TP_a28602/Utils.h"
00017
00018 // Verificar se está a ser compilado no Windows
00019 #ifdef _WIN32
00020     // Definir anotações de importar e exportar
00021 #ifdef IO_DADOS_EXPORTS
00022 #define IO_DADOS_API __declspec(dllexport)
00023 #else
00024 #define IO_DADOS_API __declspec(dllimport)
00025 #endif
00026 // Se não for Windows (Linux)
00027 #else
00028 #define IO_DADOS_API // Definir macro como macro vazio
00029 #endif
00030
00042 IO_DADOS_API int ExportarDadosFichBinario(const char* nomeFicheiro, void* dados, int tamanho, size_t
      tamanhoElemento);
00043
```

```
00056 IO_DADOS_API int ImportarDadosFichBinario(const char* nomeFicheiro, void* dados, size_t
      tamanhoRegisto, int* numRegistos);
00057
00074 IO_DADOS_API int ImportarPacientesFichTexto(FILE* file, const char* nomeFicheiro, char*
      formatoFicheiro, Paciente pacientes[TAMANHO_MAX_ARRAY], int* numPacientes);
00075
00092 IO_DADOS_API int ImportarDietasFichTexto(FILE* file, const char* nomeFicheiro, char* formatoFicheiro,
      Dieta dietas[TAMANHO_MAX_ARRAY], int* numDietas);
00093
00111 IO_DADOS_API int ImportarPlanosFichTexto(FILE* file, const char* nomeFicheiro, char* formatoFicheiro,
      Plano planos[TAMANHO_MAX_ARRAY], int* numPlanos);
00112
00113 #endif // IO_DADOS_API_H
```

# Índice