



Universidade de Coimbra  
Faculdade de Ciências e Tecnologia  
Departamento de Engenharia Informática

UNIDADE CURRICULAR  
**Princípios de Programação Procedimental**  
2015-2016

**Mini-Projeto** – Desenvolvimento de uma aplicação em linguagem C para gerir reservas de uma agência de viagens: TravelingInC.exe

Docente:  
João P. Vilela

Alunos:  
Ricardo Cruz  
Gilberto Rouxinol

## ÍNDICE

1	Introdução .....	3
2	Implementação das Estruturas de Dados Dinâmica .....	3
2.1	Ficheiro cliente.h .....	3
2.2	Ficheiro destino.h.....	4
2.3	Ficheiro save.h .....	4
2.4	Ficheiro datas.h.....	5
3	Funções e Procedimentos .....	5
3.1	Ficheiro cliente.c.....	5
3.2	Ficheiro destino.c.....	6
3.3	Ficheiro save.c .....	7
3.4	Ficheiro cadeias.c.....	8
3.5	Ficheiro gestão.c .....	8
3.6	Função main.c .....	9
4	Conclusão .....	10

# Mini – Projeto

## Agência de Viagens

### 1 Introdução

O trabalho apresenta uma aplicação escrita em C que permite a gestão de uma agência de viagens. Para tal, foram utilizadas as competências adquiridas nas aulas de PPP, nomeadamente, a implementação de listas ligadas, manipulação de ficheiros, manipulação de memória, manipulação de ficheiros ponto h e ponto c, verificação de dados, métodos auxiliares (strings, concatenação de data para comparação de datas), includes, etc.

### 2 Implementação das Estruturas de Dados Dinâmica

Nas secções seguintes apresentam-se as estruturas implementadas nos diversos ficheiros ponto h.

#### 2.1 Ficheiro cliente.h

No ficheiro cliente.h foram definidas quatro estruturas. As primeiras duas são ponteiros para estruturas do tipo nó destino e nó cliente. A primeira é aqui incluída porque a estrutura que se apresenta seguidamente (estrutura destino) também necessita da definição de estrutura cliente. A terceira estrutura (estrutura cliente) apresenta dados

genéricos “encapsulados” de diversos tipos (nome, profissão, morada, número de cidadão (id) e data de nascimento) e duas listas do tipo lista destinos (comprados e espera). A quarta e última estrutura (nó cliente) é definida por quatro ponteiros. O primeiro ponteiro do tipo cliente (info) seguido de dois ponteiros do tipo datação (info pedido d e info espera d) e por último um ponteiro do tipo lista clientes (next). No decurso do trabalho verificou-se que os dois ponteiros (info pedido d e info espera d) são redundantes uma vez que guardam a mesma informação. No entanto optou-se pela sua continuidade, dado o avanço dos trabalhos e o facto de se estar em fase de aprendizagem.

## **2.2 Ficheiro destino.h**

No ficheiro destino.h foram definidas também quatro estruturas. As primeiras duas são ponteiros para estruturas do tipo nó destino e nó cliente. A segunda é aqui incluída porque a estrutura apresentada anteriormente (estrutura cliente) também necessita da definição de estrutura destino. A terceira estrutura (estrutura destino) apresenta dados genéricos “encapsulados” destino (id, nome, limite de clientes, vendidos, em fila e data de partida) e duas listas clientes (compraram e espera). A quarta e última estrutura (nó destino) é definida por quatro ponteiros. Um ponteiro do tipo destino (info), mais dois ponteiros do tipo datação (info pedido c e info espera c) e por último um ponteiro do tipo lista destinos (next). Tal como referido anteriormente os dois ponteiros (info pedido c e info espera c) são redundantes.

## **2.3 Ficheiro save.h**

No ficheiro save.h foram definidas duas estruturas. Uma estrutura do tipo ponteiro para nó save e outra estrutura nó save. A estrutura nó save é definida por: (1) dois inteiros (id cliente e id destino); (2) um ponteiro do tipo datação (dat); e (3) um ponteiro do tipo lista save (next).

## 2.4 Ficheiro datas.h

No ficheiro datas.h foram definidas três estruturas: (1) calendário; (2) horas; e (3) datação. Esta última guarda dois ponteiros um para estrutura calendário e outro para estrutura horas.

# 3 Funções e Procedimentos

Apresenta-se a seguir de uma forma muito breve as funções e os procedimentos relevantes de cada um dos ficheiros ponto c correspondentes aos ficheiros ponto h anteriormente referidos. Complementarmente apresenta-se também informação relativa a outros ficheiros ponto c cujos ficheiros ponto h não foram referidos anteriormente.

## 3.1 Ficheiro cliente.c

### *Registar Cliente*

Função que devolve um inteiro (id cliente) e que permite adicionar um cliente à lista de clientes inicial.

### *Imprime Clientes de um Destino*

Procedimento que permite a impressão dos destinos comprados e em espera de um determinado cliente por ordem decrescente da data de aquisição do destino, i.e., os mais recentes primeiro.

### *Imprime Lista de Clientes*

Procedimento que imprime por ordem alfabética os clientes da agência.

### *Inserir na lista Cliente por ordem alfabética*

Procedimento que insere os dados de um determinado cliente numa lista ligada de um só sentido por ordem alfabética.

### *Inserir na lista Cliente por ordem decrescente da data de aquisição*

Procedimento que insere os dados de um determinado cliente numa lista ligada de um só sentido por ordem decrescente da data da aquisição.

*Procura (Último / Data de Aquisição / Nome / id) da Lista Cliente*

Função que, com o auxílio de dois ponteiros iteradores, i (anterior) e j (atual), ambos do tipo lista cliente, percorre a lista de clientes. O ponteiro j funciona como decisor e o ponteiro i atualiza a ligação.

*Cria lista*

Função que cria uma nova lista do tipo lista cliente.

Outras funções e procedimentos foram implementados para o conjunto funcionar.

### **3.2 Ficheiro destino.c**

*Registar Destino*

Função que devolve um inteiro (id destino) e que permite verificar se um determinado destino existe na lista de destinos.

*Imprime voos de um destino*

Procedimento que permite imprimir todos os voos de um determinado destino por ordem decrescente da data de partida.

*Imprime Destinos de um Cliente*

Procedimento que permite a impressão dos clientes que compraram e que estão em espera de um determinado destino por ordem crescente da data de aquisição do destino, i.e., os mais antigos primeiro.

*Imprime Lista de Destinos*

Procedimento que imprime por ordem alfabética os destinos da agência.

*Inserir na lista Destino por ordem alfabética*

Procedimento que insere os dados de um determinado destino numa lista ligada de um só sentido por ordem alfabética.

#### *Inserir na lista Destino por ordem crescente da data de aquisição*

Procedimento que insere os dados de um determinado destino numa lista ligada de um só sentido por ordem crescente da data da aquisição.

#### *Procura (Último / Data de Aquisição / Data Voo / Nome / id) da Lista Destino*

Função que, com o auxílio de dois ponteiros iteradores, i (anterior) e j (atual), ambos do tipo lista destino, percorre a lista de destinos. O ponteiro j funciona como decisor e o ponteiro i atualiza a ligação.

#### *Cria lista*

Função que cria uma nova lista do tipo lista destino.

Outras funções e procedimentos foram implementados para o conjunto funcionar.

### **3.3 Ficheiro save.c**

#### *Salvar lista de clientes*

Função que guarda num ficheiro do tipo txt os dados relativos a clientes (ficheiro que funciona como base de dados). A função é chamada quando o utilizador introduz um novo cliente manualmente. A função abre o ficheiro inicial (modo “a”) e acrescenta no final do mesmo uma linha com a informação.

#### *Salvar sessão*

A cada modificação (desistência de uma reserva, cancelamento de uma viagem, introdução de um cliente) uma sessão é salva num ficheiro sessão.txt, nomeadamente, id cliente, id destino e data de aquisição. Para tal é usada uma lista ligada auxiliar nova com toda informação. No final da escrita do ficheiro, aberto inicialmente em modo “w”) esta lista é destruída libertando assim toda a memória.

#### *Imprime lista save*

Procedimento que imprime a lista auxiliar save

*Percorre a lista de clientes*

Função que vai percorrer a lista de clientes e para cada cliente lê o respetivo conteúdo das listas de destinos comprados e de destinos em espera.

*Inserir em ordem da data de aquisição*

Função que insere na lista ligada auxiliar save o nó save por ordem crescente da data de aquisição, i.e., da data mais antiga para a data mais próxima de hoje.

Outras funções e procedimentos foram implementados para o conjunto funcionar.

### **3.4 Ficheiro cadeias.c**

Para manipular as strings optou-se por usar o menos possível as funções já implementadas nos ficheiros <\*.h> da linguagem C. Exercício que atrasou a elaboração do trabalho mas que permitiu uma compreensão plena das cadeias de caracteres. Apenas se enumeram uma vez que é fácil entender o seu propósito através da leitura do seu próprio nome.

- Apaga um Carater String
- Comprimento String
- Concatenar Strings
- Conta dígitos
- Atoi melhorado sem sinal
- String Trim
- Tokenizer String

### **3.5 Ficheiro gestão.c**

O ficheiro gestão.c apresenta todas as funções necessárias para a manipulação em conjunto das listas ligadas clientes e destinos. Enumeram-se resumidamente essas funções.



#### *Atribui uma venda*

Função que recebe uma linha do ficheiro compras.txt com o id do cliente, o id do destino e a data da aquisição. Como o id cliente faz uma pesquisa na lista ligada cliente e devolve o seu correspondente ponteiro. Com o id destino faz uma pesquisa na lista destino e devolve o seu correspondente ponteiro. Uma função insere compras vai percorrer a lista ligada de destinos deste cliente e caso haja vaga insere na lista de destinos comprados, caso contrário, insere na lista de destinos em espera. Ao mesmo tempo e consoante a inserção anterior as listas de clientes compraram e de clientes espera do respetivo destino são atualizadas. A inserção em lista de clientes é por data de aquisição decrescente e a inserção em lista de destinos é por data de aquisição crescente. Os dados relativos à estrutura destino, nomeadamente, viagens compradas e viagens em fila são atualizados.

#### *Cancelar viagem*

Função que percorre a lista de destinos de um dado cliente que quer desistir da viagem e elimina o destino (libertação de memória). O mesmo é feito na lista de clientes que compraram este destino. Uma verificação é efetuada na lista de clientes em espera do destino em causa. Se o destino apresenta clientes em espera o cliente que adquiriu em primeiro o destino é promovido para a lista de clientes compraram do destino. O mesmo é feito nas listas de destino do cliente promovido. Os dados relativos à estrutura destino, nomeadamente, viagens compradas e viagens em fila são atualizados.

#### *Cancelar reserva*

Função elimina da lista de destino em espera de um dado cliente a reserva (libertação de memória). Os dados relativos à estrutura destino, nomeadamente, viagens compradas e viagens em fila são atualizados.

Para todas estas funções outras há para o pleno funcionamento das mesmas.

### **3.6 Função main.c**

Função que lê três ficheiros do tipo txt, nomeadamente, Clientes.txt, Destinos.txt e Compras.txt. Os ficheiros Clientes.txt e Destinos.txt permitem construir as listas ligadas

lista de clientes (mais de 200 nós) e lista de destinos (mais de 300 nós), respetivamente. O ficheiro Compras.txt apresenta a atribuição de determinado destino a determinado cliente (mais de 7600 atribuições). As listas foram criadas descarregando dados disponibilizados na internet e gerando combinações aleatórias através do excel.

Os métodos disponibilizados são os seguintes:

1. Vender uma viagem
2. Cancelar uma viagem
3. Cancelar uma reserva
4. Listar clientes
5. Listar destinos
6. Listar destinos de um cliente
7. Listar clientes de um destino
8. Listar voos de um destino
9. Salvar sessão
10. Exit

Todas as entradas de dados na consola são testadas. Testes mais finos poderiam ser implementados em determinadas situações, no entanto, optou-se pelo estudo de matérias mais exigentes.

## 4 Conclusão

Atendendo à interpretação do enunciado do mini projeto é-se da opinião de que o trabalho aqui apresentado permite responder na plenitude às questões. A implementação da função regista cliente novo e atribuí-lhe um determinado destino não é exigida.