Escola Superior de Tecnologia e Gestão

Engenharia Informática

Ano letivo de 2017/2018

Gestão da distribuição de encomendas

Gonçalo Vicente

Tiago Silva

19 de dezembro de 2017

Introdução

O projeto que nos foi apresentado para desenvolver no âmbito da disciplina de Programação 1 consiste em desenvolver um programa em linguagem C para auxiliar uma empresa de transportes na gestão da distribuição de encomendas.

Para tal, o programa deverá registar a informação referente a todos os veículos que a empresa possui (a empresa não pode possuir mais de 10 veiculos) e as encomendas a transportar, sendo que estas não podem ultrapassar o limite de 50. Assim que uma encomenda chega ao centro de distribuição, o primeiro passo a fazer é efetuar o seu registo, sendo que depois ficará disponível para ser distribuída para o seu destino.

Ainda antes de se iniciar a entrega das encomendas é necessário ter em conta o número de veículos necessários para levar as encomendas com o mesmo destino. Quando 80% da carga total dos veículos estiver completa o veículo pode seguir viagem.

Índice

Lista de Figuras

Realização do Projeto

Estruturas de dados

Para a realização do projeto, foi necessário usarmos diferentes estruturas de dados. Abaixo pode-se ver quais as estruturas utilizadas e a respetiva explicação.

Começamos por criar a estrutura de dados referente ao tipo da encomenda que a distribuidora recebe.

```
typedef struct
{
    int numRegisto;
    tipoData dataRegisto;
    int peso;
    char destino [TEXTO_BREVE];
    char estado [TEXTO_BREVE];
    tipoData dataEntrega;
    char observacoes [TEXTO_LONGO];
} tipoEncomenda;
```

Figura 1 - Estrutura de dados tipoEncomenda

Nesta estrutura foi necessário introduzir um numRegisto para a encomenda e esse número será do tipo inteiro, depois para adicionar a data foi necessário introduzir uma nova estrutura de dados, esta referente à data. A estrutura tipoData deverá ser colocada antes desta estrutura e a data de registo corresponde à data quando a encomenda irá ser registada. O peso irá ser uma variável do tipo inteiro, uma vez que vamos considerar valores aproximados das encomendas. O destino da encomenda será uma variável do tipo char e terá o limite de 20 carateres. Já o estado corresponde ao estado em que se encontra a encomenda, podendo este ser registada, carregada, entregue ou caso exista algum problema de devolvida. Volta a utilizar a estrutura de dados do tipoData para introduzir a data em que a encomenda foi entregue ao cliente. Por fim, as observações são uma variável do tipo char e tem o limite máximo de 100 carateres.

Na imagem abaixo, podemos ver a estrutura de dados utilizada para o a data em que o temos o ano, o mês e o dia, todas variáveis do tipo inteiro.

```
typedef struct
{
    int dia;
    int mes;
    int ano;
} tipoData;
```

Figura 2 - Estrutura de dados tipoData

Na imagem abaixo, podemos ver a estrutura de dados utilizada para os veículos.

```
typedef struct
{
    char matricula[MAX_MATRICULA];
    tipoData dataFabrico;
    char estado[TEXTO_BREVE];
    int carga ;
    int numViagensEfetuadas;
    int quantidadeEncomendasTransportadas;
    int numRegisto[MAX_ENCOMENDAS];
} tipoVeiculo ;
```

Figura 3 - Estrutura de dados do tipoVeiculo

Nesta estrutura temos a matricula como uma variável do tipo char, uma vez que recebe tanto letras como números e está limitada a 6 carateres, pois as matrículas portuguesas tem apenas 6 carateres. Para a data de fabrico do veículo irá ser necessário voltar a recorrer à estrutura de dados tipoData, já explicada anteriormente. O estado irá ser uma variável do tipo char e limitada a 20 carateres. O estado pode ser disponível, em carga, a transportar, de regresso ou avariado. O número de viagens efetuadas pelo veículo corresponde a uma variável do tipo inteiro, bem como a quantidade de encomendas transportadas e o número de registo da encomenda, que no máximo pode ser 100.

Para a realização do trabalho foi necessário fazer algumas restrições como as que se podem ver na imagem a seguir.

```
#ifndef ESTRUTURAS_H_INCLUDED
#define ESTRUTURAS_H_INCLUDED
#define TEXTO_BREVE 20
#define TEXTO_LONGO 100
#define MAX_MATRICULA 7
#define MAX_CARGA 100
#define MIN_CARGA 80
#define MAX_STRING 50
#define ESTADO_DISPONIVEL "Disponivel"
#define ESTADO_EM_CARGA "Em carga"
#define ESTADO_TRANSPORTAR "A transportar"
#define ESTADO_DE_REGRESSO "De regresso"
#define ESTADO_AVARIADO "Avariado"
#define MAX_VEICULOS 9
#define NAO_EXISTE -1
#define ESTADO_REGISTADA 1
#define ESTADO_CARREGADA 2
#define ESTADO_ENTREGUE 3
#define ESTADO_DEVOLVIDA 4
#define MAX_ENCOMENDAS 99
```

Figura 4 - Restrições no programa

Como se pode verificar através da imagem, estão aqui limitadas algumas das variáveis que usamos ao longo do programa. Por exemplo, o TEXTO_BREVE está limitado a 20 carateres e o TEXTO_LONGO a 100 carateres. A matrícula encontra-se limitada a 7 carateres, ou seja só irão aparecer os 6 primeiros. O MIN_CARGA e o MAX_CARGA corresponde à quantidade máxima e mínima que pode o veículo pode ter para seguir viagem. No que diz respeito ao número máximo de encomendas o vetor está limitado a 99, uma vez que começa a contagem no 0, logo existem no máximo 100 encomendas. O mesmo acontece como o número de veículos. Além disto, quando a encomenda está registada é equivalente ao número, quando se encontra carregada equivale ao número 2, quando é entregue equivale ao número 3 e, caso seja devolvida fica com o número 4.

O que falta fazer?

Para a completa realização deste projeto falta apenas listar o início da viagem e o regresso ao centro de distribuição de um determinado veículo.