

Processamento de Linguagens (3º ano de Curso)

Trabalho Prático I

Relatorio de Desenvolvimento

Bruno Cancelinha
(A75428)

José Bastos
(A74696)

Marcelo Miranda
(A74817)

15 de Março de 2017

Resumo

Um ficheiro de XML, embora bom para guardar informação, não é bom para apresenta-la nem para tirar conclusões acerca dessa mesma informação. O presente relatório apresenta uma solução a este problema, com um conjunto de *scripts* que processam a informação contida no XML e a apresenta no formato HTML, dando uma interface agradável ao consultante.

Conteúdo

| | | |
|----------|--|-----------|
| 1 | Introdução | 2 |
| 1.1 | Estrutura do Relatório | 2 |
| 2 | Análise e Especificação | 3 |
| 2.1 | Análise do Problema | 3 |
| 2.2 | Objetivos | 3 |
| 2.2.1 | Dados | 3 |
| 2.2.2 | Pedidos | 4 |
| 3 | Implementação | 7 |
| 3.1 | Estrutura genérica dos módulos | 8 |
| 4 | Codificação e Testes | 9 |
| 4.1 | Testes realizados e Resultados | 13 |
| 5 | Conclusão | 16 |
| A | Código do Programa | 17 |
| A.0.1 | Index | 17 |
| A.0.2 | Gastos mensais | 19 |
| A.0.3 | Gastos mensais em parques | 20 |
| A.0.4 | Gastos mensais em portangens | 21 |
| A.0.5 | Locais de saída | 22 |
| A.0.6 | Número de entradas por dia | 23 |
| A.0.7 | Transações realizadas | 24 |
| A.0.8 | Biblioteca | 25 |

Capítulo 1

Introdução

1.1 Estrutura do Relatório

O relatório começa por fazer uma contextualização do assunto tratado neste projeto, assim como uma análise detalhada do problema proposto. Esta é seguida por uma introdução onde são apresentados os vários objetivos pretendidos.

Depois, será ainda exposto o tema sobre o qual o trabalho será desenvolvido, fazendo um levantamento dos dados que serão processados e dos vários processos pelos quais estes vão passar, cuja implementação será apresentada detalhadamente nos futuros capítulos.

O relatório termina fazendo uma análise crítica relativa quer ao desenvolvimento do projeto quer ao seu estado final.

Capítulo 2

Análise e Especificação

2.1 Análise do Problema

Por questões de organização, a *Via Verde* mantém o extrato do cliente, registrando cada transação efetuada num ficheiro XML. Este ficheiro contém vários dados relativos ao cliente, os quais precisam de ser tratados para permitirem uma visualização mais natural, para além de processar estes dados de forma a simplificar a sua análise e permitir uma conclusão rápida.

2.2 Objetivos

Com a finalização deste projeto será possível, dado um ficheiro XML com a informação relativa ao extrato de um cliente, gerar um conjunto de páginas HTML que permitem uma fácil consulta dos dados do primeiro ficheiro. Capaz de apresentar respostas para os seguintes pedidos:

1. Total gasto durante o mês
2. Total gasto durante o mês só em parques
3. Total gasto durante o mês só em portagens
4. Lista de locais de saída
5. Número de entradas ao longo do mês
6. Lista total de transferências

Para calcular estes resultados, é necessário analisarmos o ficheiro XML e identificar toda a informação que ele possui.

2.2.1 Dados

O extrato contém informação relativa ao cliente, bem como dados quem compõem cada transação.

Dados relativos ao cliente

```
1 <cliente id="14932">
2   <nif>321973123</nif>
3   <nome>Fernando Alberto dos Campos Pessoa</nome>
4   <morada>Rua 1 de Janeiro n 9</morada>
5   <localidade>Porto</localidade>
6   <codigo_postal>4100-367</codigo_postal>
7 </cliente>
```

O excerto de cima apresenta o elemento *cliente* e todos os seus elementos filho. O elemento *cliente* possui o atributo *id*, bem como um conjunto de elementos filhos com informação relativa ao nif, nome, morada, localidade e código postal do cliente.

Dados relativos à transação

```
1 <transacao>
2   <data_entrada>26-07-2016</data_entrada>
3   <hora_entrada>11:33</hora_entrada>
4   <entrada>Povoa N-S</entrada>
5   <data_saida>26-07-2016</data_saida>
6   <hora_saida>11:50</hora_saida>
7   <saida>Ponte Pedra</saida>
8   <importancia>0,65</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>05-08-2016</data_debito>
14  <cartao>6749036</cartao>
15 </transacao>
```

Agora conseguimos distinguir a informação relativa à transação. A transação simboliza um evento como, por exemplo, percorrer um troço de estrada ou estacionar o carro num parque de estacionamento. De forma a manter os dados relativos a este evento, é necessário guardar informação com a data, hora e designação quer da entrada como da saída, bem como o preço final (designado acima por *importância*), o valor de desconto, a taxa de iva, o operador, o tipo de evento bem com o a data do débito e o número de cartão do cliente.

2.2.2 Pedidos

Dos dados acima, os que de facto são relevantes para estatísticas, análise e tomadas de decisão, são os relativos à transação, que descrevem detalhadamente toda a situação. Assim, vamos analisar os diferentes pedidos a que o sistema pode responder e identificar qual a informação relevante para formular uma resposta oportuna.

Total gasto durante o mês

```
1 <data_saida>26-07-2016</data_saida>
2 <importancia>0,65</importancia>
3 <valor_desconto>0,00</valor_desconto>
4 <taxa_iva>23</taxa_iva>
```

Decidimos apresentar a informação separada mensalmente, por exemplo, no caso apresentado acima o gasto é relativo ao mês 7 (Julho). Para calcular o gasto total e o total de iva, são usadas as seguintes formulas:

$$gasto(mês) = \sum_{i=1}^N imp$$
$$total\ iva(mês) = \sum_{i=1}^N imp \times (iva \div 100)$$

Em que *mês* é o mês especificado na entidade **data_saida**, *N* é o total de transações a serem analisadas, *imp* é o valor na entidade **importancia** do código acima e o *iva* é o valor na entidade **taxa_iva**. O valor de desconto não é considerado visto que o valor *imp* já contém o desconto bem como o iva.

Total gasto em parques e portagens

Visto que este pedido é uma especificação do pedido anterior, a única informação adicional que é necessária é o tipo, como é possível observar no exemplo seguinte:

```
1 <tipo>Portagens</tipo>
```

Baseado na entidade **tipo**, podemos distinguir gastos e especificar as fórmulas acima para serem aplicadas apenas quando o tipo coincide com o pretendido.

Lista de locais de saída

Para calcular uma lista com todos os locais de saída, é apenas necessário olhar para a entidade **saida** que está demonstrada no exemplo que se segue:

```
1 <saida>Ponte Pedra</saida>
```

Basta armazenar a informação de cada **saida** num array.

Número de entradas ao longo do mês

Para este pedido, registamos o número de viagens que foram efetuadas em cada data.

```
1 <data_entrada>26-07-2016</data_entrada>
```

Como é um pedido bastante simples, basta associar o valor dentro da entidade **data_entrada** ao número de vezes que aparece no estrato deste mês.

Lista total de transações

Para obter uma lista total de transações, guardamos toda a informação contida numa transação e apresentamos-la numa tabela. Assim, todas as entidades-filho da entidade **transacao** são importantes.

Capítulo 3

Implementação

A implementação da aplicação foi realizada de forma modular, procurando resolver cada um dos objetivos acima referidos isoladamente. Assim sendo, desenvolvemos um conjunto de módulos opcionais que correspondem a cada uma das funcionalidades oferecidas pela aplicação, produzindo cada um deles uma página *html* distinta com o respetivo *output*. Por questões de organização, os vários ficheiros *output* são agregados numa diretoria *pages*. Dadas as semelhanças na implementação dos vários módulos, foi também criada uma biblioteca evitando assim as definições repetidas entre os vários módulos.

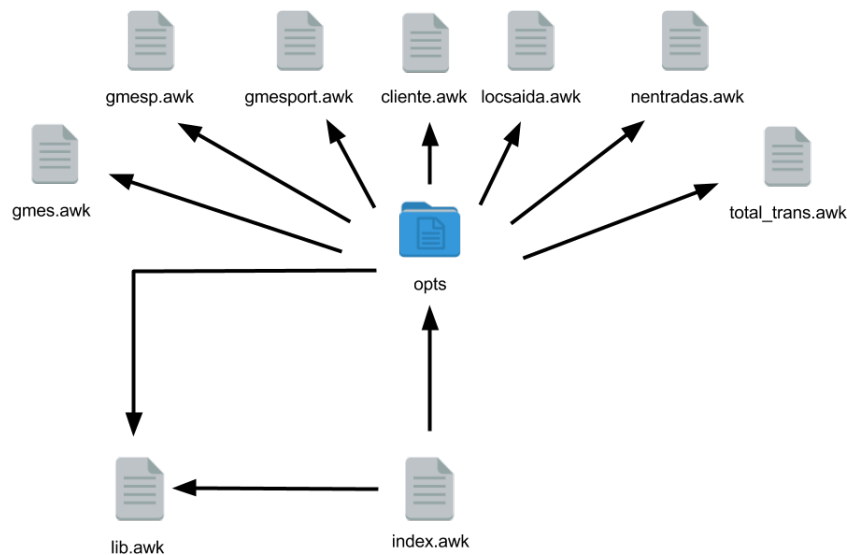


Figura 3.1: Árvore do projeto

Por fim, de forma a agregar os vários módulos, foi ainda necessário desenvolver o *index.awk*, o qual é responsável por incluir os respetivos módulos e disponibiliza ainda uma interface que permite uma fácil navegação entre estes.

3.1 Estrutura genérica dos módulos

Todos os módulos desenvolvidos obedecem, até um certo ponto, à estrutura apresentada. No início do programa, são declaradas as variáveis locais ao módulo, tal como a ordem pela qual devem ser efetuadas as travessias. É ainda inicializado o respetivo ficheiro *html* para o qual será direcionado o *output*.

Para extrair os dados pretendidos escolhemos usar a função *match*, a qual retorna a posição em que uma dada expressão regular ocorre, permitindo assim que esta atue como uma condição. Quando a expressão regular não ocorre, a função *match* retorna zero e a ação não é executada. O *match* admite ainda que os grupos capturados na expressão regular sejam guardados para um *array*, o que permite extrair o conteúdo das *tags* conforme o desejado. Conforme especificado no *lib.awk* a procura de correspondências para as expressões regulares é feita de forma *case insensitive*.

```
1 @include "lib.awk"
2
3 BEGIN {
4     PROCINFO["sorted_in"] = "(...)"
5
6     path = "opts/module.html"
7     title = "Module title"
8
9     print "<html> header "<body>" title > path
10 }
11
12 match($0, /<TAG>(.*?)</TAG>/, arr) {
13     (...)
14 }
15
16 END {
17     print "(...)"
18     print back_link "</body></html>" > path
19 }
```

Após o processamento do *input*, os dados obtidos são imprimidos para o respetivo ficheiro *output*, ao qual é ainda adicionado tanto o *link* que permite a navegação entre os módulos como as *tags* necessárias para terminar devidamente o ficheiro *html*.

Capítulo 4

Codificação e Testes

Neste capítulo pretendemos mostrar o output do ficheiro dado da via verde assim como outros exemplos de input e outras partes que considerámos relevantes do output de ficheiros teste.

-
- [Informações do Cliente](#)
 - [Total gasto durante o mês](#)
 - [Total gasto em parques](#)
 - [Total gasto em portagens](#)
 - [Lista de locais de saída](#)
 - [Número de entradas ao longo do mês](#)
 - [Lista total de transferências](#)

Figura 4.1: Index

Informação do cliente

- **Nome:** PEDRO MANUEL RANGEL SANTOS HENRIQUES
- **Nif:** 987653210
- **Morada:** RUA XXX
- **Localidade:** BRAGA
- **Codigo Postal:** 4715-012 BRAGA

Figura 4.2: Informações do Cliente

Gastos ao longo do mês

| | Mês | Total | Total IVA |
|----------------|------|-------|-----------|
| Julho: | 20.3 | | 4.669 |
| Agosto: | 57.1 | | 13.133 |

[Voltar](#)

Figura 4.3: Total gasto durante o mês

Gasto em parques

| | Mês | Total | Total IVA |
|----------------|------|-------|-----------|
| Julho: | 2.6 | | 0.598 |
| Agosto: | 3.75 | | 0.8625 |

[Voltar](#)

Figura 4.4: Total gasto durante o mês em parques de estacionamento

Gasto em portagens

| | Mês | Total | Total IVA |
|----------------|-------|-------|-----------|
| Julho: | 17.7 | | 4.071 |
| Agosto: | 53.35 | | 12.2705 |

[Voltar](#)

Figura 4.5: Total gasto em portagens

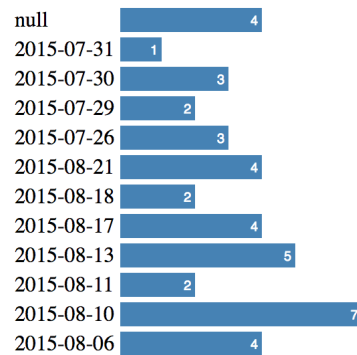
Locais de saída

- Aeroporto
- Angeiras N-S
- Braga Sul
- Custoias
- EN 205 PV
- EN107
- Ermesinde PV
- Ferreiros
- Freixieiro
- Lipor
- Maia II
- Maia PV
- Neiva N-S
- Neiva S-N
- Ponte Pedra
- Povia S-N
- PQ A Sa Carn.I
- PQ Av. Central
- Valongo

[Voltar](#)

Figura 4.6: Lista de locais de saída

Número de entradas durante o mês



[Voltar](#)

Figura 4.7: Número de entradas ao longo do mês

Todas as transações

[Voltar](#)

| Data Entrada | Hora Entrada | Entrada | Data Saída | Hora Saída | Saída | Importância | Valor Desconto | Taxa IVA | Operador | Tipo | Data Débito | Cartão |
|--------------|--------------|---------------|------------|------------|----------------|-------------|----------------|----------|---------------------------------------|---------------------------|-------------|---------|
| 26-07-2015 | 11:33 | Povoa N-S | 26-07-2015 | 11:42 | Angeiras N-S | 2,00 | 0,00 | 23 | I. de Portugal (N1) | Portagens | 05-08-2015 | 6749036 |
| 26-07-2015 | 11:45 | Aeroporto | 26-07-2015 | 11:50 | Ponte Pedra | 0,65 | 0,00 | 23 | I. de Portugal (P3) | Portagens | 05-08-2015 | 6749036 |
| 26-07-2015 | 23:26 | Angeiras S-N | 26-07-2015 | 23:37 | Povoa S-N | 2,00 | 0,00 | 23 | I. de Portugal (N1) | Portagens | 05-08-2015 | 6749036 |
| 29-07-2015 | 09:55 | EN 205 PV | 29-07-2015 | 10:07 | Ferreiros | 2,60 | 0,00 | 23 | I. de Portugal (E1) | Portagens | 07-08-2015 | 6749036 |
| 29-07-2015 | 20:06 | Ferreiros | 29-07-2015 | 20:18 | EN 205 PV | 2,60 | 0,00 | 23 | I. de Portugal (E1) | Portagens | 07-08-2015 | 6749036 |
| 30-07-2015 | 15:35 | EN 205 PV | 30-07-2015 | 15:46 | Ferreiros | 2,60 | 0,00 | 23 | I. de Portugal (E1) | Portagens | 07-08-2015 | 6749036 |
| 30-07-2015 | 19:13 | Braga Sul | 30-07-2015 | 19:31 | Maia PV | 3,25 | 0,00 | 23 | Portagens | 07-08-2015 | 6749036 | |
| 31-07-2015 | 00:10 | Angeiras S-N | 31-07-2015 | 00:29 | Povoa S-N | 2,00 | 0,00 | 23 | I. de Portugal (N1) | Portagens | 07-08-2015 | 6749036 |
| 06-08-2015 | 15:39 | Povoa N-S | 06-08-2015 | 15:50 | Angeiras N-S | 2,00 | 0,00 | 23 | I. de Portugal (N1) | Portagens | 14-08-2015 | 6749036 |
| null | | | 06-08-2015 | 15:53 | Aeroporto | 0,20 | 0,00 | 23 | I. de Portugal (P3) | Portagens | 14-08-2015 | 6749036 |
| 06-08-2015 | 17:13 | EN13 O/E | 06-08-2015 | 17:14 | EN107 | 0,80 | 0,00 | 23 | I. de Portugal (P3) | Portagens | 14-08-2015 | 6749036 |
| 06-08-2015 | 17:18 | Maia II | 06-08-2015 | 17:39 | Braga Sul | 3,25 | 0,00 | 23 | Portagens | 14-08-2015 | 6749036 | |
| 10-08-2015 | 13:00 | Braga Sul | 10-08-2015 | 13:19 | Maia II | 3,25 | 0,00 | 23 | Portagens | 14-08-2015 | 6749036 | |
| 10-08-2015 | 15:35 | Ermesinde PV | 10-08-2015 | 15:38 | Valongo | 0,35 | 0,00 | 23 | Portagens | 14-08-2015 | 6749036 | |
| 10-08-2015 | 13:22 | EN14 E/O | 10-08-2015 | 13:24 | Lipor | 0,80 | 0,00 | 23 | I. de Portugal (P3) | Portagens | 21-08-2015 | 6749036 |
| 10-08-2015 | 15:31 | Via Norte O/E | 10-08-2015 | 15:31 | Ponte Pedra | 0,45 | 0,00 | 23 | I. de Portugal (P3) | Portagens | 21-08-2015 | 6749036 |
| 10-08-2015 | 18:02 | Valongo | 10-08-2015 | 18:05 | Ermesinde PV | 0,35 | 0,00 | 23 | Portagens | 21-08-2015 | 6749036 | |
| 10-08-2015 | 18:09 | Via Norte E/O | 10-08-2015 | 18:13 | Freixieiro | 0,65 | 0,00 | 23 | I. de Portugal (P3) | Portagens | 21-08-2015 | 6749036 |
| 10-08-2015 | 18:16 | Angeiras S-N | 10-08-2015 | 18:29 | Povoa S-N | 2,00 | 0,00 | 23 | I. de Portugal (N1) | Portagens | 21-08-2015 | 6749036 |
| 11-08-2015 | 13:55 | EN 205 PV | 11-08-2015 | 14:07 | Ferreiros | 2,60 | 0,00 | 23 | I. de Portugal (E1) | Portagens | 21-08-2015 | 6749036 |
| 11-08-2015 | 17:52 | Ferreiros | 11-08-2015 | 18:04 | EN 205 PV | 2,60 | 0,00 | 23 | I. de Portugal (E1) | Portagens | 21-08-2015 | 6749036 |
| 13-08-2015 | 10:36 | Povoa N-S | 13-08-2015 | 10:45 | Angeiras N-S | 2,00 | 0,00 | 23 | I. de Portugal (N1) | Portagens | 21-08-2015 | 6749036 |
| 13-08-2015 | 11:26 | Maia PV | 13-08-2015 | 11:42 | Braga Sul | 3,25 | 0,00 | 23 | Portagens | 21-08-2015 | 6749036 | |
| 13-08-2015 | 13:03 | Ferreiros | 13-08-2015 | 13:16 | EN 205 PV | 2,60 | 0,00 | 23 | I. de Portugal (E1) | Portagens | 21-08-2015 | 6749036 |
| 13-08-2015 | 19:21 | EN 205 PV | 13-08-2015 | 19:33 | Ferreiros | 2,60 | 0,00 | 23 | I. de Portugal (E1) | Portagens | 21-08-2015 | 6749036 |
| 13-08-2015 | 20:09 | Ferreiros | 13-08-2015 | 20:20 | EN 205 PV | 2,60 | 0,00 | 23 | I. de Portugal (E1) | Portagens | 21-08-2015 | 6749036 |
| null | | | 14-08-2015 | 12:29 | Neiva S-N | 0,65 | 0,00 | 23 | I. de Portugal (N1) | Portagens | 21-08-2015 | 6749036 |
| null | | | 14-08-2015 | 17:31 | Neiva N-S | 0,65 | 0,00 | 23 | I. de Portugal (N1) | Portagens | 21-08-2015 | 6749036 |
| 17-08-2015 | 10:49 | Povoa N-S | 17-08-2015 | 10:59 | Angeiras N-S | 2,00 | 0,00 | 23 | I. de Portugal (N1) | Portagens | 31-08-2015 | 6749036 |
| 17-08-2015 | 11:01 | Aeroporto | 17-08-2015 | 11:06 | Ponte Pedra | 0,65 | 0,00 | 23 | I. de Portugal (P3) | Portagens | 31-08-2015 | 6749036 |
| 17-08-2015 | 14:11 | Via Norte E/O | 17-08-2015 | 14:12 | Custoias | 0,45 | 0,00 | 23 | I. de Portugal (P3) | Portagens | 31-08-2015 | 6749036 |
| null | | | 17-08-2015 | 14:28 | Freixieiro | 0,20 | 0,00 | 23 | I. de Portugal (P3) | Portagens | 31-08-2015 | 6749036 |
| 17-08-2015 | 14:31 | Angeiras S-N | 17-08-2015 | 14:41 | Povoa S-N | 2,00 | 0,00 | 23 | I. de Portugal (N1) | Portagens | 31-08-2015 | 6749036 |
| 18-08-2015 | 12:44 | Povoa N-S | 18-08-2015 | 12:54 | Angeiras N-S | 2,00 | 0,00 | 23 | I. de Portugal (N1) | Portagens | 31-08-2015 | 6749036 |
| 18-08-2015 | 16:10 | Angeiras S-N | 18-08-2015 | 16:32 | Povoa S-N | 2,00 | 0,00 | 23 | I. de Portugal (N1) | Portagens | 31-08-2015 | 6749036 |
| 21-08-2015 | 10:43 | EN 205 PV | 21-08-2015 | 10:56 | Ferreiros | 2,60 | 0,00 | 23 | I. de Portugal (E1) | Portagens | 31-08-2015 | 6749036 |
| 21-08-2015 | 11:19 | Ferreiros | 21-08-2015 | 11:32 | EN 205 PV | 2,60 | 0,00 | 23 | I. de Portugal (E1) | Portagens | 31-08-2015 | 6749036 |
| 21-08-2015 | 18:09 | EN 205 PV | 21-08-2015 | 18:22 | Ferreiros | 2,60 | 0,00 | 23 | I. de Portugal (E1) | Portagens | 31-08-2015 | 6749036 |
| 21-08-2015 | 19:23 | Ferreiros | 21-08-2015 | 19:35 | EN 205 PV | 2,60 | 0,00 | 23 | I. de Portugal (E1) | Portagens | 31-08-2015 | 6749036 |
| 30-07-2015 | 16:54 | | 30-07-2015 | 18:34 | PQ Av. Central | 2,60 | 0,00 | 23 | BRAGAPARQUES (BP) | Parques de estacionamento | 07-08-2015 | 6749036 |
| 06-08-2015 | 15:57 | | 06-08-2015 | 17:08 | PQ A Sa Cam.1 | 3,75 | 0,00 | 23 | ANA - Aeroportos de Portugal. SA (AP) | Parques de estacionamento | 14-08-2015 | 6749036 |

Figura 4.8: Lista total de transferências

4.1 Testes realizados e Resultados

Mostram-se a seguir alguns testes feitos (ficheiros xml) e os respectivos resultados obtidos. Primeiro fez-se um teste em que algumas das tags do ficheiro input, na parte da informação do cliente continham letras maiúsculas, minúsculas ou ambas. Tais mudanças não refletiram, como esperado, o output, que permaneceu igual.

```
1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <EXTRACTO id="011123232/02/2017">
3   <MES_EMISSAO>Fev-2017</MES_EMISSAO>
4   <CLIENTE id="916095177">
5     <NIF>222555816</NIF>
6     <nome>JosE DIogo PaiVa BaStos</NoMe>
7     <mOraDA>Rua Dali</Morada>
8     <LocaLIDade>BRAGA</LOCALIDADE>
9     <CODIGO_POSTAL>4715-080 braga</CODIGO_POSTAL>
10  </CLIENTE>
11  <IDENTIFICADOR id="38263829188">
12    <MATRICULA>32-ZB-96</MATRICULA>
13    <REF_PAGAMENTO>1234569</REF_PAGAMENTO>
14    <TRANSACCAO>
15      <DATA_ENTRADA>6-02-2017</DATA_ENTRADA>
16      <HORA_ENTRADA>11:33</HORA_ENTRADA>
17      <ENTRADA>Abrantes N-S</ENTRADA>
18      <DATA_SAIDA>6-02-2017</DATA_SAIDA>
19      <HORA_SAIDA>11:42</HORA_SAIDA>
20      <SAIDA>carrinheiros N-S</SAIDA>
21      <IMPORTANCIA>8,00</IMPORTANCIA>
22      <VALOR_DESCONTO>4,00</VALOR_DESCONTO>
23      <TAXA_IVA>23</TAXA_IVA>
24      <OPERADOR>I. de Portugal (N1)</OPERADOR>
25      <TIPO>Portagens</TIPO>
26      <DATA_DEBITO>08-02-2017</DATA_DEBITO>
27      <CARTAO>6749037</CARTAO>
28    </TRANSACCAO>
29  <TRANSACCAO>
30    <DATA_ENTRADA>09-02-2017</DATA_ENTRADA>
31    <HORA_ENTRADA>15:57</HORA_ENTRADA>
32    <ENTRADA/>
33    <DATA_SAIDA>09-02-2017</DATA_SAIDA>
34    <HORA_SAIDA>17:08</HORA_SAIDA>
35    <SAIDA>Parque Cidade Braga</SAIDA>
36    <IMPORTANCIA>3,75</IMPORTANCIA>
37    <VALOR_DESCONTO>0,00</VALOR_DESCONTO>
38    <TAXA_IVA>23</TAXA_IVA>
39    <OPERADOR>Camara Municipal Braga</OPERADOR>
40    <TIPO>Parques de estacionamento</TIPO>
41    <DATA_DEBITO>14-02-2017</DATA_DEBITO>
```

```

42     <CARTAO>6749037</CARTAO>
43 </TRANSACCAO>
44 <TOTAL>12,75</TOTAL>
45 </IDENTIFICADOR>
46 <TOTAL>12,75</TOTAL>
47 <TOTAL_IVA>2.93</TOTAL_IVA>
48 </EXTRACTO>
49 }

```

De seguida fez-se outra teste em que o ficheiro não continha informação relativamente ao cliente.

```

1  <?xml version="1.0" encoding="ISO-8859-1"?>
2  <EXTRACTO id="011123232/02/2017">
3    <MES_EMISSAO>Fev-2017</MES_EMISSAO>
4    <IDENTIFICADOR id="38263829188">
5      <MATRICULA>32-ZB-96</MATRICULA>
6      <REF_PAGAMENTO>1234569</REF_PAGAMENTO>
7      <TRANSACCAO>
8        <DATA_ENTRADA>6-02-2017</DATA_ENTRADA>
9        <HORA_ENTRADA>11:33</HORA_ENTRADA>
10       <ENTRADA>Abrantes N-S</ENTRADA>
11       <DATA_SAIDA>6-02-2017</DATA_SAIDA>
12       <HORA_SAIDA>11:42</HORA_SAIDA>
13       <SAIDA>carrinheiros N-S</SAIDA>
14       <IMPORTANCIA>8,00</IMPORTANCIA>
15       <VALOR_DESCONTO>4,00</VALOR_DESCONTO>
16       <TAXA_IVA>23</TAXA_IVA>
17       <OPERADOR>I. de Portugal (N1)</OPERADOR>
18       <TIPO>Portagens</TIPO>
19       <DATA_DEBITO>08-02-2017</DATA_DEBITO>
20       <CARTAO>6749037</CARTAO>
21     </TRANSACCAO>
22   <TRANSACCAO>
23     <DATA_ENTRADA>09-02-2017</DATA_ENTRADA>
24     <HORA_ENTRADA>15:57</HORA_ENTRADA>
25     <ENTRADA/>
26     <DATA_SAIDA>09-02-2017</DATA_SAIDA>
27     <HORA_SAIDA>17:08</HORA_SAIDA>
28     <SAIDA>Parque Cidade Braga</SAIDA>
29     <IMPORTANCIA>3,75</IMPORTANCIA>
30     <VALOR_DESCONTO>0,00</VALOR_DESCONTO>
31     <TAXA_IVA>23</TAXA_IVA>
32     <OPERADOR>Camara Municipal Braga</OPERADOR>
33     <TIPO>Parques de estacionamento</TIPO>
34     <DATA_DEBITO>14-02-2017</DATA_DEBITO>
35     <CARTAO>6749037</CARTAO>
36   </TRANSACCAO>
37   <TOTAL>12,75</TOTAL>
38 </IDENTIFICADOR>

```



```
39 | <TOTAL>12,75</TOTAL>
40 | <TOTAL_IVA>2.93</TOTAL_IVA>
41 | </EXTRACTO>
```

No output, na página html relativa à informação do cliente, esta encontrava-se sem dados, como esperado.

Informação do cliente

- **Nome:**
- **Nif:**
- **Morada:**
- **Localidade:**
- **Codigo Postal:**

Figura 4.9: Sem informação do cliente

Capítulo 5

Conclusão

A realização deste projeto foi fundamental para se consolidar a matéria lecionada nas aulas práticas. Tornou-nos mais à vontade com expressões regulares e a linguagem AWK que foram cruciais no desenvolvimento de este trabalho.

Temos agora o nosso conjunto de *scripts* a levantarem a informação desejada de um ficheiro e transforma-la para um formato bem mais legível e que descomplica a posterior análise. Somos obrigados reconhecer o potencial de AWK e regex, que facilitam o processamento e interpretação de texto.

Apêndice A

Código do Programa

A.0.1 Index

```
#!/usr/bin/gawk -f
@include "opts/nentradas.awk"
@include "opts/cliente.awk"
@include "opts/locsaida.awk"
@include "opts/gmes.awk"
@include "opts/gmesp.awk"
@include "opts/gmesport.awk"
@include "opts/total_trans.awk"

BEGIN {
    PROCINFO["sorted_in"] = std_sorted_in

    opts["nentradas"] = "Número de entradas ao longo do mês"
    opts["locsaida"] = "Lista de locais de saída"
    opts["gmes"] = "Total gasto durante o mês"
    opts["gmesp"] = "Total gasto em parques"
    opts["gmesport"] = "Total gasto em portagens"
    opts["total_trans"] = "Lista total de transferências"
    opts["cliente"] = "Informações do Cliente"

    print_index()
    print_css()
}

function print_index() {
    header = "<head><meta charset='UTF-8'></meta></head>"
    link = "<li><a href=%s>%s</a></li>"

    print "<html>" header "<body>" > "pages/index.html"
```

```

    for (i in opts)
        printf (link, i ".html" , opts[i]) > "pages/index.html"

    print "</body></html>" > "pages/index.html"
}

function print_css() {
    print"\
.bar {\
    font: 10px sans-serif;\
    background-color: steelblue;\
    text-align: right;\
    display: inline-block;\
    padding: 3px;\
    margin: 1px;\
    color: white;\
}" > "pages/all.css"
}

```

A.0.2 Gastos mensais

```
@include "lib.awk"

BEGIN {
    gmes_table = "<table><tr><th>Mês</th> <th>Total</th> <th>Total IVA</th>"
    gmes_path = "pages/gmes.html"
    gmes_title = "<h1>Gastos ao longo do mês</h1>"

    print "<html>" header "<body>" gmes_title gmes_table > gmes_path
}

match($0, /<DATA_SAIDA>(.)<\/DATA_SAIDA>/, m) {
    mes = get_mes(m[1])
}

match($0, /<IMPORTANCIA>(.)<\/IMPORTANCIA>/, m) {
    imp = normFloat(m[1])
}

match($0, /<TAXA_IVA>(.)<\/TAXA_IVA>/, m) {
    iva = m[1]
}

match($0, /<\/TRANSACCAO>/) {
    total[mes] += imp
    total_iva[mes] += imp * ("0." iva)
}

END {
    for (mes in total)
        printf table_tr_3, meshr(mes), total[mes], total_iva[mes] > gmes_path

    print "</table>" back_link "</body></html>" > gmes_path
}
```

A.0.3 Gastos mensais em parques

```
@include "lib.awk"

BEGIN {
    gmesp_path = "pages/gmesp.html"
    gmesp_title = "<h1>Gasto em parques</h1>"

    print "<html>" header "<body>" gmesp_title gmes_table > gmesp_path
}

match($0, /<TIPO>(.*?)<\/TIPO>/, m) && tolower(m[1]) == "parques de estacionamento" {
    totalp[mes] += imp
    totalp_iva[mes] += imp * ("0." iva)
}

END {
    for (mes in totalp)
        printf table_tr_3, meshr(mes), totalp[mes], totalp_iva[mes] > gmesp_path

    print "</table>" back_link "</body></html>" > gmesp_path
}
```

A.0.4 Gastos mensais em portagens

```
@include "lib.awk"

BEGIN {
    gmesport_path = "pages/gmesport.html"
    gmesport_title = "<h1>Gasto em portagens</h1>"

    print "<html>" header "<body>" gmesport_title gmes_table > gmesport_path
}

match($0, /<TIPO>(.*?)<\/TIPO>/, m) && tolower(m[1]) == "portagens" {
    totalport[mes] += imp
    totalport_iva[mes] += imp * ("0." iva)
}

END {
    for (mes in totalport)
        printf table_tr_3, meshr(mes), totalport[mes], totalport_iva[mes] > gmesport_path

    print "</table>" back_link "</body></html>" > gmesport_path
}
```

A.0.5 Locais de saída

```
@include "lib.awk"

BEGIN {
    locsaida_path = "pages/locsaida.html"
    locsaida_li = "<li>%s</li>\n"
    locsaida_title = "<h1>Locais de saída</h1>"

    print "<html>" header "<body>" locsaida_title "<ul>" > locsaida_path
}

match($0, /<SAIDA>(.*?)<\/SAIDA>/, m) {
    !saidas[m[1]]++;
}

END {
    for (s in saidas)
        printf (locsaida_li, s) > locsaida_path

    print "</ul>" back_link "</body></html>" > locsaida_path
}
```


A.0.6 Número de entradas por dia

```
@include "lib.awk"

BEGIN {

    nentradas_path = "pages/nentradas.html"
    entradas_fmt = "<tr><td>%s</td><td class='bar' style='width: %s'>%s</td>"
    entradas_title = "<h1>Número de entradas durante o mês</h1>"

    print "<html>" header "<body>" entradas_title "<table>" > nentradas_path
}

match($0, /<DATA_ENTRADA>([~>]*)<\/DATA_ENTRADA>/, m) {
    dates[m[1]]++
}

END {
    PROCINFO["sorted_in"] = "@ind_str_desc"

    for (date in dates)
        printf(entradas_fmt, normalize(date), dates[date]*25, dates[date]) > nentradas_path

    print "</table>" back_link "</body></html>" > nentradas_path

    PROCINFO["sorted_in"] = std_sorted_in
}
```

A.0.7 Transações realizadas

```
@include "lib.awk"

BEGIN {
    trans = 0
    total_trans_path = "pages/total_trans.html"
    total_trans_title = "<h1>Todas as transações</h1>"
    trans_table_header = "<table> <tr> <th>Data Entrada</th> <th>Hora Entrada</th><th>Entrada</th>"

    print "<html>" header "<body>" total_trans_title "<ul>" > total_trans_path
    print trans_table_header > total_trans_path
}

match($0, /<TRANSACCAO>/, m) {
    trans++
}

match($0, /<(.*?)>(.*?)<\/.*>/, m) && trans {
    trans_info[++p] = m[2]
}

match($0, /<.*\/>/) && trans {
    trans_info[++p] = ""
}

match($0, /<\/TRANSACCAO>/, m) && trans {
    PROCINFO["sorted_in"] = "@ind_num_asc"
    for (desc in trans_info)
        trans_row_data = trans_row_data "<td>" trans_info[desc] "</td>"

    print "<tr>" trans_row_data "</tr>" > total_trans_path

    trans_row_data = ""
    delete_array(trans_info)
    trans = 0
    p = 0
    PROCINFO["sorted_in"] = std_sorted_in
}

END {
    print end_table > total_trans_path
    print "</ul>" back_link "</body></html>" > total_trans_path
}
```

A.0.8 Biblioteca

```
BEGIN {
    IGNORECASE = 1
    system("mkdir -p pages")
    header = "<head><meta charset='UTF-8' /><link rel='stylesheet' href='all.css' /></head>"
    back_link = "<p><i><a href='index.html'>Voltar</a></i></p>"

    table_tr_3 = "<tr><td><b>%s:</b></td> <td> %s </td> <td> %s </td></tr>\n"
    table_tr_13 = "<tr> <td>%s</td> <td>%s</td> <td>%s</td> <td>%s</td> <td>%s</td> <td>%s</td>"
    std_sorted_in = "@ind_str_asc";
}

function normalize(date) {
    split(date, d, "-", sp)
    return d[3] sp[1] d[2] sp[1] d[1]
}

function get_mes(date) {
    split(date, d, "-")
    return d[2]
}

function normFloat(float) {
    sub(/,//, ".", float)
    return float
}

function delete_array(array) {
    for (i in array)
        delete array[i]
}

function meshr(mes) {
    switch (mes) {
        case "01": return "Janeiro"
        case "02": return "Fevereiro"
        case "03": return "Março"
        case "04": return "Abril"
        case "05": return "Maio"
        case "06": return "Junho"
        case "07": return "Julho"
        case "08": return "Agosto"
        case "09": return "Setembro"
        case "10": return "Outubro"
    }
}
```

```
        case "11": return "Novembro"  
        case "12": return "Dezembro"  
    }  
}
```