Processamento de Linguagens e Compiladores

3º ano de LCC

Trabalho Prático 1

Relatório de Desenvolvimento

João Gomes	Luís Martins	Luís Ventuzelos
a70400	a66093	a73002

8 de Outubro de 2016

Resumo

Este primeiro trabalho prático consiste no tratamento de dados existentes num ficheiro em formato **XML** mais especificamente no ficheiro *viaverde.xml*, que corresponde às leituras da *Via Verde* mensais de um utilizador, usando a linguagem **AWK**.

Conteúdo

Capítulo 1

Introdução

Este relatório foi realizado no âmbito da unidade curricular de *Processamento de Linguagens e Compiladores*, no curso *Ciências da Computação* da *Universidade do Minho*, sendo o trabalho proposto à componente prática da mesma, que consiste em:

- aumentar o nosso conhecimento das ferramentas de apoio à programação em ambiente Linux;
- aumentar a capacidade de escrever Expressões Regulares (ER) para descrição de padrões de frases;
- praticar a partir da criação de ERs, com o intuito de filtrarem ou transformem textos;
- utilizar o sistema de produção para filtros de texto **AWK**.

Pretende-se, assim, dar uma ideia de todo o trabalho envolvido, durante a produção do mesmo, nomeadamente da:

- preparação;
- estudo do problema;
- modo de abordagem do problema;
- solução para os problemas encontrados.

Estrutura do Relatório

O documento, está organizado em 5 capitulos, sendo:

- *capitulo 1* a *Introdução* e *Estrutura*
- *capitulo 2 Análise* e *Especificação* em que analisamos os padrões de frases que pretendiamos encontrar através de Expressões Regulares.
- *capitulo 3 Conceção* em que identificar as estruturas de dados a usar no decorrer do processamento do texto fonte.
- *capitulo 4 Codificação* consiste no desenvolvimento de Filtros de Texto, que recebem ERs para reconhecer padrões, e posteriormente, serem trabalhados, com recurso ao **AWK**.
- *capitulo 5 Conclusão* síntese do que foi estudado, as conclusões mais importantes, e o trabalho futuro.

Capítulo 2

Análise e Especificação

2.1 Descrição informal do problema

Competia-nos manipular um ficheiro em formato **XML** de modo a extrairmos o conteudo através de expressões regulares. O objetivo (mínimo) consiste em:

- Programa A - Listar o número de pórticos *Via Verde* de entrada diários;
- Programa B - Listar o nome dos pórticos *Via Verde* de saída;
- Programa C - Listar o total gasto em cada mês, sem distinção entre parques de estacionamento e portagens;
- Programa D - Listar o total gasto em cada mês, em parques de estacionamento.

Outros processamentos:

- Programa E e F - Uma pequena interação com o utilizador, com o objetivo de lhe apresentar os trajetos , feitos, no **Google Maps**;
- Programa G - Total de tempo passado na autoestrada;
- Programa H - Total de tempo passado em parques de estacionamento.

2.2 Especificação dos Requisitos

2.2.1 Dados

Os dados relativos ao utilizador estavam disponíves nas 13 primeiras linhas, sendo as seguintes relativas às transações, como mostram os seguintes exemplos.

Dados Cliente

```
<EXTRACTO id="011114056/08/2015">
<MES_EMISSAO>Ago-2015</MES_EMISSAO>
<CLIENTE id="514714936">
<NIF>987653210</NIF>
<NOME>PEDRO MANUEL RANGEL SANTOS HENRIQUES</NOME>
<MORADA>RUA XXX</MORADA>
<LOCALIDADE>BRAGA</LOCALIDADE>
<CODIGO_POSTAL>4715-012 BRAGA</CODIGO_POSTAL>
</CLIENTE>
<IDENTIFICADOR id="28876820811">
<MATRICULA>00-LJ-11</MATRICULA>
<REF_PAGAMENTO>1234567</REF_PAGAMENTO>
<TRANSACCAO>
```

Portagens

```
<TRANSACCAO>
<DATA_ENTRADA>26-07-2015</DATA_ENTRADA>
<HORA_ENTRADA>11:45</HORA_ENTRADA>
<ENTRADA>Aeroporto</ENTRADA>
<DATA_SAIDA>26-07-2015</DATA_SAIDA>
<HORA_SAIDA>11:50</HORA_SAIDA>
<SAIDA>Ponte Pedra</SAIDA>
<IMPORTANCIA>0,65</IMPORTANCIA>
<VALOR_DESCONTO>0,00</VALOR_DESCONTO>
<TAXA_IVA>23</TAXA_IVA>
<OPERADOR>I. de Portugal (P3)</OPERADOR>
<TIPO>Portagens</TIPO>
<DATA_DEBITO>05-08-2015</DATA_DEBITO>
<CARTAO>6749036</CARTAO>
</TRANSACCAO>
```

Capítulo 3

Conceção

3.1 Estruturas de Dados

Em todos os programas foi usada a seguinte estrutra, própria do **AWK**:

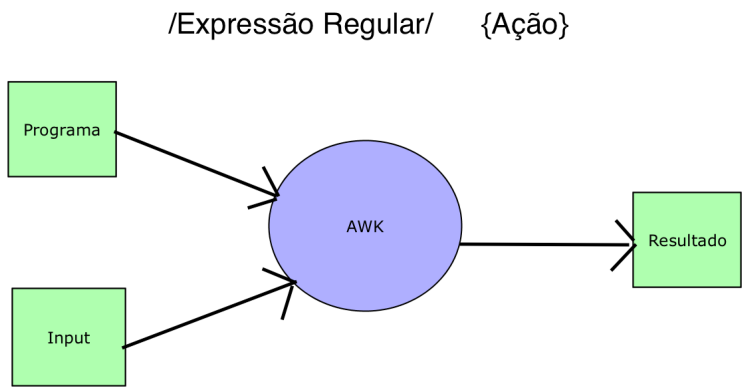

Figura 3.1: Funcionamento do **AWK**

1. Detetar a expressão regular adequada ao problema
2. Desenvolver a ação de maneira a satisfazer o nosso problema
3. Resultado final

No exemplo seguinte, temos como expressão regular /<DATA_ENTRADA>/, seguida de uma ação.

```
/<DATA_ENTRADA>/ {split ($0,data,"<|>");contas[data[3]]++ }
```

Capítulo 4

Codificação e Testes

```
/<DATA_SAIDA>/ {split ($0,data,"<|>");
          split(data[3],mes,"-");
          while(data[2]!= "IMPORTANCIA"){
             getline; # proxima linha
             split ($0,data,"<|>");
          }
          split($0, valor,"<|>");
          sub(",",".",valor[3]); #para substituir a virgula pelo ponto
          datas[mes[2]]+=valor[3];
          total+=valor[3];
          }
```

4.1 Alternativas, Decisões e Problemas de Implementação

Na realização deste trabalho deparamo-nos inicialmente com problemas menores que facilmente foram resolvidos.

O primeiro problema que nos surgiu foi, depois de usarmos uma expressão regular para trabalhar numa certa linha do nosso ficheiro, como é que iriamos percorrer as linhas pelo ficheiro e, ao mesmo tempo, preservar a informação encontrada anteriormente. Contornamos essa situaçao usando um *getline* dentro de um ciclo *while* como se pode ver no exemplo em baixo:

```
while(data[2]!= "IMPORTANCIA"){
             getline; # proxima linha
             split ($0,data,"<|>");
          }
```

Mais para a frente, já na parte de valorização, queriamos implementar um programa que fosse capaz de traçar no mapa (**Google Maps**) o trajeto entre cada cidade, de modo a conseguirmos extrair a distancia entre essas duas cidades, para depois calcularmos a velocidade médida do condutor. Infelizmente enão conseguimos concretizar este objetivo. Por um lado, para termos acesso ao website correspondente ao trajeto desejado, tinhamos que abrir uma janela do browser e, caso estivessemos a tratar muitos trajetos não fazia sentido abrirmos o mesmo número de janelas no browser. Por outro lado, não sabiamos como retirar a informação desejada utilizando apenas a linguagem **AWK**. Assim sendo, decidimos realizar uma tarefa mais simples ,que apenas abria os links correspondentes aos trajetos escolhidos pelo utilizador. Apesar da grande simplicidade desta nossa escolha, em relação à anterior, encontramos-nos com outro problema: o **AWK** não tem uma função de *stdin* e por isso tivemos que usar novamente a função *getline* do **AWK**, do seguinte modo:

```
getline x < "/dev/tty";
```

Assim, podemos quardar os seguintes dados:

- A resposta, sim ou não, do utilizador à pergunta: **Pretende visualizar algum dos seus percursos ?**
- Os trajetos que o utilizador pretende visualizar, após lhe ser apresentada uma lista com todos os percursos por ele elaborados.

4.2 Testes realizados e Resultados

Mostram-se a seguir alguns testes feitos (valores introduzidos) e os respectivos resultados obtidos:

```
ventuzelos@ventuzelos-PC ~/Área de Trabalho/PLC/Trabalhos Praticos/TP1 $ gawk -f entradas.gawk viaverde.xml
Total: 41 entradas
Em 06-08-2015 fez 4 entradas
Em 10-08-2015 fez 7 entradas
Em 11-08-2015 fez 2 entradas
Em 13-08-2015 fez 5 entradas
Em 17-08-2015 fez 4 entradas
Em 18-08-2015 fez 2 entradas
Em 21-08-2015 fez 4 entradas
Em 26-07-2015 fez 3 entradas
Em 29-07-2015 fez 2 entradas
Em 30-07-2015 fez 3 entradas
Em 31-07-2015 fez 1 entradas
Em null fez 4 entradas
ventuzelos@ventuzelos-PC ~/Área de Trabalho/PLC/Trabalhos Praticos/TP1 $
```

Figura 4.1: Execução do Programa A

```
ventuzelos@ventuzelos-PC ~/Área de Trabalho/PLC/Trabalhos Praticos/TP1 $ gawk -f saidas.gawk viaverde.xml
saida no portico: Aeroporto
saida no portico: Angeiras N-S
saida no portico: Braga Sul
saida no portico: Custoias
saida no portico: EN107
saida no portico: EN 205 PV
saida no portico: Ermesinde PV
saida no portico: Ferreiros
saida no portico: Freixieiro
saida no portico: Lipor
saida no portico: Maia II
saida no portico: Maia PV
saida no portico: Neiva N-S
saida no portico: Neiva S-N
saida no portico: Ponte Pedra
saida no portico: Povoa S-N
saida no portico: PQ A Sa Carn.I
saida no portico: PQ Av. Central
saida no portico: Valongo
```

Figura 4.2: Execução do Programa B

```
ventuzelos@ventuzelos-PC ~/Área de Trabalho/PLC/Trabalhos Praticos/TP1 $ gawk -f total_mes.gawk viaverde.xml
Gasto do mês 08 => 57.1
Gasto do mês 07 => 20.3
Total gasto: 77.4
ventuzelos@ventuzelos-PC ~/Área de Trabalho/PLC/Trabalhos Praticos/TP1 $
```

Figura 4.3: Execução do Programa C

```
ventuzelos@ventuzelos-PC ~/Área de Trabalho/PLC/Trabalhos Praticos/TP1 $ gawk -f parques.gawk viaverde.xml
Gasto do mês 08 => 3
Gasto do mês 07 => 2
Total gasto em parques: 5
ventuzelos@ventuzelos-PC ~/Área de Trabalho/PLC/Trabalhos Praticos/TP1 $
```

Figura 4.4: Execução do Programa D

Ao executarmos **gawk -f parques.gawk viaverde.xml** o nosso programa vai criar o ficheiro de texto **trajetos.txt**, que serve para se fazer a ponte entre o ficheiro **viaverde.xml** e o ficheiro **links.txt**, que será explicado mais à frente.

```
Braga Sul --> Maia PV
Angeiras S-N --> Povoa S-N
Povoa N-S --> Angeiras N-S
 --> Aeroporto
EN13 O/E --> EN107
Maia II --> Braga Sul
Braga Sul --> Maia II
Ermesinde PV --> Valongo
EN14 E/O --> Lipor
Via Norte O/E --> Ponte Pedra
Valongo --> Ermesinde PV
Via Norte E/O --> Freixieiro
Angeiras S-N --> Povoa S-N
EN 205 PV --> Ferreiros
Ferreiros --> EN 205 PV
Povoa N-S --> Angeiras N-S
Maia PV --> Braga Sul
Ferreiros --> EN 205 PV
EN 205 PV --> Ferreiros
Ferreiros --> EN 205 PV
 --> Neiva S-N
 --> Neiva N-S
Povoa N-S --> Angeiras N-S
Aeroporto --> Ponte Pedra
Via Norte E/O --> Custoias
```

Figura 4.5: Excerto do ficheiro **trajetos.txt**

Ao executarmos **gawk -f mapa.awk trajetos.txt** o nosso programa vai criar o ficheiro **links.txt**, mencionado acima. Este ficheiro vai conter os links correspondentes a cada trajeto feito pelo utilizador, e como explicado no capítulo **4.1** o utilizador terá a oportunidade de observar esses mesmos percursos no mapa.

```
https://www.google.pt/maps/dir/Angeiras S-N/Povoa S-N
https://www.google.pt/maps/dir/EN 205 PV/Ferreiros
https://www.google.pt/maps/dir/Ferreiros/EN 205 PV
https://www.google.pt/maps/dir/EN 205 PV/Ferreiros
https://www.google.pt/maps/dir/Braga Sul/Maia PV
https://www.google.pt/maps/dir/Angeiras S-N/Povoa S-N
https://www.google.pt/maps/dir/Povoa N-S/Angeiras N-S
https://www.google.pt/maps/dir//Aeroporto
https://www.google.pt/maps/dir/EN13 O/E/EN107
https://www.google.pt/maps/dir/Maia II/Braga Sul
https://www.google.pt/maps/dir/Braga Sul/Maia II
https://www.google.pt/maps/dir/Ermesinde PV/Valongo
https://www.google.pt/maps/dir/EN14 E/O/Lipor
https://www.google.pt/maps/dir/Via Norte O/E/Ponte Pedra
https://www.google.pt/maps/dir/Valongo/Ermesinde PV
https://www.google.pt/maps/dir/Via Norte E/O/Freixieiro
https://www.google.pt/maps/dir/Angeiras S-N/Povoa S-N
https://www.google.pt/maps/dir/EN 205 PV/Ferreiros
https://www.google.pt/maps/dir/Ferreiros/EN 205 PV
https://www.google.pt/maps/dir/Povoa N-S/Angeiras N-S
https://www.google.pt/maps/dir/Maia PV/Braga Sul
https://www.google.pt/maps/dir/Ferreiros/EN 205 PV
https://www.google.pt/maps/dir/EN 205 PV/Ferreiros
```

Figura 4.6: Excerto do ficheiro **links.txt**

Ao executar o programa **gawk -f mapa.awk trajetos.txt** perguntamos ao utilizador se pretende ver a lista de percursos dado pelo ficheiro **trajetos.txt**. Caso o utilizador queira ver os trajetos, o programa imprime no terminal os trajetos numerados de 1,...,N em que N é o numero total de trajetos efectuados.

```
13 --> Ermesinde PV --> Valongo
14 --> EN14 E/O --> Lipor
15 --> Via Norte O/E --> Ponte Pedra
16 --> Valongo --> Ermesinde PV
17 --> Via Norte E/O --> Freixieiro
18 --> Angeiras S-N --> Povoa S-N
19 --> EN 205 PV --> Ferreiros
20 --> Ferreiros --> EN 205 PV
21 --> Povoa N-S --> Angeiras N-S
22 --> Maia PV --> Braga Sul
23 --> Ferreiros --> EN 205 PV
24 --> EN 205 PV --> Ferreiros
25 --> Ferreiros --> EN 205 PV
26 --> Povoa N-S --> Angeiras N-S
27 --> Aeroporto --> Ponte Pedra
28 --> Via Norte E/O --> Custoias
29 --> Angeiras S-N --> Povoa S-N
30 --> Povoa N-S --> Angeiras N-S
31 --> Angeiras S-N --> Povoa S-N
32 --> EN 205 PV --> Ferreiros
33 --> Ferreiros --> EN 205 PV
34 --> EN 205 PV --> Ferreiros
35 --> Ferreiros --> EN 205 PV
36 --> Ferreiros --> EN 205 PV
Insira os trajetos que pretende ver? (0 para cancelar)
1
Nova janela criada na sessão de browser existente
```

Figura 4.7: Execução do Programa F

De seguida pede-se ao utilizador para escolher o número do trajeto que prentende observar. Por exemplo caso escolha o trajeto 1 o programa vai abrir o link correspondente ao trajeto 1 (Povoa N-S - Angeiras N-S).

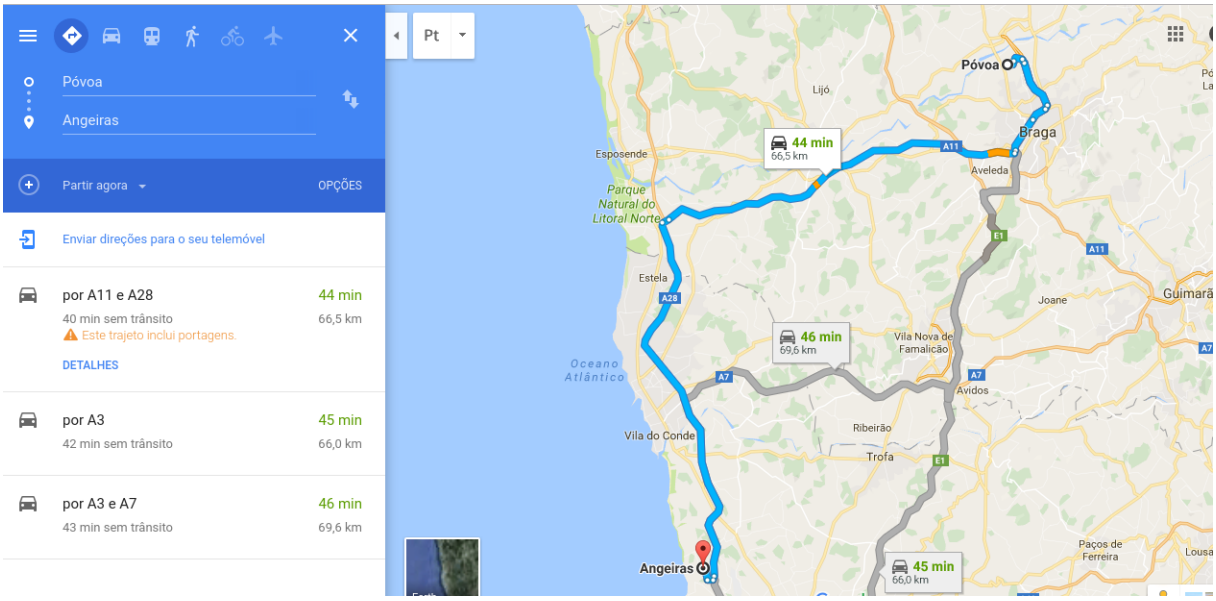

Figura 4.8: Trajeto Povoa N-S – Angeiras N-S

De seguida são apresentados dois exemplos do funcionamento dos programas **horas_estrada.awk** e **horas_parque.awk**, que calculam e apresentam o tempo total passado em autoestrada e em parque de estacionamento.

```
ventuzelos@ventuzelos-PC ~/Área de Trabalho/PLC/Trabalhos Praticos/TP1 $ gawk -f horas_estrada.awk viaverde.xml
Passou 66 horas e 29 minunos na autoestrada
ventuzelos@ventuzelos-PC ~/Área de Trabalho/PLC/Trabalhos Praticos/TP1 $
```

Figura 4.9: Execução do Programa G

```
ventuzelos@ventuzelos-PC ~/Área de Trabalho/PLC/Trabalhos Praticos/TP1 $ gawk -f horas_parques.awk viaverde.xml
Passou 2 horas e 51 minunos em parques de estacionamento
ventuzelos@ventuzelos-PC ~/Área de Trabalho/PLC/Trabalhos Praticos/TP1 $
```

Figura 4.10: Execução do Programa H

Capítulo 5

Conclusão

Com este trabalho foi-nos possível atingir os seguintes obejetivos:

- Fortalecer o conhecimento das ferramentas de programação em ambiente linux;
- Melhorar o conhecimento/manipulação das Expressões Regulares;
- Aumentar a capacidade de programação em **AWK**;
- Valorizar o nosso trabalho;
- Superar dificuldades e resultados menos optimistas;
- Melhorar o relacionamento e a capacidade de trabalhar em equipa;
- Aumentar o conhecimento, para um melhor desempenho no decurso da disciplina.

Apesar das dificuldades encontradas, ultrapassadas com mérito, esses obstáculos tornaram o trabalho mais desafiante e atrativo.
Com um conhecimento mais profundo das capacidades da programação em **AWK**, certamente conseguiríamos ter realizado um trabalho pratico com resultados de maior interesse para tarefas futuras.
Apesar de tudo, consideramos que o nosso desempenho foi excelente, e os resultados apresentados cativantes e de certo modo originais.

Apêndice A

Código do Programa

Programa A

```
/<DATA_ENTRADA>/{
      split ($0,data,"<|>");
      contas[data[3]]++
      }
END      { for (var in contas)
      {n+=contas[var];
         printf "Em " var " fez " contas[var] " entradas" "\n" |"sort" };
         printf "Total: " n " entradas" "\n";

}
```

Programa B

```
/<SAIDA>/ {
   split ($0,saida,"<|>");
   saidas[saida[3]]++}

END {for (s in saidas){printf "saida no portico: " s "\n" |"sort"}}
```

Programa C

```
/<DATA_SAIDA>/ {split ($0,data,"<|>");
           split(data[3],mes,"-");#mes[2] = mes
           while(data[2]!= "IMPORTANCIA"){
              getline; # proxima linha
              split ($0,data,"<|>");
           }
           split($0, valor,"<|>");
           sub(",",".",valor[3]);#para substituir a virgula pelo ponto
           datas[mes[2]]+=valor[3];
           total+=valor[3];
           }

END    {for (d in datas){print "Gasto do mês "d " => " datas[d]};
      print "Total gasto: " total;
   }
```

Programa D

```
/<DATA_SAIDA>/ {split ($0,data,"<|>");
           split(data[3],mes,"-");#mes[2] = mes
           while(data[2]!= "IMPORTANCIA"){
              getline; # proxima linha
              split ($0,data,"<|>");
           }
           split($0, valor,"<|>");
           while(data[2] != "TIPO"){
              getline;
              split ($0,data,"<|>");
           }
           if(data[3] == "Parques de estacionamento"){
              datas[mes[2]]+=valor[3];
              total+=valor[3];
           }
           }

END    {for (d in datas){print "Gasto do mês "d " => " datas[d]};
     print "Total gasto em parques: " total;
    }
```

Programa E

```
/<ENTRADA>/ {split ($0,local,"<|>");
         entrada = local[3];
         while(local[2] != "SAIDA"){
           getline;
           split ($0,local,"<|>")
         }
         print entrada " --> " local[3] > "trajetos.txt";
        }

END    {}
```

Programa F

```
BEGIN {link = "https://www.google.pt/maps/dir";
     l=0;
    }
   NR > 0 {
      split($0,trajeto," --> ");
      origem = "/" trajeto[1];
      destino = "/" trajeto[2];
      mapa = link origem destino
      l++;
      print "\""mapa"\"" > "links.txt";
   }
END {
   l++;
   print "Qual o seu sistema operativo ?(1 - Linux / 2 - Mac OSX)\n";
   getline so < "/dev/tty"; # para ler mesmo do teclado
   print "Pretende visualizar algum dos seus percursos ?(S/N)\n";
   getline x < "/dev/tty"; # para ler mesmo do teclado

   if (x == "S"){
      print "Esta é a lista de percursos que fez:\n";
      i = 1;
      while(i <= l){
         getline t < "trajetos.txt";
         print i " --> " t;
         i++;
      }
   print "Insira os trajetos que pretende ver? (0 para cancelar) "
   mp=1
   while (mp!=0){
      getline mp < "/dev/tty"
      if (mp==0) exit 1
      else
         if (so==1) system("awk 'NR==" mp "'" " links.txt | xargs google-chrome");
         if (so==2) system("awk 'NR==" mp "'" " links.txt | xargs open"); # xargs poe os argumentos a frente
             do open
      }

   }
   }
```

Programa G

```
/<HORA_ENTRADA>/{
              split ($0,aux,"<|>");
              split(aux[3],hora1,":")
              while(aux[2] != "HORA_SAIDA"){ # para chegar a linha certa
                 getline;
                 split ($0,aux,"<|>")
              }
              split(aux[3],hora2,":")
              hora1[1] = hora1[1]*60
              hora2[1] = hora2[1]*60
              while(aux[2] != "TIPO"){
                 getline;
                 split ($0,aux,"<|>")
              }
```

```
            if (aux[3] == "Portagens"){
               dif = (hora2[1]+hora2[2])-(hora1[1]+hora1[2])
               total+=dif
            }
         }

END  {
    print "Passou "int(total/60) " horas e " total%60 " minunos na autoestrada";
   }
```

Programa H

```
/<HORA_ENTRADA>/{
              split ($0,aux,"<|>");
              split(aux[3],hora1,":")
              while(aux[2] != "HORA_SAIDA"){ # para chegar a linha certa
                 getline;
                 split ($0,aux,"<|>")
              }
              split(aux[3],hora2,":")
              hora1[1] = hora1[1]*60
              hora2[1] = hora2[1]*60
              while(aux[2] != "TIPO"){
                 getline;
                 split ($0,aux,"<|>")
              }
              if (aux[3] == "Parques de estacionamento"){
                 dif = (hora2[1]+hora2[2])-(hora1[1]+hora1[2])
                 total+=dif
              }
           }

END  {
    print "Passou "int(total/60) " horas e " total%60 " minunos em parques de estacionamento";
   }
```