

Processamento de Linguagens (3º ano de Curso)

Trabalho Prático 1

Relatório de Desenvolvimento

Ricardo Gomes
(a93785)

Hugo Pereira
(a93752)

5 de abril de 2021

Resumo

O trabalho realizado pelo nosso grupo, na cadeira de Processamento de Linguagens, foi referente ao enunciado 1, *Processador de Inscritos numa atividade Desportiva*, que consistia em receber um ficheiro *JSON* e processar a informação nele contido.

Conteúdo

1	Introdução	3
2	Descrição do Problema	5
3	Concepção/desenho da Resolução	6
3.1	Estruturas de Dados	6
3.2	Algoritmos	7
3.2.1	Alínea 1	7
3.2.2	Alínea 2	7
3.2.3	Alínea 3	7
3.2.4	Alínea 4	8
3.2.5	Alínea 5	8
4	Testes	12
4.1	Alternativas, Decisões e Problemas de Implementação	12
4.2	Testes realizados e Resultados	12
5	Conclusão	21
A	Código do Programa	22

Lista de Figuras

4.1	Menu	12
4.2	Resultados da Opção 1	13
4.3	Alguns resultados da Opção 2	14
4.4	Alguns resultados da Opção 3	15
4.5	Resultados da Opção 4	16
4.6	Página HTML Principal	17
4.7	Página HTML - Individuais	18
4.8	Página HTML - Equipa Turbulentos	19
4.9	Página HTML - Equipa EDV Viana Trail	20
4.10	Página HTML - Equipa Print Team	20

Capítulo 1

Introdução

Este trabalho foi proposto no âmbito da unidade curricular de Processamento de Linguagens, e tem como objectivos, aumentar a capacidade de escrever expressões regulares para serem descritos padrões de frases em textos, desenvolver a partir de expressões regulares processadores de linguagens que filtrem e transformem textos contidos num ficheiro JSON.

Para a sua elaboração utilizamos a linguagem Python e as bibliotecas `re` e `diffib`. Nas secções que se seguem iremos explicar de forma mais detalhada os problemas encontrados e as suas respectivas soluções.

Estrutura do Relatório

No decorrer deste relatório, podemos verificar que no capítulo 1 está presente a introdução referente a este trabalho, no capítulo 2 faz-se uma análise do problema proposto, no capítulo 3 detalhamos pormenorizadamente a resolução do problema e mostrámos a forma como respondemos a cada questão do enunciado, no capítulo 4 apresentamos os resultados das funções do programas e algumas páginas HTML geradas e por último no capítulo 5 termina-se o relatório com uma conclusão.

Capítulo 2

Descrição do Problema

No enunciado escolhido pretende-se aceder a informações de um documento criado por uma organização desportiva. Neste documento encontram-se atletas de várias equipas, inscritos nas mais diversas competições, sendo que o programa criado tem de ser capaz não só de imprimir informações de atletas, equipas e competições, como também de gerar páginas html com todas as informações sobre as equipas com atletas inscritos.

Capítulo 3

Concepção/desenho da Resolução

3.1 Estruturas de Dados

No início do trabalho, o grupo decidiu retirar a informação necessária do documento *JSON* através de duas expressões regulares e guardar tudo num array.

```
1 inscricao = re.findall(r'(".*":(^[^"]*))', docJSON)
2
3 for elem in inscricao:
4     queroisto = re.search(r'"((\w*|\s|\W*))"', elem[1])
5     inscritos.append(queroisto.group(1))
```

Após termos toda a informação guardada num array, pretendemos passar toda essa informação para um array, em que cada elemento do array era um array contendo apenas as informações de um inscrito. Tendo já toda a informação organizada, decidimos guardar a informação em dicionários para posteriormente ser mais fácil de utilizar.

```
1 for val in range(0, len(inscritos), 7):
2     next_jump = val + 7
3     shot_values = inscritos[val:next_jump]
4     inscricoes.append(shot_values)
5
6 dados = {}
7
8 for e in inscricoes:
9     dados['nome'] = e[0].upper()
10    dados['dataNasc'] = e[1].upper()
11    dados['morada'] = e[2].upper()
12    dados['email'] = e[3].upper()
13    dados['prova'] = e[4].upper()
14    dados['escalao'] = e[5].upper()
15    dados['equipa'] = e[6].upper()
16    dic.append(dados)
17    dados = {}
```

3.2 Algoritmos

De seguida, listam-se todos os algoritmos utilizados para responder às questões colocadas.

3.2.1 Alínea 1

```
1     if opcao == '1':
2         for i in dic:
3             if re.search(r'(?i)valongo', i['morada']) and re.search(r'(?i)individual', i['equipa']):
4                 nome = i['nome']
5                 print("Nome: " + nome.upper() + "\n\n")
```

Listing 1: Algoritmo da alínea 1

3.2.2 Alínea 2

```
1     if opcao == '2':
2         for i in dic:
3             if (re.match(r'(?i)ricardo', i['nome']) or re.match(r'(?i)paulo', i['nome'])) and re.search(r'(?i)gmail', i['email']):
4                 nome = i['nome']
5                 prova = i['prova']
6                 print("Nome: " + nome.upper() + "\n" + "Prova : " + prova.upper() + "\n\n")
```

Listing 2: Algoritmo da alínea 2

3.2.3 Alínea 3

```
1     if opcao == '3':
2         for i in dic:
3             if re.search(r'(?i)turbulentos', i['equipa']):
4                 nome = i['nome']
5                 data = i['dataNasc']
6                 morada = i['morada']
7                 email = i['email']
8                 prova = i['prova']
9                 escalao = i['escalao']
10                equipa = i['equipa']
11                print("Nome: " + nome.upper() + "\nData: " + data.upper() +
12                    "\nMorada: " + morada.upper() + "\nEmail: " + email.upper() + "\nProva: " + prova.upper()
13                    + "\nEscalao: " + escalao.upper() + "\nEquipa: " + equipa.upper() + "\n\n")
```

Listing 3: Algoritmo da alínea 3

3.2.4 Alínea 4

```
1     if opcao == '4':
2         escaloes = []
3         escalaoAtletas = []
4
5         for i in dic:
6             escalao = i['escalao']
7             escalaoAtletas.append(escalao)
8
9         escaloes = escalaoAtletas.copy() # Copia Lista para se operar sem perder info
10        escaloes.sort() # Ordena Lista
11        escaloes = list(dict.fromkeys(escaloes)) # Remove Duplicados
12
13        for k in escaloes:
14            if k == "":
15                print("\nSem Escalao: " + str(escalaoAtletas.count(k))) # Conta Atletas Sem Escalao
16            else:
17                print("\n" + k + ": " + str(escalaoAtletas.count(k))) # Conta Atletas por cada Escalao
```

Listing 4: Algoritmo da alínea 4

3.2.5 Alínea 5

```
1  if opcao == '5':
2      original = []
3      dup = []
4
5      for i in dic:
6          equipa = i['equipa']
7          original.append(equipa.upper())
8
9      dup = list(dict.fromkeys(original))
10
11
12     for index, k in enumerate(original):
13         for j in dup:
14             string1 = k
15             string2 = j
16             matches = difflib.SequenceMatcher(
17                 None, string1, string2).ratio()
18
19             if(matches >= 0.7):
20                 original[index] = j
21
22
23     equipasOficias = original.copy() #cópia da lista original já com os nomes corretos
24     equipasOficias = list(dict.fromkeys(equipasOficias))
25
26
27
28
29     #atualiza o dicionário com os nomes iguais
30     for d in dic:
31         for e in equipasOficias:
32             string1 = d['equipa']
33             string2 = e.upper() #transforma em maiúscula para comparar com as
34             #equipas do dic que estão em maiúsculas
35             matches = difflib.SequenceMatcher(
36                 None, string1, string2).ratio()
37             if (matches >= 0.7):
38                 d['equipa'] = e
39
40     # troca os / e / por espaços nas equipas para ser mais fácil criar o respetivo ficheiro html
41     for d in dic:
42         if (re.search(r'\|', d['equipa'])):
43             d['equipa'] = re.sub(r'\|', ' ', d['equipa'])
44         elif (re.search(r'\/', d['equipa'])):
45             d['equipa'] = re.sub(r'\/', ' ', d['equipa'])
46
47
48     equipasOficiasAtual2 = [] #coném as equipas sem / e /
49     for d in dic:
50         equipasOficiasAtual2.append(d['equipa'])
51
52     equipasOficiasAtual = list(dict.fromkeys(equipasOficiasAtual2))
```

```

1     dicEquipas = {} #vai conter o nome das equipas e o numero de atletas de cada equipa
2
3     for ele in equipasOficiasAtual:
4         somatorio = 0
5         somatorio = somatorio + equipasOficiasAtual2.count(ele) #soma o numeroa de elementos de equipas com o nome
6         dicEquipas[ele] = somatorio #coloca o numeroa de elementos na equipa com nome na posicao 0 de ele
7     equipasOrdenadas = sorted(dicEquipas.items(), key=lambda x: x[1], reverse=True) #cria tuplos com os elementos
8
9
10    f = open("equipas.html", "w")
11
12    f.write('''<!DOCTYPE html>
13    <html>
14        <head meta charset="UTF-8">
15            <title>Equipas</title>
16        </head>
17        <body>
18            <h1>Equipas</h1>
19            <ul>
20                '''
21
22    #adiciona as equipas no ficheiro principal
23    for i in equipasOrdenadas: #tuplo
24        if (re.search(r'\s', i[0])): # cria os nomes dos ficheiros em html com _ em vez de espaço
25            nomeEquipa2 = re.sub(r'\s',
26                                r'_ ',
27                                i[0])
28            f.write("<li><a href=\"" + str(nomeEquipa2) + ".html\">" + str(i[0]) + "</a>: " + str(i[1]) + "</li><br>")
29        else:
30            f.write("<li><a href=\"" + str(i[0]) + ".html\">" + str(i[0]) + "</a>: " + str(i[1]) + "</li><br>")
31    f.write(''''</ul>
32    </body>
33    </html>''')
34
35    f.close()
36
37    # cria os ficheiros das equipas e adiciona os inscritos de cada equipa
38    for i in equipasOficiasAtual:
39        if (re.search(r'\s', i)):
40            nomeEquipa = re.sub(r'\s',
41                                r'_ ',
42                                i)
43            newfile = open(str(nomeEquipa) + ".html", "w")
44        else:
45            newfile = open(str(i) + ".html", "w")
46        newfile.write('''<!DOCTYPE html>
47        <html>
48            <head meta charset="UTF-8">
49                <title>''' + str(i) + ''''</title>
50            </head>
51            <body>
52                <h1>''' + str(i) + ''''</h1><br>
53                <ul>
54                    '''

```

```
1     for d in dic:
2         if i == d['equipa']:
3             newfile.write("<li>" + d['nome'] + "</li>")
4 newfile.write(''</ul>
5             </body>
6             </html>'')
7 newfile.close()
```

Listing 5: Algoritmo da alínea 5

Capítulo 4

Testes

4.1 Alternativas, Decisões e Problemas de Implementação

Durante a elaboração do trabalho o grupo apercebeu-se da existência de equipas parecidas presentes no ficheiro *JSON*. Algumas diferiam em acentos, outras em espaços e algumas até em vogais/consoantes, tratando-se claramente de erros gramaticais.

Como tal, e para facilitar a interpretação de resultados, o grupo decidiu que todas as equipas cujos nomes tivessem mais de 70% de *matches* seriam consideradas as mesmas equipas, sendo atribuído o mesmo nome a ambas.

4.2 Testes realizados e Resultados

Mostram-se a seguir alguns resultados obtidos referentes às questões colocadas:

```
Escolha uma opcao:
  1:Concorrentes inscritos como Individuais e de Valongo
  2:Paulos ou Ricardos que usam Gmail
  3:Informações dos atletas da equipa Turbulentos
  4:Lista dos escalões
  5:Gerar página HTML

6:Sair do Programa
```

Figura 4.1: Menu

```
1
Nome: VERA CRISTINA MOREIRA DELGADO

Nome: PAULO DOMINGUES

Nome: DULCE MOREDA
```

Figura 4.2: Resultados da Opção 1

```
2
Nome: PAULO DE CASTRO ROCHA
Prova :ULTRA TRAIL

Nome: PAULO SERRA
Prova :ULTRA TRAIL

Nome: PAULO VILAÇA
Prova :ULTRA TRAIL

Nome: PAULO DOMINGUES
Prova :ULTRA TRAIL

Nome: RICARDO JORGE DIAS OLIVEIRA
Prova :ULTRA TRAIL

Nome: RICARDO REIS
Prova :ULTRA TRAIL

Nome: PAULO FÉLIX
Prova :ULTRA TRAIL
```

Figura 4.3: Alguns resultados da Opção 2

3

Nome: JOÃO COSTA
Data: 04/04/70
Morada: RUA CAROLINA ROSA ALVES N°27 BRAGA
Email: JFSCOSTA@GMAIL.COM
Prova: ULTRA TRAIL
Escalao: M40
Equipa: TURBULENTOS

Nome: PAULO PIMENTEL TORRES
Data: 10/09/59
Morada: RUA COSTA SOARES, 39
Email: GERAL@VIEIRAFREITAS.PT
Prova: CORRIDA DA GEIRA
Escalao: SENIOR MASC
Equipa: TURBULENTOS

Nome: JOÃO PIMENTEL TORRES
Data: 11/09/88
Morada: RUA COSTA SOARES, 39
Email: GERAL@VIEIRAFREITAS.PT
Prova: CORRIDA DA GEIRA
Escalao: SENIOR MASC
Equipa: TURBULENTOS

Nome: VASCO MANUEL DE SEQUEIROS BARRETO MARTINS DE ARAÚJO
Data: 26/05/64

Figura 4.4: Alguns resultados da Opção 3

```
4  
  
Sem Escalao: 12  
  
F40: 3  
  
M40: 36  
  
M50: 12  
  
SENIOR FEM: 43  
  
SENIOR MASC: 175
```

Figura 4.5: Resultados da Opção 4

Equipas

- [INDIVUDUAL](#): 67
 - [OS TURBOLENTOS](#): 44
 - [EDV VIANA TRAIL](#): 13
 - [NAST](#): 12
 - [PRINT TEAM](#): 9
 - [CLUBE NAUTICO DE PONTE DE LIMA](#): 9
 - [ARRASTASSOLAS](#): 8
 - [XICO RUNNERS](#): 7
 - [MULTIPOWER GAIA TRAIL](#): 7
 - [CLUBE DE VETERANOS DO PORTO](#): 6
 - [COMPANHIA DO BAZÓFIAS](#): 6
 - [CLUBE ATLETISMO DE LAMAS](#): 5
 - [TUGAS NA ESTRADA](#): 5
 - [,](#): 5
 - [CLUBE SPIRIDON DE GAIA](#): 4
 - [OS BARRIGUITAS](#): 3
 - [OS CAGA TACOS RUNNING TEAN](#): 3
 - [BRACARA RUNNERS](#): 3
 - [SOBRE BUN LEE](#): 2
-

Figura 4.6: Página HTML Principal

INDIVIDUAL

- MARIO PIRES
- FRANCISCO NETO SILVA
- ARTUR BERNARDO
- VERA CRISTINA MOREIRA DELGADO
- JORGE YONG
- PAULO SERRA
- TIAGO JOSÉ CADIMA BORGES
- JORGE MANUEL MARTINS SILVA
- PAULO DOMINGUES
- DULCE MOREDA
- ANTÓNIO FERNANDES
- VERA CRISTINA MOREIRA DELGADO
- ANGELO SENRA
- PAULO JORGE
- ANTÓNIO FERNANDES
- VITOR MARQUES
- JOSE CARLINDO DUARTE CARVALHO
- NUNO LIMA
- SÉRGIO MARTINS
- NUNO LIMA
- PAULA CRISTINA OLIVEIRA
- NUNO COELHO SILVA
- ARTUR FERREIRA
- VASCO ORTIGÃO DE OLIVEIRA
- RICARDO JORGE DIAS OLIVEIRA
- NUNO CARVALHAL
- ANDRE SOUSA
- SUSANA DIAS
- MARCO TIAGO MARQUES RODRIGUES
- MARCO TIAGO MARQUES RODRIGUES
- ELISABETE COSTA
- ANTONIO LAFUENTE COELHO DE AZEVEDO
- RUI ANDRÉ SANTOS OLIVEIRA
- CARLOS FREITAS

Figura 4.7: Página HTML - Individuais

OS TURBOLENTOS

- JOÃO COSTA
- PAULO PIMENTEL TORRES
- JOÃO PIMENTEL TORRES
- VASCO MANUEL DE SEQUEIROS BARRETO MARTINS DE ARAÚJO
- CELESTE CRUZ
- JORGE LOURENÇO
- LUÍS MELO
- MARIO JORGE AORIMM
- LUIS B PEREIRA
- PAULO ANTUNES
- FERNANDO MANUEL TAVARES DA COSTA ANTUNES
- CARLOS ALBERTO GOMES TEIXEIRA
- MANEL ALBERTO LEITE DE SOUSA
- GIUSEPPE BALDASSARRE
- HELENA OLIVEIRA
- PAULO ANTUNES
- MARINHA PAULA FERNANDES
- FERNANDO MANUEL TAVARES DA COSTA ANTUNES
- CÁTIA RAQUEL MACEDO OLIVEIRA
- MANUEL COUTINHO MACHADO
- TIAGO DOMINGUES FRADA
- ANTÓNIO JOSÉ COSTA
- TIAGO DOMINGUES FRADA
- MARIO JORGE AORIMM
- JOSÉ FERREIRA LOUREIRO
- CARLOS MANUEL GONÇALVES VIEIRA FERNANDES
- JOSE ALBERTO MARTINS
- SANDRA MARIA FINSTERRA TRAVESSAS
- NUNO AMORIM SILVA
- LUIS REGO DA SILVA
- FRANCISCO ANTÓNIO DOS SANTOS COSTAS
- CARLOS BRITO
- ELISABETE COSTA
- ANTÓNIO FERREIRA

Figura 4.8: Página HTML - Equipa Turbulentos

EDV VIANA TRAIL

- SONIA PARENTE
- CÉLIA AZEVEDO
- GABRIEL FAGUNDES MEIRA
- JOSÉ DOMINGOS FERNANDES DOS SANTOS
- MARIA DO ROSÁRIO SANTOS
- NATALINA SANTOS
- JOSÉ MANUEL SANTOS FARIA
- ANDRÉ SALEIRO
- RUI COUTO
- ROSA LOBATO
- ANABELA LOBATO
- ALEXANDRO RIBEIRO
- ARLINDO ALVES RODRIGUES

Figura 4.9: Página HTML - Equipa EDV Viana Trail

PRINT TEAM

- RICARDO CAMPOS
- RUI CAMPOS
- CARLOS FIGUEIREDO
- ABILIO ARAUJO
- VERA COSTA
- EUGENIA SOUTO
- SUSANA FIGUEIREDO
- CLAUDIA ANGEIRA
- PEDRO AVELINO SILVA

Figura 4.10: Página HTML - Equipa Print Team

Capítulo 5

Conclusão

De um modo geral, podemos afirmar que os objetivos foram concluídos. Fomos capazes de responder a todas as questões presentes no enunciado, sendo importante realçar o interesse e empenho de ambos os elementos no decorrer do trabalho, para que tal fosse possível.

Este trabalho serviu para que os elementos do grupo aprimorassem as suas capacidades, no que ao processamento de um ficheiro diz respeito e, assim, perceber melhor o uso de expressões regulares e quando as utilizar.

Apêndice A

Código do Programa

Lista-se a seguir o código do programa que foi desenvolvido.

```
1 import re
2 import difflib
3
4 inscritos = []
5 inscricoes = []
6 dic = []
7
8 f = open("inscritos-form.json", encoding='UTF-8')
9
10 docJSON = f.read()
11
12 inscricao = re.findall(r'(".*":([^\[].*))', docJSON)
13
14
15 for elem in inscricao:
16     queroisto = re.search(r'"((\w*|\s|\\W*))"', elem[1])
17     inscritos.append(queroisto.group(1))
18
19
20 for val in range(0, len(inscritos), 7):
21     next_jump = val + 7
22     shot_values = inscritos[val:next_jump]
23     inscricoes.append(shot_values)
24
25
26 dados = {} #dados que contem um elemento do dicionario
27
28 for e in inscricoes:
29     dados['nome'] = e[0].upper()
30     dados['dataNasc'] = e[1].upper()
31     dados['morada'] = e[2].upper()
32     dados['email'] = e[3].upper()
33     dados['prova'] = e[4].upper()
34     dados['escalao'] = e[5].upper()
35     dados['equipa'] = e[6].upper()
36     dic.append(dados)
37     dados = {}
```



```

38
39
40 print("Boa Noite de Estudo!!!\n\n")
41 print(
42     "Escolha uma opcao:\n1: Pessoas Individuais de Valongo\n2: Paulos ou Ricardos
        \n3: Equipa Turbulentos\n4: Escaloes\n5: Site Bonitao LEGIT!!!\n6: Sair
        \ndo Programa\n")
43 opcao = input()
44
45 while opcao != '6':
46     if opcao == '1':
47         for i in dic:
48             if re.search(r'(?i)valongo', i['morada']) and re.search(r'(?i)individual',
                i['equipa']):
49                 nome = i['nome']
50                 print("Nome:_" + nome.upper() + "\n\n")
51
52     if opcao == '2':
53         for i in dic:
54             if (re.match(r'(?i)ricardo', i['nome']) or re.match(r'(?i)paulo', i['nome']
                )) and re.search(r'(?i)gmail', i['email']):
55                 nome = i['nome']
56                 prova = i['prova']
57                 print("Nome:_" + nome.upper() + "\n" + "Prova:_" + prova.upper() + "\n\n")
58
59     if opcao == '3':
60         for i in dic:
61             if re.search(r'(?i)turbulentos', i['equipa']):
62                 nome = i['nome']
63                 data = i['dataNasc']
64                 morada = i['morada']
65                 email = i['email']
66                 prova = i['prova']
67                 escalao = i['escalao']
68                 equipa = i['equipa']
69                 print("Nome:_" + nome.upper() + "\nData:_" + data.upper() + "\nMorada:_"
                    + morada.upper() + "\nEmail:_" + email.upper() + "\nProva:_" +
                    prova.upper()
70                     + "\nEscalao:_" + escalao.upper() + "\nEquipa:_" + equipa.upper()
                    + "\n\n")
71
72     if opcao == '4':
73         escaloes = []
74         escalaoAtletas = []
75
76         for i in dic:
77             escalao = i['escalao']
78             escalaoAtletas.append(escalao)
79
80         escaloes = escalaoAtletas.copy() # Copia Lista para se operar sem perder
            info
81         escaloes.sort() # Ordena Lista
82         escaloes = list(dict.fromkeys(escaloes)) # Remove Duplicados

```

```

83
84     for k in escaloes:
85         if k == "":
86             print("\nSem_Escalao:_ " + str(escalaoAtletas.count(k))) # Conta
87                                     Atletas Sem Escalao
88         else:
89             print("\n" + k + ":_ " + str(escalaoAtletas.count(k))) # Conta
90                                     Atletas por cada Escalao
91
92 if opcao == '5':
93     original = []
94     dup = []
95
96     for i in dic:
97         equipa = i['equipa']
98         original.append(equipa.upper())
99
100     dup = list(dict.fromkeys(original))
101
102     for index, k in enumerate(original):
103         for j in dup:
104             string1 = k
105             string2 = j
106             matches = difflib.SequenceMatcher(
107                 None, string1, string2).ratio()
108
109             if(matches >= 0.7):
110                 original[index] = j
111
112     equipasOficias = original.copy() #c pia da lista original j com os nomes
113                                     corretos
114     equipasOficias = list(dict.fromkeys(equipasOficias)) # Remove Duplicados
115
116     #atualiza o dicionario com os nomes iguais
117     for d in dic:
118         for e in equipasOficias:
119             string1 = d['equipa']
120             string2 = e.upper() #transforma em maiuscula para comparar com as
121                                     equipas do dic que est o em maiusculas
122             matches = difflib.SequenceMatcher(
123                 None, string1, string2).ratio()
124             if (matches >= 0.7):
125                 d['equipa'] = e
126
127     # troca os | e / por espa os nas equipas para ser mais facil criar o
128     # respectivo ficheiro html
129     for d in dic:
130         if (re.search(r'\\|', d['equipa'])):
131             d['equipa'] = re.sub(r'\\|', '\\_ ', d['equipa'])
132         elif (re.search(r'\\/ ', d['equipa'])):
133             d['equipa'] = re.sub(r'\\/ ', '\\_ ', d['equipa'])

```

```

132
133
134 equipasOficiasAtual2 = [] #con m as equipas sem | e /
135 for d in dic:
136     equipasOficiasAtual2.append(d['equipa'])
137
138 equipasOficiasAtual = list(dict.fromkeys(equipasOficiasAtual2))
139
140
141
142 dicEquipas = {} #vai conter o nome das equipas e o numero de atletas de cada
    equipa
143
144 for ele in equipasOficiasAtual:
145     somatorio = 0
146     somatorio = somatorio + equipasOficiasAtual2.count(ele) #soma o numeroa
        de elementos de equipas com o nome parecido
147     dicEquipas[ele] = somatorio #coloca o numeroa de elementos na equipa com
        nome na posicao 0 de ele
148 equipasOrdenadas = sorted(dicEquipas.items(), key=lambda x: x[1], reverse=
    True) #cria tuplos com os elementos do dicEquipas ordenados de forma
        decrecente pelo numero de atletas
149
150
151 f = open("equipas.html", "w")
152
153 f.write(' '<!DOCTYPE html>
154 <html>
155     <head meta charset="UTF-8">
156         <title>Equipas</title>
157     </head>
158     <body>
159         <h1>Equipas</h1>
160         <ul>
161             '''
162
163 #adiciona as equipas no ficheiro principal
164 for i in equipasOrdenadas: #tuplo
165     if (re.search(r'\s', i[0])): # cria os nomes dos ficheiros em html com _
        em vez de espa o
166         nomeEquipa2 = re.sub(r'\s',
167                               r'_ ',
168                               i[0])
169         f.write("<li><a href=\"\" + str(nomeEquipa2) + \".html\">\" + str(i[0])
170               + "</a>:_\" + str(i[1]) + "</li><br>")
171     else:
172         f.write("<li><a href=\"\" + str(i[0]) + \".html\">\" + str(i[0]) + "</a>
173               >:_\" + str(i[1]) + "</li><br>")
174 f.write(' '</ul>
175     </body>
176 </html>''')
177
178 f.close()

```

```

178     # cria os ficheiros das equipas e adiciona os inscritos de cada equipa
179     for i in equipasOficiasAtual:
180         if (re.search(r'\s', i)):
181             nomeEquipa = re.sub(r'\s',
182                                 r' - ',
183                                 i)
184             newfile = open(str(nomeEquipa) + ".html", "w")
185         else:
186             newfile = open(str(i) + ".html", "w")
187         newfile.write( '''<!DOCTYPE html>
188             <html>
189                 <head meta charset="UTF-8">
190                     <title>''' + str(i) + '''</title>
191                 </head>
192                 <body>
193                     <h1>''' + str(i) + '''</h1><br>
194                     <ul>
195                         '''
196         for d in dic:
197             if i == d['equipa']:
198                 newfile.write("<li>" + d['nome'] + "</li>")
199         newfile.write( '''</ul>
200             </body>
201             </html>''' )
202         newfile.close()
203
204     print(
205         "\n\nEscolha uma opcao:\n\n1: Pessoas Individuais de Valongo\n\n2: Paulos ou
206         Ricardos\n\n3: Equipa Turbulentos\n\n4: Escaloes\n\n5: Site Bonitao LEGIT
207         !!!\n\n6: Sair do Programa\n")
208     opcao = input()
209
210     print(" Ate uma proxima")

```
