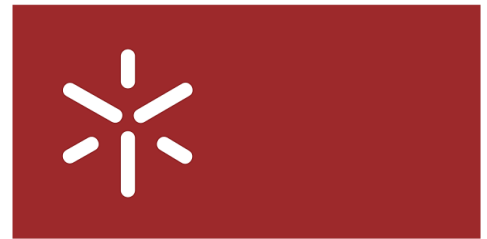


Relatório

MP2-MIETI



Universidade do Minho

HORÁRIOS DOS COMBOIOS BRAGA-LISBOA



Intercidades / Alfa Pendular	Braga
	Famalicão
	Porto
	Aveiro
	Coimbra
	Entroncamento
	Santarém
	Lisboa

Trabalho realizado por:

-Marcos Martins **A84481**;

-Rui Freitas **A84121**;

Índice

- Introdução (p.3);
- Enunciado do problema (p.4);
- Descrição da solução (p.5);
- Código (p.6-13);
- Exemplo (p.14);
- Conclusão (p.15);

Introdução

Neste trabalho prático de MP2 foi-nos proposto realizar um projeto em C cujo objetivo é apresentar os horários de comboio que ligam Braga a Lisboa com paragens em Famalicão, Porto, Aveiro, Coimbra, Entroncamento e Santarém.

Os objetivos de aprendizagem deste projeto têm por base a utilização de estruturas e de arrays assim como aprender a utilizar ficheiros.

Após a elaboração deste trabalho esperamos dominar minimamente o uso de estruturas e também praticar o uso de ficheiros em C.

Para a realização deste trabalho prático iremos utilizar conceitos lecionados na cadeira de Métodos de Programação II assim como na cadeira de Métodos de Programação I, visto que são 2 cadeiras que se complementam na aprendizagem da linguagem “C”.

Enunciado do problema

Neste problema pretende-se armazenar o horário diário dos comboios rápidos (AP-alfa ou IC-intercidades) que ligam Braga a Lisboa, com paragens em Famalicão, Porto, Aveiro, Coimbra, Entroncamento e Santarém.

O programa deve ler o identificador e o tipo de cada comboio (IC ou AP), e a seguir a lista de cidades onde pára e a respetiva hora de partida. Desta forma o programa poderá responder a perguntas tais como:

- Qual o próximo comboio para Lisboa, a partir de uma estação X e depois da hora atual?
- Quais os comboios diários, genéricos ou de um dado tipo (IC/AP), que ligam duas cidades X e Y do percurso?
- Qual o comboio mais rápido de Braga a Lisboa?

Os dados introduzidos devem ser gravados num ficheiro binário para que possam ser recuperados numa sessão de trabalho posterior.

Descrição da solução

Visto que o objetivo deste projeto seria a análise de horários de comboio que ligam as cidades de Braga e Porto, começamos por introduzir horários tanto Intercidades como Alfa Pendular no código em C.

Após termos os horários definidos tivemos de arranjar uma solução para responder a perguntas como: Qual o comboio mais próximo a sair da estação X após a hora atual, assim como outras perguntas relacionadas com os horários.

Para responder a estas questões tivemos de utilizar matéria lecionada nas aulas de MP2, como estruturas e arrays, que foi no que se baseou este trabalho prático. Depois tivemos de utilizar ficheiros de modo a guardar a ultima pesquisa executada pelo utilizador para que na próxima sessão estivesse guardada.

Após a elaboração deste projeto sentimo-nos mais confortáveis na manipulação de arrays e estruturas em C, sentindo que cumprimos o objetivo do trabalho.

Código

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4  #include <string.h>
5  #define MAX 50
6  #define SMAX 100
7
8  FILE *lr;          //declaração do file
9
10 typedef struct {
11     char ap_nome[MAX];
12     char ap_paragens[MAX];
13 } ComboiosAp; // declara tipo e paragens dos comboios
14 ComboiosAp AP[SMAX];
15
16 char aps_nome[][MAX]={"AP_102","AP_204","AP_320","AP_450"};
17
18 typedef struct {
19     char ic_nome[MAX];
20     char ic_paragens[MAX];
21 } ComboioIC;
22 ComboioIC IC[SMAX];
23
24 char ics_nome[][MAX]={"IC_089","IC_160","IC_203","IC_344"};
25 char paragens[][MAX]={"Braga","Famalicão","Porto","Aveiro","Coimbra","Entroncamento","Santarem","Lisboa"};
26
27 void def_variaveis() {
28     int i;
29
30     for(i=0;i<4;i++){
31         strcpy(AP[i].ap_nome,aps_nome[i]);
32         strcpy(IC[i].ic_nome,ics_nome[i]);
33     }
34
35     for(i=0;i<8;i++){
36         strcpy(AP[i].ap_paragens,paragens[i]);
37         strcpy(IC[i].ic_paragens,paragens[i]);
38     }
39 } // funcao que atribui os nomes e paragens para as funcoes
40
41 int tp(int *times){
42     int hours, minutes;
43     time_t now;
44     time(&now);
45
46     struct tm *local = localtime(&now);
47
48     hours = local->tm_hour;
49     minutes = local->tm_min;
50
51     *times=(hours%100)*100+(minutes%100);
52
53     return *times;
54 } // verifica a hora do sistema
55
56 void first_case(int AP_H[8][4], int IC_H[8][4], int *hora, int *minutos, int *partida, int *destino, int *aux_AP, int *aux_IC, int *tipe){
57     int i;
58     int tempo_M=(*hora)*60+(*minutos);
59     int horas, min;
60     int reg_AP=0;
61     int reg_IC=0;
62
63     for(i=0;i<4;i++){
64         if(AP_H[*partida][i]>=tempo_M){
65             *aux_AP=i;
66             break;
67         }
68         if((i==3)&&(AP_H[*partida][i]<tempo_M)&&(tempo_M<=1439)){
69             tempo_M=0;
70             i=0;
71             req_AP=1;
72         }
73     }
74 }
```

```

75         *aux_AP=0;
76         break;
77     }
78
79 }
80 tempo_M=(*hora)*60+(*minutos);
81 for(i=0;i<4;i++){
82     if(IC_H[*partida][i]>=tempo_M){
83         *aux_IC=i;
84         break;
85     }
86     if((i==3)&&(IC_H[*partida][i]<tempo_M)&&(tempo_M<=1439)){
87         tempo_M=0;
88         i=0;
89         reg_IC=1;
90         *aux_IC=0;
91         break;
92     }
93 }
94
95
96 if((reg_AP!=1)&&(reg_IC!=1)|| (reg_AP==1)&&(reg_IC==1)){
97     if((AP_H[*partida][*aux_AP]<(IC_H[*partida][*aux_IC])){
98         horas=AP_H[*partida][*aux_AP]/60;
99         min=AP_H[*partida][*aux_AP]%60;
100         printf("\n\nO comboio mais proximo sera o %s.\n",AP[*aux_AP].ap_nome);
101         printf("Partida de %s as %02d:%02dh.\n",AP[*partida].ap_paragens,horas,min);
102         horas=AP_H[*destino][*aux_AP]/60;
103         min=AP_H[*destino][*aux_AP]%60;
104         printf("Chegada a %s as %02d:%02dh.\n\n",AP[*destino].ap_paragens,horas,min);
105         *tipe=0;
106     }
107     if((AP_H[*partida][*aux_AP])>(IC_H[*partida][*aux_IC])){
108         horas=IC_H[*partida][*aux_IC]/60;
109         min=IC_H[*partida][*aux_IC]%60;
110         printf("\n\nO comboio mais proximo sera o %s.\n",IC[*aux_IC].ic_nome);
111         printf("Partida de %s as %02d:%02dh.\n",IC[*partida].ic_paragens,horas,min);

```

```

112         horas=IC_H[*destino][*aux_IC]/60;
113         min=IC_H[*destino][*aux_IC]%60;
114         printf("Chegada a %s as %02d:%02dh.\n\n",IC[*destino].ic_paragens,horas,min);
115         *tipe=1;
116     }
117 }
118
119 if((reg_AP==1)&&(reg_IC!=1)){
120     horas=IC_H[*partida][*aux_IC]/60;
121     min=IC_H[*partida][*aux_IC]%60;
122     printf("\n\nO comboio mais proximo sera o %s.\n",IC[*aux_IC].ic_nome);
123     printf("Partida de %s as %02d:%02dh.\n",IC[*partida].ic_paragens,horas,min);
124     horas=IC_H[*destino][*aux_IC]/60;
125     min=IC_H[*destino][*aux_IC]%60;
126     printf("Chegada a %s as %02d:%02dh.\n\n",IC[*destino].ic_paragens,horas,min);
127     *tipe=1;
128 }
129 if((reg_AP!=1)&&(reg_IC==1)){
130     horas=AP_H[*partida][*aux_AP]/60;
131     min=AP_H[*partida][*aux_AP]%60;
132     printf("\n\nO comboio mais proximo sera o %s.\n",AP[*aux_AP].ap_nome);
133     printf("Partida de %s as %02d:%02dh.\n",AP[*partida].ap_paragens,horas,min);
134     horas=AP_H[*destino][*aux_AP]/60;
135     min=AP_H[*destino][*aux_AP]%60;
136     printf("Chegada a %s as %02d:%02dh.\n\n",AP[*destino].ap_paragens,horas,min);
137     *tipe=0;
138 }
139
140 }
141
142 void expanded_case(int AP_H[6][4], int IC_H[8][4], int *hora, int *minutos, int *partida, int *destino,int *aux_AP, int *aux_IC,int *tipe){
143     int i;
144     int horas,min;
145
146     for(i=*partida;i<=*destino;i++){
147
148         if(*tipe==0){

```

```

149         horas=AP_H[i][*aux_AP]/60;
150         min=AP_H[i][*aux_AP]%60;
151         printf("%02d:%02d\n -> %s\n",horas,min,AP[i].ap_paragens);
152     }
153     if(*tipe==1){
154         horas=IC_H[i][*aux_IC]/60;
155         min=IC_H[i][*aux_IC]%60;
156         printf("%02d:%02d\n -> %s\n",horas,min,IC[i].ic_paragens);
157     }
158 }
159
160
161 }
162
163 void second_case(int AP_H[8][4],int IC_H[8][4],int *hora, int *minutos){
164
165     int i,j;
166     int choice;
167
168     printf("\n\n Qual a linha de comboios quer apresentar?\n[0] AP - alpha    \n[1] IC - intercidade    \n[2] Ambos\n\nOpcao: ");
169     scanf("%d",&choice);
170     for(i=0;i<5;i++){
171         if((choice<0)|| (choice>2)){
172             printf("\nOpcao invalida. Introduza novamente.\nOpcao: ");
173             scanf("%d",&choice);
174         }else{
175             break;
176         }
177     }
178     if(choice==0){
179         printf("\n\n ");
180         for(i=0;i<4;i++){
181             printf("%s",AP[i].ap_nome);
182         }
183         for(i=0;i<8;i++){
184             printf("\n ");
185             for(j=0;j<4;j++){

```

```

186                 *hora=AP_H[i][j]/60;
187                 *minutos=AP_H[i][j]%60;
188                 printf("%02d:%02d",*hora,*minutos);
189             }
190             printf("%s\n",AP[i].ap_paragens);
191         }
192         printf("\n\n");
193     }
194
195     if(choice==1){
196         printf("\n\n ");
197         for(i=0;i<4;i++){
198             printf("%s",IC[i].ic_nome);
199         }
200         for(i=0;i<8;i++){
201             printf("\n ");
202             for(j=0;j<4;j++){
203                 *hora=IC_H[i][j]/60;
204                 *minutos=IC_H[i][j]%60;
205                 printf("%02d:%02d",*hora,*minutos);
206             }
207             printf("%s\n",IC[i].ic_paragens);
208         }
209         printf("\n\n");
210     }
211
212     if(choice==2){
213         printf("\n\n ");
214         for(i=0;i<4;i++){
215             printf("%s",AP[i].ap_nome);
216         }
217         for(i=0;i<6;i++){
218             printf("\n ");
219             for(j=0;j<4;j++){
220                 *hora=AP_H[i][j]/60;
221                 *minutos=AP_H[i][j]%60;
222                 printf("%02d:%02d",*hora,*minutos);

```



```

223         }
224         printf("%s\n", AP[i].ap_paragens);
225     }
226     printf("\n\n");
227
228     for(i=0; i<4; i++){
229         printf("  %s ", IC[i].ic_nome);
230     }
231     for(i=0; i<8; i++){
232         printf("\n  ");
233         for(j=0; j<4; j++){
234             *hora=IC_H[i][j]/60;
235             *minutos=IC_H[i][j]%60;
236             printf("%02d:%02d  ", *hora, *minutos);
237         }
238         printf("%s\n", IC[i].ic_paragens);
239     }
240     printf("\n\n");
241
242 }
243
244 }
245
246 int main() {
247
248     int i=0;
249     int times;
250     int AP_H[8][4]={ {241, 601, 841, 1081}, // {0401, 1001, 1401, 1801}
251                      {257, 617, 857, 1097}, // {0417, 1017, 1417, 1817}
252                      {280, 640, 880, 1120}, // {0440, 1040, 1440, 1840}
253                      {315, 675, 915, 1155}, // {0515, 1115, 1515, 1915}
254                      {344, 704, 944, 1184}, // {0544, 1144, 1544, 1944}
255                      {348, 708, 948, 1188}, // {0548, 1148, 1548, 1948}
256                      {367, 727, 967, 1207}, // {0607, 1207, 1607, 2007}
257                      {438, 805, 1042, 1282}}; // {0718, 1325, 1722, 2122}
258
259

```

```

260     int IC_H[8][4]={ {485, 785, 965, 1205}, // {0805, 1305, 1605, 2005}
261                     {501, 801, 981, 1221}, // {0821, 1321, 1621, 2021}
262                     {525, 825, 1005, 1245}, // {0845, 1345, 1645, 2045}
263                     {565, 865, 1045, 1285}, // {0925, 1425, 1725, 2125}
264                     {597, 897, 1077, 1317}, // {0957, 1457, 1757, 2157}
265                     {661, 961, 1141, 1381}, // {1101, 1601, 1901, 2301}
266                     {680, 980, 1160, 1400}, // {1120, 1620, 1920, 2320}
267                     {708, 1012, 1197, 1425}}; // {1148, 1652, 1457, 2345}
268
269     def_variaveis();
270     tp(&times);
271
272     int input;
273     int partida;
274     int destino;
275     int escolha;
276     int hora, minutos, tempo;
277     int aux_AP, aux_IC;
278     int ret_menu;
279     int menu=0;
280     int choice;
281     int tipe;
282
283     while(menu==0) {
284
285         printf("*****\n");
286         printf("----- Choose an option: ----- \n");
287         printf(" \n");
288         printf(" [1] -> Partida e Destino \n");
289         printf(" [2] -> Horarios \n");
290         printf(" [3] -> Comboio mais rapido \n");
291         printf(" [4] -> Ultima pesquisa \n");
292         printf(" [0] -> Terminar o programa \n");
293         printf("\n*****\n");
294
295         printf("\nOpcao: ");
296         scanf("%d", &input);

```

```

297
298     if(input==0){
299         return 0;
300     }
301
302     switch(input){
303
304     case 1:
305         if(input==1){
306             system ("cls");
307             printf("Partida:          \n");
308             for(i=0;i<7;i++){
309                 printf("[%d] -> %s\n",i,IC[i].ic_paragens);
310             }
311             printf("\nOpcao: ");
312             scanf("%d",&partida);
313             for(i=0;i<5;i++){
314                 if((partida<0)|| (partida>6)){
315                     printf("Opcao invalida. Escolha novamente.\n\nOpcao: ");
316                     scanf("%d",&partida);
317                 }else{
318                     break;
319                 }
320             }
321
322             printf("\nDestino: \n");
323             for(i=partida+1;i<8;i++){
324                 printf("[%d] -> %s\n",i,IC[i].ic_paragens);
325             }
326             printf("\nOpcao: ");
327             scanf("%d",&destino);
328             for(i=0;i<5;i++){
329                 if((destino<partida+1)|| (destino>7)){
330                     printf("Opcao invalida. Escolha novamente.\n\nOpcao: ");
331                     scanf("%d",&destino);
332                 }else{
333                     break;

```

```

334     }
335 }
336
337 printf("\nIntroduza:\n1-Usar hora atual\n2-Introduzir hora manualmente\n\nOpcao: ");
338 scanf("%d",&escolha);
339
340 for(i=0;i<5;i++){
341     if((escolha<1)|| (escolha>2)){
342         printf("Opcao invalida. Escolha novamente.\n");
343         scanf("%d",&escolha);
344     }else{
345         break;
346     }
347 }
348
349 if(escolha==1){
350     hora=times/100;
351     minutos=times%100;
352 }
353 if(escolha==2){
354     for(i=0;i<10;i++){
355         printf("Horas: ");
356         scanf("%d",&hora);
357         if((hora>=0)&&(hora<24)){
358             break;
359         }else{
360             printf("Hora introduzida inválida. Introduza novamente.\n");
361         }
362     }
363     for(i=0;i<10;i++){
364         printf("Minutos: ");
365         scanf("%d",&minutos);
366         if((minutos>=0)&&(minutos<60)){
367             break;
368         }else{
369             printf("Hora introduzida invalida. Introduza novamente.\n");
370         }

```



```

519     first_case(AP_H, IC_H, &hora, &minutos, &partida, &destino, &aux_AP, &aux_IC, &tipo);
520
521     printf("Deseja apresentar a extensao do horario?\n[0]NAO    [1]SIM\n\nOpcao: ");
522     scanf("%d", &choice);
523
524     if(choice==1){
525         system("cls");
526         printf("ULTIMA PESQUISA:");
527         printf("    PARTIDA: %s.    DESTINO: %s.    HORAS: %02d:%02d.\n", IC[partida].ic_paragens, IC[destino].ic_paragens, hora, minutos);
528
529         first_case(AP_H, IC_H, &hora, &minutos, &partida, &destino, &aux_AP, &aux_IC, &tipo);
530         expanded_case(AP_H, IC_H, &hora, &minutos, &partida, &destino, &aux_AP, &aux_IC, &tipo);
531     }
532     fclose(lr);
533
534     printf("\nDESEJA RETORNAR AO MENU ANTERIOR?\n[0]NAO    [1]SIM\n\nOpcao: ");
535     scanf("%d", &ret_menu);
536
537     if(ret_menu==1){
538         system("cls");
539     }else{
540         return;
541     }
542
543     break;
544 }
545
546 default:
547     printf("\nOpcao invalida. Introduza novamente.");
548     for (i=1; i<2; i++){
549         sleep(1);
550         fflush(stdout);
551     }
552     system("cls");
553 }
554 return;
555 }

```

Exemplo

Vamos supor que o utilizador está interessado em saber qual o comboio mais rápido com partida em Aveiro e com destino Lisboa a partir das 10 da manhã, este seria o output esperado:

```
PARTIDA: Aveiro. DESTINO: Lisboa. HORAS: 10:00.

O comboio mais proximo sera o AP_204.
Partida de Aveiro as 11:15h.
Chegada a Lisboa as 13:25h.

11:15h -> Aveiro
11:44h -> Coimbra
11:48h -> Entroncamento
12:07h -> Santarem
13:25h -> Lisboa

DESEJA RETORNAR AO MENU ANTERIOR?
[0]NAO [1]SIM

Opcao:
```

Por outro lado, se o utilizador quiser apenas ver os horários disponíveis com comboio do tipo Intercidades (IC) num certo dia, este seria o output esperado:

```
Qual a linha de comboios quer apresentar?
[0] AP - alpha
[1] IC - intercidade
[2] Ambos

Opcao: 1

IC_089 IC_160 IC_203 IC_344
08:05 13:05 16:05 20:05 Braga
08:21 13:21 16:21 20:21 Famalicao
08:45 13:45 16:45 20:45 Porto
09:25 14:25 17:25 21:25 Aveiro
09:57 14:57 17:57 21:57 Coimbra
11:01 16:01 19:01 23:01 Entroncamento
11:20 16:20 19:20 23:20 Santarem
11:48 16:52 19:57 23:45 Lisboa

DESEJA RETORNAR AO MENU ANTERIOR?
[0]NAO [1]SIM

Opcao: █
```

Conclusão

Este trabalho prático proporcionou-nos uma ampla abordagem e uma grande capacidade de programação na linguagem C, através do uso da manipulação de structs e arrays.

Após a elaboração deste projeto podemos dizer que concluímos com sucesso o que foi pedido pelo professor da cadeira que era criar um código em C que permitisse analisar horários de comboio entre Braga e Lisboa.

Podemos também afirmar que após a finalização do trabalho conseguimos dominar minimamente o uso de estruturas e de ficheiros, que eram os objetivos principais deste TP.