

Projeto ED: Portagens

Generated by Doxygen 1.13.2



<b>1 Sistema de Gestão de Portagens</b>	<b>1</b>
1.1 Funcionalidades Principais	1
1.1.1 1. Gestão de Donos	1
1.1.2 2. Gestão de Veículos	1
1.1.3 3. Passagens/Viagens	1
1.1.4 4. Estatísticas e Rankings	2
1.1.5 5. Sistema de Infrações	2
1.1.6 6. Análise de Dados	2
1.1.7 7. Exportação de Dados	2
1.1.8 8. Configurações	2
1.2 Requisitos Técnicos	3
1.3 Compilação	3
1.3.1 Em Windows	3
1.3.2 Em Linux	3
1.4 Autores	3
1.5 Agradecimentos	3
<b>2 Class Index</b>	<b>5</b>
2.1 Class List	5
<b>3 File Index</b>	<b>7</b>
3.1 File List	7
<b>4 Class Documentation</b>	<b>9</b>
4.1 Bdados Struct Reference	9
4.2 Carro Struct Reference	10
4.3 CodPostal Struct Reference	11
4.4 Data Struct Reference	11
4.5 Dict Struct Reference	12
4.6 Distancias Struct Reference	12
4.7 Dono Struct Reference	13
4.8 Lista Struct Reference	13
4.9 no Struct Reference	14
4.10 noHash Struct Reference	14
4.11 noRankings Struct Reference	15
4.12 Passagem Struct Reference	15
4.13 Ranking Struct Reference	16
4.14 Sensor Struct Reference	16
4.15 Viagem Struct Reference	17
<b>5 File Documentation</b>	<b>19</b>
5.1 bdados.h	19
5.2 carro.h	19
5.3 configs.h	21

5.4 constantes.h . . . . .	21
5.5 dados.h . . . . .	22
5.6 distancias.h . . . . .	22
5.7 dono.h . . . . .	23
5.8 menus.h . . . . .	23
5.9 passagens.h . . . . .	24
5.10 sensores.h . . . . .	25
5.11 structsGenericas.h . . . . .	25
5.12 uteis.h . . . . .	26
5.13 validacoes.h . . . . .	27

# Chapter 1

## Sistema de Gestão de Portagens

Sistema integrado para gestão de portagens rodoviárias desenvolvido em C, permitindo o controlo de veículos, donos, passagens e análise de dados estatísticos e rankings com as estruturas de dados devidas.

Aplicação de consola desenvolvida no âmbito da cadeira "Estrutura de Dados" da licenciatura de Engenharia Informática da ESTGV no 2º semestre do ano letivo 2024-2025

### 1.1 Funcionalidades Principais

#### 1.1.1 1. Gestão de Donos

- Registar novos proprietários
- Listar donos por ordem alfabética
- Listar donos por NIF

#### 1.1.2 2. Gestão de Veículos

- Registar novos veículos
- Mudar proprietário
- Listar veículos por:
  - Matrícula
  - Marca
  - Modelo
- Consultar veículos por período de circulação

#### 1.1.3 3. Passagens/Viagens

- Registar passagens em portagens
- Consultar histórico de viagens

#### 1.1.4 4. Estatísticas e Rankings

- [Ranking](#) de KMs por período
- [Ranking](#) de KMs por marca
- [Ranking](#) de infrações
- Velocidades médias:
  - Por condutor
  - Por marca
  - Por código postal

#### 1.1.5 5. Sistema de Infrações

- Consulta de veículos com infrações
- Análise de infrações por período

#### 1.1.6 6. Análise de Dados

- Monitorização da memória utilizada em tempo real
- Outras estatísticas de marcas de veículos

#### 1.1.7 7. Exportação de Dados

- Formato CSV
- Formato XML
- Formato HTML
- Formato TXT (apenas em listagens)

#### 1.1.8 8. Configurações

- Autosave configurável
- Personalização de listagens
- Gestão de ficheiros de dados
- Backup e restauro de dados

## 1.2 Requisitos Técnicos

- Sistema operativo: Windows/Unix
- Compilador C compatível (Recomendado GCC, usando o C23)
- Suporte a UTF-8 para caracteres especiais
- 300MB de espaço mínimo em disco (1GB, pelo menos, recomendado)
- 1GB RAM no mínimo (Recomendado 8 ou 16 GB para um funcionamento fluido)

## 1.3 Compilação

### 1.3.1 Em Windows

- Compilar com: `gcc -Wall -Wextra -g -O0 -std=c23 -o filename main.c uteis.c validacoes.c sensores.c passagens.c menus.c structsGenericas.c dono.c distancias.c dados.c carro.c bdados.c configs.c`
- Testado em ambiente Windows 11 Home 23H2 (64 bits) com o compilador GCC em C23
- Especificações do computador utilizado:
  - 16GB RAM DDR4 (3200MHz)
  - Processador Intel i5-12450H 2.00GHz 12MB Cache
  - SSD 512GB

### 1.3.2 Em Linux

- Compilar com: `gcc -std=c2x -Wall -Wextra -o FILENAME main.c uteis.c validacoes.c sensores.c passagens.c menus.c structsGenericas.c dono.c distancias.c dados.c carro.c bdados.c configs.c -D_XOPEN_SOURCE=700`
- Testado em ambiente Linux Ubuntu 20.04.6 LTS (Garantir que estamos a usar gcc13 (C23) - Testado na versão 13.1.0)
- Especificações do computador (VM):
  - 8GB RAM
  - 7 CPU's no VirtualBox

## 1.4 Autores

- Hugo Afonso (Número mecanográfico 30032)
- Mateus Silva (Número mecanográfico 29989)
- Guilherme Pais (Número mecanográfico 30007)
- Todas as versões do programa estão disponíveis no seguinte repositório do github: <https://github.com/huger6/ProjetoED>
- Notar que o repositório apenas estará público durante o período de entrega do projeto

## 1.5 Agradecimentos

- Gostávamos de expressar os nossos agradecimentos ao professor Carlos Simões, e especialmente ao Francisco Morgado, por toda a ajuda durante a realização deste projeto.





## Chapter 2

# Class Index

### 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Bdados	9
Carro	10
CodPostal	11
Data	11
Dict	12
Distancias	12
Dono	13
Lista	13
no	14
noHash	14
noRankings	15
Passagem	15
Ranking	16
Sensor	16
Viagem	17



## Chapter 3

# File Index

### 3.1 File List

Here is a list of all documented files with brief descriptions:

<a href="#">bdados.h</a>	19
<a href="#">carro.h</a>	19
<a href="#">configs.h</a>	21
<a href="#">constantes.h</a>	21
<a href="#">dados.h</a>	22
<a href="#">distancias.h</a>	22
<a href="#">dono.h</a>	23
<a href="#">menus.h</a>	23
<a href="#">passagens.h</a>	24
<a href="#">sensores.h</a>	25
<a href="#">structsGenericas.h</a>	25
<a href="#">uteis.h</a>	26
<a href="#">validacoes.h</a>	27

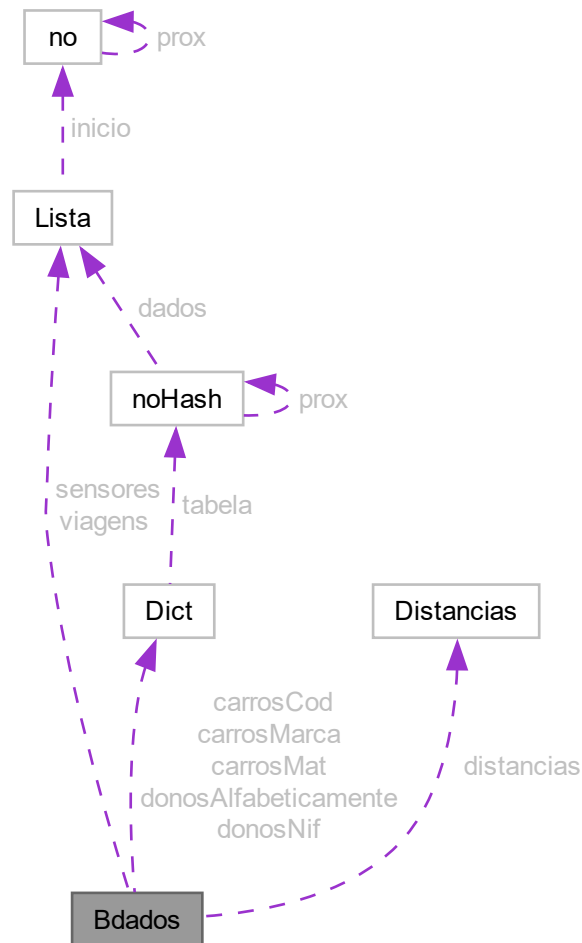


## Chapter 4

# Class Documentation

### 4.1 Bdados Struct Reference

Collaboration diagram for Bdados:



### Public Attributes

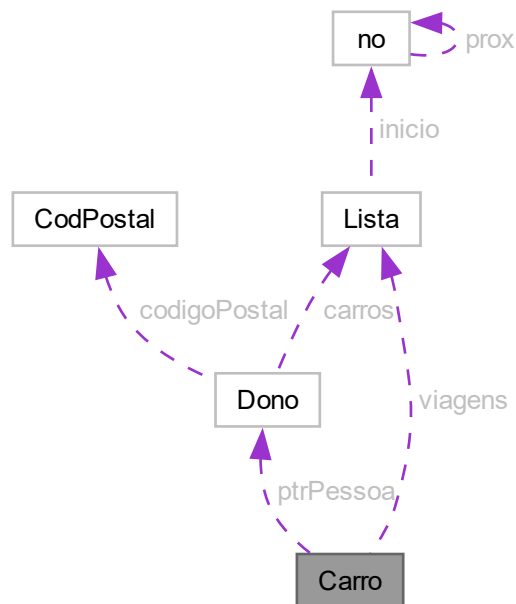
- Dict \* **donosNif**
- Dict \* **donosAlfabeticamente**
- Dict \* **carrosCod**
- Dict \* **carrosMarca**
- Dict \* **carrosMat**
- Lista \* **sensores**
- Distancias \* **distancias**
- Lista \* **viagens**

The documentation for this struct was generated from the following file:

- bdados.h

## 4.2 Carro Struct Reference

Collaboration diagram for Carro:



### Public Attributes

- char **matricula** [MAX\_MATRICULA+1]
- char \* **marca**
- char \* **modelo**
- short **ano**

- int **codVeiculo**
- [Dono](#) \* **ptrPessoa**
- [Lista](#) \* **viagens**

The documentation for this struct was generated from the following file:

- carro.h

## 4.3 CodPostal Struct Reference

### Public Attributes

- short **zona**
- short **local**

The documentation for this struct was generated from the following file:

- dono.h

## 4.4 Data Struct Reference

### Public Attributes

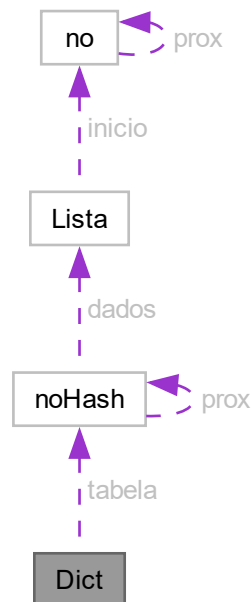
- short **ano**
- short **mes**
- short **dia**
- short **hora**
- short **min**
- float **seg**

The documentation for this struct was generated from the following file:

- uteis.h

## 4.5 Dict Struct Reference

Collaboration diagram for Dict:



### Public Attributes

- [NoHashing](#) \* **tabela** [TAMANHO\_TABELA\_HASH]
- int **nelDict**

The documentation for this struct was generated from the following file:

- structsGenericas.h

## 4.6 Distancias Struct Reference

### Public Attributes

- float \* **matriz**
- int **nColunas**

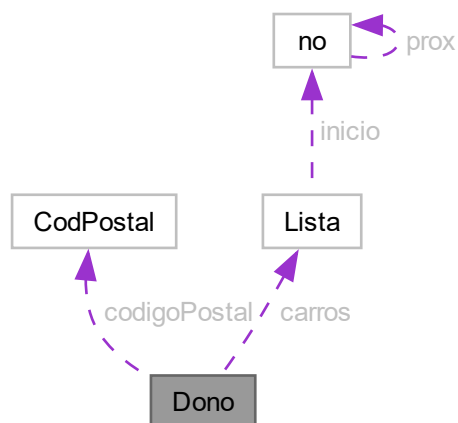
The documentation for this struct was generated from the following file:

- distancias.h



## 4.7 Dono Struct Reference

Collaboration diagram for Dono:



### Public Attributes

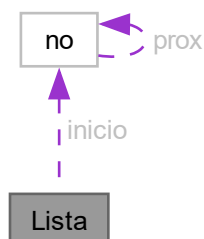
- int **nif**
- char \* **nome**
- [CodPostal](#) **codigoPostal**
- [Lista](#) \* **carros**

The documentation for this struct was generated from the following file:

- dono.h

## 4.8 Lista Struct Reference

Collaboration diagram for Lista:



**Public Attributes**

- `No * inicio`
- `int nel`

The documentation for this struct was generated from the following file:

- `structsGenericas.h`

## 4.9 no Struct Reference

Collaboration diagram for no:

**Public Attributes**

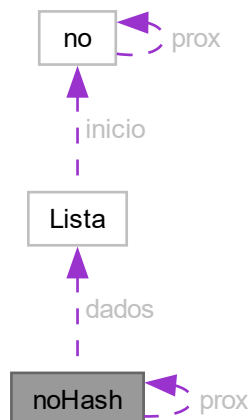
- `void * info`
- `struct no * prox`

The documentation for this struct was generated from the following file:

- `structsGenericas.h`

## 4.10 noHash Struct Reference

Collaboration diagram for noHash:



**Public Attributes**

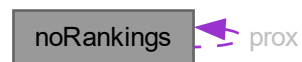
- void \* **chave**
- [Lista](#) \* **dados**
- struct [noHash](#) \* **prox**

The documentation for this struct was generated from the following file:

- structsGenericas.h

## 4.11 noRankings Struct Reference

Collaboration diagram for noRankings:

**Public Attributes**

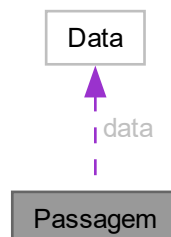
- void \* **mainInfo**
- void \* **complInfo**
- struct [noRankings](#) \* **prox**

The documentation for this struct was generated from the following file:

- structsGenericas.h

## 4.12 Passagem Struct Reference

Collaboration diagram for Passagem:



### Public Attributes

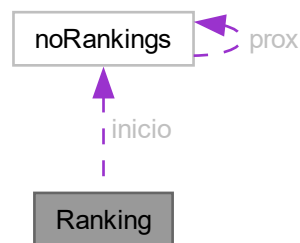
- int **idSensor**
- [Data](#) **data**
- char **tipoRegisto**

The documentation for this struct was generated from the following file:

- passagens.h

## 4.13 Ranking Struct Reference

Collaboration diagram for Ranking:



### Public Attributes

- [NoRankings](#) \* **inicio**
- int **nel**

The documentation for this struct was generated from the following file:

- structsGenericas.h

## 4.14 Sensor Struct Reference

### Public Attributes

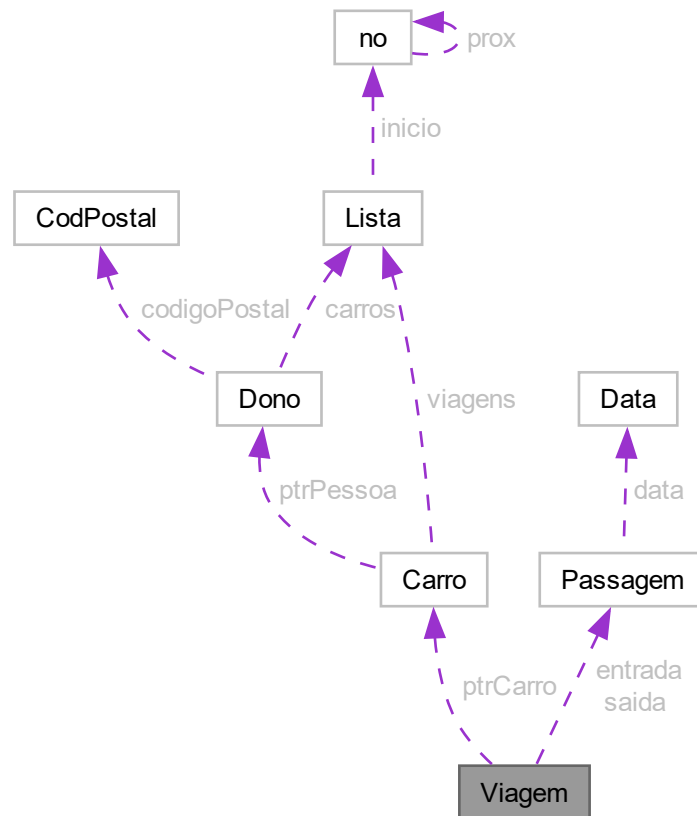
- int **codSensor**
- char \* **designacao**
- char \* **latitude**
- char \* **longitude**

The documentation for this struct was generated from the following file:

- sensores.h

## 4.15 Viagem Struct Reference

Collaboration diagram for Viagem:



### Public Attributes

- **Carro** \* **ptrCarro**
- **Passagem** \* **entrada**
- **Passagem** \* **saida**
- float **kms**
- float **tempo**
- float **velocidadeMedia**

The documentation for this struct was generated from the following file:

- **passagens.h**



## Chapter 5

# File Documentation

### 5.1 bdados.h

```
00001 #ifndef BD_HEADERS
00002 #define BD_HEADERS
00003
00004 #include <stdio.h>
00005 #include <stdlib.h>
00006
00007 #include "structsGenericas.h"
00008 #include "distancias.h"
00009
00010 typedef struct Bdados{
00011     Dict *donosNif;
00012     Dict *donosAlfabeticamente;
00013     Dict *carrosCod;
00014     Dict *carrosMarca;
00015     Dict *carrosMat;
00016     Lista *sensores;
00017     Distancias *distancias;
00018     Lista *viagens;
00019 } Bdados;
00020
00021
00022 int inicializarBD(Bdados *bd);
00023 void freeTudo(Bdados *bd);
00024
00025 // Exportação fica aqui
00026 void exportarTudoXML(Bdados *bd, const char *filename);
00027 void exportarTudoCSV(Bdados *bd, const char *donosFilename, const char *carrosFilename, const char
    *sensoresFilename, const char *distanciasFilename, const char *viagensFilename);
00028 void exportarTudoHTML(Bdados *bd, const char *donosFilename, const char *carrosFilename, const char
    *sensoresFilename, const char *distanciasFilename, const char *viagensFilename);
00029 size_t memUsageTudo(Bdados *bd);
00030
00031
00032 #endif
```

### 5.2 carro.h

```
00001 #ifndef CARRO_HEADERS
00002 #define CARRO_HEADERS
00003
00004 #include <stdio.h>
00005 #include <stdlib.h>
00006 #include <string.h>
00007
00008 #include "dono.h"
00009 #include "constantes.h"
00010 #include "structsGenericas.h"
00011
00012 struct Bdados;
00013
00014 typedef struct {
00015     char matricula[MAX_MATRICULA + 1];
00016     char *marca;
```

```

00017     char *modelo;
00018     short ano;
00019     int codVeiculo; //PRIMARY KEY
00020     Dono *ptrPessoa;
00021     Lista *viagens;
00022 } Carro;
00023
00024
00025 int compararCarros(void *carro1, void *carro2);
00026 int inserirCarroLido(struct Bdados *bd, char *matricula, char *marca, char *modelo, short ano, int
    nif, int codVeiculo);
00027 int compCodCarro(void *carro, void *codigo);
00028 void freeCarro(void *carro);
00029 void printCarro(void *carro, FILE *file);
00030 void guardarCarroBin(void *carro, FILE *file);
00031 void *readCarroBin(FILE *file);
00032 void guardarChaveCarroMarca(void *chaveMarca, FILE *file);
00033 void *criarChaveCarroMarca(void *carro);
00034 int hashChaveCarroMarca(void *chave);
00035 void freeChaveCarroMarca(void *chave);
00036 int compChaveCarroMarca(void *chave, void *chave2);
00037 int compCarroMarca (void *carro1, void *carro2);
00038 int compCarroMatricula(void *carro1, void *carro2);
00039 int compCarroModelo(void *carro1, void *carro2);
00040 void guardarChaveCarroCod(void *chave, FILE *file);
00041 void *readChaveCarroCod(FILE *file);
00042 void *criarChaveCarroCod(void *carro);
00043 int hashChaveCarroCod(void *carro);
00044 void freeChaveCarroCod(void *chave);
00045 int compChaveCarroCod(void *chave, void *chave2);
00046 void *criarChaveCarroMatricula(void *carro);
00047 int hashChaveCarroMatricula(void *chave);
00048 void freeChaveCarroMatricula(void *chave);
00049 int compChaveCarroMatricula(void *chave, void *chave2);
00050 int compMatCarro(void *carro, void *matricula);
00051 int compChaveCarroRankingInt(void *chave, void *chave2);
00052 int compChaveCarroRankingFloat(void *chave, void *chave2);
00053 void freeChaveCarroRankingInt(void *chave);
00054 void freeChaveCarroRankingFloat(void *chave);
00055 void printCarroXML(void *carro, int indentacao, FILE *file);
00056 void printHeaderCarrosCSV(FILE *file);
00057 void printCarroCSV(void *carro, FILE *file);
00058 void printHeaderCarrosTXT(FILE *file);
00059 void printCarroTXT(void *carro, FILE *file);
00060 void printHeaderCarrosHTML(FILE *file);
00061 void printCarroHTML(void *carro, FILE *file);
00062 void printCarroRanking(NoRankings *no, void (*printCompObj)(void *compInfo, FILE *file), FILE *file);
00063 void printCarroRankingTXT(NoRankings *no, void (*printCompObj)(void *compInfo, FILE *file), FILE
    *file);
00064 void printCarroRankingCSV(NoRankings *no, void (*printCompObj)(void *compInfo, FILE *file), FILE
    *file);
00065 void printMarcaRanking(NoRankings *no, void (*printCompObj)(void *compInfo, FILE *file), FILE *file);
00066 void printMarcaRankingTXT(NoRankings *no, void (*printCompObj)(void *compInfo, FILE *file), FILE
    *file);
00067 void printMarcaRankingCSV(NoRankings *no, void (*printCompObj)(void *compInfo, FILE *file), FILE
    *file);
00068 void printHeaderCarroMaisInfracoes(FILE *file);
00069 void printMaisInfracoes(void *compInfo, FILE *file);
00070 void printHeaderCarroMaisInfracoesTXT(FILE *file);
00071 void printHeaderCarroMaisInfracoesCSV(FILE *file);
00072 void printHeaderCarroMaisKMS(FILE *file);
00073 void printMaisKMS(void *compInfo, FILE *file);
00074 void printMaisKMS_CSV(void *compInfo, FILE *file);
00075 void printHeaderCarroMaisKMS_TXT(FILE *file);
00076 void printHeaderCarroMaisKMS_CSV(FILE *file);
00077 void printHeaderMarcaMaisKMS(FILE *file);
00078 char *obterMarcaMaisComum(Dict *carrosMarca);
00079 char *obterMarcaMaisVelocidadeMedia(struct Bdados *bd);
00080 size_t memUsageCarro(void *carro);
00081 size_t memUsageChaveCarroCod(void *chave);
00082 size_t memUsageChaveCarroMarca(void *chave);
00083 size_t memUsageChaveCarroMatricula(void *chave);
00084 int obterCodVeiculoNovo(Dict *carrosCod);
00085 void registrarCarro(struct Bdados *bd);
00086 void mudarDonoCarro(struct Bdados *bd);
00087 void listarCarrosTodos(struct Bdados *bd);
00088 void listarCarrosPorMatricula(struct Bdados *bd);
00089 void listarCarrosPorMarca(struct Bdados *bd);
00090 void listarCarrosPorModelo(struct Bdados *bd);
00091 void listarCarrosPorPeriodoTempo(struct Bdados *bd);
00092 void listarInfracoesPorPeriodoTempo(struct Bdados *bd);
00093 void rankingInfracoes(struct Bdados *bd);
00094 void rankingKMSPeriodoTempo(struct Bdados *bd);
00095 void rankingKMSMarca(struct Bdados *bd);
00096 void listarCarrosComInfracoes(struct Bdados *bd);
00097
00098

```



```
00099 #endif
```

## 5.3 configs.h

```
00001 #ifndef CONFIGS_HEADERS
00002 #define CONFIGS_HEADERS
00003
00004 #include <stdio.h>
00005 #include <stdlib.h>
00006 #ifdef _WIN32
00007     #include <windows.h>
00008 #endif
00009 #include <locale.h>
00010
00011 struct Bdados;
00012
00013 extern int autosaveON;
00014 extern int backupsON;
00015 extern int pausaListagem;
00016
00017 extern char *donosFilename;
00018 extern char *carrosFilename;
00019 extern char *sensoresFilename;
00020 extern char *distanciasFilename;
00021 extern char *passagensFilename;
00022
00023 extern char *donosExportacaoFilename;
00024 extern char *carrosExportacaoFilename;
00025 extern char *sensoresExportacaoFilename;
00026 extern char *distanciasExportacaoFilename;
00027 extern char *viagensExportacaoFilename;
00028 extern char *databaseExportacaoXML;
00029
00030 void colocarTerminalUTF8();
00031 int faseInstalacao(const char *flag, const char abrir);
00032 void reset(struct Bdados *bd);
00033 void setPausaListagem();
00034 void setFileNames();
00035 void setExportacaoFileNames();
00036 void freeFileNames();
00037 void freeExportacaoFileNames();
00038 size_t memUsageVarGlobais();
00039 void autosave(struct Bdados *bd);
00040
00041
00042 #endif
```

## 5.4 constantes.h

```
00001 #ifndef CONSTANTES_HEADERS
00002 #define CONSTANTES_HEADERS
00003
00004 #define SEPARADOR '\\t'
00005 #define SEPARADOR_STR "\\t" //Para usar no strtok
00006 #define MAX_MATRICULA 8
00007 #define TAMANHO_INICIAL_BUFFER 128
00008 #define TAMANHO_BUFFER_LISTAGEM 150
00009 #define TAMANHO_FORMATO_LISTAGEM 12
00010 #define PAUSA_LISTAGEM 1000
00011 #define MAX_VELOCIDADE_AE 120
00012 #define MIN_VELOCIDADE_AE 50
00013
00014 //Nomes default para os ficheiros
00015 #define LOGS_TXT "logs.txt"
00016 #define DONOS_TXT "Dados/donos.txt"
00017 #define CARROS_TXT "Dados/carros.txt"
00018 #define SENSORES_TXT "Dados/sensores.txt"
00019 #define DISTANCIAS_TXT "Dados/distancias.txt"
00020 #define PASSAGEM_TXT "Dados/passagem.txt"
00021 #define DONOS "donos"
00022 #define DONOS_CSV "donos.csv"
00023 #define DONOS_HTML "donos.html"
00024 #define CARROS "carros"
00025 #define SENSORES "sensores"
00026 #define DISTANCIAS "distancias"
00027 #define VIAGENS "viagens"
00028 #define DATABASE_XML "database.xml"
00029 #define CONFIG_TXT "config.txt"
```

```

00030 #define AUTOSAVE_BIN "autosave.bin"
00031
00032 //Número de parâmetros por cada ficheiro
00033 #define PARAM_DONOS 3
00034 #define PARAM_CARROS 6
00035 #define PARAM_SENSORES 4
00036 #define PARAM_DISTANCIAS 3
00037 #define PARAM_PASSAGEM 4
00038
00039 //Extensões
00040 #define DOT_CSV ".csv"
00041 #define DOT_HTML ".html"
00042 #define DOT_BIN ".bin"
00043
00044
00045
00046 #endif

```

## 5.5 dados.h

```

00001 #ifndef DADOS_HEADERS
00002 #define DADOS_HEADERS
00003
00004 #include <stdio.h>
00005 #include <stdlib.h>
00006 #include <string.h>
00007
00008 #include "bdados.h"
00009
00010 int carregarDadosTxt(Bdados *bd, char *fDonos, char *fCarros, char *fSensores, char *fDistancias, char
*fPassagem, char *fLogs);
00011 int carregarDonosTxt(Bdados *bd, char *donosFilename, FILE *logs);
00012 int carregarCarrosTxt(Bdados *bd, char *carrosFilename, FILE *logs);
00013 int carregarSensoresTxt(Bdados *bd, char *sensoresFilename, FILE *logs);
00014 int carregarDistanciasTxt(Bdados *bd, char *distanciasFilename, FILE *logs);
00015 int carregarPassagensTxt(Bdados *bd, char *passagensFilename, FILE *logs);
00016
00017 void removerEspacos(char * str);
00018 void separarParametros(char *linha, char **parametros, int *numParametros, const int paramEsperados);
00019 void linhaInvalida(const char *linha, int nLinha, FILE *logs);
00020 int contarLinhas(const char *filename);
00021
00022 int guardarDadosBin(Bdados *bd, const char *nome);
00023 void guardarDadosBinFicheiro(Bdados *bd);
00024 int carregarDadosBin(Bdados *bd, const char *nome);
00025 int carregarDadosBinFicheiro(Bdados **bd);
00026 unsigned long checksum(Bdados *bd);
00027
00028
00029 #endif

```

## 5.6 distancias.h

```

00001 #ifndef DISTANCIAS_HEADERS
00002 #define DISTANCIAS_HEADERS
00003
00004 #include <stdio.h>
00005 #include <stdlib.h>
00006
00007 #include "passagens.h"
00008
00009 struct Bdados;
00010
00011 typedef struct {
00012     float *matriz;
00013     int nColunas;
00014 } Distancias;
00015
00016 int inserirDistanciaLido(struct Bdados *bd, int codSensor1, int codSensor2, float distancia);
00017 void inicializarMatrizDistancias(struct Bdados *bd);
00018 int realocarMatrizDistancias(struct Bdados *bd, int nColunas);
00019 void freeMatrizDistancias(Distancias *distancia);
00020 void guardarDistanciasBin(Distancias *distancia, FILE *file);
00021 Distancias *readDistanciasBin(FILE *file);
00022 void exportarDistanciasXML(Distancias *d, int indentacao, FILE *file);
00023 void exportarDistanciasCSV(Distancias *d, FILE *file);
00024 void exportarDistanciasHTML(Distancias *d, char *pagename, FILE *file);
00025 size_t memUsageDistancias(Distancias *d);
00026
00027 #endif

```

## 5.7 dono.h

```

00001 #ifndef DONO_HEADERS
00002 #define DONO_HEADERS
00003
00004 #include <stdio.h>
00005 #include <stdlib.h>
00006 #include <string.h>
00007 #include <ctype.h>
00008
00009 #include "structsGenericas.h"
00010
00011 struct Bdados;
00012
00013 typedef struct {
00014     short zona; //Primeiros 4 dígitos
00015     short local; //Últimos 3 dígitos
00016 } CodPostal;
00017
00018 typedef struct {
00019     int nif; //PRIMARY KEY
00020     char *nome;
00021     CodPostal codigoPostal;
00022     Lista *carros;
00023 } Dono, Pessoa, *ptDono, *ptPessoa;
00024
00025 int inserirDonoLido(struct Bdados *bd, char *nome, int nif, CodPostal codigoPostal);
00026 int compDonosNif(void *dono1, void *dono2);
00027 int compDonosNome(void *dono1, void *dono2);
00028 int compCodDono(void *dono, void *codigo);
00029 int compararCodPostal(const CodPostal cod1, const CodPostal cod2);
00030 void freeDono(void *dono);
00031 void printDono(void *dono, FILE *file);
00032 void guardarDonoBin(void *obj, FILE *file);
00033 void *readDonoBin(FILE *file);
00034 void *criarChaveDonoNif(void *dono);
00035 void freeChaveDonoNif(void *chave);
00036 int compChaveDonoNif(void *chave, void *chave2);
00037 int hashChaveDonoNif(void *chave);
00038 void *criarChaveDonoAlfabeticamente(void *dono);
00039 void freeChaveDonoAlfabeticamente(void *chave);
00040 int compChaveDonoAlfabeticamente(void *chave, void *chave2);
00041 void guardarChaveDonoAlfabeticamente(void *chave, FILE *file);
00042 int hashChaveDonoAlfabeticamente(void *chave);
00043 void printDonoXML(void *dono, int indentacao, FILE *file);
00044 void printHeaderDonosCSV(FILE *file);
00045 void printDonoCSV(void *dono, FILE *file);
00046 void printDonoVelocidadesCSV(Dono *dono, float velocidadeMedia, FILE *file);
00047 void printHeaderDonosTXT(FILE *file);
00048 void printDonoTXT(void *dono, FILE *file);
00049 void printHeaderDonosHTML(FILE *file);
00050 void printDonoHTML(void *dono, FILE *file);
00051 void printDonoVelocidadesTXT(Dono *dono, float velocidadeMedia, FILE *file);
00052 void printDonoVelocidadesTXT(Dono *dono, float velocidadeMedia, FILE *file);
00053 size_t memUsageDono(void *dono);
00054 size_t memUsageChaveDonoCod(void *chave);
00055 size_t memUsageChaveDonoAlfabeticamente(void *chave);
00056 void registarDono(struct Bdados *bd);
00057 void listarDonosNIF(struct Bdados *bd);
00058 void listarDonosAlfabeticamente(struct Bdados *bd);
00059 Dono *obterConductorMaisVelocidadeMedia(struct Bdados *bd);
00060 void listarDonosVelocidadesMedias(struct Bdados *bd);
00061 void velocidadeMediaPorCodPostal(struct Bdados *bd);
00062
00063
00064
00065 #endif

```

## 5.8 menus.h

```

00001 #ifndef MENUS_HEADERS
00002 #define MENUS_HEADERS
00003
00004 #include <stdio.h>
00005
00006 #include "bdados.h"
00007
00008 char mostrarMenu(void (*escrever_menu)(), char min_opcao, char max_opcao);
00009 int validacaoMenus(const char opcao, const char limInf, const char limSup);
00010
00011 void menuPrincipal();
00012 void menuDonos();

```

```

00013 void menuVeiculos();
00014 void menuListagemVeiculos();
00015 void menuPassagens();
00016 void menuEstatisticas();
00017 void menuVelocidadesMedias();
00018 void menuInfracoes();
00019 void menuAnaliseDados();
00020 void menuExportacao();
00021 void menuOpcoes();
00022 void menuFormatosListagem();
00023 void menuNomeFicheirosDados();
00024 void menuNomeFicheirosExportacao();
00025 void menuGuiaUtilizacao();
00026
00027 void the_architect(Bdados *bd);
00028 void processarMenuDonos(Bdados *bd);
00029 void processarMenuVeiculos(Bdados *bd);
00030 void processarMenuListagemVeiculos(Bdados *bd);
00031 void processarMenuPassagens(Bdados *bd);
00032 void processarMenuEstatisticas(Bdados *bd);
00033 void processarMenuVelocidades(Bdados *bd);
00034 void processarMenuInfracoes(Bdados *bd);
00035 void processarMenuAnaliseDados(Bdados *bd);
00036 void processarMenuExportacao(Bdados *bd);
00037 void processarMenuOpcoes(Bdados *bd);
00038
00039
00040
00041
00042 #endif

```

## 5.9 passagens.h

```

00001 #ifndef PASSAGENS_HEADERS
00002 #define PASSAGENS_HEADERS
00003
00004 #include <stdio.h>
00005 #include <stdlib.h>
00006
00007 #include "sensores.h"
00008 #include "carro.h"
00009 #include "uteis.h"
00010
00011 struct Bdados;
00012
00013 typedef struct {
00014     int idSensor; //PRIMARY KEY / FOREIGN KEY(idSensor) - REFERENCES Sensor(codSensor)
00015     Data data;
00016     char tipoRegistro;
00017 } Passagem, Pass, *ptPassagem, *ptPass;
00018
00019 typedef struct {
00020     Carro *ptrCarro;
00021     Passagem *entrada;
00022     Passagem *saida;
00023     float kms;
00024     float tempo; //Em minutos
00025     float velocidadeMedia; // km/h
00026 } Viagem;
00027
00028
00029 Passagem *obterPassagem(int idSensor, Data date, char tipoRegistro);
00030 int inserirViagemLido(struct Bdados *bd, Passagem *entrada, Passagem *saida, int codVeiculo);
00031 int compararPassagens(void *passagem1, void *passagem2);
00032 int compCodPassagem(void *passagem, void *codigo);
00033 void freePassagem(void *passagem);
00034 void freeViagem(void *viagem);
00035 void guardarViagemBin(void *viagem, FILE *file);
00036 void *readViagemBin(FILE *file);
00037 void guardarPassagemBin(void *passagem, FILE *file);
00038 void *readPassagemBin(FILE *file);
00039 void getStatsViagem(struct Bdados *bd, Viagem *v);
00040 void printViagem(void *viagem, FILE *file);
00041 void printViagemXML(void *viagem, int indentacao, FILE *file);
00042 void printHeaderViagensCSV(FILE *file);
00043 void printViagemCSV(void *viagem, FILE *file);
00044 void printHeaderViagensTXT(FILE *file);
00045 void printViagemTXT(void *viagem, FILE *file);
00046 void printHeaderViagensHTML(FILE *file);
00047 void printViagemHTML(void *viagem, FILE *file);
00048 size_t memUsagePassagem(void *passagem);
00049 size_t memUsageViagem(void *viagem);
00050 void registrarViagem(struct Bdados *bd);

```

```

00051 Passagem *pedirPassagem(struct Bdados *bd);
00052 void listarViagensTodas(struct Bdados *bd);
00053
00054
00055
00056 #endif

```

## 5.10 sensores.h

```

00001 #ifndef SENSORES_HEADERS
00002 #define SENSORES_HEADERS
00003
00004 #include <stdio.h>
00005 #include <stdlib.h>
00006 #include <string.h>
00007
00008 struct Bdados;
00009
00010 typedef struct {
00011     int codSensor; //PRIMARY KEY
00012     char *designacao;
00013     char *latitude;
00014     char *longitude;
00015 } Sensor, *ptSensor;
00016
00017 int inserirSensorLido(struct Bdados *bd, int codSensor, char *designacao, char *latitude, char
    *longitude);
00018 int compararSensores(void *sensor1, void *sensor2);
00019 int compIdSensor(void *sensor, void *idSensor);
00020 void freeSensor(void *sensor);
00021 void printSensor(void *sensor);
00022 void guardarSensorBin(void *sensor, FILE *file);
00023 void *readSensorBin(FILE *file);
00024 void printSensorXML(void *sensor, int indentacao, FILE *file);
00025 void printHeaderSensoresCSV(FILE *file);
00026 void printSensorCSV(void *sensor, FILE *file);
00027 void printHeaderSensoresHTML(FILE *file);
00028 void printSensorHTML(void *sensor, FILE *file);
00029 size_t memUsageSensor(void *sensor);
00030
00031
00032 #endif

```

## 5.11 structsGenericas.h

```

00001 #ifndef STRUCTS_GENERICAS_HEADERS
00002 #define STRUCTS_GENERICAS_HEADERS
00003
00004 #include <stdio.h>
00005 #include <stdlib.h>
00006 #include <string.h>
00007
00008 #define TAMANHO_TABELA_HASH 16007 // load factor 0,62; N°primo para não haver tantas colisões
00009
00010 extern int listagemFlag;
00011 extern int printingDict;
00012
00013 extern void pressEnter();
00014 extern int enter_espaco_esc();
00015
00016 typedef struct no {
00017     void *info;
00018     struct no *prox;
00019 } No;
00020
00021 typedef struct {
00022     No *inicio;
00023     int nel;
00024 } Lista;
00025
00026 typedef struct noHash {
00027     void *chave;
00028     Lista *dados;
00029     struct noHash *prox; // Necessário caso chaves diferentes gerem o mesmo índice
00030 } NoHashing;
00031
00032 typedef struct {
00033     NoHashing *tabela[TAMANHO_TABELA_HASH];

```

```

00034     int nelDict;
00035 } Dict; // Cada Dict ocupa cerca de 125KB em memória (sem elementos ou chaves)
00036
00037 typedef struct noRankings {
00038     void *mainInfo;
00039     void *compInfo;
00040     struct noRankings *prox;
00041 } NoRankings;
00042
00043 typedef struct {
00044     NoRankings *inicio;
00045     int nel;
00046 } Ranking;
00047
00048
00049 // Listas
00050
00051 Lista *criarLista();
00052 int addInicioLista(Lista *li, void *elemento);
00053 void printLista(Lista *li, void (*printObj)(void *obj, FILE *file), FILE *file, int pausa);
00054 void exportarListaXML(Lista *li, char *nomeLista, void (*printObj)(void *obj, int indentacao, FILE
    *file), int indentacao, FILE *file);
00055 void exportarListaCSV(Lista *li, void (*printHeader)(FILE *file), void (*printObj)(void *obj, FILE
    *file), FILE *file);
00056 void exportarListaHTML(Lista *li, char *pagename, void (*printTableHead)(FILE *file), void
    (*printObj)(void *obj, FILE *file), FILE *file);
00057 void freeLista(Lista *li, void (*freeObj)(void *obj));
00058 void mergeSortLista(Lista *li, int (*compObjs)(void *obj1, void *obj2));
00059 void ordenarLista(Lista *li, int (*compObjs)(void *obj1, void *obj2));
00060 void *searchLista(Lista *li, int (*compCod)(void *codObj, void *chave), void *chave);
00061 void guardarListaBin(Lista *li, void (*saveInfo)(void *obj, FILE *fileObj), FILE *file);
00062 Lista *readListaBin(void *(*readInfo)(FILE *fileObj), FILE *file);
00063 size_t listaMemUsage(Lista *li, size_t (*objMemUsage)(void *obj));
00064
00065 // Hashing
00066
00067 Dict *criarDict();
00068 NoHashing *posicaoInsercao(Dict *has, int indice, void *chave, int (*compChave)(void *chave, void
    *chave2));
00069 Lista *obterListaDoDict(Dict *has, void *chave, int (*compChave)(void *chave, void *chave2), int
    (*hashChave)(void *obj));
00070 int appendToDict(Dict *has, void *obj, int (*compChave)(void *chave, void *obj), void
    *(*criarChave)(void *obj), int (*hashChave)(void *obj), void (*freeObj)(void *obj), void
    (*freeChave)(void *chave));
00071 void printDict(Dict *has, void (*printObj)(void *obj, FILE *file), FILE *file, int pausa);
00072 void exportarDictXML(Dict *has, char *nomeDict, void (*printObj)(void *obj, int indentacao, FILE
    *file), int indentacao, FILE *file);
00073 void exportarDictCSV(Dict *has, void (*printHeader)(FILE *file), void (*printObj)(void *obj, FILE
    *file), FILE *file);
00074 void exportarDictHTML(Dict *has, char *pagename, void (*printTableHead)(FILE *file), void
    (*printObj)(void *obj, FILE *file), FILE *file);
00075 void freeDict(Dict *has, void (*freeChave)(void *chave), void (*freeObj)(void *obj));
00076 void *searchDict(Dict *has, void *chave, int (*compChave)(void *chave, void *obj), int (*compCod)(void
    *codObj, void *chave), int (*hashChave)(void *chave));
00077 void guardarDadosDictBin(Dict *has, void (*saveInfo)(void *obj, FILE *fileObj), FILE *file);
00078 Dict *readToDictBin(void *(*criarChave)(void *obj), int (*hashChave)(void *obj), void (*freeObj)(void
    *obj), void (*freeChave)(void *chave), void *(*readInfo)(FILE *fileObj), FILE *file);
00079 size_t dictMemUsage(Dict *has, size_t (*objMemUsage)(void *obj), size_t (*chaveMemUsage)(void
    *chave));
00080 Lista *dictToLista(Dict *has);
00081
00082 // Listas Rankings
00083
00084 Ranking *criarRanking();
00085 int addToRanking(Ranking *r, void *mainInfo, void *compInfo);
00086 void printRanking(Ranking *r, void printCarroRanking(NoRankings *no, void (*printCompObj)(void
    *compInfo, FILE *file), FILE *file), void (*printHeaderCompObj)(FILE *file), void (*printCompObj)(void
    *compInfo, FILE *file), FILE *file, int pausa);
00087 void freeRanking(Ranking *r, void (*freeMainObj)(void *obj), void (*freeCompObj)(void *obj));
00088 void mergeSortRanking(Ranking *r, int (*compObjs)(void *obj1, void *obj2));
00089
00090
00091 #endif

```

## 5.12 uteis.h

```

00001 #ifndef UTEIS_HEADERS
00002 #define UTEIS_HEADERS
00003
00004 #include <stdio.h>
00005 #include <stdlib.h>
00006 #if defined(_WIN32) || defined(_WIN64)
00007     #include <windows.h>

```

```

00008     #include <conio.h>
00009     #include <psapi.h>
00010 #else
00011     #include <sys/resource.h>
00012 #endif
00013 #include <locale.h>
00014 #include <time.h>
00015 #include <ctype.h>
00016 #include <limits.h>
00017 #include <string.h>
00018
00019 #include "constantes.h"
00020
00021 extern char mostrarMenu(void (*escrever_menu)(), char min_opcao, char max_opcao);
00022 extern void menuFormatosListagem();
00023
00024 typedef struct {
00025     short ano, mes, dia, hora, min;
00026     float seg;
00027 } Data;
00028
00029 extern Data DATA_ATUAL;
00030
00031
00032 void limpar_buffer();
00033 void limpar_terminal();
00034 int getKeyStroked();
00035 void pressEnter();
00036 int enter_espaco_esc();
00037 int randomInt(int min, int max);
00038 void data_atual();
00039 int sim_nao(char *mensagem);
00040 char *normString(const char *str);
00041 char *strlwrSafe(const char *str);
00042 int strcmpSafe(const char *s1, const char *s2);
00043 int stringToInt(const char *str, int *resultado);
00044 int stringToShort(const char *str, short *resultado);
00045 int stringToFloat(const char *str, float *resultado);
00046 void converterPontoVirgulaDecimal(char *str);
00047 void replaceStrObj(char *str, const char out, const char in);
00048 char *lerLinhaTxt(FILE *ficheiro, int *n_linhas);
00049 void pedirInt(int *num, char *mensagem, int (*validarInput)(int input));
00050 void pedirShort(short *num, char *mensagem, int (*validarInput)(short input));
00051 void pedirData(Data *data, char *mensagem);
00052 void pedirPeriodoTempo(Data *inicio, Data *fim, char *mensagemInicial, char *mensagemFinal);
00053 int converterCodPostal(const char *codPostal, short *zona, short *local);
00054 int compararDatas(Data data1, Data data2);
00055 char *converterParaData(const char *strData, Data *data);
00056 float calcularIntervaloTempo(Data *data1, Data *data2);
00057 int hashString(const char *str);
00058 int deleteFile(const char *nome, const char modo);
00059 void indent(int indentacao, FILE *file);
00060 char *floatToStringPontoDecimal(float valor, int casasDecimais);
00061 int validarNomeFicheiro(const char *filename);
00062 FILE *abrirFicheiroComValidacao(const char *nome);
00063 FILE *pedirListagemFicheiro(char *formatoSelecioneado);
00064 char *appendFileExtension(const char *filename, const char *extension);
00065
00066
00067
00068
00069 #endif

```

## 5.13 validacoes.h

```

00001 #ifndef VALIDACOES_HEADERS
00002 #define VALIDACOES_HEADERS
00003
00004 #include <stdio.h>
00005 #include <stdlib.h>
00006 #include <string.h>
00007 #include <wctype.h>
00008
00009 #include "uteis.h"
00010
00011 int validarNif(const int nif);
00012 char *validarNome(char *nome);
00013 int validarCodPostal(const short zona, const short local);
00014 int validarAnoCarro(const short ano);
00015 int validarMatricula(const char *matricula);
00016 char *validarMarca(char *marca);
00017 char *validarModelo(char *modelo);
00018 int validarCodVeiculo(const int codigo);

```

```
00019 int validarCodSensor(const int codSensor);
00020 int validarDistancia(const float distancia);
00021 int validarData(const Data date, const char modo);
00022 int validarTipoRegistro(const char tipoRegistro);
00023
00024 #endif
```