

Processamento de Linguagens
(3º ano de Mestrado Integrado em Engenharia Informática)
Trabalho Prático nº1 - Parte A
Relatório de Desenvolvimento

Daniel Malhadas
(A72293)

Alexandre Silva
(A72502)

13 de Março de 2017

Resumo

O presente documento descreve um estudo de desenvolvimento de filtros de texto em **GAWK**. Como caso prático de estudo foi escolhido como fonte a filtrar um conjunto de dados relativos a transacções no contexto da tecnologia **Via-Verde**. A fonte a filtrar é então um ficheiro de texto no formato **XML** com várias transacções e, os filtros idealizados, consistem na organização de uma grande quantidade de informação vinda do referido ficheiro de forma a que o olho humano facilmente encontre pormenores do seu interesse, sendo assim apresentada a informação de forma clara, limpa e sucinta.

Área: Processamento de Linguagens, GAWK, XML

Conteúdo

1	Introdução	2
2	Análise e Especificação	4
2.1	Descrição informal do problema	4
2.2	Especificação dos Requisitos	4
2.2.1	Dados	4
2.2.2	Pedidos	5
3	Concepção/desenho da Resolução	7
3.1	Estruturas de Dados	7
3.2	Algoritmos	9
4	Codificação e Testes	10
4.1	Alternativas, Decisões e Problemas de Implementação	10
4.2	Testes realizados e Resultados	10
5	Conclusão e Trabalho Futuro	18
A	Código do Programa	19

Capítulo 1

Introdução

Supervisor: Pedro Rangel Henriques

Supervisor: José João

No decorrer da Unidade Curricular **Processamento de Linguagens**, foi proposto aos autores, pelos docentes supervisores, um trabalho prático com o intuito de desenvolver os seguintes objetivos:

- Aumentar a experiência de uso do ambiente Linux e de algumas ferramentas de apoio à programação;
- Aumentar a capacidade de escrever Expressões Regulares (ER) para descrição de padrões de frases;
- Desenvolver, a partir de ERs, sistemática e automaticamente Processadores de Linguagens Regulares, que filtrem ou transformem textos;
- Utilizar o sistema de produção para filtragem de texto GAWK.

É importante referir que no decorrer deste relatório teve-se também em consideração que grande parte dos problemas com o software hoje em dia baseia-se na má comunicação entre os desenvolvedores e os seus empregadores tentou-se então que todos as imagens e explicações neste relatório fossem sucintas, diretas ao assunto e simples de perceber. Desta forma emula-se a estrutura de um relatório que poderia ser relativo a um trabalho no mundo de trabalho fora da redoma que é a Universidade e onde o nosso empregador, mesmo com poucos conhecimentos de processamento de linguagens seria, mesmo assim, capaz de ler, perceber e fazer uma crítica construtiva sobre a direcção do projeto. Isto possibilita evitar interpretações erradas do objetivo por parte dos programadores e também permite ao empregador compreender melhor se realmente os seus objectivos podem ser todos realizados da forma que idealizou ou se são demasiado optimistas.

Estrutura do Relatório

explicar como está organizado o documento, referindo os capítulos existentes em e a sua articulação explicando o conteúdo de cada um. No capítulo 2 faz-se uma análise detalhada do problema proposto

Organizamos então o relatório nos seguintes capítulos :

- **Capítulo 1 - Introdução** Aqui são explicados sucintamente os objetivos gerais que se pretende com o projecto. Não os os objetivos concretos relativos á tecnologia **ViaVerde**, mas sim as capacidades globais que é expectável que os autores, como desenvolvedores do projeto, melhorem/desenvolvam. Fala-se também um pouco nos cuidados *burucráticos* tidos no processo de escrita do relatório bem como a sua organização estrutura.
- **Capítulo 2 - Análise e Especificação** Começa-se por fazer uma descrição informal do problema a resolver, isto consiste na formulação do projecto já tocando nos pontos específicos que são relativos á tecnologia **ViaVerde**. O input usado é depois especificado assim como os requisitos mínimos propostos pelos docentes supervisores do projeto, assim como os requisitos extra que os autores desenvolvedores acharam relevante adicionar.

- **Capítulo 3 - Concepção/Desenho da Solução** De seguida é explicada a forma de iteração pelas linhas do ficheiro, sendo explicada a estrutura usada para guardar os dados necessários à medida que se itera pelo ficheiro e explicado também o raciocínio geral por detrás das soluções.
- **Capítulo 4 - Codificação e Testes** Agora é mostrado o output de cada programa pormenorizadamente e demonstrado o seu respetivo método de uso. Referem-se também um pouco as dificuldades encontradas
- **Capítulo 5 - Conclusão e Trabalho Futuro** Por fim faz uma breve análise de todo o projeto num panorama geral e avalia-se a solução obtida, mencionando também melhorias futuras que se poderiam vir a fazer caso o projeto não terminasse ainda.

Capítulo 2

Análise e Especificação

2.1 Descrição informal do problema

De forma a praticar os objetivos referidos no capítulo 1 realizou-se um projecto onde, de forma a operar sobre um ficheiro **XML** com informações relativas a transacções no contexto da tecnologia ViaVerde, se desenvolveram filtros de texto em **GAWK** que permitem a apresentação de informação pertinente ao utilizador de forma fácil, limpa e simples.

2.2 Especificação dos Requisitos

2.2.1 Dados

No ficheiro **XML** referido anteriormente temos um extracto onde é identificado o cliente ao qual se refere e, de seguida, todas as transacções do mesmo.

Mostra-se agora o exemplo de dados que adjetivam um extracto:

```
1 <EXTRACTO id="011114056/08/2015">
2 <MES_EMISSAO>Ago-2015</MES_EMISSAO>
3 <CLIENTE id="514714936">...</CLIENTE>
4 <IDENTIFICADOR id="28876820811">
5 <MATRICULA>00-LJ-11</MATRICULA>
6 <REF_PAGAMENTO>1234567</REF_PAGAMENTO>
```

O campo **CLIENTE** expande para as seguintes linhas:

```
1 <CLIENTE id="514714936">
2 <NIF>987653210</NIF>
3 <NOME>PEDRO MANUEL RANGEL SANTOS HENRIQUES</NOME>
4 <MORADA>RUA XXX</MORADA>
5 <LOCALIDADE>BRAGA</LOCALIDADE>
6 <CODIGO_POSTAL>4715-012 BRAGA</CODIGO_POSTAL>
7 </CLIENTE>
```

De seguida temos ainda várias transacções relativas ao extrato. Cada uma destas transacções pode ser de vários tipos distintos. No ficheiro utilizado surgem apenas dois tipos: **Portagens** e **Parques de estacionamento**. Mostra-se agora o exemplo de uma portagem:

```
1 <TRANSACCAO>
2 <DATA_ENTRADA>26-07-2015</DATA_ENTRADA>
3 <HORA_ENTRADA>11:33</HORA_ENTRADA>
4 <ENTRADA>Povoa N-S</ENTRADA>
5 <DATA_SAIDA>26-07-2015</DATA_SAIDA>
6 <HORA_SAIDA>11:42</HORA_SAIDA>
7 <SAIDA>Angeiras N-S</SAIDA>
8 <IMPORTANCIA>2,00</IMPORTANCIA>
9 <VALOR_DESCONTO>0,00</VALOR_DESCONTO>
10 <TAXA_IVA>23</TAXA_IVA>
11 <OPERADOR>I. de Portugal (N1)</OPERADOR>
12 <TIPO>Portagens</TIPO>
13 <DATA_DEBITO>05-08-2015</DATA_DEBITO>
14 <CARTAO>6749036</CARTAO>
15 </TRANSACCAO>
```

Note-se que é referido o local do prtico de sada e o de entrada (que iremos referir de agora para a frente como sada e entrada apenas). Vemos agora um exemplo de um parque de estacionamento:

```
1 <TRANSACCAO>
2 <DATA_ENTRADA>06-08-2015</DATA_ENTRADA>
3 <HORA_ENTRADA>15:57</HORA_ENTRADA>
4 <ENTRADA/>
5 <DATA_SAIDA>06-08-2015</DATA_SAIDA>
6 <HORA_SAIDA>17:08</HORA_SAIDA>
7 <SAIDA>PQ A Sa Carn.I</SAIDA>
8 <IMPORTANCIA>3,75</IMPORTANCIA>
9 <VALOR_DESCONTO>0,00</VALOR_DESCONTO>
10 <TAXA_IVA>23</TAXA_IVA>
11 <OPERADOR>ANA - Aeroportos de Portugal. SA (AP)</OPERADOR>
12 <TIPO>Parques de estacionamento</TIPO>
13 <DATA_DEBITO>14-08-2015</DATA_DEBITO>
14 <CARTAO>6749036</CARTAO>
15 </TRANSACCAO>
```

Aqui vemos que no temos o campo relativo  entrada. Sendo ento o campo de sada referente ao local do parque de estacionamento.

2.2.2 Pedidos

Certos requisitos mnimos foram propostos pelos docentes supervisores do projeto:

- A) Calcular o nmero de 'entradas' em cada dia do ms.
- B) A lista de locais de 'sada'.
- C) Calcular o total gasto no ms.
- D) Calcular o total gasto no ms apenas em 'parques'.

para alm destes requisitos mnimos, desenvolveram-se tambm alguns extras. Extendeu-se os quatro requisitos iniciais e adicionou-se ainda um outro.

- A) Calcular o nmero de 'entradas' em cada data distinta, calcular quantas em cada dia distinto, quantas em cada ms distinto e quantas em cada ano distinto.

- B)** Determinar lista de locais de 'saída' e, para cada local de saída distinto, quantas saídas no mesmo se realizaram.
- C)** Calcular o total gasto em cada data, em cada dia distinto, em cada mês distinto e em cada ano distinto.
- D)** Calcular o total gasto em cada data, em cada dia distinto, em cada mês distinto e em cada ano distinto, mas apenas em 'parques'.
- E)** Construir uma árvore de ficheiros **HTML**, ligados entre si através de links, que apresentam de forma apelativa toda a informação presente no ficheiro **XML**. Deve ainda indicar o tempo de duração de cada 'viagem', o tempo de estadia em cada 'parque', assim como o tempo total para ambos. As viagens devem incluir um link para o respetivo percurso no google maps e os parques um link para o respetivo local no google maps.

Capítulo 3

Concepção/desenho da Resolução

3.1 Estruturas de Dados

Todos os programas desenvolvidos neste projeto partilham uma estrutura semelhante. Inicialmente têm um bloco **BEGIN** onde são feitos prints iniciais de apoio ao utilizador e declaradas algumas variáveis importantes para todo o programa, como contadores e nomes de ficheiros.

Segue-se um exemplo de um bloco deste tipo usado no programa A:

```
#####
BEGIN{
print "\t -----"
print "\t/                                     \"
print "\t| ENTRADAS                | por: Daniel Malhadas |"
print "\t| -----                | & Alexandre Silva |"
print "\t|                |                |"
print "\t|-----|-----|"
print "\t|                | Determinar quantas |"
print "\t|    objectivo    | entradas em cada |"
print "\t|    -----    | dia, em cada mes e |"
print "\t|                | em cada ano.      |"
print "\t\\-----/"
print "\t/                                     \"

#Contador de Entradas. Inicializado a 0
entradas = 0;
}
#####
```

De seguida têm um bloco intermédio que será o usado para iterar pelo ficheiro de input. Este bloco consiste na seguinte estrutura: **\EXPRESSAO REGULAR\{Operar sobre a linha com a expressão regular procurada}**. Com este bloco exemplificado as instruções no interior das chavetas serão aplicadas para cada linha que contenha a expressão regular dentro dos 'backlashes'.

Mostra-se agora um exemplo deste bloco, também tirado do programa A:

```
#####
#Expressao regular que identifica uma entrada
/<DATA_ENTRADA>/{

#Contador de data
split ($0,data,">|<");
```

```
datas[data[3]]++;
```

```
#Contadores dia, mes e ano
```

```
split (data[3],dma,"-");
```

```
    dia[dma[1]]++;
```

```
    mes[dma[2]]++;
```

```
    ano[dma[3]]++;
```

```
    entradas++;
```

```
}
```

```
#####
```

Por fim temos um bloco **END** onde fazemos os prints finais de apresentação de resultados ao utilizador. De novo mostramos um exemplo tirado do programa A:

```
#####
```

```
END{
```

```
    print "\t|-----"
```

```
    print "\t| Entradas em cada data"
```

```
    print "\t|-----"
```

```
    for (i in datas)
```

```
        print "\t| Na Data " i ": " datas[i] " entradas";
```

```
    print "\t|-----"
```

```
    print "\t| Entradas em cada dia"
```

```
    print "\t|-----"
```

```
    for (i in dia)
```

```
        print "\t| No dia " i ": " dia[i] " entradas";
```

```
    print "\t|-----"
```

```
    print "\t| Entradas em cada mes"
```

```
    print "\t|-----"
```

```
    for (i in mes) {
```

```
        if(length(i)==0) print "\t| No mes null: " mes[i] " entradas";
```

```
        else          print "\t| No mes " i ": " mes[i] " entradas";
```

```
    }
```

```
    print "\t|-----"
```

```
    print "\t| Entradas em cada ano"
```

```
    print "\t|-----"
```

```
    for (i in ano) {
```

```
        if(length(i)==0) print "\t| No ano null: " ano[i] " entradas";
```

```
        else          print "\t| No ano " i ": " ano[i] " entradas";
```

```
    }
```

```
    print "\t|-----"
```

```
    print "\t| Total Entradas: " entradas
```

```
    print "\t\\_-----/"
```

```
}
```

```
#####
```

3.2 Algoritmos

Podemos observar na secção anterior que no interior do bloco intermédio do programa A se utiliza a função `split` para operar sobre a linha. Este método de operar sobre a linha repete-se da mesma forma nos restantes programas, segue-se então uma explicação do uso da função `split`:

```
Exemplo do uso de split em gawk:
-----
{split($0, a, "<|>")}
  ^^  ^  ^^^^^
  |   |   |
  |   |   | delimitador, pode ser uma expressao regular
  |   |   |
  |   |   | array para guardar string dividida
  |   |   |
  |   |   | string para fazer split, $0 e a linha toda
-----
```

Figura 3.1: Funcionamento da Função `split` em GAWK

Esta função irá guardar o resultado no segundo argumento e, este resultado será um array de strings com a string resultado (primeiro argumento) dividida pelo delimitador que é o terceiro argumento.

Nos outros programas surge por vezes também a necessidade de operar na linha abaixo da atual. Para isso temos mais uma função que podemos usar chamada **getline** que coloca a linha diretamente abaixo na variável `$0`. Em certas alturas esta função é o ideal, mas noutras alturas em que a linha está muito em baixo em relação à nossa posição atual ou então quando não sabemos quantas linhas ainda faltam até à linha que pretendemos encontrar, usamos a função **index** em conjunção com **getline**. Por exemplo:

```
#####
for(getline; (i=index($0,"DATA_DEBITO"))==0; getline);
#####
```

Com este ciclo vamos sempre percorrendo linha a linha até a linha atual (`$0`) tiver a string pretendida (`DATA_DEBITO`). Em situações em que não temos a certeza se uma linha realmente existe podemos antes fazer algo como:

```
#####
for(getline; ((i=index($0,"Parques de estacionamento"))==0) &&
(index($0,"/TRANSACCAO>")==0);
getline);
#####
```

Neste caso procuramos a linha da mesma forma que no caso anterior, no entanto, paramos caso cheguemos a uma linha com `/TRANSACCAO`, já que esta é a última linha da transacção então se a variável `i` for zero sabemos que a linha com `Parque de estacionamento` não existe nesta transacção.

Capítulo 4

Codificação e Testes

4.1 Alternativas, Decisões e Problemas de Implementação

Com o intuito de completar o programa extra E foi necessário estudar um pouco o formato de um ficheiro **HTML** com o qual não estávamos familiarizados a cem por cento. Inicialmente notamos que a informação era demasiada e ficava demasiado confusa ao colocar tudo numa só página, daí optou-se por dividi-la em diferentes secções que seriam apresentadas em diferentes páginas acedidas através de links. Isto significa que um input com mais dados pode gerar mais ficheiros **HTML**, no entanto, já que a intenção é apresentar a informação ao utilizador de forma simples, fácil e limpa, achou-se que os benefícios justificam o facto de o número de ficheiros criados variar com o input.

Mesmo dividindo em várias páginas, certos conjuntos de informações geraram páginas muito extensas, por exemplo uma página com todas as transacções de um extracto pode tornar-se bastante extensa. A nossa solução foi separar os dados referentes a cada transacção com texto com cor vermelha e um tamanho de letra maior que o restante texto chamando assim a atenção de que se iniciará uma nova transacção a partir daquela linha. Este texto atribui também um número sequencial a cada transacção a começar no um para facilmente se encontrar determinada transacção relevante. Outra opção poderia ser organizar estes dados numa tabela ou então dividir as transacções por várias páginas e ter um link que permitiria avançar para a página seguinte.

4.2 Testes realizados e Resultados

Mostra-se agora imagens que demonstram o comando usado para cada programa e o output correspondente.

Programa A:

```
xubuntu@xubuntu-VirtualBox:~/Desktop/PL/TP1a$ gawk -f TP1a_ViaVerde_exe.gawk va
averde.xml
```

ENTRADAS	por: Daniel Malhadas & Alexandre Silva
objectivo	Determinar quantas entradas em cada dia, em cada mes e em cada ano.

```

Entradas em cada data
Na Data 30-07-2015: 3 entradas
Na Data 18-08-2015: 2 entradas
Na Data 29-07-2015: 2 entradas
Na Data 11-08-2015: 2 entradas
Na Data 10-08-2015: 7 entradas
Na Data 26-07-2015: 3 entradas
Na Data 17-08-2015: 4 entradas
Na Data 06-08-2015: 4 entradas
Na Data null: 4 entradas
Na Data 13-08-2015: 5 entradas
Na Data 21-08-2015: 4 entradas
Na Data 31-07-2015: 1 entradas

Entradas em cada dia
No dia 17: 4 entradas
No dia 26: 3 entradas
No dia 18: 2 entradas
No dia 29: 2 entradas
No dia 10: 7 entradas
No dia 11: 2 entradas
No dia 21: 4 entradas
No dia 30: 3 entradas
No dia 13: 5 entradas
No dia null: 4 entradas
No dia 31: 1 entradas
No dia 06: 4 entradas

Entradas em cada mes
No mes null: 4 entradas
No mes 08: 28 entradas
No mes 07: 9 entradas

Entradas em cada ano
No ano null: 4 entradas
No ano 2015: 37 entradas

Total Entradas: 41

```

4.1 Funcionamento do programa A

Programa B:

```
xubuntu@xubuntu-VirtualBox:~/Desktop/PL/TP1a$ gawk -f TP1a_ViaVerde_exec.gawk via
verde.xml

| SAIDAS | por: Daniel Malhadas |
| -----| & Alexandre Silva |
|         |                       |
|-----|
| objectivo | Determinar que saidas |
| -----| se realizaram, quantas |
|         | de cada e quantas no |
|         | total.                |
|         |                       |
|-----|
| Locais de saidas |
|-----|
| Saidas em Braga Sul: 2 entradas |
| Saidas em Maia PV: 1 entradas |
| Saidas em Neiva S-N: 1 entradas |
| Saidas em Angeiras N-S: 5 entradas |
| Saidas em Freixieiro: 2 entradas |
| Saidas em Valongo: 1 entradas |
| Saidas em PQ A Sa Carn.I: 1 entradas |
| Saidas em Ermesinde PV: 1 entradas |
| Saidas em Aeroporto: 1 entradas |
| Saidas em Maia II: 1 entradas |
| Saidas em EN107: 1 entradas |
| Saidas em EN 205 PV: 6 entradas |
| Saidas em Neiva N-S: 1 entradas |
| Saidas em Lipor: 1 entradas |
| Saidas em Ponte Pedra: 3 entradas |
| Saidas em PQ Av. Central: 1 entradas |
| Saidas em Custoias: 1 entradas |
| Saidas em Povoia S-N: 5 entradas |
| Saidas em Ferreiros: 6 entradas |
|-----|
| Total Saidas: 41 |
|-----|

xubuntu@xubuntu-VirtualBox:~/Desktop/PL/TP1a$
```

4.2 Funcionamento do programa B

Programa C:

```
xubuntu@xubuntu-VirtualBox:~/Desktop/PL/TP1a$ gawk -f TP1a_ViaVerde_exec.gawk via
verde.xml

| Total Gasto | por: Daniel Malhadas |
| -----| & Alexandre Silva |
|         |                       |
|-----|
| objectivo | Determinar total gasto |
| -----| por dia/mes/ano. |
|         |                       |
|-----|
| Pago em cada data |
|-----|
| Na Data 07-08-2015 pago 15.65 |
| Na Data 14-08-2015 pago 13.6 |
| Na Data 05-08-2015 pago 4.65 |
| Na Data 31-08-2015 pago 19.7 |
| Na Data 21-08-2015 pago 23.8 |
|-----|
| Pago em cada dia |
|-----|
| No dia 21 pago 23.8 |
| No dia 31 pago 19.7 |
| No dia 14 pago 13.6 |
| No dia 05 pago 4.65 |
| No dia 07 pago 15.65 |
|-----|
| Pago em cada mes |
|-----|
| No ano 2015 pago 77.4 |
|-----|
| Total Pago : 77.4 |
| Total Desconto : 0 |
| Total Pagamentos: 41 |
|-----|

xubuntu@xubuntu-VirtualBox:~/Desktop/PL/TP1a$
```

4.3 Funcionamento do programa C

Programa D:

```
xubuntu@xubuntu-VirtualBox:~/Desktop/PL/TP1a$ gawk -f TP1a_ViaVerde_exed.gawk via
verde.xml

/
Total Gasto          por: Daniel Malhadas
-----            & Alexandre Silva
/
/
objectivo            Determinar total gasto
-----            por dia/mes/ano apenas
/                  em parques.
\
\
/
Pago em cada data (apenas em Parques)
-----
Na Data 07-08-2015 pago 2.6
Na Data 14-08-2015 pago 3.75
-----
Pago em cada dia (apenas em Parques)
-----
No dia 14 pago 3.75
No dia 07 pago 2.6
-----
Pago em cada mes (apenas em Parques)
-----
No mes 08 pago 6.35
-----
Pago em cada ano (apenas em Parques)
-----
No ano 2015 pago 6.35
-----
Total Pago apenas em Parques      : 6.35
Total Desconto apenas em Parques  : 0
Total Pagamentos apenas em Parques: 2
/
xubuntu@xubuntu-VirtualBox:~/Desktop/PL/TP1a$
```

4.4 Funcionamento do programa D

Programa E:

```
xubuntu@xubuntu-VirtualBox:~/Desktop/PL/TP1a$ gawk -f TP1a_ViaVerde_EXTRA.gawk v
iaverde.xml

/
Registos ViaVerde    por: Daniel Malhadas
-----            & Alexandre Silva
/
/
objectivo            Constrói ficheiros
-----            html que organizam de
/                  uma maneira facil de
\                  ler toda a informacao
\                  em viaverde.xml.
\
/
Ficheiros Construidos com Sucesso
-----
/
xubuntu@xubuntu-VirtualBox:~/Desktop/PL/TP1a$
```

4.5 Funcionamento do programa E

Agora que se gerou os ficheiros do programa E podemos observá-los. O ficheiro base que se deve abrir primeiro é o ficheiro **index.html**. Aqui podemos escolher a funcionalidade que pretendemos, nesta versão apenas se pode consultar extractos.



4.5 ficheiro index.html

Ao carregar no link podemos observar todos os extractos. Como o ficheiro de input apenas contém um extracto então apenas vemos um na seguinte imagem, no entanto, se o input tivesse vários, eles apareceriam aqui.



Lista de Extractos

Extracto 1:

ID do Extracto:

"011114056/08/2015"

Mes de Emissao:

Ago-2015

ID do Cliente:

"514714936"

NIF do Cliente:

987653210

Nome do Cliente:

PEDRO MANUEL RANGEL SANTOS HENRIQUES

Morada:

RUA XXX

Localidade:

BRAGA

Codigo Postal:

4715-012 BRAGA

Identificador:

"28876820811"

Matricula:

00-LJ-11

Referencia de Pagamento:

1234567

[Consultar transaccoes deste extracto](#)


[Consultar viagens realizadas neste extracto](#)

[Consultar parques utilizados neste extracto](#)

4.6 ficheiro de extractos

Temos agora 3 links para cada extracto. O primeiro mostra todas as transacções realizadas nesse extracto, o segundo todas as viagens (através das transacções que resultam de portagens) e o último os parques de estacionamento que visitou.

Olhamos agora para o link das transacções:



VIA VERDE

Lista de Transaccoes do Extracto 1

Transacao 1:

Data de Entrada:
26-07-2015

Hora de Entrada:
11:33

Local de Entrada:
Povoia N-S

Data de Saída:

4.7 ficheiro de transacções

Aqui temos a lista de todas as transacções, sendo que as restantes se encontram por baixo, separadas pelo texto vermelho indicando o número de cada uma como referido na subsecção anterior. Olhamos agora para o link das viagens:



VIA VERDE

Lista de Viagens do Extracto 1

Viagem 1:

Data de Entrada:
26-07-2015

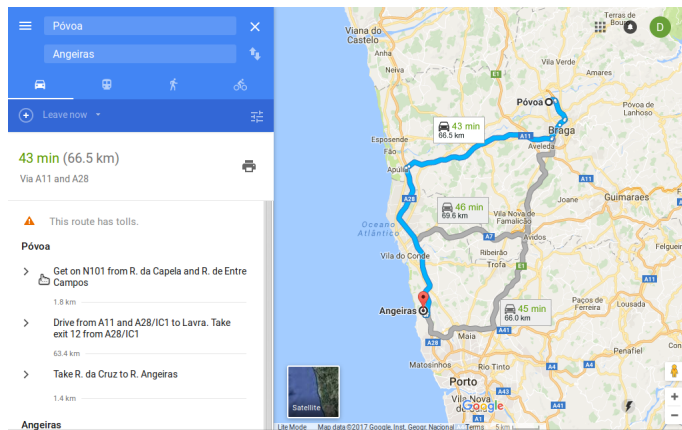
Data de Saída:
26-07-2015

Tempo de Viagem:
9 minutos

Percurso Realizado (clique para ver no google maps):
[De Povoia N-S ate Angeiras N-S](#)


4.8 ficheiro de viagens

Aqui temos as viagens separadas também por texto vermelho indicando o número de cada uma. É ainda indicado o tempo de duração da mesma, sendo que no final da página temos o tempo total de todas as viagens. Clicando no percurso da viagem podemos observar o mesmo no google maps. Como exemplo temos o que aparece ao clicar no percurso da primeira viagem:



4.9 percurso da viagem 1 no google maps

Olhando agora por último para o link dos parques vemos os parques também divididos pelo texto vermelho. Temos também o tempo de estadia em cada parque e o tempo total passado em parques no fim da página. Ao clicar no local do parque este pode ser observado no google maps.



Lista de Parques do Extracto 1

Parque de Estacionamento 1:

Data de Entrada:
30-07-2015

Data de Saída:
30-07-2015

Tempo de Estadia:
100 minutos

Local do Parque (clique para ver no google maps):
[PQ Av. Central](#)

4.10 ficheiro de parques

Capítulo 5

Conclusão e Trabalho Futuro

Terminado o projeto entende-se que o mesmo cumpre com sucesso tudo o que era pretendido e adiciona ainda funcionalidades extra que são realmente relevantes para o contexto do problema. Em particular o programa E foi pensado com especial cuidado para que fosse realizado algo que teria algum propósito real e não só académico já que, poderia ser do interesse de alguém organizar a informação guardada num ficheiro **XML** de uma forma mais apelativa e fácil de entender para um humano. O facto de se ter escolhido uma interface com recurso a ficheiros **HTML** ao invés do terminal permitiria que estes fossem usados para, por exemplo, serem adicionados a um site onde o cliente poderia consultar os seus extractos.

Embora se tenha conseguido completar tudo o que se pretendeu, poder-se-ia ainda melhorar já que nunca se pode dar nenhum programa por realmente terminado ou perfeito. Poder-se-ia, por exemplo, organizar as transacções/viagens/parques por vários ficheiros, sendo que cada um apresentaria por exemplo dez itens de cada vez, dando depois a opção ao utilizador de navegar para um outro ficheiro com os próximos dez. Embora isto pudesse tornar as coisas mais organizadas o número de ficheiros ia ser muito maior por isso também teria as suas desvantagens.

Por fim, conclui-se que o projeto final ficou do nosso agrado tendo permitido que, com a prática, desenvolvessemos capacidades em GAWK que não tínhamos anteriormente. Desenvolvemos também uma melhor percepção do que realmente são expressões regulares e achamos estar agora mais aptos para perceber melhor o decorrer da matéria na unidade curricular em questão.

Apêndice A

Código do Programa

Programa A

```
BEGIN{
    print "\t -----"
    print "\t/"
    print "\t| ENTRADAS          | por: Daniel Malhadas |"
    print "\t| -----          | & Alexandre Silva |"
    print "\t|                      |                      |"
    print "\t|-----|-----|"
    print "\t|                      | Determinar quantas |"
    print "\t|    objetivo          | entradas em cada  |"
    print "\t|    -----          | dia, em cada mes e |"
    print "\t|                      | em cada ano.      |"
    print "\t\\-----/"
    print "\t/"

    #Contador de Entradas. Inicializado a 0
    entradas = 0;
}

#Expressao regular que identifica uma entrada
/<DATA_ENTRADA>/{

    #Contador de data
    split ($0,data,">|<");
    datas[data[3]]++;

    #Contadores dia, mes e ano
    split (data[3],dma,"-");
    dia[dma[1]]++;
    mes[dma[2]]++;
    ano[dma[3]]++;

    entradas++;
}

END{
    print "\t|-----"
```

```

print "\t| Entradas em cada data"
print "\t|-----"
for (i in datas)
    print "\t| Na Data " i ": " datas[i] " entradas";

print "\t|-----"
print "\t| Entradas em cada dia"
print "\t|-----"
for (i in dia)
    print "\t| No dia " i ": " dia[i] " entradas";

print "\t|-----"
print "\t| Entradas em cada mes"
print "\t|-----"
for (i in mes) {
    if(length(i)==0) print "\t| No mes null: " mes[i] " entradas";
    else             print "\t| No mes " i ": " mes[i] " entradas";
}

print "\t|-----"
print "\t| Entradas em cada ano"
print "\t|-----"
for (i in ano) {
    if(length(i)==0) print "\t| No ano null: " ano[i] " entradas";
    else             print "\t| No ano " i ": " ano[i] " entradas";
}

print "\t|-----"
print "\t| Total Entradas: " entradas
print "\t\\_-----/"
}

```

Programa B

```

BEGIN{
    print "\t -----"
    print "\t/                                     \\"
    print "\t| SAIDAS                               | por: Daniel Malhadas |"
    print "\t| -----                               | & Alexandre Silva |"
    print "\t|                                     |                       |"
    print "\t|-----|-----|"
    print "\t|                                     | Determinar que saidas |"
    print "\t|    objetivo                               | se realizaram, quantas |"
    print "\t|    -----                               | de cada e quantas no  |"
    print "\t|                                     | total.                 |"
    print "\t\\_-----/"
    print "\t/                                     \\"

    #Contador de saidas. Inicializado a 0
    saidas = 0;
}

#Expressao regular que identifica uma saida
/<SAIDA>/{
    #Contador de data

```

```

split ($0,local,">|<");
locais[local[3]]++;

saidas++;
}

END{
    print "\t|-----"
    print "\t| Locais de saidas"
    print "\t|-----"
    for (i in locais)
        print "\t| Saidas em " i ": " locais[i] " entradas";
    print "\t|-----"
    print "\t| Total Saidas: " saidas
    print "\t\\_-----/"
}

```

Programa C

```
BEGIN{
    print "\t -----"
    print "\t/"
    print "\t| Total Gasto          | por: Daniel Malhadas |"
    print "\t| -----                | & Alexandre Silva |"
    print "\t|                          |                          |"
    print "\t|-----|-----|"
    print "\t|                          |                          |"
    print "\t|      objetivo          | Determinar total gasto |"
    print "\t|      -----          | por dia/mes/ano.      |"
    print "\t|                          |                          |"
    print "\t\\-----/"
    print "\t/"

    #Contador de total gasto. Inicializado a 0
    total_gasto = 0;

    #Contador de total desconto. Inicializado a 0
    total_desconto = 0;

    #Total pagamentos. Inicializado a 0
    pagamentos = 0;

}

#Expressao regular que identifica uma importancia
/<IMPORTANCIA>/{

    split ($0,importancia,">|<");
    sub(",",".",importancia[3]);
    total_gasto += importancia[3];
    gasto_agora = importancia[3];

    getline;
    split ($0,desconto,">|<");
    sub(",",".",desconto[3]);
    total_desconto += desconto[3];
    gasto_agora -= desconto[3];

    for(getline; (i=index($0,"DATA_DEBITO"))==0; getline);

    split($0,data,">|<");
    gasto_data[data[3]]+=gasto_agora;

    split(data[3],dma,"-");
    dia[dma[1]]+=gasto_agora;
    mes[dma[2]]+=gasto_agora;
    ano[dma[3]]+=gasto_agora;

    pagamentos++;
}
```



```

END{
    print "\t|-----"
    print "\t| Pago em cada data"
    print "\t|-----"
    for (i in gasto_data)
        print "\t| Na Data " i " pago " gasto_data[i];

    print "\t|-----"
    print "\t| Pago em cada dia"
    print "\t|-----"
    for (i in dia)
        print "\t| No dia " i " pago " dia[i];

    print "\t|-----"
    print "\t| Pago em cada mes"
    print "\t|-----"
    for (i in mes) {
        print "\t| No mes " i " pago " mes[i];
    }

    print "\t|-----"
    print "\t| Pago em cada ano"
    print "\t|-----"
    for (i in ano) {
        print "\t| No ano " i " pago " ano[i];
    }

    print "\t|-----"
    print "\t| Total Pago      : " total_gasto
    print "\t| Total Desconto   : " total_desconto
    print "\t| Total Pagamentos: " pagamentos
    print "\t\\_____/"
}

```

Programa D

```

BEGIN{
    print "\t _____"
    print "\t/                                     \\"
    print "\t| Total Gasto          | por: Daniel Malhadas |"
    print "\t| -----              | & Alexandre Silva |"
    print "\t|                      |                      |"
    print "\t| -----              | -----"
    print "\t|                      | Determinar total gasto |"
    print "\t|      objetivo        | por dia/mes/ano apenas |"
    print "\t| -----              | em parques.         |"
    print "\t|                      |                      |"
    print "\t\\_____/"
    print "\t/                                     \\"

    #Contador de total gasto. Inicializado a 0
    total_gasto = 0;

    #Contador de total desconto. Inicializado a 0
    total_desconto = 0;
}

```

```

#Total pagamentos. Inicializado a 0
pagamentos = 0;

}

#Expressao regular que identifica uma importancia
/<IMPORTANCIA>/{

    split ($0,importancia,">|<");

    getline;
    split ($0,desconto,">|<");

    for(getline; ((i=index($0,"Parques de estacionamento"))==0) &&
        (index($0,"/TRANSACCAO>")==0);
        getline);

    if (i!=0){
        sub(",",".",importancia[3]);
        sub(",",".",desconto[3]);
        total_gasto += importancia[3];
        gasto_agora = importancia[3];
        total_desconto += desconto[3];
        gasto_agora -= desconto[3];

        getline;
        split($0,data,">|<");
        gasto_data[data[3]]+=gasto_agora;

        split(data[3],dma,"-");
        dia[dma[1]]+=gasto_agora;
        mes[dma[2]]+=gasto_agora;
        ano[dma[3]]+=gasto_agora;

        pagamentos++;
    }
}

END{
    print "\t|-----"
    print "\t| Pago em cada data (apenas em Parques)"
    print "\t|-----"
    for (i in gasto_data)
        print "\t| Na Data " i " pago " gasto_data[i];

    print "\t|-----"
    print "\t| Pago em cada dia (apenas em Parques)"
    print "\t|-----"
    for (i in dia)
        print "\t| No dia " i " pago " dia[i];

    print "\t|-----"

```

```

print "\t| Pago em cada mes (apenas em Parques)"
print "\t|-----"
for (i in mes) {
    print "\t| No mes " i " pago " mes[i];
}

print "\t|-----"
print "\t| Pago em cada ano (apenas em Parques)"
print "\t|-----"
for (i in ano) {
    print "\t| No ano " i " pago " ano[i];
}

print "\t|-----"
print "\t| Total Pago apenas em Parques      : " total_gasto
print "\t| Total Desconto apenas em Parques   : " total_desconto
print "\t| Total Pagamentos apenas em Parques: " pagamentos
print "\t\\_____/"
}

```

Programa E

```

BEGIN{
    print "\t _____"
    print "\t/                                     \\"
    print "\t| Registos ViaVerde      |   por: Daniel Malhadas |"
    print "\t| -----                |       & Alexandre Silva |"
    print "\t|                          |                          |"
    print "\t|-----|-----|"
    print "\t|                          | Constroi ficheiros   |"
    print "\t|      objectivo          | html que organizam de |"
    print "\t|      -----            | uma maneira facil de |"
    print "\t|                          | ler toda a informacao |"
    print "\t|                          | em viaverde.xml.    |"
    print "\t\\_____/"
    print "\t/                                     \\"

    #Constroi a base/home page dos registos ViaVerde
    base      = "index.html"      #ficheiro base
    extractos  = "extractos.html"  #ficheiro com os extractos
    transaccoes = "transaccoes.html" #ficheiro com transaccoes
    viagens    = "viagens.html"    #ficheiro com viagens
    parques    = "parques.html"    #ficheiro com parques

    print "<!DOCTYPE html>" > base;
    print "<html>" > base;
    print "<body>" > base;
    print "<body style=\"background-color:lightgreen\">" > base;
    print "<img src=\"viaverde.jpg\" width=\"200\" height=\"200\">" > base;
    print "<h1 style=\"color:darkgreen;font-size:300%;text-align:center\">Registos ViaVerde</h1>" > base;
    print "<p style=\"color:red;font-size:160%;text-align:center\">Funcionalidades:</p>" > base;
    print "<a href=\"\" extractos \"\">Consultar extractos</a>" > base;
    print "</body>" > base;
    print "</html>" > base;
}

```

```

#Ficheiro base terminado

}

#Expressao regular que identifica uma importancia
/<EXTRACTO/{

#Identificar extracto
split ($0,idaux,">|<");
split (idaux[2],extracto_id,"=");
id = extracto_id[2];

getline;
split ($0,data,">|<");
data_emissao[id] = data[3];

getline;
split ($0,id_clienteaux,">|<");
split (id_clienteaux[2],id_cliente,"=");
clientes_id[id] = id_cliente[2];

getline;
split ($0,nif,">|<");
nifs[id] = nif[3];

getline;
split ($0,nome_cliente,">|<");
nome_clientes[id] = nome_cliente[3];

getline;
split ($0,morada,">|<");
moradas[id] = morada[3];

getline;
split ($0,localidade,">|<");
localidades[id] = localidade[3];

getline;
split ($0,cod_postal,">|<");
codigos_postais[id] = cod_postal[3];

getline;
getline;
split ($0,identificador_aux,">|<");
split (identificador_aux[2],identificador,"=");
identificadores[id] = identificador[2];

getline;
split ($0,mat,">|<");
matriculas[id] = mat[3];

getline;
split ($0,ref,">|<");
referencia_pagamento[id] = ref[3];

```

```

#Identificar transaccoes
t=0;
while(1){
    for(getline; (i=index($0,"<TRANSACCAO>"))==0 &&
        (j=index($0,"</EXTRACTO>"))==0 ;getline);

    if (i==0) break;

    t++;

    getline;
    split ($0,t_data_e,">|<");
    t_datas_e[id,t] = t_data_e[3];

    getline;
    split ($0,horas_e,">|<");
    t_horas_entrada[id,t] = horas_e[3];

    getline;
    split ($0,e,">|<");
    t_entradas[id,t] = e[3];

    getline;
    split ($0,data_s,">|<");
    t_datas_s[id,t] = data_s[3];

    getline;
    split ($0,horas_s,">|<");
    t_horas_saida[id,t] = horas_s[3];

    getline;
    split ($0,s,">|<");
    t_saidas[id,t] = s[3];

    getline;
    split ($0,imp,">|<");
    sub(",",".",imp[3]);
    t_importancias[id,t] = imp[3];

    getline;
    split ($0,desc,">|<");
    sub(",",".",desc[3]);
    t_descontos[id,t] = desc[3];

    getline;
    split ($0,iva,">|<");
    t_ivas[id,t] = iva[3];

    getline;
    split ($0,o,">|<");
    t_operadores[id,t] = o[3];

    getline;

```

```

split ($0, tipo, ">|<");
t_tipos[id,t] = tipo[3];

getline;
split ($0, deb, ">|<");
t_dadas_debito[id,t] = deb[3];

getline;
split ($0, cart, ">|<");
t_cartao[id,t] = cart[3];
}

n_transaccoes[id] = t;

}

END{
    n = 0;

    #Construir ficheiro de extractos
    print "<!DOCTYPE html>" > extractos;
    print "<html>" > extractos;
    print "<body>" > extractos;
    print "<body style=\"background-color:lightgreen\">" > extractos;
    print "<img src=\"viaverde.jpg\" width=\"200\" height=\"200\">" > extractos;
    print "<h1 style=\"font-size:300%\">Lista de Extractos</h1>" > extractos;

    for (i in data_emissao){
        n++;
        print "" > (n transaccoes);
        print "<p style=\"color:red;font-size:160%;\">Extracto " n " :</p>" > extractos;
        print "<p><b>ID do Extracto:</b></p>" > extractos;
        print "<p>i " </p>" > extractos;
        print "<p><b>Mes de Emissao:</b></p>" > extractos;
        print "<p>data_emissao[i] " </p>" > extractos;
        print "<p><b>ID do Cliente:</b></p>" > extractos;
        print "<p>clientes_id[i] " </p>" > extractos;
        print "<p><b>NIF do Cliente:</b></p>" > extractos;
        print "<p>nifs[i] " </p>" > extractos;
        print "<p><b>Nome do Cliente:</b></p>" > extractos;
        print "<p>nome_clientes[i] " </p>" > extractos;
        print "<p><b>Morada:</b></p>" > extractos;
        print "<p>moradas[i] " </p>" > extractos;
        print "<p><b>Localidade:</b></p>" > extractos;
        print "<p>localidades[i] " </p>" > extractos;
        print "<p><b>Codigo Postal:</b></p>" > extractos;
        print "<p>codigos_postais[i] " </p>" > extractos;
        print "<p><b>Identificador:</b></p>" > extractos;
        print "<p>identificadores[i] " </p>" > extractos;
        print "<p><b>Matricula:</b></p>" > extractos;
        print "<p>matriculas[i] " </p>" > extractos;
        print "<p><b>Referencia de Pagamento:</b></p>" > extractos;
        print "<p>referencia_pagamento[i] " </p>" > extractos;
    }
}

```

```

print "<a href=\"\" (n transaccoes) \"\">Consultar transaccoes deste extracto</a>" > extractos;
print "<p></p>" > extractos;
print "<a href=\"\" (n viagens) \"\">Consultar viagens realizadas neste extracto</a>" > extractos;
print "<p></p>" > extractos;
print "<a href=\"\" (n parques) \"\">Consultar parques utilizados neste extracto</a>" > extractos;

#Construir ficheiro de transaccoes para este extracto
print "<!DOCTYPE html>" > (n transaccoes);
print "<html>" > (n transaccoes);
print "<body>" > (n transaccoes);
print "<body style=\"background-color:lightgreen\">" > (n transaccoes);
print "<img src=\"viaverde.jpg\" width=\"200\" height=\"200\">" > (n transaccoes);
print "<h1 style=\"font-size:300%\">Lista de Transaccoes do Extracto "n"</h1>" > (n transaccoes);

for (j=1;j<=n_transaccoes[id];j++){
    print "" > (n transaccoes);
    print "<p style=\"color:red;font-size:160%;\">Transacciao " j "</p>" > (n transaccoes);
    print "<p><b>Data de Entrada:</b></p>" > (n transaccoes);
    print "<p>" t_datas_e[i,j] "</p>" > (n transaccoes);
    print "<p><b>Hora de Entrada:</b></p>" > (n transaccoes);
    print "<p>" t_horas_entrada[i,j] "</p>" > (n transaccoes);
    print "<p><b>Local de Entrada:</b></p>" > (n transaccoes);
    print "<p>" t_entradas[i,j] "</p>" > (n transaccoes);
    print "<p><b>Data de Saida:</b></p>" > (n transaccoes);
    print "<p>" t_datas_s[i,j] "</p>" > (n transaccoes);
    print "<p><b>Hora de Saida:</b></p>" > (n transaccoes);
    print "<p>" t_horas_saida[i,j] "</p>" > (n transaccoes);
    print "<p><b>Local de Saida:</b></p>" > (n transaccoes);
    print "<p>" t_saidas[i,j] "</p>" > (n transaccoes);
    print "<p><b>Importancia:</b></p>" > (n transaccoes);
    print "<p>" t_importancias[i,j] "</p>" > (n transaccoes);
    print "<p><b>Desconto:</b></p>" > (n transaccoes);
    print "<p>" t_descontos[i,j] "</p>" > (n transaccoes);
    print "<p><b>IVA:</b></p>" > (n transaccoes);
    print "<p>" t_ivas[i,j] "</p>" > (n transaccoes);
    print "<p><b>Operador:</b></p>" > (n transaccoes);
    print "<p>" t_operadores[i,j] "</p>" > (n transaccoes);
    print "<p><b>Tipo de Transacciao:</b></p>" > (n transaccoes);
    print "<p>" t_tipos[i,j] "</p>" > (n transaccoes);
    print "<p><b>Data de Debito:</b></p>" > (n transaccoes);
    print "<p>" t_datas_debito[i,j] "</p>" > (n transaccoes);
    print "<p><b>Cartao Usado:</b></p>" > (n transaccoes);
    print "<p>" t_cartao[i,j] "</p>" > (n transaccoes);
}
print "</body>" > (n transaccoes);
print "</html>" > (n transaccoes);
#Ficheiro de transaccoes deste extracto terminado

#Construir ficheiro de viagens para este extracto
print "<!DOCTYPE html>" > (n viagens);
print "<html>" > (n viagens);
print "<body>" > (n viagens);
print "<body style=\"background-color:lightgreen\">" > (n viagens);
print "<img src=\"viaverde.jpg\" width=\"200\" height=\"200\">" > (n viagens);

```

```

print "<h1 style=\"font-size:300%\">Lista de Viagens do Extracto "n"</h1>" > (n viagens);
tempo_total = 0;
jj=0;
for (j=1;j<=n_transaccoes[id];j++){
    if (t_tipos[i,j]!="Portagens") continue;
    jj++;
    print "<p style=\"color:red;font-size:160%;\">Viagem " jj " :</p>" > (n viagens);
    print "<p><b>Data de Entrada:</b></p>" > (n viagens);
    print "<p>" t_datas_e[i,j] "</p>" > (n viagens);
    print "<p><b>Data de Saida:</b></p>" > (n viagens);
    print "<p>" t_datas_s[i,j] "</p>" > (n viagens);
    if (t_datas_e[i,j]!="null" && t_datas_s[i,j]!="null"){
        print "<p><b>Tempo de Viagem:</b></p>" > (n viagens);
        split(t_datas_e[i,j],d,"-");
        split(t_horas_entrada[i,j],h,":");
        de = mktime(d[3] " " d[2] " " d[1] " " h[1] " " h[2] " 00");
        split(t_datas_s[i,j],d,"-");
        split(t_horas_saida[i,j],h,":");
        ds = mktime(d[3] " " d[2] " " d[1] " " h[1] " " h[2] " 00");
        tempo_total+=((ds - de)/60);
        print "<p>((ds - de)/60) minutos</p>" > (n viagens);
    }
    print "<p><b>Percurso Realizado (clique para ver no google maps):</b></p>" > (n viagens);
    print "<a href=\"https://www.google.pt/maps/dir/\"t_entradas[i,j]\"/\"t_saidas[i,j]\"/\"/>De " t_entr
}
print "<p><p><p style=\"color:red;font-size:160%;\">Tempo Total de Viagem \"tempo_total\" minutos</p>";
print "</body>" > (n viagens);
print "</html>" > (n viagens);
#Ficheiro de viagens deste extracto terminado

#Construir ficheiro de parques para este extracto
print "<!DOCTYPE html>" > (n parques);
print "<html>" > (n parques);
print "<body>" > (n parques);
print "<body style=\"background-color:lightgreen\">" > (n parques);
print "<img src=\"viaverde.jpg\" width=\"200\" height=\"200\">" > (n parques);
print "<h1 style=\"font-size:300%\">Lista de Parques do Extracto "n"</h1>" > (n parques);
tempo_total = 0;
jj=0;
for (j=1;j<=n_transaccoes[id];j++){
    if (t_tipos[i,j]!="Parques de estacionamento") continue;
    jj++;
    print "<p style=\"color:red;font-size:160%;\">Parque de Estacionamento " jj " :</p>" > (n parques);
    print "<p><b>Data de Entrada:</b></p>" > (n parques);
    print "<p>" t_datas_e[i,j] "</p>" > (n parques);
    print "<p><b>Data de Saida:</b></p>" > (n parques);
    print "<p>" t_datas_s[i,j] "</p>" > (n parques);
    if (t_datas_e[i,j]!="null" && t_datas_s[i,j]!="null"){
        print "<p><b>Tempo de Estadia:</b></p>" > (n parques);
        split(t_datas_e[i,j],d,"-");
        split(t_horas_entrada[i,j],h,":");
        de = mktime(d[3] " " d[2] " " d[1] " " h[1] " " h[2] " 00");
        split(t_datas_s[i,j],d,"-");
        split(t_horas_saida[i,j],h,":");
    }
}

```



```

        ds = mktime(d[3] " " d[2] " " d[1] " " h[1] " " h[2] " 00");
        tempo_total+=((ds - de)/60);
        print "<p>"((ds - de)/60)" minutos</p>" > (n parques);
    }
    print "<p><b>Local do Parque (clique para ver no google maps):</b></p>" > (n parques);
    print "<a href=\"https://www.google.pt/maps/place/"t_saidas[i,j]"/\"/>"t_saidas[i,j]"</a>" > (n p
}
print "<p><p><p style=\"color:red;font-size:160%;\">Tempo Total em Parques "tempo_total" minutos</p>
print "</body>" > (n parques);
print "</html>" > (n parques);
#Ficheiro de parques deste extracto terminado

}
print "</body>" > extractos;
print "</html>" > extractos;
#Ficheiro de extractos terminado


print "\t|          Ficheiros Construidos com Sucesso          |";
print "\t|          -----          |";
print "\t\\_____/"
}

```