

Sistema de Gestão do Espaço Social

1.0

Gerado por Doxygen 1.13.1

1 Índice das estruturas de dados	1
1.1 Estruturas de dados	1
2 Índice dos ficheiros	3
2.1 Lista de ficheiros	3
3 Documentação da estruturas de dados	5
3.1 Referência à estrutura Ementa	5
3.1.1 Documentação dos campos e atributos	5
3.1.1.1 data	5
3.1.1.2 dia	5
3.1.1.3 prato_carne	5
3.1.1.4 prato_peixe	5
3.2 Referência à estrutura Escolha	6
3.2.1 Documentação dos campos e atributos	6
3.2.1.1 dia	6
3.2.1.2 num_funcionario	6
3.2.1.3 tipo_prato	6
3.3 Referência à estrutura Funcionario	6
3.3.1 Descrição detalhada	7
3.3.2 Documentação dos campos e atributos	7
3.3.2.1 nif [1/2]	7
3.3.2.2 nif [2/2]	7
3.3.2.3 nome	7
3.3.2.4 numero	7
3.3.2.5 telefone [1/2]	7
3.3.2.6 telefone [2/2]	7
3.4 Referência à estrutura NodeEmenta	8
3.4.1 Documentação dos campos e atributos	8
3.4.1.1 ementa	8
3.4.1.2 prox	8
3.5 Referência à estrutura NodeEscolha	8
3.5.1 Documentação dos campos e atributos	8
3.5.1.1 escolha	8
3.5.1.2 prox	8
3.6 Referência à estrutura NodeFunc	9
3.6.1 Descrição detalhada	9
3.6.2 Documentação dos campos e atributos	9
3.6.2.1 func	9
3.6.2.2 prox	9
3.7 Referência à estrutura Prato	9
3.7.1 Descrição detalhada	10
3.7.2 Documentação dos campos e atributos	10

3.7.2.1 calorias	10
3.7.2.2 nome	10
4 Documentação do ficheiro	11
4.1 Referência ao ficheiro src/funcionarios.h	11
4.1.1 Descrição detalhada	12
4.1.2 Documentação dos tipos	12
4.1.2.1 NodeFunc	12
4.1.3 Documentação das funções	12
4.1.3.1 buscarFuncionario()	12
4.1.3.2 carregarFuncionarios()	12
4.1.3.3 criarNoFunc()	13
4.1.3.4 liberarFuncionarios()	13
4.1.3.5 listarFuncionarios()	13
4.1.3.6 salvarFuncionarios()	14
4.2 funcionarios.h	14
4.3 Referência ao ficheiro src/include/estruturas.h	15
4.3.1 Documentação das macros	16
4.3.1.1 MAX_LINE	16
4.3.1.2 MAX_NOME	16
4.3.2 Documentação dos tipos	16
4.3.2.1 NodeEmenta	16
4.3.2.2 NodeEscolha	16
4.3.2.3 NodeFunc	16
4.3.3 Documentação das funções	16
4.3.3.1 calcularMediasCalorias()	16
4.3.3.2 carregarEmentas()	17
4.3.3.3 carregarEscolhas()	17
4.3.3.4 carregarFuncionarios()	17
4.3.3.5 criarNoFunc()	18
4.3.3.6 gerarTabelaSemanal()	18
4.3.3.7 liberarMemoria()	18
4.3.3.8 listarRefeicoesCalorias()	19
4.3.3.9 listarRefeicoesDia()	19
4.3.3.10 listarUtentesOrdenados()	20
4.4 estruturas.h	21
4.5 Referência ao ficheiro src/lib/funcoes.c	22
4.5.1 Descrição detalhada	22
4.5.2 Documentação das funções	23
4.5.2.1 calcularMediasCalorias()	23
4.5.2.2 carregarEmentas()	23
4.5.2.3 carregarEscolhas()	23

4.5.2.4 carregarFuncionarios()	24
4.5.2.5 criarNoFunc()	24
4.5.2.6 gerarTabelaSemanal()	24
4.5.2.7 liberarMemoria()	25
4.5.2.8 listarRefeicoesCalorias()	25
4.5.2.9 listarRefeicoesDia()	25
4.5.2.10 listarUtentesOrdenados()	26
4.6 Referência ao ficheiro src/main.c	26
4.6.1 Descrição detalhada	27
4.6.2 Documentação das funções	27
4.6.2.1 limparBuffer()	27
4.6.2.2 main()	27
4.6.2.3 pausar()	27

Capítulo 1

Índice das estruturas de dados

1.1 Estruturas de dados

Lista das estruturas de dados com uma breve descrição:

Ementa	5
Escolha	6
Funcionario	
Estrutura que representa um funcionário	6
NodeEmenta	8
NodeEscolha	8
NodeFunc	
Estrutura de nó para lista ligada de funcionários	9
Prato	
@	9

Capítulo 2

Índice dos ficheiros

2.1 Lista de ficheiros

Lista de todos os ficheiros com uma breve descrição:

src/ funcionarios.h	
Módulo de gestão de funcionários do sistema de refeições	11
src/ main.c	
Sistema de Gestão do Espaço Social	26
src/include/ estruturas.h	15
src/lib/ funcoes.c	
Implementação das funções do sistema de gestão do espaço social	22

Capítulo 3

Documentação da estruturas de dados

3.1 Referência à estrutura Ementa

```
#include <estruturas.h>
```

Campos de Dados

- char [dia](#) [10]
- char [data](#) [10]
- [Prato](#) [prato_carne](#)
- [Prato](#) [prato_peixe](#)

3.1.1 Documentação dos campos e atributos

3.1.1.1 data

```
char Ementa::data[10]
```

3.1.1.2 dia

```
char Ementa::dia[10]
```

3.1.1.3 prato_carne

```
Prato Ementa::prato_carne
```

3.1.1.4 prato_peixe

```
Prato Ementa::prato_peixe
```

A documentação para esta estrutura foi gerada a partir do seguinte ficheiro:

- [src/include/estruturas.h](#)

3.2 Referência à estrutura Escolha

```
#include <estruturas.h>
```

Campos de Dados

- char [dia](#) [10]
- int [num_funcionario](#)
- char [tipo_prato](#)

3.2.1 Documentação dos campos e atributos

3.2.1.1 dia

```
char Escolha::dia[10]
```

3.2.1.2 num_funcionario

```
int Escolha::num_funcionario
```

3.2.1.3 tipo_prato

```
char Escolha::tipo_prato
```

A documentação para esta estrutura foi gerada a partir do seguinte ficheiro:

- [src/include/estruturas.h](#)

3.3 Referência à estrutura Funcionario

Estrutura que representa um funcionário.

```
#include <funcionarios.h>
```

Campos de Dados

- int [numero](#)
- char [nome](#) [100]
- char [nif](#) [10]
- char [telefone](#) [15]
- int [nif](#)
- int [telefone](#)

3.3.1 Descrição detalhada

Estrutura que representa um funcionário.

3.3.2 Documentação dos campos e atributos

3.3.2.1 nif [1/2]

```
char Funcionario::nif[10]
```

NIF do funcionário

3.3.2.2 nif [2/2]

```
int Funcionario::nif
```

3.3.2.3 nome

```
char Funcionario::nome
```

Nome completo do funcionário

3.3.2.4 numero

```
int Funcionario::numero
```

Número identificador do funcionário

3.3.2.5 telefone [1/2]

```
char Funcionario::telefone[15]
```

Número de telefone do funcionário

3.3.2.6 telefone [2/2]

```
int Funcionario::telefone
```

A documentação para esta estrutura foi gerada a partir dos seguintes ficheiros:

- [src/funcionarios.h](#)
- [src/include/estruturas.h](#)

3.4 Referência à estrutura NodeEmenta

```
#include <estruturas.h>
```

Campos de Dados

- [Ementa ementa](#)
- struct [NodeEmenta](#) * [prox](#)

3.4.1 Documentação dos campos e atributos

3.4.1.1 ementa

```
Ementa NodeEmenta::ementa
```

3.4.1.2 prox

```
struct NodeEmenta* NodeEmenta::prox
```

A documentação para esta estrutura foi gerada a partir do seguinte ficheiro:

- src/include/[estruturas.h](#)

3.5 Referência à estrutura NodeEscolha

```
#include <estruturas.h>
```

Campos de Dados

- [Escolha escolha](#)
- struct [NodeEscolha](#) * [prox](#)

3.5.1 Documentação dos campos e atributos

3.5.1.1 escolha

```
Escolha NodeEscolha::escolha
```

3.5.1.2 prox

```
struct NodeEscolha* NodeEscolha::prox
```

A documentação para esta estrutura foi gerada a partir do seguinte ficheiro:

- src/include/[estruturas.h](#)

3.6 Referência à estrutura NodeFunc

Estrutura de nó para lista ligada de funcionários.

```
#include <funcionarios.h>
```

Campos de Dados

- `Funcionario func`
- `struct NodeFunc * prox`

3.6.1 Descrição detalhada

Estrutura de nó para lista ligada de funcionários.

3.6.2 Documentação dos campos e atributos

3.6.2.1 func

`Funcionario NodeFunc::func`

Dados do funcionário

3.6.2.2 prox

`struct NodeFunc * NodeFunc::prox`

Ponteiro para o próximo nó

A documentação para esta estrutura foi gerada a partir dos seguintes ficheiros:

- `src/funcionarios.h`
- `src/include/estruturas.h`

3.7 Referência à estrutura Prato

@

```
#include <estruturas.h>
```

Campos de Dados

- `char nome [MAX_NOME]`
- `int calorias`

3.7.1 Descrição detalhada

@

3.7.2 Documentação dos campos e atributos

3.7.2.1 calorias

```
int Prato::calorias
```

3.7.2.2 nome

```
char Prato::nome[MAX\_NOME]
```

A documentação para esta estrutura foi gerada a partir do seguinte ficheiro:

- [src/include/estruturas.h](#)

Capítulo 4

Documentação do ficheiro

4.1 Referência ao ficheiro src/funcionarios.h

Módulo de gestão de funcionários do sistema de refeições.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

Estruturas de Dados

- struct [Funcionario](#)
Estrutura que representa um funcionário.
- struct [NodeFunc](#)
Estrutura de nó para lista ligada de funcionários.

Definições de tipos

- typedef struct NodeFunc [NodeFunc](#)
Estrutura de nó para lista ligada de funcionários.

Funções

- [NodeFunc](#) * [criarNoFunc](#) ([Funcionario](#) func)
Cria um novo nó de funcionário.
- [NodeFunc](#) * [carregarFuncionarios](#) (const char *filename, int formato_tab)
Carrega funcionários de um arquivo.
- void [listarFuncionarios](#) ([NodeFunc](#) *lista)
Lista todos os funcionários cadastrados.
- [NodeFunc](#) * [buscarFuncionario](#) ([NodeFunc](#) *lista, int numero)
Busca um funcionário pelo número.
- void [liberarFuncionarios](#) ([NodeFunc](#) *lista)
Libera a memória alocada para a lista de funcionários.
- int [salvarFuncionarios](#) ([NodeFunc](#) *lista, const char *filename, int formato_bin)
Salva a lista de funcionários em arquivo.

4.1.1 Descrição detalhada

Módulo de gestão de funcionários do sistema de refeições.

Autor

[Seu Nome]

Data

[Data]

4.1.2 Documentação dos tipos

4.1.2.1 NodeFunc

```
typedef struct NodeFunc NodeFunc
```

Estrutura de nó para lista ligada de funcionários.

4.1.3 Documentação das funções

4.1.3.1 buscarFuncionario()

```
NodeFunc * buscarFuncionario (  
    NodeFunc * lista,  
    int numero)
```

Busca um funcionário pelo número.

Parâmetros

<i>lista</i>	Ponteiro para o início da lista de funcionários
<i>numero</i>	Número do funcionário a ser buscado

Retorna

Ponteiro para o nó do funcionário encontrado ou NULL se não encontrado

4.1.3.2 carregarFuncionarios()

```
NodeFunc * carregarFuncionarios (  
    const char * filename,  
    int formato_tab)
```

Carrega funcionários de um arquivo.

Parâmetros

<i>filename</i>	Nome do arquivo a ser lido
<i>formato_tab</i>	Flag indicando se o arquivo usa tabulação como separador

Retorna

Ponteiro para o início da lista de funcionários ou NULL em caso de erro

4.1.3.3 criarNoFunc()

```
NodeFunc * criarNoFunc (  
    Funcionario func)
```

Cria um novo nó de funcionário.

Parâmetros

<i>func</i>	Estrutura <code>Funcionario</code> com os dados do funcionário
-------------	--

Retorna

Ponteiro para o novo nó criado ou NULL em caso de erro

Cria um novo nó de funcionário.

Parâmetros

<i>func</i>	Estrutura com os dados do funcionário
-------------	---------------------------------------

Retorna

NodeFunc* Ponteiro para o novo nó criado ou NULL em caso de falha

Aloca memória dinamicamente para um novo nó e inicializa seus campos

4.1.3.4 liberarFuncionarios()

```
void liberarFuncionarios (  
    NodeFunc * lista)
```

Libera a memória alocada para a lista de funcionários.

Parâmetros

<i>lista</i>	Ponteiro para o início da lista de funcionários
--------------	---

4.1.3.5 listarFuncionarios()

```
void listarFuncionarios (  
    NodeFunc * lista)
```

Lista todos os funcionários cadastrados.

Parâmetros

<i>lista</i>	Ponteiro para o início da lista de funcionários
--------------	---

4.1.3.6 salvarFuncionarios()

```
int salvarFuncionarios (  
    NodeFunc * lista,  
    const char * filename,  
    int formato_bin)
```

Salva a lista de funcionários em arquivo.

Parâmetros

<i>lista</i>	Ponteiro para o início da lista de funcionários
<i>filename</i>	Nome do arquivo onde salvar
<i>formato_bin</i>	Flag indicando se deve salvar em formato binário

Retorna

1 se sucesso, 0 se erro

4.2 funcionarios.h

[Ir para a documentação deste ficheiro.](#)

```
00001  
00007  
00008 #ifndef FUNCIONARIOS_H  
00009 #define FUNCIONARIOS_H  
00010  
00011 #include <stdio.h>  
00012 #include <stdlib.h>  
00013 #include <string.h>  
00014  
00018 typedef struct {  
00019     int numero;  
00020     char nome[100];  
00021     char nif[10];  
00022     char telefone[15];  
00023 } Funcionario;  
00024  
00028 typedef struct NodeFunc {  
00029     Funcionario func;  
00030     struct NodeFunc* prox;  
00031 } NodeFunc;  
00032  
00038 NodeFunc* criarNoFunc(Funcionario func);  
00039  
00046 NodeFunc* carregarFuncionarios(const char* filename, int formato_tab);  
00047  
00052 void listarFuncionarios(NodeFunc* lista);  
00053  
00060 NodeFunc* buscarFuncionario(NodeFunc* lista, int numero);  
00061  
00066 void liberarFuncionarios(NodeFunc* lista);  
00067  
00075 int salvarFuncionarios(NodeFunc* lista, const char* filename, int formato_bin);  
00076  
00077 #endif /* FUNCIONARIOS_H */
```

4.3 Referência ao ficheiro src/include/estruturas.h

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

Estruturas de Dados

- struct [Funcionario](#)
Estrutura que representa um funcionário.
- struct [Prato](#)
@
- struct [Ementa](#)
- struct [Escolha](#)
- struct [NodeFunc](#)
Estrutura de nó para lista ligada de funcionários.
- struct [NodeEmenta](#)
- struct [NodeEscolha](#)

Macros

- #define [MAX_NOME](#) 50
- #define [MAX_LINE](#) 256

Definições de tipos

- typedef struct NodeFunc [NodeFunc](#)
- typedef struct NodeEmenta [NodeEmenta](#)
- typedef struct NodeEscolha [NodeEscolha](#)

Funções

- [NodeFunc](#) * [criarNoFunc](#) ([Funcionario](#) func)
Cria um novo nó para a lista de funcionários.
- [NodeFunc](#) * [carregarFuncionarios](#) (const char *filename)
Carrega dados dos funcionários a partir de um arquivo.
- [NodeEmenta](#) * [carregarEmentas](#) (const char *filename)
Carrega dados das ementas a partir de um arquivo.
- [NodeEscolha](#) * [carregarEscolhas](#) (const char *filename)
Carrega as escolhas dos utentes a partir de um arquivo.
- void [listarRefeicoesDia](#) ([NodeFunc](#) *funcionarios, [NodeEmenta](#) *ementas, [NodeEscolha](#) *escolhas, const char *dia)
Lista todas as refeições de um dia específico.
- void [listarUtentesOrdenados](#) ([NodeFunc](#) *funcionarios, [NodeEscolha](#) *escolhas)
Lista os utentes ordenados por número de refeições.
- void [listarRefeicoesCalorias](#) ([NodeFunc](#) *funcionarios, [NodeEmenta](#) *ementas, [NodeEscolha](#) *escolhas, int num_funcionario, const char *data_inicio, const char *data_fim)
Lista as refeições e calorias consumidas por um utente.

- void `calcularMediasCalorias` (`NodeEmenta` *ementas, `NodeEscolha` *escolhas, const char *data_inicio, const char *data_fim)
Calcula as médias de calorias consumidas por dia.
- void `gerarTabelaSemanal` (`NodeFunc` *funcionarios, `NodeEmenta` *ementas, `NodeEscolha` *escolhas, int num_funcionario)
Gera uma tabela semanal de refeições para um utente.
- void `liberarMemoria` (`NodeFunc` *funcionarios, `NodeEmenta` *ementas, `NodeEscolha` *escolhas)
Libera toda a memória alocada para as listas.

4.3.1 Documentação das macros

4.3.1.1 MAX_LINE

```
#define MAX_LINE 256
```

4.3.1.2 MAX_NOME

```
#define MAX_NOME 50
```

4.3.2 Documentação dos tipos

4.3.2.1 NodeEmenta

```
typedef struct NodeEmenta NodeEmenta
```

4.3.2.2 NodeEscolha

```
typedef struct NodeEscolha NodeEscolha
```

4.3.2.3 NodeFunc

```
typedef struct NodeFunc NodeFunc
```

4.3.3 Documentação das funções

4.3.3.1 calcularMediasCalorias()

```
void calcularMediasCalorias (  
    NodeEmenta * ementas,  
    NodeEscolha * escolhas,  
    const char * data_inicio,  
    const char * data_fim)
```

Calcula as médias de calorias consumidas por dia.

Parâmetros

<i>ementas</i>	Lista de ementas
<i>escolhas</i>	Lista de escolhas
<i>data_inicio</i>	Data inicial do período
<i>data_fim</i>	Data final do período

Calcula e exibe a média de calorias consumidas por dia no período especificado

4.3.3.2 carregarEmentas()

```
NodeEmenta * carregarEmentas (
    const char * filename)
```

Carrega dados das ementas a partir de um arquivo.

Parâmetros

<i>filename</i>	Nome do arquivo contendo as ementas
-----------------	-------------------------------------

Retorna

NodeEmenta* Ponteiro para o início da lista de ementas ou NULL em caso de erro

Lê o arquivo CSV com dados das ementas e cria uma lista ligada O arquivo deve estar no formato: dia;data;prato↵_carne;calorias_carne;prato_peixe;calorias_peixe

4.3.3.3 carregarEscolhas()

```
NodeEscolha * carregarEscolhas (
    const char * filename)
```

Carrega as escolhas dos utentes a partir de um arquivo.

Parâmetros

<i>filename</i>	Nome do arquivo contendo as escolhas
-----------------	--------------------------------------

Retorna

NodeEscolha* Ponteiro para o início da lista de escolhas ou NULL em caso de erro

Lê o arquivo CSV com as escolhas e cria uma lista ligada O arquivo deve estar no formato: dia;num_↵funcionario;tipo_prato

4.3.3.4 carregarFuncionarios()

```
NodeFunc * carregarFuncionarios (
    const char * filename)
```

Carrega dados dos funcionários a partir de um arquivo.

Parâmetros

<i>filename</i>	Nome do arquivo contendo os dados dos funcionários
-----------------	--

Retorna

NodeFunc* Ponteiro para o início da lista de funcionários ou NULL em caso de erro

Lê o arquivo CSV com dados dos funcionários e cria uma lista ligada. O arquivo deve estar no formato: numero;nome;nif;telefone

4.3.3.5 criarNoFunc()

```
NodeFunc * criarNoFunc (
    Funcionario func)
```

Cria um novo nó para a lista de funcionários.

Cria um novo nó de funcionário.

Parâmetros

<i>func</i>	Estrutura com os dados do funcionário
-------------	---------------------------------------

Retorna

NodeFunc* Ponteiro para o novo nó criado ou NULL em caso de falha

Aloca memória dinamicamente para um novo nó e inicializa seus campos

4.3.3.6 gerarTabelaSemanal()

```
void gerarTabelaSemanal (
    NodeFunc * funcionarios,
    NodeEmenta * ementas,
    NodeEscolha * escolhas,
    int num_funcionario)
```

Gera uma tabela semanal de refeições para um utente.

Parâmetros

<i>funcionarios</i>	Lista de funcionários
<i>ementas</i>	Lista de ementas
<i>escolhas</i>	Lista de escolhas
<i>num_funcionario</i>	Número do funcionário

Gera e exibe uma tabela com as escolhas de refeições do funcionário para a semana

4.3.3.7 liberarMemoria()

```
void liberarMemoria (
    NodeFunc * funcionarios,
    NodeEmenta * ementas,
    NodeEscolha * escolhas)
```

Libera toda a memória alocada para as listas.

Parâmetros

<i>funcionarios</i>	Ponteiro para a lista de funcionários
<i>ementas</i>	Ponteiro para a lista de ementas
<i>escolhas</i>	Ponteiro para a lista de escolhas

Percorre todas as listas liberando a memória alocada para cada nó

4.3.3.8 listarRefeicoesCalorias()

```
void listarRefeicoesCalorias (  
    NodeFunc * funcionarios,  
    NodeEmenta * ementas,  
    NodeEscolha * escolhas,  
    int num_funcionario,  
    const char * data_inicio,  
    const char * data_fim)
```

Lista as refeições e calorias consumidas por um utente.

Parâmetros

<i>funcionarios</i>	Lista de funcionários
<i>ementas</i>	Lista de ementas
<i>escolhas</i>	Lista de escolhas
<i>num_funcionario</i>	Número do funcionário
<i>data_inicio</i>	Data inicial do período
<i>data_fim</i>	Data final do período

Mostra todas as refeições escolhidas por um funcionário em um período específico

4.3.3.9 listarRefeicoesDia()

```
void listarRefeicoesDia (  
    NodeFunc * funcionarios,  
    NodeEmenta * ementas,  
    NodeEscolha * escolhas,  
    const char * dia)
```

Lista todas as refeições de um dia específico.

Parâmetros

<i>funcionarios</i>	Lista de funcionários
<i>ementas</i>	Lista de ementas
<i>escolhas</i>	Lista de escolhas
<i>dia</i>	Dia para listar as refeições

Mostra as escolhas de refeições feitas pelos funcionários em um determinado dia

4.3.3.10 listarUtentesOrdenados()

```
void listarUtentesOrdenados (  
    NodeFunc * funcionarios,  
    NodeEscolha * escolhas)
```

Lista os utentes ordenados por número de refeições.

Parâmetros

<i>funcionarios</i>	Lista de funcionários
<i>escolhas</i>	Lista de escolhas

Ordena e exibe os funcionários com base no número de refeições escolhidas

4.4 estruturas.h

[Ir para a documentação deste ficheiro.](#)

```

00001 #ifndef ESTRUTURAS_H
00002 #define ESTRUTURAS_H
00003
00004 #include <stdio.h>
00005 #include <stdlib.h>
00006 #include <string.h>
00007
00008 #define MAX_NOME 50
00009 #define MAX_LINE 256
00010
00011 // Estrutura para Funcionário
00012 typedef struct {
00013     int numero;
00014     char nome[MAX_NOME];
00015     int nif;
00016     int telefone;
00017 } Funcionario;
00023 typedef struct {
00024     char nome[MAX_NOME];
00025     int calorias;
00026 } Prato;
00027
00028 // Estrutura para Ementa
00029 typedef struct {
00030     char dia[10];
00031     char data[10];
00032     Prato prato_carne;
00033     Prato prato_peixe;
00034 } Ementa;
00035
00036 // Estrutura para Escolha
00037 typedef struct {
00038     char dia[10];
00039     int num_funcionario;
00040     char tipo_prato; // 'C' para carne, 'P' para peixe
00041 } Escolha;
00042
00043 // Nó para lista de funcionários
00044 typedef struct NodeFunc {
00045     Funcionario func;
00046     struct NodeFunc* prox;
00047 } NodeFunc;
00048
00049 // Nó para lista de ementas
00050 typedef struct NodeEmenta {
00051     Ementa ementa;
00052     struct NodeEmenta* prox;
00053 } NodeEmenta;
00054
00055 // Nó para lista de escolhas
00056 typedef struct NodeEscolha {
00057     Escolha escolha;
00058     struct NodeEscolha* prox;
00059 } NodeEscolha;
00060
00061 // Protótipos das funções
00062 NodeFunc* criarNoFunc(Funcionario func);
00063 NodeFunc* carregarFuncionarios(const char* filename);
00064 NodeEmenta* carregarEmentas(const char* filename);
00065 NodeEscolha* carregarEscolhas(const char* filename);
00066 void listarRefeicoesDia(NodeFunc* funcionarios, NodeEmenta* ementas, NodeEscolha* escolhas, const
char* dia);
00067 void listarUtentesOrdenados(NodeFunc* funcionarios, NodeEscolha* escolhas);
00068 void listarRefeicoesCalorias(NodeFunc* funcionarios, NodeEmenta* ementas, NodeEscolha* escolhas, int
num_funcionario, const char* data_inicio, const char* data_fim);

```

```
00069 void calcularMediasCalorias(NodeEmenta* ementas, NodeEscolha* escolhas, const char* data_inicio, const
char* data_fim);
00070 void gerarTabelaSemanal(NodeFunc* funcionarios, NodeEmenta* ementas, NodeEscolha* escolhas, int
num_funcionario);
00071 void liberarMemoria(NodeFunc* funcionarios, NodeEmenta* ementas, NodeEscolha* escolhas);
00072
00073 #endif
```

4.5 Referência ao ficheiro src/lib/funcoes.c

Implementação das funções do sistema de gestão do espaço social.

```
#include "estruturas.h"
```

Funções

- **NodeFunc** * **criarNoFunc** (**Funcionario** func)
Cria um novo nó para a lista de funcionários.
- **NodeFunc** * **carregarFuncionarios** (const char *filename)
Carrega dados dos funcionários a partir de um arquivo.
- **NodeEmenta** * **carregarEmentas** (const char *filename)
Carrega dados das ementas a partir de um arquivo.
- **NodeEscolha** * **carregarEscolhas** (const char *filename)
Carrega as escolhas dos utentes a partir de um arquivo.
- void **liberarMemoria** (**NodeFunc** *funcionarios, **NodeEmenta** *ementas, **NodeEscolha** *escolhas)
Libera toda a memória alocada para as listas.
- void **listarRefeicoesDia** (**NodeFunc** *funcionarios, **NodeEmenta** *ementas, **NodeEscolha** *escolhas, const char *dia)
Lista todas as refeições de um dia específico.
- void **listarUtentesOrdenados** (**NodeFunc** *funcionarios, **NodeEscolha** *escolhas)
Lista os utentes ordenados por número de refeições.
- void **listarRefeicoesCalorias** (**NodeFunc** *funcionarios, **NodeEmenta** *ementas, **NodeEscolha** *escolhas, int num_funcionario, const char *data_inicio, const char *data_fim)
Lista as refeições e calorias consumidas por um utente.
- void **calcularMediasCalorias** (**NodeEmenta** *ementas, **NodeEscolha** *escolhas, const char *data_inicio, const char *data_fim)
Calcula as médias de calorias consumidas por dia.
- void **gerarTabelaSemanal** (**NodeFunc** *funcionarios, **NodeEmenta** *ementas, **NodeEscolha** *escolhas, int num_funcionario)
Gera uma tabela semanal de refeições para um utente.

4.5.1 Descrição detalhada

Implementação das funções do sistema de gestão do espaço social.

Autor

[Seu Nome]

Data

2025-01-04

4.5.2 Documentação das funções

4.5.2.1 calcularMediasCalorias()

```
void calcularMediasCalorias (
    NodeEmenta * ementas,
    NodeEscolha * escolhas,
    const char * data_inicio,
    const char * data_fim)
```

Calcula as médias de calorias consumidas por dia.

Parâmetros

<i>ementas</i>	Lista de ementas
<i>escolhas</i>	Lista de escolhas
<i>data_inicio</i>	Data inicial do período
<i>data_fim</i>	Data final do período

Calcula e exibe a média de calorias consumidas por dia no período especificado

4.5.2.2 carregarEmentas()

```
NodeEmenta * carregarEmentas (
    const char * filename)
```

Carrega dados das ementas a partir de um arquivo.

Parâmetros

<i>filename</i>	Nome do arquivo contendo as ementas
-----------------	-------------------------------------

Retorna

NodeEmenta* Ponteiro para o início da lista de ementas ou NULL em caso de erro

Lê o arquivo CSV com dados das ementas e cria uma lista ligada O arquivo deve estar no formato: dia;data;prato↵_carne;calorias_carne;prato_peixe;calorias_peixe

4.5.2.3 carregarEscolhas()

```
NodeEscolha * carregarEscolhas (
    const char * filename)
```

Carrega as escolhas dos utentes a partir de um arquivo.

Parâmetros

<i>filename</i>	Nome do arquivo contendo as escolhas
-----------------	--------------------------------------

Retorna

NodeEscolha* Ponteiro para o início da lista de escolhas ou NULL em caso de erro

Lê o arquivo CSV com as escolhas e cria uma lista ligada O arquivo deve estar no formato: dia;num_↵funcionario;tipo_prato

4.5.2.4 carregarFuncionarios()

```
NodeFunc * carregarFuncionarios (
    const char * filename)
```

Carrega dados dos funcionários a partir de um arquivo.

Parâmetros

<i>filename</i>	Nome do arquivo contendo os dados dos funcionários
-----------------	--

Retorna

NodeFunc* Ponteiro para o início da lista de funcionários ou NULL em caso de erro

Lê o arquivo CSV com dados dos funcionários e cria uma lista ligada O arquivo deve estar no formato: numero;nome;nif;telefone

4.5.2.5 criarNoFunc()

```
NodeFunc * criarNoFunc (
    Funcionario func)
```

Cria um novo nó para a lista de funcionários.

Cria um novo nó de funcionário.

Parâmetros

<i>func</i>	Estrutura com os dados do funcionário
-------------	---------------------------------------

Retorna

NodeFunc* Ponteiro para o novo nó criado ou NULL em caso de falha

Aloca memória dinamicamente para um novo nó e inicializa seus campos

4.5.2.6 gerarTabelaSemanal()

```
void gerarTabelaSemanal (
    NodeFunc * funcionarios,
    NodeEmenta * ementas,
    NodeEscolha * escolhas,
    int num_funcionario)
```

Gera uma tabela semanal de refeições para um utente.

Parâmetros

<i>funcionarios</i>	Lista de funcionários
<i>ementas</i>	Lista de ementas
<i>escolhas</i>	Lista de escolhas
<i>num_funcionario</i>	Número do funcionário

Gera e exibe uma tabela com as escolhas de refeições do funcionário para a semana

4.5.2.7 liberarMemoria()

```
void liberarMemoria (
    NodeFunc * funcionarios,
    NodeEmenta * ementas,
    NodeEscolha * escolhas)
```

Libera toda a memória alocada para as listas.

Parâmetros

<i>funcionarios</i>	Ponteiro para a lista de funcionários
<i>ementas</i>	Ponteiro para a lista de ementas
<i>escolhas</i>	Ponteiro para a lista de escolhas

Percorre todas as listas liberando a memória alocada para cada nó

4.5.2.8 listarRefeicoesCalorias()

```
void listarRefeicoesCalorias (
    NodeFunc * funcionarios,
    NodeEmenta * ementas,
    NodeEscolha * escolhas,
    int num_funcionario,
    const char * data_inicio,
    const char * data_fim)
```

Lista as refeições e calorias consumidas por um utente.

Parâmetros

<i>funcionarios</i>	Lista de funcionários
<i>ementas</i>	Lista de ementas
<i>escolhas</i>	Lista de escolhas
<i>num_funcionario</i>	Número do funcionário
<i>data_inicio</i>	Data inicial do período
<i>data_fim</i>	Data final do período

Mostra todas as refeições escolhidas por um funcionário em um período específico

4.5.2.9 listarRefeicoesDia()

```
void listarRefeicoesDia (
    NodeFunc * funcionarios,
    NodeEmenta * ementas,
    NodeEscolha * escolhas,
    const char * dia)
```

Lista todas as refeições de um dia específico.

Parâmetros

<i>funcionarios</i>	Lista de funcionários
<i>ementas</i>	Lista de ementas
<i>escolhas</i>	Lista de escolhas
<i>dia</i>	Dia para listar as refeições

Mostra as escolhas de refeições feitas pelos funcionários em um determinado dia

4.5.2.10 listarUtentesOrdenados()

```
void listarUtentesOrdenados (  
    NodeFunc * funcionarios,  
    NodeEscolha * escolhas)
```

Lista os utentes ordenados por número de refeições.

Parâmetros

<i>funcionarios</i>	Lista de funcionários
<i>escolhas</i>	Lista de escolhas

Ordena e exibe os funcionários com base no número de refeições escolhidas

4.6 Referência ao ficheiro src/main.c

Sistema de Gestão do Espaço Social.

```
#include "estruturas.h"  
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>
```

Funções

- void `limparBuffer` ()
Limpa o buffer do teclado.
- void `pausar` ()
Pausa a execução do programa até que o usuário pressione Enter.
- int `main` ()
Função principal do programa.

4.6.1 Descrição detalhada

Sistema de Gestão do Espaço Social.

Autor

[Seu Nome]

Data

2025-01-04

4.6.2 Documentação das funções

4.6.2.1 limparBuffer()

```
void limparBuffer ()
```

Limpa o buffer do teclado.

Remove caracteres residuais do buffer de entrada

4.6.2.2 main()

```
int main ()
```

Função principal do programa.

Retorna

0 em caso de sucesso, outro valor em caso de erro

Implementa o menu principal do sistema com as seguintes funcionalidades:

- Carregamento de dados de funcionários
- Carregamento de ementas semanais
- Carregamento de escolhas dos utentes
- Listagem de refeições por dia
- Listagem de utentes ordenados
- Consulta de refeições por utente
- Cálculo de médias de calorias

4.6.2.3 pausar()

```
void pausar ()
```

Pausa a execução do programa até que o usuário pressione Enter.

Exibe uma mensagem e aguarda input do usuário

