

Processamento de Linguagens
(3 ano de Mestrado Integrado em Engenharia Informática)
Trabalho Prático Nº1
Relatório de Desenvolvimento

Carlos Manuel Magalhães da Silva
(A75107)

João Rui de Sousa Miguel
(A74237)

Paulo Jorge Machado Guedes
(A74411)

15 de Março de 2017

Resumo

Neste primeiro trabalho, a partir do tratamento de um ficheiro *xml* com informações relativas a um extracto mensal da Via Verde de uma determinada matrícula, foram reunidas algumas informações que puderam ser obtidas a partir desse mesmo ficheiro.

Estas informações, tais como, o número de vezes que foi usado o dispositivo ViaVerde em cada dia do mês, os locais onde o dispositivo saiu, o que foi gasto durante o mês, entre outras, foram tratadas com a ferramenta de processamento de texto *gawk* e apresentados os resultados das queries executadas, em *html*.

Este *html* terá portanto, de forma visualmente apelativa, toda a informação que foi recolhida com a ferramenta usada.

Conteúdo

| | | |
|----------|---|-----------|
| 1 | Introdução | 2 |
| 2 | Análise e Especificação | 3 |
| 2.1 | Problema | 3 |
| 2.2 | Requisitos | 3 |
| 3 | Concepção/desenho da Resolução | 4 |
| 3.1 | Estruturas de Dados | 4 |
| 3.2 | Algoritmos | 4 |
| 4 | Codificação e Testes | 7 |
| 4.1 | Alternativas, Decisões e Problemas de Implementação | 7 |
| 4.1.1 | Alternativas | 7 |
| 4.1.2 | Decisões | 7 |
| 4.1.3 | Problemas de Implementação | 7 |
| 4.2 | Testes realizados e Resultados | 8 |
| 5 | Conclusão | 10 |

Capítulo 1

Introdução

Neste projeto foi nos apresentado um ficheiro, sobre o extrato mensal da Via Verde de um dado veículo no formato de *xml*, e foi nos solicitado que desenvolvêssemos um Processador de Texto que processasse o seu conteúdo de modo a ser apresentado no formato *html*, apresentando as informações processadas no ficheiro. O Processador de Texto desenvolvido utiliza o sistema de produção para filtragem do texto *gawk* que identifica as expressões regulares do ficheiro *xml* e filtra a informação. O ficheiro *html* criado pelo processador apresenta:

1. Os dados gerais do veículo, de modo a facilitar a identificação ou a obtenção de qualquer informação conectada á matrícula, assim como o mês do extrato em questão.
2. O número de entradas por dia do veículo incluindo as informações gerais de cada entrada.
3. Os gastos do extrato especificando os tipos de gastos, e a sua quantia.
4. Os detalhes de cada percurso percorrido pelo veículo em questão, assim como o trajeto, se possível, no Google Maps, a frequência que percorreu cada trajeto e os gastos de cada trajeto.

Realçamos o facto da apresentação dos dados estarem muito acessíveis, muito bem organizados, de fácil navegação e de grande agrado em termos estéticos.

Capítulo 2

Análise e Especificação

2.1 Problema

A empresa Via Verde guarda todas as informações sobre os veículos que estão registrados na empresa, de modo a facultar um extrato mensal de cada veículo sobre os gastos desse mês, apresentando também os detalhes sobre os gastos, os trajetos que percorreu, os detalhes de cada trajeto incluindo a hora de entrada e saída do percurso, entre outros.

Neste projeto foi nos solicitado que desenvolvêssemos um Processador de Texto com o sistema de produção para filtragem de texto *gawk* para processar todos os dados do extrato criado pela empresa Via Verde e que após analisadas todas as informações, que as apresentasse em formato *html*.

2.2 Requisitos

O extrato mensal providenciado pela empresa Via Verde contém, em formato *xml*, as informações do veículo com a matrícula especificada pelo extrato, assim como todas as transações provocadas pelo veículo, e as informações acerca das transações, como o local e hora de entrada e de saída, o operador e a quantia do trajeto. Com estas informações e após as analisar, o Processador de Texto desenvolvido cria ficheiros *html* onde vai guardar:

- As informações sobre os percursos do veículo, assim como a hora de entrada e as informações sobre o trajeto desse percurso, referenciando o número de vezes que efetuou o mesmo trajeto e a quantia gasta nesse mesmo, mostrando também, se possível, o trajeto no Google Maps;
- Os dados sobre os trajetos efetuados pelo veículo em cada dia, como o operador, local de entrada e saída e hora de entrada;
- Os gastos do extrato em relação ao mês especificado, especificando o tipo de gastos;

Criamos um ficheiro *index* em *html*, que apresenta todas as informações como especificado em cima tentando facilitar ao máximo a consulta das informações do extrato, contendo, no entanto, todas as informações relevantes do extrato.

Capítulo 3

Concepção/desenho da Resolução

3.1 Estruturas de Dados

Durante a realização do trabalho foram sendo encontradas soluções, para armazenamento de dados e para a apresentação dos mesmos.

Com o tratamento do *xml* e devido a algumas informações que se queria apresentar, surge a necessidade de, para cada linha lida com os parâmetros de divisão estabelecidos, guardar campos que mais tarde serão necessários na composição do resultado. Para que tal fosse possível, ao longo do tempo, foi-se guardando, em arrays, a informação que mais tarde iria ser usada. Esta estrutura de dados permite que também sejam criadas relações entre os vários campos lidos, como por exemplo, contar as frequências que alguns dados surgem.

Na apresentação de resultados, e para uma mais fácil leitura e consulta, decidiu-se usar *html*, em vez da consola. Desta forma, foi possível personalizar mais facilmente a informação retirada do ficheiro *xml*, adicionando até algumas funcionalidades que de outro modo não era possível. Esta informação é dividida por tipos, e, cada um destes tipos, corresponde a um ficheiro *html* diferente. Esta distribuição, permite que sejam organizados mais facilmente os resultados com links entre ficheiros *html*.

3.2 Algoritmos

O objectivo deste trabalho prático baseava-se no tratamento de um documento *xml*, para obtenção de informação sobre um extrato mensal da Via Verde de um cliente. Para se alcançar esse objetivo, e como já referido, foi usado o processador de texto *gawk*, tirando partido das suas funcionalidades. O processo de desenvolvimento do trabalho baseou-se nos seguintes pontos:

- Definir os caracteres pelo qual o ficheiro *xml* vai ser filtrado. Neste caso seriam os limitadores das *tags* em *xml*, como podemos ver na figura. Também definir construções de *html* para mais tarde facilitar a composição de ficheiros. Estas intruções, são inseridas no bloco BEGIN e são executadas uma só vez no início do programa.

```
BEGIN {
  PROCINFO["sorted_in"] = "@val_str_asc"
  FS = "[><]"
  enc = "<!DOCTYPE html><html><head><meta charset=\"utf-8\"/><title> %s </title>
<link rel=\"stylesheet\" type=\"text/css\" href=\"style.css\"/></head><body>"
  end = "</body></html>"
  fmt = "<li> <a href='%s'> %s </a></li>"
}
```

Nesta imagem vemos o início do programa, os separadores de campo para a filtragem do *xml* e a construção de

algum código *html* que irá ser invocada durante o programa.

- Para cada uma das linhas, é testado quando o campo corresponde ao nome da respetiva *tag* do *xml*. Quando este campo corresponde ao que é testado, como vemos no excerto exemplo do código abaixo, é guardada a informação pretendida, para ser usada mais tarde para a apresentação de resultados.

```
$2 ~ /OPERADOR/ && horaEntrada != "" && dataEntrada != "null" {  
  if(entrada == "") entrada = "Desconhecido"  
  ent[idx] = entrada  
  saidas[idx] = saida  
  operadores[idx] = $3  
  precos[ent[idx] saidas[idx] operadores[idx]] += preco  
  relationString[ent[idx] saidas[idx] operadores[idx]]++  
  infoEntradas[idx] = dataEntrada "''" entrada "''" saida "''" $3 "''" horaEntrada  
  idx++  
}
```

No exemplo da imagem vemos o caso em que o campo pretendido, faz *match* com a string atual. Neste caso, se o campo nº 2 for igual a "OPERADOR" é guardado onde a matrícula entrou e saiu para passar naquele operador, o próprio operador, entre outros.

- No fim de reunir toda a informação necessária, esta é preparada para ser apresentada em *html*. Como os resultados apresentados são vários, decidiu-se implementar diferentes funções para construir os ficheiros *html* para a consulta dos dados. Cada uma destas funções acede aos valores guardados anteriormente como no ponto acima é explicitado.

```
function printEntradasHTML() {  
  printf (enc, "Entradas") > "entradas.html"  
  print "<h2>Número de Entradas no Mês " mes "</h2>" > "entradas.html"  
  print "<table style=\"width:30%\">" > "entradas.html"  
  print "<tr><th><b>Data</b></th> <th><b>Nº de Entradas</b></th></tr>" > "entradas.html"  
  
  asorti(tempEntradas)  
  for(i in tempEntradas) {  
    split(tempEntradas[i], n2, "-")  
    t = n2[3] "-" n2[2] "-" n2[1]  
  
    tot += entradas[t]  
    print "<tr><td>" t "</td> <td><a href=\" t \".html\"> entradas[t] "</a></td></tr>" > "entradas.html"  
  }  
  
  print "<tr><td><b>Total</b></td> <td><b><a href= totVa\" matricula \".html\"> tot "</a></b></td></tr></table>" > "entradas.html"  
  print end > "entradas.html"  
}
```

Esta função, por exemplo, tem o objectivo de criar um ficheiro **html** com datas e com frequência que o cliente utilizou o serviço da Via Verde nesse específico dia.

- Todas as funções são chamadas no bloco END, que é executado no fim de tudo o resto, garantindo, que quando há acesso a informação, esta, está já completa e pronta a ser apresentada. São criados então ficheiros *html* que são ligados entre si para facilitar a consultar do utilizador.

```

END {
  for(i in entradas) {
    split(i, n, "-")
    t = n[3] "-" n[2] "-" n[1]
    tempEntradas[t] = entradas[i]
  }

  printInfoData()
  printInfoMatriculaHTML()
  printEntradasHTML()
  printTotalHTML()
  printRelationsHTML()
  printInfoSaidas()
  printIndexHTML()

  printEntradas()
  print "Ficheiros HTML criados com sucesso! (Abrir index.html)"
}

```

Nesta imagem são chamadas todas as função do género já apresentado, que geram os ficheiros *html* com a informação pretendida, tornando apelativa a consulta.

Capítulo 4

Codificação e Testes

4.1 Alternativas, Decisões e Problemas de Implementação

4.1.1 Alternativas

Para uma fase inicial, o nosso grupo tinha tomado a decisão de apresentar o resultado do processamento dos dados em *xml* no terminal. O utilizador apenas tinha de executar o *script* (processador de texto) em *gawk*, fornecendo o devido ficheiro em *xml* e toda a informação estaria imediatamente visível. Este método tornava-se muito eficaz quando apenas pretendíamos apresentar os resultados para as questões explicitadas pelo enunciado deste trabalho prático.

Contudo, à medida que o nosso grupo foi integrando respostas a cada vez mais questões, deixou de fazer sentido continuar a apresentar todas as informações no terminal, sendo que este estava a ficar sobrecarregado de texto, não sendo apelativo a qualquer utilizador que o pretenda utilizar. Além disto, caso se desejasse saber uma informação específica, era necessário procura-la em todo o texto apresentado no ecrã, o que se tornava pouco prático e ineficiente.

4.1.2 Decisões

Dada a implementação de mais questões, além daquelas enunciadas, decidiu-se fazer a migração da apresentação de todas as informações de um formato texto apresentado em terminal para um formato mais apelativo apresentado, desta vez, em ficheiros *html*.

Esta migração para um formato em *html* mostrou-se razoavelmente simples de se realizar, visto que apenas foi necessário redirecionar as informações para os devidos ficheiros. Para que seja possível ter o formato de apresentação em *html* totalmente implementado, decidimos, ainda criar vários ficheiros, dividindo todas as informações por categorias facilmente identificáveis pelos utilizadores. Garantimos, posteriormente, que todas as hiperligações entre ficheiros estavam de acordo com as nossas preferências.

Numa terceira fase, decidimos implementar uma interface com base em tabelas em cada um dos ficheiros com dados relevantes, para que seja possível fazer uma extração quase imediata por qualquer tipo de informação.

Finalmente, optámos por dar ao utilizador a possibilidade de visualizar o percurso que referente à cada uma das suas viagens, para facilitar, ainda mais, todo o processo de gestão das suas viagens realizadas.

4.1.3 Problemas de Implementação

O nosso grupo não se deparou com nenhum problema grave a nível de implementação do código. Planeamos atempadamente todas as informações que seriam possíveis de obter através da geração dos ficheiros *html* precisamente para evitar qualquer tipo de infortúnios.

O primeiro problema, de fácil resolução, era a codificação do ficheiro *xml*, que por algum motivo não aceitava caracteres *utf-8*.

Tivemos ainda outro pequeno entrave na fase inicial, quando ainda apresentávamos os dados no ecrã. Este problema

era apenas de apresentação, não afetando a veracidade dos dados. Tudo surgiu porque quando decidimos apresentar o nome da operadora de um dado percurso, existiam evidentemente nomes diferentes e por isso, agora numa perspectiva de programação, com comprimentos de *string* diferentes. Isto fez com que as informações não se encontrassem alinhadas umas com as outras, sendo difícil perceber a que campo nos referíamos, que podiam ser os nomes de uma entrada, de uma saída, ou das entidades operadoras.

4.2 Testes realizados e Resultados

Numa primeira fase, imprimíamos no ecrã as informações do cliente.

| Total gasto em parques | 6.35€ |
|--------------------------|----------------|
| Total gasto em portagens | 71.05€ |
| Total gasto no mês | 77.4€ |
| | |
| Dia | Nº de entradas |
| 30-07-2015 | 3 |
| 18-08-2015 | 2 |
| 29-07-2015 | 2 |
| 11-08-2015 | 2 |
| 10-08-2015 | 7 |
| 26-07-2015 | 3 |
| 17-08-2015 | 4 |
| 06-08-2015 | 4 |
| 13-08-2015 | 5 |
| 21-08-2015 | 4 |
| 31-07-2015 | 1 |
| Total: | 37 |

De seguida, decidimos que seria possível o utilizador obter mais informações, caso desejasse.

Extracto do Mês Ago-2015 da Matricula 00-LJ-11

- [Informações da Matrícula.](#)
- [Número de entradas por dia.](#)
- [Total pago.](#)
- [Dados gerais.](#)

Após possibilitar a pesquisa de mais informações, decidiu-se tornar o ambiente gráfico na apresentação de resultados mais agradável. Introduzimos também a possibilidade do utilizador visualizar o percurso a que nos referíamos.


Informações Gerais no Mês Ago-2015

| Entrada | Saída | Operador | Frequência | Gasto | Percurso |
|--------------|----------------|---------------------------------------|------------|-------|----------------------------|
| Aeroporto | Ponte Pedra | I. de Portugal (P3) | 2 | 1.3€ | Visualizar |
| Angeiras S-N | Povoa S-N | I. de Portugal (N1) | 5 | 10€ | Visualizar |
| Braga Sul | Maia II | Brisa Concessão Rodoviária (BR) | 1 | 3.25€ | Visualizar |
| Braga Sul | Maia PV | Brisa Concessão Rodoviária (BR) | 1 | 3.25€ | Visualizar |
| EN 205 PV | Ferreiros | I. de Portugal (E1) | 6 | 15.6€ | Nao Disponivel |
| EN13 O/E | EN107 | I. de Portugal (P3) | 1 | 0.8€ | Nao Disponivel |
| EN14 E/O | Lipor | I. de Portugal (P3) | 1 | 0.8€ | Nao Disponivel |
| Ermesinde PV | Valongo | Brisa Concessão Rodoviária (BR) | 1 | 0.35€ | Visualizar |
| Ferreiros | EN 205 PV | I. de Portugal (E1) | 6 | 15.6€ | Nao Disponivel |
| Ferreiros | PQ Av. Central | BRAGAPARQUES (BP) | 1 | 2.6€ | Visualizar |
| Ferreiros | Neiva S-N | I. de Portugal (N1) | 1 | 0.65€ | Visualizar |
| Ferreiros | PQ A Sa Cam.I | ANA - Aeroportos de Portugal. SA (AP) | 1 | 3.75€ | Visualizar |
| Ferreiros | Neiva N-S | I. de Portugal (N1) | 1 | 0.65€ | Visualizar |
| Maia II | Braga Sul | Brisa Concessão Rodoviária (BR) | 1 | 3.25€ | Visualizar |

Finalmente, decidimos tornar o website mais apelativo, tornando o ambiente mais navegável.

Extracto do Mês Ago-2015 da Matricula 00-LJ-11

[Informações da Matrícula](#)
[Número de entradas por dia](#)
[Total Pago](#)
[Dados gerais](#)
[Locais de Saída](#)



Informações Gerais no Mês Ago-2015

| Entrada | Saída | Operador | Frequência | Gasto | Percurso |
|--------------|----------------|---------------------------------------|------------|-------|----------------------------|
| Aeroporto | Ponte Pedra | I. de Portugal (P3) | 2 | 1.3€ | Visualizar |
| Angeiras S-N | Povoa S-N | I. de Portugal (N1) | 5 | 10€ | Visualizar |
| Braga Sul | Maia PV | Brisa Concessão Rodoviária (BR) | 1 | 3.25€ | Visualizar |
| Braga Sul | Maia II | Brisa Concessão Rodoviária (BR) | 1 | 3.25€ | Visualizar |
| EN 205 PV | Ferreiros | I. de Portugal (E1) | 6 | 15.6€ | Nao Disponivel |
| EN13 O/E | EN107 | I. de Portugal (P3) | 1 | 0.8€ | Nao Disponivel |
| EN14 E/O | Lipor | I. de Portugal (P3) | 1 | 0.8€ | Nao Disponivel |
| Ermesinde PV | Valongo | Brisa Concessão Rodoviária (BR) | 1 | 0.35€ | Visualizar |
| Ferreiros | PQ Av. Central | BRAGAPARQUES (BP) | 1 | 2.6€ | Visualizar |
| Ferreiros | EN 205 PV | I. de Portugal (E1) | 6 | 15.6€ | Nao Disponivel |
| Ferreiros | PQ A Sa Cam.I | ANA - Aeroportos de Portugal. SA (AP) | 1 | 3.75€ | Visualizar |
| Maia II | Braga Sul | Brisa Concessão Rodoviária (BR) | 1 | 3.25€ | Visualizar |
| Maia PV | Braga Sul | Brisa Concessão Rodoviária (BR) | 1 | 3.25€ | Visualizar |

Capítulo 5

Conclusão

Este trabalho integra os conhecimentos adquiridos na disciplina de Processamento de Linguagens, relacionados com o uso de expressões regulares e comando *gawk*.

Conseguimos desenvolver um *script* que fosse simples e capaz de fornecer ao utilizador, todo tipo de informação disponível no ficheiro *xml*. Enaltecemos o nosso agrado relativamente à forma como apresentamos os dados ao utilizador.

Futuramente seria abordada a possibilidade de povoamento do ficheiro *xml*, integrando mais clientes e mais informação no geral.