



**ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO**

Projeto de Laboratório de Programação

Licenciatura em Engenharia Informática

2018/2019

Grupo 71

Francisco Matos – 8180463

Jéssica Coelho – 8170180

Paula Silva – 8160583

1. Introdução

A proposta feita para este trabalho tem a ver com o desenvolvimento de um sistema de informação com o objetivo de gerir operações de portagens da empresa Via Verde. Na realização deste trabalho temos de ter em contas as funcionalidades que nos são pedidas e propor mais funcionalidades de modo a obtermos um sistema mais completo e mais específico.

Sendo a Via Verde uma empresa no setor de infraestruturas de transporte, mais verificada na utilização de portagens eletrónicas. Para evitar a paragem do condutor foram desenvolvidos identificadores eletrónico colocados nos veículos que, no momento de passagem na portagem, identificam o veículo e o contrato associado para futura faturação. Assim, é necessário ter em conta alguns conceitos para a perceção deste sistema de informação, sendo estas as mais relevantes:

- Pórticos - que representam os pontos da autoestrada em que o veículo passa, bem como as distâncias entre os pórticos complementando com os preços tabelados dos lanços entre esses pares de pórticos.
- Cliente - que representam um cliente da Via Verde e que tem de ter um veículo registado e também tem de fazer passagens nesses pórticos. A Via Verde assume como cliente alguém que apresente os dados requeridos que são: nome, NIF, número do CC, NIB, morada e a quantidade de pontos Via Verde.
- Veículo - este tem de estar associado a uma conta cliente específica com registo da sua matrícula, marca e modelo do mesmo.
- Viagem - apresenta a passagem de um determinado veículo, num par de pórticos, com a data e hora de saída exata da mesma mostrando também o custo associado a esse lanço.
- Utilizador - funcionário da Via Verde, quer humano quer computador, que vai utilizar o sistema de informação desenvolvido.

2. Funcionalidades requeridas

Neste trabalho foram-nos apresentadas um conjunto de funcionalidades que têm obrigatoriamente de constar neste trabalho, sendo elas:

```
printf("Bem vindo a Via Verde\n");
printf("Selecione a operacao que pretende efetuar\n");

do
{
    do
    {
        printf("1 - Gestao de utilizadores\n");
        printf("2 - Gestao de viagens\n");
        printf("3 - Gestao de precos\n");
        printf("4 - Gestao de faturas\n");
        printf("5 - Presistencia de dados\n");
        printf("6 - Informacoes extra\n");
        printf("0 - Sair\n");
        printf("Selecione a opcao: ");
        readInt(&opcao, 0, 6, "");
    } while(opcao < 0 || opcao > 6);
```

- Gestão de Utilizadores - que tem de ser composta por: criar, editar, remover, pesquisar e listar os clientes. E em que temos de ter em conta que ao eliminar um cliente deverá de ser eliminada toda a informação relativa à mesma.

```

case 1: //Gestão de utilizadores
do
{
do
{
printf("1 - Criar cliente\n");
printf("2 - Editar cliente\n");
printf("3 - Remover cliente\n");
printf("4 - Pesquisar por cliente\n");
printf("5 - Listar clientes\n");
printf("0 - Voltar ao inicio\n");
printf("Selecione a opcao: ");
readInt(&a, 0, 5, "");
} while(a < 0 || a > 5);

switch (a)
{
case 1: //Adicionar um cliente
add_client(&clientes, &quantos, &tam_max);
break;

case 2: //Editar dados de um cliente já existente
editar_user(&clientes, &quantos, &tam_max);
break;

case 3: //Remover cliente já existente
removeclient(&clientes, &quantos, &tam_max);
break;

case 4: //Pesquisar por cliente a partir do número de cartão de cidadão
pesquisar_cliente(&clientes, &quantos, tam_max);
break;

case 5: //Imprimir a lista dos clientes registados
listar_user(&clientes, &quantos);
break;
}
} while(a != 0);
break;

```

- Gestão de Viagens - tem de ser possível acrescentar uma viagem entre um determinado par de pórtilos (em que pode ou não existir lanço), para um veículo específico, em certa data e hora, também tem de ser possível realizar pesquisas viagens por veículo, data, pórtilos ou por combinações de vários fatores.

```

case 2: //Gestão de viagens
do
{
do
{
printf("1 - Adicionar uma viagem\n");
printf("2 - Pesquisar viagens\n");
printf("0 - Voltar ao inicio\n");
printf("Selecione a opcao: ");
readInt(&b, 0, 2, "");
} while(b < 0 || b > 2);

switch(b)
{
case 1: //Adicionar uma viagem a um determinado cliente
add_viagem(&clientes, &quantos);
break;

case 2: //Pesquisar por uma determinada viagem já existente
pesquisar_viagens(&clientes, &quantos, &quantviagem);
break;
}
} while (b != 0);
break;

```

- Gestão de Preços - tem de possibilitar a edição dos preços tabelados para os diversos lanços, sendo que os lanços existentes e os pórtilos ou o número dos mesmo não pode ser alterado, para além disso, a alteração dos preços não pode ser revelada nas viagens já registadas.

```

case 3: //Gestão de preços
do
{
    do
    {
        printf("1 - Editar preços\n");
        printf("2 - Listar preços\n");
        printf("0 - Voltar ao inicio\n");
        printf("Selecione a opcao: ");
        readInt(&x, 0, 2, "");
    } while (x < 0 || x > 2);

    switch(x)
    {
        case 1: //Edição de preços
            edit_precos();
            break;

        case 2: //listagem de preços
            list_precos();
            break;
    }

} while (x != 0);
break;

```

- Geração de Faturas - a aplicação tem de ser capaz de gerar uma fatura para um determinado utilizar num determinado mês, essa mesma fatura tem de ser gerada no ecrã e/ou escrita para um ficheiro contendo uma listagem de todas as viagens efetuadas nesse mês com identificação do pórtilco de entrada e saída e respetiva data e hora, do cliente e do total devido.

```

case 4: //Gestão de faturas
printf("Gerar fatura: \n");
fatura(&clientes, &viagens, &quantos, &quantviagem);
break;

```

- Persistência de dados: a aplicação tem de possibilitar guardar e carregar dados em ficheiros e de ficheiros e deve perseverar esses dados no programa ao longo das diferentes utilizações.

```

case 5: //Presistência de dados - exportar
do
{
    do
    {
        printf("1 - Gravar Dados dos Clientes\n");
        printf("2 - Gravar Dados das Viagens\n");
        printf("0 - Voltar ao inicio\n");
        printf("Escolha uma opcao: ");
        readInt(&opt, 0, 2, "");
    } while(opt < 0 || opt > 2);

    switch(opt)
    {
        case 1: guardarCliente(&clientes, &quantos);
            break;

        case 2: guardarViagem(&clientes, &viagens, &quantos, &quantviagem);
            break;
    }

} while(opt!=0);

break;

```

3. Funcionalidades propostas

Para além das funcionalidades pedidas, propomos mais algumas que achamos complementar e beneficiar os clientes da Via Verde e são elas:

- Radares - permite adicionar a um ficheiro que contém a descrição de um determinado lanço com a sua entrada e saída e o utilizador informa ao cliente em que quilómetro aparece o radar nesse determinado lanço, consiste também na presença de uma lista de radares que vão sendo adicionados pelo utilizador
- Multas - que consiste na identificação do tipo da multa que pode ser por excesso de velocidade e alto nível de álcool no sangue e apresenta o valor da multa tal como a penalização por pontos na carta de condução.
- Reclamações: não só permite ao cliente fazer uma reclamação e guardá-la para que possa ser mais tarde consultada como também ver uma lista com todas as reclamações feitas por todos os clientes que utilizam o sistema.

```
case 6: //Funções extra
do
{
    do
    {
        printf("1 - Informacao sobre multas\n");
        printf("2 - Informacao sobre radares\n");
        printf("3 - Reclamacoes\n");
        printf("0 - Voltar ao inicio\n");
        printf("Selecione a opcao: ");
        readInt(&y, 0, 3, "");
    }while(y < 0 || y > 3);

    switch(y)
    {
        case 1: //Multas de velocidade e excesso de alcool
            tipomulta();
            break;

        case 2: //Radares
            menuRadar();
            break;

        case 3: //Reclamações
            menuReclama();
            break;
    }

    }while (y != 0);

    break;
```

4. Estrutura analítica do projeto

- Gestão de Utilizadores:
 - Criar utilizador;
 - Editar utilizador;
 - Remover utilizador;
 - Pesquisar e listar clientes.
- Gestão de Viagens:
 - Acrescentar viagens entre um determinado par de pórticos com os dados:
 - Veículo;
 - Data/hora.
 - Pesquisar viagens por:
 - Veículo;
 - Data;
 - Pórticos;
 - ou outras combinações de fatores.
- Gestão de preços:
 - Editar os preços tabelados nos diferentes lanços.

- Gestão de Faturas;
 - Gerar fatura para um determinado utilizador num mês específico;
 - A fatura deve conter:
 - Todas as viagens efetuadas com a identificação do:
 - Mês;
 - Pórtico de entrada e de saída;
 - Data e hora;
 - Identificação do cliente;
 - Valor a dever.
- Persistência dos dados:
 - Guardar e carregar os dados em e de ficheiros.
- Radares:
 - Adicionar radar:
 - Lanço em que está o radar;
 - Quilómetro em que está o radar.
 - Lista de radares.
- Multas:
 - Tipo de multa
 - Valor da multa
 - Penalização na carta de condução
- Reclamações:
 - Criar reclamação;
 - Lista de reclamações.

5. Funcionalidades implementadas

Todas as funcionalidades requeridas foram implementadas, tal como as funcionalidades propostas, apesar de não ser no número pedido. Note-se que o código produzido apresenta erros na parte da gestão de viagens e não está a funcionar corretamente o que faz com que as partes relacionadas com estas também não estejam a funcionar corretamente.

6. Conclusão

Concluindo, pensamos que este trabalho foi realizado com sucesso, e que tudo o que nos foi proposto terá sido bem implementado. Adquirimos, ao longo do trabalho, e complementamos os mais diversos conceitos aprendidos ao longo do semestre.