

# Processador de Inscritos numa Atividade Desportiva

Processamento de Linguagens e Compiladores

Bárbara Faria  
A85774

José Pires  
A84552

Tiago Lima  
A85126

(7 de novembro de 2021)

### **Resumo**

No âmbito da Unidade Curricular de Processamento de Linguagens e Compiladores, foi proposto ao grupo, como forma de aplicarmos o conhecimento adquirido nas aulas práticas a realização de um trabalho sobre filtragem de texto utilizando expressões regulares e o módulo **re** de Python. O grupo ficou com o trabalho n.º 1 - Processador de Inscritos numa Atividade Desportiva.

## Conteúdo

<b>1</b>	<b>Introdução</b>	<b>4</b>
<b>2</b>	<b>Exposição do problema</b>	<b>5</b>
2.1	Enunciado . . . . .	5
<b>3</b>	<b>Funcionamento</b>	<b>5</b>
3.1	Input e Output . . . . .	5
<b>4</b>	<b>Conceção/Resolução</b>	<b>6</b>
4.1	InscritosA . . . . .	6
4.1.1	Resultados . . . . .	8
4.2	InscritosB . . . . .	8
4.2.1	Resultados . . . . .	10
4.3	InscritosC . . . . .	10
4.3.1	Resultados . . . . .	12
4.4	InscritosD . . . . .	12
4.4.1	Resultados . . . . .	14
4.5	InscritosE . . . . .	14
4.5.1	Resultados . . . . .	16
<b>5</b>	<b>Conclusão</b>	<b>20</b>

# 1 Introdução

Estando no 3.<sup>o</sup> ano da Licenciatura de Ciências da Computação, foi-nos proposto a construção de um ou vários programas Python para processar o texto 'inscritos.txt' conforme solicitado nas alíneas do enunciado.

Neste relatório vamos explicar como os implementamos (as Expressões Regulares necessárias, estratégias de armazenamento de dados, estruturação) assim como o seu funcionamento.

## 2 Exposição do problema

### 2.1 Enunciado

Neste projeto, foi-nos pedido que criássemos um ou vários programas capazes de satisfazer as seguintes alíneas:

- a) imprimir o nome e o email dos concorrentes inscritos entre a 5<sup>a</sup> e a 15<sup>a</sup> posições.
- b) imprimir o nome dos concorrentes que se inscrevem como 'Individuais' e são de 'Valongo'.
- c) imprimir o telemóvel e a prova em que está inscrito cada concorrente cujo nome contenha 'Paulo' ou 'Ricardo', desde que seja da Vodafone (nº de telemóvel começado por 91).
- d) imprimir os nomes dos 20 primeiros Responsáveis de Equipe, sem repetições, convertidos para minúsculas.
- e) imprimir os 20 primeiros registos num novo ficheiro de output mas em formato 'Json'.

## 3 Funcionamento

Para a realização deste trabalho foram criados 5 programas diferentes:

- 1) InscritosA.py;
- 2) InscritosB.py;
- 3) InscritosC.py
- 4) InscritosD.py
- 5) InscritosE.py

Cada um satisfaz as alíneas do enunciado.

### 3.1 Input e Output

Os programas criados para cada alínea recebem como input o ficheiro fornecido "inscritos.txt". Estes imprimem no terminal os dados exigidos, excetuando o programa correspondente à última alínea, que cria um ficheiro em formato 'Json'.

## 4 Conceção/Resolução

O grupo começou por analisar o ficheiro de input "inscritos.txt". De imediato nos apercebemos que a primeira linha é uma mera formalidade. A segunda contém vários campos, como por exemplo `nome`, `morada`. As restantes linhas correspondem aos registos em atividades desportivas.

Depois de uma análise minuciosa, chegamos à conclusão que os vários campos estão separados por `tabs`, o que nos permitirá extrair a informação pretendida nas várias alíneas.

Nas alíneas a), b) e c) adotamos um método semelhante para a sua resolução:

Através da observação consciente e atenta do ficheiro que recebemos, decidimos guardar e organizar a informação que temos de recolher numa lista. Definimos, assim, a seguinte estrutura.

```
6 part = []
```

Reparamos que os dados que queremos extrair encontram-se a partir da terceira linha do ficheiro. Assim criamos uma lista com as linhas a partir dessa, inclusive, recorrendo ao método `readlines()`.

```
8 text = f.readlines()[2:]
```

Como foi mencionado acima, os vários campos estão separados por `tabs`. Consequentemente, usamos a função `split(pattern, string, maxsplit=0, flags=0)` do módulo `re` do Python para separar uma linha do ficheiro pelas ocorrências de um padrão. Sendo este o `tab`.

### 4.1 InscritosA

A alínea a) pede-nos para imprimir o nome e o email dos concorrentes inscritos entre a 5<sup>a</sup> e a 15<sup>a</sup> posições. Depois de fazer o `split` iremos verificar se a posição 8 da lista retornada - correspondente ao campo `Num. Participante no Evento` - é um valor desse intervalo. Para tal utilizámos a função `fullmatch(pattern, string, flag=0)` do módulo `re` do Python. O padrão empregado foi a seguinte expressão regular: `[5-9]|1[0-5]`.

Devido à maneira como a função funciona, se a `string` toda em análise fizer `match` com o padrão, então temos um número válido. Neste caso, esta é um caractere entre '5' e '9' ou é composta pelo caractere '1' seguido de um compreendido entre '0' e '5'.

Se estivermos perante um caso válido, geramos uma lista auxiliar onde iremos guardar os campos pedidos: `Nome`, `Email`, `Num. Participante no Evento`, situados nas posições 20, 28 e 8, respetivamente. Depois da recolha, adicionamos esta lista auxiliar à lista `part`.

Repetimos o processo para as várias linhas do ficheiro, aplicando um ciclo que percorre a lista `text`.

```

10 for line in text:
11
12     #As colunas do ficheiro a ler estao separadas por \t
13     y = re.split(r'\t', line)
14
15     #O concorrente tem de estar numa posicao entre 5 e 15
16     x = re.fullmatch(r'[5-9]|1[0-5]', y[7])
17     if x:
18         aux=[]
19         #Nome (Coluna 20)
20         aux.append(y[19])
21         #Email (Coluna 28) (-1 remove o \n)
22         aux.append(y[27][: -1])
23         #Num. Participante no Evento (Coluna 8)
24         aux.append(int(y[7]))
25         part.append(aux)

```

Para a apresentação da informação, ordenamos a lista `part` por posição de cada participante e imprimimos no ecrã desenhando uma tabela.

```

27 #Concorrentes organizados por posicao
28 part.sort(key = lambda x : x[2])
29
30 #Limites da tabela
31 p_max, n_max, e_max = 7, 4, 5
32 for l in part:
33     if len(str(l[2])) > p_max:
34         p_max = len(str(l[2]))
35     if len(l[0]) > n_max:
36         n_max = len(l[0])
37     if len(l[1]) > e_max:
38         e_max = len(l[1])
39
40 #Desenhar a tabela
41 print("Nome e Email dos concorrentes inscritos entre a 5a e a 15a
      posicao\n")
42 print("| Posicao+"+' '*(p_max-7)+" | Nome+"+' '*(n_max-4)+" | Email
      "+"+' '*(e_max-5)+" | ")
43 print("-+"+'-'*(p_max)+"-|-+"+'-'*(n_max)+"-|-+"+'-'*(e_max)+"-|- ")
44 for ins in part:
45     print('| '+str(ins[2])+' '*(p_max-len(str(ins[2])))+" | "+ins
      [0]+' '*(n_max-len(ins[0]))+" | "+ins[1]+' '*(e_max-len(ins[1]))
      )+' | ')
46
47 f.close()

```

#### 4.1.1 Resultados

Nome e Email dos concorrentes inscritos entre a 5ª e a 15ª posição

Posição	Nome	Email
5	Carlos Alberto da Silva Cardoso	casc1972@hotmail.com
6	Ricardo Salgueiro	rs@selcofootwear.com
7	Rui Gilberto Santos Correia	gilcorreia@yahoo.com
8	Luis Santos Poeira	lspoeira@gmail.com
9	paulo de castro rocha	pcastrorocha@gmail.com
10	JOSE COELHO	rota_pontual@sapo.pt
11	José Moreira	jmoreiracdf@gmail.com
12	João Costa	jfscosta@gmail.com
13	Paulo Pimentel Torres	geral@vieirafreitas.pt
14	João Pimentel Torres	geral@vieirafreitas.pt
15	Vasco Manuel de Sequeiros Barreto Martins de Araújo	vascosequeiros@yahoo.com

Figura 1: ResA.jpg

## 4.2 InscritosB

A alínea b) pede-nos para imprimir o nome dos concorrentes que se inscrevem como 'Individuais' e são de 'Valongo'. Depois de fazer o `split` faremos três verificações:

- 1) O concorrente tem de se inscrever como "Individual". Para tal recorremos à função `search(pattern, string, flags=0)` do módulo `re` do Python; o padrão a ser usado será a expressão regular `(?i:individual)`; e a `string` analisada encontra-se na posição 10.
- 2) O concorrente deve habitar em "Valongo". Mais uma vez usamos a função `search`; o padrão a ser usado será a expressão regular `(?i:valongo)`; e a `string` analisada encontra-se na posição 21.
- 3) Por último, a inscrição não pode estar "Expirada", i.e., poderá ser "Confirmada" ou "Registada". Assim como nos casos anteriores, iremos empregar a função `search`; o padrão a ser usado será a expressão regular `(?i:confirmada)|(?i:registada)`; e a `string` analisada encontra-se na posição 3.

Se estas três condições forem válidas, geramos uma lista auxiliar onde iremos guardar os campos pedidos: `Nome`, `Morada`, situados nas posições 20 e 21, respetivamente. Depois da recolha, adicionamos esta lista auxiliar à lista `part`.

Repetimos o processo para as várias linhas do ficheiro, aplicando um ciclo que percorre a lista `text`.



```

10 for line in text:
11
12     #As colunas do ficheiro a ler estao separadas por \t
13     y = re.split(r'\t', line)
14
15     #Expressoes regulares que vao filtrar os dados pretendidos
16     tipo = re.search(r'(?i:individual)', y[9])
17     morada = re.search(r'(?i:valongo)', y[20])
18     insc = re.search(r'(?i:confirmada)|(?i:registada)', y[2])
19
20     #0 concorrente tem de ser individual, ser de Valongo e nao ter
    a inscricao expirada
21     if(tipo and morada and insc):
22         aux = []
23         #Nome (Coluna 20)
24         aux.append(y[19])
25         #Morada (Coluna 21)
26         aux.append(y[20])
27         part.append(aux)

```

Para a apresentação da informação, ordenamos a lista `part` pelo nome de cada participante e imprimimos no ecrã desenhando uma tabela.

```

29 #Concorrentes organizados por nome
30 part.sort(key = lambda x : x[0])
31
32 #Limites da tabela
33 n_max, m_max = 4, 6
34 for l in part:
35     if len(l[0]) > n_max:
36         n_max = len(l[0])
37     if len(l[2]) > m_max:
38         m_max = len(l[2])
39
40 #Desenhar a tabela
41 print("Concorrentes de Valongo que se inscrevem como \"Individual
    \"\n")
42 print("| Nome" + ' '*(n_max-4) + " | Morada" + ' '*(m_max-6) + ' |')
43 print("| Nome" + ' '*(n_max-4) + " | Morada" + ' '*(m_max-6) + ' |')
44 for ins in part:
45     print('| ' + ins[0] + ' '*(n_max-len(ins[0])) + " | " + ins[2] + ' '*(
        m_max-len(ins[2])) + ' |')
46
47 f.close()

```

### 4.2.1 Resultados

Nome	Morada
Dulce Moreda	Valongo
Paulo Domingues	Valongo

Figura 2: ResB.jpg

## 4.3 InscritosC

A alínea c) pede-nos para imprimir o telemóvel e a prova em que está inscrito cada concorrente cujo nome contenha 'Paulo' ou 'Ricardo', desde que seja da Vodafone (nº de telemóvel começado por 91). Depois de fazer o `split` faremos três verificações:

- 1) O concorrente tem de conter "Paulo" ou "Ricardo" no nome. Para tal recorreremos à função `search`; o padrão a ser usado será a expressão regular `(?i:paulo)|(?i:ricardo)`; e a `string` analisada encontra-se na posição 20.
- 2) A inscrição não pode estar "Expirada", i.e., poderá ser "Confirmada" ou "Registada". Portanto, iremos empregar a função `search`; o padrão a ser usado será a expressão regular `(?i:confirmada)|(?i:registada)`; e a `string` analisada encontra-se na posição 3.
- 3) Por fim, o concorrente tem de ser tarifário Vodafone, i.e., o número telefone começa por "91". Mais uma vez usamos a função `search`; o padrão a ser usado será a expressão regular `91[0-9]7`; e a `string` analisada encontra-se na posição 11.

Se estas três condições forem válidas, geramos uma lista auxiliar onde iremos guardar os campos pedidos: `Nome`, `Tipo de Inscrição`, `Telefone/Telemóvel`, situados nas posições 20, 3 e 11, respetivamente. Depois da recolha, adicionamos esta lista auxiliar à lista `part`.

Repetimos o processo para as várias linhas do ficheiro, aplicando um ciclo que percorre a lista `text`.

```
10 for line in text:
11
12     #As columnas do ficheiro a ler estao separadas por \t
13     y = re.split(r'\t', line)
```

```

14
15 #Expressoes regulares que vao filtrar os dados pretendidos
16 nome = re.search(r'(?i:paulo)|(?i:ricardo)', y[19])
17 insc = re.search(r'(?i:confirmada)|(?i:registada)', y[2])
18 num = re.search(r'91[0-9]{7}', y[10])
19
20 #O concorrente tem de ter a inscricao confirmada, tarifario
21 Vodafone e Paulo ou Ricardo como nome
22 if(nome and insc and num):
23     aux = []
24     #Nome (Coluna 20)
25     aux.append(y[19])
26     #Tipo de Inscricao (Coluna 5)
27     aux.append(y[4])
28     #Telefone/ Telemovel (Coluna 11)
29     aux.append(y[10])
30     part.append(aux)

```

Para a apresentação da informação, ordenamos a lista `part` pelo nome da prova seguido do nome de cada participante e imprimimos no ecrã desenhando uma tabela.

```

31 #Concorrentes organizados por prova e por nome
32 part.sort(key = lambda x : (x[1],x[0]))
33
34 #Limites da tabela
35 n_max, p_max, v_max = 4, 5, 6
36 for l in part:
37     if len(l[0]) > n_max:
38         n_max = len(l[0])
39     if len(l[1]) > p_max:
40         p_max = len(l[1])
41     if len(l[2]) > v_max:
42         v_max = len(l[2])
43
44 #Desenhar a tabela
45 print("Provas dos concorrentes cujo nome contem Paulo ou Ricardo\n")
46 print("| Nome" + ' '*(n_max-4) + " | Prova" + ' '*(p_max-5) + " | Numero" + ' '*(v_max-6) + ' | ')
47 print("-" + '-'*(n_max) + "-" + '-'*(p_max) + "-" + '-'*(v_max) + "-" + ' ')
48 for ins in part:
49     print(' | ' + ins[0] + ' '*(n_max-len(ins[0])) + " | " + ins[1] + ' '*(p_max-len(ins[1])) + " | " + ins[2] + ' '*(v_max-len(ins[2])) + ' | ')
50
51 f.close()

```

### 4.3.1 Resultados

Provas dos concorrentes cujo nome contém Paulo ou Ricardo

Nome	Prova	Número
Ricardo Campos	Corrida da Geira	917804565
Ricardo Ernesto dos Santos Geraldes Domingues	Corrida da Geira	919121762
J Paulo Marques	Ultra Trail	918657054
Paulo Domingues	Ultra Trail	917067362
Paulo Jorge	Ultra Trail	917300483
Paulo Serra	Ultra Trail	914465667
Ricardo Couto	Ultra Trail	914536925

Figura 3: ResC.jpg

## 4.4 InscritosD

A alínea d) pede-nos para imprimir os nomes dos 20 primeiros Responsáveis de Equipe, sem repetições, convertidos para minúsculas. Fazendo uso da função `next()` saltamos as duas primeiras linhas do ficheiro.

```
5 #salta as duas primeira linhas (titulo e cabecalho)
6 next(f)
7 next(f)
```

Para armazenar os dados, criamos uma lista `nomes`, onde iremos inserir os Nomes dos Responsáveis.

```
9 #lista com os nomes dos responsaveis sem repeticoes
10 nomes = []
```

Desta vez usamos a função `split` com a *keyword* `maxsplit` igualada a 1, visto que só nos interessa extrair o primeiro de cada linha do ficheiro

```
17 #so precisamos de ir buscar o primeiro campo (nome do responsavel)
18 campos = re.split('\t', line, maxsplit=1)
```

Para evitar repetições de nomes, verificamos se o nome a ser tratado (convertido para minúsculas) ainda não foi inserido na lista. Em caso afirmativo, ele é adicionado.

```
20 #se o nome do responsavel que esta a ser tratado ainda nao estiver
    na lista e adicionado
21 if campos[0].lower() not in nomes:
22     nomes.append(campos[0].lower())
```

Repetimos o processo para as várias linhas do ficheiro, aplicando um ciclo que o percorre `text`. No início de cada iteração, averiguamos se o comprimento da lista `nomes` é vinte. Se sim, paramos o ciclo.

```
15 for line in f:
16     #se a lista possuir 20 elementos o ciclo para
17     if len(nomes)==20:
18         break
19
20     #so precisamos de ir buscar o primeiro campo (nome do
    responsavel)
21     campos = re.split('\t', line, maxsplit=1)
22
23     #se o nome do responsavel que esta a ser tratado ainda nao
    estiver na lista e adicionado
24     if campos[0].lower() not in nomes:
25         nomes.append(campos[0].lower())
26
27     #limites da tabela
28     if len(campos[0])>n_max:
29         n_max = len(campos[0])
```

Para a apresentação da informação, ordenamos a lista `nomes` no ecrã desenhando uma tabela.

```
31 #desenhar a tabela
32 print("Os 20 primeiros Responsaveis de Equipa (nome em minusculas s
    / repeticoes)\n")
33 print("| Nome do Responsavel"+' '*(n_max-19)+' |')
34 print("-" + "-"*(n_max)+'-|')
35 for n in nomes:
36     print('| '+n+' '*(n_max-len(n))+ ' |')
37
38 f.close()
```

#### 4.4.1 Resultados

```
Os 20 primeiros Responsáveis de Equipa (nome em minúsculas s/ repetições)
```

Nome do Responsável
mario pires
francisco neto silva
luis santos poeira
jose moura de sousa
fernando jorge mendes carneiro
carlos alberto da silva cardoso
ricardo salgueiro
rui gilberto santos correia
josé moreira
hélter silva
jose coelho
paulo de castro rocha
artur bernardo
j paulo marques
bruno manuel henriques maceda
joão costa
bruno filipe de sa campelo
paulo pimentel torres
vasco manuel de sequeiros barreto martins de Araújo
helder matos

Figura 4: ResD.jpg

#### 4.5 InscritosE

A alínea e) pede-nos para imprimir os 20 primeiros registos num novo ficheiro de output mas em formato 'Json'.

Começamos por criar um ficheiro nesse formato e com o auxílio da função `next()` ignoramos a primeira linha do ficheiro. A seguir guardamos a segunda linha na `string` `cabecalho` e os vinte primeiros registos na lista `text`.

```
4 #Cria/Reescreve ficheiro .JSON para escrita
5 g = open("E.JSON", "w", encoding="utf8")
6
7 #Ignora a primeira linha
8 next(f)
9 cabecalho = f.readline()
10 text = f.readlines()[0:20]
```

Fazemos o split de `cabecalho` usando o `tab` como padrão e armazenamos essa informação na lista `cab`. O proximo passo é escrever a primeira linha no ficheiro output. Cada registo será um objeto, portanto o ficheiro será composto por um `array` de objetos. Consequentemente, a primeira linha será um '[' seguido de uma mudança de linha.

```

12 #Cada parametro do cabecalho esta separado por \t
13 cab = re.split(r'\t', cabecalho)
14
15 g.write("[\n")

```

Logo depois, vamos percorrer a lista `text` até ao penúltimo elemento. No início de cada iteração do ciclo escrevemos no ficheiro output um '{', precedido de um `tab` para dar inicio ao nosso objeto. Seguidamente, executamos o `split` na linha a ser lida, guardando os dados na lista `y`. Percorremos cada componente dela até ao penúltimo e registamos no ficheiro `json` essa informação associada ao respetivo campo do cabeçalho, i.e., na forma **"campo": "dados"** . Além disso, dois `tabs` devem antecede-los a fim de pertencerem ao objeto.

Chegando ao ultimo dado de `y` aplicamos o mesmo processo, mas com o cuidado de remover o `\n` que fica depois do `split`. Para "fechar"o objeto imprimimos um '}' seguido de uma vírgula e uma mudança de linha.

```

17 #A variavel 'j' vai percorrer os registos que nos interessam
18 for j in range(len(text)-1):
19
20     g.write("\t{\n")
21     y = re.split(r'\t', text[j])
22
23     #A variavel 'i' vai percorrer cada posicao do cabecalho
24     for i in range(len(y)-1):
25         g.write("\t\t" + "\"" + cab[i] + "\"" + ":" + ' ' + "\"" +
26             y[i] + "\"" + "," + "\n")
27
28     g.write("\t\t" + "\"" + cab[i+1][-1] + "\"" + ":" + ' ' + "\"" +
29         y[i+1][-1] + "\"" + "\n")
30     g.write("\t},\n")

```

Chegando à última linha repetimos o processo, mas desta vez, ao "fechar"o objeto, não colocamos uma vírgula e mudamos logo de linha. A fim de concluir o `array`, escrevemos no ficheiro de ouput um ']' .

```

30 #0 ultimo elemento
31 g.write("\t{\n")
32 w = re.split(r'\t', text[j + 1])
33
34 for i in range(len(w)-1):
35     g.write("\t\t" + "\"" + cab[i] + "\"" + ":" + ' ' + "\"" + w[i]
36         + "\"" + "," + "\n")
37
38 g.write("\t\t" + "\"" + cab[i+1][-1] + "\"" + ":" + ' ' + "\"" + w
39     [i+1][-1] + "\"" + "\n")
40 g.write("\t}\n")
41
42 g.write("]")

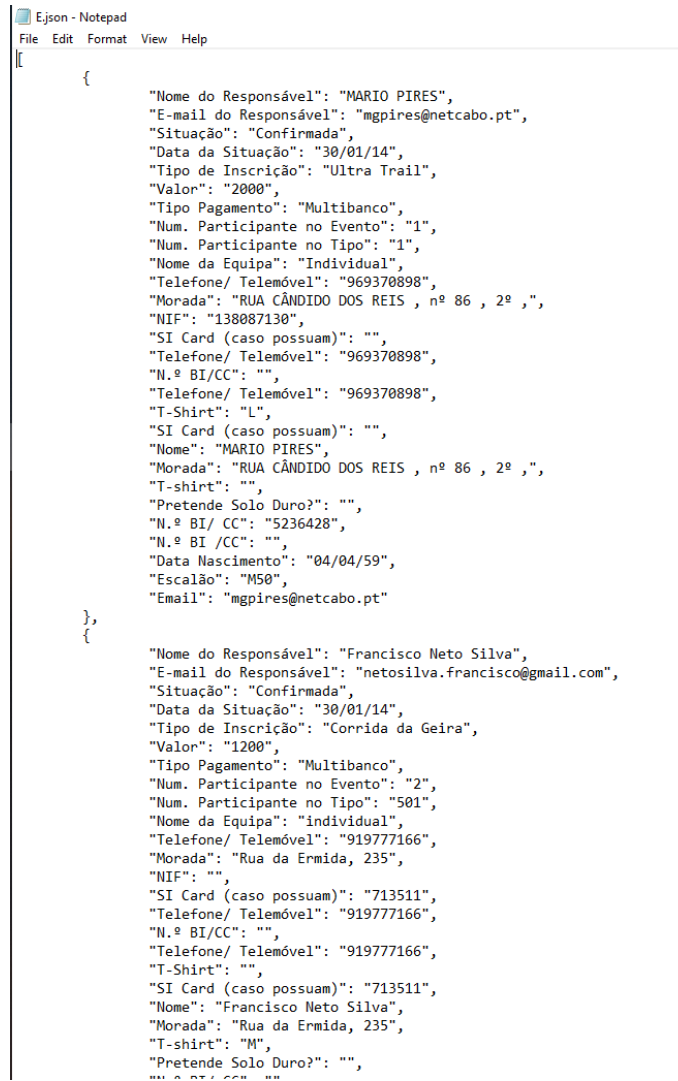
```

```

42 f.close()
43 g.close()

```

#### 4.5.1 Resultados



```

Ejson - Notepad
File Edit Format View Help
[
  {
    "Nome do Responsável": "MARIO PIRES",
    "E-mail do Responsável": "mgpires@netcabo.pt",
    "Situação": "Confirmada",
    "Data da Situação": "30/01/14",
    "Tipo de Inscrição": "Ultra Trail",
    "Valor": "2000",
    "Tipo Pagamento": "Multibanco",
    "Num. Participante no Evento": "1",
    "Num. Participante no Tipo": "1",
    "Nome da Equipa": "Individual",
    "Telefone/ Telemóvel": "969370898",
    "Morada": "RUA CÂNDIDO DOS REIS , nº 86 , 2º ,",
    "NIF": "138087130",
    "SI Card (caso possuam)": "",
    "Telefone/ Telemóvel": "969370898",
    "N.º BI/CC": "",
    "Telefone/ Telemóvel": "969370898",
    "T-Shirt": "L",
    "SI Card (caso possuam)": "",
    "Nome": "MARIO PIRES",
    "Morada": "RUA CÂNDIDO DOS REIS , nº 86 , 2º ,",
    "T-shirt": "",
    "Pretende Solo Duro?": "",
    "N.º BI/ CC": "5236428",
    "N.º BI /CC": "",
    "Data Nascimento": "04/04/59",
    "Escalão": "M50",
    "Email": "mgpires@netcabo.pt"
  },
  {
    "Nome do Responsável": "Francisco Neto Silva",
    "E-mail do Responsável": "netosilva.francisco@gmail.com",
    "Situação": "Confirmada",
    "Data da Situação": "30/01/14",
    "Tipo de Inscrição": "Corrida da Geira",
    "Valor": "1200",
    "Tipo Pagamento": "Multibanco",
    "Num. Participante no Evento": "2",
    "Num. Participante no Tipo": "501",
    "Nome da Equipa": "individual",
    "Telefone/ Telemóvel": "919777166",
    "Morada": "Rua da Ermida, 235",
    "NIF": "",
    "SI Card (caso possuam)": "713511",
    "Telefone/ Telemóvel": "919777166",
    "N.º BI/CC": "",
    "Telefone/ Telemóvel": "919777166",
    "T-Shirt": "",
    "SI Card (caso possuam)": "713511",
    "Nome": "Francisco Neto Silva",
    "Morada": "Rua da Ermida, 235",
    "T-shirt": "M",
    "Pretende Solo Duro?": "",
    "N.º BI/ CC": ""
  }
]

```

Figura 5: ResE1.jpg



```

Ejson - Notepad
File Edit Format View Help
},
{
  "Nome do Responsável": "Francisco Neto Silva",
  "E-mail do Responsável": "netosilva.francisco@gmail.com",
  "Situação": "Confirmada",
  "Data da Situação": "30/01/14",
  "Tipo de Inscrição": "Corrida da Geira",
  "Valor": "1200",
  "Tipo Pagamento": "Multibanco",
  "Num. Participante no Evento": "2",
  "Num. Participante no Tipo": "501",
  "Nome da Equipa": "individual",
  "Telefone/ Telemóvel": "919777166",
  "Morada": "Rua da Ermida, 235",
  "NIF": "",
  "SI Card (caso possuam)": "713511",
  "Telefone/ Telemóvel": "919777166",
  "N.º BI/CC": "",
  "Telefone/ Telemóvel": "919777166",
  "T-Shirt": "",
  "SI Card (caso possuam)": "713511",
  "Nome": "Francisco Neto Silva",
  "Morada": "Rua da Ermida, 235",
  "T-shirt": "M",
  "Pretende Solo Duro?": "",
  "N.º BI/ CC": "",
  "N.º BI /CC": "13474034",
  "Data Nascimento": "22/04/89",
  "Escalação": "SENIOR Masc",
  "Email": "netosilva.francisco@gmail.com"
},
{
  "Nome do Responsável": "Luis Santos Poeira",
  "E-mail do Responsável": "lspoeira@gmail.com",
  "Situação": "Confirmada",
  "Data da Situação": "03/02/14",
  "Tipo de Inscrição": "Ultra Trail",
  "Valor": "2000",
  "Tipo Pagamento": "Multibanco",
  "Num. Participante no Evento": "8",
  "Num. Participante no Tipo": "5",
  "Nome da Equipa": "Porto Runners",
  "Telefone/ Telemóvel": "963036120",
  "Morada": "R. Soc. Nacional Fósforos, 160 - 1º Andar Hab. 6 4150 Porto",
  "NIF": "",
  "SI Card (caso possuam)": "",
  "Telefone/ Telemóvel": "963036120",
  "N.º BI/CC": "",
  "Telefone/ Telemóvel": "963036120",
  "T-Shirt": "M",
  "SI Card (caso possuam)": "",
  "Nome": "Luis Santos Poeira",
  "Morada": "R. Soc. Nacional Fósforos, 160 - 1º Andar Hab. 6 4150 Porto",
  "T-shirt": "",
  "Pretende Solo Duro?": "",
  "N.º BI/ CC": "73202258"
}

```

Figura 6: ResE2.jpg

```

Ejson - Notepad
File Edit Format View Help

{
  "Nome do Responsável": "Paulo Pimentel Torres",
  "E-mail do Responsável": "geral@vieirafreitas.pt",
  "Situação": "Confirmada",
  "Data da Situação": "03/02/14",
  "Tipo de Inscrição": "Corrida da Geira",
  "Valor": "1200",
  "Tipo Pagamento": "Multibanco",
  "Num. Participante no Evento": "14",
  "Num. Participante no Tipo": "505",
  "Nome da Equipa": "TURBULENTOS",
  "Telefone/ Telemóvel": "966040136",
  "Morada": "Rua Costa Soares, 39",
  "NIF": "132287641",
  "SI Card (caso possuam)": "",
  "Telefone/ Telemóvel": "966040136",
  "N.º BI/CC": "",
  "Telefone/ Telemóvel": "966040136",
  "T-Shirt": "",
  "SI Card (caso possuam)": "",
  "Nome": "João Pimentel Torres",
  "Morada": "Rua Costa Soares, 39",
  "T-shirt": "S",
  "Pretende Solo Duro?": "",
  "N.º BI/ CC": "",
  "N.º BI /CC": "13358723",
  "Data Nascimento": "11/09/88",
  "Escalaõ": "SENIOR Masc",
  "Email": "geral@vieirafreitas.pt"
},
{
  "Nome do Responsável": "Vasco Manuel de Sequeiros Barreto Martins de Araújo",
  "E-mail do Responsável": "vascosequeiros@yahoo.com",
  "Situação": "Confirmada",
  "Data da Situação": "03/02/14",
  "Tipo de Inscrição": "Corrida da Geira",
  "Valor": "1200",
  "Tipo Pagamento": "Multibanco",
  "Num. Participante no Evento": "15",
  "Num. Participante no Tipo": "506",
  "Nome da Equipa": "TURBULENTOS",
  "Telefone/ Telemóvel": "919421515",
  "Morada": "Rua S. Domingos 174 3º esq.",
  "NIF": "192410148",
  "SI Card (caso possuam)": "337363",
  "Telefone/ Telemóvel": "919421515",
  "N.º BI/CC": "",
  "Telefone/ Telemóvel": "919421515",
  "T-Shirt": "",
  "SI Card (caso possuam)": "337363",
  "Nome": "Vasco Manuel de Sequeiros Barreto Martins de Araújo",
  "Morada": "Rua S. Domingos 174 3º esq.",
  "T-shirt": "XL",
  "Pretende Solo Duro?": "",
  "N.º BI/ CC": "",
  "N.º BI /CC": "13358723"
}

```

Figura 7: ResE3.jpg

```

Ejson - Notepad
File Edit Format View Help

"Valor": "1200",
"Tipo Pagamento": "Multibanco",
"Num. Participante no Evento": "14",
"Num. Participante no Tipo": "505",
"Nome da Equipa": "TURBULENTOS",
"Telefone/ Telemovel": "966040136",
"Morada": "Rua Costa Soares, 39",
"NIF": "132287641",
"SI Card (caso possuam)": "",
"Telefone/ Telemovel": "966040136",
"N.º BI/CC": "",
"Telefone/ Telemovel": "966040136",
"T-Shirt": "",
"SI Card (caso possuam)": "",
"Nome": "João Pimentel Torres",
"Morada": "Rua Costa Soares, 39",
"T-shirt": "S",
"Pretende Solo Duro?": "",
"N.º BI/ CC": "",
"N.º BI /CC": "13358723",
"Data Nascimento": "11/09/88",
"Escalão": "SENIOR Masc",
"Email": "geral@vieirafreitas.pt"
},
{
  "Nome do Responsável": "Vasco Manuel de Sequeiros Barreto Martins de Araújo",
  "E-mail do Responsável": "vascosequeiros@yahoo.com",
  "Situação": "Confirmada",
  "Data da Situação": "03/02/14",
  "Tipo de Inscrição": "Corrida da Geira",
  "Valor": "1200",
  "Tipo Pagamento": "Multibanco",
  "Num. Participante no Evento": "15",
  "Num. Participante no Tipo": "506",
  "Nome da Equipa": "TURBULENTOS",
  "Telefone/ Telemovel": "919421515",
  "Morada": "Rua S. Domingos 174 3º esq.",
  "NIF": "192410148",
  "SI Card (caso possuam)": "337363",
  "Telefone/ Telemovel": "919421515",
  "N.º BI/CC": "",
  "Telefone/ Telemovel": "919421515",
  "T-Shirt": "",
  "SI Card (caso possuam)": "337363",
  "Nome": "Vasco Manuel de Sequeiros Barreto Martins de Araújo",
  "Morada": "Rua S. Domingos 174 3º esq.",
  "T-shirt": "XL",
  "Pretende Solo Duro?": "",
  "N.º BI/ CC": "",
  "N.º BI /CC": "6672944",
  "Data Nascimento": "26/05/64",
  "Escalão": "SENIOR Masc",
  "Email": "geral@vieirafreitas.pt"
}
}
1

```

Figura 8: ResE4.jpg

## 5 Conclusão

Através deste primeiro trabalho prático foi-nos possível pôr em prática o conhecimento sobre Expressões Regulares e o módulo `re` de Python dados durante as aulas, bem como a sua aplicação real em situações de filtragem de grandes porções de texto.

Fazemos uma avaliação positiva do nosso desempenho neste projeto pois achamos que o objetivo do mesmo foi concluído. Apesar alguns desafios encontrados na sua realização, estes foram superados com distinção, revelando assim um trabalho consistente e bem estruturado.