Universidade do Minho

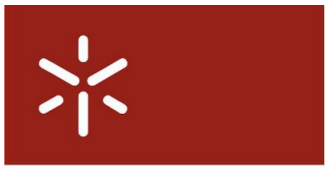
Mestrado Integrado em Engenharia Informática

Processamento de Linguagens

Processador de Via Verde

Autores:
A75625 André Almeida Gonçalves
A74576 José António Dantas Silva
A75315 Ricardo Jorge Barroso Certo

13 de Março de 2017



Universidade do Minho

Resumo

Um ficheiro XML, embora eficiente e desenvolvido para guardar informação de forma a permitir uma leitura globalizada, não tem vantagens na apresentação que dispõe da informação. O relatório e trabalho desenvolvidos demonstram uma solução para este problema, utilizando a matéria lecionada nas aulas, processando a informação contida nas tags XML e procedendo à sua apresentação no formato HTML, permitindo uma interface agradável ao utilizador comum.

Conteúdo

1 Introdução

A realização deste primeiro trabalho prático da unidade curricular de Processamento de Linguagens consiste no desenvolvimento de um *Filtro de Texto GAWK*, através do uso de expressões regulares. Para isso utilizamos os conteúdos lecionados nas aulas, tendo como objetivos com a elaboração deste trabalho aperfeiçoar a nossa capacidade de escrita das Expressões Regulares (ER) e aprender a utilizar o sistema de produção para a filtragem de texto GAWK.

Ao longo deste relatório vamos apresentar todas as etapas e decisões tomadas durante todo o processo de elaboração do filtro de texto GAWK que desenvolvemos, iniciando com uma breve descrição do problema e na parte dos anexos, será colocado o código fonte dos exercícios desenvolvidos.

2 Análise e Especificação

2.1 Descrição do Problema

Dos enunciados propostos a escolha por parte do grupo recaiu sobre o **Processador de Transações Via Verde**, pois era um exercício interessante, de aplicação no mundo real que permitia uma manipulação dos dados para variadas extrações estatísticas. Neste enunciado eram requeridas algumas funcionalidades, que passam pela apresentação de diversas informações tais como o número de entradas registadas para cada dia, dados estes possíveis de extrair através do extrato mensal de um Cliente na Via Verde no formato de um ficheiro XML, sendo para tal necessária a interpretação, filtragem e tratamento do ficheiro de input. Para além destes requisitos mínimos o grupo, devido à variedade de informação inerente ao ficheiro decidiu produzir mais *queries* tais como um registo dos percursos e as respetivas passagens no mesmo, tendo tais dados sido apresentados em **html** para uma mais fácil visualização por parte do utilizador.

2.2 Enunciado

A Via Verde envia a cada um dos seus utentes um extrato mensal no formato XML como se exemplifica no ficheiro anexo viaverde.xml. Depois de analisar com cuidado o formato desse ficheiro anexo, pretende-se que desenvolva um Processador de Texto com o GAWK para ler um extrato mensal da Via Verde e:

- **a)** calcular o número de 'entradas' em cada dia do mês
- **b)** escrever a lista de locais de saída
- **c)** calcular o total gasto no mês
- **d)** calcular o total no mês apenas em 'parques'.

2.3 Dados a tratar

Após a analise do ficheiro XML do qual será extraída a informação verificamos dois campos gerais e distintos que contêm a informação necessária para a resolução das *Queries* propostas, os campos **CLIENTE** e **TRANSACCAO**.

2.3.1 Dados relativos ao cliente

O campo **CLIENTE** mantém registo das informações do cliente ao qual a fatura se refere, tornando a identidade do mesmo única através dos seus campos de informação.

```
<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>
```

2.3.2 Dados relativos à transação

O campo **TRANSACCAO**, permite manter um registo completo das informações importantes sobre um "percurso"entre dois postos Via Verde, bem como das permanências nos parques de estacionamento.

```
<TRANSACCAO>
<DATA_ENTRADA>26-07-2015</DATA_ENTRADA>
<HORA_ENTRADA>23:26</HORA_ENTRADA>
<ENTRADA>Angeiras S-N</ENTRADA>
<DATA_SAIDA>26-07-2015</DATA_SAIDA>
<HORA_SAIDA>23:37</HORA_SAIDA>
<SAIDA>Povoa S-N</SAIDA>
<IMPORTANCIA>2,00</IMPORTANCIA>
<VALOR_DESCONTO>0,00</VALOR_DESCONTO>
<TAXA_IVA>23</TAXA_IVA>
<OPERADOR>I. de Portugal (N1)</OPERADOR>
<TIPO>Portagens</TIPO>
<DATA_DEBITO>05-08-2015</DATA_DEBITO>
<CARTAO>6749036</CARTAO>
</TRANSACCAO>
```

3 Desenho e Implementação da Solução

Apesar do ficheiro XML fornecido para o exercício proposto ter a vantagem de cada *tag* ser tratada numa única linha e estarem todas devidamente alinhadas na esquerda, o grupo decidiu optar por uma solução capaz de lidar com estas vantagens como não estando consideradas. Para tal foi instrumental o uso da função ***match***, função que permite recebendo uma string e uma expressão regular colocar num array os campos pretendidos explicitados entre parênteses na expressão regular. Sendo os dados que pretendíamos obtidos maioritariamente através do campo geral **transaccao**, foram mantidos registos desses mesmos campos para cada transaccao efetuada, sendo guardados todos sob a forma de arrays cuja variavel em comum era a transaccao em questão, incrementada quando a tag de fim de transação era encontrada.

```
if(match($0,/<\/TRANSACCAO>/))
    var++;
```

3.1 Número de Entradas em cada dia do mês

Nesta querie eram pedidas o numero de entradas para cada dia do mês, ou seja, teria de ser verificada para cada transação a data da sua entrada, dados contidos na tag **DATA_ENTRADA**. Como tal usando a função match com a expressão regular apresentada de seguida conseguimos obter o valor entre as duas tags, neste caso a data. Criando um array em que as datas são os índices evitando a repetição dos mesmos e o valor é a soma de uma variavel que começa a 0 por cada ocorrência da dada data, obtemos assim o numero de entradas para cada dia do mês.

```
if(match($0,/<DATA_ENTRADA>(.*)<\/DATA_ENTRADA>/, d)) {
    if(d[1] != "null")
        dias[reordena(d[1])]++;

    split(d[1],a,"-");
    meses[var] = a[2];
}
```

Esta expressão regular define que entre as tags pretendidas queremos registar no array a ocorrência de qualquer caracter 0 ou mais vezes como verificamos com **(.*)**.

3.2 Lista de locais de saída

Seguindo o mesmo raciocínio da alínea anterior e sabendo que na tag **SAIDA** se encontra o local de saída do percurso, facilmente conseguimos obter um array com todos os locais de saida através da seguinte chamada da função match.

```
    if(match($0,/<SAIDA>(.*)<\/SAIDA>/, l)) {
        locais[l[1]]++;
```

3.3 Total gasto no mês

Para o total gasto no mês o grupo considerou o mês não apenas o referente à mensalidade em si, tendo também efetuado a separação por mês como o mesmo é disposto na data. Como tal a partir da data de entrada registamos para cada mês transação o mês a que esta se refere, bem como o valor pago, alcançado no campo **IMPORTANCIA**. Para que contas sobre floats fossem possiveis de efetuar nos valores contidos nesse campo foi criada a seguinte função:

```
function normaliza_float(num) {
    gsub(/,/, ".",num)
    return num;
}
```

No fim de todas as transações controladas foi apenas necessário adicionar e posteriormente percorrer um array no qual a soma dos valores da importância eram adicionados num array cujo índice era o mês da mesma.

```
    if(match($0,/<IMPORTANCIA>(.*)<\/IMPORTANCIA>/, v))
        precos[var] = normaliza_float(v[1]);
```

```
    for (i = 0;i < var; i++) {
        total[meses[i]] += precos[i];
```

3.4 Total gasto no mês em parques

Para esta ultima alínea, os dados da anterior praticamente permitiam chegar ao resultado, faltando apenas o campo que diferenciava as transações, o campo **TIPO**. Foi então para cada transação registado o tipo de transação efetuada, sendo apenas necessário no fim percorrer as transações e efetuar a soma da importância daquelas cujo tipo era **'parque de estacionamento'**.

```
if(match($0,/<TIPO>(.*)<\/TIPO>/, t))
    tipo[var] = t[1];
```

```
for (i = 0;i < var; i++) {
    total[meses[i]] += precos[i];

    if(tipo[i] == "Parques de estacionamento")
        parques += precos[i];
```

3.5 Registo das viagens mais efetuadas

Como foi referido o grupo decidiu que uma estatística interessante a nível real seria a verificação dos percursos mais efetuados num dado mês. Tal foi obtido guardando registo do local de entrada e saída de cada transação, permitindo posteriormente criar um array em que para cada transação teria como índice a entrada seguida de - "seguida do local de saída, cujo valor seria incrementado por cada ocorrência do mesmo.

```
    if(length(entrada[i]) > 0 && length(saida[i]) > 0)
        viagens[entrada[i]" - "saida[i]]++;
```

4 Exemplos de utilização

Como foi referido o grupo decidiu que a apresentação dos dados obtidos seria mais agradável se apresentada num browser através de um ficheiro html. Como tal foram obtidos os seguintes resultados, utilizando o ficheiro disponibilizado pelo professor.

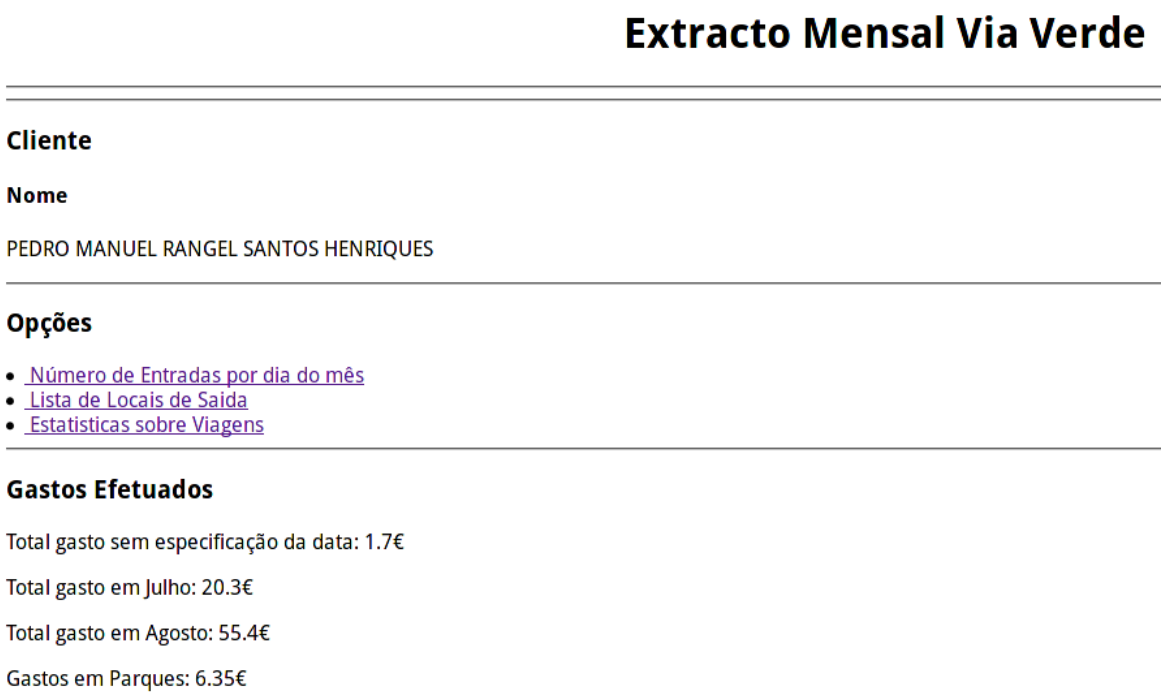


Figura 1: Página principal

Lista de Entradas por dia

Entradas de cada dia

Dia	Nr de Entradas
26 de Julho de 2015	3
29 de Julho de 2015	2
30 de Julho de 2015	3
31 de Julho de 2015	1
06 de Agosto de 2015	4
10 de Agosto de 2015	7
11 de Agosto de 2015	2
13 de Agosto de 2015	5
17 de Agosto de 2015	4
18 de Agosto de 2015	2
21 de Agosto de 2015	4

Figura 2: Entradas diárias

Lista de Locais de Saida

Locais de Saida

Local
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
PQ A Sa Carn.I
PQ Av. Central
Ponte Pedra
Povoa S-N

Figura 3: Locais de Saida

Estatisticas sobre viagens

Viagens efetuadas no mês por percurso

Percurso	Nr de Viagens
Aeroporto - Ponte Pedra	2
Angeiras S-N - Povoa S-N	5
Braga Sul - Maia II	1
Braga Sul - Maia PV	1
EN 205 PV - Ferreiros	6
EN13 O/E - EN107	1
EN14 E/O - Lipor	1
Ermesinde PV - Valongo	1
Ferreiros - EN 205 PV	6
Maia II - Braga Sul	1
Maia PV - Braga Sul	1
Povoa N-S - Angeiras N-S	5
Valongo - Ermesinde PV	1
Via Norte E/O - Custoias	1
Via Norte E/O - Freixieiro	1
Via Norte O/E - Ponte Pedra	1

Figura 4: Percursos efetuados

5 Conclusão

Após descrito todo o processo de desenvolvimento deste trabalho prático, desde a descrição do problema em causa até á implementação da solução em *HTML* passando pela análise de dados resta agora apresentar uma breve conclusão sobre todo o processo.

A realização deste projeto desempenhou um papel importante para consolidarmos a matéria lecionada nas aulas apesar do grau de dificuldade não ter sido elevado. As técnicas de utilização de expressões regulares aí aprendidas e aperfeiçoadas ao longo deste trabalho, serviram para resolver de uma forma bastante prática e simplista os enunciados propostos mostrando-se assim como uma ferramenta poderosa. Para além do uso das ER's foi também possível ao grupo melhorar e aprofundar os seus conhecimentos sobre *HTML*, que tornou a experiência ainda mais enriquecedora e permitiu uma visualização dos resultados mais cómoda para a vista.

A Via Verde

A.1 Código do programa

A.1.1 Geração HMTL

```
#!/usr/bin/gawk -f

@include "via_verde.gawk"

BEGIN {
    html_start = "<html> <head> <meta charset='UTF-8'/> <style>
        table, th, td { border: 1px solid black; border-collapse:
        collapse; } th, td { padding: 5px; } th { text-align: left;
        } </style> </head> <body>"
    html_end = "</body>\n </html>"
    table_locais_start = "<table style=\"width:20%\">
        <caption>Locais de Saida</caption> <tr> <th>Local</th> </tr>"
    table_entradas_start = "<table style=\"width:30%\">
        <caption>Entradas de cada dia</caption> <tr> <th>Dia</th>
        <th>Nr de Entradas</th> </tr>"
    table_viagens_start = "<table style=\"width:30%\">
        <caption>Viagens efetuadas no m[U+FFFD]s por
        percurso</caption> <tr> <th>Percurso</th> <th>Nr de
        Viagens</th> </tr>"
    table_end = "</table>\n"
    h1 = "<h1> %s </h1> <hr>\n"
    h3 = "<hr> <h3> %s </h3>\n\n"
    h4 = "<h4> %s </h4>\n"
    par = "<p> %s </p>"
    link = "<li> <a href='%s'> %s </a></li>\n"
}

END {
    print html_start > "index.html"
    print "<h1 align=\"center\"> Extracto Mensal Via Verde </h1>
        <hr> " > "index.html"
    printf(h3,"Cliente") > "index.html"
    printf(h4,"Nome") > "index.html"
    printf(par,nome) > "index.html"
    printf(h3,"Op[U+FFFD][U+FFFD]es") > "index.html"
    printf(link,"entradas.html","N[U+FFFD]mero de Entradas por dia
        do m[U+FFFD]s") > "index.html"
    printf(link,"locais.html","Lista de Locais de Saida") >
        "index.html"
```

```
    printf(link,"viagens.html","Estatisticas sobre Viagens") >
        "index.html"
    printf(h3,"Gastos Efetuados") > "index.html"
    for(p in total) {
        if(length(p) > 0)
            print "<p> Total gasto em " get_mes(p) ": " total[p]
                "[U+FFFD] </p> " > "index.html"
        else print "<p> Total gasto sem especifica[U+FFFD][U+FFFD]o
            da data: " total[p] "[U+FFFD] </p>" > "index.html"
    }
    print "<p> Gastos em Parques: " parques "[U+FFFD] </p>" >
        "index.html"
    print html_end > "index.html"


    print html_start > "entradas.html"
    printf (h1,"Lista de Entradas por dia") > "entradas.html"
    print table_entradas_start > "entradas.html"
    for(e in dias)
        print "<tr> <td> " controi(e) " </td> <td> " dias[e] "</td>
            </tr>" > "entradas.html"
    print table_end > "entradas.html"
    print html_end > "entradas.html"


    print html_start > "locais.html"
    printf (h1,"Lista de Locais de Saida") > "locais.html"
    print table_locais_start > "locais.html"
    for(i in locais)
        print "<tr> <td>" i "</td> </tr>" > "locais.html"
    print table_end > "locais.html"
    print html_end > "locais.html"


    print html_start > "viagens.html"
    printf (h1,"Estatisticas sobre viagens") > "viagens.html"
    print table_viagens_start > "viagens.html"
    for(viagem in viagens)
        print "<tr> <td>" viagem "</td> <td>" viagens[viagem] "</td>
            </tr>" > "viagens.html"
    print table_end > "viagens.html"
    print html_end > "viagens.html"

}
```

A.1.2 Gerador de dados

```
#!/usr/bin/gawk -f

@include "functions.gawk"

BEGIN {
    PROCINFO["sorted_in"]= "@ind_str_asc";
    var = 0;
}

{
    if(match($0,/<NOME>(.*)<\/NOME>/, n))
        nome = n[1];

    if(match($0,/<ENTRADA>(.*)<\/ENTRADA>/, en))
        entrada[var] = en[1];

    if(match($0,/<SAIDA>(.*)<\/SAIDA>/, l)) {
        locais[l[1]]++;
        saida[var] = l[1];
    }

    if(match($0,/<DATA_ENTRADA>(.*)<\/DATA_ENTRADA>/, d)) {
        if(d[1] != "null")
            dias[reordena(d[1])]++;

        split(d[1],a,"-");
        meses[var] = a[2];
    }

    if(match($0,/<IMPORTANCIA>(.*)<\/IMPORTANCIA>/, v))
        precos[var] = normaliza_float(v[1]);

    if(match($0,/<TIPO>(.*)<\/TIPO>/, t))
        tipo[var] = t[1];

    if(match($0,/<\/TRANSACCAO>/))
        var++;
}


END {

    for (i = 0;i < var; i++) {
```

```
        total[meses[i]] += precos[i];

        if(tipo[i] == "Parques de estacionamento")
            parques += precos[i];

        if(length(entrada[i]) > 0 && length(saida[i]) > 0)
            viagens[entrada[i]" - "saida[i]]++;
    }

}
```

A.1.3 Funções auxiliares

```
#!/usr/bin/gawk -f

function normaliza_float(num) {
    gsub(/,/, ".",num)
    return num;
}

function get_mes(valor) {
    mes["01"] = "Janeiro";
    mes["02"] = "Fevereiro";
    mes["03"] = "Mar[U+FFFD]o";
    mes["04"] = "Abril";
    mes["05"] = "Maio";
    mes["06"] = "Junho";
    mes["07"] = "Julho";
    mes["08"] = "Agosto";
    mes["09"] = "Setembro";
    mes["10"] = "Outubro";
    mes["11"] = "Novembro";
    mes["12"] = "Dezembro";
    return mes[valor];
}

function controi(data) {
    split(data,a,"-");
    return a[3] " de " get_mes(a[2]) " de " a[1];
}

function reordena(data) {
    split(data,a,"-");
    return a[3] "-" a[2] "-" a[1]
```

```
}
```

B Autores Musicais

O grupo decidiu ainda realizar o exercicio 3 sobre os Autores Musicais como maneira de aprofundar ainda mais os seus conhecimentos em Gawk. Deixamos de seguida o código fonte criado bem como alguns prints do resultado obtido.

B.0.1 Ficheiro Gawk

```
#!/usr/bin/gawk -f

BEGIN {
   FS = ":"
   html_start = "<html> <head> <meta charset='UTF-8'/> <link
       rel=\"stylesheet\"
       href=\"https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css\"
       integrity=\"sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u\"
       crossorigin=\"anonymous\"> <style> table, th, td { border:
       1px solid black; border-collapse: collapse; } th, td {
       padding: 5px; } th { text-align: left; } body{
       text-align:center; font-size: 40px; margin-top:20px;}</style>
       </head> <body>"
    html_end = "</body>\n </html>"
}

{
   if($1 == "singer"){
   x = split($2,a,";")
   cant[a[1]]++
   }
}

{
   if($1 == "author"){
   x = split($2,b,";")
   autor[b[1]]++
   }
}

/^title/{
```

```
   title = $2;
}

{
   if($1 == "author"){
      if(listaTit[$2] == null) {
         listaTit[$2] = title;
      }
      else {
         listaTit[$2] = listaTit[$2] " , " title;
      }
   }

}



END {

   #HTML MAIN


   print html_start > "index.html"
   print "<h1> Projeto 3: Autores Musicais<h1>" > "index.html"
   print "<h2> Processamento de Linguagens</h2>" > "index.html"
   print "<div style=\" margin-top:20px; margin-bottom:20px;
       text-align:justify;\" class=\"col-md-8
       col-md-offset-2\"><h4>Al[U+FFFD]m da cole[U+FFFD][U+FFFD]o de
       entrevista e fotografias do npMP, o Professor Jos[U+FFFD]
       Jo[U+FFFD]o Almeida tem uma diretoria (de nome musica, que
       [U+FFFD] anexada em formato ZIP) com dezenas de ficheiros de
       extens[U+FFFD]o [U+FFFD].lyr[U+FFFD] que cont[U+FFFD]m a
       letra de can[U+FFFD][U+FFFD]es famosas precedidas de 2 ou
       mais campos de meta-informa[U+FFFD][U+FFFD]o (1 por linha)
       com o t[U+FFFD]tulo da can[U+FFFD][U+FFFD]o, o autor da letra
       (pode ser 1 ou mais pessoas), o cantor, etc. Uma linha em
       branco separa a meta-informa[U+FFFD][U+FFFD]o da letra.
       Podendo ainda ter em alguns casos um terceiro bloco
       (igualmente separado da letra por uma linha em branco) com a
       m[U+FFFD]sica escrita na nota[U+FFFD][U+FFFD]o midi.</br>
       Depois de analisar com cuidado o formato desse ficheiro
       anexo, pretende-se que desenvolva um Processador de Texto com
       o GAWK para ler todos os ficheiros [U+FFFD].lyr[U+FFFD] da
       diretoria musica e: </h4></div>" > "index.html"
```

```
print "<div class=\"col-md-4\">" > "index.html"
print "<h1><a href=\"indexA.html\"> Alinea A </a></h1>" >
    "index.html"
print "<h3> Calcular o total de cantores e a lista com seus
    nomes</h3>" > "index.html"
print "</div> <div class=\"col-md-4\">" > "index.html"
print "<a href=\"indexB.html\"> Alinea B </a>" > "index.html"
print "</h1><h3>Calcular o total de can[U+FFFD][U+FFFD]es do
    mesmo autor</h3></h1>" > "index.html"
print "</div> <div class=\"col-md-4\">" > "index.html"
print "<h1><a href=\"indexC.html\"> Alinea C </a></h1>" >
    "index.html"
print "<h3> Escrever o nome de cada autor seguido do
    t[U+FFFD]tulo das suas can[U+FFFD][U+FFFD]es</h3>" >
    "index.html"
print "</div>" > "index.html"
print html_end > "index.html"



#Alinea A HTML

print html_start > "indexA.html"
print "<div style=\"text-align:left; margin-left:20px;\"<h5><a
    href=\"index.html\"> Voltar </a> </h5></div> " > "indexA.html"
print "<h1> Al[U+FFFD]nea A) </h1>" > "indexA.html"
print "<h2> Calcular o total de cantores e a lista com seus
    nomes.</h2>" > "indexA.html"
print "<h3>Total: <b>"length(cant) "</b></h3>" > "indexA.html"
for (i in cant) print "<div class=\"col-md-3\" style=\" height:
    70px; margin-top: 20px; \"><h3>" i "</h3></div>" >
    "indexA.html"
print html_end > "indexA.html"


#Alinea B HTML

print html_start > "indexB.html"
n=asorti(autor, sorted)
print "<div style=\"text-align:left; margin-left:20px;\"<h5><a
    href=\"index.html\"> Voltar </a> </h5></div> " > "indexB.html"
print "<h1> Al[U+FFFD]ena B) </h1>" > "indexB.html"
print "<h2> Calcular o total de can[U+FFFD][U+FFFD]es de cada
    autor autor. </h2>" > "indexB.html"
```

```
    print "<ul>" > "indexB.html"
    for (i=1; i<=n; i++) {
        print "<div class=\"col-md-3\" style=\" height: 70px;
            margin-top: 20px; \"><h3><b>" sorted[i] "</b>:"
            autor[sorted[i]] "</h3></div>" > "indexB.html"
    }
    print "</ul>" > "indexB.html"
    print "</div>" > "indexB.html"
    print html_end > "indexB.html"


    #Alinea C HTML

    print html_start > "indexC.html"
    print "<div style=\"text-align:left; margin-left:20px;\"<h5><a
        href=\"index.html\"> Voltar </a> </h5></div> " > "indexC.html"
    print "<h1> Alinea C) </h1>" > "indexC.html"
    print "<h2 style=\"margin-bottom:50px;\"> Escrever o nome de
        cada autor seguido das suas can[U+FFFD][U+FFFD]es. </h2>" >
        "indexC.html"
    print "<div class=\"col-md-8 col-md-offset-2\"> <table
        style=\"margin-bottom: 40px;\">" > "indexC.html"
       for (i=1; i<=n; i++) {
          print "<tr>" > "indexC.html"
          print "<td><h4 style=\"text-align:center;\"><b>" sorted[i]
              "</b></h4> </td>" > "indexC.html"
          print "<td style=\"padding:20px;\">" listaTit[sorted[i]]
              "</td>" > "indexC.html"
          print "</tr>" > "index.html"
    }

    print "</table></div>" > "indexC.html"
    print html_end > "indexC.html"
}
```

B.0.2 Resultado Obtido

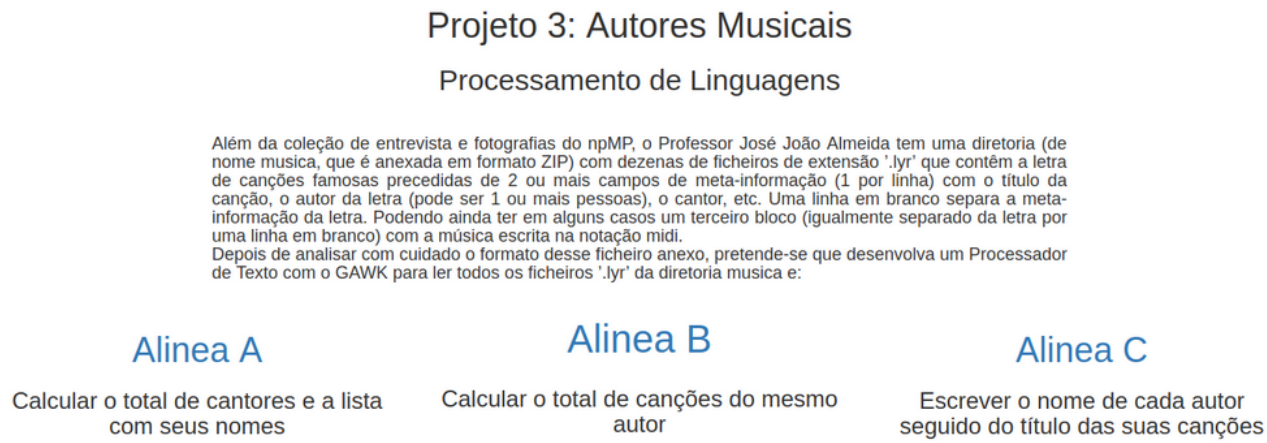


Figura 5: Menu principal

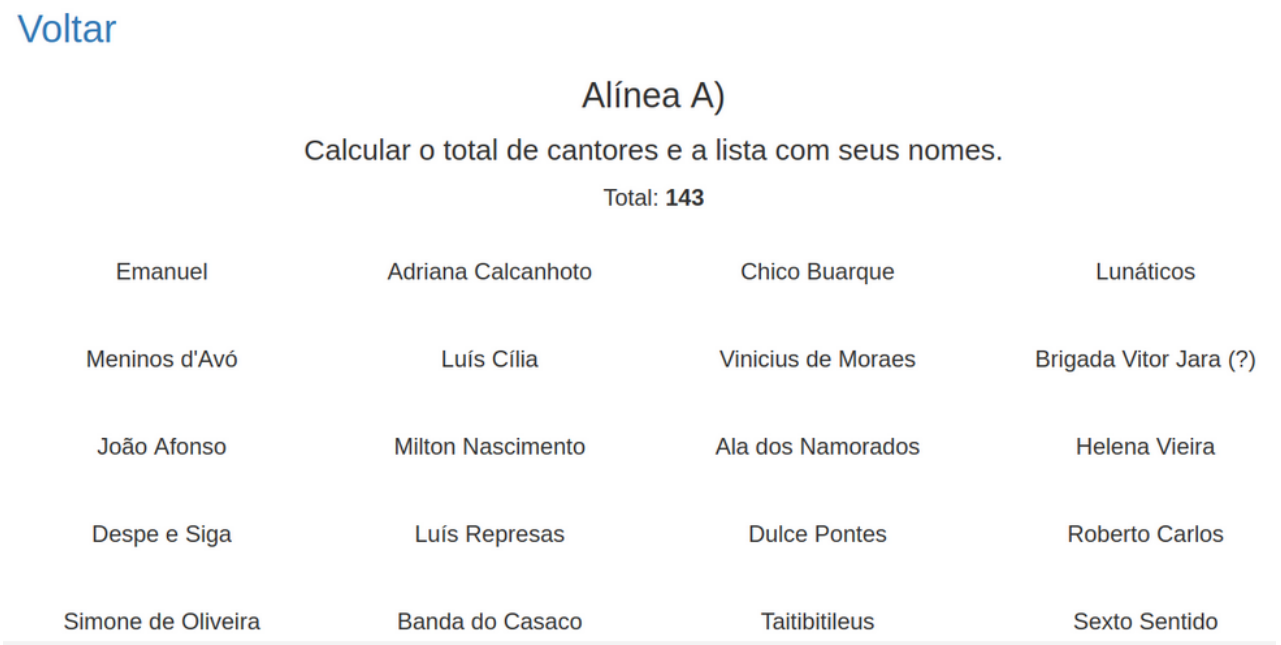


Figura 6: Alínea A

Voltar

Alíena B)

Calcular o total de canções de cada autor autor.

?:36	**?** :4	**A. Amargo**:1	**A. P. Braga**:1
Alberto Janes:1	**Alexandre**:1	**Alfredo Duarte**:1	**Alves Coelho**:1
Amadeu do Vale:1	**António Luís Melo**:1	**António Macedo**:1	**António Mafra**:2
António Variações:3	**Aníbal Nazaré**:2	**Arlindo Duarte Carvalho**:1	**Arnaldo Antunes** :1
Artur Ribeiro:1	**Ary Barroso**:2	**Augusto Hilário**:1	**Belchior**:1

Figura 7: Alínea B

Voltar

Alinea C)

Escrever o nome de cada autor seguido das suas canções.

?	* Não sei se mereço , * De 2.ª a 6.ª Feira , * Filhos da Nação , Modinha , Adeus Sé velha , * À meia noite ao luar , A minha capa rasgada , * Ao cair da tarde , Ao morrer os olhos dizem , Balada da despedida (2) , * Balada da despedida , Balada do Mondego , * Coimbra , * Coimbra menina e moça , Esmeralda verde , * Feiticeira , * O meu desejo , O meu menino , O sol anda lá no céu , Os teus olhos são tão verdes , Torre de Santa Cruz , * Vira de Coimbra , Protesto , * Cais , Vídeo Maria , Quase tudo , O cuco , = Era uma vez um cuco que não gostava de couves , O porquinho , Júlia Florista , * Neve , Da cor do pecado , A lua nasceu , Mestre de culinária , * O anzol , * Um caso mais
?	Chuva de Prata , Lua-de-Mel (Como o Diabo Gosta) , Me Faz Bem , * Não voltarei a ser fiel
A. Amargo	
A. P. Braga	Canção para desfazer equívocos
Alberto Janes	Dar de beber à dor
Alexandre	Serei uma sombra a voar

Figura 8: Alínea C