Laboratório de Informática

2025

Gerado por Doxygen 1.14.0

Capítulo 1

Índice das estruturas de dados

1.1 Estruturas de dados

Lista das estruturas de dados com uma breve descrição:

Capítulo 2

Índice dos ficheiros

2.1 Lista de ficheiros

Lista de todos os ficheiros com uma breve descrição:

Capítulo 3

Documentação da estruturas de dados

3.1 Referência à estrutura AtualizacaoValor

Campos de Dados

- float valor
- char data_inicio [11]

3.1.1 Documentação dos campos e atributos

3.1.1.1 data_inicio

```
char AtualizacaoValor::data_inicio[11]
```

3.1.1.2 valor

```
float AtualizacaoValor::valor
```

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

- src/lib/funcoes.c

3.2 Referência à estrutura ControleRefeicao

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

Campos de Dados

- float valor
- char data_ultima_atualizacao [11]

3.2.1 Descrição detalhada

Estrutura para armazenar o valor da refeição

Esta estrutura é utilizada para controlar o valor monetário das refeições, armazenando o valor atual e a data da última atualização.

3.2.2 Documentação dos campos e atributos

3.2.2.1 data_ultima_atualizacao

```
char ControleRefeicao::data_ultima_atualizacao[11]
```

Data da última atualização do valor

3.2.2.2 valor

```
float ControleRefeicao::valor
```

Valor da refeição

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

- src/include/estruturas.h

3.3 Referência à estrutura Ementa

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

Campos de Dados

- char dia [10]
- char data [11]
- Prato prato_carne
- Prato prato_peixe
- Prato prato_dieta
- Prato prato_vegetariano

3.3.1 Descrição detalhada

Estrutura para Ementa

A estrutura Ementa contém a descrição de uma ementa diária, incluindo o dia da semana, a data e os pratos disponíveis: carne, peixe, dieta e vegetariano.

3.3.2 Documentação dos campos e atributos

3.3.2.1 data

```
char Ementa::data[11]
```

Data no formato AAAA-MM-DD

3.3.2.2 dia

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

Dia da semana

3.3.2.3 prato_carne

```
Prato Ementa::prato_carne
```

Prato de carne

3.3.2.4 prato_dieta

```
Prato Ementa::prato_dieta
```

Prato da dieta

3.3.2.5 prato_peixe

```
Prato Ementa::prato_peixe
```

Prato de peixe

3.3.2.6 prato_vegetariano

```
Prato Ementa::prato_vegetariano
```

Prato vegetariano

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

- src/include/estruturas.h

3.4 Referência à estrutura Escolha

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

Campos de Dados

- char dia [10]
- int num_funcionario
- char tipo_prato

3.4.1 Descrição detalhada

Estrutura para Escolha

A estrutura Escolha regista a escolha de um funcionário para um determinado dia, incluindo o tipo de prato selecionado (carne, peixe, etc.).

3.4.2 Documentação dos campos e atributos

3.4.2.1 dia

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

Dia da semana

3.4.2.2 num_funcionario

```
int Escolha::num_funcionario
```

Número do funcionário que fez a escolha

3.4.2.3 tipo_prato

```
char Escolha::tipo_prato
```

Tipo de prato escolhido: 'C' para carne, 'P' para peixe, etc.

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

- src/include/estruturas.h

3.5 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

3.5.1 Descrição detalhada

Estrutura que representa um funcionário.

3.5.2 Documentação dos campos e atributos

3.5.2.1 nif [1/2]

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

NIF do funcionário

3.5.2.2 nif [2/2]

```
int Funcionario::nif
```

Número de Identificação Fiscal do funcionário

3.5.2.3 nome

```
char Funcionario::nome
```

Nome completo do funcionário

Nome do funcionário

3.5.2.4 numero

```
int Funcionario::numero
```

Número identificador do funcionário

Número do funcionário

3.5.2.5 telefone [1/2]

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

Número de telefone do funcionário

3.5.2.6 telefone [2/2]

```
int Funcionario::telefone
```

Número de telefone do funcionário

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

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

3.6 Referência à estrutura NodeEmenta

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

Campos de Dados

- Ementa ementa
- struct NodeEmenta * prox

3.6.1 Documentação dos campos e atributos

3.6.1.1 ementa

```
Ementa NodeEmenta::ementa
```

Estrutura da ementa

3.6.1.2 prox

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

Ponteiro para o próximo nó da lista

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

- src/include/estruturas.h

3.7 Referência à estrutura NodeEscolha

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

Campos de Dados

- Escolha escolha
- struct NodeEscolha * prox

3.7.1 Documentação dos campos e atributos

3.7.1.1 escolha

```
Escolha NodeEscolha::escolha
```

Estrutura da escolha

3.7.1.2 prox

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

Ponteiro para o próximo nó da lista

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

- src/include/estruturas.h

3.8 Referência à estrutura NodeFunc

Estrutura de nó para lista ligada de funcionários.

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

Campos de Dados

- Funcionario func
- struct NodeFunc * prox

3.8.1 Descrição detalhada

Estrutura de nó para lista ligada de funcionários.

3.8.2 Documentação dos campos e atributos

3.8.2.1 func

```
Funcionario NodeFunc::func
```

Dados do funcionário

Estrutura do funcionário

3.8.2.2 prox

```
struct NodeFunc * NodeFunc::prox
```

Ponteiro para o próximo nó

Ponteiro para o próximo nó da lista

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

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

3.9 Referência à estrutura Prato

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

Campos de Dados

- char nome [MAX_NOME]
- int calorias

3.9.1 Descrição detalhada

Estrutura para Prato

Esta estrutura representa um prato disponível na ementa, contendo informações sobre o seu nome e o número de calorias que possui.

3.9.2 Documentação dos campos e atributos

3.9.2.1 calorias

```
int Prato::calorias
```

Número de calorias do prato

3.9.2.2 nome

```
char Prato::nome[MAX_NOME]
```

Nome do prato

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 libertarFuncionarios (NodeFunc ∗lista)

  Liberta 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]

Este ficheiro define as estruturas e protótipos de funções para a gestão de funcionários, incluindo operações de criação, carregamento, listagem, busca, libertação de memória e salvamento de dados.

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

	lista	Ponteiro para o início da lista de funcionários
	numero	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

	filename	Nome do arquivo a ser lido
	formato_tab	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

	func	Estrutura Funcionario 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

	func	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 libertarFuncionarios()

```
void libertarFuncionarios (
            NodeFunc * lista)
```

Liberta a memória alocada para a lista de funcionários.

Parâmetros

	lista	Ponteiro para o início da lista de funcionários

4.1.3.5 listarFuncionarios()

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

Lista todos os funcionários cadastrados.

Parâmetros

	lista	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

	lista	Ponteiro para o início da lista de funcionários
	filename	Nome do arquivo onde salvar
	formato_bin	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
00011
00012 #ifndef FUNCIONARIOS_H
00013 #define FUNCIONARIOS_H
00014
00015 #include <stdio.h>
00016 #include <stdlib.h>
00017 #include <string.h>
00018
00022 typedef struct {
00023     int numero;
00024     char nome[100];
00025     char nif[10];
00026     char telefone[15];
00027 } Funcionario;
00028
00032 typedef struct NodeFunc {
00033     Funcionario func;
00034     struct NodeFunc* prox;
00035 } NodeFunc;
00036
00042 NodeFunc* criarNoFunc(Funcionario func);
00043
00050 NodeFunc* carregarFuncionarios(const char* filename, int formato_tab);
00051
00056 void listarFuncionarios(NodeFunc* lista);
00057
00064 NodeFunc* buscarFuncionario(NodeFunc* lista, int numero);
00065
00070 void libertarFuncionarios(NodeFunc* lista);
00071
00079 int salvarFuncionarios(NodeFunc* lista, const char* filename, int formato_bin);
00080
00081 #endif /* FUNCIONARIOS_H */
```

4.3 Referência ao ficheiro src/include/estruturas.h

Definições de estruturas e protótipos para o sistema de gestão do espaço social.

```
#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 ControleRefeicao
- struct NodeFunc

  Estrutura de nó para lista ligada de funcionários.
- struct NodeEmenta
- struct NodeEscolha

Macros

- #define MAX_NOME 50
- #define MAX_LINE 256
- #define LIMITE_REFEICOES_DIARIO 200

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 a lista de funcionários a partir de um ficheiro.
- NodeEmenta ∗ carregarEmentas (const char ∗filename)

  Carrega a lista de ementas a partir de um ficheiro.
- NodeEscolha ∗ carregarEscolhas (const char ∗filename)

  Carrega a lista de escolhas a partir de um ficheiro.
- void atualizarEmentas (NodeEmenta ∗nova_lista)

  Atualiza as ementas com uma nova lista.
- NodeEmenta ∗ obterEmentas ()

  Obtém a lista atual de ementas.
- void listarRefeicoesDia (NodeFunc ∗funcionarios, NodeEmenta ∗ementas, NodeEscolha ∗escolhas, const char ∗dia)

Lista as refeições de um determinado dia.

- void listarRefeicoesServidasSemana (NodeFunc *funcionarios, NodeEmenta *ementas, NodeEscolha *escolhas)

  Lista as refeições servidas durante uma semana.
- void listarRefeicoesCalorias (NodeFunc *funcionarios, NodeEmenta *ementas, NodeEscolha *escolhas, int num_funcionario, const char *data_inicio, const char *data_fim)

  Lista as refeições com um determinado número de calorias.
- void calcularMediasCalorias (NodeEmenta *ementas, NodeEscolha *escolhas, const char *data_inicio, const char *data_fim)

  Calcula as médias de calorias das refeições.
- void gerarTabelaSemanal (NodeFunc *funcionarios, NodeEmenta *ementas, NodeEscolha *escolhas, int num_funcionario)

  Gera uma tabela semanal de refeições.
- void gerarTabelaSemanalDetalhada (NodeFunc *funcionarios, NodeEmenta *ementas, NodeEscolha *escolhas, int num_funcionario)

  Gera uma tabela semanal detalhada de refeições.
- void libertarMemoria (NodeFunc *funcionarios, NodeEmenta *ementas, NodeEscolha *escolhas)

  Liberta a memória utilizada pelas listas.
- void inicializarControleRefeicoes ()

  Inicializa o controlo de refeições.
- int validarRefeicaoDia (const char *dia)

  Valida um dia para registo de refeição.
- void atualizarValorRefeicao (float novo_valor)

  Atualiza o valor da refeição.
- void mostrarValorRefeicao ()

  Mostra o valor atual da refeição.
- void limparBuffer ()

  Limpa o buffer do teclado.
- void pausar ()

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

Variáveis

- NodeFunc * funcionarios
- NodeEmenta * ementas
- NodeEscolha * escolhas

4.3.1 Descrição detalhada

Definições de estruturas e protótipos para o sistema de gestão do espaço social.

Este ficheiro contém as definições das estruturas de dados principais (Funcionário, Prato, Ementa, Escolha, listas ligadas) e os protótipos das funções utilizadas em todo o sistema.

4.3.2 Documentação das macros

4.3.2.1 LIMITE_REFEICOES_DIARIO

```
#define LIMITE_REFEICOES_DIARIO 200
```

4.3.2.2 MAX_LINE

```
#define MAX_LINE 256
```

4.3.2.3 MAX_NOME

```
#define MAX_NOME 50
```

4.3.3 Documentação dos tipos

4.3.3.1 NodeEmenta

```
typedef struct NodeEmenta NodeEmenta
```

4.3.3.2 NodeEscolha

```
typedef struct NodeEscolha NodeEscolha
```

4.3.3.3 NodeFunc

```
typedef struct NodeFunc NodeFunc
```

4.3.4 Documentação das funções

4.3.4.1 atualizarEmentas()

```
void atualizarEmentas (
            NodeEmenta * nova_lista)
```

Atualiza as ementas com uma nova lista.

Esta função substitui a lista atual de ementas por uma nova lista, libertando a memória da lista antiga.

Parâmetros

	nova_lista	Ponteiro para a nova lista de ementas

Atualiza as ementas com uma nova lista.

Parâmetros

	nova_lista	Nova lista de ementas

4.3.4.2 atualizarValorRefeicao()

```
void atualizarValorRefeicao (
            float novo_valor)
```

Atualiza o valor da refeição.

Esta função atualiza o valor monetário da refeição, armazenando a nova avaliação e a data da atualização.

Parâmetros

	novo_valor	Novo valor da refeição
	novo_valor	Novo valor da refeição

4.3.4.3 calcularMediasCalorias()

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

Calcula as médias de calorias das refeições.

Esta função calcula e exibe as médias de calorias das refeições servidas dentro de um intervalo de datas especificado.

Parâmetros

	ementas	Lista de ementas
	escolhas	Lista de escolhas
	data_inicio	Data de início do intervalo
	data_fim	Data de fim do intervalo

Calcula as médias de calorias das refeições.

Parâmetros

	ementas	Lista de ementas
	escolhas	Lista de escolhas
	data_inicio	Data inicial do período
	data_fim	Data final do período

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

4.3.4.4 carregarEmentas()

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

Carrega a lista de ementas a partir de um ficheiro.

Esta função lê um ficheiro com a informação das ementas e cria a lista de ementas correspondente.

Parâmetros

	filename	Nome do ficheiro a ser lido

Retorna

Ponteiro para a lista de ementas carregada

Carrega a lista de ementas a partir de um ficheiro.

Parâmetros

	filename	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

4.3.4.5 carregarEscolhas()

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

Carrega a lista de escolhas a partir de um ficheiro.

Esta função lê um ficheiro com a informação das escolhas e cria a lista de escolhas correspondente.

Parâmetros

	filename	Nome do ficheiro a ser lido

Retorna

Ponteiro para a lista de escolhas carregada

Carrega a lista de escolhas a partir de um ficheiro.

Parâmetros

	filename	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

4.3.4.6 carregarFuncionarios()

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

Carrega a lista de funcionários a partir de um ficheiro.

Esta função lê um ficheiro com a informação dos funcionários e cria a lista de funcionários correspondente.

Parâmetros

	filename	Nome do ficheiro a ser lido

Retorna

Ponteiro para a lista de funcionários carregada

Carrega a lista de funcionários a partir de um ficheiro.

Parâmetros

	filename	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

4.3.4.7 criarNoFunc()

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

Cria um novo nó para a lista de funcionários.

Esta função aloca memória para um novo nó da lista de funcionários, inicializa os seus valores e devolve o ponteiro para o nó criado.

Parâmetros

	func	Estrutura do funcionário a ser armazenado no nó

Retorna

Ponteiro para o novo nó da lista de funcionários

Cria um novo nó de funcionário.

Parâmetros

	func	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

4.3.4.8 gerarTabelaSemanal()

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

Gera uma tabela semanal de refeições.

Esta função gera uma tabela com todas as refeições servidas durante uma semana, para um determinado funcionário.

Parâmetros

	funcionarios	Lista de funcionários
	ementas	Lista de ementas
	escolhas	Lista de escolhas
	num_funcionario	Número do funcionário

Gera uma tabela semanal de refeições.

Parâmetros

	funcionarios	Lista de funcionários
	ementas	Lista de ementas
	escolhas	Lista de escolhas
	num_funcionario	Número do funcionário

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

4.3.4.9 gerarTabelaSemanalDetalhada()

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

Gera uma tabela semanal detalhada de refeições.

Esta função gera uma tabela detalhada com todas as refeições servidas durante uma semana, incluindo informações sobre calorias, para um determinado funcionário.

Parâmetros

	funcionarios	Lista de funcionários
	ementas	Lista de ementas
	escolhas	Lista de escolhas
	num_funcionario	Número do funcionário

Gera uma tabela semanal detalhada de refeições.

Parâmetros

	funcionarios	Lista de funcionários
	ementas	Lista de ementas
	escolhas	Lista de escolhas
	num_funcionario	Número do funcionário

Exibe uma tabela com colunas para cada tipo de prato e calorias, linhas de 2ª a 6ª feira

4.3.4.10 inicializarControleRefeicoes()

```
void inicializarControleRefeicoes ()
```

Inicializa o controlo de refeições.

Esta função inicializa o sistema de controlo de refeições, definindo o valor padrão da refeição e a data da última atualização.

Inicializa o controlo de refeições.

Configura o valor inicial da refeição e reseta o contador diário

4.3.4.11 libertarMemoria()

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

Liberta a memória utilizada pelas listas.

Esta função liberta a memória alocada para as listas de funcionários, ementas e escolhas.

Parâmetros

	funcionarios	Lista de funcionários
	ementas	Lista de ementas
	escolhas	Lista de escolhas

Liberta a memória utilizada pelas listas.

Parâmetros

	funcionarios	Ponteiro para a lista de funcionários
	ementas	Ponteiro para a lista de ementas
	escolhas	Ponteiro para a lista de escolhas

Percorre todas as listas libertando a memória alocada para cada nó

4.3.4.12 limparBuffer()

```
void limparBuffer ()
```

Limpa o buffer do teclado.

Remove caracteres residuais do buffer de entrada

4.3.4.13 listarRefeicoesCalorias()

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

Lista as refeições com um determinado número de calorias.

Esta função exibe todas as refeições que têm um número de calorias dentro de um intervalo especificado, para um determinado funcionário e intervalo de datas.

Parâmetros

funcionarios	Lista de funcionários
ementas	Lista de ementas
escolhas	Lista de escolhas
num_funcionario	Número do funcionário
data_inicio	Data de início do intervalo
data_fim	Data de fim do intervalo

4.3.4.14 listarRefeicoesDia()

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

Lista as refeições de um determinado dia.

Esta função exibe todas as refeições servidas num determinado dia, para todos os funcionários.

Parâmetros

funcionarios	Lista de funcionários
ementas	Lista de ementas
escolhas	Lista de escolhas
dia	Dia para o qual se pretende listar as refeições

Lista as refeições de um determinado dia.

Parâmetros

funcionarios	Lista de funcionários
ementas	Lista de ementas
escolhas	Lista de escolhas
dia	Dia para listar as refeições

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

4.3.4.15 listarRefeicoesServidasSemana()

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

Lista as refeições servidas durante uma semana.

Esta função exibe todas as refeições servidas durante uma semana, para todos os funcionários.

Parâmetros

	funcionarios	Lista de funcionários
	ementas	Lista de ementas
	escolhas	Lista de escolhas

Lista as refeições servidas durante uma semana.

Parâmetros

	funcionarios	Lista de funcionários
	ementas	Lista de ementas
	escolhas	Lista de escolhas

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

4.3.4.16 mostrarValorRefeicao()

```
void mostrarValorRefeicao ()
```

Mostra o valor atual da refeição.

Esta função exibe o valor monetário atual da refeição.

4.3.4.17 obterEmentas()

```
NodeEmenta * obterEmentas ()
```

Obtém a lista atual de ementas.

Esta função devolve a lista atual de ementas.

Retorna

Ponteiro para a lista atual de ementas

Obtém a lista atual de ementas.

Retorna

Ponteiro para a lista de ementas

4.3.4.18 pausar()

```
void pausar ()
```

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

Exibe uma mensagem e aguarda input do usuário

4.3.4.19 validarRefeicaoDia()

```
int validarRefeicaoDia (
            const char * dia)
```

Valida um dia para registo de refeição.

Esta função verifica se um determinado dia é válido para o registo de uma refeição, com base nas regras do sistema.

Parâmetros

	dia	Dia da semana a ser validado

Retorna

1 se o dia for válido, 0 caso contrário

Valida um dia para registo de refeição.

Parâmetros

	dia	Dia da semana para validação

Retorna

1 se pode servir, 0 se não pode

4.3.5 Documentação das variáveis

4.3.5.1 ementas

```
NodeEmenta* ementas  [extern]
```

4.3.5.2 escolhas

```
NodeEscolha* escolhas  [extern]
```

4.3.5.3 funcionarios

```
NodeFunc* funcionarios  [extern]
```

4.4 estruturas.h

Ir para a documentação deste ficheiro.

```
00001
00009
00010 #ifndef ESTRUTURAS_H
00011 #define ESTRUTURAS_H
00012
00013 #include <stdio.h>
00014 #include <stdlib.h>
00015 #include <string.h>
00016
00017 #define MAX_NOME 50
00018 #define MAX_LINE 256
00019 #define LIMITE_REFEICOES_DIARIO 200
00020
00021 // Estrutura para Funcionário
00022 typedef struct {
00023     int numero;
```

```
00024     char nome[MAX_NOME];
00025     int nif;
00026     int telefone;
00027 } Funcionario;
00028
00034 typedef struct {
00035     char nome[MAX_NOME];
00036     int calorias;
00037 } Prato;
00038
00045 typedef struct {
00046     char dia[10];
00047     char data[11];
00048     Prato prato_carne;
00049     Prato prato_peixe;
00050     Prato prato_dieta;
00051     Prato prato_vegetariano;
00052 } Ementa;
00053
00059 typedef struct {
00060     char dia[10];
00061     int num_funcionario;
00062     char tipo_prato;
00063 } Escolha;
00064
00070 struct ControleRefeicao {
00071     float valor;
00072     char data_ultima_atualizacao[11];
00073 };
00074
00075 // Nó para lista de funcionários
00076 typedef struct NodeFunc {
00077     Funcionario func;
00078     struct NodeFunc* prox;
00079 } NodeFunc;
00080
00081 // Nó para lista de ementas
00082 typedef struct NodeEmenta {
00083     Ementa ementa;
00084     struct NodeEmenta* prox;
00085 } NodeEmenta;
00086
00087 // Nó para lista de escolhas
00088 typedef struct NodeEscolha {
00089     Escolha escolha;
00090     struct NodeEscolha* prox;
00091 } NodeEscolha;
00092
00101 NodeFunc* criarNoFunc(Funcionario func);
00102
00111 NodeFunc* carregarFuncionarios(const char* filename);
00112
00121 NodeEmenta* carregarEmentas(const char* filename);
00122
00131 NodeEscolha* carregarEscolhas(const char* filename);
00132
00140 void atualizarEmentas(NodeEmenta* nova_lista);
00141
00148 NodeEmenta* obterEmentas();
00149
00160 void listarRefeicoesDia(NodeFunc* funcionarios, NodeEmenta* ementas, NodeEscolha* escolhas, const
      char* dia);
00161
00171 void listarRefeicoesServidasSemana(NodeFunc* funcionarios, NodeEmenta* ementas, NodeEscolha*
      escolhas);
00172
00186 void listarRefeicoesCalorias(NodeFunc* funcionarios, NodeEmenta* ementas, NodeEscolha* escolhas, int
      num_funcionario, const char* data_inicio, const char* data_fim);
00187
00198 void calcularMediasCalorias(NodeEmenta* ementas, NodeEscolha* escolhas, const char* data_inicio, const
      char* data_fim);
00199
00210 void gerarTabelaSemanal(NodeFunc* funcionarios, NodeEmenta* ementas, NodeEscolha* escolhas, int
      num_funcionario);
00211
00223 void gerarTabelaSemanalDetalhada(NodeFunc* funcionarios, NodeEmenta* ementas, NodeEscolha* escolhas,
      int num_funcionario);
00224
00234 void libertarMemoria(NodeFunc* funcionarios, NodeEmenta* ementas, NodeEscolha* escolhas);
00235
00241 void inicializarControleRefeicoes();
00242
00251 int validarRefeicaoDia(const char* dia);
00252
00260 void atualizarValorRefeicao(float novo_valor);
00261
00266 void mostrarValorRefeicao();
```

```
00267
00268 // Protótipos de funções uteis
00269 void limparBuffer();
00270 void pausar();
00271
00272 extern NodeFunc* funcionarios;
00273 extern NodeEmenta* ementas;
00274 extern NodeEscolha* escolhas;
00275
00276 #endif
```

4.5 Referência ao ficheiro src/lib/funcoes.c

Implementação das funções do sistema de gestão do espaço social.

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

Estruturas de Dados

- struct AtualizacaoValor

Macros

- #define MAX_ATUALIZACOES 10

Funções

- void atualizarEmentas (NodeEmenta *nova_lista)

  Atualiza a variável global das ementas.
- NodeEmenta * obterEmentas ()

  Obtém a lista global de ementas.
- int compararDatas (const char *data1, const char *data2)

  Compara duas datas no formato DD.MM.AAAA.
- void limparBuffer ()

  Limpa o buffer do teclado.
- void pausar ()

  Pausa a execução do programa até que o usuário pressione Enter.
- void inicializarControleRefeicoes ()

  Inicializa o controle de refeições.
- int validarRefeicaoDia (const char *dia)

  Valida se é possível servir mais uma refeição no dia.
- void atualizarValorRefeicao (float novo_valor)

  Atualiza o valor da refeição.
- void mostrarValorRefeicao ()

  Mostra o valor atual da refeição.
- 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 libertarMemoria (NodeFunc *funcionarios, NodeEmenta *ementas, NodeEscolha *escolhas)

  Liberta 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 listarRefeicoesServidasSemana (NodeFunc *funcionarios, NodeEmenta *ementas, NodeEscolha *escolhas)

  Lista o consumo semanal de refeições, dos utentes ordenados por ordem decrescente,.
- void listarRefeicoesCalorias (NodeFunc *funcionarios, NodeEmenta *ementas, NodeEscolha *escolhas, int num_funcionario, const char *data_inicio, const char *data_fim)

  Lista as refeições com um determinado número de calorias.
- 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 gerarTabelaSemanalDetalhada (NodeFunc *funcionarios, NodeEmenta *ementas, NodeEscolha *escolhas, int num_funcionario)

  Gera uma tabela semanal detalhada da ementa usufruída por um utente.
- int stricmp (const char *s1, const char *s2)

Variáveis

- struct ControleRefeicao valorRefeicao = {5.00, "29.06.2025"}
- int refeicoes_servidas_por_dia = 0
- char dia_atual [11] = ""
- NodeFunc * funcionarios = NULL
- NodeEmenta * ementas = NULL
- NodeEscolha * escolhas = NULL

4.5.1 Descrição detalhada

Implementação das funções do sistema de gestão do espaço social.

Autor

[G36 - Epoca Especial]

Data

2025-06-20

Este ficheiro contém a implementação das funções auxiliares e principais para a gestão de funcionários, ementas, escolhas e controlo de refeições. Inclui manipulação de listas ligadas, operações de leitura e escrita, validação de dados e controlo de limites diários.

4.5.2 Documentação das macros

4.5.2.1 MAX_ATUALIZACOES

```
#define MAX_ATUALIZACOES 10
```

4.5.3 Documentação das funções

4.5.3.1 atualizarEmentas()

```
void atualizarEmentas (
            NodeEmenta * nova_lista)
```

Atualiza a variável global das ementas.

Atualiza as ementas com uma nova lista.

Parâmetros

	nova_lista	Nova lista de ementas

4.5.3.2 atualizarValorRefeicao()

```
void atualizarValorRefeicao (
            float novo_valor)
```

Atualiza o valor da refeição.

Parâmetros

	novo_valor	Novo valor da refeição

4.5.3.3 calcularMediasCalorias()

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

Calcula as médias de calorias consumidas por dia.

Calcula as médias de calorias das refeições.

Parâmetros

	ementas	Lista de ementas
	escolhas	Lista de escolhas
	data_inicio	Data inicial do período
	data_fim	Data final do período

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

4.5.3.4 carregarEmentas()

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

Carrega dados das ementas a partir de um arquivo.

Carrega a lista de ementas a partir de um ficheiro.

Parâmetros

	filename	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

4.5.3.5 carregarEscolhas()

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

Carrega as escolhas dos utentes a partir de um arquivo.

Carrega a lista de escolhas a partir de um ficheiro.

Parâmetros

	filename	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

4.5.3.6 carregarFuncionarios()

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

Carrega dados dos funcionários a partir de um arquivo.

Carrega a lista de funcionários a partir de um ficheiro.

Parâmetros

	filename	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

4.5.3.7 compararDatas()

```
int compararDatas (
            const char * data1,
            const char * data2)
```

Compara duas datas no formato DD.MM.AAAA.

Parâmetros

	data1	Primeira data
	data2	Segunda data

Retorna

1 se data1 > data2, -1 se data1 < data2, 0 se são iguais

4.5.3.8 criarNoFunc()

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

Cria um novo nó para a lista de funcionários.

Cria um novo nó de funcionário.

Parâmetros

	func	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

4.5.3.9 gerarTabelaSemanal()

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

Gera uma tabela semanal de refeições para um utente.

Gera uma tabela semanal de refeições.

Parâmetros

	funcionarios	Lista de funcionários
	ementas	Lista de ementas
	escolhas	Lista de escolhas
	num_funcionario	Número do funcionário

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

4.5.3.10 gerarTabelaSemanalDetalhada()

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

Gera uma tabela semanal detalhada da ementa usufruída por um utente.

Gera uma tabela semanal detalhada de refeições.

Parâmetros

	funcionarios	Lista de funcionários
	ementas	Lista de ementas
	escolhas	Lista de escolhas
	num_funcionario	Número do funcionário

Exibe uma tabela com colunas para cada tipo de prato e calorias, linhas de 2ª a 6ª feira

4.5.3.11 inicializarControleRefeicoes()

```
void inicializarControleRefeicoes ()
```

Inicializa o controle de refeições.

Inicializa o controlo de refeições.

Configura o valor inicial da refeição e reseta o contador diário

4.5.3.12 libertarMemoria()

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

Liberta toda a memória alocada para as listas.

Liberta a memória utilizada pelas listas.

Parâmetros

	funcionarios	Ponteiro para a lista de funcionários
	ementas	Ponteiro para a lista de ementas
	escolhas	Ponteiro para a lista de escolhas

Percorre todas as listas libertando a memória alocada para cada nó

4.5.3.13 limparBuffer()

```
void limparBuffer ()
```

Limpa o buffer do teclado.

Remove caracteres residuais do buffer de entrada

4.5.3.14 listarRefeicoesCalorias()

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

Lista as refeições com um determinado número de calorias.

Esta função exibe todas as refeições que têm um número de calorias dentro de um intervalo especificado, para um determinado funcionário e intervalo de datas.

Parâmetros

	funcionarios	Lista de funcionários
	ementas	Lista de ementas
	escolhas	Lista de escolhas
	num_funcionario	Número do funcionário
	data_inicio	Data de início do intervalo
	data_fim	Data de fim do intervalo

4.5.3.15 listarRefeicoesDia()

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

Lista todas as refeições de um dia específico.

Lista as refeições de um determinado dia.

Parâmetros

funcionarios	Lista de funcionários
ementas	Lista de ementas
escolhas	Lista de escolhas
dia	Dia para listar as refeições

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

4.5.3.16 listarRefeicoesServidasSemana()

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

Lista o consumo semanal de refeições, dos utentes ordenados por ordem decrescente,.

Lista as refeições servidas durante uma semana.

Parâmetros

funcionarios	Lista de funcionários
ementas	Lista de ementas
escolhas	Lista de escolhas

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

4.5.3.17 mostrarValorRefeicao()

```
void mostrarValorRefeicao ()
```

Mostra o valor atual da refeição.

Esta função exibe o valor monetário atual da refeição.

4.5.3.18 obterEmentas()

```
NodeEmenta * obterEmentas ()
```

Obtém a lista global de ementas.

Obtém a lista atual de ementas.

Retorna

Ponteiro para a lista de ementas

4.5.3.19 pausar()

```
void pausar ()
```

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

Exibe uma mensagem e aguarda input do usuário

4.5.3.20 stricmp()

```
int stricmp (
            const char * s1,
            const char * s2)
```

4.5.3.21 validarRefeicaoDia()

```
int validarRefeicaoDia (
            const char * dia)
```

Valida se é possível servir mais uma refeição no dia.

Valida um dia para registo de refeição.

Parâmetros

	dia	Dia da semana para validação

Retorna

1 se pode servir, 0 se não pode

4.5.4 Documentação das variáveis

4.5.4.1 dia_atual

```
char dia_atual[11] = ""
```

4.5.4.2 ementas

```
NodeEmenta* ementas = NULL
```

4.5.4.3 escolhas

```
NodeEscolha* escolhas = NULL
```

4.5.4.4 funcionarios

```
NodeFunc* funcionarios = NULL
```

4.5.4.5 refeicoes_servidas_por_dia

```
int refeicoes_servidas_por_dia = 0
```

4.5.4.6 valorRefeicao

```
struct ControleRefeicao valorRefeicao = {5.00, "29.06.2025"}
```

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

- int main ()

  Função principal do programa.

4.6.1 Descrição detalhada

Sistema de Gestão do Espaço Social.

Autor

[G36 - Epoca Especial]

Data

2025-06-20

Este ficheiro implementa o menu principal do sistema, permitindo ao utilizador carregar dados, consultar informações e efetuar operações relacionadas com a gestão de refeições e utentes.

4.6.2 Documentação das funções

4.6.2.1 main()

```
int main ()
```

Função principal do programa.

Retorna

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

Esta função apresenta o menu principal ao utilizador, permitindo-lhe:

- Carregar dados dos funcionários
- Carregar a ementa semanal
- Carregar as escolhas dos utentes
- Listar refeições requeridas por dia
- Apresentar o resumo semanal de consumo por utente
- Consultar refeições de um utente
- Calcular médias de calorias
- Gerar tabela semanal detalhada de um utente
- Sair do programa

O menu é apresentado em ciclo até o utilizador escolher a opção de sair. $<$ Variável que armazena a opção escolhida pelo utilizador no menu principal

Carrega os dados dos funcionários a partir do ficheiro

Se a leitura for bem-sucedida, atualiza a lista global de funcionários.

Carrega a ementa semanal a partir do ficheiro

Se a leitura for bem-sucedida, atualiza a lista global de ementas.

Carrega as escolhas dos utentes a partir do ficheiro

Se a leitura for bem-sucedida, atualiza a lista global de escolhas.

Lista as refeições requeridas por dia

Solicita ao utilizador o dia da semana e apresenta as refeições desse dia. Só é possível se todos os dados estiverem carregados.

$<$ Armazena o dia da semana introduzido pelo utilizador

Apresenta o resumo semanal de consumo por utente

Só é possível se todos os dados estiverem carregados.

Consulta as refeições de um utente

Permite ao utilizador consultar refeições por intervalo de datas ou semana completa. Só é possível se todos os dados estiverem carregados.

$<$ Número do funcionário e opção de consulta

$<$ Datas para consulta por intervalo

Calcula as médias de calorias num intervalo de datas

Só é possível se os dados de ementas e escolhas estiverem carregados.

$<$ Datas para o cálculo das médias

Gera a tabela semanal detalhada de um utente

Só é possível se todos os dados estiverem carregados.

$<$ Número do funcionário a consultar

Sai do programa, libertando toda a memória alocada

Opção inválida introduzida pelo utilizador